



Rodrigo Mosconi de Gouvêa

Serviços, Processos e Máquinas: Um estudo de metodologias para realocação de processos nas máquinas

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática da PUC-Rio.

Orientador: Prof. Marcus Vinicius Soledade Poggi de Aragão

Rio de Janeiro
abril de 2018



Rodrigo Mosconi de Gouvêa

Serviços, Processos e Máquinas: Um estudo de metodologias para realocação de processos nas máquinas

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marcus Vinicius Soledade Poggi de Aragão

Orientador

Departamento de Informática – PUC-Rio

Prof. Rafael Martinelli Pinto

Departamento de Engenharia Industrial – PUC-Rio

Prof. Anderson Oliveira da Silva

Departamento de Informática – PUC-Rio

Prof. Márcio da Silveira Carvalho

Coordenador Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 26 de abril de 2018

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Rodrigo Mosconi de Gouvêa

Graduou-se em Matemática na Universidade do Estado do Rio de Janeiro como bacharel. Tem pós-graduação *lato sensu* de Especialização em Redes de Computadores pelo CCE PUC-Rio. Trabalhou em *data center* comercial e atualmente trabalha na manutenção dos sistemas do núcleo computacional do Tribunal Regional do Trabalho da Primeira Região.

Ficha Catalográfica

Mosconi de Gouvêa, Rodrigo

Serviços, Processos e Máquinas: Um estudo de metodologias para realocação de processos nas máquinas / Rodrigo Mosconi de Gouvêa; orientador: Marcus Vinicius Soledade Poggi de Aragão. – Rio de Janeiro: PUC-Rio, Departamento de Informática, 2018.

v., 145 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Inclui bibliografia

1. Informática – Teses. 2. Otimização combinatória – Teses. 3. Atribuição de máquinas;. 4. Programação inteira;. 5. Geração de colunas;. I. Soledade Poggi de Aragão, Marcus Vinicius. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

Agradecimentos

Agradeço ao apoio de minha esposa e de meus pais. Agradeço também à CAPES pela bolsa fornecida.

Resumo

Mosconi de Gouvêa, Rodrigo; Soledade Poggi de Aragão, Marcus Vinicius. **Serviços, Processos e Máquinas: Um estudo de metodologias para realocação de processos nas máquinas.** Rio de Janeiro, 2018. 145p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A organização lógica de *data centers* recai principalmente na questão estratégica de distribuir os serviços nos equipamentos de forma que os custos operacionais sejam os menores possíveis. Além desses custos, devem ser considerados outros aspectos que envolvem a interdependência de seus serviços internos e a distribuição entre suas localidades, visando assim melhorar a qualidade de seu produto aos seus clientes. Este trabalho explora o problema de atribuição de processos a máquinas do desafio ROADEF de 2012 pelos métodos de programação inteira e geração de colunas. Apresenta estratégias para lidar com as dificuldades numéricas encontradas. Na geração de colunas, analisa técnicas para acelerar a convergência, por meio de resolver o mestre restrito após cada variável, geração prévia de colunas e estabilização das variáveis duais. Ao final do trabalho, são comparados os resultados obtidos com os melhores resultados oficiais.

Palavras-chave

Atribuição de máquinas; Programação inteira; Geração de colunas;

Abstract

Mosconi de Gouvêa, Rodrigo; Soledade Poggi de Aragão, Marcus Vinicius (Advisor). **Services, Processes and Machines: A methodologies study for machine reassignment problem.** Rio de Janeiro, 2018. 145p. Dissertação de mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A data center logic organization lies mainly by the strategic decision on how distribute services between machines, so the operational costs should be the smallest as possible. Beside those costs, must also consider the interdependence of their own services, the distribution between their localities, to improve the quality of their product to their customers. This work explores the challenge ROADEF 2012 machine assignment problem by the means of integer programming and column generation. Shows strategies to address numeric issues. At column generation, it analyzes techniques to speed up the convergence, by solving after each variable addition, a previous generation of columns and stabilization of duals variables. At the end of the work, it compares the results obtained are compared with the best official results.

Keywords

Machine assignment; Integer programming; Column generation;

Sumário

Lista de figuras	9
Lista de tabelas	13
1 Introdução	15
1.1 Objetivos	16
1.2 Organização do trabalho	17
1.3 Recursos computacionais	17
2 O desafio do ROADEF de 2012	18
2.1 Variáveis de Decisão	18
2.2 Restrições Rígidas	19
2.3 Objetivos	21
2.4 Avaliação	23
2.5 Instâncias	23
2.6 Instância de exemplo	25
2.7 Trabalhos anteriores	27
3 Formulação de Programação Inteira	30
3.1 Variáveis de decisão para programação Inteira	31
3.2 Constantes usados no modelo	31
3.3 Variáveis auxiliares	32
3.4 Restrições para Programação Inteira	37
3.5 Objetivo	39
3.6 Ensaios preliminares para avaliação	41
3.7 Formulação	43
3.8 Comentários	49
4 Geração de Colunas	51
4.1 Variáveis de Decisão do Problema Principal	53
4.2 Variáveis Auxiliares do Problema Principal	54
4.3 Restrições do Problema Principal	55
4.4 Objetivo do Problema Principal	58
4.5 Custo reduzido das Variáveis	59
4.6 Formulação do Problema Principal	60
4.7 Variáveis de Decisão do Problema de Precificação	61
4.8 Variáveis Auxiliares do Problema de Precificação	61
4.9 Restrições do Problema de Precificação	64
4.10 Objetivo do Problema de Precificação	66
4.11 Formulação do Problema de Precificação	69
4.12 Critério de Parada	70
4.13 Programação inteira	71
4.14 Ensaios preliminares de geração de colunas	71
4.15 Otimização após adição de coluna	95

4.16	Geração prévia de colunas	105
4.17	Estabilização das Variáveis duais	115
4.18	Ensaio combinados	124
4.19	Modelo CG3	133
4.20	Comentários	134
5	Análise de Resultados	137
5.1	Programação inteira	137
5.2	Geração de colunas	138
5.3	Comparação com resultados oficiais	140
6	Conclusão e Trabalhos Futuros	141
6.1	Sugestão para trabalhos futuros	142
A	Referências bibliográficas	143
	Referências bibliográficas	143

Lista de figuras

Figura 4.1	Objetivo e densidade de CG1, a1_1	79
Figura 4.2	Tempos e iterações de CG1, a1_1	79
Figura 4.3	Custo reduzido de CG1, a1_1	79
Figura 4.4	Objetivo e densidade de CG1, a1_2	80
Figura 4.5	Tempos e iterações de CG1, a1_2	80
Figura 4.6	Custo reduzido de CG1, a1_2	80
Figura 4.7	Objetivo e densidade de CG1, a1_3	82
Figura 4.8	Tempos e iterações de CG1, a1_3	82
Figura 4.9	Custo reduzido de CG1, a1_3	82
Figura 4.10	Objetivo e densidade de CG1, a1_4	83
Figura 4.11	Tempos e iterações de CG1, a1_4	83
Figura 4.12	Custo reduzido de CG1, a1_4	83
Figura 4.13	Objetivo e densidade de CG1, a1_5	85
Figura 4.14	Tempos e iterações de CG1, a1_5	85
Figura 4.15	Custo reduzido de CG1, a1_5	85
Figura 4.16	Objetivo e densidade de CG2, a1_1	88
Figura 4.17	Tempos e iterações de CG2, a1_1	88
Figura 4.18	Custo reduzido de CG2, a1_1	88
Figura 4.19	Objetivo e densidade de CG2, a1_2	89
Figura 4.20	Tempos e iterações de CG2, a1_2	89
Figura 4.21	Custo reduzido de CG2, a1_2	89
Figura 4.22	Objetivo e densidade de CG2, a1_3	91
Figura 4.23	Tempos e iterações de CG2, a1_3	91
Figura 4.24	Custo reduzido de CG2, a1_3	91
Figura 4.25	Objetivo e densidade de CG2, a1_4	92
Figura 4.26	Tempos e iterações de CG2, a1_4	92
Figura 4.27	Custo reduzido de CG2, a1_4	92
Figura 4.28	Objetivo e densidade de CG2, a1_5	94
Figura 4.29	Tempos e iterações de CG2, a1_5	94
Figura 4.30	Custo reduzido de CG2, a1_5	94
Figura 4.31	Objetivo e densidade de CG1, a1_1, com otimização após cada nova coluna	98
Figura 4.32	Tempos e iterações de CG1, a1_1, com otimização após cada nova coluna	98
Figura 4.33	Custo reduzido de CG1, a1_1, com otimização após cada nova coluna	98
Figura 4.34	Objetivo e densidade de CG1, a1_2, com otimização após cada nova coluna	99
Figura 4.35	Tempos e iterações de CG1, a1_2, com otimização após cada nova coluna	99
Figura 4.36	Custo reduzido de CG1, a1_2, com otimização após cada nova coluna	99
Figura 4.37	Objetivo e densidade de CG1, a1_3, com otimização após cada nova coluna	101

Figura 4.38 Tempos e iterações de CG1, a1_3, com otimização após cada nova coluna	101
Figura 4.39 Custo reduzido de CG1, a1_3, com otimização após cada nova coluna	101
Figura 4.40 Objetivo e densidade de CG1, a1_4, com otimização após cada nova coluna	103
Figura 4.41 Tempos e iterações de CG1, a1_4, com otimização após cada nova coluna	103
Figura 4.42 Custo reduzido de CG1, a1_4, com otimização após cada nova coluna	103
Figura 4.43 Objetivo e densidade de CG1, a1_5, com otimização após cada nova coluna	104
Figura 4.44 Tempos e iterações de CG1, a1_5, com otimização após cada nova coluna	104
Figura 4.45 Custo reduzido de CG1, a1_5, com otimização após cada nova coluna	104
Figura 4.46 Objetivo e densidade de CG1, a1_1, com geração de colunas prévia	108
Figura 4.47 Tempos e iterações de CG1, a1_1, com geração de colunas prévia	108
Figura 4.48 Custo reduzido de CG1, a1_1, com geração de colunas prévia	108
Figura 4.49 Objetivo e densidade de CG1, a1_2, com geração de colunas prévia	109
Figura 4.50 Tempos e iterações de CG1, a1_2, com geração de colunas prévia	109
Figura 4.51 Custo reduzido de CG1, a1_2, com geração de colunas prévia	109
Figura 4.52 Objetivo e densidade de CG1, a1_3, com geração de colunas prévia	111
Figura 4.53 Tempos e iterações de CG1, a1_3, com geração de colunas prévia	111
Figura 4.54 Custo reduzido de CG1, a1_3, com geração de colunas prévia	111
Figura 4.55 Objetivo e densidade de CG1, a1_4, com geração de colunas prévia	113
Figura 4.56 Tempos e iterações de CG1, a1_4, com geração de colunas prévia	113
Figura 4.57 Custo reduzido de CG1, a1_4, com geração de colunas prévia	113
Figura 4.58 Objetivo e densidade de CG1, a1_5, com geração de colunas prévia	114
Figura 4.59 Tempos e iterações de CG1, a1_5, com geração de colunas prévia	114
Figura 4.60 Custo reduzido de CG1, a1_5, com geração de colunas prévia	114
Figura 4.61 Objetivo e densidade de CG1, a1_2, com estabilização constante=0,9	119

Figura 4.62 Tempos e iterações de CG1, a1_2, com estabilização constante=0,9	119
Figura 4.63 Custo reduzido de CG1, a1_2, com estabilização constante=0,9	119
Figura 4.64 Objetivo e densidade de CG1, a1_3, com estabilização constante=0,9	120
Figura 4.65 Tempos e iterações de CG1, a1_3, com estabilização constante=0,9	120
Figura 4.66 Custo reduzido de CG1, a1_3, com estabilização constante=0,9	120
Figura 4.67 Objetivo e densidade de CG1, a1_4, com estabilização constante=0,9	122
Figura 4.68 Tempos e iterações de CG1, a1_4, com estabilização constante=0,9	122
Figura 4.69 Custo reduzido de CG1, a1_4, com estabilização constante=0,9	122
Figura 4.70 Objetivo e densidade de CG1, a1_5, com estabilização constante=0,9	123
Figura 4.71 Tempos e iterações de CG1, a1_5, com estabilização constante=0,9	123
Figura 4.72 Custo reduzido de CG1, a1_5, com estabilização constante=0,9	123
Figura 4.73 Objetivo e densidade de CG1, a1_1, com estabilização constante=0,9 e pré geração de colunas	126
Figura 4.74 Tempos e iterações de CG1, a1_1, com estabilização constante=0,9 e pré geração de colunas	126
Figura 4.75 Custo reduzido de CG1, a1_1, com estabilização constante=0,9 e pré geração de colunas	126
Figura 4.76 Objetivo e densidade de CG1, a1_2, com estabilização constante=0,9 e pré geração de colunas	127
Figura 4.77 Tempos e iterações de CG1, a1_2, com estabilização constante=0,9 e pré geração de colunas	127
Figura 4.78 Custo reduzido de CG1, a1_2, com estabilização constante=0,9 e pré geração de colunas	127
Figura 4.79 Objetivo e densidade de CG1, a1_3, com estabilização constante=0,9 e pré geração de colunas	129
Figura 4.80 Tempos e iterações de CG1, a1_3, com estabilização constante=0,9 e pré geração de colunas	129
Figura 4.81 Custo reduzido de CG1, a1_3, com estabilização constante=0,9 e pré geração de colunas	129
Figura 4.82 Objetivo e densidade de CG1, a1_4, com estabilização constante=0,9 e pré geração de colunas	131
Figura 4.83 Tempos e iterações de CG1, a1_4, com estabilização constante=0,9 e pré geração de colunas	131
Figura 4.84 Custo reduzido de CG1, a1_4, com estabilização constante=0,9 e pré geração de colunas	131
Figura 4.85 Objetivo e densidade de CG1, a1_5, com estabilização constante=0,9 e pré geração de colunas	132

Figura 4.86 Tempos e iterações de CG1, a1_5, com estabilização constante=0,9 e pré geração de colunas	132
Figura 4.87 Custo reduzido de CG1, a1_5, com estabilização constante=0,9 e pré geração de colunas	132

