

**SEQUENCIAMENTO DA PRODUÇÃO E
DESPACHO DE CAMINHÕES EM PLANTAS DE
CONCRETO**

MATEUS VILELA SOUZA

**SEQUENCIAMENTO DA PRODUÇÃO E
DESPACHO DE CAMINHÕES EM PLANTAS DE
CONCRETO**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais - Departamento de Ciência da Computação. como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: GERALDO ROBSON MATEUS

Belo Horizonte

Agosto de 2016

© 2016, Mateus Vilela Souza.
Todos os direitos reservados.

Souza, Mateus Vilela

S729s Sequenciamento da produção e despacho de
caminhões em plantas de concreto / Mateus Vilela
Souza. — Belo Horizonte, 2016
xxvii, 85 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais - Departamento de Ciência da
Computação.

Orientador: Geraldo Robson Mateus

1. Computação - Tese. 2. Ready mixed concrete
(RMC). 3. Programação linear inteira . 4. Logística.
5. Heurística. I. Título.

CDU 519.6*61(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Sequenciamento da produção e despacho de caminhões em plantas de concreto

MATEUS VILELA SOUZA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. GERALDO ROBSON MATEUS - Orientador
Departamento de Ciência da Computação - UFMG

PROF. ABÍLIO PEREIRA DE LUCENA FILHO
Faculdade de Economia e Administração - UFRJ

PROF. ANDRÉ GUSTAVO DOS SANTOS
Departamento de Informática - UFV

PROF. MARVIN GÓMEZ RAVETTI
Departamento de Engenharia de Produção - UFMG

Belo Horizonte, 04 de agosto de 2016.

Dedico este trabalho a minha mãe, que venceu a maior de todas as batalhas que se pode travar. Perto dessa batalha a minha foi quase insignificante. Obrigado por tudo, Maria de Fátima.

Agradecimentos

Inicialmente agradeço a UFMG. Em especial a meus professores da Engenharia de Produção que me mostraram esse caminho que estou trilhando hoje. Ao meu orientador, Geraldo Robson Mateus, por toda a paciência e por ter respondido todas as minhas inúmeras dúvidas. Também ao professor Alexandre, pela questão 3 da primeira prova de combinatória que me fez entender melhor o par Primal-Dual.

Agradeço à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pelo suporte financeiro que me possibilitou focar todos meus esforços nos meus estudos durante o período de mestrado.

Agradeço a todos meus amigos e familiares por todo o apoio e suporte que me deram. Ao meu falecido cachorro, que me acompanhou por toda essa trajetória. Ao meu pai e meu irmão, que sempre me apoiaram. À Larissa, por todo o apoio e suporte durante o mestrado.

Sempre me lembrarei dos amigos que fiz durante o mestrado no LaPO e que irei levar para o resto da minha vida.

*“Eu canto porque o instante existe
e a minha vida está completa.
Não sou alegre nem sou triste:
sou poeta.”*

(Cecília Meireles)

Resumo

A demanda por concreto com maior qualidade e as políticas de *Just-In-Time* das construções modernas vêm mudando o mercado de concreto na atualidade, principalmente em países com grandes potenciais de crescimento econômico como Brasil, Índia e China. O tipo de concreto RMC (Ready Mixed Concrete) tem ganhado muito espaço nesse mercado pela garantia de qualidade e grande flexibilidade de fornecimento. Isso permite a entrega de grandes quantidades em um curto intervalo de tempo.

O concreto RMC é produzido em plantas, carregado em veículos especializados que por sua vez realizarão o atendimento das demandas de um conjunto de construções. A determinação de onde e quando produzir de RMC, e qual deve ser a sequência de atendimento das demandas, tem sido realizada por profissionais experientes. No entanto, não tem sido utilizadas ferramentas que os auxiliem na tomada de decisão para obter a melhor solução possível.

Com isso, realizamos uma abrangente revisão na literatura e propomos uma nova abordagem sobre o problema de sequenciamento da produção e distribuição de RMC quando as plantas compartilham demandas entre si. Criamos um conjunto de instâncias, propomos um modelo de Programação Linear Inteiro Misto e uma heurística construtiva randômica para solucionar o problema. Os resultados mostram que o melhor desempenho é obtido pelo modelo iniciado por uma solução gerada pela heurística.

Palavras-chave: Ready Mixed Concrete (RMC), Programação Linear Inteira, Logística, Heurística.

Abstract

The demand for high quality concrete and a Just-In-Time policy in modern constructions have been changing the concrete market lately, mainly in countries with big potentials of economic growth like Brazil, India and China. The RMC (Ready Mixed Concrete) has had a great deal of market penetration because of its quality guarantee and large supply flexibility. It allows the supply of large quantities in a short time.

The RMC is produced in plants and transported by specialized vehicles to meet the demands of a set of constructions. Production planning and how the demands will be met are mostly performed by experienced employees. And, in general, they are elaborated without the use of a tool that assists them in the best possible solution.

In this context, we make a large literature review and propose a new approach to the scheduling production and distribution of RMC problem, considering that the plants can meet each demand alone or together. We also generate a set of instances, propose a Mixed-Integer Linear Programming model and a randomic constructive heuristic to solve this problem. The best approach is utilizing the model started with a solution from the heuristic.

Keywords: Ready Mixed Concrete (RMC), Mixed Integer Linear Programming, Logistics, Heuristic.

Lista de Figuras

5.1	Comparativo entre as três Buscas Locais	53
5.2	Comparativo entre as quatro Buscas Locais	54
5.3	Comparativo entre modelos que permite e não permite violação nas janelas de tempo	55
5.4	Comparativo do GAP obtido dos modelos que não permite violação nas janelas de tempo e o que permite	55
5.5	Comparativo entre modelos que não permite violações nas janelas de tempo iniciado com a solução da heurística e o modelo que não permite violações nas janelas de tempo	57
5.6	Comparativo do GAP obtido dos modelos que não permite violação nas janelas de tempo iniciado com a solução da heurística e o que não permite violação nas janelas de tempo	57

Lista de Tabelas

2.1	O tipo de veículo, o número de plantas e o método de solução em cada estudo	8
5.1	Valores dos dados das instâncias	50
5.2	Porcentagem de intâncias com soluções viáveis	52
A.1	Porcentagem de intâncias p-C101-P1C5 com soluções viáveis	67
A.2	Porcentagem de intâncias p-C101-P1C8 com soluções viáveis	68
A.3	Porcentagem de intâncias p-C101-P2C10 com soluções viáveis	68
A.4	Porcentagem de intâncias p-C101-P2C15 com soluções viáveis	68
A.5	Porcentagem de intâncias p-C101-P3C15 com soluções viáveis	68
A.6	Porcentagem de intâncias p-C101 com soluções viáveis	69
A.7	Porcentagem de intâncias p-R101-P1C5 com soluções viáveis	69
A.8	Porcentagem de intâncias p-R101-P1C8 com soluções viáveis	69
A.9	Porcentagem de intâncias p-R101-P2C10 com soluções viáveis	69
A.10	Porcentagem de intâncias p-R101-P2C15 com soluções viáveis	70
A.11	Porcentagem de intâncias p-R101-P3C15 com soluções viáveis	70
A.12	Porcentagem de intâncias p-R101 com soluções viáveis	70
A.13	Porcentagem de intâncias com soluções viáveis	71
B.1	Resultados da heurística executada cem vezes com os parâmetros Ve-2 Co-4 Pl-1 seguido de buscas locais para instâncias p-C101	74
B.2	Resultados da heurística executada cem vezes com os parâmetros Ve-2 Co-4 Pl-1 seguido de buscas locais para instâncias p-R101	75
C.1	Resultados do modelo que não permite violações nas janelas de tempo para instâncias p-C101	78
C.2	Resultados do modelo que não permite violações nas janelas de tempo para instâncias p-R101	79

C.3	Resultados do modelo que permite violações nas janelas de tempo para instâncias p-C101	81
C.4	Resultados do modelo que permite violações nas janelas de tempo para instâncias p-R101	82
C.5	Resultados do modelo que não permite violações nas janelas de tempo da construção iniciando com a solução da heurística para instâncias p-C101 .	84
C.6	Resultados do modelo que não permite violações nas janelas de tempo da construção iniciando com a solução da heurística para instâncias p-R101 .	85

Lista de Algoritmos

1	HEURÍSTICA CONSTRUTIVA	24
2	ADICIONA DEMANDA	25
3	ATRASADA DEMANDAS PARA ATENDER	28
4	ATRASADA DEMANDAS RECURSÃO	30
5	ADICIONA DEMANDA HORA FIM	32
6	VIABILIDADE 1	34
7	VIABILIDADE 2	37
8	PROCEDIMENTO DE VIABILIZAÇÃO	39
9	BUSCA LOCAL VEÍCULO	41
10	BUSCA LOCAL CONSTRUÇÃO	43
11	BUSCA LOCAL PLANTA	45

Lista de Siglas

LB *Local Branch*

RMC *Ready Mixed Concrete*

VLNS *Very Large Neighbourhood Search Strategy*

VNS *Variable Neighborhood Search*

Sumário

Agradecimentos	ix
Resumo	xiii
Abstract	xv
Lista de Figuras	xvii
Lista de Tabelas	xix
Lista de Siglas	xxiii
1 Introdução	1
1.1 Histórico do RMC	1
1.2 O problema de sequenciamento da produção e alocação do transporte de RMC	3
1.3 Problemas relacionados e complexidade	4
1.4 Objetivo	5
1.5 Estrutura do texto	5
2 Estado da arte	7
2.1 Trabalhos relacionados	7
2.2 Modelos matemáticos	10
2.3 Métodos de solução	11
3 Modelo matemático	13
3.1 Notação utilizada	13
3.2 Modelo	15
3.3 Valores de M	17
3.4 Modelo que permite violações nas janelas de tempo das construções . .	18

4	Heurística	21
4.1	Variáveis e Parâmetros utilizados pela heurística	21
4.2	Ordem de escolha dos veículos, construções e plantas	22
4.3	Heurística Construtiva	23
4.4	Procedimento AdicionaDemanda	24
4.4.1	Procedimento AtrasaDemandasParaAtender	28
4.4.2	Procedimento AtrasaDemandasRecursão	29
4.4.3	Procedimento AdicionaDemandaHoraFim	31
4.5	Algoritmos de Viabilidade	33
4.5.1	Algoritmo de Viabilidade 1	33
4.5.2	Algoritmo de Viabilidade 2	35
4.5.3	Execução dos procedimentos de Viabilização	38
4.6	Busca Local	39
4.6.1	Procedimento de Busca Local Veículos	40
4.6.2	Procedimento de Busca Local Construção	42
4.6.3	Procedimento de Busca Local Planta	44
5	Resultados	49
5.1	Descrição das instâncias	49
5.1.1	Descrição e criação das instâncias	49
5.1.2	Nomeclatura das instâncias	51
5.2	Ordem de escolha dos veículos, construções e plantas	51
5.2.1	Parâmetros do algoritmo para escolha dos veículos, construções e plantas	52
5.3	Buscas Locais	53
5.4	Modelos	54
5.5	Modelo iniciando com resultado da heurística	56
6	Conclusão	59
	Referências Bibliográficas	61
	Apêndice A Análise de uma execução da heurística construtiva seguida dos processos de viabilização	67
	Apêndice B Resultados da execução da heurística executada 100 vezes seguida das buscas locais	73
	Apêndice C Resultados da execução dos modelos	77

C.1	Testes do modelo que não permite violação das janelas de tempo das construções	77
C.2	Testes do modelo que permite violação das janelas de tempo das construções	80
C.3	Testes do modelo que não permite violação das janelas de tempo das construções iniciado com o valor da solução da heurística	83

Capítulo 1

Introdução

Neste capítulo apresentamos um histórico sobre o *Ready Mixed Concrete* (RMC) e como é realizado o sequenciamento de sua produção e seu transporte até as construções. Em seguida introduzimos o problema do sequenciamento da produção e alocação do transporte de RMC, problemas relacionados e uma análise de complexidade. Por fim, são apresentados os objetivos e a estrutura do nosso trabalho.

1.1 Histórico do RMC

O concreto é o material de construção mais utilizado no mundo [Aitcin, 2000], e é considerado indispensável para as necessidades modernas de infraestrutura, indústria e habitação [Damtoft et al., 2008]. A previsão é que sua demanda se mantenha no futuro [Imbabi et al., 2012], principalmente em países com grandes populações e com grandes potenciais de crescimento. A China é o país que mais consome concreto no mundo, seguido por Índia, Estados Unidos e Brasil [Armstrong, 2015].

A produção de concreto é realizada pela mistura de cimento, água e agregados, como areia e cascalho. Em muitos casos são adicionadas certas substâncias na mistura para retardar ou antecipar sua solidificação, ou mudar certas propriedades como sua cor e permeabilidade [Schmid, 2007; Schmid et al., 2009].

O RMC é um concreto produzido em plantas específicas e transportado para as construções através de veículos apropriados. Foi inicialmente introduzido na construção civil no início do século XX e desde então sua utilização vem aumentando cada vez mais [Feng et al., 2004]. Isso aconteceu pelas vantagens do RMC sobre o concreto misturado manualmente. Ele possui uma maior qualidade [Mayteekrieangkrai & Wongthatsanekorn, 2015], é considerado menos agressivo ambientalmente e necessita de um menor espaço nas construções para sua utilização [Lu & Lam, 2005, 2009].

Segundo Syverson [2008], nos últimos anos ocorreram cinco grandes mudanças na produção de RMC. São elas:

- Automatização da mistura do RMC nas plantas de produção;
- Aumento da capacidade dos veículos que transportam RMC;
- Avanços tecnológicos dos agregados a serem misturados ao concreto;
- Ganhos com a logística de distribuição de RMC ao centralizar as decisões de distribuição de várias plantas em apenas um local;
- Melhorias na sua aplicação através da utilização de bombas de concreto.

Apesar dos avanços, o sequenciamento da produção e alocação do transporte de RMC ainda é dependente de funcionários experientes [Lin et al., 2010; Liu et al., 2010; Yan et al., 2012; Maghrebi et al., 2014d; Maghrebi & Travis Waller, 2014] e é desempenhado de uma maneira ineficiente [Feng et al., 2004; Matsatsinis, 2004; Lin et al., 2010; Liu et al., 2010; Yan et al., 2012; Maghrebi et al., 2014d; Maghrebi & Travis Waller, 2014]. Feng et al. [2004]; Naso et al. [2004]; Maghrebi et al. [2013b,c, 2014b] relatam a falta de ferramentas para a automação destas tarefas. Ainda mais, Feng & Wu [2006]; Naso et al. [2007b] alegam que o setor de construção civil sofre por falta de soluções práticas para este problema.

Segundo Langevin & Riope [2005], em Lin et al. [2010], companhias de RMC possuem custos de manufatura similares. Com isso, o sequenciamento da produção e alocação do transporte de RMC mais eficientes são os principais fatores que podem possibilitar reduções de custos, podendo gerar maiores margens de lucro, ou uma redução do preço do RMC. Além dos ganhos financeiros, a criação de uma ferramenta para auxiliar a tomada de decisão diminuiria o estresse dos funcionários que as desempenham, reduziria os tempos de treinamento, e possibilitaria uma maior facilidade na substituição e realocação destes empregados [Durbin, 2003].

A resolução deste problema ainda pode auxiliar na solução de problemas semelhantes, como transporte de produtos agrícolas manufaturados [Naso et al., 2007a; Hanif & Holvoet, 2014], materiais perigosos [Naso et al., 2007a], e outros bens perecíveis, como o gelo [Syverson, 2008].

1.2 O problema de sequenciamento da produção e alocação do transporte de RMC

O problema de sequenciamento da produção e alocação do transporte de RMC pode ser caracterizado como um processo *Just-In-Time* [Tommelein & Li, 1999; Lu & Lam, 2009]. Isso é devido ao fato do RMC ter que ser consumido logo após sua produção e a impossibilidade de estoques por ser um produto perecível [Feng & Wu, 2000; Garcia et al., 2002; Schmid, 2007; Yan & Lai, 2007; Srichandum & Rujirayanyong, 2010; Liu et al., 2014]. Com isso, existe a restrição do produto ter que ser consumido até um certo limite de tempo após sua produção. Caso esse limite não seja respeitado, o concreto pode perder certas qualidades, podendo até se solidificar dentro do veículo transportador e causar grandes prejuízos [Asbach et al., 2009].

Este problema de sequenciamento e alocação de RMC pode ser dividido em cinco etapas: produção; carregamento do veículo transportador; transporte da planta até a construção; descarregamento na construção; retorno do veículo da construção até a planta [Garcia et al., 2002; Yan et al., 2008; Liu et al., 2014]. Com isso, cada veículo só atende uma construção em cada rota, ou viagem; apenas um veículo é carregado por vez em cada planta; e apenas um veículo pode descarregar por vez em cada construção [Schmid et al., 2010]. Ou seja, os carregamentos e descarregamentos em cada planta e construção não podem se sobrepor [Schmid et al., 2009].

Caso a demanda de uma construção seja superior à capacidade de um veículo, é preciso designar a essa construção um número de descarregamentos de veículos de RMC que satisfaça essa demanda [Garcia et al., 2002; Schmid, 2007; Schmid et al., 2009; Liu et al., 2014]. Esses descarregamentos têm que ser consecutivos e o intervalo entre eles não pode ultrapassar um limite máximo de tempo. Isso se deve ao fato do concreto não se solidificar homoganeamente na construção caso esse intervalo de tempo limite entre descarregamentos não seja respeitado [Asbach et al., 2009; Schmid et al., 2010; Kinable et al., 2014].

Para a resolução deste problema deve-se realizar a escolha de quais plantas irão atender quais demandas das construções, quais veículos irão realizar os atendimentos, e determinar os horários que as atividades serão realizadas. Os carregamentos e descarregamentos devem ainda respeitar as janelas de tempo de funcionamento das plantas e as janelas de tempo de recebimento de RMC das construções [Kinable et al., 2014].

Temos como premissas na abordagem desse estudo os seguintes fatos:

- Todos os caminhões possuem a mesma capacidade de transporte de RMC;
- Os tempos de deslocamento entre as construções e as plantas são pré-definidos;

- Os tempos de carregamento e descarregamento são também conhecidos;
- A planta pode produzir a quantidade que for solicitada, cabendo apenas respeitar o tempo de carregamento dos veículos;
- Cada planta possui um conjunto de veículos próprios que são utilizados somente por essa planta, que é uma premissa forte.

Vale resaltar que é muito usual os responsáveis pelas construções solicitarem que haja filas de veículos com RMC nas construções para evitar descontinuidades na entrega [Feng & Wu, 2006]. No entanto, filas de veículos nas construções são uma ineficiência do sistema. Como mencionado em Feng et al. [2004], todo tempo de espera do veículo na construção pode ser considerado como um custo de oportunidade e as filas forçam as plantas a possuírem e usarem mais veículos do que realmente precisam. Tendo isso em vista, escolhemos em nosso estudo por não permitir que veículos carregados com RMC fiquem ociosos.

Em nosso trabalho visamos minimizar os tempos que a última entrega de RMC é realizada em cada construção. Ou seja, visamos que o último descarregamento seja realizado o mais brevemente possível em cada construção. Vale ressaltar também que em nosso estudo todas as demandas das construções tem que ser atendidas para que a solução seja viável.

1.3 Problemas relacionados e complexidade

Várias abordagens sobre o problema de sequenciamento da produção e alocação do transporte de RMC já foram avaliadas. Schmid [2007]; Schmid et al. [2010] o consideraram como um problema de otimização combinatória e de logística altamente complexo. Maghrebi et al. [2015, 2016] o consideraram um desafiador problema de cadeia de suprimentos. Este problema é considerado como uma especialização do problema de roteamento de veículos por Asbach et al. [2009]; Maghrebi et al. [2013c] e já foi modelado como um problema de roteamento com restrições adicionais por Naso et al. [2007b]; Schmid [2007]; Chen et al. [2009]; Hertz et al. [2012]. Uma das grandes diferenças deste problema ao problema de roteamento de veículos é que seus veículos só podem visitar um cliente por rota [Zhang et al., 2011]. Kinable et al. [2014] mencionam a relação desse problema com o problema de sequenciamento de máquinas paralelas com janelas de tempo. Naso et al. [2007a]; Asbach et al. [2009]; Misir et al. [2011] o consideraram como um problema que interrelaciona os problemas de roteamento e de sequenciamento.

O problema de distribuição de RMC é NP-Difícil, [Naso et al., 2007b; Yan & Lai, 2007; Yan et al., 2008; Asbach et al., 2009; Maghrebi et al., 2013b, 2014d], para o qual foi feito uma redução do problema do caixeiro viajante em Asbach et al. [2009]. Uma solução ótima em tempo polinomial é considerada computacionalmente intratável [Asbach et al., 2009; Maghrebi et al., 2013b, 2014b,e, 2015, 2016].

1.4 Objetivo

Nosso trabalho tem como objetivo formular este problema em uma etapa intermediária entre os cenários nos quais as plantas de RMC competem entre si, não compartilhando demandas, e nos de completa integração entre as plantas de RMC, compartilhando demandas e veículos. Em nosso caso, os veículos ainda não são compartilhados entre as plantas de concreto, mas as demandas de RMC podem ser atendidas por mais de uma planta. Essa abordagem ainda não foi aplicada na literatura e acreditamos que venha a ser útil na integração de plantas de uma mesma empresa ou mesmo de diferentes empresas de RMC. A integração das empresas de RMC é uma tendência por ganhos de eficiência como é mencionado em Syverson [2008].

Consideramos que todas as demandas são atendidas e que devem ser respeitados: o tempo de vida do concreto, o tempo limite entre entregas consecutivas na mesma construção, as janelas de tempo das construções e das plantas. Os veículos podem atender mais de uma demanda e não consideramos tempos ociosos de caminhões carregados, tanto nas construções e nas plantas.

Os estudos com mais de uma planta já realizados utilizam veículos compartilhados entre as plantas. Estes veículos iniciam suas jornadas em certas garagens ou plantas, realizam os atendimentos de RMC e finalizam as jornadas na garagem ou planta de onde iniciaram. Por esse fato, os autores acima consideram este problema como um problema de roteamento de veículos. Isso não acontece em nosso caso pelo fato das rotas já serem predeterminadas e a tarefa que nos cabe é determinar se elas serão realizadas e quando, sendo um problema basicamente de sequenciamento.

1.5 Estrutura do texto

No capítulo 2 apresentamos uma revisão sobre os estudos já realizados sobre o tema. No capítulo 3 propomos um modelo em programação linear inteiro misto para a resolução do problema. No capítulo 4 discutimos a heurística proposta. No capítulo 5

descrevemos as intâncias criadas para o estudo e apresentamos os resultados obtidos. Finalizamos com o capítulo 6 que apresenta uma conclusão sobre o trabalho realizado.

Capítulo 2

Estado da arte

Inicialmente realizamos uma análise de artigos da literatura e identificamos para cada um os tipos de veículos empregados, o número de plantas abordadas e os métodos de solução utilizados. Em seguida apresentamos algumas características diferentes abordadas entre os estudos. Por fim, se tem duas sessões, uma contendo modelos matemáticos para resolver problemas de RMC e outra relatando as técnicas utilizadas nestes trabalhos.

2.1 Trabalhos relacionados

Foram realizados vários esforços para a resolução deste problema. Uma das dificuldades enfrentadas é o fato dos trabalhos levarem em consideração características diferentes do problema, [Kinable et al., 2014]. Alguns artigos levam em conta apenas o processo produtivo de uma planta de RMC, outros apenas o despacho dos veículos de RMC, e outros consideram os dois aspectos, [Yan et al., 2012].

A tabela 2.1 possui uma relação de trabalhos publicados em que a primeira coluna contém os trabalhos, a segunda coluna o tipo de frota de veículos que estes abordam, o número de plantas na terceira coluna e os métodos de solução na quarta coluna. Esses métodos são melhores explicados nas sessões 2.2 e 2.3.