Lista de tabelas

Tabela 2.1	Tamanho das instâncias a	23
Tabela 2.2	Tamanho das instâncias b	24
Tabela 2.3	Tamanho das instâncias x	24
Tabela 2.4	Solução trivial das instâncias	24
Tabela 2.5	Exemplo: Capacidade e Capacidade segura	26
Tabela 2.6	Exemplo: Custo de Movimentação de Máquina - $MMC(m_i, m_j)$	26
Tabela 2.7	Exemplo: Requisitos dos processos e Custo de Movimentação de Processo - $PMC(p_i)$	26
Tabela 3.1	Resultado parcial usando modelos $MIP1$	42
Tabela 3.2	Resultado parcial usando modelos $MIP2$	45
Tabela 3.3	Faixa de valores no modelo $MIP2$	46
Tabela 3.4	Resultado parcial usando modelos $MIP3$	47
Tabela 3.5	Resultado parcial usando modelos $MIP4$	48
Tabela 3.6	Faixa de valores no modelo $MIP4$	48
Tabela 4.1	Resultado parcial por geração de colunas, CG1	77
Tabela 4.2	Resultado parcial por geração de colunas, CG2	86
Tabela 4.3	Resultado parcial por geração de colunas, CG1, com otimização após cada nova coluna	95
Tabela 4.4	Resultado parcial por geração de colunas, CG1, com geração de colunas prévia	105
Tabela 4.5	Tempo de geração dos modelos, variáveis e densidade em CG1	106
Tabela 4.6	Resultado parcial por geração de colunas, CG1, com estabilização constante=0,9	116
Tabela 4.7	Resultados de CG1, a1_1, com vários coeficientes de estabilização constante	117
Tabela 4.8	Resultado parcial por geração de colunas, CG1, com estabilização constante=0,9 e geração prévia de variáveis	124
Tabela 4.9	Resultado parcial por geração de colunas, CG3, com estabilização constante=0,9 e geração prévia de variáveis	133
Tabela 4.10	Resultado de geração de colunas, a1_1	135
Tabela 4.11	Resultado de geração de colunas de a1_2	135
Tabela 4.12	Resultado de geração de colunas a1_3	135
Tabela 4.13	Resultado de geração de colunas a1_4	135
Tabela 4.14	Resultado de geração de colunas a1_5	136
Tabela 5.1	Resultado de a1 por MIP	137
Tabela 5.2	Resultado de a1 por geração de colunas	139
Tabela 5.3	Resultado de a1	140

Lista de Abreviaturas

C	<i>Capacity</i>
CG	<i>Column Generation</i>
EURO	<i>European Operational Research Society</i>
LP	<i>Linear Programing</i>
MIP	<i>Mixed Integer Programing</i>
MMC	<i>Machine Move Cost</i>
PMC	<i>Process Move Cost</i>
ROADEF	<i>Société française de Recherche Opérationnelle et Aide à la Décision</i>
SC	<i>Safe Capacity</i>
SMC	<i>Service Move Cost</i>

1

Introdução

Os serviços de nuvem são uma realidade inegável. Cada vez mais serviços são alocados em alguma infraestrutura de terceiros na Internet, criando uma abstração de todas as camadas inferiores. Esta é a *Cloud*, com seus modelos de *Infraestrutura como serviço*, *Plataforma como serviço* e *Software como serviço*. Porém, para atenderem as demandas, os provedores de cada camada podem necessitar de mais recursos das camadas inferiores. Alguns desses provedores ainda trabalham com os *data centers* e seus serviços providos podem ter como requisitos uma distribuição entre localidades, visando uma resiliência a falhas. São os hardwares componentes custosos e de lenta escalabilidade, podendo conter um tempo de entrega e de ativação elevados. A correta utilização dos recursos atuais pode resultar em prevenção de gastos futuros na aquisição de novos computadores.

A gestão de um *data center* é uma questão estratégica para as empresas que lidam com as necessidades de gerirem as demandas físicas e operacionais. A importância é suficientemente grande, a ponto de existirem entidades¹ que realizam certificações de projeto, construção e manutenção de data center. A certificação de projeto, por exemplo, considera o isolamento dos corredores quente e frio, reduzindo assim a troca de calor, e melhorando a eficiência de ar condicionado.

A questão energética de data center físico é um tópico vital, onde ainda é abordado na literatura técnica moderna² para melhorar o consumo de energia. A redução do consumo de energia, além de reduzir a conta de eletricidade, pode resultar em uma redução de carga nos sistemas de aquecimento, ventilação ou ar condicionado (HVAC), podendo reduzir também a carga nos geradores de emergência, caso ocorra falha na distribuidora local de energia.

Outra característica de gestão estratégica está na continuidade de servi-

¹Uptime Institute <https://uptimeinstitute.com/>

²por exemplo o livro *Energy Efficient Servers: Blueprints for Data Center Optimization*, publicado em 2015. ISBN-13: 978-1-4302-6637-2

ços, onde o serviço provido pode ser realocado para outros lugares em caso de falha da localidade onde estava. Ambientes com alta disponibilidade normalmente possuem seus serviços dispostos em várias localidades, o que resulta, em contra-partida, em lidar com problemas de consistência de dados distribuídos.

Em 2012, houve um desafio com o escopo de melhorar a organização dos processos em um conjunto de máquinas. Os organizadores forneceram um conjunto de instâncias que são cenários hipotéticos de processos e máquinas, com demanda e capacidade de recursos, assim como relações de dependência entre os serviços providos.

1.1

Objetivos

A base deste trabalho é o desafio do ROADEF de 2012 (ROADEF, 2012)³, que consiste em um problema de redistribuição de tarefas de múltiplos recursos, onde deve decidir que processos devem ser alocados em quais máquinas, sendo que cada máquina possui recursos limitados. Há também uma característica de *constraint programming*, onde há critérios que devem ser cumpridos, porém estes não possuem métricas de custo. Este trabalho visa identificar as configurações ótimas das instâncias deste desafio e foram usadas metodologias distintas para resolver este problema.

A *Programação Inteira* foi a abordagem utilizada para formular o problema. O modelo resultante é a base da interpretação do problema para as demais etapas de estudo.

As dimensões do modelo de programação inteiro resultante, assim como a qualidade em termos de limites inferiores das sua relaxação linear, sugeriram a utilização de técnicas de decomposição. A resolução do modelo resultante é feita utilizando a técnica de *Geração de Colunas*. Esta é uma técnica tradicionalmente eficiente para resolver problemas de distribuição de tarefas, roteamento ou corte de chapas. Os primeiros usos deste método para esses tipos de problema datam desde 1958 (Ford & Fulkerson, 1958) e posteriormente em (Dantiz & Wolf, 1960), (Gilmore & Gomory, 1961), (Gilmore & Gomory, 1963) e (Barnhart et al., 1998).

³<http://www.roadef.org/challenge/2012/en/>

1.2

Organização do trabalho

Esta dissertação está organizada da seguinte forma: o Capítulo 2 descreve o desafio, que é a base deste trabalho; o Capítulo 3 apresenta uma formulação de programação inteira para o problema e mostra também algumas dificuldades em sua formulação pura; o Capítulo 4 mostra uma abordagem de geração de colunas.

Durante os capítulos, são mostrados resultados das metodologias aplicadas para as instâncias **a1** do desafio. A compilação de resultados para as instâncias de cada método está presente no Capítulo 5. As conclusões e trabalhos futuros estão presentes no Capítulo 6.

1.3

Recursos computacionais

Os ensaios computacionais são realizados em um computador com processador AMD A10-6800K, 16GB de RAM, 2 discos SSD em RAID0, sistema operacional Ubuntu 16.04.2 LTS de 64 bits, Anaconda 4.3.30 (com Python 3.6.3 e NumPy 1.13.3) e Gurobi 7.5.1. Como este trabalho visa a otimalidade, os ensaios foram realizados com um tempo máximo de execução de 6 horas. Para validação dos resultados, a configuração resultante foi validada pelo programa `checker`⁴ fornecido pela equipe proponente do desafio.

⁴http://www.roadef.org/challenge/2012/files/solution_checker/

O desafio do ROADEF de 2012

Em 2012, a *Société française de Recherche Operationnelle et Aide à la Décision* (ROADEF) em conjunto com a *European Operational Research Society* (EURO) e em parceria com a Google elaboraram um desafio para o problema de alocação de máquina. As seções 2.1, 2.2 e 2.3 são transcrições parciais do documento (ROADEF, 2012).

Do mesmo documento, definiu o desafio como:

O objetivo do desafio é melhorar a utilização de um conjunto de máquinas. Uma máquina possui recursos, como por exemplo *CPU* e *RAM*, e possui processos rodando que consomem esses recursos. Inicialmente cada processo está alocado a uma máquina. Para melhorar a utilização de uma máquina, processos podem ser migrados de uma máquina para outra. Os movimentos possíveis são limitados pelas restrições rígidas, por exemplo restrições de capacidade de recursos e possuem um custo. A solução do problema é uma nova atribuição processo-máquina, que satisfaça todas as restrições rígidas e minimize um determinado objetivo de custo.

2.1

Variáveis de Decisão

Seja \mathcal{M} o conjunto de máquinas e \mathcal{P} o conjunto de processos. Uma solução são as alocações de cada processo $p \in \mathcal{P}$ para uma única máquina $m \in \mathcal{M}$; esta alocação será notada por $M(p) = m$. A alocação original de um processo p será notada por $M_0(p)$. Observe que a alocação original é viável, isto é, todas as restrições rígidas são satisfeitas.

2.2

Restrições Rígidas

2.2.1

Restrições de Capacidade

Seja \mathcal{R} o conjunto de recursos que são comuns em todas as máquinas, $C(m, r)$ a capacidade do recurso $r \in \mathcal{R}$ da máquina $m \in \mathcal{M}$ e $R(p, r)$ o requisito de recurso $r \in \mathcal{R}$ do processo $p \in \mathcal{P}$. Então, dada uma alocação M , a utilização U da máquina m do recurso r é definida por:

$$U(m, r) = \sum_{\substack{p \in \mathcal{P} \text{ tal que} \\ M(p)=m}} R(p, r) \quad (2-1)$$

Um processo pode rodar em uma máquina se e somente se a máquina possui capacidade disponível suficiente em cada recurso. Formalmente, uma alocação viável deve satisfazer as restrições de capacidade:

$$U(m, r) \leq C(m, r), \quad \forall m \in \mathcal{M}, r \in \mathcal{R} \quad (2-2)$$

2.2.2

Restrições de conflito

Processos são particionados em serviços. Seja \mathcal{S} o conjunto de serviços. Um serviço $s \in \mathcal{S}$ é um conjunto de processos que devem rodar em máquinas distintas. Observe que todos os serviços são disjuntos.

$$(p_i, p_j) \in s^2, p_i \neq p_j \implies M(p_i) \neq M(p_j), \quad \forall s \in \mathcal{S} \quad (2-3)$$

2.2.3

Restrições de espalhamento

Seja \mathcal{L} o conjunto de localidades, uma localidade $l \in \mathcal{L}$ é um conjunto de máquinas. Observe que localidades são conjuntos disjuntos. Para cada $s \in \mathcal{S}$, seja $spreadMin(s) \in \mathbb{N}$ o número mínimo de localidades distintas onde pelo

menos um processo do serviço s deve estar presente. A restrição é definida por:

$$\sum_{l \in \mathcal{L}} \min \left(1, \left\| \{p \in s \mid M(p) \in l\} \right\| \right) \geq \text{spreadMin}(s), \quad \forall s \in \mathcal{S} \quad (2-4)$$

2.2.4

Restrições de dependência

Seja \mathcal{N} um conjunto de vizinhanças, uma vizinhança $n \in \mathcal{N}$ é um conjunto de máquinas. Observe que vizinhanças são conjuntos disjuntos. Se um serviço s^a depende do serviço s^b , então cada processo de s^a deve rodar em uma vizinhança de algum processo de s^b :

$$\forall p^a \in s^a, \exists p^b \in s^b \wedge \exists n \in \mathcal{N} \implies M(p^a) \in n \wedge M(p^b) \in n \quad (2-5)$$

Observe que as dependências não são simétricas, isto é, um serviço s^a depende do serviço s^b não é equivalente do serviço s^b depender do serviço s^a .

2.2.5

Restrições de utilização transitória

Quando um processo p é movido de uma máquina m para outra máquina m' , alguns recursos são consumidos duas vezes; por exemplo espaço em disco não está disponível na máquina m durante a cópia de m para m' e m' deve obviamente ter espaço disponível em disco suficiente para a cópia. Seja $\mathcal{TR} \subset \mathcal{R}$ um subconjunto de recursos que precisam de utilização transitória, isto é, requer a capacidade tanto na alocação inicial $M_0(p)$ quanto na atual $M(p)$. Então a restrição de utilização transitória é:

$$\sum_{\substack{p \in \mathcal{P} \text{ tal que} \\ M(p)=m \vee M_0(p)=m}} R(p, r) \leq C(m, r), \quad \forall m \in \mathcal{M}, r \in \mathcal{TR} \quad (2-6)$$

Observe que não há dimensão de tempo no problema, ou seja, assume-se que todas as movimentações são realizadas exatamente ao mesmo tempo. Então para os recursos em \mathcal{TR} esta restrição soma com a restrição de

capacidade.

2.3

Objetivos

O objetivo é melhorar o uso de um conjunto de máquinas. Para isso, um objetivo de custo total é construído pela combinação de um custo de carga, um custo de balanceamento e vários custos de movimentação.

2.3.1

Custo de carga

Seja $SC(m, r)$ a capacidade segura de recurso $r \in \mathcal{R}$ da máquina $m \in \mathcal{M}$. O custo por carga é definido por recurso e corresponde à capacidade usada acima da capacidade segura; formalmente:

$$loadCost(r) = \sum_{m \in \mathcal{M}} \max(0, U(m, r) - SC(m, r)) \quad (2-7)$$

2.3.2

Custo de balanceamento

Como ter disponíveis recursos CPU sem ter recurso RAM disponível é inútil para futuras alocações, um dos objetivos desse problema é equilibrar recursos disponíveis. A ideia é atingir uma determinada meta na proporção disponível de dois recursos diferentes. Seja \mathcal{B} o conjunto de tuplas definidas em $\mathbb{N} \times \mathcal{R}^2$. Para uma determinada tupla $b = \langle r_1, r_2, target \rangle \in \mathcal{B}$, o custo de balanceamento é:

$$balanceCost(b) = \sum_{m \in \mathcal{M}} \max(0, target \cdot A(m, r_1) - A(m, r_2)) \quad (2-8)$$

onde

$$A(m, r) = C(m, r) - U(m, r) \quad (2-9)$$

2.3.3

Custo de movimentação de processo

Alguns processos são mais difíceis de mover, para modelar esta restrição suave, um custo de movimentação de processo é definido. Seja $PMC(p)$ o custo de mover o processo p de sua máquina original $M_0(p)$.

$$processMoveCost = \sum_{\substack{p \in \mathcal{P} \text{ tal que} \\ M_0(p) \neq M(p)}} PMC(p) \quad (2-10)$$

2.3.4

Custo de movimentação de serviço

Para balancear movimentações entre serviços, um custo de movimentação de serviço foi definido como o número máximo de processo movidos por serviço. Formalmente:

$$serviceMoveCost = \max \left(\left\| \left\{ p \in s \mid M_0(p) \neq M(p) \right\} \right\| \right) \quad (2-11)$$

2.3.5

Custo de movimentação de máquina

Seja $MMC(m_{source}, m_{destination})$ o custo de mover qualquer processo p da máquina m_{source} para a máquina $m_{destination}$. Obviamente para qualquer máquina $m \in \mathcal{M}$, $MMC(m, m) = 0$. O custo de movimentação de máquina é então a soma de todas as movimentações ponderada pelo MMC relevante:

$$machineMoveCost = \sum_{p \in \mathcal{P}} MMC(M_0(p), M(p)) \quad (2-12)$$

2.3.6

Custo objetivo total

O custo total é uma soma ponderada de todos os custos anteriores. É esta soma que deve ser minimizada.

$$\begin{aligned}
totalCost = & \sum_{r \in \mathcal{R}} weight_{loadCost}(r) \cdot loadCost(r) \\
& + \sum_{b \in \mathcal{B}} weight_{balanceCost}(b) \cdot balanceCost(b) \\
& + weight_{processMoveCost} \cdot processMoveCost \\
& + weight_{serviceMoveCost} \cdot serviceMoveCost \\
& + weight_{machineMoveCost} \cdot machineMoveCost
\end{aligned} \tag{2-13}$$

2.4

Avaliação

Os organizadores definiram os critérios de avaliação das respostas das implementações das equipes participantes. Também estipularam um limite de tempo para execução de 5 minutos.

2.5

Instâncias

Os organizadores forneceram um conjunto de instâncias para testes das implementações, assim como avaliação das propostas. As tabelas 2.1, 2.2 e 2.3 contêm os tamanhos das instâncias disponibilizadas no desafio, cujas soluções triviais estão expostas na tabela 2.4. A solução trivial corresponde à solução em que nenhum processo é movido para outra máquina.

Tabela 2.1: Tamanho das instâncias a

Instância	Número de					
	recursos	máquinas	processos	serviços	vizinhança	localidades
a1_1	2	4	100	79	1	4
a1_2	4	100	1000	980	2	4
a1_3	3	100	1000	216	5	25
a1_4	3	50	1000	142	50	50
a1_5	4	12	1000	981	2	4
a2_1	3	100	1000	1000	1	1
a2_2	12	100	1000	170	5	25
a2_3	12	100	1000	129	5	25
a2_4	12	50	1000	180	5	25
a2_5	12	50	1000	153	5	25

Tabela 2.2: Tamanho das instâncias b

Instância	Número de					
	recursos	máquinas	processos	serviços	vizinhança	localidades
b_1	12	100	5000	2512	5	10
b_2	12	100	5000	2462	5	10
b_3	6	100	20000	15025	5	10
b_4	6	500	20000	1732	5	50
b_5	6	100	40000	35082	5	10
b_6	6	200	40000	14680	5	50
b_7	6	4000	40000	15050	5	50
b_8	3	100	50000	45030	5	10
b_9	3	1000	50000	4609	5	100
b_10	3	5000	50000	4896	5	100

Tabela 2.3: Tamanho das instâncias x

Instância	Número de					
	recursos	máquinas	processos	serviços	vizinhança	localidades
x_1	12	100	5000	2529	5	10
x_2	12	100	5000	2484	5	10
x_3	6	100	20000	14928	5	10
x_4	6	500	20000	1190	5	50
x_5	6	100	40000	34872	5	10
x_6	6	200	40000	14504	5	50
x_7	6	4000	40000	15273	5	50
x_8	3	100	50000	44950	5	10
x_9	3	1000	50000	4871	5	100
x_10	3	5000	50000	4615	5	100

Tabela 2.4: Solução trivial das instâncias

Instância	Solução trivial	Instância	Solução trivial	Instância	Solução trivial
a1_1	49.528.750	b_1	7.644.173.180	x_1	7.422.426.760
a1_2	1.061.649.570	b_2	5.181.493.830	x_2	5.103.634.830
a1_3	583.662.270	b_3	6.336.834.660	x_3	6.119.933.380
a1_4	632.499.600	b_4	9.209.576.380	x_4	9.207.188.610
a1_5	782.189.690	b_5	12.426.813.010	x_5	12.369.526.590
a2_1	391.189.190	b_6	12.749.861.240	x_6	12.753.566.360
a2_2	1.876.768.120	b_7	37.946.901.700	x_7	37.763.791.230
a2_3	2.272.487.840	b_8	14.068.207.250	x_8	11.611.565.600
a2_4	3.223.516.130	b_9	23.234.641.520	x_9	23.146.106.380
a2_5	787.355.300	b_10	42.220.868.760	x_10	42.201.640.770

2.6

Instância de exemplo

Durante o transcorrer deste trabalho, para exemplificações, será usada a mesma instância de exemplo presente da proposta do desafio. Esta consiste em quatro máquinas (2-14), três processos (2-15), dois recursos (2-16) dois serviços (2-17), duas vizinhanças (2-18) e três localidades (2-19)

$$\mathcal{M} = \{m_1, m_2, m_3, m_4\} \quad (2-14)$$

$$\mathcal{P} = \{p_1, p_2, p_3\} \quad (2-15)$$

$$\begin{aligned} \mathcal{R} &= \{r_1, r_2\} \\ \mathcal{T} &= \{r_1\} \\ \mathcal{B} &= \{(r_1, r_2, 20)\} \end{aligned} \quad (2-16)$$

$$\begin{aligned} \mathcal{S} &= \{s^a, s^b\} \\ s^a &= \{p_1, p_2\} \\ s^b &= \{p_3\} \\ s^b &\text{ depende de } s^a \\ \text{spreadMin}(s^a) &= 2 \\ \text{spreadMin}(s^b) &= 1 \end{aligned} \quad (2-17)$$

$$\begin{aligned} \mathcal{N} &= \{n_1, n_2\} \\ n_1 &= \{m_1, m_2\} \\ n_2 &= \{m_3, m_4\} \end{aligned} \quad (2-18)$$

$$\begin{aligned} \mathcal{L} &= \{l_1, l_2, l_3\} \\ l_1 &= \{m_1, m_2\} \\ l_2 &= \{m_3\} \\ l_3 &= \{m_4\} \end{aligned} \quad (2-19)$$

As capacidades e capacidades segura das máquinas estão descritas na Tabela 2.5. A Tabela 2.6 apresenta os custos de movimentação de máquina. Os requisitos dos processos, custo de movimentação estão na Tabela 2.7. Os pesos são mostrados em (2-20). A solução inicial é dada por (2-21)

Os pesos dos custos parciais são dados por (2-20)

Tabela 2.5: Exemplo: Capacidade e Capacidade segura

	C		SC	
	r_1	r_2	r_1	r_2
m_1	30	400	16	80
m_2	10	240	8	160
m_3	15	100	12	80
m_4	10	100	8	80

Tabela 2.6: Exemplo: Custo de Movimentação de Máquina - $MMC(m_i, m_j)$

Origem	Destino			
	m_1	m_2	m_3	m_m
m_1	0	1	4	5
m_2	1	0	3	4
m_3	4	3	0	2
m_4	5	4	2	0

Tabela 2.7: Exemplo: Requisitos dos processos e Custo de Movimentação de Processo - $PMC(p_i)$

	R		PMC
	r_1	r_2	
p_1	12	10	1000
p_2	10	20	100
p_3	6	200	1

$$\begin{aligned}
 weight_{loadCost}(r_1) &= 100 \\
 weight_{loadCost}(r_2) &= 10 \\
 weight_{balanceCost}(\langle r_1, r_2, 20 \rangle) &= 10 \\
 weight_{processMoveCost} &= 1 \\
 weight_{serviceMoveCost} &= 10 \\
 weight_{machineMoveCost} &= 100
 \end{aligned} \tag{2-20}$$

A solução inicial é mostrada em 2-21.

$$\begin{aligned}
 M_0(p_1) &= m_1 \\
 M_0(p_2) &= m_4 \\
 M_0(p_3) &= m_1
 \end{aligned} \tag{2-21}$$

Uma nova solução poderia ser, por exemplo:

$$M_0(p_1) = m_3$$

$$M_0(p_2) = m_1$$

$$M_0(p_3) = m_2$$

A solução inicial possui custo total de:

$$\underbrace{(400 + 1300)}_{\text{carga}} + \underbrace{2500}_{\text{balanceamento}} + \underbrace{0}_{\text{movimentação de processos}} + \underbrace{0}_{\text{movimentação de serviços}} + \underbrace{0}_{\text{movimentação de máquinas}} = 4200$$

Ao mover o processo p_2 da máquina m_4 para m_3 , por exemplo, reduz-se o custo de balanceamento de 2500 para 1700 e o custo de carga do recurso r_1 de 400 para 200. O novo custo passa a ser:

$$\underbrace{(200 + 1300)}_{\text{carga}} + \underbrace{1700}_{\text{balanceamento}} + \underbrace{100}_{\text{movimentação de processos}} + \underbrace{10}_{\text{movimentação de serviços}} + \underbrace{200}_{\text{movimentação de máquinas}} = 3510$$

A solução ótima é obtida ao mover o processo p_3 da máquina m_1 para m_2 , resultando em um custo total de:

$$\underbrace{(0 + 400)}_{\text{carga}} + \underbrace{1600}_{\text{balanceamento}} + \underbrace{101}_{\text{movimentação de processos}} + \underbrace{10}_{\text{movimentação de serviços}} + \underbrace{300}_{\text{movimentação de máquinas}} = 2411$$

Portanto a solução ótima é dada por:

$$M_0(p_1) = m_1$$

$$M_0(p_2) = m_3 \tag{2-22}$$

$$M_0(p_3) = m_2$$

2.7

Trabalhos anteriores

O desafio original foi proposto em 2012, para atender o biênio 2012-2013, portanto, o desafio está encerrado. O trabalho de (Afsar et al., 2016) apresenta as equipes finalistas, assim como cita características dos métodos utilizados para resolução. Muitos utilizaram combinações de heurísticas ou métodos exatos. Algumas equipes utilizaram somente um único tipo de heurística em suas soluções propostas. O artigo (Gabay & Zaourar, 2012) expõe a metodologia

utilizada pela equipe dos autores.

No site do proponente, há um documento que explica a metodologia usada para gerar as instâncias de classificação e avaliação (b e x respectivamente) e também o resultado das equipes classificadas. Foram fornecidas premiações, nos valores de 3.000, 1.500 e 500 euros para as três melhores equipes *júniors*, cujos grupos foram compostos somente por estudantes. Houve uma premiação de 5.000 euros para melhor equipe *sênior*, que não possuía restrição de membros. Também houve uma premiação extra de 10.000 euros para a equipe cuja solução usou somente componentes de código livre. Tanto os grupos *júniors*, quanto o grupo *sênior* premiados participavam também desta terceira categoria. A premiação foi para a mesma equipe que ganhou no grupo *sênior*.

Mesmo com o desafio encerrado e resolvido, o material do desafio é utilizado como base para outros trabalhos relacionados após 2012. Em (Mehta et al., 2012), os autores propõem uma seleção de um subproblema, por meio de uma heurística de *large neighborhood search* que escolhe um subconjunto de máquinas e processos. Aplica-se então uma otimização no subproblema. Um método parecido é apresentado em (Jaskowski et al., 2015) e em (Mrad et al., 2015).

No artigo (Gabay & Zaourar, 2013), os autores propõem uma heurística para identificar alocação de processos factíveis. Também adaptam heurísticas clássicas e propõem novas para a resolução de problema de empacotamento.

Em (Wang et al., 2016) as vizinhanças de são foram particionadas em conjuntos menores, onde aplica-se uma heurística de busca local entre essas sub vizinhanças.

O trabalho de (Butelle et al., 2015), propõe uma metodologia paralela, onde uma heurística de *Adaptive Variable Neighborhood Search* trabalha em conjunto com uma hiper-heurística de *Simulated Annealing*. A hiper-heurística trabalha na seleção de um método e aceitação desse método escolhido. O trabalho de (Hoffman et al., 2015) também expõe uma hiper-heurística baseada em *Simulated annealing*, onde uma heurística busca uma vizinhança de soluções maiores e outra busca soluções dentro de um conjunto.

Em (Gavranovic et al., 2012), os autores aplicam *Variable Neighborhood Search* no problema. Nos resultados obtidos considerou-se o melhor objetivo obtido após várias execuções, usando sementes com sementes diferentes para

cada execução.

No trabalho (Masson et al., 2013), os autores propõe-se uma heurística de *multi-start iterated local search* para o problema de empacotamento.

Alguns trabalhos derivados propõem uma versão multi-objetivo do problema, como em (Saber et al., 2015). Usam uma programação inteira para gerar soluções iniciais não-dominadas, que são a base para a aplicação de uma meta-heurística multi-objetivo, no caso *GeNePi* descrito em (Saber et al., 2014). Outro algoritmo evolutivo para uma outra variação multi-objetivo é apresentado em (Legillon et al, 2015).

Como o patrocinador das premiações do desafio foi a Google, que é considerada como referência no *estado da prática* em relação à gestão de *data center* e serviços de nuvens, há outros trabalhos que exploram otimizações desses recursos ou serviços, como por exemplo (Xu & Foters, 2011) e (Ries et al., 2012), porém desvinculadas dos dados do desafio.

Formulação de Programação Inteira

O problema do desafio contém vários elementos. Estes fazem parte de algum dos seguintes conjuntos:

- \mathcal{P} Conjunto dos processos
- \mathcal{M} Conjunto das máquinas
- \mathcal{R} Conjunto dos recursos comuns de todas as máquinas
- \mathcal{T} Conjunto dos recursos que possuem utilização transitória
- \mathcal{S} Conjunto dos serviços
- \mathcal{L} Conjunto das localidades
- \mathcal{N} Conjunto das vizinhanças

Todos os conjuntos enumerados no problema são finitos, portanto possuem cardinalidade P , M , R , S , L e N respectivamente. Em outras palavras, o número de elementos de cada um é finito, portanto são enumeráveis. Assim, existe uma relação bijetiva entre \mathcal{P} e $I_P = \{1, 2, 3, \dots, P\} \in \mathbb{N}$, por exemplo. Dessa forma, pode-se atribuir valores numéricos para cada um dos processos p de \mathcal{P} . No restante deste trabalho, será utilizada a notação $p \in \mathcal{P}$ tanto para representar os processos propriamente quanto sua equivalência com os Naturais. De forma análoga vale para os demais conjuntos.