Dentre estes trabalhos existem artigos que lidam com a possibilidade de contratar veículos terceirizados, [Naso et al., 2004, 2007b], casos em que cada veículo somente pode ser carregado por uma única planta, [Matsatsinis, 2004], ou em qualquer planta, [Feng & Wu, 2000; Garcia et al., 2002; Durbin, 2003; Naso et al., 2004; Silva et al., 2005; Naso et al., 2007a,b; Schmid, 2007; Asbach et al., 2009; Schmid et al., 2009, 2010; Misir et al., 2011; Yan et al., 2011; Hertz et al., 2012; Maghrebi et al., 2014a,b,d,e; Kinable et al., 2014; Narayanan et al., 2015; Maghrebi et al., 2016]. Outro atributo explorado é

Tabela 2.1. O tipo de veículo, o número de plantas e o método de solução em cada estudo

Estudo	Tipo de Veículo	Número de Plantas	Métodos de Solução
Feng & Wu [2000]	homogêneos	Uma planta	Algoritmo Genético
Zayed & Halpin [2001]		Uma planta	Simulação
Garcia et al. [2002]	homogêneos	Mais de uma	Algoritmo Genético
Durbin [2003]	homogêneos	Mais de uma	Busca Tabu
Lu et al. [2003]	heterogêneos	Uma planta	Simulação
Feng et al. [2004]	homogêneos	Uma planta	Simulação
Matsatsinis [2004]	heterogêneos	Mais de uma	Sistema de Suporte à Decisão
Naso et al. [2004]	homogêneos	Mais de uma	Algoritmo Evolucionário
Lu & Lam [2005]	heterogêneos	Uma planta	Simulação
Silva et al. [2005]	homogêneos	Mais de uma	Colônia de Formigas
Feng & Wu [2006]	homogêneos	Uma planta	Simulação
Naso et al. [2007a]	homogêneos	Mais de uma	Algoritmo Genético
Naso et al. [2007b]	homogêneos	Mais de uma	Algoritmo Genético
Schmid [2007]	heterogêneos	Mais de uma	<i>Local Branch</i>
Yan & Lai [2007]	homogêneos	Uma planta	Heurística
Yan et al. [2008]	homogêneos	Uma planta	<i>Branch-and-Bound</i>
Asbach et al. [2009]	heterogêneos	Mais de uma	Heurística
Lu & Lam [2009]	heterogêneos	Uma planta	Simulação
Schmid et al. [2009]	heterogêneos	Mais de uma	<i>Variable Neighborhood Search</i>
Lin et al. [2010]	homogêneos	Uma planta	Solução por solver
Liu et al. [2010]	homogêneos	Uma planta	Enxame de Partículas Discreta
Schmid et al. [2010]	heterogêneos	Mais de uma	<i>Variable Neighborhood Search</i>
Srichandum & Rujirayanyong [2010]	homogêneos	Uma planta	Colônia de Abelhas
Misir et al. [2011]	heterogêneos	Mais de uma	Hyper-Heurística
Park et al. [2011]	homogêneos	Uma planta	Simulação
Yan et al. [2011]	homogêneos	Mais de uma	Enxame de Partículas Discreta
Zhang et al. [2011]	heterogêneos	Uma planta	Algoritmo Genético
Hertz et al. [2012]	heterogêneos	Mais de uma	Algoritmo Hierárquico
Shi & Wang [2012]	homogêneos	Uma planta	Algoritmo Genético
Yan et al. [2012]	homogêneos	Uma planta	Heurística
Maghrebi et al. [2013a]	heterogêneos	Mais de uma	Geração de Colunas
Maghrebi et al. [2013b]		Uma planta	Aprendizado de Máquinas
Maghrebi et al. [2013c]		Mais de uma	Aprendizado de Máquinas
Liu et al. [2014]	heterogêneos	Uma planta	Algoritmo Genético
Maghrebi et al. [2014a]	heterogêneos	Mais de uma	Decomposição de Benders
Maghrebi et al. [2014b]	heterogêneos	Mais de uma	Geração de Colunas
Maghrebi et al. [2014c]	homogêneos	Uma planta	Nova Formulação
Maghrebi et al. [2014d]	heterogêneos	Mais de uma	Algoritmo Genético
Maghrebi et al. [2014e]	heterogêneos	Uma planta	Algoritmo Genético
Maghrebi & Travis Waller [2014]		Mais de uma	Aprendizado de Máquinas
Kinable et al. [2014]	heterogêneos	Mais de uma	Heurística
Cheng & Hoc-Tran [2015]	homogêneos	Uma planta	Simulação
Maghrebi et al. [2015]	heterogêneos	Uma planta	Aprendizado de Máquinas
Mayteekriangkrai & Wongthatsanekorn [2015]	homogêneos	Uma planta	Colônia de Abelhas
Narayanan et al. [2015]	homogêneos	Mais de uma	Relaxação Lagrangeana
Maghrebi et al. [2016]	homogêneos	Mais de uma	Geração de Colunas

o roteamento de bombas de concreto, [Matsatsinis, 2004; Zhang et al., 2011; Liu et al., 2014] e veículos com acessórios, como bombas e correias transportadoras, [Schmid et al., 2009, 2010].

Em Kinable et al. [2014] é mencionada a variação nas definições do problema entre os trabalhos já realizados. Um exemplo são as funções objetivo, que variam de trabalho para trabalho. Por exemplo: Garcia et al. [2002]; Kinable et al. [2014] visam atender o maior número de ordens na janela de tempo fornecida e penalizam quando isso não acontece. Feng & Wu [2000]; Feng et al. [2004]; Feng & Wu [2006]; Liu et al. [2010]; Srichandum & Rujiranyong [2010]; Shi & Wang [2012]; Cheng & Hoc-Tran [2015]; Mayteekriengkrai & Wongthatsanekorn [2015] visam minimizar os tempos ociosos dos veículos de RMC nas construções. Lin et al. [2010] têm como foco minimizar o tempo total de atraso das entregas de RMC. Naso et al. [2004, 2007b] são multiobjetivos, visando minimizar vários aspectos do sistema como: o tempo de espera por RMC nas construções, a distância percorrida pelos veículos e as entregas de RMC realizadas por terceiros. Yan et al. [2008]; Liu et al. [2014] minimizam os custos operacionais do sistema de despacho de RMC. Schmid [2007]; Maghrebi et al. [2014e] têm como foco minimizar as distâncias percorridas pelos veículos que transportam o RMC. Schmid et al. [2010] além de minimizarem as distâncias percorridas pelos veículos, penalizam na função objetivo os atrasos de entregas consecutivas na mesma construção.

Além destes atributos, alguns estudos levam em consideração outras características. Por exemplo: Yan & Lai [2007] contabilizam horas extra. Misir et al. [2011] penalizam caso a construção não seja atendida pela planta de produção de RMC desejada e caso ocorra desperdício de RMC. Schmid et al. [2009]; Maghrebi et al. [2013a] permitem que clientes não sejam atendidos. Lin et al. [2010] levam em conta demandas incertas. Garcia et al. [2002] assumem uma frota de veículos de transporte de RMC ilimitada. Yan et al. [2011] levam em conta tempos de deslocamento dos veículos de RMC estocásticos. Yan et al. [2012]; Hanif & Holvoet [2014]; Cheng & Hoc-Tran [2015] consideram quebras de equipamentos. Hanif & Holvoet [2014] consideram congestionamentos. Srichandum & Rujiranyong [2010]; Mayteekriengkrai & Wongthatsanekorn [2015] consideram que os veículos possuem tempos diferentes para executarem os percursos da planta até a construção e da construção até a planta. Srichandum & Rujiranyong [2010]; Park et al. [2011] consideram tipos diferentes de descarregamentos de concreto, o que faz com que o tempo de descarregamento e o intervalo máximo entre descarregamentos consecutivos sejam diferentes entre as construções. Yan & Lai [2007]; Yan et al. [2008] consideram o tempo de descarregamento diferente entre as construções. Yan et al. [2008] consideram construções com diferentes tempos máximos entre descarregamentos.

Esses fatos fazem a comparação entre os métodos serem extremamente difíceis, ou até mesmo inviáveis.

2.2 Modelos matemáticos

Vários modelos matemáticos foram propostos, [Durbin, 2003; Feng et al., 2004; Naso et al., 2007b; Schmid, 2007; Yan & Lai, 2007; Yan et al., 2008; Asbach et al., 2009; Lin et al., 2010; Zhang et al., 2011; Yan et al., 2011; Hertz et al., 2012; Yan et al., 2012; Liu et al., 2014; Maghrebi et al., 2014c,d; Kinable et al., 2014; Narayanan et al., 2015]. Um dos mais importantes na área é o proposto por Asbach et al. [2009]. Eles propuseram um modelo matemático mais simples ao dividir as plantas de produção de RMC e as construções em sub-plantas e em sub-construções. Com isso, a formulação não precisa mais lidar com clientes que demandam mais de um veículo de RMC, diminuindo assim o número de restrições. Contudo, o número de variáveis de decisão aumenta.

O modelo proposto em Asbach et al. [2009] foi utilizado, ou levou ao desenvolvimento de outros modelos matemáticos, nos estudos de Zhang et al. [2011]; Maghrebi et al. [2013a,c, 2014a,b,d,e]; Narayanan et al. [2015]; Maghrebi et al. [2016].

Naso et al. [2007b] apresentam um modelo não linear que é estendido em Naso et al. [2007a] incluindo novas variáveis e restrições. Durbin [2003] utiliza uma formulação de rede espaço-tempo. Schmid [2007] propõe um modelo fluxo *multi-commodity* baseado no modelo de Durbin [2003]. Yan & Lai [2007]; Yan et al. [2008, 2011, 2012] abordaram como um problema inteiro misto de fluxo em rede com restrições adicionais (*side constraints*) e utilizaram a modelagem espaço-tempo. Liu et al. [2014] propõem um modelo de programação inteiro misto também baseado em fluxo.

Hertz et al. [2012] propõem um modelo linear inteiro misto e uma heurística que decompõe o problema de RMC em dois modelos de programação inteiro misto. Maghrebi et al. [2014d] propõem dois modelos matemáticos para o problema de produção e transporte de RMC, sendo que um considera janelas de tempo e o outro não. Maghrebi et al. [2014c] propõem um modelo inteiro misto que foca nos clientes e não nas entregas individuais de RMC. Kinable et al. [2014] apresentam três modelos matemáticos. Um baseado em programação inteira mista derivado dos modelos de roteamento de veículos e modelos de sequenciamento de máquinas paralelas. Os outros dois são extensões desse primeiro modelo, pois incorporam a exigência que um mesmo veículo faça mais de uma entrega a um certo cliente. Lin et al. [2010] apresentam um modelo linear inteiro misto abordando o problema como sendo um *job-shop* com recirculação, considerando os veículos como máquinas e as construções como *jobs*.

2.3 Métodos de solução

Yan & Lai [2007] abordam o problema com uma heurística de decomposição. Eles resolvem o problema relaxado, fixam algumas variáveis, e em seguida resolvem o problema simplificado que não contempla as variáveis fixadas. Yan et al. [2008] introduzem um método *Branch-and-Bound* e realizam cortes no espaço de soluções.

A técnica de geração de colunas foi aplicada por Maghrebi et al. [2013a, 2014b, 2016]. Maghrebi et al. [2014a] utilizam a decomposição de Benders. Narayanan et al. [2015] abordam o problema utilizando a relaxação Lagrangeana.

Schmid [2007] soluciona o modelo criado com um *Local Branch* (LB) baseado em *Variable Neighborhood Search* (VNS). Isso foi realizado para aumentar a velocidade de convergência do método. Também apresenta inequações válidas e cortes no espaço de soluções. Schmid et al. [2009] implementa um modelo de fluxo *multi-commodity* inteiro junto com um VNS. Schmid et al. [2010] utilizam o VNS para gerar soluções iniciais com a finalidade de serem inseridas em um modelo inteiro misto. Também realizam um procedimento que fixa algumas variáveis do modelo através de *Very Large Neighbourhood Search Strategy* (VLNS).

Durbin [2003] aplica uma técnica de otimização de fluxos em rede de custo mínimo e uma busca Tabu para resolver o problema. Asbach et al. [2009] aplicam buscas locais e métodos de decomposição para a obtenção de soluções. Kinable et al. [2014] apresentam duas heurísticas: uma heurística construtiva juntamente com uma busca local; e uma heurística que utiliza um modelo inteiro misto.

Hanif & Holvoet [2014] utilizam o *Delegate Multi-Agents System* que é um mecanismo de coordenação bio-inspirado. Feng & Wu [2000]; Garcia et al. [2002]; Feng & Wu [2006]; Naso et al. [2007a,b]; Zhang et al. [2011]; Shi & Wang [2012]; Liu et al. [2014]; Maghrebi et al. [2014b,e] utilizam algoritmos genéticos para solucionar o problema. Naso et al. [2007a,b] utilizam um algoritmo genético juntamente com heurísticas construtivas. Silva et al. [2005] utilizam um algoritmo genético juntamente com um algoritmo de colônia de formigas. Heurísticas baseadas em colônia de abelhas foram utilizadas por Srichandum & Rujiranyong [2010]; Mayteekrieangkrai & Wongthatsanekorn [2015]. Liu et al. [2010]; Yan et al. [2011] optaram por uma abordagem via enxame de partículas discretas (*Discrete Particle Swarm Optimization*). Naso et al. [2004] propõem uma meta-heurística baseada em um algoritmo evolucionário híbrido combinado com uma heurística construtiva.

Técnicas de simulação são abordadas por Zayed & Halpin [2001]; Lu et al. [2003]; Feng et al. [2004]; Lu & Lam [2005]; Feng & Wu [2006]; Lu & Lam [2009]; Park et al. [2011]; Cheng & Hoc-Tran [2015]. Sendo que Feng et al. [2004]; Lu & Lam [2005]; Feng

& Wu [2006] também utilizam algoritmos genéticos.

Técnicas de Aprendizado de Máquina (*Machine Learning*) para o problema são aplicadas nos estudos de Maghrebi et al. [2013c]; Maghrebi & Travis Waller [2014]; Maghrebi et al. [2015].

Misir et al. [2011] propõem uma hyper-heurística e Matsatsinis [2004] apresenta uma abordagem para projetar um Sistema de Suporte à Decisão visando o problema de RMC.

Em nosso estudo utilizamos uma heurística construtiva randomizada e um modelo linear inteiro misto. Abordamos o problema com mais de uma planta e veículos com capacidades homogêneas. Pelas características mencionadas em 1.4 não observamos uma correlação do nosso trabalho com os outros, reforçando a percepção de Kinable et al. [2014].

Capítulo 3

Modelo matemático

Neste capítulo propomos dois modelos de programação linear inteiro misto para a resolução do problema de sequenciamento da produção de RMC e despacho de veículos. Primeiramente mostramos o modelo que não permite adiantamento ou atrasos nas janelas de tempo das construções. Em seguida mostramos as alterações no primeiro modelo para aceitar adiantamentos ou atrasos nas janelas de tempo das construções para se obter o segundo modelo. Após a demonstração de cada modelo se tem como foi estipulado os diversos Big M usados nos modelos. A notação utilizada é apresentada antes da apresentação dos modelos.

3.1 Notação utilizada

Os conjuntos, parâmetros e variáveis utilizados ao longo do trabalho estão a seguir.

Conjuntos

C	Conjunto de construções c
D	Conjunto de demandas i
D^c	Conjunto de demandas i da construção $c \in C$
P	Conjunto de plantas p
V	Conjunto de veículos v
V^p	Conjunto de veículos associados à planta $p \in P$

Parâmetros

Tem_{cp}	Tempo para deslocar entre a construção $c \in C$ e a planta $p \in P$
Tem_{pc}	Tempo para deslocar entre a planta $p \in P$ e a construção $c \in C$
$Desc_{vi}$	Tempo para descarregar RMC do veículo $v \in V$ para suprir a demanda $i \in D$
$Carr_p$	Tempo para carregar RMC em um veículo na planta p
TVC	Tempo de vida do concreto
θ_c	Tempo máximo entre descarregamentos consecutivos na construção $c \in C$
$S_{vii'}$	Tempo de percurso do veículo v pertencente a planta p entre i até i' , sendo $i \in D^c$ e $i' \in D^{c'}$. Considera o intervalo de tempo que o veículo leva para realizar o deslocamento da construção c até a planta p , mais o tempo para carregar o veículo v na planta p e o tempo de deslocamento da planta p até a construção c'
$MaxD^c$	Última demanda i a ser atendida na construção c
M_{vi}^1	Primeira constante de valor elevado que é definida de acordo com o veículo v e a demanda i
M_{pc}^2	Segunda constante de valor elevado que é definida de acordo com a construção c e a planta p
$M_{vii'}^3$	Terceira constante de valor elevado que é definida de acordo com o veículo v e as demandas i e i'
$M_{vii'}^4$	Quarta constante de valor elevado que é definida de acordo com o veículo v e as demandas i e i'
M_c^5	Quinta constante de valor elevado que é definida de acordo com a construção c
$M_{v'i}^6$	Sexta constante de valor elevado que é definida de acordo com o veículo v' e a demanda i
M_p^7	Sétima constante de valor elevado que é definida de acordo com a planta p
M_{cp}^8	Oitava constante de valor elevado que é definida de acordo com a construção c e a planta p

Variáveis

z_c	Último horário que o último veículo termina o descarregamento na construção c
t_{vi}	Horário que inicia o descarregamento do veículo $v \in V$ para atender a demanda $i \in D^c$
tp_{vi}	Horário que inicia o carregamento do veículo $v \in V^p$ na planta $p \in P$ para atender a demanda $i \in D^c$
α_{vi}	Possuirá o valor 1 se o veículo $v \in V$ é alocado para atender a demanda $i \in D^c$, 0 caso contrário
$\beta_{vii'}$	Possuirá o valor 1 se veículo $v \in V$ for alocado para atender a demanda $i \in D^c$ antes de atender a $i' \in D^{c'}$, 0 caso contrário
$\beta'_{pvi'}$	Possuirá o valor 1 se planta $p \in P$ for alocada para atender a demanda $i \in D^c$ antes de atender a $i' \in D^{c'}$, 0 caso contrário

3.2 Modelo

O modelo em programação linear inteiro misto proposto para o problema de sequenciamento e alocação do transporte de RMC foi baseado nos trabalhos de Manne [1960]; Ravetti [2007], e associa os veículos à máquinas e as construções à tarefas nos problemas clássicos de sequenciamento.

$$MIN : \sum_c^C z_c \quad (3.1)$$

Sujeito a:

$$\sum_v^V \alpha_{vi} = 1 \quad \forall i \in D^c, \forall c \in C \quad (3.2)$$

$$z_c \geq t_{vi} + Desc_{vi} - M_{vi}^1(1 - \alpha_{vi}) \quad \forall i \in D^c, \forall c \in C, \forall v \in V \quad (3.3)$$

$$t_{vi} \geq -M_{pc}^2(1 - \alpha_{vi}) + tp_{vi} + Carr_p + Tem_{pc} \quad \forall i \in D^c, \forall c \in C, \forall v \in V^p, \forall p \in P \quad (3.4)$$

$$t_{vi} \leq M_{pc}^2(1 - \alpha_{vi}) + tp_{vi} + Carr_p + Tem_{pc} \quad \forall i \in D^c, \forall c \in C, \forall v \in V^p, \forall p \in P \quad (3.5)$$

$$M_{vii'}^3(1 - \alpha_{vi}) + M_{vii'}^3(1 - \alpha_{vi'}) + M_{vii'}^3(1 - \beta_{vii'}) + t_{vi'} \geq t_{vi} + Desc_{vi} + S_{vii'} \quad (3.6)$$

$$\forall v \in V, \forall i \in D^c, \forall i' \in D^c, \forall c \in C, i' \neq i$$

$$M_{vii'}^4(1 - \alpha_{vi}) + M_{vii'}^4(1 - \alpha_{vi'}) + M_{vii'}^4 \cdot \beta_{vii'} + t_{vi} \geq t_{vi'} + Desc_{vi'} + S_{vii'} \quad (3.7)$$

$$\forall v \in V, \forall i \in D^c, \forall i' \in D^c, \forall c \in C, i' \neq i$$

$$M_{vii'}^3(1 - \alpha_{vi}) + M_{vii'}^3(1 - \alpha_{vi'}) + M_{vii'}^3(1 - \beta_{vii'}) + t_{vi'} \geq t_{vi} + Desc_{vi} + S_{vii'} \quad (3.8)$$

$$\forall v \in V, \forall i \in D^c, \forall c \in C, \forall i' \in D^c, \forall c' \in C, c' \neq c$$

$$M_{vii'}^4(1 - \alpha_{vi}) + M_{vii'}^4(1 - \alpha_{vi'}) + M_{vii'}^4 \cdot \beta_{vii'} + t_{vi} \geq t_{vi'} + Desc_{vi'} + S_{vii'} \quad (3.9)$$

$$\forall v \in V, \forall i \in D^c, \forall c \in C, \forall i' \in D^c, \forall c' \in C, c' \neq c$$

$$M_c^5(1 - \alpha_{vi}) + t_{vi} + Desc_{vi} + \theta_c \geq -M_c^5(1 - \alpha_{v'i+1}) + t_{v'i+1} \quad (3.10)$$

$$\forall i \in \{D^c - MaxD^c\}, \forall c \in C, \forall v \in V, \forall v' \in V$$

$$M_{v'i}^6(1 - \alpha_{vi+1}) + t_{vi+1} \geq -M_{v'i}^6(1 - \alpha_{v'i}) + t_{v'i} + Desc_{v'i} \quad (3.11)$$

$$\forall i \in \{D^c - MaxD^c\}, \forall c \in C, \forall v \in V, \forall v' \in V$$

$$M_p^7(1 - \alpha_{vi}) + M_p^7(1 - \alpha_{v'i'}) + M_p^7(1 - \beta'_{pii'}) + tp_{v'i'} \geq tp_{vi} + Carr_p \quad (3.12)$$

$$\forall v \in V^p, \forall v' \in V^p, \forall p \in P, \forall i \in D, \forall i' \in D, i \neq i'$$

$$M_p^7(1 - \alpha_{vi}) + M_p^7(1 - \alpha_{v'i'}) + M_p^7 \cdot \beta'_{pii'} + tp_{vi} \geq tp_{v'i'} + Carr_p \quad (3.13)$$

$$\forall v \in V^p, \forall v' \in V^p, \forall p \in P, \forall i \in D, \forall i' \in D, i \neq i'$$

$$TVC \geq t_{vi} - tp_{vi} - M_{cp}^8(1 - \alpha_{vi}) \quad \forall i \in D^c, \forall c \in C, \forall v \in V^p, \forall p \in P \quad (3.14)$$

$$T_{MIN_c} * \alpha_{vi} \leq t_{vi} \leq T_{MAX_c} * \alpha_{vi} \quad \forall i \in D^c, \forall c \in C, \forall v \in V \quad (3.15)$$

$$T_{MIN_p} * \alpha_{vi} \leq tp_{vi} \leq T_{MAX_p} * \alpha_{vi} \quad \forall i \in D^c, \forall c \in C, \forall v \in V^p, \forall p \in P \quad (3.16)$$

$$\alpha_{vi} \in \{0, 1\} \quad \forall v \in V, \forall i \in D^c, \forall c \in C \quad (3.17)$$

$$\beta_{vii'} \in \{0, 1\} \quad \forall v \in V, \forall i \in D^c, \forall c \in C, \forall i' \in D^c, \forall c' \in C \quad (3.18)$$

$$\beta'_{pii'} \in \{0, 1\} \quad \forall p \in P, \forall i \in D^c, \forall c \in C, \forall i' \in D^c, \forall c' \in C \quad (3.19)$$

A equação 3.1 representa a função objetivo que minimiza o tempo de funcionamento das construções.

As restrições 3.2 garantem que toda demanda seja atendida por um único veículo. Pelas restrições 3.3 o último horário de funcionamento das construções são pelo menos o horário de término do último descarregamento na construção. As restrições 3.4 e 3.5 implicam que, caso o veículo v atenda a demanda i , o horário de início do descarregamento para atender i na construção seja igual a soma do horário do início do carregamento na planta mais o tempo para carregar o veículo na planta e o tempo do deslocamento do veículo da planta até a construção. As restrições 3.6, 3.7, 3.8, 3.9, 3.12 e 3.13 são baseadas nas restrições propostas em Manne [1960]. As restrições 3.6

e 3.7 fazem com que o início dos descarregamentos de um veículo na mesma construção tenham a diferença mínima de tempo referente ao tempo de descarregamento na construção mais o tempo do deslocamento de ir da construção até a planta, carregar o veículo e retornar até a construção. As restrições 3.8 e 3.9 fazem com que o início dos descarregamentos de um veículo em construções diferentes tenham uma diferença mínima de tempo referente ao tempo de descarregamento da demanda na primeira construção, mais o tempo de deslocar da primeira construção até a planta, carregar o veículo e deslocar para a segunda construção. As restrições 3.10 e 3.11 impõem que demandas sucessivas em uma mesma construção sejam atendidas respeitando os intervalos máximos e mínimos que elas devem ter entre si. Sendo o intervalo mínimo o tempo para se descarregar um veículo na construção c . As restrições 3.12 e 3.13 garantem que o início dos carregamentos em uma mesma planta tenham uma diferença mínima de tempo referente ao tempo que leva para carregar o veículo v na planta p . As restrições 3.14 impõem que o horário entre o carregamento do concreto em uma planta e o seu descarregamento em uma construção não seja superior ao tempo de vida do concreto. As restrições 3.15 limitam o horário em que é permitido iniciar o descarregamento da demanda i na construção c com o veículo v . As restrições 3.16 limitam o horário em que é permitido iniciar o carregamento da demanda i na planta p com o veículo v . As restrições 3.17, 3.18 e 3.19 exigem a integralidade das variáveis binárias α_{vi} , $\beta_{vii'}$ e $\beta'_{pii'}$.

3.3 Valores de M

Os valores dos Big M foram estipulados para que as restrições fiquem mais justas e assim facilitar o atendimento das condições de otimalidade.

$$M_{vi}^1 = T_{MAX_c} + Desc_{vi} \quad i \in D^c \quad (3.20)$$

$$M_{pc}^2 = T_{MAX_p} + Carr_p + Tem_{pc} \quad (3.21)$$

$$M_{vii'}^3 = T_{MAX_c} + Desc_{vi} + S_{vii'} \quad i \in D^c \quad (3.22)$$

$$M_{vii'}^4 = T_{MAX_{c'}} + Desc_{vii'} + S_{vii'} \quad i' \in D^{c'} \quad (3.23)$$

$$M_c^5 = T_{MAX_c} \quad (3.24)$$

$$M_{v'i}^6 = T_{MAX_c} + Desc_{v'i} \quad i \in D^c \quad (3.25)$$

$$M_p^7 = T_{MAX_p} + Carr_p \quad (3.26)$$

$$M_{cp}^8 = T_{MAX_c} - T_{MIN_p} \quad (3.27)$$

Em 3.20 estipula-se que M^1 é o horário máximo permitido para iniciar um descarregamento na construção c mais o tempo para descarregar a demanda i pelo veículo v . Em 3.21 estipula-se que M^2 é definido como o horário máximo que é possível iniciar um carregamento na planta p , mais o tempo para carregar um veículo na planta e deslocá-lo da planta p até a construção c . Em 3.22 o M^3 corresponde ao horário máximo permitido para iniciar um descarregamento na construção c , mais o tempo para descarregar a demanda i pelo veículo v e o valor do parâmetro $S_{vii'}$. Em 3.23 estipula M^4 da mesma maneira que o M^3 foi estipulado, porém o tempo para descarregar a demanda i pelo veículo v é agora para a demanda i' e em vez do parâmetro $S_{vii'}$ se tem o $S_{v'i}$. Em 3.24 é estipulado M^5 como o horário máximo permitido para iniciar um descarregamento na construção c . Em 3.25 o M^6 é calculado como o horário máximo permitido para iniciar um descarregamento na construção c mais o tempo para descarregar o veículo v' visando atender a demanda i . Em 3.26 é estipulado o M^7 como o horário máximo que é possível iniciar um carregamento na planta p mais o tempo para carregar um veículo nessa planta. Em 3.27, o M^8_{cp} corresponde a maior diferença entre o último horário que é permitido iniciar um descarregamento na construção c e o primeiro horário que é permitido iniciar um carregamento de um veículo na planta p .