Pelo fato das principais variáveis serem de escopo finito e enumerável, este problema pode ser modelado por meio de uma *Programação Inteira* (MIP de Mixed Integer Programming). Observa-se que embora as variáveis utilizadas sejam inteiras, este método também permite o uso de variáveis contínuas. Logo, pode ser representado por um conjunto de operações lineares, e o domínio das variáveis são subconjuntos dos Inteiros. Para todo problema de programação inteira, existe uma versão de relaxação linear, ou seja, onde as variáveis são contínuas.

3.1

Variáveis de decisão para programação Inteira

Para uma formulação de programação inteira, define-se a variável x_{pm} como:

$$x_{pm} = \begin{cases} 1 & \text{se o processo } p \text{ está alocado na máquina } m. \\ 0 & \text{caso contrário.} \end{cases} \quad (3-1)$$

A alocação inicial de x_{pm} é denotada por uma constante x_{pm}^0 , onde:

$$x_{pm}^0 = \begin{cases} 1 & \text{se o processo } p \text{ possui alocação inicial na máquina } m. \\ 0 & \text{caso contrário.} \end{cases} \quad (3-2)$$

3.2

Constantes usados no modelo

Além da constante x_{pm}^0 , na formulação de programação inteira deste problema, são utilizadas várias outras constantes relativas às características dos processos, máquinas e serviços.

De forma geral, as constantes já definidas no Capítulo 2 são as mesmas, apenas usando uma notação indexada. Por exemplo a constante $R(p, r)$ aparecerá como R_{pr} . Algumas constantes tiveram sua notação alterada, como por exemplo $spreadMin(s)$ que será denotada por δ_s . A constante $target$ da tupla de balanceamento $b = \langle r_1, r_2, target \rangle$ será denotada por $B_{r_1 r_2}$ para equilibrar os recursos $r_1 \in \mathcal{R}$ e $r_2 \in \mathcal{R}$.

As constantes $weight_{loadCost}(r)$, $weight_{balanceCost}(r_1, r_2)$, $weight_{processMoveCost}$, $weight_{serviceMoveCost}(r)$, $weight_{machineMoveCost}(r)$, também tiveram as notações alteradas para W_r^{lc} , $W_{r_1 r_2}^{bc}$, W^{pmc} , W^{smc} e W^{mmc} respectivamente.

A constante γ_{ss^d} indica a relação de dependência entre o serviço $s \in \mathcal{S}$ e $s^d \in \mathcal{S}$, formalmente:

$$\gamma_{ss^d} = \begin{cases} 1 & \text{se o serviço } s \text{ é dependente do serviço } s^d \\ 0 & \text{caso contrário.} \end{cases} \quad (3-3)$$

3.3

Variáveis auxiliares

Foram definidas algumas variáveis auxiliares que permitem ter conhecimento das movimentações de processos, da utilização dos recursos, da presença de serviços. Com base nesse conhecimento, é possível implementar as restrições rígidas do desafio.

Variáveis de migração de processo

Com base na definição de x_{pm} , pode-se definir as variáveis auxiliares z_{pm}^+ e z_{pm}^- que informam se o processo $p \in \mathcal{P}$ move-se para a (ou saiu da) máquina $m \in \mathcal{M}$ respectivamente. Formalmente:

$$z_{pm}^+ \geq x_{pm} - x_{pm}^0 \quad (\forall p \in \mathcal{P}, \forall m \in \mathcal{M}) \quad (3-4)$$

$$z_{pm}^- \geq x_{pm}^0 - x_{pm} \quad (\forall p \in \mathcal{P}, \forall m \in \mathcal{M}) \quad (3-5)$$

Variáveis de movimento migratório de processo

Das variáveis de migração z_{pm}^+ e z_{pm}^- de processos $p \in \mathcal{P}$ em relação à máquina $m \in \mathcal{M}$ podem-se obter as variáveis y_{pij} que indicam se o processo p migrou da máquina $i \in \mathcal{M}$ para a máquina $j \in \mathcal{M}$.

$$y_{pij} \geq z_{pi}^- + z_{pj}^+ - 1 \quad (\forall p \in \mathcal{P}, \forall i, j \in \mathcal{M}) \quad (3-6)$$

$$y_{pij} \leq z_{pi}^- \quad (\forall p \in \mathcal{P}, \forall i, j \in \mathcal{M}) \quad (3-7)$$

$$y_{pij} \leq z_{pj}^+ \quad (\forall p \in \mathcal{P}, \forall i, j \in \mathcal{M}) \quad (3-8)$$

Variáveis de fluxo migração

Consequentemente pode-se definir a variável t_{ij} que representa quantos processos migraram da máquina i para j :

$$t_{ij} = \sum_{p \in \mathcal{P}} y_{pij} \quad (\forall i, j \in \mathcal{M}) \quad (3-9)$$

Variáveis de utilização

Também definem-se as variáveis u_{mr} relativas à utilização do recurso $r \in \mathcal{R}$ na máquina $m \in \mathcal{M}$. Formalmente:

$$u_{mr} = \sum_{p \in \mathcal{P}} R_{pr} x_{pm} \quad (\forall r \in \mathcal{R}, \forall m \in \mathcal{M}) \quad (3-10)$$

Variável de utilização transitória

Também define-se a utilização transitória u_{mr}^T de um recurso $r \in \mathcal{T}$ da máquina $m \in \mathcal{M}$ como:

$$u_{mr}^T = \sum_{p \in \mathcal{P}} R_{pr} z_{pm}^- \quad (\forall r \in \mathcal{T}, \forall m \in \mathcal{M}) \quad (3-11)$$

Para os recursos que não possuem utilização transitória, pode-se considerar que:

$$u_{mr}^T = 0 \quad (\forall r \in \mathcal{R}, r \notin \mathcal{T}, \forall m \in \mathcal{M}) \quad (3-12)$$

Variáveis de sobrecarga

Com base na utilização u_{mr} , pode-se definir a variável d_{mr} que informa a quantidade de recursos $r \in \mathcal{R}$ da máquina $m \in \mathcal{M}$ acima da capacidade segura SC_{mr} . Formalmente:

$$d_{mr} \geq u_{mr} - SC_{mr} \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-13)$$

Variáveis de disponibilidade de recursos

Também define-se a disponibilidade a_{mr} do recurso $r \in \mathcal{R}$ da máquina $m \in \mathcal{M}$ como:

$$a_{mr} = C_{mr} - u_{mr} \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-14)$$

Variáveis de balanceamento

Com base na disponibilidade a_{mr} , pode-se obter a variável $b_{mr_1r_2}$ que mensura o balanceamento entre os recursos $r_1 \in \mathcal{R}$ e $r_2 \in \mathcal{R}$ da máquina $m \in \mathcal{M}$. Formalmente:

$$b_{mr_1r_2} \geq B_{r_1r_2}a_{mr_1} - a_{mr_2} \quad (\forall m \in \mathcal{M}, \forall r_1, r_2 \in \mathcal{R}) \quad (3-15)$$

Variáveis de presença de serviço na localidade

A variável binária o_{sl} indica se um serviço $s \in \mathcal{S}$ está presente na localidade $l \in \mathcal{L}$. Formalmente, tem-se como limite inferior:

$$o_{sl} \geq x_{pm} \quad (\forall p \in s \in \mathcal{S}, \forall m \in l \in \mathcal{L}) \quad (3-16)$$

O limite superior é definido por:

$$o_{sl} \leq \sum_{\substack{p \in s \\ m \in l}} x_{pm} \quad (\forall s \in \mathcal{S}, \forall l \in \mathcal{L}) \quad (3-17)$$

É necessário informar o limite superior, pois como a variável o_{sl} não possui custo no objetivo, então a solução $o_{sl} = 1$ para todos $s \in \mathcal{S}, l \in \mathcal{L}$ seria possível caso não existisse a restrição definida pela Equação (3-17).

Variável de presença de serviço na vizinhança

Definiu-se a variável h_{sn} para indicar se um serviço $s \in \mathcal{S}$ possui algum processo alocado em alguma máquina da vizinhança $n \in \mathcal{N}$. Formalmente:

$$h_{sn} \geq x_{pm} \quad (\forall p \in s \in \mathcal{S}, \forall m \in n \in \mathcal{N}) \quad (3-18)$$

De forma análoga a o_{sl} , definiu-se também um limite superior:

$$h_{sn} \leq \sum_{\substack{p \in s \\ m \in n}} x_{pm} \quad (\forall s \in \mathcal{S}, \forall n \in \mathcal{N}) \quad (3-19)$$

Variável de movimentação de serviço

Tanto as variáveis z_{pm}^+ quanto z_{pm}^- permitem contar a quantidade de processos migrados de um serviço $s \in \mathcal{S}$. Foi definida a variável g_s da seguinte forma:

$$g_s = \sum_{\substack{p \in s \\ m \in \mathcal{M}}} z_{pm}^+ \quad (\forall s \in \mathcal{S}) \quad (3-20)$$

Critério de positividade

Todas as variáveis u_{mr} , u_{mr}^T , d_{mr} e a_{mr} somente assumem valores inteiros positivos:

$$u_{mr}, u_{mr}^T, d_{mr}, a_{mr} \geq 0 \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-21)$$

3.4

Restrições para Programação Inteira

Seguem as formulações das restrições definidas no desafio e enumeradas na Seção 2.2 para o modelo proposto.

Alocação de processos

Esta é uma restrição implícita, pois está definida na definição do desafio, e não em um tópico do mesmo. A restrição de *Alocação de processos* garante que todos os processos são alocados em exatamente uma máquina conforme descrito em na Seção 2.1, formalmente:

$$\sum_{m \in \mathcal{M}} x_{pm} = 1 \quad (\forall p \in \mathcal{P}) \quad (3-22)$$

Restrição de Capacidade

A *Restrição de capacidade* apresentada na Seção 2.2.1, define um limite superior para as variáveis u_{mr} da seguinte forma:

$$u_{mr} \leq C_{mr} \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-23)$$

Como as variáveis u_{mr} , u_{mr}^T , a_{mr} e d_{mr} são relacionadas e todas mensuram alguma característica de quantidades de recursos, os limites dessas podem ser sumarizadas em:

$$u_{mr}, u_{mr}^T, a_{mr}, d_{mr} \in [0, C_{mr}] \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-24)$$

Restrição de Conflito

A restrição de conflito descrita na Seção 2.2.2 pode ser formulada como:

$$\sum_{p \in s} x_{pm} \leq 1 \quad (\forall s \in \mathcal{S}, \forall m \in \mathcal{M}) \quad (3-25)$$

Restrição de espalhamento

A restrição de espalhamento apresentada na Seção 2.2.3 pode ser formulada com base na variável o_{sl} como:

$$\sum_{l \in \mathcal{L}} o_{sl} \geq \delta_s \quad (\forall s \in \mathcal{S}) \quad (3-26)$$

Restrição de dependência

Para a restrição de dependência apresentada na Seção 2.2.4 faz-se uso das variáveis auxiliares h_{sn} :

$$h_{sn} \leq h_{s^d n} \quad (\forall s, s^d \in \mathcal{S}, \gamma_{ss^d} > 0, \forall n \in \mathcal{N}) \quad (3-27)$$

Restrição de utilização transitória

Como as variáveis u_{mr} e u_{mr}^T contabilizam a utilização do recurso $r \in \mathcal{R}$ dos processos que estão presentes e que saíram da máquina $m \in \mathcal{M}$

respectivamente, não contabilizam simultaneamente os requisitos R_{pr} de um mesmo processo $p \in \mathcal{P}$, pois este não pode estar presente e ter saído da máquina na mesma máquina. Portanto, a restrição da Equação (2.2.5) pode ser descrita como:

$$u_{mr} + u_{mr}^T \leq C_{mr} \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-28)$$

Nota-se que esta restrição domina a restrição de capacidade, Equação (3-23).

3.5

Objetivo

A função objetivo é composta por uma combinação de outras funções objetivos.

Custo de Carga

A parte relativa à *Custo de carga* é calculada com base nas variáveis de carga d_{mr} do recurso $r \in \mathcal{R}$ na máquina $m \in \mathcal{M}$, podendo ser descrita por:

$$lc_r = \sum_{m \in \mathcal{M}} d_{mr} \quad (\forall r \in \mathcal{R}) \quad (3-29)$$

Custo de balanceamento

A parte do *Custo de balanceamento* é obtida com base nas variáveis $b_{mr_1r_2}$ da seguinte forma:

$$bc_{r_1 r_2} = \sum_{m \in \mathcal{M}} b_{mr_1 r_2} \quad (\forall r_1, r_2 \in \mathcal{R}) \quad (3-30)$$

Custo de movimentação de processo

A componente do *Custo de movimentação de processo* pode ser obtida com uso de uma das variáveis de movimentação z_{pm}^+ ou z_{pm}^- . Optou-se pelo uso da variável z_{pm}^+ . Formalmente tem-se:

$$pmc = \sum_{\substack{p \in \mathcal{P} \\ m \in \mathcal{M}}} PMC_p z_{pm}^+ \quad (3-31)$$

Custo de Movimentação de Serviço

Com base nas variáveis g_s com $s \in \mathcal{S}$, pode-se obter a parcela do *custo de movimentação de serviço*:

$$smc \geq g_s \quad (\forall s \in \mathcal{S}) \quad (3-32)$$

Custo de Movimentação de Máquinas

As variáveis t_{ij} com $i, j \in \mathcal{M}$ permitem obter o *custo de movimentação de máquinas* da seguinte forma:

$$mmc = \sum_{\substack{i \in \mathcal{M} \\ j \in \mathcal{M}}} MMC_{ij} t_{ij} \quad (3-33)$$

Custo Total

O objetivo final pode então ser calculado como a soma ponderada das componentes anteriores:

$$obj = \sum_{r \in \mathcal{R}} W_r^{lc} l c_r + \sum_{\substack{r1 \in \mathcal{R} \\ r2 \in \mathcal{R}}} W_{r1r2}^{bc} \cdot b c_{r1r2} + \\ W^{pmc} p m c + W^{smc} s m c + W^{mmc} m m c \quad (3-34)$$

3.6

Ensaio preliminares para avaliação

As seções anteriores (3.1 - 3.5) enumeram e descrevem os elementos necessários para implementar um primeiro modelo por programação inteira, sendo analisados os desempenhos do modelo em relação à resposta final, ao tempo de execução e ao consumo de memória.

Este primeiro modelo considera que todas as variáveis são inteiras, pois as variáveis de decisão x_{pm} são inteiras e, por premissa, todos os dados das instâncias são inteiros. Como o produto, soma e subtração de inteiros também é inteiro, os resultados obtidos são inteiros.