3.4 Modelo que permite violações nas janelas de tempo das construções

Para o modelo permitir alterações nas janelas de tempo é necessário a adição de alguns parâmetros, variáveis e de realizar algumas modificações no modelo.

Precisamos adicionar os parâmetros e variáveis que se seguem.

Parâmetros

- M^C Penalidade caso inicie o descarregamento de um veículo em uma construção fora de sua janela de tempo
- $MaxAC_c$ Limite de tempo permitido para adiantar o horário mínimo da janela de tempo da construção c
- $MaxPC_c$ Limite de tempo permitido para postergar o horário máximo da janela de tempo da construção c

Variáveis

- γac_{vi} Possuirá o valor 1 caso o atendimento da demanda i pelo veículo v tiver que ser realizado em um horário anterior a janela de tempo da construção de i , 0 caso contrário
- γpc_{vi} Possuirá o valor 1 caso o atendimento da demanda i pelo veículo v tiver que ser realizado em um horário posterior a janela de tempo da construção de i , 0 caso contrário

A função objetivo 3.1 passa a ser a 3.28.

$$MIN : \quad \sum_c^C z_c + \sum_v^v \sum_c^C \sum_i^{D^c} (M^C * \gamma ac_{vi} + M^C * \gamma pc_{vi}) \quad (3.28)$$

As restrições 3.15 são substituídas por 3.29. Dessa forma, as janelas das construções podem ser adiantadas no máximo de $MaxAC_c$, ou atrasadas de no máximo $MaxPC_c$.

$$T_{MIN_c} * \alpha_{vi} - MaxAC_c * \gamma ac_{vi} \leq t_{vi} \leq T_{MAX_c} * \alpha_{vi} + MaxPC_c * \gamma pc_{vi} \quad (3.29)$$

$$\forall v \in V, \forall c \in C, \forall i \in D^c$$

As restrições 3.30 e 3.31 fazem uma associação entre as variáveis γac_{vi} e γpc_{vi} com o α_{vi} . As duas primeiras variáveis somente poderão ser ativas caso α_{vi} também seja unitário. As restrições 3.32 e 3.33 garantem a integralidade das variáveis binárias γac_c e γpc_c .

$$\gamma ac_{vi} \leq \alpha_{vi} \quad \forall v \in V, c \in C, \forall i \in D^c \quad (3.30)$$

$$\gamma pc_{vi} \leq \alpha_{vi} \quad \forall v \in V, c \in C, \forall i \in D^c \quad (3.31)$$

$$\gamma ac_{vi} \in \{0, 1\} \quad \forall v \in V, c \in C, \forall i \in D^c \quad (3.32)$$

$$\gamma pc_{vi} \in \{0, 1\} \quad \forall v \in V, c \in C, \forall i \in D^c \quad (3.33)$$

A escolha dos valores para M tem influência na convergência do algoritmo e nos tempos que se pode adiantar ou atrasar as janelas de tempo das construções. Devem ser elevados para não restringir os valores que as variáveis podem assumir. No entanto,

devem ser o mais justo possível. Neste sentido, foram escolhidos os seguintes valores para cada um dos M usados no modelo:

$$M_{vi}^1 = T_{MAX_c} + MaxPC_c + Desc_{vi} \quad i \in D^c \quad (3.34)$$

$$M_{pc}^2 = T_{MAX_p} + Carr_p + Tem_{pc} \quad (3.35)$$

$$M_{vii'}^3 = T_{MAX_c} + MaxPC_c + Desc_{vi} + S_{vii'} \quad i \in D^c \quad (3.36)$$

$$M_{vii'}^4 = T_{MAX_{c'}} + MaxPC_{c'} + Desc_{vii'} + S_{vii'} \quad i' \in D^{c'} \quad (3.37)$$

$$M_c^5 = T_{MAX_c} + MaxPC_c \quad (3.38)$$

$$M_{v'i}^6 = T_{MAX_c} + MaxPC_c + Desc_{v'i} \quad i \in D^c \quad (3.39)$$

$$M_p^7 = T_{MAX_p} + Carr_p \quad (3.40)$$

$$M_{cp}^8 = T_{MAX_c} + MaxPC_c - T_{MIN_p} \quad (3.41)$$

Capítulo 4

Heurística

Este capítulo descreve a heurística proposta para solucionar o problema do sequenciamento de produção e alocação do transporte de RMC. Inicialmente são definidos algumas variáveis e parâmetros utilizados pela heurística. Em seguida descrevemos os procedimentos utilizados pela heurística visando a seleção das construções, das plantas e dos veículos. Explicamos a estrutura geral da heurística como um todo, o procedimento de alocação da produção de RMC para definir qual veículo será carregado em qual planta e qual construção esse veículo irá atender, e o procedimento construtivo para a confecção do sequenciamento de produção e distribuição de RMC. Considerando que nem sempre uma solução viável (solução com todas as demandas atendidas) é obtida, apresentamos os procedimentos que buscam viabilizar os sequenciamentos criados no procedimento construtivo. Em sequência apresentamos as buscas locais criadas.

4.1 Variáveis e Parâmetros utilizados pela heurística

A execução dos procedimentos da heurística demandam a definição de algumas variáveis e parâmetros, como:

Para todas as construções/plantas:

- **Analisado** : Indica se a construção/planta já foi analisada pelo algoritmo. Terá valor 1 caso positivo, e 0 caso contrário.

Para todas as plantas:

- **ConstruçãoPodeReceber** : Menor horário que a planta pode começar o carregamento de um veículo para suprir a demanda da construção corrente no algoritmo.

Para todas as demandas:

- *AnaliseRemoção* : Representa se a tarefa referente ao atendimento a uma demanda já foi removida do sequenciamento durante um procedimento do algoritmo. Será 1 caso positivo, e 0 caso contrário.

Um vetor *Sequenciamento* armazenará o sequenciamento da produção e alocação do transporte de RMC obtida. Esse vetor contém as seguintes informações:

- *NívelInviabilidade* : Possui o número de demandas não atendidas pelo sequenciamento. Este número é considerado como nível de inviabilidade da solução. Caso seu valor seja nulo, a solução é viável (todas as demandas atendidas). Caso seja maior que 0, ela é inviável (possui o número de demandas não atendidas);
- *SomaHoráriosFinais* : Representa a soma dos horários que os últimos descarregamentos são realizados nas construções do sequenciamento.

Os procedimentos *AdicionaDemanda()*, *AtrasaDemandaParaAtender()* e *AtrasaDemandasRecursão()*, utilizam os seguintes valores durante seus procedimentos de execução:

- *InícioC* : Horário de início do descarregamento do concreto na construção corrente;
- *InícioP* : Horário de início do carregamento de concreto na planta corrente;
- *HoraFim* : Horário mínimo para que a demanda atendida anteriormente à demanda corrente termine seu descarregamento na construção;
- *SitAloc* : Representa se a demanda corrente foi ou não atendida;
- *InterTempo* : Salto de tempo que o algoritmo irá utilizar para incrementar o horário *InícioP*.

4.2 Ordem de escolha dos veículos, construções e plantas

Para a execução da heurística é necessário estabelecer uma ordem de prioridade para a escolha dos veículos, das construções e das plantas. Tendo isso em vista, foram definidas algumas ordens de prioridades:

- Escolha dos veículos

Ve-1 - Veículo com menor número de demandas atendidas

Ve-2 - Veículo com maior número de demandas atendidas

Ve-3 - Escolha pela ordem de numeração dos veículos (o veículo com menor índice é escolhido primeiro)

Ve-4 - Seleção aleatória dos veículos

- Escolha das construções

Co-1 - Construção com menor valor de $((T_{MAX_c} - T_{MIN_c})/D^c)$, ou seja, com janela de tempo reduzida e demanda elevada

Co-2 - Construção com menor janela de tempo $(T_{MAX_c} - T_{MIN_c})$

Co-3 - Construção com menor horário mínimo de janela de tempo (T_{MIN_c})

Co-4 - Seleção aleatória das construções

- Escolha das plantas

Pl-1 - Planta mais próxima da construção em análise

Pl-2 - Planta com menor número de carregamentos

Pl-3 - Planta com maior número de carregamentos

Pl-4 - Seleção aleatória das plantas

4.3 Heurística Construtiva

A heurística construtiva realiza o procedimento de construção do sequenciamento inicial. Para isso, ela irá percorrer todas as construções seguindo a ordenação adotada, atendendo todas as demandas da primeira até a última. Isso é realizado com o objetivo de atender as demandas, visando respeitar o intervalo máximo entre atendimentos consecutivos na mesma construção (θ_c). Esse procedimento retorna o sequenciamento

obtido. O pseudocódigo está no Algoritmo 1.

Algoritmo 1: HEURÍSTICA CONSTRUTIVA

Entrada: P, C
Saída: S

```

1 início
2    $S = \emptyset$ ;
3   para  $c \leftarrow \{1, \dots, C\}$  faça
4      $Atende \leftarrow 1$ ;
5      $i \leftarrow 1$ ;
6     enquanto  $Atende = 1 \wedge i \leq D^c$  faça
7        $Atende \leftarrow \text{AdicionaDemanda}(i, P, S)$ ;
8        $i \leftarrow i + 1$ ;
9     fim
10  fim
11  S.CalculaNívelInviabilidade( );
12  retorna  $S$ ;
13 fim
```

O procedimento recebe como entrada as construções C e as plantas P , e tem como saída o sequenciamento S .

Na linha 2 é iniciado o vetor S como vazio. As linhas 3 a 10 são executadas para todas as construções. A linha 4 inicia $Atende$ com o valor 1, isso é realizado para entrar no loop das linhas 6 a 9. A linha 5 inicia i com a primeira demanda a ser atendida na construção c . As linhas 6 a 9 são executadas enquanto for viável atender as demandas de c . Na linha 7, a demanda i é alocada seguindo o procedimento $\text{AdicionaDemanda}(i, P, S)$, seção 4.4, que atribui a $Atende$ o valor 1 caso a demanda i seja atendida, e -1 caso contrário. Na linha 8 passa-se para a próxima demanda a ser atendida. Nas linhas 11 e 12 é calculado o nível de inviabilidade e é retornado o sequenciamento obtido.

4.4 Procedimento AdicionaDemanda

Esse procedimento é responsável pelo atendimento de uma demanda. Explicitamos em uma seção separada por sua importância, complexidade e alta utilização, tanto na heurística construtiva, quanto nos procedimentos de viabilização e nas buscas locais.

Primeiramente tenta atender uma demanda $i \in D^c$ considerando as demandas já atendidas na construção c . Caso não consiga realizar o atendimento da demanda i , será verificado a possibilidade de atendê-la atrasando o atendimento das demandas anteriores na construção c . Caso seja possível atender a demanda, é retornado o novo sequenciamento e o valor 1. Caso contrário, é retornado o sequenciamento inicial e o valor -1, indicando inviabilidade para atender à demanda. O pseudocódigo está no

Algoritmo 2.

Algoritmo 2: ADICIONADEMANDA

```

Entrada:  $i \in D^c, P, S$ 
Saída:  $S, valor$ 
1 início
2   AtribuiSequenciamento( $S0$ );
3   InicializaHorárioPodeAtender( );
4    $P$ .OrdenaPlantas( );
5   para  $p \leftarrow \{1, \dots, P\}$  faça
6      $p$ .OrdenaVeículos( );
7     para  $v \leftarrow \{1, \dots, V^p\}$  faça
8        $InícioP \leftarrow p$ .DeterminaHorárioInicialPlanta( $c$ );
9        $InícioC \leftarrow InícioP + Carr_p + Tem_{pc}$ ;
10      enquanto  $InícioP \leq T_{MAX_p} \wedge InícioC \leq T_{MAX_c}$  faça
11        se  $p$ .DispPla( $InícioP$ ) = 1  $\wedge$   $v$ .DispVei( $InícioP, c$ ) = 1 então
12          se  $c$ .DispConst( $InícioC$ ) = 1 então
13            Aloca( $InícioP, InícioC, v, i$ );
14            retorna  $S, 1$ ; ( demanda atendida )
15          fim
16          se  $c$ .DispConstCasoAtrasa( $InícioC$ ) = 1 então
17            se  $p$ .ConstruçãoPodeReceber >  $InícioC$  então
18               $p$ .ConstruçãoPodeReceber  $\leftarrow InícioC$ ;
19            fim
20             $p$ .MarcaPlantaPodeAtendeSeAtrasar( );
21             $InícioP \leftarrow T_{MAX_p} + InterTempo$ ;
22          fim
23        fim
24         $InícioP \leftarrow InícioP + InterTempo$ ;
25         $InícioC \leftarrow InícioP + Carr_p + Tem_{pc}$ ;
26      fim
27    fim
28  fim
29  se  $PlantasPodemAtenderSeAtrasar( ) = 1$  então
30     $HoraFim \leftarrow P$ .MenorHoraEmpRec( ) -  $\theta_c + InterTempo$ ;
31    se  $AtrasaDemandasParaAtender(i, HoraFim, S) = 1$  então
32      retorna  $S, 1$ ; ( demanda atendida )
33    fim
34  fim
35  RetornaSequenciamento( $S0$ );
36  retorna  $S, -1$ ; ( demanda não atendida )
37 fim

```

O procedimento recebe como entrada a demanda $i \in D^c$, as plantas P e o sequenciamento S . E tem como saída o sequenciamento S e um valor indicativo de viabilidade do atendimento da demanda (valor 1) ou inviabilidade do atendimento da demanda (valor -1).

Na linha 2 é armazenado o sequenciamento inicial em $S0$. Na linha 3 é atribuído as variáveis *ConstruçãoPodeReceber* de todas as plantas o maior valor possível. A

linha 4 ordena as plantas de acordo com um dos critérios apresentados em 4.2. Isso é realizado para que a escolha da planta para atender a demanda i respeite essa ordem. As linhas 5 a 28 realizam o procedimento de atender a demanda i considerando as demandas já atendidas na construção c . As linhas 12 a 15 tentam atender a demanda i , sendo que as linhas 16 a 22 atribuem o menor horário para que a planta corrente possa atender a demanda desrespeitando o intervalo máximo entre atendimentos de demandas consecutivas na mesma construção (θ_c). Caso a demanda i não seja atendida, as linhas 29 a 34 tentam atendê-la atrasando o atendimento de demandas anteriores a i na construção c . Se a demanda for atendida, é retornado o novo sequenciamento S e o valor 1. Caso não, é retornado o sequenciamento inicial e o valor -1.

Entre as linhas 5 e 28, é realizado para todos os veículos de todas as plantas a busca pelo atendimento da demanda i . Para cada um dos veículos é atribuído o primeiro possível horário inicial de carregamento na planta p em $InícioP$, linha 8, e de descarregamento na construção c em $InícioC$, linha 9. Em seguida é atualizado o $InícioP$ com uma fração de tempo $InterTempo$, linha 24, e é atualizado o $InícioC$, linha 25. Isso é realizado até conseguir atender a demanda i , ou ultrapassar o horário máximo de funcionamento da planta (T_{MAX_p}), ou ultrapassar o horário máximo de funcionamento da construção (T_{MAX_c}).

Mais detalhadamente, as iterações das linhas 5 a 28 e 7 a 27 são realizados para todas as plantas e todos os veículos respectivamente. Eles consideram as ordenações realizadas nas linhas 4 e 6. Na linha 8 o horário inicial de carregamento na planta p é alocado a $InícioP$ em que o veículo v pode realizar a primeira tentativa de atendimento da demanda i . A função $p.DeterminaHorárioInicialPlanta(c)$ leva em conta os horários de início de funcionamento da planta p e da construção c para determinar o $InícioP$. A linha 9 atualiza o horário que iniciará o atendimento da demanda na construção c no $InícioC$. Esse tempo é a soma do horário de início na planta contido em $InícioP$, do tempo de carregar o veículo na planta p ($Carr_p$) e do tempo de deslocamento do veículo da planta p até a construção c (Tem_{pc}). Nas linhas 10 a 26 é executado o procedimento de atender a demanda i da construção c com o veículo v validando todas as possibilidades de horários que não violem os horários de funcionamento da construção c e da planta p . A linha 11 verifica se a planta p e o veículo v podem atender a demanda i com as respectivas funções: $p.DispPla(InícioP)$ e $v.DispVei(InícioP, c)$. Caso seja possível, o procedimento das linhas 12 a 23 é executado. A linha 12 verifica se o atendimento da demanda i não violará nenhuma restrição da construção c . Tais restrições são referentes ao intervalo máximo entre o atendimento de demandas consecutivas, a janela de tempo da construção e a sobreposição desse atendimento a um outro já realizado. Essa verificação é realizada através do procedimento $c.DispConst(InícioC)$ que retorna 1 caso

seja possível atender à demanda. Caso possível, a linha 13 irá realizar o atendimento da demanda i na construção c e a alocação dessa atividade, tanto no veículo v quanto na planta p . Isso é realizado pelo procedimento $Aloca(InícioP, InícioC, v, i)$. A linha 14 retorna o novo sequenciamento e o valor 1 caso i seja atendida. A linha 16 verifica se é possível atender a demanda i pelo veículo v e a planta p caso o atendimento das demandas anteriormente atendidas em c sejam atrasadas. Isso é realizado pelo procedimento $c.DispConstCasoAtrasa(InícioC)$. Tal fato acontece quando houver a possibilidade de atender a demanda i em um horário posterior as demandas já atendidas em c , mas violando o intervalo máximo entre o atendimento de duas demandas consecutivas na construção c . Caso seja possível, os procedimentos das linhas 17 a 21 são executados. A linha 17 verifica se o horário armazenado para atender a demanda i na construção c com a planta p , o $p.ConstruçãoPodeReceber$, é maior que $InícioC$ no momento. Caso seja maior, o valor de $p.ConstruçãoPodeReceber$ é atualizado com o valor de $InícioC$ na linha 18. Isso é realizado para garantir o menor horário em que a planta p pode atender a demanda i caso atrase o atendimento das demandas anteriormente atendidas em c . A linha 20 assinala que a planta p pode atender a demanda i caso seja atrasado o atendimento de demandas anteriores a i em c . Isso é realizado pelo procedimento $p.MarcaPlantaAtendeSeAtrasar()$. A linha 21 atualiza o valor de $InícioP$ para o valor máximo de funcionamento da planta p mais $InterTempo$, isso é realizado para sair do *loop* das linhas 10 a 27. A linha 24 realiza o incremento de $InícioP$ com o valor $InterTempo$ e a linha 25 atualiza o $InícioC$ pelo mesmo procedimento realizado na linha 9.

Nas linhas 29 a 34 é avaliado o atendimento da demanda i atrasando o atendimento das demandas já atendidas na construção c . Na linha 29, o procedimento $PlantasPodemAtenderSeAtrasar()$ verifica a existência de alguma planta que pode atender a demanda i caso as demandas já atendidas em c sejam atrasadas. Esse procedimento retorna 1 caso positivo. A linha 30 aloca o horário mínimo em que a primeira demanda, que é atendida anteriormente a i , deva terminar seu atendimento em $HoraFim$. Essa variável recebe o valor da soma de: o menor horário dentre as $ConstruçãoPodeReceber$ das plantas, que é retornado pela função $P.MenorHoraEmpRec()$; o valor negativo do intervalo entre o atendimento que duas demandas consecutivas na construção c têm que possuir (θ_c); e o valor de $InterTempo$. Na linha 31 é realizado o procedimento $AtrasaDemandasParaAtender(i, HoraFim)$. Essa avalia o atraso do atendimento das demandas anteriores a demanda i , e o posterior atendimento dessa demanda i . Caso a demanda i seja atendida com esse procedimento, um novo sequenciamento S e o valor 1 é retornado na linha 32. Caso contrário, na linha 36 é retornado o sequenciamento inicial, que é restabelecido na linha 35, e o valor -1.

4.4.1 Procedimento AtrasaDemandasParaAtender

O procedimento busca realizar o atendimento da demanda $i \in D^c$ ao atrasar o atendimento das demandas já atendidas em c . Isso é realizado enquanto houver a possibilidade de atendimento de uma demanda i com o atraso do atendimento das demandas anteriores à ela. O pseudocódigo está no Algoritmo 3.

Algoritmo 3: ATRASADEMANDASPARAATENDER

Entrada: $i \in D^c$, $HoraFim$, S
Saída: S , *valor*

```

1 início
2   AtribuiSequenciamento( $S_0$ );
3    $SitAloc \leftarrow 0$ ;
4   enquanto  $SitAloc = 0$  faça
5      $SitAloc \leftarrow$  AtrasaDemandasRecursão( $i$ ,  $HoraFim$ ,  $S$ );
6     se  $SitAloc = 1$  então
7        $SitAloc \leftarrow$  AdicionaDemandaHoraFim( $i$ ,  $HoraFim$ ,  $S$ ,  $NovaHoraFim$ );
8       se  $SitAloc = 0$  então
9          $HoraFim \leftarrow NovaHoraFim$ ;
10      fim
11     fim
12  fim
13  se  $SitAloc = 1$  então
14    retorna  $S$ , 1; ( demanda atendida )
15  fim
16  RetornaSequenciamento( $S_0$ );
17  retorna  $S$ , -1; ( demanda não pode ser atendida )
18 fim
```

O procedimento recebe como entrada a demanda $i \in D^c$, o horário $HoraFim$ e o sequenciamento S , e tem como saída o sequenciamento S e um valor indicando o atendimento de i (valor 1) ou não (valor -1).

Na linha 2 é armazenado o sequenciamento inicial em S_0 . Na linha 3 é atribuído o valor 0 a $SitAloc$. O valor 0 representa que a demanda i ainda não foi atendida, mas ela ainda pode ser atendida caso atrase o atendimento das demandas anteriores na construção c . O *loop* das linhas 4 a 12 é realizado enquanto a demanda i ainda puder ser atendida caso atrase o atendimento das demandas anteriores. Sairá do *loop* quando a demanda i for atendida e o $SitAloc$ receber o valor 1, ou quando certificar que não consegue atender a demanda i mesmo atrasando o atendimento das demandas anteriores na construção e o $SitAloc$ recebe o valor -1. A linha 5 executa $AtrasaDemandasRecursão(i, HoraFim)$ que atrasa o atendimento das demandas anteriores a i na construção c . Essa função faz com que a demanda $i - 1$ tenha seu término de atendimento na construção c de, pelo menos, o horário de $HoraFim$. A linha 6 verifica se a demanda $i - 1$ foi atendida em um horário igual ou após o $HoraFim$. Isso

é realizado verificando se o *SitAloc* recebeu o valor 1 na linha 5. Caso positivo, na linha 7 a função *AdicionaDemandaHoraFim*(i , *HoraFim*, *NovaHoraFim*) tenta atender a demanda i partindo de um horário início de carregamento na planta do qual o término do descarregamento na construção c seja, no mínimo, o horário de *HoraFim*. Se a função retornar o valor 0, a demanda i ainda pode ser atendida caso atrase o atendimento das demandas anteriores em c . Esse procedimento ainda atribuirá a variável *NovaHoraFim* o horário mínimo em que a primeira demanda atendida anteriormente a i deve ter seu término de descarregamento. A linha 8 verifica se a demanda i não foi atendida, mas ainda é possível atendê-la caso as demandas anteriores a i em c tenham seus atendimentos atrasados. Caso entre na condição da linha 8, a linha 9 atualizará o *HoraFim* com o *NovaHoraFim* fornecido pela linha 7. A linha 13 verifica se a demanda i foi atendida. Caso tenha sido, a linha 14 retorna o novo sequenciamento S e o valor 1. Caso não, a linha 17 retorna o sequenciamento inicial, restabelecido na linha 16, e o valor -1.

4.4.2 Procedimento Atrasa Demandas Recursão

O procedimento tenta atender as demandas anteriores a $i \in D^c$ de forma que a demanda $i - 1$ termine seu descarregamento no mínimo no horário de *HoraFim*. Caso não seja possível atender a demanda $i - 1$ terminando em *HoraFim*, entrará na recursão para alocar as demandas anteriores a $i - 1$ em um horário que permita o atendimento da demanda $i - 1$ terminando no mínimo em *HoraFim*. A recursão irá terminar quando realocar o atendimento das demandas anteriores a i respeitando o horário mínimo de atendimento *HoraFim*, ou certificar que isso não é possível. O pseudocódigo está no Algoritmo 4.