Ao aplicar este modelo na instância de exemplo, obtém-se como resposta que o processo p_1 deve ser atribuído para a máquina m_1 , p_2 na máquina m_3 e p_3 em m_2 , pois $x_{0,0} = x_{1,2} = x_{2,1} = 1$ e os demais x_{pm} são nulos. Esta configuração de alocação resulta em um custo final de 2411. Para resolver o exemplo, são necessários 301 linhas e 169 colunas, onde existem 768 elementos não nulos. As 169 colunas representam as variáveis, que são todas inteiras e 48 dessas são binárias.

Os resultados das execuções desse modelo nas instâncias a1 são apresentados na Tabela 3.1, onde a coluna *Objetivo* informa o valor do objetivo obtido. A coluna *Razão* é a razão deste objetivo obtido em relação à solução trivial. A coluna *Tempo* informa o tempo total de execução. A coluna *Critério* informa qual o critério de parada adotado, podendo ser:

Concluída Informa que a execução foi concluída como ótima.

Tempo Informa que a execução foi interrompida por atingir o limite de tempo.

Memória Informa que a execução foi interrompida por falta de memória.

No caso de *concluída*, a resposta é considerada como ótima quando a diferença entre os limites inferior e superior do objetivo, está na tolerância de 1.0×10^{-4} em relação ao valor absoluto do limite superior.

Além dos parâmetros de limite de tempo, são configurados no Gurobi os parâmetros para usar ponto flutuante de precisão quádrupla nas execuções dos Simplex (Quad=1) e o parâmetro para lidar com problemas numéricos (NumericFocus=3).

Tabela 3.1: Resultado parcial usando modelos *MIP1*

Instância	<i>MIP1</i>			
	Objetivo	Razão	Tempo	Critério
a1_1	44.306.501	0.894	0,706	Concluída
a1_2	não resolveu	-	1398	Memória
a1_3	não resolveu	-	1369	Memória
a1_4	não resolveu	-	3815	Memória
a1_5	727.578.610	0.930	2424,695	Concluída

Neste modelo, denominado de *MIP1*, para as instâncias a1_2, a1_3 e a1_4, não é possível obter resposta pois estes são abortados por excesso de memória. Conclui-se então que o modelo até então proposto apresenta uma ineficiência relacionada ao alto consumo de memória.

Pode-se observar que devem ser computados PM^2 valores de y_{pij} e existirão um mesmo número, em ordem de grandeza, de restrições para seu cômputo. Na instância a1_2 (com $P = 1000$ e $M = 100$), por exemplo, existem na ordem de 10^7 variáveis e restrições. Esta variável somente é utilizada para calcular t_{ij} . Uma melhoria possível para reduzir o uso de memória é redefinir a variável t_{ij} que conta a quantidade de processos migrados da máquina $i \in \mathcal{M}$ para a máquina $j \in \mathcal{M}$ como:

$$t_{ij} = \sum_{p \in \mathcal{P}} x_{pi}^0 x_{pj} \quad (\forall i \in \mathcal{M}, \forall j \in \mathcal{M}) \quad (3-35)$$

Vale ressaltar que x_{pm}^0 é uma constante que indica se o processo $p \in \mathcal{P}$ estava inicialmente alocado na máquina $m \in \mathcal{M}$ (ver Seção 3.1), portanto esta nova definição ainda é linear, quando relaxada. Não é uma restrição quadrática. Esta restrição somente soma os x_{pj} cuja origem é a máquina i .

Com essa nova definição de t_{ij} , a variável y_{pij} que informa se o processo $p \in \mathcal{P}$ migrou de máquina da máquina i para a máquina j não é mais necessária com esta nova estratégia.

Como mencionado na Seção 3.1 nas definições de g_s e de pmc , podia-se usar qualquer uma das variáveis de migração z_{pm}^+ ou z_{pm}^- . Optando por usar sempre z_{pm}^- , a variável z_{pm}^+ não é mais necessária. No resto deste documento utiliza-se a variável z_{pm} com a mesma definição de z_{pm}^- :

$$z_{pm} \geq x_{pm}^0 - x_{pm} \quad (\forall p \in \mathcal{P}, \forall m \in \mathcal{M}) \quad (3-36)$$

3.7

Formulação

Com base nos elementos previamente enumerados, pode-se construir um segundo modelo para resolver o desafio por programação inteira. Este modelo completo é descrito por:

$$\begin{aligned} \min \quad & \sum_{r \in \mathcal{R}} W_r^{lc} l c_r \\ & + \sum_{\substack{r1 \in \mathcal{R} \\ r2 \in \mathcal{R}}} W_{r1r2}^{bc} b c_{r1r2} \\ & + W^{pmc} pmc \\ & + W^{smc} smc \\ & + W^{mmc} mmc \end{aligned} \quad (3-34)$$

sujeito a:

$$\sum_{m \in \mathcal{M}} x_{pm} = 1 \quad (\forall p \in \mathcal{P}) \quad (3-22)$$

$$\sum_{p \in \mathcal{S}} x_{pm} \leq 1 \quad (\forall s \in \mathcal{S}, \forall m \in \mathcal{M}) \quad (3-25)$$

$$z_{pm} \geq x_{pm}^0 - x_{pm} \quad (\forall p \in \mathcal{P}, \forall m \in \mathcal{M}) \quad (3-36)$$

$$u_{mr} = \sum_{p \in \mathcal{P}} R_{pr} x_{pm} \quad (\forall r \in \mathcal{R}, \forall m \in \mathcal{M}) \quad (3-10)$$

$$u_{mr}^T = \sum_{p \in \mathcal{P}} R_{pr} z_{pm} \quad (\forall r \in \mathcal{T}, \forall m \in \mathcal{M}) \quad (3-11)$$

$$u_{mr}^T = 0 \quad (\forall r \in \mathcal{R}, r \notin \mathcal{T}, \forall m \in \mathcal{M}) \quad (3-12)$$

$$u_{mr} + u_{mr}^T \leq C_{mr} \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-28)$$

$$d_{mr} \geq u_{mr} - SC_{mr} \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-13)$$

$$lc_r = \sum_{m \in \mathcal{M}} d_{mr} \quad (\forall r \in \mathcal{R}) \quad (3-29)$$

$$a_{mr} = C_{mr} - u_{mr} \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-14)$$

$$b_{mr_1 r_2} = B_{r_1 r_2} a_{mr_1} - a_{mr_2} \quad (\forall m \in \mathcal{M}, \forall r_1, r_2 \in \mathcal{R}) \quad (3-15)$$

$$bc_{r_1 r_2} = \sum_{m \in \mathcal{M}} b_{mr_1 r_2} \quad (\forall r_1, r_2 \in \mathcal{R}) \quad (3-30)$$

$$pmc = \sum_{\substack{p \in \mathcal{P} \\ m \in \mathcal{M}}} PMC_p z_{pm}$$

$$o_{sl} \geq x_{pm} \quad (\forall p \in s \in \mathcal{S}, \forall m \in l \in \mathcal{L}) \quad (3-16)$$

$$o_{sl} \leq \sum_{\substack{p \in s \\ m \in l}} x_{pm} \quad (\forall s \in \mathcal{S}, \forall l \in \mathcal{L}) \quad (3-17)$$

$$\sum_{l \in \mathcal{L}} o_{sl} \geq \delta_s \quad (\forall s \in \mathcal{S}) \quad (3-26)$$

$$h_{sn} \leq h_{s^n n} \quad (\forall s, s^d \in \mathcal{S} | \gamma_{ss^d} = 1, \forall n \in \mathcal{N}) \quad (3-27)$$

$$g_s = \sum_{\substack{p \in s \\ m \in \mathcal{M}}} z_{pm} \quad (\forall s \in \mathcal{S}) \quad (3-20)$$

$$smc \geq g_s \quad (\forall s \in \mathcal{S}) \quad (3-32)$$

$$t_{ij} = \sum_{p \in \mathcal{P}} x_{pi}^0 x_{pj} \quad (\forall i, j \in \mathcal{M}) \quad (3-35)$$

$$mmc = \sum_{\substack{i \in \mathcal{M} \\ j \in \mathcal{M}}} MMC_{ij} t_{ij} \quad (3-33)$$

$$u_{mr}, u_{mr}^T \geq 0 \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-21)$$

$$d_{mr}, a_{mr} \geq 0 \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R})$$

$$b_{mr_1 r_2} \geq 0 \quad (\forall m \in \mathcal{M}, \forall r_1, r_2 \in \mathcal{R})$$

$$t_{ij} \geq 0 \quad (\forall i, j \in \mathcal{M})$$

$$g_s \geq 0 \quad (\forall s \in \mathcal{S})$$

$$lc_r \geq 0 \quad (\forall r \in \mathcal{R})$$

$$bc_{r_1 r_2} \geq 0 \quad (\forall r_1 \in \mathcal{R}, \forall r_2 \in \mathcal{R})$$

$$pmc \geq 0$$

$$smc \geq 0$$

$$\begin{aligned}
 mmc &\geq 0 \\
 x_{pm}, z_{pm} &\in \{0, 1\} && (\forall p \in \mathcal{P}, \forall m \in \mathcal{M}) \\
 o_{sl} &\in \{0, 1\} && (\forall s \in \mathcal{S} \forall l \in \mathcal{L}) \\
 h_{sn} &\in \{0, 1\} && (\forall s \in \mathcal{S}, \forall n \in \mathcal{N})
 \end{aligned}$$

Pelo mesmo motivo do primeiro modelo, neste novo modelo, *MIP2*, considera-se que todas as variáveis são inteiras.

Ao aplicar este novo modelo na instância de exemplo, obtem-se como resposta $x_{0,0} = x_{1,2} = x_{2,1} = 1$ e os demais x_{pm} são nulos. Esta configuração tem custo final de 2411. Portanto, deve-se alocar os processos p_1 , p_2 e p_3 nas máquinas m_1 , m_3 e m_2 respectivamente. Para resolver esta instância, o programa possui 145 linhas e 109 colunas, das quais 390 elementos são não nulos. Das 109 variáveis, que são todas inteiras, 36 são binárias.

Neste segundo modelo, de forma análoga ao primeiro, também será analisado o desempenho do modelo em relação à resposta final, ao tempo de execução e ao consumo de memória.

Os resultados das execuções desse modelo nas instâncias **a1** são apresentados na Tabela 3.2. Para melhor compreensão, deve ser lembrado que a coluna *Objetivo* informa o valor do objetivo obtido. A coluna *Razão* é a razão deste objetivo obtido em relação à solução trivial. A coluna *Tempo* informa o tempo total de execução. A coluna *Critério* informa qual o critério de parada adotado, que são os mesmos da Tabela 3.1.

Além dos parâmetros de limite de tempo, foram configurados no Gurobi os parâmetros para usar ponto flutuante de precisão quádrupla nas execuções dos Simplex (*Quad=1*) e o parâmetro para lidar com problemas numéricos (*NumericFocus=3*), que são os mesmos do *MIP1*.

Tabela 3.2: Resultado parcial usando modelos MIP2

Instância	MIP2			
	Objetivo	Razão	Tempo	Critério
a1_1	44.306.501	0,894	0,295	Concluída
a1_2	804.180.142	0,757	3581,654	Concluída
a1_3	583.318.987	0,999	523,072	Concluída
a1_4	383.721.957	0,606	21600,421	Tempo
a1_5	727.578.312	0,930	18,337	Concluída

Dos resultados, pode-se notar que somente a instância `a1_4` foi interrompida por tempo, apresentando uma diferença de 3.7% em relação ao limite inferior de otimalidade. As demais instâncias terminaram como ótimas dentro do limite de tolerância relativa.

O modelo *MIP2* apresenta, nas instâncias utilizadas, grande amplitude no limite de variáveis, assim como nos coeficientes e termos independentes. A Tabela 3.3 contém os limites de amplitudes do modelo nessas instâncias.

Tabela 3.3: Faixa de valores no modelo MIP2

Instância	Coeficientes	Objetivos	Limites	Termos Independentes
<code>a1_1</code>	1 a 1×10^6	1 a 1×10^2	1 a 5×10^6	1 a 5×10^6
<code>a1_2</code>	1 a 7×10^6	1 a 1×10^2	1 a 2×10^7	1 a 2×10^7
<code>a1_3</code>	1 a 6×10^6	1 a 1×10^2	1 a 1×10^7	1 a 1×10^7
<code>a1_4</code>	1 a 7×10^6	1 a 1×10^2	1 a 2×10^7	1 a 2×10^7
<code>a1_5</code>	1 a 6×10^6	1 a 1×10^2	1 a 3×10^7	1 a 3×10^7

Com base em uma apresentação da equipe do Gurobi sobre boas práticas (Mars, 2016), são sugeridas algumas abordagens para melhorar o desempenho da execução do modelo, como:

- considerar as variáveis com grande amplitude como sendo contínuas;
- reduzir a amplitude dos coeficientes.

Elabora-se então um terceiro modelo, *MIP3*, baseado no *MIP2*, porém considerando como contínuas as variáveis de utilização, disponibilidade, carga, balanceamento, utilização transitória, as de custo de movimentação (processos, serviços e máquinas). No caso das variáveis relacionadas com recursos da máquina, estas podem variar de 0 até C_{mr} , por exemplo. Já as variáveis de custos de movimentação podem assumir grandes valores, caso muitos processos sofram migração e não haja custo nulo de movimentação, ou seja, este novo modelo possui as mesmas equações do modelo anterior, porém altera o domínio de algumas variáveis.

A aplicação deste novo modelo na instância de exemplo, gera como resposta $x_{0,0} = x_{1,2} = x_{2,1} = 1$, as mesmas soluções de antes. Para resolver esta instância, o programa possui 145 linhas e 109 colunas, dos quais 390 elementos são não nulos. Das 109 variáveis, são 57 contínuas e as demais 52 são inteiras, incluindo 36 binárias.

O resultado da execução deste novo modelo está presente na Tabela 3.4.

Tabela 3.4: Resultado parcial usando modelos *MIP3*

Instância	<i>MIP3</i>			
	Objetivo	Razão	Tempo	Critério
a1_1	44.306.501	0,894	0,293	Concluída
a1_2	804.166.728	0,757	3564,085	Concluída
a1_3	583.322.849	0,999	552,797	Concluída
a1_4	381.150.275	0,602	21600,420	Tempo
a1_5	727.578.310	0,930	18,408	Concluída

Assim como no *MIP2*, a instância **a1_4** não termina por tempo, com diferença de 3,00% para o limite inferior de otimalidade. Para as instâncias **a1_2** e **a1_5**, as configurações obtidas mostram resultado melhor que o *MIP2*. Somente para a instância **a1_3** este modelo apresenta resultados piores.

Em relação ao tempo de execução, os tempos para as instâncias **a1_1** e **a1_2** no *MIP3* são praticamente os mesmos em relação ao modelo *MIP2*, houve menos de 1% de ganho. Para a instância **a1_5** o tempo também é praticamente o mesmo, 1% mais lento. Já para a instâncias **a1_3** o desempenho foi pior, onde é necessário 5,6% a mais de tempo.