Algoritmo 4: ATRASADEMANDASRECURSÃO

Entrada: $i \in D^c$, $HoraFim$, S
Saída: S , $valor$

```

1 início
2   AtribuiSequenciamento( $S0$ );
3    $c.DeletaAtividade(i - 1)$ ;
4    $SitAloc \leftarrow AdicionaDemandaHoraFim(i - 1, HoraFim, S, NovaHoraFim)$ ;
5   se  $SitAloc = 0$  então
6      $HoraFim \leftarrow NovaHoraFim$ ;
7     enquanto  $SitAloc = 0$  faça
8        $SitAloc \leftarrow AtrasaDemandasRecursão(i - 1, HoraFim, S)$ ;
9       se  $SitAloc = 1$  então
10         $SitAloc \leftarrow AdicionaDemandaHoraFim(i, HoraFim, S, NovaHoraFim)$ ;
11        se  $SitAloc = 0$  então
12           $HoraFim \leftarrow NovaHoraFim$ ;
13        fim
14      fim
15    fim
16  fim
17  se  $SitAloc = 1$  então
18    retorna  $S, 1$ ; ( demanda atendida )
19  fim
20  RetornaSequenciamento( $S0$ );
21  retorna  $S, -1$ ; ( demanda não atendida )
22 fim
```

O procedimento recebe como entrada a demanda $i \in D^c$, o horário $HoraFim$ e o sequenciamento S , e tem como saída o sequenciamento S e um valor indicando se foi possível atender a i (valor 1) ou se não é possível (valor -1).

A linha 2 armazena o sequenciamento inicial em $S0$. A linha 3 desaloca o atendimento da demanda $i - 1$. Na linha 4 é realizado a tentativa de atender a demanda $i - 1$ com o horário de término do descarregamento de no mínimo $HoraFim$. A linha 5 verifica se $SitAloc$ possui o valor 0. Ou seja, não conseguiu atender a demanda $i - 1$, mas ainda é possível atendê-la, caso atrasar o atendimento das demandas anteriores a ela em c . Caso positivo, será executado as linhas 6 a 16. A linha 6 atualiza o valor de $HoraFim$ com o valor de $NovaHoraFim$ obtido na linha 4. As linhas 7 a 15 serão executadas enquanto puder atender a demanda $i - 1$ após atrasar o atendimento das demandas anteriores a ela, ou seja, enquanto o $SitAloc$ for 0. A linha 8 chama a própria função recursivamente, mas para a demanda $i - 1$ e com o $HoraFim$ que foi atualizado. A linha 9 verifica se a demanda $i - 1$ foi atendida com o término de seu descarregamento em c igual ou superior ao horário $HoraFim$ ao verificar se o $SitAloc$ é 1. Caso tenha obtido êxito, a linha 10 realiza o atendimento da demanda i com tempo final de descarregamento de no mínimo $HoraFim$. A linha 11 verifica se o $SitAloc$ é 0, ou seja, se a demanda i não foi atendida, mas ainda pode ser atendida caso atrase

o atendimento das demandas anteriores a ela em c . A linha 12 atualiza o $HoraFim$ com o $NovaHoraFim$ fornecido pela linha 10 caso $SitAloc$ for 0. Sairá do *loop* das linhas 7 a 15 quando ocorrer o atendimento da demanda i , ou certificar que não é possível atendê-la. Será certificado que não é possível atender a demanda i quando a função $AdicionaDemandaHoraFim(i - 1, HoraFim, S, NovaHoraFim)$ da linha 4 ou 10, ou $AtrasaDemandasRecursão(i - 1, HoraFim, S)$ da linha 8, retornarem -1. A linha 17 verifica se a demanda foi atendida. Caso atendida, a linha 18 retornará o novo sequenciamento S e o valor 1. Caso contrário, a linha 21 retorna o sequenciamento inicial, restabelecido na linha 20, e o valor -1.

4.4.3 Procedimento AdicionaDemandaHoraFim

Esse procedimento é muito parecido com o $AdicionaDemanda(i, P, S)$, mas ao contrário desse, ele inicializa o horário de início do carregamento na planta de acordo com o horário $HoraFim$ fornecido. Esse procedimento também não tenta atender a demanda $i \in D^c$ atrasando o atendimento das demandas anteriores à ela na construção c . Ao invés disso, ele retorna o menor horário que a demanda anterior a i em c tem que acabar o descarregamento para que a demanda i possa ser atendida. Esse horário é retornado através de $NovaHoraFim$.

O procedimento possui três situações de término. No primeiro, o atendimento da demanda i é realizado e retorna-se o novo sequenciamento S e o valor 1. No segundo, é sinalizado que ainda é possível atender a demanda i ao atrasar o atendimento das outras demandas de c , retornando o sequenciamento inicial e o valor 0. No terceiro, é certificado que não é possível atender a i mesmo atrasando o atendimento das outras demandas de c , retornando o sequenciamento inicial e o valor -1. O pseudocódigo está no Algoritmo 5.

Algoritmo 5: ADICIONADEMANDAHORAFIM

Entrada: $i \in D^c$, $HoraFim$, S
Saída: S , $NovaHoraFim$, $valor$

```

1 início
2   AtribuiSequenciamento( $S0$ );
3   InicializaHorárioPodeAtender( );
4    $P$ .OrdenaPlantas( );
5   para  $p \leftarrow \{1, \dots, P\}$  faça
6      $p$ .OrdenaVeículos( );
7     para  $v \leftarrow \{1, \dots, V^p\}$  faça
8        $InícioP \leftarrow$  DeterminaHorárioInicialPlanta( $HoraFim$ );
9        $InícioC \leftarrow$   $InícioP + Carr_p + Tem_{pc}$ ;
10      enquanto  $InícioP \leq T_{MAX_p} \wedge InícioC \leq T_{MAX_c}$  faça
11        se  $p.DispPla(InícioP) = 1 \wedge v.DispVei(InícioP, c) = 1$  então
12          se  $c.DispConst(InícioC) = 1$  então
13             $c.Aloca(InícioP, InícioC, v, i)$ ;
14            retorna  $S, 1$ ; ( demanda atendida )
15          fim
16          se  $c.DispConstCasoAtrasa(InícioC) = 1$  então
17            se  $p.ConstruçãoPodeReceber > InícioC$  então
18               $p.ConstruçãoPodeReceber \leftarrow InícioC$ ;
19            fim
20            MarcaPlantaPodeAtendeSeAtrasar( $p$ );
21             $InícioP \leftarrow T_{MAX_p} + InterTempo$ ;
22          fim
23        fim
24      fim
25       $InícioP \leftarrow InícioP + InterTempo$ ;
26       $InícioC \leftarrow InícioP + Carr_p + Tem_{pc}$ ;
27    fim
28  fim
29  se  $P.PlantasPodemAtenderSeAtrasar( ) = 1$  então
30     $NovaHoraFim \leftarrow P.MenorHoraEmpRec( ) - \theta_c + InterTempo$ ;
31    RetornaSequenciamento( $S0$ );
32    retorna  $S, 0$ ; ( demanda ainda pode ser atendida )
33  fim
34  RetornaSequenciamento( $S0$ );
35  retorna  $S, -1$ ; ( demanda não atendida )
36 fim
```

O procedimento recebe como entrada a demanda $i \in D^c$, o horário $HoraFim$ e o sequenciamento S , e tem como saída o sequenciamento S , o horário $NovaHoraFim$ e um valor que será 1 se a demanda i for atendida, 0 caso ainda seja possível atender i , ou -1 no caso de certificar que não é possível atender i .

Uma das diferenças entre esse procedimento e o $AdicionaDemanda(i, P, S)$ está na linha 8 da qual é inicializado o valor de $InícioP$. O procedimento $DeterminaHorárioInicialPlanta(HoraFim)$ inicializa o $InícioP$ com um horário que permite finalizar o descarregamento da demanda i no horário $HoraFim$ na construção

c. A outra diferença é a partir da linha 29 na qual não se tenta mais atender a demanda *i*. A linha 30 atualiza *NovaHoraFim* que será retornado. A linha 32 retorna o sequenciamento inicial, restabelecido na linha 31, e o valor 0 representando que a demanda ainda pode ser atendida caso atrasar o atendimento das demandas anteriores a *i* em *c*. A linha 35 é executada caso for certificado que não é possível atender a demanda *i*, nela será retornado o sequenciamento inicial, restabelecido na linha 34, e o valor -1.

4.5 Algoritmos de Viabilidade

Esses algoritmos visam melhorar a viabilidade do sequenciamento aplicando procedimentos que pretendem atender demandas não atendidas inicialmente após a etapa da heurística construtiva.

4.5.1 Algoritmo de Viabilidade 1

Este procedimento busca atender uma demanda *i* não atendida na construção *x* ao alocar seu atendimento antes de certas demandas da construção *y*. Isso é realizado através do desalocamento do atendimento de certas demandas da construção *y*, do atendimento da demanda *i* e do reatendimento das demandas não atendidas da construção *y*. Esse movimento é realizado para todas as demandas atendidas no sequenciamento e todas as construções com demandas não atendidas. Caso seja possível diminuir o *NívelInviabilidade* do sequenciamento com um destes movimentos, retorna-se o novo sequenciamento obtido e o valor 1. Caso não seja possível diminuir o *NívelInviabilidade*, é retornado o sequenciamento antes de entrar nessa função e o valor 0. O pseudocódigo está no Algoritmo 6.

Algoritmo 6: VIABILIDADE 1

```

Entrada:  $S$ 
Saída:  $S$ , valor
1 início
2   AtribuiSequenciamento( $S0$ );
3    $NívelInviabilidadeAnterior \leftarrow S.NívelInviabilidade$ ;
4   NenhumaDemandaRetirada( );
5   enquanto  $DemandaRetirar(j \in D^y) = 1$  faça
6     DeletaTarefasPosterioresEmConstrução( $j$ );
7     MarcaConstruçõesParaAnalisar( );
8     enquanto  $DemandaNãoAtendida(y, i \in D^x)$  faça
9        $P.OrdenaPlantas()$ ;
10      para  $p \leftarrow \{1, \dots, P\}$  faça
11         $p.OrdenaVeículos()$ ;
12        para  $v \leftarrow \{1, \dots, V^p\}$  faça
13          AtribuiSequenciamento( $S1$ );
14          se  $AdicionaDemandaVeículo(i, v, S) = 1$  então
15            RealocaDemandas( $y$ );
16             $S.CalculaNívelInviabilidade()$ ;
17            se  $NívelInviabilidadeAnterior > S.NívelInviabilidade$  então
18              retorna  $S, 1$ ;
19            fim
20          fim
21           $RetornaSequenciamento(S1)$ ;
22        fim
23      fim
24       $MarcaConstruçãoAnalisada(x)$ ;
25    fim
26     $RetornaSequenciamento(S0)$ ;
27     $j.AnaliseRemoção \leftarrow 1$ ;
28  fim
29  retorna  $S, 0$ ;
30 fim

```

O procedimento recebe como entrada um sequenciamento S , e tem como saída um sequenciamento S e um valor que indica se melhorou o *NívelInviabilidade* (valor 1) ou não (valor 0).

Na linha 2 e 3 é armazenado o sequenciamento inicial e seu nível de inviabilidade em *NívelInviabilidadeAnterior*. A linha 4 sinaliza que nenhuma demanda atendida tenha sido marcada como retirada. O *loop* das linhas 5 a 28 é realizado enquanto tiver uma demanda atendida e que não tenha sido marcada como retirada. A função $DemandaRetirar(j \in D^y)$ retorna 1 se ainda tiver uma demanda não marcada como retirada, e retorna uma demanda j não marcada. O procedimento de escolha da demanda j segue a ordenação das construções e seleciona a primeira construção e demanda não marcada como retirada. A linha 6 desaloca o atendimento de todas as demandas após a demanda j na construção y , incluindo a demanda j . A linha 7 marca as construções

com demandas não atendidas no sequenciamento inicial que devam ser analisadas. O *loop* das linhas 8 a 25 é realizado enquanto houver uma construção x diferente de y a ser analisada. Caso a função $\text{DemandaN\~{a}oAtendida}(y, i \in D^x)$ verificar se ainda tem uma construção x diferente de y a analisar, ela retorna a primeira demanda não atendida i da construção x . A linha 9 ordena as plantas para que o *loop* das linhas 10 a 23 passe por todas as plantas respeitando essa ordenação. O mesmo vale para a linha 11 e o *loop* das linhas 12 a 22, nesse caso a ordenação é em relação aos veículos. A linha 13 armazena o sequenciamento corrente na variável $S1$. Na linha 14 é realizado uma tentativa de atendimento da demanda i da construção x utilizando a planta p e o veículo v corrente. Caso seja possível atender a demanda i , é tentado reatender as demandas desaloçadas em y na linha 15. Será calculado o *NívelInviabilidade* na linha 16. Caso decresça esse nível do sequenciamento na linha 17, será retornado o sequenciamento corrente e o valor 1 na linha 18. Caso não, será restabelecido o sequenciamento para o estado antes de ocorrer a tentativa de atender a demanda j na linha 21. A linha 24 marca a construção x como já analisada. A linha 26 restabelece o sequenciamento para a configuração inicial. A linha 27 marca a demanda j da construção y como já removida. A linha 29 retorna o sequenciamento inicial e o valor 0, sinalizando que não foi possível melhorar a viabilidade da solução.

4.5.2 Algoritmo de Viabilidade 2

Este procedimento escolhe uma demanda não atendida i da construção c e desaloca o atendimento de todas as demandas atendidas por plantas em horários iguais ou superiores aqueles em que essas plantas poderiam atender a demanda i . Com isso, é realizado o atendimento de i com um certo veículo v . Em seguida busca-se atender a todas as demandas desaloçadas, utilizando o mesmo procedimento da heurística construtiva. Caso no fim desse procedimento for obtido um nível de inviabilidade menor que o do sequenciamento inicial, será retornado o sequenciamento obtido e o valor 1. Caso contrário, será restabelecido o sequenciamento para o estado anterior ao atendimento da demanda i e é executado novamente o procedimento Viabilidade 2 para um outro veículo. Esse procedimento é executado para todos os veículos. Caso não consiga atender a demanda i com um dos veículos, o sequenciamento inicial é restabelecido e será realizado o procedimento Viabilidade 2 acima utilizando uma demanda não atendida de outra construção. Esse procedimento é executado para todas as construções com demandas não atendidas.

Também foi implementado um procedimento quando a demanda i não atendida não for a primeira demanda da construção c . Quando ocorrer a tentativa de atendi-

mento dessa demanda e não obtiver sucesso, será também desalocado o atendimento de demandas anteriores em c . Inicialmente é desalocado o atendimento da demanda $i - 1$ anterior a i . Em seguida, será executado o procedimento de desalocar o atendimento das demandas de outras construções baseando agora no horário que é possível atender a demanda $i - 1$. Após desalocar o atendimento de todas as demandas que utilizam plantas que poderiam atender a $i - 1$, procura-se reatender as demandas $i - 1$ e i . Para o atendimento da demanda i é utilizado certo veículo v . Em seguida, é realizado o reatendimento das demandas desalocadas. Caso ao final seja obtido um sequenciamento com nível de inviabilidade menor que o sequenciamento inicial, é retornado o sequenciamento obtido e o valor 1. Caso contrário, é restabelecido o sequenciamento anterior ao atendimento de i e é realizado o atendimento de i com outro veículo v . Esse procedimento é executado para todos os veículos v . Caso não consiga diminuir o nível de inviabilidade, será também desalocado o atendimento de uma demanda anterior a demanda $i - 1$ em c . Após, será repetido o procedimento de desalocar atendimentos de demandas de outras construções, baseando agora na demanda anterior a $i - 1$. É realizado até melhorar o nível de inviabilidade ou até não se ter mais demandas anteriores a desalocar em c .

O pseudocódigo está no Algoritmo 7.

Algoritmo 7: VIABILIDADE 2

```

Entrada:  $S$ 
Saída:  $S$ , valor
1 início
2   AtribuiSequenciamento( $S0$ );
3    $NívelInviabilidadeAnterior \leftarrow S.NívelInviabilidade$ ;
4   enquanto  $ExisteConstruçãoNãoAnalisada(i \in D^{c \in C}) = 1$  faça
5      $DemFinal \leftarrow i$ ;
6      $DemInício \leftarrow DemFinal$ ;
7      $Analisa \leftarrow 0$ ;
8     enquanto  $Analisa = 0$  faça
9       DeletaTodasTarefasApósDemanda( $DemInício$ );
10      ReadicionaDemandasConstrução( $c$ ,  $DemInício$ ,  $DemFinal - 1$ );
11      AtribuiSequenciamento( $S1$ );
12       $P.OrdenaPlantas()$ ;
13      para  $p \leftarrow \{1, \dots, P\}$  faça
14         $p.OrdenaVeículos()$ ;
15        para  $v \leftarrow \{1, \dots, V^p\}$  faça
16          se  $AdicionaDemandaVeículo(DemFinal, v) = 1$  então
17            ReadicionaTodasDemandas();
18             $S.CalculaNívelInviabilidade()$ ;
19            se  $NívelInviabilidadeAnterior > S.NívelInviabilidade$  então
20              retorna  $S$ ,  $1$ ;
21            fim
22          fim
23         $RetornaSequenciamento(S1)$ ;
24      fim
25    fim
26     $RetornaSequenciamento(S0)$ ;
27     $DemInício \leftarrow DemInício - 1$ ;
28    se  $DemInício < 1$  então
29       $Analisa \leftarrow 1$ ;
30    fim
31  fim
32 fim
33 retorna  $S$ ,  $0$ ;
34 fim

```

O procedimento recebe como entrada um sequenciamento S , e tem como saída um sequenciamento S e um valor indicando a melhora da viabilidade do sequenciamento (valor 1) ou não (valor 0).

Na linha 2 e 3 é armazenado o sequenciamento inicial em $S0$ e seu nível de inviabilidade em $NívelInviabilidadeAnterior$. O *loop* das linhas 4 a 32 é realizado enquanto tiver uma construção c possuindo uma demanda i que pode ser atendida e ainda não foi analisada. A linha 4 seleciona a demanda $i \in D^c$ que pode ser atendida e marca a construção c como já analisada.

As linhas 5 e 6 iniciam $DemInício$ e $DemFinal$ com a demanda i . Essas va-

riáveis representam a demanda inicial e final analisadas na construção c . A demanda $DemFinal$ é sempre a demanda não atendida na construção c no sequenciamento inicial. A linha 7 inicializa $Analisa$ com o valor 0 para entrar no *loop* das linhas 8 a 31. A variável $Analisa$ controla se a construção c foi completamente analisada pelo procedimento. A linha 9 desaloca o atendimento de todas as demandas atendidas por plantas em um horário igual ou superior a um horário que poderia atender a demanda $DemInício$ na construção c . Na linha 10 são atendidas todas as demandas anteriores a $DemFinal$ na construção c , ou seja, as demandas $DemInício$ até a $DemFinal - 1$, caso existam. A linha 11 armazena o sequenciamento corrente em $S1$. A linha 12 ordena as plantas de acordo com os critérios estabelecidos em 4.2. O *loop* das linhas 13 a 25 será executado para todas as plantas seguindo a ordenação realizada. A linha 14 ordena os veículos para que o *loop* das linhas 15 a 24 seja executado para todos os veículos seguindo essa ordem. Na linha 16 há uma tentativa de atender a demanda $DemFinal$ com a planta p e o veículo v corrente. Caso consiga atender a demanda $DemFinal$, haverá uma tentativa de atender as demandas não atendidas de outras construções na linha 17. Esse procedimento executa o mesmo procedimento da heurística construtiva. A linha 19 verifica se o sequenciamento resultante possui um nível de inviabilidade (calculado na linha 18) melhor que o do sequenciamento inicial. Caso for melhor, será retornado o sequenciamento corrente e o valor 1 na linha 20. A linha 23 restabelece o sequenciamento antes de haver uma tentativa de atender a $DemFinal$ na linha 16. A linha 26 restabelece o sequenciamento inicial. A linha 27 decresce a $DemInício$. Caso $DemInício$ seja menor que a primeira demanda da construção corrente, o valor 1 é atribuído a variável $Analisa$ para sair do *loop* das linhas 8 a 31. Caso não consiga melhorar a viabilidade da solução, será retornado o sequenciamento inicial e o valor 0 na linha 33.

4.5.3 Execução dos procedimentos de Viabilização

Os procedimentos de viabilização são executados em sequência. Primeiro é realizado o primeiro procedimento de viabilização ($Viabilidade1(S)$). Caso após este primeiro procedimento a solução continue inviável, será realizado o segundo procedimento de viabilização ($Viabilidade2(S)$). O sequenciamento obtido é retornado. O pseudocódigo está no Algoritmo 8.

Algoritmo 8: PROCEDIMENTO DE VIABILIZAÇÃO

Entrada: S **Saída:** S

```

1 início
2    $MelhorouViabilidade \leftarrow 1$ ;
3   enquanto  $MelhorouViabilidade = 1$  faça
4      $MelhorouViabilidade \leftarrow Viabilidade1(S)$ ;
5   fim
6   se  $S.NívelInviabilidade > 0$  então
7      $MelhorouViabilidade \leftarrow 1$ ;
8     enquanto  $MelhorouViabilidade = 1$  faça
9        $MelhorouViabilidade \leftarrow Viabilidade2(S)$ ;
10    fim
11  fim
12  retorna  $S$ ;
13 fim
```

O procedimento recebe como entrada o sequenciamento S , e tem como saída o sequenciamento S após realizar os procedimentos de viabilização.

A linha 2 inicia $MelhorouViabilidade$ com o valor 1 para que entre no *loop* das linhas 3 a 5. As linhas 3 a 5 realizam o procedimento $Viabilidade1$ até não obter melhorias na viabilidade do sequenciamento S . A linha 6 verifica se o sequenciamento S continua inviável. Caso sim, será realizado o procedimento $Viabilidade2$. A linha 7 tem a mesma função da linha 2, mas nesse caso é para entrar no *loop* das linhas 8 a 10. As linhas 8 a 10 executam o procedimento $Viabilidade2$ até não obter melhorias na viabilidade do sequenciamento S . A linha 12 retorna o sequenciamento S obtido.

4.6 Busca Local

Busca Local é uma abordagem geral que visa a melhoria da solução encontrada por uma heurística. Ela parte de uma solução viável inicial, realiza uma alteração local na solução e verifica se a nova solução é melhor que a solução inicial. Em nosso estudo ela é realizada até que não se encontre uma solução melhor, ou seja, até que ela alcance um ótimo local.

Apresentamos a seguir as três buscas locais criadas para esse problema que vi-

são obter uma melhora na variável do sequenciamento *SomaHoráriosFinais*, soma dos horários dos últimos descarregamentos nas construções, mantendo o *NívelInviabilidade* como nulo, ou seja, tendo todas as demandas atendidas. Vale ressaltar que quanto menor o valor de *SomaHoráriosFinais* melhor será a solução em nosso estudo.

4.6.1 Procedimento de Busca Local Veículos

Esse procedimento verifica a possibilidade de melhorar a variável *SomaHoráriosFinais* do sequenciamento corrente ao alterar o veículo utilizado para atender uma demanda. A busca inicialmente desaloca o atendimento da demanda $i \in D^c$ e das demandas posteriores a essa na construção c . Depois realiza o atendimento de i com um veículo diferente do usado no sequenciamento inicial e busca atender as outras demandas de c . Caso melhore *SomaHoráriosFinais*, retornará o sequenciamento com menor *SomaHoráriosFinais* e o valor 1. Caso contrário, restabelecerá o sequenciamento com a demanda i e as demandas posteriores a i anteriormente desalocadas. Em seguida, é repetido o procedimento de atender a demanda i utilizando um outro veículo e atender as outras demandas desatendidas. Isso é realizado para todos os veículos e para todas as demandas do sequenciamento. Caso não melhore *SomaHoráriosFinais*, será retornado o sequenciamento inicial e o valor 0. O pseudocódigo está no Algoritmo 9.

Algoritmo 9: BUSCALocalVEÍCULO

```

Entrada:  $S$ 
Saída:  $S$ , valor
1 início
2   AtribuiSequenciamento( $S0$ );
3    $SomaHoráriosFinaisAnterior \leftarrow S.CalculaSomaHoráriosFinais()$ ;
4   para  $c \leftarrow \{1, \dots, C\}$  faça
5     para  $i \leftarrow \{1, \dots, D^c\}$  faça
6       DeletaTarefasPosterioresEmConstrução( $i$ );
7       AtribuiSequenciamento( $S1$ );
8        $P.OrdenaPlantas()$ ;
9       para  $p \leftarrow \{1, \dots, P\}$  faça
10         $p.OrdenaVeículos()$ ;
11        para  $v \leftarrow \{1, \dots, V^p\}$  faça
12          se  $AdicionaDemandaVeículo(i, v) = 1$  então
13            se  $RealocaDemandas(c) = 1$  então
14               $S.CalculaSomaHoráriosFinais()$ ;
15              se  $SomaHoráriosFinaisAnterior > S.SomaHoráriosFinais$  então
16                retorna  $S, 1$ ;
17              fim
18            fim
19             $RetornaSequenciamento(S1)$ ;
20          fim
21        fim
22      fim
23       $RetornaSequenciamento(S0)$ ;
24    fim
25  fim
26  retorna  $S, 0$ ;
27 fim

```

O procedimento recebe como entrada um sequenciamento S , e tem como saída um sequenciamento S e um valor que será 1 caso melhore $SomaHoráriosFinais$, ou 0 caso não consiga melhorar $SomaHoráriosFinais$.

Nas linhas 2 e 3 é armazenado o sequenciamento inicial em $S0$ e seu $SomaHoráriosFinais$ em $SomaHoráriosFinaisAnterior$. O *loop* das linhas 4 a 25 é realizado para todas as construções. O *loop* das linhas 5 a 24 é realizado para todas as demandas da construção c . Na linha 6 é desalocado o atendimento da demanda i e das demandas posteriores na construção c . Na linha 7 é armazenado o sequenciamento corrente em $S1$. O *loop* das linhas 9 a 22 é realizado para todas as plantas, seguindo a ordenação realizada na linha 8. O *loop* das linhas 11 a 21 é realizado para todos os veículos da planta p corrente, seguindo a ordenação realizada na linha 10. Na linha 12 é realizada a tentativa de atender a demanda i com a planta p e o veículo v corrente. Caso consiga atender a demanda i , busca atender as outras demandas de c através da função $RealocaDemandas(c)$ na linha 13. Caso realize o atendimento de todas as demadas

de c , a linha 15 verifica se $SomaHoráriosFinais$ do sequenciamento corrente (calculado na linha 14) é menor que $SomaHoráriosFinais$ do sequenciamento inicial. Caso seja menor, será retornado o sequenciamento corrente e o valor 1 na linha 16. A linha 19 restabelece o sequenciamento antes do reatendimento da demanda i na linha 12. A linha 23 restabelece o sequenciamento inicial. Caso não melhore $SomaHoráriosFinais$, o sequenciamento inicial e o valor 0 são retornados na linha 26.

4.6.2 Procedimento de Busca Local Construção

Esse procedimento verifica a possibilidade de melhorar o valor da variável $SomaHoráriosFinais$ ao atender as demandas de uma construção x antes de atender certas demandas de uma construção y . Consiste em desalocar o atendimento de certas demandas de y e de x , e busca atender as demandas de x antes de atender as demandas de y . Caso no final desse procedimento seja obtida uma solução viável com um melhor $SomaHoráriosFinais$, é retornado o novo sequenciamento e o valor 1. Caso contrário, escolhe um novo conjunto de demandas que deixarão de ser atendidas em y e repete-se o procedimento. É realizado para todo par de construções. Caso não melhore $SomaHoráriosFinais$, é retornado o sequenciamento inicial e o valor 0. O pseudocódigo está no Algoritmo 10.

Algoritmo 10: BUSCALOCALCONSTRUÇÃO

```

Entrada:  $S$ 
Saída:  $S$ , valor
1 início
2   AtribuiSequenciamento( $S0$ );
3    $SomaHoráriosFinaisAnterior \leftarrow S.CalculaSomaHoráriosFinais( )$ ;
4   para  $y \leftarrow \{1, \dots, C\}$  faça
5     para  $i \leftarrow \{1, \dots, D^y\}$  faça
6       DeletaTarefasPosterioresEmConstrução( $i$ );
7       AtribuiSequenciamento( $S1$ );
8       para  $x \leftarrow \{1, \dots, C \mid x \neq y\}$  faça
9         DeletaTarefasEmConstrução( $x$ );
10        se  $RealocaDemandas(x) = 1$  então
11          se  $RealocaDemandas(y) = 1$  então
12             $S.CalculaSomaHoráriosFinais( )$ ;
13            se  $SomaHoráriosFinaisAnterior > S.SomaHoráriosFinais$  então
14              retorna  $S$ ,  $1$ ;
15            fim
16          fim
17        fim
18        RetornaSequenciamento( $S1$ );
19      fim
20      RetornaSequenciamento( $S0$ );
21    fim
22  fim
23  retorna  $S$ ,  $0$ ;
24 fim

```

O procedimento recebe como entrada um sequenciamento S , e tem como saída um sequenciamento S e um valor que será 1 caso melhore $SomaHoráriosFinais$, ou 0 caso não consiga melhorar $SomaHoráriosFinais$.

Na linha 2 e 3 é armazenado o sequenciamento inicial em $S0$ e $SomaHoráriosFinais$ em $SomaHoráriosFinaisAnterior$. O *loop* das linhas 4 a 22 é executado para todas as construções $y \in C$. O *loop* das linhas 5 a 21 é realizado para todas as demandas de y . A linha 6 desaloca o atendimento da demanda i e das demandas posteriores na construção y . A linha 7 armazena o sequenciamento corrente em $S1$. O *loop* das linhas 8 a 19 é realizado para todas as construções $x \in C$, exceto para a construção y corrente. A linha 9 desaloca o atendimento de todas as demandas da construção x . Nas linhas 10 e 11 é realizado um novo atendimento das demandas desalocadas, mas o atendimento das demandas da construção x será realizado antes das demandas de y . Caso sejam atendidas todas as demandas que foram desalocadas, será verificado se o novo sequenciamento tem $SomaHoráriosFinais$ (calculada na linha 12) menor que $SomaHoráriosFinais$ do sequenciamento inicial. Caso seja menor, será retornado o sequenciamento corrente e o valor 1 na linha 14. A linha 18 restabelece o se-

quenciamento para a situação antes de desalocar o atendimento das demandas de x . A linha 20 restabelece o sequenciamento inicial. Caso não melhore *SomaHoráriosFinais*, o sequenciamento inicial e o valor 0 são retornados na linha 23.

4.6.3 Procedimento de Busca Local Planta

Este procedimento verifica a possibilidade de melhorar o *SomaHoráriosFinais* do sequenciamento reconstruindo o sequenciamento ao atender uma demanda i da construção c com uma planta p diferente da qual a atende. É executado ao desalocar o atendimento de todas as demandas atendidas depois da demanda i em c , atender a demanda i com uma planta p e veículo v diferentes daqueles do sequenciamento inicial, e, por fim, reatender as demandas desalocadas. Esse procedimento é realizado para todas as demandas do sequenciamento. É executado com a expectativa de atender cada demanda i utilizando todos os veículos de todas as plantas. Caso seja obtida uma solução com melhor *SomaHoráriosFinais*, será retornado o novo sequenciamento e o valor 1. Caso não melhore *SomaHoráriosFinais*, retorna o sequenciamento inicial e o valor 0. O pseudocódigo está no Algoritmo 11.

Algoritmo 11: BUSCALocalPLANTA

```

Entrada:  $S$ 
Saída:  $S$ , valor
1 início
2   AtribuiSequenciamento( $S_0$ );
3    $SomaHoráriosFinaisAnterior \leftarrow S.CalculaSomaHoráriosFinais( )$ ;
4    $P.OrdenaPlantas( )$ ;
5   para  $p_1 \leftarrow \{1, \dots, P\}$  faça
6     para  $c_1 \leftarrow \{1, \dots, C\}$  faça
7       para  $d_1 \leftarrow \{1, \dots, D^{c_1}\}$  faça
8         se  $PlantaAtendeDemanda(p_1, d_1) = 1$  então
9            $DeletaTarefasPosterioresEmConstrução(d_1)$ ;
10        fim
11      fim
12    fim
13    para  $c_2 \leftarrow \{1, \dots, C\}$  faça
14      AtribuiSequenciamento( $S_1$ );
15      enquanto  $c_2.DemandaNãoAnalisada(d_2) = 1$  faça
16        AtribuiSequenciamento( $S_2$ );
17         $P.OrdenaPlantas( )$ ;
18        para  $p_2 \leftarrow \{1, \dots, P \mid p_1 \neq p_2\}$  faça
19           $p_2.OrdenaVeículos( )$ ;
20          para  $v \leftarrow \{1, \dots, V^{p_2}\}$  faça
21            AtribuiSequenciamento( $S_3$ );
22            se  $AdicionaDemandaVeículo(d_2, v) = 1$  então
23              se  $RealocaDemandas(c_2) = 1$  então
24                se  $ReadicionaTodasDemandas( ) = 1$  então
25                   $S.CalculaSomaHoráriosFinais( )$ ;
26                  se  $SomaHoráriosFinaisAnterior >$ 
27                     $S.SomaHoráriosFinais$  então
28                     $retorna\ S, 1$ ;
29                  fim
30                fim
31              fim
32             $RetornaSequenciamento(S_3)$ ;
33          fim
34        fim
35         $RetornaSequenciamento(S_2)$ ;
36         $AdicionaDemanda(d_2, P)$ ;
37      fim
38       $RetornaSequenciamento(S_1)$ ;
39    fim
40     $RetornaSequenciamento(S_0)$ ;
41  fim
42   $retorna\ S, 0$ ;
43 fim

```

O procedimento recebe como entrada um sequenciamento S , e tem como saída um sequenciamento S e um valor que será 1 caso melhore $SomaHoráriosFinais$, ou 0

caso não consiga melhorar *SomaHoráriosFinais*.

Na linha 2 e 3 são armazenados o sequenciamento inicial em $S0$ e seu *SomaHoráriosFinais* em *SomaHoráriosFinaisAnterior*. O *loop* das linhas 5 a 41 é executado para todas as plantas, seguindo a ordenação realizada na linha 4.

A partir deste ponto é possível dividir o restante do procedimento em duas partes. Na primeira, todas as demandas supridas por uma determinada planta $p1$, incluindo as demandas atendidas posteriormente a essas demandas atendidas por $p1$ nas construções, são desalocadas. Na segunda, é realizada a tentativa de atender uma demanda atendida por $p1$ com uma planta $p2$ diferente. Ainda na segunda parte, é realizada a tentativa de reatender todas as outras demandas desalocadas.

A primeira parte possui o *loop* das linhas 6 a 12 que é realizado para todas as construções $c1$. Interno, o *loop* das linhas 7 a 11 é realizado para todas as demandas $d1$ da construção $c1$. A linha 9 desaloca o atendimento da demanda $d1$ e das demandas posteriores a ela em $c1$, caso constate que a demanda $d1$ é atendida pela planta $p1$ corrente na linha 8.

A segunda parte está contida dentro do *loop* das linhas 13 a 39. Ela começa na linha 13 que irá realizar um *loop* para todas as construções $c2$. Na linha 14 é armazenado o sequenciamento corrente em $S1$. O *loop* das linhas 15 a 37 é executado enquanto houver uma demanda $d2$ não atendida na construção $c2$, a função *DemandaNãoAnalisada()* atribui $d2$ a primeira demanda não atendida em $c2$ e retorna o valor 1. Caso não tenha uma demanda não atendida em $c2$, a função *DemandaNãoAnalisada()* retorna 0. Na linha 16 é armazenado o sequenciamento corrente em $S2$. O *loop* das linhas 18 a 34 é executado para todas as plantas, seguindo a ordenação realizada na linha 17. O *loop* das linhas 20 a 34 é realizado para todos os veículos v da planta $p2$, seguindo a ordenação realizada na linha 19. Na linha 21 é armazenado o sequenciamento corrente em $S3$. É realizada a tentativa de atendimento a demanda $d1$ com a planta $p2$ e o veículo v na linha 22. Caso consiga atender a demanda $d2$, será realizada a tentativa de reatender as outras demandas desalocadas na construção $c2$ para depois tentar reatender as demandas desalocadas nas outras construções, linhas 23 e 24. Uma verificação se *SomaHoráriosFinais* do novo sequenciamento (calculado na linha 25) possui melhor valor que o da solução inicial é realizada na linha 26, caso seja atendidas todas as demandas desalocadas. Caso seja melhor, será retornado o sequenciamento corrente e o valor 1 na linha 27. Na linha 32 é restabelecido o sequenciamento ao estado antes de reatender a demanda $d2$ na linha 22. Nas linhas 35 e 36 é restabelecido o sequenciamento ao estado da linha 16, e é atendida a demanda $d1$ corrente para que ela não seja selecionada novamente para $d2$ na linha 15. A linha 38 restabelece o sequenciamento após desalocar o atendimento das demandas

na primeira parte. O sequenciamento inicial é restabelecido na linha 40.

Caso não consiga melhorar *SomaHoráriosFinais* do sequenciamento, será retornado o sequenciamento inicial e o valor 0 na linha 42.

Capítulo 5

Resultados

Neste capítulo apresentamos uma avaliação e uma comparação das heurísticas e modelos explorados. Para isso, introduzimos um conjunto de instâncias a serem testadas e os parâmetros escolhidos para a ordenação das construções, plantas e veículos da heurística. Apresentamos os resultados obtidos pelas heurísticas e pelos modelos permitindo e não permitindo violações nas janelas de tempo. Finalizamos com os resultados da execução do modelo tendo como ponto de partida o resultado da heurística.

Todos os testes foram executados usando um computador com 32 Gigabytes de memória RAM, processador Intel Xeon(R) CPU E5645 2.40GHz x 24 e sistema operacional Ubuntu 14.04 LTS. A linguagem utilizada, tanto na heurística como no modelo de programação linear inteiro misto, foi C++.

5.1 Descrição das instâncias

As instâncias utilizadas nos trabalhos já realizados sobre esse tema não possuem todos os dados necessários para a utilização em nosso estudo. O motivo está no fato do problema ser tratado de diferentes modos entre os muitos estudos já realizados, [Kinable et al., 2014]. Em alguns trabalhos as instâncias nem foram disponibilizadas ao público como mencionado em Kinable et al. [2014]. Com isso, tivemos a necessidade da criação de um conjunto de instâncias. A descrição das instâncias, o modo como elas foram criadas e a nomenclatura usada estão a seguir.

5.1.1 Descrição e criação das instâncias

Os dados de tempo de vida do concreto, velocidade dos veículos, tempo que um veículo leva para descarregar em uma construção, tempo de carregamento de um veículo em

uma planta e o tempo máximo entre descarregamentos consecutivos em uma mesma construção foram obtidos através de dados encontrados em trabalhos da literatura. Os valores mais comuns destes dados foram os utilizados em nossas instâncias. Estes valores estão na tabela 5.1.

Tabela 5.1. Valores dos dados das instâncias

Dados	Valores
Tempo de vida do concreto	90 minutos
Velocidade dos veículos	30 Km/h
Tempo de descarregamento de um veículo em uma construção	10 minutos
Tempo de carregamento de um veículo em uma planta	5 minutos
Tempo máximo entre descarregamentos consecutivos em uma mesma construção	10 minutos

A localização das plantas e das construções foram escolhidas utilizando as coordenadas contidas nas instâncias propostas em Solomon [1987] com 25 clientes. Foram utilizados dois dos três grupos propostos, o dos subconjuntos de clientes agrupados (C101) e dos clientes distribuídos aleatoriamente (R101). Estas instâncias foram obtidas no site *Networking and Emerging Optimization*¹ e as distâncias euclidianas entre os nós foram também obtidas da mesma referência. Vale ressaltar que a unidade de distância considerada entre os nós foi a de quilômetros (Km).

Para os dois grupos a primeira planta se situa no nó 0. As outras plantas e construções foram escolhidas aleatoriamente² entre os nós restantes. Tivemos a precaução de que nenhum nó representasse diferentes construções ou plantas em uma mesma instância criada.

O número de descarregamentos necessários para suprir cada construção foi gerado aleatoriamente² entre os valores inteiros de 1 a 5. Os intervalos para atender essas demandas nas construções também foram selecionados aleatoriamente² entre os horários de 8:00 até 12:00 e 13:30 até 17:00. Eles sempre começam em horários com término em 00 minutos ou 30 minutos (exemplo: 8:00, 8:30, 9:00, 9:30, 10:00, em diante). A janela de tempo para atender as demandas de cada construção é um múltiplo do tempo necessário para atendê-las descarregando os veículos seguidamente na construção. Por exemplo, caso o número de demandas necessárias em uma construção for 3 (o tempo de atender todas as demandas seguidamente é de 30 minutos) e o múltiplo usado for

¹Networking and Emerging Optimization: <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-with-time-windows-instances/>

²Seleção aleatória foi realizada através de uma função pseudo-aleatória *rand()* da biblioteca *cstdlib.h* da linguagem C++.

3, a janela de tempo que a construção dispõe é de 90 minutos. Sendo assim, caso o horário inicial seja 9:00, a janela de tempo na construção será de 9:00 até as 10:30. Os múltiplos usados foram 2, 3 e 4. Essa estratégia de geração de janelas de atendimento foi adotada para simular horários mais factíveis de recebimento de concreto nas construções.

A janela de tempo em que as plantas podem começar a carregar um veículo é de 7:00 até as 18:00. Isso visa simular o horário de funcionamento mais factível de uma planta de produção de RMC. Cada planta em nossas instâncias possui dez veículos visando refletir um número compatível com a realidade de uma planta.

5.1.2 Nomeclatura das instâncias

A nomeclatura das instâncias criadas começa com os caracteres “p-” para sinalizar que elas foram criadas para o problema estudado nesse trabalho. Em seguida tem-se os caracteres “C101-” ou “R101-”. No caso de “C101-”, a instância foi criada utilizando as coordenadas dos nós da instância C101 de Solomon [1987]. No caso de “R101-”, a instância utilizou os nós da instância R101.

A próxima parte da nomeclatura sinaliza o número de plantas e construções da instância. Essa parte é iniciada com a letra P seguida pelo número de plantas e a letra C seguida pelo número de construções. Foram criadas instâncias com 1 planta e 5 e 8 construções, 2 plantas e 10 e 15 construções, e 3 plantas e 15 construções.

Em seguida se tem os caracteres “-I” e o número que indica o múltiplo do tempo utilizado para calcular a janela de tempo das construções. Também são especificados os caracteres “-V” seguido do número que foi utilizado como semente aleatória do processo de criação da instância. Foram utilizados os valores 1, 2 e 3 em nosso estudo.

Um exemplo para o melhor entendimento da nomeclatura das instâncias é o caso da instância “p-C101-P1C5-I2-V1”. Ela foi criada a partir das coordenadas da instância C101 e possui uma planta e cinco construções. O multiplicador do tempo utilizado para calcular a janela de tempo das construções é 2, e a semente aleatória do processo de criação foi o número 1.

5.2 Ordem de escolha dos veículos, construções e plantas

Executamos uma série de testes para se definir o conjunto de parâmetros a usar na heurística. Visamos um conjunto de parâmetros que produzam mais soluções viáveis,

soluções que todas as demandas são atendidas nas intâncias, e em caso de empate, escolheremos as que produzem os melhores valores das soluções.

5.2.1 Parâmetros do algoritmo para escolha dos veículos, construções e plantas

Para determinarmos quais são os melhores parâmetros para a ordem de escolha na heurística citada na seção 4.2, executamos 100 vezes a heurística construtiva seguida dos procedimentos de viabilização para cada instância. Fixamos em meio minuto o *InterTempo*. Sendo que quando um, ou mais, dos parâmetros é aleatório (ex: Ve-4, Co-4 ou Pl-4) nos executamos 30 vezes cada uma das 100 execuções da heurística seguida dos procedimentos de viabilização. Por exemplo, caso os parâmetros sejam Ve-1 Co-1 Pl-4 para a instância p-C101-P1C5-I2-V1, ao executarmos 30 vezes a heurística seguida dos procedimentos de viabilização e ao final tivermos 24 soluções viáveis dentre as 30, a porcentagem de soluções viáveis encontradas para esta intância é 8/10 ou 80%. Como executamos 100 vezes essas 30 execuções, pegamos a porcentagem de soluções viáveis dentre essas 100. No caso de 80 execuções com 80% e 20 com 100% de soluções viáveis, o número de soluções viáveis será de 84%. Os tempos de execução não fizeram parte do escopo de nosso experimento, pois possuem em sua maioria tempos de execução inferiores a 2 segundos e de no máximo 4 segundos.

A tabela 5.2 contém a porcentagem de soluções viáveis encontradas para todo o conjunto de instâncias variando os parâmetros da ordem de escolha. A primeira linha refere a escolha do veículo, sendo da segunda coluna até quinta coluna referente a escolha Ve-1, da sexta a nona da escolha Ve-2, da décima até a décima terceira a escolha Ve-3, da décima quarta até a décima sétima a Ve-4. A segunda linha se refere a escolha das construções, sendo a segunda, sexta, décima e décima quarta refere a Pl-1; a terceira, sétima, décima primeira e décima quinta refere a Pl-2; a quarta, oitava, décima segunda e décima sexta referente a Pl-3; a quinta, nona, décima terceira e décima sétima refere a Pl-4. A terceira linha se refere a escolha Co-1, a quarta linha a Co-2, a quinta linha a escolha Co-3 e a sexta linha a escolha Co-4.

Tabela 5.2. Porcentagem de intâncias com soluções viáveis

	Ve-1				Ve-2				Ve-3				Ve-4			
	Pl-1	Pl-2	Pl-3	Pl-4												
Co-1	80,0%	81,1%	80,0%	85,6%	83,3%	87,8%	82,2%	87,8%	83,3%	87,8%	82,2%	87,8%	87,6%	88,9%	87,3%	88,9%
Co-2	82,2%	82,2%	78,9%	86,7%	84,4%	85,6%	81,1%	87,8%	85,6%	86,7%	81,1%	87,8%	87,8%	88,9%	86,7%	88,9%
Co-3	85,6%	86,7%	85,6%	89,4%	84,4%	87,8%	82,2%	89,5%	84,4%	86,7%	83,3%	89,7%	89,5%	89,9%	87,6%	89,3%
Co-4	89,5%	89,5%	89,4%	89,3%	90,0%	89,9%	89,5%	89,8%	90,0%	89,8%	89,8%	89,9%	89,9%	89,8%	89,3%	89,4%

A melhor combinação de parâmetros foi Ve-2 Co-4 Pl-1 com 90,0% de soluções

viáveis. Apesar da combinação Ve-3 Co-4 Pl-1 também possuir 90,0% de soluções viáveis, ao se compara-las a primeira gera soluções superiores em 66,4% das instâncias, sendo que Ve-3 Co-4 Ve-1 apresentou superioridade apenas em 18,3% e ocorreu empate em 15,3% das instâncias.

Uma análise com apenas uma execução e uma análise mais detalhada dos resultados das instâncias são apresentadas no apêndice A.

5.3 Buscas Locais

Foram executadas 100 iterações da heurística seguidas dos processos de viabilização e buscas locais. Realizamos testes com a busca local veículos (seção 4.6.1), a busca local construções (seção 4.6.2), a busca local plantas (seção 4.6.3), e na última executamos as três buscas locais anteriores em sequencia. Esta última chamamos de buscas locais conjuntas.

Os resultados destes testes estão no apêndice B. Na Figura 5.1 temos nas três primeiras colunas o número de instâncias da qual as buscas locais veículo, construção e planta encontraram a melhor solução obtida entre as três buscas. Na quarta o número em que estas três buscas obtiveram o mesmo resultado e a quinta o número de instâncias que não se conseguiu obter uma solução.

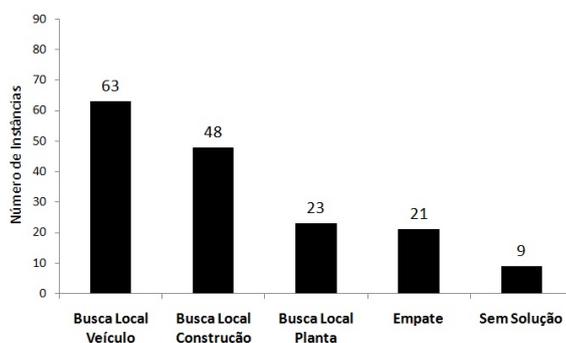


Figura 5.1. Comparativo entre as três Buscas Locais

Se compararmos as três primeiras buscas; veículo, construção e plantas; vemos que a veículos possui o melhor desempenho entre as três obtendo 63 melhores soluções, contra 48 e 23 das buscas construção e plantas. Ocorreu um empate entre os valores das três em 21 das instâncias e não foram conseguidas soluções em 9 instâncias.

Na Figura 5.2 incluímos na análise a busca local conjunta. Sendo agora a quarta coluna a busca local conjunta, a quinta o número de empates do valor da solução entre

as quatro buscas e a sexta o número de instâncias onde não se obteve uma solução. A comparação é realizada sobre a melhor solução encontrada entre as quatro buscas.

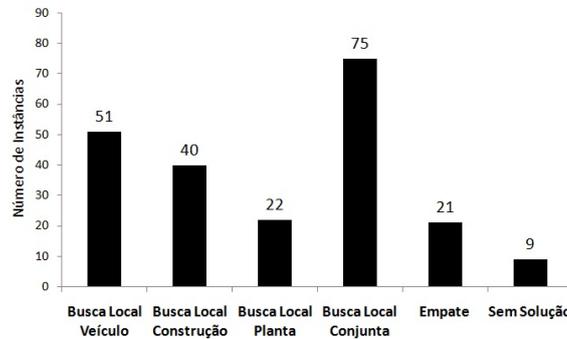


Figura 5.2. Comparativo entre as quatro Buscas Locais

Podemos observar que a busca conjunta obteve um resultado superior as outras três buscas, tendo 75 casos dos quais encontrou a melhor solução. Isto já era esperado, pois ela executa as três buscas locais em sequencia.

5.4 Modelos

Para solucionar os modelos de programação inteiro misto contido no capítulo 3 foi utilizado o programa CPLEX versão 12.5 utilizando 12 *threads* e limite de tempo de 1 hora (3600 segundos). O modelo que considera violações nas janelas de tempo nas construções, mediante penalizações, permite variações que fazem as janelas de tempo irem de 0 até as 24 horas. A penalidade por violar a janela de tempo permitida inicialmente é de 100.000.000. Ou seja, os parâmetros AC_c e PC_c possuem o valor de 100.000.000.

Os resultados dos testes para os modelos se encontram no apêndice C. A figura 5.3 possui um comparativo do número de soluções ótimas, viáveis, inviáveis ou ilimitadas, desconhecidas e inviáveis entre os modelos que não permitem e permitem violações nas janelas de tempo.

O modelo que não permite violações nas janelas de tempo obteve um número maior de soluções ótimas comparado ao resultado do modelo que permite violações, 9 soluções ótimas a mais. Contudo, o modelo que permite violação obteve 14 soluções viáveis a mais que o outro modelo, nenhuma solução Inviável ou Ilimitada e três soluções desconhecidas a menos. Esse desempenho é devido ao modelo encontrar soluções penalizando a função objetivo, tal fato aconteceu em 39 casos de suas soluções viáveis. Isso pode ser observado nas tabelas C.3 e C.4 quando as soluções são acima de 100.000.000.