O modelo *MIP3* ainda contém as mesmas variações de amplitudes do *MIP2*. Visando reduzir essa amplitude, é elaborado um quarto modelo, *MIP4*, tendo como base o *MIP3*. Procura-se reduzir os valores dos pesos dos objetivos e os limites de variáveis e coeficientes do problema.

Para reduzir as amplitudes dos coeficientes do objetivo, pode-se dividir os pesos W^{lc} , W^{bc} , W^{pmc} , W^{smc} e W^{mmc} por uma constante W^{min} que é o menor peso não nulo.

Para a matriz de coeficientes, usa-se o fato que os recursos possuem uma natureza vetorial, os requisitos no recurso r_1 somente somam com outros requisitos no mesmo recurso. Por exemplo, um requisito de CPU não pode ser suprido pela RAM, e vice-versa. Os requisitos são somados entre si quando há balanceamento (Equação (3-15)) e no objetivo parcial de carga (Equação (3-29)).

Portanto, escala-se os requisitos R_{pr} de cada recurso em relação ao menor requisito de cada recurso. Com isso, em cada recurso r , o menor requisito passa a ser 1. A redução de R_{pr} , resulta na redução da variação das variáveis u_{mr} , a_{mr} , d_{mr} . Também é necessário escalar as constantes C_{mr} e SC_{mr} . Então, para cada recurso $r \in \mathcal{R}$, obtém-se novos coeficientes R_{pr}^* , C_{mr}^* e SC_{mr}^* :

$$R_{pr}^* = \frac{R_{pr}}{\min_{p \in \mathcal{P}}(R_{pr})} \quad (\forall p \in \mathcal{P}, \forall r \in \mathcal{R}) \quad (3-37)$$

$$C_{mr}^* = \frac{C_{mr}}{\min_{p \in \mathcal{P}}(R_{pr})} \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-38)$$

$$SC_{mr}^* = \frac{SC_{mr}}{\min_{p \in \mathcal{P}}(R_{pr})} \quad (\forall m \in \mathcal{M}, \forall r \in \mathcal{R}) \quad (3-39)$$

A aplicação deste novo modelo na instância de exemplo, gera uma resposta dada por $x_{0,0} = x_{1,3} = x_{2,1} = 1$, que resulta em um custo final de 3111, ou seja, a solução obtida não é ótima. O programa constitui-se de 145 linhas e 109 colunas, das quais 390 elementos são não nulos. Das 109 variáveis, tem-se 57 contínuas e as demais 52 são inteiras, incluindo 36 binárias.

O resultado deste novo modelo é apresentado na Tabela 3.5 e a variação dos coeficientes na Tabela 3.6.

Tabela 3.5: Resultado parcial usando modelos MIP4

Instância	MIP4			
	Objetivo	Razão	Tempo	Critério
a1_1	44.306.501	0,894	0,288	Concluída
a1_2	804.713.563	0,757	4080,830	Concluída
a1_3	583.304.086	0,999	560,361	Concluída
a1_4	393.439.494	0,622	3566,204	Concluída
a1_5	727.603.785	0,930	14,530	Concluída

Tabela 3.6: Faixa de valores no modelo MIP4

Instância	Coefficientes	Objetivos	Limites	Termos Independentes
a1_1	1 a 2×10^3	1 a 1×10^2	1 a 7×10^3	1 a 7×10^3
a1_2	1 a 5×10^4	1 a 1×10^2	1 a 1×10^5	1 a 1×10^5
a1_3	1 a 4×10^4	1 a 1×10^2	1 a 7×10^4	1 a 7×10^4
a1_4	1 a 5×10^4	1 a 1×10^2	1 a 1×10^5	1 a 1×10^5
a1_5	1 a 4×10^4	1 a 1×10^2	1 a 2×10^5	1 a 2×10^5

A redução nos coeficientes permite a conclusão da instância a1_4, ou seja, o modelo termina com uma solução ótima. A solução fornecida é pior, de maior custo, se comparada aos demais métodos onde a mesma instância é interrompida por tempo. A solução fornecida também possui custo maior para as instâncias a1_2 e a1_5. Somente para a instância a1_3 o escalonamento permitiu ao modelo resultar em uma resposta melhor, de custo menor.

Em relação à tempo, para as instâncias *a1_1*, *a1_4* e *a1_5*, o tempo de execução é menor, se comparado aos modelos *MIP2* e *MIP3*. Para as instâncias *a1_2* e *a1_3*, o tempo é maior.

Há redução de amplitude nos coeficientes, nos termos independentes e limites de variáveis, onde ocorre uma redução de duas ordens de grandeza (10^2). Com isso, o intervalo de valores, nas instâncias utilizadas, possuem magnitude inferior a 10^6 . Levando-se em conta que a tolerância de erro padrão do Gurobi é de 10^{-6} , então o extremo inferior não está mais na tolerância de erro do extremo superior. Não há redução nos pesos dos objetivos, pois o menor peso já é a unidade.

3.8

Comentários

Com base nos resultados parciais obtidos, somente a instância *a1_1* teve a mesma resposta em todos os modelos. O modelo *MIP1* não é aplicável para instâncias maiores, pois termina por falta de memória.

Já os demais modelos, quando há resposta por conclusão no tempo, apresentaram respostas diferentes entre eles, dentro de uma margem de 0.002%. Uma possível causa seria uma dificuldade numérica resultante da precisão padrão do critério de parada. Como os objetivos são da ordem de 10^9 , então a tolerância relativa padrão do Gurobi de 10^{-4} pode resultar em erros absolutos na ordem de 10^5 . Comparando os resultados entre os modelos *MIP2* e *MIP3*, pode-se perceber que a diferença das respostas está dentro dessa ordem de grandeza.

Com base nos resultados apresentados, nas instâncias utilizadas, os modelos propostos não se mostram eficientes, pois há múltiplas respostas para o custo ótimo. Esta falta de unicidade de resposta está relacionada a uma dificuldade numérica devido à diferença de grandeza nos coeficientes e limites de variáveis. Mas como não se utilizam ajustes nos parâmetros de tolerância, as respostas de uma mesma instância em modelos distintos estão na mesma faixa de erro.

Também vale citar que alterar os dados dos problemas, a fim de reduzir a magnitude de variação não gera resultados na qualidade da resposta, uma vez que a instância de exemplo não obtém a resposta ótima. Somente há ganho de tempo para algumas instâncias.

Pode-se concluir que uma modelagem por programação inteira apresentou dificuldades para resolver o problema, apesar de ser uma abordagem direta de modelagem. Esta dificuldade é resultante das características dos dados. Portanto, deve-se buscar outras abordagens para lidar com essas dificuldades, que são principalmente pela amplitude dos números envolvidos

Geração de Colunas

O problema do desafio é em seu cerne um problema de alocação de tarefas em máquinas. Ao invés de decidir qual processo é alocado a qual máquina, como apresentado no capítulo anterior, se o problema contivesse todas as possíveis configurações de alocação de processos de cada máquina então a decisão passaria a ser: quais destas configurações fornece a solução de menor custo, atendendo aos requisitos.

Esta mudança de estratégia altera a modelagem envolvida e resulta que, para cada máquina, existam na ordem de 2^P configurações possíveis, onde P é o número de processos do problema, ou seja, neste caso, o problema teria um número exponencial de variáveis.

O método de *geração de colunas* (Ford & Fulkerson, 1958, Dantiz & Wolf, 1960) é uma técnica para se resolver problemas de grande porte onde é possível se considerar implicitamente as variáveis de um problema. Em particular, quando existe um problema cuja resolução equivale a determinar a variável de menor custo reduzido, pois é um problema de minimização. Para problemas de maximização, o problema deve determinar a variável de maior custo reduzido.

Este é um método iterativo, na qual a relaxação linear do problema principal, normalmente chamado de *mestre restrito* (*restricted master*), é resolvida seguido de um subproblema de precificação para determinar se há outras variáveis com *custo reduzido negativo*. Caso exista alguma variável com custo reduzido negativo, esta é adicionada ao problema e executa-se a próxima iteração. Caso contrário é ótimo, pois a solução corrente não terá colunas com potencial de reduzir o valor do objetivo corrente entre as que não estão presentes no mestre restrito.

Para calcular o custo reduzido de uma variável x_i de um modelo descrito pelo programa linear de minimização:

$$\begin{aligned} & \min \sum_{i \in I} c_i x_i \\ & \text{sujeito a:} \\ & \sum_{i \in I} a_{li} x_i \leq b_l \quad (\forall l \in L) \\ & x_i \geq 0 \quad (\forall i \in I) \end{aligned} \tag{4-1}$$

Este possui a versão dual:

$$\begin{aligned} & \max \sum_{l \in L} b_l \gamma_l \\ & \text{sujeito a:} \sum_{l \in L} a_{li} \gamma_l \leq c_i \quad (\forall i \in I) \\ & \gamma_l \leq 0 \quad (\forall l \in L) \end{aligned} \tag{4-2}$$

Se o problema for viável, então as soluções ótimas de ambos são iguais, ou seja existem x_i^* com $i \in I$ e γ_l^* com $l \in L$ tal que $\sum_{i \in I} c_i x_i^* = \sum_{l \in L} b_l \gamma_l^*$.

O custo reduzido \bar{c}_i da variável x_i pode ser calculado como:

$$\bar{c}_i = c_i - \sum_{l \in L} a_{li} \gamma_l \tag{4-3}$$

Como o método de geração de colunas busca novas variáveis com custo reduzido negativo que deveriam pertencer ao escopo do problema, o problema de precificação, dado as variáveis duais do problema principal, resolve um modelo cuja solução é uma nova variável x_i , seu custo c_i e os coeficientes a_{li} com custo reduzido mínimo.

No cenário do desafio, o problema principal é resolver o sistema que atenda às restrições da especificação. O problema auxiliar será reduzido para identificar configurações de processos para cada $m \in \mathcal{M}$, sem ter muita interação com as outras máquinas, sendo assim, o problema de precificação pode ser interpretado como: dado os valores das variáveis duais do problema principal, quais processos a máquina m pode alocar, sem exaurir a capacidade, e minimizando o custo reduzido. E como cada máquina possui características distintas, então haverá um problema de precificação para cada máquina m . Por simplicidade, nos problemas auxiliares serão omitidos os índices m , n e l das constantes.

De forma análoga ao capítulo anterior, serão mostrados as variáveis utilizadas, as restrições, o cálculo do objetivo e do custo reduzido antes da

formulação completa do problema principal e do auxiliar de precificação.

4.1

Variáveis de Decisão do Problema Principal

Seja Λ_m o conjunto de configurações possíveis para a máquina $m \in \mathcal{M}$, então define-se a variável λ_{ma} como:

$$\lambda_{ma} = \begin{cases} 1 & \text{se a } a\text{-ésima configuração da máquina } m \in \mathcal{M} \\ & \text{foi escolhida} \\ 0 & \text{caso contrário} \end{cases} \quad (4-4)$$

Define-se também constantes auxiliares q_{ma}^p , v_{ma}^s e z_{ma}^s como:

$$q_{ma}^p = \begin{cases} 1 & \text{se o processo } p \text{ está na configuração } a \text{ da máquina } m. \\ 0 & \text{caso contrário.} \end{cases} \quad (4-5)$$

$$v_{ma}^s = \begin{cases} 1 & \text{se algum processo } p \text{ do serviço } s \text{ está na configuração } a \\ & \text{da máquina } m. \\ 0 & \text{caso contrário.} \end{cases} \quad (4-6)$$

$$z_{ma}^s = \begin{cases} 1 & \text{se o serviço } s \text{ teve algum processo } p \in s \text{ que estava na} \\ & \text{configuração inicial da máquina } m \text{ e foi movido para} \\ & \text{outra máquina.} \\ 0 & \text{caso contrário.} \end{cases} \quad (4-7)$$

Os conjuntos Λ_m possuem um elemento que é o estado inicial da instância que, por convenção, será a primeira configuração a de cada conjunto Λ_m . Portanto, o problema principal sempre terá pelo menos uma solução inteira válida, que é a solução trivial.

Pela definição de conflito na Seção 2.2.2, as constantes q_{ma}^p e v_{ma}^s possuem a seguinte relação:

$$v_{ma}^s = \sum_{p \in s} q_{ma}^p \quad (\forall s \in \mathcal{S}, \forall a \in \Lambda_m, \forall m \in \mathcal{M}) \quad (4-8)$$

Como dois processos de um mesmo serviço não podem ser alocados na mesma máquina, então na mesma configuração a da máquina m a soma $\sum_{p \in s} q_{ma}^p$ não pode ser maior que 1. Pelo mesmo motivo, z_{ma}^s também não pode ser maior que 1, pois dois processos do mesmo serviço não podem estar presentes ao mesmo tempo, portanto mais de um processo de um mesmo serviço não pode sair da mesma máquina.

Vale mencionar que é possível que ocorra uma permuta entre dois processos do mesmo serviço, ou seja, o processo p_1 que estava inicialmente alocado saia da máquina e outro processo p_2 é alocado. Neste cenário, as constantes relacionadas aos serviços serão 1.

Para cada configuração a da máquina m , existe um valor Δ_{ma} referente às métricas de custo definidas no desafio que são diretamente calculáveis pela configuração.

4.2

Variáveis Auxiliares do Problema Principal

As variáveis de decisão têm seu contexto no que uma máquina possui. Porém, o desafio propõe que existam relações entre grupos de máquinas. Assim como na modelagem por MIP, definem-se as variáveis h_{sn} e o_{sl} , com o mesmo sentido do MIP, se o serviço $s \in \mathcal{S}$ está presente em uma vizinhança $n \in \mathcal{N}$ ou localidade $l \in \mathcal{L}$ respectivamente.

A variável h_{sn} se relaciona com λ_{ma} por meio de:

$$h_{sn} \leq \sum_{\substack{a \in \Lambda_m \\ m \in n}} v_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}, \forall n \in \mathcal{N}) \quad (4-9)$$

$$h_{sn} \geq \sum_{a \in \Lambda_m} v_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}, \forall m \in n \in \mathcal{N}) \quad (4-10)$$

As Equações (4-9), na relaxação linear, produzem as variáveis duais η_{sn}^u com $s \in \mathcal{S}$ e $n \in \mathcal{N}$. As Equações (4-10), na relaxação linear, produzem as

variáveis duais η_{sm}^l com $s \in \mathcal{S}$ e $m \in \mathcal{M}$.

Observa-se que para a restrição de limite superior resultam para cada serviço em N variáveis duais e para o limite inferior em M duais. Para o limite inferior, há a comparação de cada vizinhança n com cada um das máquinas integrantes, ou seja, para cada serviço há uma quantidade de limites inferior iguais à cardinalidade da vizinhança. E como cada máquina m pertence somente a uma única vizinhança n , então a soma de todas as cardinalidades das vizinhanças é igual ao conjunto de todas as máquinas. Por isso, para cada serviço existem N variáveis duais de limite superior e M variáveis de limite inferior.

A variável o_{sl} relaciona-se com λ_{ma} por meio de:

$$o_{sl} \leq \sum_{\substack{a \in \Lambda_m \\ m \in l}} v_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}, \forall l \in \mathcal{L}) \quad (4-11)$$

$$o_{sl} \geq \sum_{a \in \Lambda_m} v_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}, \forall m \in l \in \mathcal{L}) \quad (4-12)$$

Estas relações, na relaxação linear, produzem as variáveis duais v_{sl}^u (com $s \in \mathcal{S}$ e $l \in \mathcal{L}$) e v_{sm}^l (com $s \in \mathcal{S}$ e $m \in \mathcal{M}$) respectivamente.

Como cada máquina m está localizada em uma única localidade l , por motivos análogos existem \mathcal{L} variáveis duais de limite superior e \mathcal{M} variáveis de limite inferior para cada serviço.

Também são definidas variáveis auxiliares g_s com $s \in \mathcal{S}$ tais que:

$$g_s = \sum_{\substack{a \in \Lambda_m \\ m \in \mathcal{M}}} z_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}) \quad (4-13)$$

4.3

Restrições do Problema Principal

Como o problema foi decomposto em um principal e outro auxiliar, algumas restrições do desafio são implementadas no problema principal e

outras no problema auxiliar.

Alocação de máquina

Esta é uma restrição implícita, pois está na definição desta metodologia adotada. Como cada máquina $m \in \mathcal{M}$ deve possuir uma alocação válida do conjunto Λ_m , portanto:

$$\sum_{a \in \Lambda_m} \lambda_{ma} = 1 \quad (\forall m \in \mathcal{M}) \quad (4-14)$$

Estas restrições, na relaxação linear, produzem as variáveis duais μ_m com $m \in \mathcal{M}$.

Alocação de Processos

Assim como no MIP, esta é uma restrição implícita, onde todos os processos devem ser alocados, então:

$$\sum_{\substack{a \in \Lambda_m \\ m \in \mathcal{M}}} q_{ma}^p \lambda_{ma} = 1 \quad (\forall p \in \mathcal{P}) \quad (4-15)$$

Estas restrições garantem que cada processo somente pode ser alocado uma única vez entre todas as configurações de todas as máquinas e na relaxação linear, produzem as variáveis duais π_p com $p \in \mathcal{P}$.

Capacidade da máquina

No problema principal, esta restrição é provida pela própria definição de configuração de alocação. Logo, por definição, $a \in \Lambda_m$ é uma configuração

de alocação de processos válida, portanto não pode ter utilização acima da capacidade da máquina.

Esta restrição está presente no problema auxiliar de precificação.

Conflito

No problema principal, esta restrição é provida pela própria definição de configuração de alocação. Logo, por definição, $a \in \Lambda_m$ é uma configuração de alocação de processos válida, portanto não pode ter mais de um processo do mesmo serviço alocados.

Esta restrição está presente no problema auxiliar de precificação.

Espalhamento dos Serviços

De forma semelhante à formulação MIP, a restrição de dependência de serviços é definida por:

$$\sum_{l \in \mathcal{L}} o_{sl} \geq \delta_s \quad (\forall s \in \mathcal{S}) \quad (4-16)$$

Dependência de Serviços

Por sua vez, a restrição de dependência de serviços é definida por:

$$h_{sn} \leq h_{s^d n} \quad (\forall s, s^d \in \mathcal{S} \gamma_{ss^d} = 1, \forall n \in \mathcal{N}) \quad (4-17)$$

Utilização de recursos transitória

No problema principal, esta restrição é provida pela própria definição de configuração de alocação. Portanto, por definição, $a \in \Lambda_m$ é uma configuração de alocação de processos válida, portanto não pode ter utilização acima da capacidade da máquina.

Esta restrição está presente no problema auxiliar de precificação.

4.4

Objetivo do Problema Principal

Ao se adotarem configurações de processos predefinidas, é possível calcular previamente a parcela no custo dessas configurações. Com isso, vários componentes de custos presentes no desafio estão agrupados.

Custo de Carga

Esta métrica relativa a uma configuração a da máquina m está contida no coeficiente Δ_{ma} , que é calculada no problema auxiliar de precificação.

Custo de Balanceamento

Refere-se ao balanceamento da configuração a da máquina m que está contida no coeficiente Δ_{ma} . Esta parcela é calculada no problema auxiliar de precificação.

Custo de Movimentação de Processos

Esta métrica está contida no coeficiente Δ_{ma} , relativo a uma configuração a da máquina m , calculada pelo problema auxiliar de precificação.

Custo de Movimentação de Serviço

Esta parcela somente pode ser calculada no problema principal, onde o cálculo do smc é definido por:

$$smc \geq g_s \quad (\forall s \in \mathcal{S}) \quad (4-18)$$

Custo de Movimentação de Máquinas

O custo de movimentação para a máquina m na configuração a está presente no coeficiente Δ_{ma} . Este custo é calculado no problema auxiliar de precificação.

Custo Total

O custo total no problema principal, que também será igual ao custo da solução no desafio, é definido por:

$$\begin{aligned} \min \quad & \sum_{\substack{a \in \Lambda_m \\ m \in \mathcal{M}}} \Delta_{ma} \lambda_{ma} \\ & + W^{smc} smc \end{aligned} \quad (4-19)$$

4.5

Custo reduzido das Variáveis

Para cada variável λ_{ma} , com $a \in \Lambda_m$ e $m \in \mathcal{M}$, o custo reduzido Δ_{ma}^- é definido por:

$$\begin{aligned}
 \Delta_{ma}^- &= \Delta_{ma} - \mu_m - \sum_{p \in \mathcal{P}} \pi_p q_{ma}^p \\
 &\quad - \sum_{s \in \mathcal{S}} \eta_{sn}^u v_{ma}^s - \sum_{s \in \mathcal{S}} \eta_{sm}^l v_{ma}^s \quad (\forall a \in \Lambda_m, \forall m \in \mathcal{M}) \\
 &\quad - \sum_{s \in \mathcal{S}} v_{sl}^u v_{ma}^s - \sum_{s \in \mathcal{S}} v_{sm}^l v_{ma}^s \quad (n \in \mathcal{N} | m \in n) \quad (4-20) \\
 &\quad - \sum_{s \in \mathcal{S}} \gamma_s z_{ma}^s \quad (l \in \mathcal{L} | m \in l)
 \end{aligned}$$

Quando $\Delta_{ma}^- < 0$ então a variável λ_{ma} pode entrar na base do simplex.

4.6

Formulação do Problema Principal

Com base nas informações previamente expostas, segue a formulação do problema principal:

$$\begin{aligned}
 \min \quad & \sum_{\substack{a \in \Lambda_m \\ m \in \mathcal{M}}} \Delta_{ma} \lambda_{ma} \\
 & + W^{smc} smc
 \end{aligned} \quad (4-19)$$

sujeito a:

$$\sum_{a \in \Lambda_m} \lambda_{ma} = 1 \quad (\forall m \in \mathcal{M}) \quad (4-14)$$

$$\sum_{\substack{a \in \Lambda_m \\ m \in \mathcal{M}}} q_{ma}^p \lambda_{ma} = 1 \quad (\forall p \in \mathcal{P}) \quad (4-15)$$

$$h_{sn} \leq \sum_{\substack{a \in \Lambda_m \\ m \in n}} v_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}, \forall n \in \mathcal{N}) \quad (4-9)$$

$$h_{sn} \geq \sum_{a \in \Lambda_m} v_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}, \forall m \in n \in \mathcal{N}) \quad (4-10)$$

$$h_{sn} \leq h_{s^d n} \quad (\forall s, s^d \in \mathcal{S} | \gamma_{ss^d} = 1, \forall n \in \mathcal{N}) \quad (4-17)$$

$$o_{sl} \leq \sum_{\substack{a \in \Lambda_m \\ m \in l}} v_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}, \forall l \in \mathcal{L}) \quad (4-11)$$

$$o_{sl} \geq \sum_{a \in \Lambda_m} v_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}, \forall m \in l \in \mathcal{L}) \quad (4-12)$$

$$\sum_{l \in \mathcal{L}} o_{sl} \geq \delta_s \quad (\forall s \in \mathcal{S}) \quad (4-16)$$

$$g_s = \sum_{\substack{a \in \Lambda_m \\ m \in \mathcal{M}}} z_{ma}^s \lambda_{ma} \quad (\forall s \in \mathcal{S}) \quad (4-13)$$

$$smc \geq g_s \quad (\forall s \in \mathcal{S}) \quad (4-18)$$

$$\lambda_{ma} \in \{0, 1\} \quad (\forall a \in \Lambda_m, \forall m \in \mathcal{M})$$

$$h_{sn} \in \{0, 1\} \quad (\forall s \in \mathcal{S}, \forall n \in \mathcal{N})$$

$$o_{sl} \in \{0, 1\} \quad (\forall s \in \mathcal{S}, \forall l \in \mathcal{L})$$

$$g_s \geq 0 \quad (\forall s \in \mathcal{S})$$

$$smc \geq 0$$

4.7

Variáveis de Decisão do Problema de Precificação

Para cada máquina $m \in \mathcal{M}$, sejam as variáveis x_p com $p \in \mathcal{P}$ definidas por:

$$x_p = \begin{cases} 1 & \text{se o processo } p \text{ está alocado;} \\ 0 & \text{caso contrário.} \end{cases} \quad (4-21)$$

De forma análoga ao MIP, existem constantes x_p^0 tais que:

$$x_p^0 = \begin{cases} 1 & \text{se o processo } p \text{ está alocação inicial;} \\ 0 & \text{caso contrário.} \end{cases} \quad (4-22)$$

4.8

Variáveis Auxiliares do Problema de Precificação

São definidas variáveis que permitem ter conhecimento do estado da máquina m , como a utilização e mudança de processos.

Variáveis de migração

Define-se a variável z_p como:

$$z_p = \begin{cases} 1 & \text{se o processo } p \text{ estava inicialmente alocado} \\ & \text{e saiu da máquina} \\ 0 & \text{caso contrário.} \end{cases} \quad (4-23)$$

Estas se relacionam com x_p por meio de:

$$z_p \geq x_p^0 - x_p \quad (\forall p \in \mathcal{P}) \quad (4-24)$$

Variáveis de Utilização

Define-se a variável u_r como a utilização do recurso $r \in \mathcal{R}$. Formalmente:

$$u_r = \sum_{p \in \mathcal{P}} R_{pr} x_p \quad (\forall r \in \mathcal{R}) \quad (4-25)$$

Variáveis de Utilização Transiente

Define-se a variável u_r^T como a utilização transiente do recurso $r \in \mathcal{T}$. Formalmente:

$$u_r^T = \sum_{p \in \mathcal{P}} R_{pr} z_p \quad (\forall r \in \mathcal{T}) \quad (4-26)$$

Pode-se assumir que $u_r^T = 0$ quando $r \notin \mathcal{T}$

Variáveis de Sobrecarga

De forma análoga ao MIP, com base em u_r , definem-se as variáveis d_r como a quantidade de utilização de recurso r acima da capacidade segura SC_r . Matematicamente:

$$d_r \geq u_r - SC_r \quad (\forall r \in \mathcal{R}) \quad (4-27)$$

Variáveis de Disponibilidade de Recursos

Também define-se a disponibilidade a_r do recurso $s \in \mathcal{R}$ como:

$$a_r = C_r - u_r \quad (\forall r \in \mathcal{R}) \quad (4-28)$$

Variáveis de Balanceamento

Com base nas variáveis de disponibilidade a_r , podem-se definir variáveis $b_{r_1 r_2}$ para mensurar o balanceamento entre os recursos $r_1, r_2 \in \mathcal{R}$ por meio de:

$$b_{r_1 r_2} \geq B_{r_1 r_2} a_{r_1} - a_{r_2} \quad (\forall r_1, r_2 \in \mathcal{R}) \quad (4-29)$$

Variáveis de Presença de Serviço

Definem-se as variáveis h_s com $s \in \mathcal{S}$ como:

$$h_s = \begin{cases} 1 & \text{se algum processo } p \text{ do serviço } s \\ & \text{está presente na máquina} \\ 0 & \text{caso contrário.} \end{cases} \quad (4-30)$$

Também definem-se as variáveis g_s como:

$$g_s = \begin{cases} 1 & \text{se algum processo } p \text{ do serviço } s \\ & \text{inicialmente alocado saiu da máquina} \\ 0 & \text{caso contrário.} \end{cases} \quad (4-31)$$

Estas se relacionam com x_p e z_p por meio de:

$$h_s = \sum_{p \in s} x_p \quad (\forall s \in \mathcal{S}) \quad (4-32)$$

$$g_s = \sum_{p \in s} z_p \quad (\forall s \in \mathcal{S}) \quad (4-33)$$

4.9

Restrições do Problema de Precificação

Como mencionado anteriormente, algumas restrições do desafio estão implementadas no problema principal e outras serão implementadas nos problemas auxiliares.

Capacidade da máquina

Esta restrição define um limite superior para a utilização de recursos $r \in \mathcal{R}$ da máquina m :

$$u_r \leq C_r \quad (\forall r \in \mathcal{R}) \quad (4-34)$$

Conflito

A restrição de conflito é implementada por meio de:

$$\sum_{p \in s} x_p \leq 1 \quad (\forall s \in \mathcal{S}) \quad (4-35)$$

Espalhamento dos serviços

Esta restrição está implementada no problema principal, pois os problemas auxiliares só lidam com aspectos de uma única máquina por vez.

Dependência dos serviços

Esta restrição está implementada no problema principal, pois os problemas auxiliares só lidam com aspectos de uma única máquina por vez.

Utilização de recursos transitória

Esta restrição define um limite superior para a utilização de recursos $r \in \mathcal{T}$ da máquina m :

$$u_r + u_r^T \leq C_r \quad (\forall r \in \mathcal{T}) \quad (4-36)$$

Esta restrição domina a restrição de capacidade.

4.10

Objetivo do Problema de Precificação

O objetivo dos problemas de precificação é identificar uma configuração de processos para a máquina, que deveria estar em Λ_m e possui custo reduzido negativo. O valor Δ é o coeficiente de custo no objetivo do problema principal.

Custo de Carga

O custo de carga do recurso $r \in \mathcal{R}$ é dado pelas variáveis d_r .