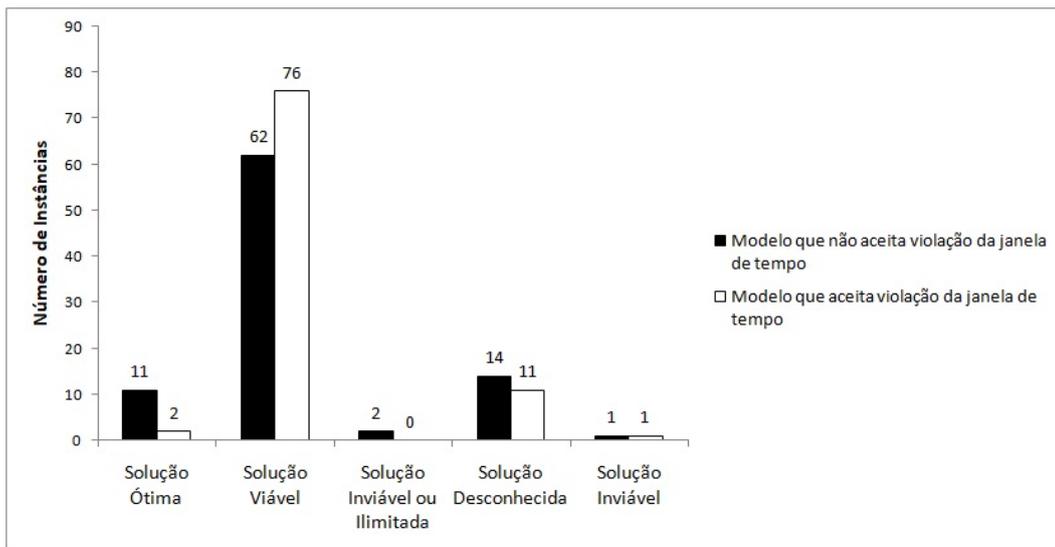


Figura 5.3. Comparativo entre modelos que permite e não permite violação nas janelas de tempo

A Figura 5.4 faz uma comparação entre o GAP obtido entre as soluções do modelo que não permite violações nas janelas de tempo e do modelo que permite violações. A primeira coluna é o número de soluções das quais o GAP do modelo que não permite violações fornece um melhor GAP, a segunda coluna é quando o modelo que permite violações possui um melhor GAP, e a terceira é quando ocorre um empate entre o GAP obtido entre os dois modelos.

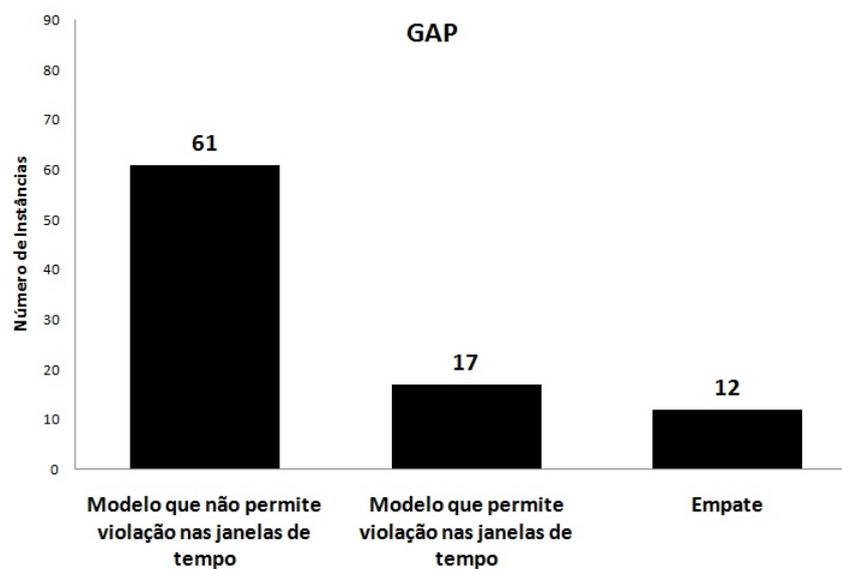


Figura 5.4. Comparativo do GAP obtido dos modelos que não permite violação nas janelas de tempo e o que permite

Ao analisar a Figura 5.4 vemos que o modelo que não permite violações possui um desempenho superior ao do modelo que permite violações ao ter um GAP superior em 44 instâncias do que o que permite. Isso se deve ao fato do modelo que permite violações gerar mais soluções viáveis, mas com penalizações na função objetivo. Este desempenho se deve também ao fato do modelo que não permite violações ter menos variáveis e restrições em comparação ao modelo que permite.

5.5 Modelo iniciando com resultado da heurística

O modelo que não permite violações nas janelas de tempo das construções obteve melhores resultados que o modelo que permite violações nas janelas de tempo das construções. Tendo isso em vista, realizamos experimentos iniciando esse modelo com a solução gerada pela heurística. A heurística consiste na execução do procedimento construtivo e os processos de viabilização 100 vezes, sobre a melhor solução dentre as 100 execuções realizamos a busca local conjunta. Os parâmetros de escolha dos veículos, construções e plantas utilizados foram Ve-2 Co-4 Pl-1. Os resultados se encontram no Apêndice C.3.

A Figura 5.5 contém um comparativo das soluções encontradas pelo modelo iniciado com a solução da heurística e o modelo que não. A Figura 5.6 contém um comparativo dos GAPs encontrados entre os dois modelos. Sendo a primeira coluna o número de instâncias onde o modelo iniciado com a heurística obtém um melhor GAP, a segunda coluna o número de instâncias que o modelo sem iniciar com a heurística possui um melhor GAP, e a última o número de casos dos quais os GAPs encontrados nos dois modelos foram iguais.

Como é verificado na Figura 5.5, o número de soluções ótimas, inviável ou ilimitada, e solução inviável se mantém nos dois modelos. A diferença é que o modelo que é iniciado com a solução da heurística encontra mais soluções viáveis que o modelo não iniciado. O modelo iniciado com a heurística encontra 8 soluções viáveis a mais que o modelo não iniciado.

Na Figura 5.6 o modelo que é iniciado com a solução da heurística possui em 33 casos um GAP melhor que o modelo não iniciado. Sendo que ele perde para o modelo que não é iniciado com a heurística em 31 casos, e se mantém igual em 26 casos.

Após essas análises vemos que a melhor estratégia para solucionar o problema é utilizar em conjunto a heurística e o modelo linear inteiro misto que não permite violação na janela de tempo das construções.

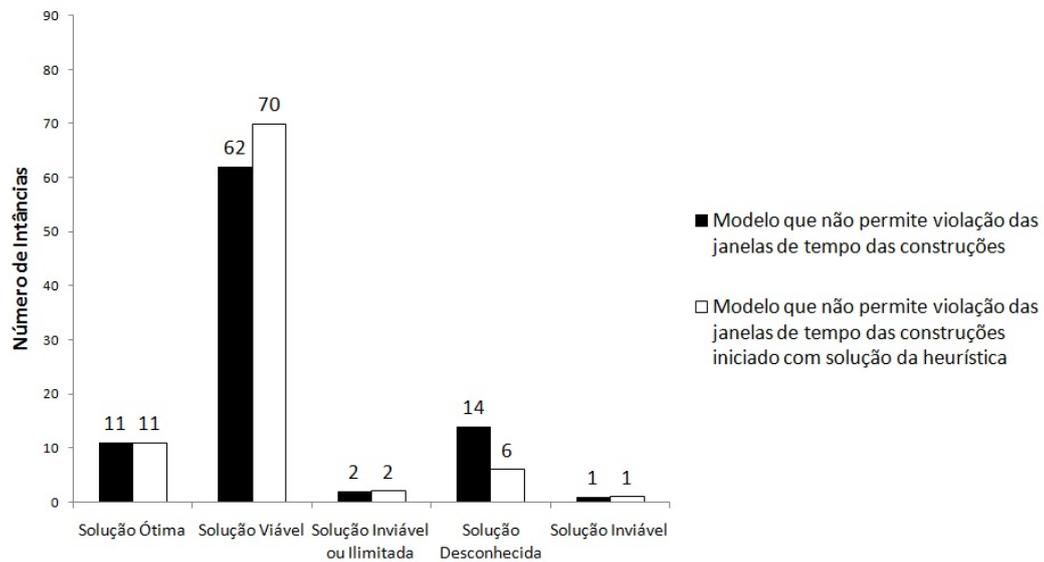


Figura 5.5. Comparativo entre modelos que não permite violações nas janelas de tempo iniciado com a solução da heurística e o modelo que não permite violações nas janelas de tempo

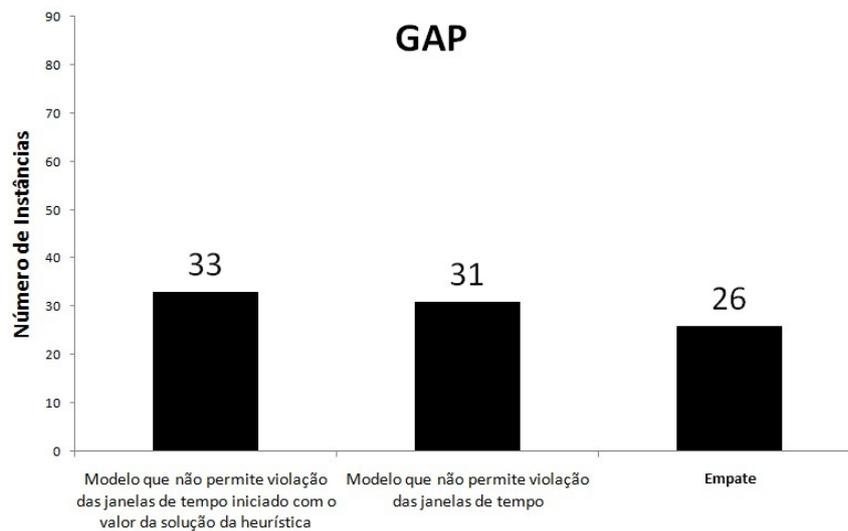


Figura 5.6. Comparativo do GAP obtido dos modelos que não permite violação nas janelas de tempo iniciado com a solução da heurística e o que não permite violação nas janelas de tempo

Capítulo 6

Conclusão

Nesse trabalho abordamos o problema do sequenciamento da produção e alocação de RMC. Realizamos uma abrangente revisão dos esforços já realizados sobre o tema e propomos uma nova abordagem. Essa abordagem realiza o sequenciamento e alocação do RMC no período em que empresas cooperam entre si mas não compartilham veículos entre elas.

Como não encontramos instâncias disponíveis na literatura, optamos por gerar um conjunto de instâncias para tratar o problema. Para criação dessas instâncias foi realizado uma grande pesquisa sobre quais parâmetros eram mais comuns nos estudos já realizados para serem usados em nosso estudos. Parâmetros como a velocidade dos veículos, tempo de descarregamento e carregamento de RMC, dentre outros.

Propomos uma heurística construtiva, dois processos de viabilização das soluções encontradas e três buscas locais. Para a heurística teve-se a necessidade da decisão da ordem de escolha dos veículos, construções e plantas. Também propomos dois modelos de programação inteiro misto para o problema. Um que não aceita violação nas janelas de tempo das construções e um que aceita violações nas janelas das construções mediante penalizações da função objetivo. Foi desenvolvido também um modelo que não aceita violação nas janelas de tempo das construções que é inicializado com uma solução da heurística.

A heurística obteve bons resultados conseguindo soluções viáveis em 90,0% das instâncias. O modelo que não permite violações nas janelas de tempo das construções se mostrou superior ao modelo que permite violações nas janelas de tempo mediante penalizações na função objetivo. Isso se deve ao primeiro modelo ter obtido menores GAPs e mais soluções ótimas. Contudo, a melhor abordagem foi o modelo que não permite violações nas janelas de tempo das construções iniciando com uma solução da heurística proposta. Esta abordagem obteve mais soluções viáveis e melhores GAPs

que o modelo que não permite violações nas janelas de tempo e não é inicializado com uma solução da heurística.

Muitos esforços podem ser realizados para continuar este estudo. Esforços como: instâncias mais fidedignas para o problema com um retrato mais fiel de como é o padrão de pedidos de RMC, a aceitação de que os veículos carregados de RMC fiquem ociosos certo período de tempo, veículos heterogêneos, veículos compartilhados entre as plantas, implementações de inequações válidas para o modelo, criação de um modelo indexado no tempo e uma abordagem com um cenário dinâmico.

Referências Bibliográficas

- Aitcin, P.-C. (2000). Cements of yesterday and today: Concrete of tomorrow. *Cement and Concrete Research*, 30(9):1349 – 1359. ISSN 0008-8846.
- Armstrong, T., editor (2015). *The Global Cement Report 11th Edition*. International Cement Review.
- Asbach, L.; Dorndorf, U. & Pesch, E. (2009). Analysis, modeling and solution of the concrete delivery problem. *European Journal of Operational Research*, 193(3):820 – 835. ISSN 0377-2217.
- Chen, H.-K.; Hsueh, C.-F. & Chang, M.-S. (2009). Production scheduling and vehicle routing with time windows for perishable food products. *Computers & Operations Research*, 36(7):2311 – 2319. ISSN 0305-0548.
- Cheng, M.-Y. & Hoc-Tran, D. (2015). Integrating chaotic initialized opposition multiple-objective differential evolution and stochastic simulation to optimize Ready-Mixed concrete truck dispatch schedule. *Journal of Management in Engineering*, 32(1).
- Damtoft, J.; Lukasik, J.; Herfort, D.; Sorrentino, D. & Gartner, E. (2008). Sustainable development and climate change initiatives. *Cement and Concrete Research*, 38(2):115 – 127. ISSN 0008-8846. Special Issue - The 12th International Congress on the Chemistry of Cement. Montreal, Canada.
- Durbin, M. T. (2003). *The Dance of the Thirty-Ton Trucks: Demand Dispatching in a Dynamic Environment*. Tese de doutorado, George Mason University.
- Feng, C.-W.; Cheng, T.-M. & Wu, H.-T. (2004). Optimizing the schedule of dispatching RMC trucks through genetic algorithms. *Automation in Construction*, 13(3):327 – 340. ISSN 0926-5805.

- Feng, C.-W. & Wu, H.-T. (2000). Using genetic algorithms to optimize the dispatching scheduling of RMC cars. *Proceedings of the 17th ISARC*, pp. 1 – 6.
- Feng, C.-W. & Wu, H.-T. (2006). Integrating fnga and cyclone to optimize the schedule of dispatching RMC trucks. *Automation in Construction*, 15(2):186 – 199. ISSN 0926-5805.
- Garcia, J. M.; Lozano, S.; Smith, K.; Kwok, T. & Villa, G. (2002). Coordinated scheduling of production and delivery from multiple plants and with time windows using genetic algorithms. *Proceedings of the 9th International Conference on Neural Information Processing 2002 (ICONIP '02)*, 3:1153 – 1158.
- Hanif, S. & Holvoet, T. (2014). Dynamic scheduling of ready mixed concrete delivery problem using delegate mas. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8473 LNAI:146 – 158.
- Hertz, A.; Uldry, M. & Widmer, M. (2012). Integer linear programming models for a cement delivery problem. *European Journal of Operational Research*, 222(3):623 – 631. ISSN 0377-2217.
- Imbabi, M. S.; Carrigan, C. & McKenna, S. (2012). Trends and developments in green cement and concrete technology. *International Journal of Sustainable Built Environment*, 1(2):194 – 216. ISSN 2212-6090.
- Kinable, J.; Wauters, T. & Berghe, G. V. (2014). The concrete delivery problem. *Computers & Operations Research*, 48:53 – 68. ISSN 0305-0548.
- Langevin, A. & Riope, D. (2005). *Logistics Systems: Design and Optimization*, capítulo Routing propane deliveries, pp. 299–322. Berlin Heidelberg.
- Lin, P.-C.; Wang, J.; Huang, S.-H. & Wang, Y.-T. (2010). Dispatching ready mixed concrete trucks under demand postponement and weight limit regulation. *Automation in Construction*, 19(6):798 – 807. ISSN 0926-5805.
- Liu, P.; Wang, L.; Ding, X. & Gao, X. (2010). Scheduling of dispatching ready mixed concrete trucks trough discrete particle swarm optimization. *2010 IEEE International Conference on Systems Man and Cybernetics (SMC)*, pp. 4086 – 4090.
- Liu, Z.; Zhang, Y. & Li, M. (2014). Integrated scheduling of ready-mixed concrete production and delivery. *Automation in Construction*, 48:31 – 43. ISSN 0926-5805.

- Lu, M. & Lam, H.-C. (2005). Optimized concrete delivery scheduling using combined simulation and genetic algorithms. Em *Winter Simulation Conference*, pp. 2572 – 2580.
- Lu, M. & Lam, H.-C. (2009). Simulation-optimization integrated approach to planning ready mixed concrete production and delivery: Validation and applications. *Winter Simulation Conference (WSC)*, pp. 2593 – 2604.
- Lu, M.; Tang, M. A. S. L.; & Ying, Y. C. (2003). HKCONSIM: A practical simulation solution to planning concrete plant operations in Hong Kong. *Journal of Construction Engineering and Management*, 129(5):547 – 554.
- Maghrebi, M.; Periaraj, V.; Waller, S. c. & Sammut, C. (2014a). Using benders decomposition for solving ready mixed concrete dispatching problems. Em *The 31st International Symposium on Automation and Robotics in Construction and Mining (ISARC 2014)*, pp. 680 – 688.
- Maghrebi, M.; Periaraj, V.; Waller, S. T. & Sammut, C. (2013a). Using column generation for solving largescale concrete dispatching problems. Relatório técnico UNSW-CSE-TR-201334, The University of New South Wales, School of Computer Science and Engineering.
- Maghrebi, M.; Periaraj, V.; Waller, S. T. & Sammut, C. (2014b). Solving ready-mixed concrete delivery problems: Evolutionary comparison between column generation and robust genetic algorithm. *2014 International Conference on Computing in Civil and Building Engineering*, pp. 1417 – 1424.
- Maghrebi, M.; Periaraj, V.; Waller, S. T. & Sammut, C. (2016). Column generation-based approach for solving large-scale ready mixed concrete delivery dispatching problems. *Computer-Aided Civil and Infrastructure Engineering*, 31(2):145 – 159. ISSN 1467-8667.
- Maghrebi, M.; Rey, D.; Waller, S. T. & Sammut, C. (2014c). Reducing the number of decision variables in ready mixed concrete for optimally solving small instances in a practical time. *CSCE 2014 General Conference*.
- Maghrebi, M.; Sammut, C.; & Waller, T. (2013b). Reconstruction of an expert's decision making expertise in concrete dispatching by machine learning. *Journal of Civil Engineering and Architecture*, 7(20):1540 – 1547.

- Maghrebi, M.; Sammut, C. & Waller, S. T. (2015). Feasibility study of automatically performing the concrete delivery dispatching through machine learning techniques. *Engineering, Construction and Architectural Management*, 22(5):573 – 590.
- Maghrebi, M. & Travis Waller, S. (2014). Exploring experts decisions in concrete delivery dispatching systems using bayesian network learning techniques. Em *2014 Second International Conference on Artificial Intelligence, Modelling and Simulation*, pp. 103 – 108.
- Maghrebi, M.; Travis Waller, S. & Sammut, C. (2014d). Assessing the accuracy of expert-based decisions in dispatching ready mixed concrete. *Journal of Construction Engineering and Management*, 140(6).
- Maghrebi, M.; Waller, S. T. & Sammut, C. (2014e). Sequential meta-heuristic approach for solving large-scale ready-mixed concrete dispatching problems. *Journal of Computing in Civil Engineering*, 30(1).
- Maghrebi, M.; Waller, T. S. & Sammut, C. (2013c). Scheduling concrete delivery problems by a robust meta heuristic method. Em *2013 European Modelling Symposium (EMS)*, pp. 375 – 380.
- Manne, A. S. (1960). On the job-shop scheduling problem. *Operations Research*, 8(2):219–223.
- Matsatsinis, N. F. (2004). Towards a decision support system for the ready concrete distribution system: A case of a greek company. *European Journal of Operational Research*, 152(2):487 – 499. ISSN 0377-2217. New Technologies in Transportation Systems.
- Mayteekrieangkrai, N. & Wongthatsanekorn, W. (2015). Optimized ready mixed concrete truck scheduling for uncertain factors using bee algorithm. *Songklanakarin Journal of Science and Technology*, 37(2):221 – 230. ISSN 01253395.
- Misir, M.; Vancroonenburg, W.; Verbeeck, K. & Berghe, G. V. (2011). A selection hyper-heuristic for scheduling deliveries of ready-mixed concrete. *Proceedings of the Metaheuristics International Conference (MIC 2011)*, pp. 289 – 298.
- Narayanan, P. K.; Rey, D.; Maghrebi, M. & Waller, S. T. (2015). Using lagrangian relaxation to solve ready mixed concrete dispatching problems. *Transportation Research Record*, pp. 84 – 90.

- Naso, D.; Surico, M. & Turchiano, B. (2007a). Reactive scheduling of a distributed network for the supply of perishable products. *IEEE Transactions on Automation Science and Engineering*, 4(3):407–423.
- Naso, D.; Surico, M.; Turchiano, B. & Kaymak, U. (2004). Just-in-time production and delivery in supply chains: a hybrid evolutionary approach. *2004 IEEE International Conference on Systems, Man and Cybernetics*, 2:1932 – 1937.
- Naso, D.; Surico, M.; Turchiano, B. & Kaymak, U. (2007b). Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. *European Journal of Operational Research*, 177(3):2069 – 2099. ISSN 0377-2217.
- Park, M.; Kim, W.-Y.; Lee, H.-S. & Han, S. (2011). Supply chain management model for ready mixed concrete. *Automation in Construction*, 20(1):44 – 55. ISSN 0926-5805. Global convergence in construction.
- Ravetti, M. G. (2007). *Algoritmos para o problema de sequenciamento com máquinas paralelas e tempos de preparação dependentes da sequência*. Tese de doutorado, Universidade Federal de Minas Gerais.
- Schmid, M. V. (2007). *Trucks in Movement: Hybridization of Exact Approaches and Variable Neighborhood Search for the Delivery of Ready-Mixed Concrete*. Tese de doutorado, University of Vienna.
- Schmid, V.; Doerner, K. F.; Hartl, R. F. & Salazar-González, J.-J. (2010). Hybridization of very large neighborhood search for ready-mixed concrete delivery problems. *Computers & Operations Research*, 37(3):559 – 574. ISSN 0305-0548. Hybrid Metaheuristics.
- Schmid, V.; Doerner, K. F.; Hartl, R. F.; Savelsbergh, M. W. P. & Stoecher, W. (2009). A hybrid solution approach for ready-mixed concrete delivery. *Transportation Science*, 43(1):70 – 85. ISSN 1526-5447.
- Shi, C. b. & Wang, X. (2012). Scheduling model of dispatching ready mixed concrete trucks based on GA. *Advances in Information Sciences and Service Sciences*, 4(8):131 – 136. ISSN 19763700.
- Silva, C. A.; Faria, J. M.; Abrantes, P.; Sousa, J. M. C.; Surico, M. & Naso, D. (2005). Concrete delivery using a combination of GA and ACO. *44th IEEE Conference on Decision and Control and the European Control Conference 2005*, pp. 7633 – 7638.

- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254 – 265. ISSN 0030-364X.
- Srichandum, S. & Rujirayanyong, T. (2010). Production scheduling for dispatching ready mixed concrete trucks using bee colony optimization. *American J. of Engineering and Applied Sciences* 3, 3(1):7 – 14.
- Syverson, C. (2008). Markets: Ready-mixed concrete. *The Journal of Economic Perspectives*, 1(22):217 – 234.
- Tommelein, I. D. & Li, A. E. Y. (1999). Just-in-time concrete delivery: mapping alternatives for vertical supply chain integration. *Seventh Conference of the International Group for Lean Construction (IGLC-7)*, pp. 97 – 108.
- Yan, S. & Lai, W. (2007). An optimal scheduling model for ready mixed concrete supply with overtime considerations. *Automation in Construction*, 16(6):734 – 744. ISSN 0926-5805.
- Yan, S.; Lai, W. & Chen, M. (2008). Production scheduling and truck dispatching of ready mixed concrete. *Transportation Research Part E: Logistics and Transportation Review*, 44(1):164 – 179. ISSN 1366-5545.
- Yan, S.; Lin, H. C. & Jiang, X. Y. (2012). A planning model with a solution algorithm for ready mixed concrete production and truck dispatching under stochastic travel times. *Engineering Optimization*, 44(4):427 – 447.
- Yan, S.; Lin, H.-C. & Liu, Y.-C. (2011). Optimal schedule adjustments for supplying ready mixed concrete following incidents. *Automation in Construction*, 20(8):1041 – 1050. ISSN 0926-5805.
- Zayed, T. M. & Halpin, D. (2001). Simulation of concrete batch plant production. *Journal of Construction Engineering and Management*, 127(2):132 – 141.
- Zhang, Y.; Li, M. & Liu, Z. (2011). Vehicle scheduling and dispatching of ready mixed concrete. Em *2011 Fourth International Workshop on Advanced Computational Intelligence (IWACI)*, pp. 465 – 472.

Apêndice A

Análise de uma execução da heurística construtiva seguida dos processos de viabilização

A seguir as tabelas A.1, A.2, A.3, A.4 e A.5 possuem os valores percentuais do número de soluções viáveis encontradas para os conjuntos de instâncias p-C101-P1C5, p-C101-P1C8, p-C101-P2C10, p-C101-P2C15 e p-C101-P3C15. Para todas as instâncias foi executado o algoritmo apenas uma vez, menos nos casos que possuía um parâmetro de escolha aleatório (Ve-4, Co-4 ou Pl-4). Nesses últimos casos foi realizado 30 execuções com a mesma instância e computado a média de soluções viáveis encontradas dentre as 30. Na primeira linha se tem o valor do parâmetro de escolha dos veículos, na segunda linha o parâmetro de escolha das plantas e na primeira coluna o parâmetro de escolha das construções. Estes parâmetros são encontrados na seção 4.2. O restante contém o percentual de soluções viáveis encontrados com a configuração dos parâmetros respectivos.

Tabela A.1. Porcentagem de instâncias p-C101-P1C5 com soluções viáveis

	Ve-1				Ve-2				Ve-3				Ve-4			
	Pl-1	Pl-2	Pl-3	Pl-4												
Co-1	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
Co-2	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
Co-3	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
Co-4	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%

Tabela A.2. Porcentagem de intâncias p-C101-P1C8 com soluções viáveis

	Ve-1				Ve-2				Ve-3				Ve-4			
	Pl-1	Pl-2	Pl-3	Pl-4												
Co-1	55,6%	55,6%	55,6%	55,6%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	55,6%	66,7%	55,6%	63,3%
Co-2	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	63,7%	66,7%	66,7%
Co-3	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%
Co-4	62,2%	55,6%	57,8%	54,1%	66,7%	63,7%	57,8%	66,7%	49,6%	65,6%	61,5%	66,7%	66,7%	65,9%	57,4%	51,5%

Tabela A.3. Porcentagem de intâncias p-C101-P2C10 com soluções viáveis

	Ve-1				Ve-2				Ve-3				Ve-4			
	Pl-1	Pl-2	Pl-3	Pl-4												
Co-1	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%
Co-2	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%
Co-3	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%
Co-4	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%

Tabela A.4. Porcentagem de intâncias p-C101-P2C15 com soluções viáveis

	Ve-1				Ve-2				Ve-3				Ve-4			
	Pl-1	Pl-2	Pl-3	Pl-4	Pl-1	Pl-2	Pl-3	Pl-4	Pl-1	Pl-2	Pl-3	Pl-4	Pl-1	Pl-2	Pl-3	Pl-4
Co-1	88,9%	88,9%	100,0%	100,0%	100,0%	100,0%	100,0%	99,6%	100,0%	100,0%	100,0%	99,6%	100,0%	95,6%	100,0%	98,1%
Co-2	88,9%	88,9%	100,0%	87,0%	100,0%	88,9%	100,0%	96,7%	100,0%	100,0%	100,0%	97,4%	95,6%	88,5%	100,0%	90,0%
Co-3	100,0%	89,9%	100,0%	93,0%	88,9%	100,0%	88,9%	100,0%	88,9%	88,9%	100,0%	96,3%	98,9%	92,6%	98,1%	93,3%
Co-4	94,8%	92,6%	93,0%	91,1%	97,4%	95,9%	95,6%	93,3%	98,1%	94,4%	97,0%	98,9%	97,0%	92,2%	96,3%	94,1%

Tabela A.5. Porcentagem de intâncias p-C101-P3C15 com soluções viáveis

	Ve-1				Ve-2				Ve-3				Ve-4			
	Pl-1	Pl-2	Pl-3	Pl-4												
Co-1	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
Co-2	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
Co-3	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
Co-4	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%

Como se pode observar todos os conjuntos p-C101-P1C5 e p-C101-P3C15 obtiveram para todas as instâncias soluções viáveis, sendo assim, que os parâmetros não tiveram influência na obtenção das soluções viáveis. As intâncias p-C101-P2C10 obtiveram todas o mesmo percentual de soluções viáveis, indicando que a escolha de parâmetros não tem influência. As intâncias p-C101-P1C8 possuem em sua maioria a mesma porcentagem de soluções viáveis, sendo que as escolhas Co-1 e Co-4, com Ve-1 ou Ve-4, apresenta maiores variações no número de soluções viáveis encontradas. Aconteceu a maior variação entre a porcentagem de soluções viáveis nas intâncias p-C101-P2C15, sendo dentre os cinco grupos a que mais sofreu influência das configurações dos parâmetros de escolha.

Na tabela A.6 se tem os valores para todas as intâncias p-C101.

Tabela A.6. Porcentagem de intâncias p-C101 com soluções viáveis

	Ve-1				Ve-2				Ve-3				Ve-4			
	Pl-1	Pl-2	Pl-3	Pl-4												
Co-1	86,7%	86,7%	88,9%	88,9%	91,1%	91,1%	91,1%	91,0%	91,1%	91,1%	91,1%	91,0%	88,9%	90,2%	88,9%	90,1%
Co-2	88,9%	88,9%	91,1%	88,5%	91,1%	88,9%	91,1%	90,4%	91,1%	91,1%	91,1%	90,6%	90,2%	88,2%	91,1%	89,1%
Co-3	91,1%	88,9%	91,1%	89,7%	88,9%	91,1%	88,9%	91,1%	88,9%	88,9%	91,1%	90,4%	90,9%	89,6%	90,7%	89,8%
Co-4	89,2%	87,4%	87,9%	86,8%	90,6%	89,7%	88,4%	89,8%	87,3%	89,8%	89,5%	90,9%	90,5%	89,4%	88,5%	86,9%

Nesta tabela podemos observar que as melhores configurações para solucionar as intâncias p-C101 foram de 91,1% e se encontram em varias combinações de parâmetros de escolha. A variação na performance encontradas são devidas principalmente aos resultados das intâncias p-C101-P1C8 e p-C101-P2C15 das quais os fatores influenciaram mais intensamente na obtenção de soluções viáveis.

As tabelas A.7, A.8, A.9, A.10 e A.11 possuem os valores percentuais do número de soluções viáveis encontradas para os conjuntos de instâncias p-R101-P1C5, p-R101-P1C8, p-R101-P2C10, p-R101-P2C15 e p-R101-P3C15.

Tabela A.7. Porcentagem de intâncias p-R101-P1C5 com soluções viáveis

	V1				V2				V3				V4			
	P1	P2	P3	P4												
C1	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%
C2	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%
C3	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%
C4	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%	88,9%

Tabela A.8. Porcentagem de intâncias p-R101-P1C8 com soluções viáveis

	V1				V2				V3				V4			
	P1	P2	P3	P4												
C1	44,4%	44,4%	44,4%	44,4%	55,6%	55,6%	55,6%	55,6%	55,6%	55,6%	55,6%	55,6%	66,7%	66,7%	66,7%	60,7%
C2	44,4%	44,4%	44,4%	44,4%	55,6%	55,6%	55,6%	55,6%	55,6%	55,6%	55,6%	55,6%	66,7%	66,7%	61,1%	59,6%
C3	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	66,7%	65,9%	62,6%	66,7%	63,7%
C4	57,4%	52,6%	45,6%	58,5%	59,6%	65,2%	55,6%	57,4%	52,2%	62,2%	54,4%	61,5%	50,4%	65,6%	55,9%	54,8%

Tabela A.9. Porcentagem de intâncias p-R101-P2C10 com soluções viáveis

	V1				V2				V3				V4			
	P1	P2	P3	P4												
C1	100,0%	100,0%	88,9%	100,0%	100,0%	100,0%	88,9%	100,0%	100,0%	100,0%	88,9%	100,0%	100,0%	100,0%	100,0%	100,0%
C2	100,0%	100,0%	88,9%	100,0%	100,0%	100,0%	88,9%	100,0%	100,0%	100,0%	88,9%	100,0%	100,0%	100,0%	100,0%	100,0%
C3	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%	100,0%
C4	96,7%	94,4%	84,1%	93,3%	94,8%	95,9%	93,0%	88,9%	100,0%	93,7%	96,3%	94,1%	98,1%	95,6%	79,6%	94,4%

Tabela A.10. Porcentagem de intâncias p-R101-P2C15 com soluções viáveis

	V1				V2				V3				V4			
	P1	P2	P3	P4												
C1	44,4%	44,4%	44,4%	53,3%	44,4%	77,8%	44,4%	73,3%	44,4%	77,8%	44,4%	62,2%	48,5%	77,8%	37,4%	66,3%
C2	55,6%	44,4%	33,3%	59,3%	55,6%	66,7%	44,4%	73,3%	66,7%	66,7%	44,4%	73,3%	64,1%	58,5%	40,0%	63,7%
C3	55,6%	66,7%	55,6%	61,9%	55,6%	66,7%	44,4%	58,9%	55,6%	66,7%	44,4%	66,7%	48,9%	76,7%	44,1%	67,0%
C4	45,6%	60,0%	45,2%	44,8%	54,4%	67,0%	50,4%	65,2%	55,6%	59,3%	40,4%	67,0%	50,4%	64,4%	45,9%	59,6%

Tabela A.11. Porcentagem de intâncias p-R101-P3C15 com soluções viáveis

	V1				V2				V3				V4			
	P1	P2	P3	P4												
C1	88,9%	100,0%	88,9%	97,0%	88,9%	100,0%	88,9%	99,3%	88,9%	100,0%	88,9%	97,8%	88,9%	97,4%	81,1%	94,1%
C2	88,9%	100,0%	77,8%	89,6%	88,9%	100,0%	77,8%	90,4%	88,9%	100,0%	77,8%	90,4%	88,9%	98,9%	88,1%	88,9%
C3	88,9%	100,0%	88,9%	90,4%	88,9%	100,0%	77,8%	93,7%	88,9%	100,0%	77,8%	90,7%	88,9%	100,0%	85,2%	90,4%
C4	92,6%	98,9%	87,4%	93,7%	88,9%	98,5%	84,8%	93,7%	90,7%	98,5%	90,4%	93,0%	90,7%	99,3%	91,1%	93,3%

As intâncias p-R101-P1C5 obtiveram todas o mesmo desempenho de 88,9%, sendo assim, os parâmetros de escolha não tem efeito sobre a obtenção de soluções viáveis nessas instâncias. As intâncias p-R101-P1C8 o parâmetro Co-3 nos casos com Ve-1, Ve-2 e Ve-3 influenciam positivamente o número de soluções encontradas em detrimento de Co-1, Co-2 e Co-4. Nesses casos se encontram o maior valor de soluções viáveis que é de 66,7%. Esse valor também é encontrado com Ve-4, selecionando Pl-1 com Co-1 ou Co-2, Pl-2 com Co-1 ou Co-2, e Pl-3 com Co-1. O parâmetro Ve-1 combinado com Co-1 e Co-2 obtiveram os piores resultados. As intâncias p-R101-P2C10 obtiveram 100% na maioria dos casos, sendo que a combinação de Ve-1, Ve-2 ou Ve-3, com Co-1 ou Co-2, com Pl-3 gera uma queda de desempenho da obtenção de soluções viáveis. Com o parâmetro Co-4 não se obteve o valor de 100% em nenhum dos casos. As intâncias p-R101-P2C15, caso se avaliar os parâmetros Co-1, Co-2, Co-3 e Co-4 fixos, observasse em Ve-1, Ve-2, Ve-3 e Ve-4 que o Pl-3 tem um desempenho inferior ou igual a Pl-1, Pl-2 e Pl-4. Sendo Ve-1 Pl-3 Co-4 e Ve-1 Pl-4 Co-4 as únicas excessões. Os maiores valores se encontram na escolha Co-1 com Pl-2 e Ve-2, Ve-3 ou Ve-4. As intâncias p-R101-P3C15 obtiveram valor de 100% com o Pl-2 com Co-1, Co-2 ou Co-3 para Ve-1, Ve-2 e Ve-3, e Ve-4 Pl-2 Co-3.

Na tabela A.12 se tem os valores para todas as intâncias p-R101.

Tabela A.12. Porcentagem de intâncias p-R101 com soluções viáveis

	V1				V2				V3				V4			
	P1	P2	P3	P4												
C1	73,3%	75,6%	71,1%	76,7%	75,6%	84,4%	73,3%	83,4%	75,6%	84,4%	73,3%	80,9%	78,6%	86,1%	74,8%	82,0%
C2	75,6%	75,6%	66,7%	76,4%	77,8%	82,2%	71,1%	81,6%	80,0%	82,2%	71,1%	81,6%	81,7%	82,6%	75,6%	80,2%
C3	80,0%	84,4%	80,0%	81,6%	80,0%	84,4%	75,6%	81,6%	80,0%	84,4%	75,6%	82,6%	78,5%	85,6%	77,0%	82,0%
C4	76,2%	79,0%	70,2%	75,9%	77,3%	83,1%	74,5%	78,8%	77,5%	80,5%	74,1%	80,9%	75,7%	82,7%	72,3%	78,2%

As intâncias p-R101 tem como a melhor escolha de parâmetros a configuração

Ve-4 Pl-2 Co-1 com 86.1% de soluções viáveis encontradas.

Na tabela A.13 se tem os valores para todas as intâncias p.

Tabela A.13. Porcentagem de intâncias com soluções viáveis

	Ve-1				Ve-2				Ve-3				Ve-4			
	Pl-1	Pl-2	Pl-3	Pl-4												
Co-1	80,0%	81,1%	80,0%	82,8%	83,3%	87,8%	82,2%	87,2%	83,3%	87,8%	82,2%	86,0%	83,7%	88,2%	81,9%	86,0%
Co-2	82,2%	82,2%	78,9%	82,5%	84,4%	85,6%	81,1%	86,0%	85,6%	86,7%	81,1%	86,1%	86,0%	85,4%	83,4%	84,7%
Co-3	85,6%	85,6%	85,6%	85,6%	84,4%	87,8%	82,2%	86,4%	84,4%	86,7%	83,3%	86,5%	84,7%	87,6%	83,9%	85,9%
Co-4	82,7%	83,2%	79,1%	81,3%	84,0%	86,4%	81,5%	84,3%	82,4%	85,1%	81,8%	85,9%	83,1%	86,1%	80,4%	82,6%

A melhor escolha de configuração de parâmetros para as instâncias p com uma execução da heurística como se verifica na tabela A.13 é Ve-4 Pl-2 Co-1 com 88,2%. Sendo Ve-1 Pl-3 Co-2 com 78,9% a pior escolha. Esta escolha de parâmetros se deve mais as intâncias p-R101 do que as p-C101 que não sofrem muita influência dos parâmetros de escolha.

Apêndice B

Resultados da execução da heurística executada 100 vezes seguida das buscas locais

Temos nas tabelas B.1 e B.2 o valor encontrado para a execução de 100 iterações da heurística construtiva com os parâmetros Ve-2 Co-4 Pl-1 seguido de uma busca local para as intâncias p-C101 e p-R101 respectivamente. Na primeira coluna se tem o nome das intâncias, na segunda coluna o valor da solução da execução da heurística utilizando a busca local veículos, na terceira utilizando a busca local construções, na quarta utilizando a busca local planta e na quinta utilizando a busca local conjunta. Valendo que a busca local conjunta é a execução das buscas locais anteriores em sequencia.

Tabela B.1. Resultados da heurística executada cem vezes com os parâmetros Ve-2 Co-4 Pl-1 seguido de buscas locais para instâncias p-C101

Instâncias	Buscas Locais			
	Veículo	Construção	Planta	Conjunta
p-C101-P1C5-I2-V1	4100	4100	4100	4100
p-C101-P1C5-I2-V2	3383	3383	3383	3383
p-C101-P1C5-I2-V3	4003	4003	4003	4003
p-C101-P1C5-I3-V1	2802	2802	2812	2802
p-C101-P1C5-I3-V2	3061	3061	3061	3061
p-C101-P1C5-I3-V3	3860	3860	3860	3860
p-C101-P1C5-I4-V1	3521	3521	3589	3521
p-C101-P1C5-I4-V2	3450	3435	3460	3450
p-C101-P1C5-I4-V3	3063	3059	3063	3059
p-C101-P1C8-I2-V1	—	—	—	—
p-C101-P1C8-I2-V2	—	—	—	—
p-C101-P1C8-I2-V3	5817	5817	5817	5817
p-C101-P1C8-I3-V1	5362	5340	5402	5344
p-C101-P1C8-I3-V2	—	—	—	—
p-C101-P1C8-I3-V3	5427	5427	5427	5427
p-C101-P1C8-I4-V1	4991	4993	5003	4991
p-C101-P1C8-I4-V2	5538	5540	5540	5538
p-C101-P1C8-I4-V3	5417	5417	5422	5411
p-C101-P2C10-I2-V1	7470	7475	7470	7470
p-C101-P2C10-I2-V2	7561	7566	7566	7561
p-C101-P2C10-I2-V3	—	—	—	—
p-C101-P2C10-I3-V1	6551	6552	6552	6551
p-C101-P2C10-I3-V2	6511	6517	6528	6511
p-C101-P2C10-I3-V3	7104	7111	7140	7104
p-C101-P2C10-I4-V1	5969	5965	5994	5965
p-C101-P2C10-I4-V2	6264	6271	6315	6264
p-C101-P2C10-I4-V3	7215	7224	7266	7209
p-C101-P2C15-I2-V1	10287	10287	10287	10287
p-C101-P2C15-I2-V2	10292	10322	10313	10292
p-C101-P2C15-I2-V3	11123	11132	11131	11123
p-C101-P2C15-I3-V1	9834	9883	9906	9834
p-C101-P2C15-I3-V2	9821	9803	9872	9815
p-C101-P2C15-I3-V3	11609	11614	11685	11607
p-C101-P2C15-I4-V1	10941	10956	10996	10941
p-C101-P2C15-I4-V2	9977	9926	9983	9922
p-C101-P2C15-I4-V3	9136	9113	9149	9113
p-C101-P3C15-I2-V1	11031	11040	11039	11031
p-C101-P3C15-I2-V2	10265	10271	10271	10260
p-C101-P3C15-I2-V3	11071	11082	11077	11071
p-C101-P3C15-I3-V1	10281	10281	10291	10281
p-C101-P3C15-I3-V2	10814	10856	10822	10814
p-C101-P3C15-I3-V3	10600	10605	10634	10600
p-C101-P3C15-I4-V1	9193	9195	9281	9181
p-C101-P3C15-I4-V2	9636	9649	9653	9636
p-C101-P3C15-I4-V3	10182	10217	10293	10182

Tabela B.2. Resultados da heurística executada cem vezes com os parâmetros Ve-2 Co-4 Pl-1 seguido de buscas locais para instâncias p-R101

Instâncias	Buscas Locais			
	Veículo	Construção	Planta	Conjunta
p-R101-P1C5-I2-V1	—	—	—	—
p-R101-P1C5-I2-V2	3294	3294	3294	3294
p-R101-P1C5-I2-V3	4253	4253	4253	4253
p-R101-P1C5-I3-V1	3449	3449	3449	3449
p-R101-P1C5-I3-V2	3824	3824	3824	3824
p-R101-P1C5-I3-V3	3690	3690	3690	3690
p-R101-P1C5-I4-V1	3951	3951	3951	3951
p-R101-P1C5-I4-V2	3980	3980	3980	3980
p-R101-P1C5-I4-V3	2784	2784	2784	2784
p-R101-P1C8-I2-V1	—	—	—	—
p-R101-P1C8-I2-V2	—	—	—	—
p-R101-P1C8-I2-V3	6399	6399	6399	6399
p-R101-P1C8-I3-V1	5544	5544	5544	5544
p-R101-P1C8-I3-V2	—	—	—	—
p-R101-P1C8-I3-V3	5479	5479	5479	5479
p-R101-P1C8-I4-V1	5467	5463	5467	5463
p-R101-P1C8-I4-V2	5679	5679	5689	5679
p-R101-P1C8-I4-V3	5572	5572	5651	5572
p-R101-P2C10-I2-V1	7842	7842	7842	7842
p-R101-P2C10-I2-V2	6932	6932	6932	6932
p-R101-P2C10-I2-V3	6885	6884	6885	6884
p-R101-P2C10-I3-V1	7747	7744	7752	7741
p-R101-P2C10-I3-V2	6724	6688	6701	6687
p-R101-P2C10-I3-V3	7012	7012	7076	7012
p-R101-P2C10-I4-V1	7055	7059	7059	7055
p-R101-P2C10-I4-V2	6569	6567	6594	6563
p-R101-P2C10-I4-V3	6431	6443	6485	6429
p-R101-P2C15-I2-V1	11564	11573	11583	11564
p-R101-P2C15-I2-V2	10152	10152	10161	10152
p-R101-P2C15-I2-V3	11878	11851	11884	11845
p-R101-P2C15-I3-V1	10666	10666	10690	10666
p-R101-P2C15-I3-V2	—	—	—	—
p-R101-P2C15-I3-V3	11278	11262	11278	11260
p-R101-P2C15-I4-V1	9453	9432	9466	9442
p-R101-P2C15-I4-V2	9964	9845	10061	9954
p-R101-P2C15-I4-V3	10562	10640	10730	10557
p-R101-P3C15-I2-V1	11345	11347	11347	11345
p-R101-P3C15-I2-V2	12026	12029	12026	12022
p-R101-P3C15-I2-V3	10653	10654	10658	10649
p-R101-P3C15-I3-V1	11156	11170	11246	11152
p-R101-P3C15-I3-V2	10499	10504	10506	10494
p-R101-P3C15-I3-V3	10505	10430	10484	10444
p-R101-P3C15-I4-V1	10320	10283	10359	10281
p-R101-P3C15-I4-V2	9711	9795	9797	9689
p-R101-P3C15-I4-V3	10211	10219	10354	10211

Apêndice C

Resultados da execução dos modelos

Apresentaremos os resultados da execução das instâncias criadas primeiramente para o modelo não permitindo violação nas janelas de tempo das construções. Em seguida os resultados do modelo que permite violação das janelas de tempo das construções mediante penalização na função objetivo é apresentado. Por fim os teste do modelo que não permite violações nas janelas de tempo das construções mas é iniciado com a solução da heurística criada neste trabalho é demonstrado. A heurística consiste na execução de 100 vezes o procedimento construtivo seguidos dos dois de viabilidade, sobre a melhor solução encontrada dentre as 100 execuções é realizado a busca local conjunta.

C.1 Testes do modelo que não permite violação das janelas de tempo das construções

A tabela C.1 apresenta os resultados para as instâncias p-C101 e a tabela C.2 para as instâncias p-R101. A primeira coluna se possui o nome das intâncias, a segunda o status da solução encontrada, a terceira a solução primal encontrada, a quarta a solução dual, a quinta o valor do GAP encontrado, e na última o tempo em segundos da execução. Quando não se obteve um valor de solução o carácter “ - ” é sinalizado nas colunas três, quatro e cinco.

Foram obtidas 11 soluções ótimas, 62 soluções viáveis, 2 inviável ou ilimitadas, 14 com o status de desconhecido, e 1 foi considerada inviável. Sendo que as soluções ótimas foram encontradas com instâncias contendo apenas uma planta. Foi observado

Tabela C.1. Resultados do modelo que não permite violações nas janelas de tempo para instâncias p-C101

Instância	Status	Solução Primal	Solução Dual	GAP	Tempo (segundos)
p-C101-P1C5-I2-V1	Ótimo	4100,0	4100,0	0,0%	5
p-C101-P1C5-I2-V2	Ótimo	3383,0	3383,0	0,0%	39
p-C101-P1C5-I2-V3	Ótimo	4003,0	4003,0	0,0%	56
p-C101-P1C5-I3-V1	Viável	2802,0	2772,0	1,1%	3600
p-C101-P1C5-I3-V2	Ótimo	3061,0	3061,0	0,0%	27
p-C101-P1C5-I3-V3	Ótimo	3860,0	3860,0	0,0%	1040
p-C101-P1C5-I4-V1	Viável	3521,0	3460,0	1,7%	3601
p-C101-P1C5-I4-V2	Viável	3430,0	3320,0	3,2%	3600
p-C101-P1C5-I4-V3	Viável	3055,0	3000,0	1,8%	3601
p-C101-P1C8-I2-V1	Inviável ou Ilimitado	-	-	-	328
p-C101-P1C8-I2-V2	Desconhecido	-	-	-	3600
p-C101-P1C8-I2-V3	Viável	5816,0	5736,2	1,4%	3600
p-C101-P1C8-I3-V1	Viável	5332,0	5231,0	1,9%	3600
p-C101-P1C8-I3-V2	Desconhecido	-	-	-	3603
p-C101-P1C8-I3-V3	Viável	5427,0	5377,0	0,9%	3601
p-C101-P1C8-I4-V1	Ótimo	4979,0	4978,6	0,0%	1136
p-C101-P1C8-I4-V2	Viável	5528,0	5370,0	2,9%	3600
p-C101-P1C8-I4-V3	Viável	5415,0	4880,0	9,9%	3600
p-C101-P2C10-I2-V1	Viável	7470,0	7380,0	1,2%	3607
p-C101-P2C10-I2-V2	Viável	7564,0	7300,0	3,5%	3601
p-C101-P2C10-I2-V3	Inviável	-	-	-	3
p-C101-P2C10-I3-V1	Viável	6617,0	6310,0	4,6%	3601
p-C101-P2C10-I3-V2	Viável	6529,0	6340,0	2,9%	3600
p-C101-P2C10-I3-V3	Viável	7210,0	6820,0	5,4%	3601
p-C101-P2C10-I4-V1	Viável	6042,0	5781,0	4,3%	3601
p-C101-P2C10-I4-V2	Viável	6265,0	6110,0	2,5%	3601
p-C101-P2C10-I4-V3	Viável	7193,0	7000,0	2,7%	3602
p-C101-P2C15-I2-V1	Viável	10339,0	10032,0	3,0%	3601
p-C101-P2C15-I2-V2	Desconhecido	-	-	-	3602
p-C101-P2C15-I2-V3	Viável	11190,0	10800,0	3,5%	3602
p-C101-P2C15-I3-V1	Viável	10250,0	-8363,7	181,6%	3601
p-C101-P2C15-I3-V2	Viável	10248,0	9390,0	8,4%	3601
p-C101-P2C15-I3-V3	Viável	11846,0	-9666,3	181,6%	3601
p-C101-P2C15-I4-V1	Viável	10936,0	10710,0	2,1%	3601
p-C101-P2C15-I4-V2	Viável	10365,0	-5888,0	156,8%	3601
p-C101-P2C15-I4-V3	Desconhecido	-	-	-	3601
p-C101-P3C15-I2-V1	Viável	11126,0	-9748,7	187,6%	3601
p-C101-P3C15-I2-V2	Viável	10462,0	-9339,4	189,3%	3603
p-C101-P3C15-I2-V3	Viável	11082,0	10860,0	2,0%	3604
p-C101-P3C15-I3-V1	Desconhecido	-	-	-	3604
p-C101-P3C15-I3-V2	Viável	11008,0	-8957,7	181,4%	3605
p-C101-P3C15-I3-V3	Viável	10861,0	-9853,8	190,7%	3601
p-C101-P3C15-I4-V1	Viável	9707,0	-8910,4	191,8%	3601
p-C101-P3C15-I4-V2	Viável	9786,0	-7913,8	180,9%	3602
p-C101-P3C15-I4-V3	Viável	10718,0	-8872,5	182,8%	3607

Tabela C.2. Resultados do modelo que não permite violações nas janelas de tempo para instâncias p-R101

Instância	Status	Solução Primal	Solução Dual	GAP	Tempo (segundos)
p-R101-P1C5-I2-V1	Inviável ou Ilimitado	-	-	-	70
p-R101-P1C5-I2-V2	Ótimo	3294,0	3294,0	0,0%	61
p-R101-P1C5-I2-V3	Ótimo	4253,0	4253,0	0,0%	285
p-R101-P1C5-I3-V1	Ótimo	3441,0	3441,0	0,0%	28
p-R101-P1C5-I3-V2	Viável	3817,0	3729,0	2,3%	3600
p-R101-P1C5-I3-V3	Viável	3690,0	3640,0	1,4%	3601
p-R101-P1C5-I4-V1	Viável	3951,0	3920,0	0,8%	3601
p-R101-P1C5-I4-V2	Viável	3980,0	3900,0	2,0%	3601
p-R101-P1C5-I4-V3	Ótimo	2784,0	2784,0	0,0%	59
p-R101-P1C8-I2-V1	Desconhecido	-	-	-	3600
p-R101-P1C8-I2-V2	Desconhecido	-	-	-	3600
p-R101-P1C8-I2-V3	Ótimo	6399,0	6399,0	0,0%	191
p-R101-P1C8-I3-V1	Viável	5541,0	5344,0	3,6%	3617
p-R101-P1C8-I3-V2	Desconhecido	-	-	-	3600
p-R101-P1C8-I3-V3	Viável	5479,0	5327,0	2,8%	3602
p-R101-P1C8-I4-V1	Viável	5463,0	5390,0	1,3%	3600
p-R101-P1C8-I4-V2	Viável	5657,0	5310,0	6,1%	3600
p-R101-P1C8-I4-V3	Viável	5568,0	5432,0	2,4%	3600
p-R101-P2C10-I2-V1	Viável	7838,0	7728,0	1,4%	3600
p-R101-P2C10-I2-V2	Viável	6932,0	6749,1	2,6%	3600
p-R101-P2C10-I2-V3	Viável	6892,0	6640,0	3,7%	3606
p-R101-P2C10-I3-V1	Viável	7742,0	7510,0	3,0%	3706
p-R101-P2C10-I3-V2	Viável	6685,0	6590,0	1,4%	3600
p-R101-P2C10-I3-V3	Viável	6997,0	6833,0	2,3%	3600
p-R101-P2C10-I4-V1	Viável	7052,0	6909,0	2,0%	3600
p-R101-P2C10-I4-V2	Viável	6940,0	6129,0	11,7%	3601
p-R101-P2C10-I4-V3	Viável	6493,0	6272,0	3,4%	3600
p-R101-P2C15-I2-V1	Viável	11575,0	11330,0	2,1%	3608
p-R101-P2C15-I2-V2	Desconhecido	-	-	-	3600
p-R101-P2C15-I2-V3	Viável	11870,0	11560,0	2,6%	3601
p-R101-P2C15-I3-V1	Desconhecido	-	-	-	3602
p-R101-P2C15-I3-V2	Desconhecido	-	-	-	3601
p-R101-P2C15-I3-V3	Viável	11297,0	11010,0	2,5%	3711
p-R101-P2C15-I4-V1	Viável	9872,0	9042,0	8,4%	3601
p-R101-P2C15-I4-V2	Desconhecido	-	-	-	3602
p-R101-P2C15-I4-V3	Desconhecido	-	-	-	3602
p-R101-P3C15-I2-V1	Viável	11381,0	-8962,8	178,8%	3602
p-R101-P3C15-I2-V2	Viável	12019,0	11755,0	2,2%	3602
p-R101-P3C15-I2-V3	Viável	10703,0	10364,0	3,1670%	3601
p-R101-P3C15-I3-V1	Viável	11532,0	-9869,4	185,6%	3602
p-R101-P3C15-I3-V2	Viável	10612,0	10230,0	3,6%	3602
p-R101-P3C15-I3-V3	Desconhecido	-	-	-	3601
p-R101-P3C15-I4-V1	Viável	10372,0	9940,0	4,2%	3601
p-R101-P3C15-I4-V2	Viável	9935,0	9450,0	4,9%	3601
p-R101-P3C15-I4-V3	Viável	10320,0	9900,0	4,1%	3602

que quanto maior a janela de tempo nas construções para instâncias com uma planta e cinco construções mais difícil fica para o método obter a solução ótima.

C.2 Testes do modelo que permite violação das janelas de tempo das construções

A tabela C.3 apresenta os resultados para as instâncias p-C101 e a tabela C.4 para as instâncias p-R101. A primeira coluna possui o nome das instâncias, a segunda o status da solução encontrada, a terceira a solução primal encontrada, a quarta a solução dual, a quinta o número de demandas não atendidas dentro da janela de tempo inicial da construção, a sexta o valor do GAP encontrado, e na última o tempo em segundos da execução. Quando não se obteve um valor de solução o carácter “ - ” é sinalizado nas colunas três, quatro, cinco e seis.

Foram obtidas 2 soluções ótimas, 76 soluções viáveis, 11 com o status de desconhecido, e 1 foi considerada inviável. Ocorreu uma redução na obtenção de soluções ótimas em relação ao modelo que não permite violação das janelas de tempo pelo fato do modelo conter mais restrições e variáveis que o torna mais difícil de ser resolvido.

Tabela C.3. Resultados do modelo que permite violações nas janelas de tempo para instâncias p-C101

Instância	Status	Solução Primal	Solução Dual	Demandas não atendidas	GAP	Tempo (segundos)
p-C101-P1C5-I2-V1	Viável	4100,0	4060,0	0	1,0%	3601
p-C101-P1C5-I2-V2	Viável	3383,0	3300,0	0	2,5%	3602
p-C101-P1C5-I2-V3	Ótimo	4003,0	4003,0	0	0,0%	74
p-C101-P1C5-I3-V1	Viável	2802,0	2725,0	0	2,7%	3601
p-C101-P1C5-I3-V2	Viável	3061,0	2960,0	0	3,3%	3601
p-C101-P1C5-I3-V3	Ótimo	3860,0	3860,0	0	0,0%	477
p-C101-P1C5-I4-V1	Viável	3521,0	3450,0	0	2,0%	3601
p-C101-P1C5-I4-V2	Viável	3430,0	3290,0	0	4,1%	3602
p-C101-P1C5-I4-V3	Viável	3055,0	2970,0	0	2,8%	3601
p-C101-P1C8-I2-V1	Viável	500005913,0	5298,2	5	100,0%	3600
p-C101-P1C8-I2-V2	Viável	200005956,0	5280,1	2	100,0%	3603
p-C101-P1C8-I2-V3	Viável	5816,0	5660,0	0	2,7%	3600
p-C101-P1C8-I3-V1	Viável	5336,0	5160,0	0	3,3%	3600
p-C101-P1C8-I3-V2	Desconhecido	-	-	-	-	3601
p-C101-P1C8-I3-V3	Viável	5427,0	5333,0	0	1,7%	3600
p-C101-P1C8-I4-V1	Viável	4979,0	4711,0	0	5,4%	3602
p-C101-P1C8-I4-V2	Viável	5528,0	5330,0	0	3,6%	3692
p-C101-P1C8-I4-V3	Viável	400005433,0	4548,0	4	100,0%	3600
p-C101-P2C10-I2-V1	Viável	7470,0	7360,0	0	1,5%	3604
p-C101-P2C10-I2-V2	Viável	7572,0	7300,0	0	3,6%	3604
p-C101-P2C10-I2-V3	Inviável	-	-	-	-	3
p-C101-P2C10-I3-V1	Viável	100006736,0	4970,3	1	100,0%	3601
p-C101-P2C10-I3-V2	Viável	6516,0	6340,0	0	2,7%	3603
p-C101-P2C10-I3-V3	Viável	200007278,0	4821,4	2	100,0%	3601
p-C101-P2C10-I4-V1	Viável	5976,0	5771,0	0	3,4%	3613
p-C101-P2C10-I4-V2	Viável	6260,0	6100,0	0	2,6%	3604
p-C101-P2C10-I4-V3	Viável	7192,0	7000,0	0	2,7%	3708
p-C101-P2C15-I2-V1	Viável	900009776,0	7328,4	9	100,0%	3601
p-C101-P2C15-I2-V2	Desconhecido	-	-	-	-	3601
p-C101-P2C15-I2-V3	Viável	1200010117,0	7310,8	12	100,0%	3603
p-C101-P2C15-I3-V1	Viável	900009593,0	-8698,0	9	100,0%	3601
p-C101-P2C15-I3-V2	Viável	1600009823,0	-8646,7	16	100,0%	3601
p-C101-P2C15-I3-V3	Viável	200011858,0	-9962,7	2	100,0%	3602
p-C101-P2C15-I4-V1	Viável	3700009792,0	-9646,3	37	100,0%	3602
p-C101-P2C15-I4-V2	Viável	4300009941,0	-8068,3	43	100,0%	3606
p-C101-P2C15-I4-V3	Desconhecido	-	-	-	-	3603
p-C101-P3C15-I2-V1	Viável	700009806,0	-10137,8	7	100,0%	3604
p-C101-P3C15-I2-V2	Viável	4800010049,0	-9620,1	48	100,0%	3601
p-C101-P3C15-I2-V3	Viável	2200009856,0	-10121,5	22	100,0%	3602
p-C101-P3C15-I3-V1	Desconhecido	-	-	-	-	3602
p-C101-P3C15-I3-V2	Desconhecido	-	-	-	-	3601
p-C101-P3C15-I3-V3	Viável	400010792,0	7124,0	4	100,0%	3601
p-C101-P3C15-I4-V1	Viável	1000009067,0	-9161,2	10	100,0%	3602
p-C101-P3C15-I4-V2	Viável	1100009496,0	-9484,6	11	100,0%	3602
p-C101-P3C15-I4-V3	Desconhecido	-	-	-	-	3602

Tabela C.4. Resultados do modelo que permite violações nas janelas de tempo para instâncias p-R101

Instância	Status	Solução Primal	Solução Dual	Demandas não atendidas	GAP	Tempo (segundos)
p-R101-P1C5-I2-V1	Viável	300003781,0	3522,9	3	100,0%	3600
p-R101-P1C5-I2-V2	Viável	3294,0	3217,0	0	2,3%	3601
p-R101-P1C5-I2-V3	Viável	4253,0	4180,0	0	1,7%	3615
p-R101-P1C5-I3-V1	Viável	3441,0	3370,0	0	2,1%	3601
p-R101-P1C5-I3-V2	Viável	3817,0	3650,0	0	4,4%	3601
p-R101-P1C5-I3-V3	Viável	3690,0	3600,0	0	2,4%	3601
p-R101-P1C5-I4-V1	Viável	3951,0	3910,0	0	1,0%	3601
p-R101-P1C5-I4-V2	Viável	3980,0	3900,0	0	2,0%	3601
p-R101-P1C5-I4-V3	Viável	2784,0	2691,0	0	3,3%	3601
p-R101-P1C8-I2-V1	Viável	800006141,0	5254,4	8	100,0%	3600
p-R101-P1C8-I2-V2	Viável	900006579,0	5418,7	9	100,0%	3601
p-R101-P1C8-I2-V3	Viável	6399,0	6235,0	0	2,6%	3600
p-R101-P1C8-I3-V1	Viável	5541,0	5359,0	0	3,3%	3676
p-R101-P1C8-I3-V2	Viável	400005520,0	4882,9	4	100,0%	3600
p-R101-P1C8-I3-V3	Viável	5479,0	5307,0	0	3,1%	3600
p-R101-P1C8-I4-V1	Viável	5463,0	5360,0	0	1,9%	3600
p-R101-P1C8-I4-V2	Viável	5663,0	5280,0	0	6,8%	3608
p-R101-P1C8-I4-V3	Viável	5568,0	5400,0	0	3,0%	3600
p-R101-P2C10-I2-V1	Viável	200007477,0	6174,6	2	100,0%	3602
p-R101-P2C10-I2-V2	Viável	6932,0	6762,0	0	2,5%	3603
p-R101-P2C10-I2-V3	Viável	500006875,0	5550,2	5	100,0%	3601
p-R101-P2C10-I3-V1	Viável	200007581,0	5119,9	2	100,0%	3601
p-R101-P2C10-I3-V2	Viável	6684,0	6551,0	0	2,0%	3600
p-R101-P2C10-I3-V3	Viável	7005,0	6790,0	0	3,1%	3604
p-R101-P2C10-I4-V1	Viável	7049,0	6889,0	0	2,3%	3600
p-R101-P2C10-I4-V2	Viável	6957,0	5298,6	0	23,8%	3602
p-R101-P2C10-I4-V3	Viável	100006431,0	5897,4	1	100,0%	3600
p-R101-P2C15-I2-V1	Viável	1000011206,0	7556,1	10	100,0%	3601
p-R101-P2C15-I2-V2	Viável	700009820,0	7762,5	7	100,0%	3608
p-R101-P2C15-I2-V3	Viável	4500010929,0	-8733,0	45	100,0%	3601
p-R101-P2C15-I3-V1	Desconhecido	-	-	-	-	3602
p-R101-P2C15-I3-V2	Viável	2100010328,0	7561,6	21	100,0%	3601
p-R101-P2C15-I3-V3	Viável	700010787,0	7870,7	7	100,0%	3602
p-R101-P2C15-I4-V1	Viável	400009598,0	7308,2	4	100,0%	3601
p-R101-P2C15-I4-V2	Desconhecido	-	-	-	-	3602
p-R101-P2C15-I4-V3	Desconhecido	-	-	-	-	3601
p-R101-P3C15-I2-V1	Viável	4800010140,0	-10198,4	48	100,0%	3602
p-R101-P3C15-I2-V2	Viável	800011322,0	7563,1	8	100,0%	3601
p-R101-P3C15-I2-V3	Viável	1100010585,0	7406,8	11	100,0%	3601
p-R101-P3C15-I3-V1	Desconhecido	-	-	-	-	3603
p-R101-P3C15-I3-V2	Viável	400010466,0	7323,2	4	100,0%	3602
p-R101-P3C15-I3-V3	Desconhecido	-	-	-	-	3605
p-R101-P3C15-I4-V1	Viável	1700009785,0	7387,9	17	100,0%	3601
p-R101-P3C15-I4-V2	Viável	300009910,0	7775,3	3	100,0%	3608
p-R101-P3C15-I4-V3	Viável	700009968,0	7647,9	7	100,0%	3602

C.3 Testes do modelo que não permite violação das janelas de tempo das construções iniciado com o valor da solução da heurística

A tabela C.5 apresenta os resultados para as instâncias p-C101 e a tabela C.6 para as instâncias p-R101. A primeira coluna possui o nome das instâncias, a segunda o valor final da solução obtida pela heurística, a terceira o tempo de execução da heurística em segundos, a quarta o status da solução encontrada pelo modelo, a quinta a solução primal encontrada, a sexta a solução dual, a sétima o valor do GAP encontrado, e na última o tempo em segundos da execução do modelo. Quando não se obteve um valor de solução o carácter “ - ” é sinalizado nas colunas dois, cinco, seis e sete.

Foram obtidas 11 soluções ótimas, 70 soluções viáveis, 6 com o status de desconhecido, 2 como inviáveis ou ilimitadas, e 1 como inviável. Ocorreu um aumento na obtenção de soluções viáveis em relação ao modelo que não permite violação das janelas de tempo pelo fato do modelo já iniciar em muitos casos com soluções viáveis geradas pela heurística.

Tabela C.5. Resultados do modelo que não permite violações nas janelas de tempo da construção iniciando com a solução da heurística para instâncias p-C101

Instância	Solução Heurística	Tempo da Heurística (Segundos)	Status	Solução Primal	Solução Dual	GAP	Tempo (segundos)
p-C101-P1C5-I2-V1	4100	0	Ótimo	4100	4100	0%	12
p-C101-P1C5-I2-V2	3383	0	Ótimo	3383	3383	0%	42
p-C101-P1C5-I2-V3	4003	0	Ótimo	4003	4003	0%	51
p-C101-P1C5-I3-V1	2802	0	Ótimo	2802	2802	0%	58
p-C101-P1C5-I3-V2	3061	0	Ótimo	3061	3061	0%	26
p-C101-P1C5-I3-V3	3860	0	Ótimo	3860	3860	0%	1406
p-C101-P1C5-I4-V1	3521	0	Viável	3521	3460	2%	3601
p-C101-P1C5-I4-V2	3450	1	Viável	3430	3310	3%	3600
p-C101-P1C5-I4-V3	3059	0	Viável	3055	3020	1%	3601
p-C101-P1C8-I2-V1	-	5	Inviável ou Ilimitado	-	-	-	451
p-C101-P1C8-I2-V2	-	6	Desconhecido	-	-	-	3600
p-C101-P1C8-I2-V3	5817	1	Viável	5816	5774	1%	3600
p-C101-P1C8-I3-V1	5344	2	Viável	5334	5224	2%	3600
p-C101-P1C8-I3-V2	-	17	Desconhecido	-	-	-	3601
p-C101-P1C8-I3-V3	5427	1	Viável	5427	5397	1%	3601
p-C101-P1C8-I4-V1	4991	4	Viável	4979	4951	1%	3600
p-C101-P1C8-I4-V2	5554	1	Viável	5528	5381	3%	3600
p-C101-P1C8-I4-V3	5400	10	Viável	5395	4860	10%	3601
p-C101-P2C10-I2-V1	7470	1	Viável	7470	7380	1%	3606
p-C101-P2C10-I2-V2	7561	3	Viável	7558	7300	3%	3602
p-C101-P2C10-I2-V3	-	1	Inviável	-	-	-	3
p-C101-P2C10-I3-V1	6551	2	Viável	6551	6310	4%	3603
p-C101-P2C10-I3-V2	6516	2	Viável	6514	6340	3%	3602
p-C101-P2C10-I3-V3	7104	3	Viável	7104	6820	4%	3607
p-C101-P2C10-I4-V1	5966	2	Viável	5966	5772	3%	3603
p-C101-P2C10-I4-V2	6267	2	Viável	6262	6100	3%	3601
p-C101-P2C10-I4-V3	7209	2	Viável	7205	7010	3%	3602
p-C101-P2C15-I2-V1	10287	4	Viável	10287	-3886	138%	3601
p-C101-P2C15-I2-V2	10311	5	Viável	10311	9960	3%	3601
p-C101-P2C15-I2-V3	11129	5	Viável	11123	-9029	181%	3600
p-C101-P2C15-I3-V1	9871	6	Viável	9871	-8390	185%	3600
p-C101-P2C15-I3-V2	9812	9	Viável	9812	-5678	158%	3601
p-C101-P2C15-I3-V3	11607	5	Viável	11607	-9570	182%	3601
p-C101-P2C15-I4-V1	10938	4	Viável	10938	10710	2%	3604
p-C101-P2C15-I4-V2	9933	4	Viável	9933	9540	4%	3607
p-C101-P2C15-I4-V3	9101	46	Viável	9100	-6893	176%	3602
p-C101-P3C15-I2-V1	11031	5	Viável	11031	-9853	189%	3601
p-C101-P3C15-I2-V2	10264	8	Viável	10264	-9398	192%	3600
p-C101-P3C15-I2-V3	11071	5	Viável	11071	10860	2%	3602
p-C101-P3C15-I3-V1	10287	6	Viável	10287	-9867	196%	3600
p-C101-P3C15-I3-V2	10803	9	Viável	10803	-10238	195%	3601
p-C101-P3C15-I3-V3	10612	7	Viável	10612	-9902	193%	3600
p-C101-P3C15-I4-V1	9182	7	Viável	9182	8940	3%	3601
p-C101-P3C15-I4-V2	9630	6	Viável	9630	-8118	184%	3601
p-C101-P3C15-I4-V3	10172	8	Viável	10172	-10243	201%	3603

C.3. TESTES DO MODELO QUE NÃO PERMITE VIOLAÇÃO DAS JANELAS DE TEMPO DAS CONSTRUÇÕES INICIADO COM O VALOR DA SOLUÇÃO DA HEURÍSTICA 85

Tabela C.6. Resultados do modelo que não permite violações nas janelas de tempo da construção iniciando com a solução da heurística para instâncias p-R101

Instância	Solução Heurística	Tempo da Heurística (Segundos)	Status	Solução Primal	Solução Dual	GAP	Tempo (segundos)
p-R101-P1C5-I2-V1	-	3	Inviável ou Ilimitado	-	-	-	71
p-R101-P1C5-I2-V2	3294	1	Ótimo	3294	3294	0%	55
p-R101-P1C5-I2-V3	4253	0	Ótimo	4253	4253	0%	242
p-R101-P1C5-I3-V1	3449	0	Ótimo	3441	3441	0%	25
p-R101-P1C5-I3-V2	3824	1	Viável	3817	3775	1%	3600
p-R101-P1C5-I3-V3	3690	0	Viável	3690	3640	1%	3601
p-R101-P1C5-I4-V1	3951	1	Viável	3951	3920	1%	3601
p-R101-P1C5-I4-V2	3980	0	Viável	3980	3900	2%	3601
p-R101-P1C5-I4-V3	2784	1	Ótimo	2784	2784	0%	38
p-R101-P1C8-I2-V1	-	7	Desconhecido	-	-	-	3600
p-R101-P1C8-I2-V2	-	8	Desconhecido	-	-	-	3600
p-R101-P1C8-I2-V3	6399	1	Ótimo	6399	6399	0%	265
p-R101-P1C8-I3-V1	5544	4	Viável	5541	5319	4%	3600
p-R101-P1C8-I3-V2	-	12	Desconhecido	-	-	-	3600
p-R101-P1C8-I3-V3	5479	2	Viável	5479	5317	3%	3600
p-R101-P1C8-I4-V1	5463	1	Viável	5463	5380	2%	3601
p-R101-P1C8-I4-V2	5678	8	Viável	5657	5280	7%	3600
p-R101-P1C8-I4-V3	5572	1	Viável	5568	5432	2%	3600
p-R101-P2C10-I2-V1	7842	5	Viável	7842	7670	2%	3600
p-R101-P2C10-I2-V2	6932	3	Viável	6932	6747	3%	3601
p-R101-P2C10-I2-V3	6884	2	Viável	6884	6640	4%	3603
p-R101-P2C10-I3-V1	7746	3	Viável	7746	7510	3%	3604
p-R101-P2C10-I3-V2	6712	3	Viável	6685	6587	1%	3601
p-R101-P2C10-I3-V3	7012	2	Viável	7012	6820	3%	3601
p-R101-P2C10-I4-V1	7052	2	Viável	7049	6919	2%	3600
p-R101-P2C10-I4-V2	6579	6	Viável	6575	6129	7%	3604
p-R101-P2C10-I4-V3	6429	3	Viável	6425	6272	2%	3601
p-R101-P2C15-I2-V1	11567	4	Viável	11567	11330	2%	3606
p-R101-P2C15-I2-V2	10156	10	Viável	10156	9899	3%	3602
p-R101-P2C15-I2-V3	11845	10	Viável	11845	11560	2%	3607
p-R101-P2C15-I3-V1	10655	27	Viável	10655	10155	5%	3600
p-R101-P2C15-I3-V2	-	22	Desconhecido	-	-	-	3605
p-R101-P2C15-I3-V3	11284	7	Viável	11284	11010	2%	3601
p-R101-P2C15-I4-V1	9440	11	Viável	9440	9042	4%	3606
p-R101-P2C15-I4-V2	9914	31	Viável	9914	-4347	144%	3603
p-R101-P2C15-I4-V3	10550	43	Viável	10550	9817	7%	3600
p-R101-P3C15-I2-V1	11345	8	Viável	11345	-9990	188%	3602
p-R101-P3C15-I2-V2	12019	5	Viável	12019	11755	2%	3600
p-R101-P3C15-I2-V3	10647	6	Viável	10647	-6559	162%	3606
p-R101-P3C15-I3-V1	11156	23	Viável	11156	-9933	189%	3600
p-R101-P3C15-I3-V2	10493	6	Viável	10493	10230	3%	3600
p-R101-P3C15-I3-V3	10430	41	Viável	10430	-6517	162%	3607
p-R101-P3C15-I4-V1	10284	16	Viável	10284	-6507	163%	3601
p-R101-P3C15-I4-V2	9716	8	Viável	9716	9450	3%	3607
p-R101-P3C15-I4-V3	10204	9	Viável	10204	9900	3%	3601