Custo de Balanceamento

O custo de balanceamento entre os recursos $r_1, r_2 \in \mathcal{R}$ é dado pelas variáveis $b_{r_1 r_2}$.

Custo de Movimentação de Processos

O custo de movimentação de processos pmc é dado por:

$$pmc = \sum_{\substack{p \in \mathcal{P} \\ x_p^0 = 0}} PMC_p x_p \quad (4-37)$$

Custo de Movimentação de Serviço

No problema auxiliar, esta parcela não é computada. Somente é considerada no problema principal.

Custo de Movimentação de Máquinas

O custo de movimentação de processos mmc é dado por:

$$mmc = \sum_{\substack{p \in \mathcal{P} \\ x_p^0 = 0}} MMC_p x_p \quad (4-38)$$

Onde MMC_p são constantes por problema auxiliar, calculadas por meio de :

$$MMC_p = MMC(m_p^0, m)$$

Sendo m_p^0 a máquina na qual o processo p está inicialmente alocado.

Custo Total

O custo total Δ das parcelas do desafio é dado por:

$$\Delta = \sum_{r \in \mathcal{R}} W_r^{lc} d_r + \sum_{r1, r2 \in \mathcal{R}} W_{r1r2}^{bc} b_{r1r2} + W^{pmc} pmc + W^{mmc} mmc \quad (4-39)$$

O objetivo do problema de precificação é minimizar o custo reduzido definido por:

$$\begin{aligned} \min \quad & \Delta - \mu - \sum_{p \in \mathcal{P}} \pi_p x_p \\ & - \sum_{s \in \mathcal{S}} \eta_s^u h_s - \sum_{s \in \mathcal{S}} \eta_s^l h_s \\ & - \sum_{s \in \mathcal{S}} v_s^u h_s - \sum_{s \in \mathcal{S}} v_s^l h_s \\ & - \sum_{s \in \mathcal{S}} \gamma_s g_s \end{aligned} \quad (4-40)$$

Sendo π_p , μ , η_s^u , η_s^l , v_s^u , v_s^l e γ_s as variáveis duais pertinentes à máquina m do problema de precificação.

A função pode ser simplificada para:

$$\begin{aligned} \min \quad & \Delta - \mu - \sum_{p \in \mathcal{P}} \pi_p x_p \\ & - \sum_{s \in \mathcal{S}} \sigma_s h_s \\ & - \sum_{s \in \mathcal{S}} \gamma_s g_s \end{aligned} \quad (4-41)$$

onde:

$$\sigma_s = \eta_s^u + \eta_s^l + v_s^u + v_s^l \quad (4-42)$$

4.11

Formulação do Problema de Precificação

Com base nas informações previamente expostas, segue a formulação dos problemas auxiliares.

$$\min \quad \Delta - \mu - \sum_{p \in \mathcal{P}} \pi_p x_p - \sum_{s \in \mathcal{S}} \sigma_s h_s - \sum_{s \in \mathcal{S}} \gamma_s g_s \quad (4-41)$$

sujeito a:

$$z_p \geq x_p^0 - x_p \quad (\forall p \in \mathcal{P}) \quad (4-24)$$

$$u_r = \sum_{p \in \mathcal{P}} R_{pr} x_p \quad (\forall r \in \mathcal{R}) \quad (4-25)$$

$$u_r^T = \sum_{p \in \mathcal{P}} R_{pr} z_p \quad (\forall r \in \mathcal{T}) \quad (4-26)$$

$$u_r^T = 0 \quad (\forall r \notin \mathcal{T})$$

$$d_r \geq u_r - SC_r \quad (\forall r \in \mathcal{R}) \quad (4-27)$$

$$a_r = C_r - u_r \quad (\forall r \in \mathcal{R}) \quad (4-28)$$

$$b_{r_1 r_2} \geq B_{r_1 r_2} a_{r_1} - a_{r_2} \quad (\forall r_1, r_2 \in \mathcal{R}) \quad (4-29)$$

$$h_s = \sum_{p \in \mathcal{S}} x_p \quad (\forall s \in \mathcal{S}) \quad (4-32)$$

$$g_s = \sum_{p \in \mathcal{S}} z_p \quad (\forall s \in \mathcal{S}) \quad (4-33)$$

$$\sum_{p \in \mathcal{S}} x_p \leq 1 \quad (\forall s \in \mathcal{S}) \quad (4-35)$$

$$u_r + u_r^T \leq C_r \quad (\forall r \in \mathcal{R}) \quad (4-36)$$

$$pmc = \sum_{\substack{p \in \mathcal{P} \\ x_p^0 = 0}} PMC_p x_p \quad (4-37)$$

$$mmc = \sum_{\substack{p \in \mathcal{P} \\ x_p^0 = 0}} MMC_p x_p \quad (4-38)$$

$$\begin{aligned} \Delta = & \sum_{r \in \mathcal{R}} W_r^{lc} d_r + \sum_{r_1, r_2 \in \mathcal{R}} W_{r_1 r_2}^{bc} b_{r_1 r_2} \\ & + W^{pmc} pmc + W^{mmc} mmc \end{aligned} \quad (4-39)$$

$$u_r, u_r^T \in [0, C_r] \quad (\forall r \in \mathcal{R})$$

$$d_r, a_r \in [0, C_r] \quad (\forall r \in \mathcal{R})$$

$$b_{r_1 r_2} \geq 0 \quad (\forall r_1, r_2 \in \mathcal{R})$$

$$g_s \geq 0 \quad (\forall s \in \mathcal{S})$$

$$pmc \geq 0$$

$$\begin{aligned}
mmc &\geq 0 \\
x_p, z_p &\in \{0, 1\} && (\forall p \in \mathcal{P}) \\
h_s &\in \{0, 1\} && (\forall s \in \mathcal{S})
\end{aligned}$$

Onde:

$$\begin{aligned}
MMC_p &= MMC(m_p^0, m) \\
\sigma_s &= \eta_s^u + \eta_s^l + v_s^u + v_s^l
\end{aligned}$$

4.12

Critério de Parada

Como mencionado anteriormente, a geração de colunas é um método iterativo, ou seja, há um conjunto de operações em laço que ocorrem até que uma condição seja atendida. Quando há somente um único problema de precificação, o critério de parada é quando o custo reduzido da nova variável não for negativo.

Entretanto, para o problema principal descrito, há M problemas auxiliares de precificação e cada um gerando novas configurações para Λ_m . Cada problema produz uma nova variável λ_m , com custo reduzido $\bar{\Delta}_m$ negativo, pois o problema mestre é de minimização. Um primeiro critério de parada adotado neste trabalho é quando a soma de todos os custos reduzidos for não negativo, ou seja a iteração ocorre enquanto:

$$\sum_{m \in \mathcal{M}} \bar{\Delta}_m < 0 \tag{4-43}$$

Também é adotado como critério de parada quando uma iteração não produzir novas configurações em todas as máquinas. Esta característica de repetição de configuração ocorre pois cada máquina tem propriedades distintas e somente um subconjunto de valores de duais é aplicado.

4.13

Programação inteira

Uma vez que a geração de colunas termine, pois o critério de parada foi atingido, o problema principal é convertido para uma versão inteira. As variáveis de decisão λ_{ma} e as auxiliares h_{sn} e o_{sl} passam a assumir somente valores inteiros 0 ou 1, conforme a definição original. Em outras palavras, o problema deixa de ser linear e é necessário resolver de forma discreta.

Este novo programa inteiro é então resolvido, cuja solução é a seleção de configurações de um conjunto predefinido de possibilidades. Todas as restrições anteriormente descritas para o problema principal continuam válidas. Com a mudança do tipo de variável, o resolvidor Gurobi passa a usar algoritmos para programação inteira.

A resolução deste problema inteiro proveniente da geração de colunas não garante que a solução inteira é ótima, portanto trata-se de uma heurística. Contudo, pode-se aplicar procedimentos auxiliares que resultam comprovadamente uma solução ótima, por meio da obtenção de colunas que não foram geradas na resolução da relaxação linear. Estes procedimentos podem ser por meio de enumeração de todas as colunas com potencial de pertencer a solução ótima ou por meio de *branching*.

4.14

Ensaio preliminares de geração de colunas

As seções anteriores contêm todos os elementos necessários para resolver o problema pelo método de geração de colunas. De forma semelhante à realizada na programação inteira, os modelos são analisados em relação à resposta final, tempo de execução e consumo de memória. São mostrados dois modelos, sendo o primeiro uma versão simplificada do problema principal. O segundo modelo apresentado é a versão completa.

Por questões numéricas, o critério de parada do ciclo de geração de colunas é ajustado para:

$$\sum_{m \in \mathcal{M}} \Delta_m^* > -\epsilon \quad (4-44)$$

onde ϵ é um valor positivo suficientemente pequeno para considerar a soma dos custos reduzidos como nula, dentro de uma margem de erro aceitável. Lembrando que o custo reduzido deve ser um valor negativo para entrar na base. Caso o custo seja nulo, então a variável pode estar na base e não há variável com potencial de redução no valor do objetivo.

Como no problema principal final os custos são inteiros de altas ordens de grandeza e as variáveis também são inteiras, então usou-se $\epsilon = 0,1$ em todos os ensaios. Optou-se pelo uso dos mesmos parâmetros em todos os ensaios para todas as instâncias, de forma análoga ao que foi aplicado na avaliação do desafio pelos organizadores.

4.14.1

Modelo simplificado - CG1

Um primeiro modelo, *CG1*, é construído como uma prova de conceito desta metodologia. O programa principal é uma versão simplificada, onde somente possui as restrições que todas as máquinas devem ter alguma alocação selecionada e todos os processos devem estar alocados (restrições das Equações (4-14) e (4-15) respectivamente). Como há uma simplificação do problema principal, o custo reduzido Δ_{ma}^- também foi alterado para:

$$\Delta_{ma}^- = \Delta_{ma} - \mu_m - \sum_{p \in \mathcal{P}} \pi_p q_{ma}^p \quad (4-45)$$

Na instância de exemplo, para cada máquina $m \in \mathcal{M}$ tem como conjunto inicial Λ_m :

$$\begin{aligned} \Lambda_{m_1} &= \{\{p_1, p_3\}\} \\ \Lambda_{m_2} &= \{\{\}\} \\ \Lambda_{m_3} &= \{\{\}\} \\ \Lambda_{m_4} &= \{\{p_2\}\} \end{aligned}$$

Do conjunto inicial obtêm-se os coeficientes q_{ma}^p , relacionando cada variável λ_{ma} com o processo p da configuração inicial $a = 0$.

Para a máquina m_1 tem-se:

$$q_{m_1,0}^{p_1} = 1$$

$$q_{m_1,0}^{p_2} = 0$$

$$q_{m_1,0}^{p_3} = 1$$

Como as máquinas m_2 e m_3 não possuem processos inicialmente atribuídos, então:

$$q_{m_2,0}^{p_1} = q_{m_2,0}^{p_2} = q_{m_2,0}^{p_3} = q_{m_3,0}^{p_1} = q_{m_3,0}^{p_2} = q_{m_3,0}^{p_3} = 0$$

E para a máquina m_4 :

$$q_{m_4,0}^{p_1} = 0$$

$$q_{m_4,0}^{p_2} = 1$$

$$q_{m_4,0}^{p_3} = 0$$

Resultando no sistema linear com objetivo:

$$\min \quad 2000\lambda_{m_1,0} + 0\lambda_{m_2,0} + 2000\lambda_{m_3,0} + 200\lambda_{m_4,0}$$

Vale notar que, apesar da máquina m_3 não possuir processos alocados, esta apresenta um custo resultante da falta de balanceamento entre os recursos.

As restrições de alocação de processo $\sum_{\substack{a \in \Lambda_m \\ m \in \mathcal{M}}} q_{ma}^p \lambda_{ma} = 1$ para cada processo $p \in \mathcal{P}$:

$$\lambda_{m_1,0} = 1$$

$$\lambda_{m_4,0} = 1$$

$$\lambda_{m_1,0} = 1$$

E nas restrições de máquina $\sum_{a \in \Lambda_m} \lambda_{ma} = 1$ para cada máquina $m \in \mathcal{M}$:

$$\lambda_{m_1,0} = 1$$

$$\lambda_{m_2,0} = 1$$

$$\lambda_{m_3,0} = 1$$

$$\lambda_{m_4,0} = 1$$

Cuja solução pode ser facilmente verificável $\lambda_{m_1,0} = \lambda_{m_2,0} = \lambda_{m_3,0} = \lambda_{m_4,0} = 1$. As variáveis duais $\pi_{p_1} = 2000$, $\pi_{p_2} = 200$, $\pi_{p_3} = 0$, $\mu_{m_1} = 0$, $\mu_{m_2} = 0$, $\mu_{m_3} = 2000$ e $\mu_{m_4} = 0$.

Ao aplicar estes valores nos problemas auxiliares de precificação, para a máquina m_1 , resulta que esta deve conter o processo p_1 somente, cujo custo desta configuração é nulo. Para a máquina m_2 , o problema de precificação retorna a configuração já existente, possuir nenhum processo. Para a máquina m_3 , é retornada que esta deve conter o processo p_1 , e possui custo 1400. Na máquina m_4 , é retornada a configuração corrente. O custo reduzido total desta iteração é -4600 . O problema mestre passa a ser:

$$\min \quad 2000\lambda_{m_1,0} + 2000\lambda_{m_3,0} + 200\lambda_{m_4,0} + 1400\lambda_{m_3,1}$$

$$\lambda_{m_1,0} + \lambda_{m_1,1} + \lambda_{m_3,1} = 1$$

$$\lambda_{m_4,0} = 1$$

$$\lambda_{m_1,0} = 1$$

$$\lambda_{m_1,0} + \lambda_{m_1,1} = 1$$

$$\lambda_{m_2,0} = 1$$

$$\lambda_{m_3,0} + \lambda_{m_3,1} = 1$$

$$\lambda_{m_4,0} = 1$$

Cuja a solução também é a configuração inicial e resulta nas duais $\pi_{p_1} = -600$, $\pi_{p_2} = 200$, $\pi_{p_3} = 2600$, $\mu_{m_1} = 0$, $\mu_{m_2} = 0$, $\mu_{m_3} = 2000$ e $\mu_{m_4} = 0$. Ao aplicar estes novos valores nos problemas auxiliares, resulta na configurações iniciais para m_1 e m_4 . E nas novas configurações $\{p_3\}$ e $\{p_2\}$ para m_2 e m_3 com custos 901 e 500 respectivamente. O custo reduzido total da iteração é -3399 . O problema mestre passa a ser:

$$\begin{aligned} \min \quad & 2000\lambda_{m_1,0} + 2000\lambda_{m_3,0} + 200\lambda_{m_4,0} + 1400\lambda_{m_3,1} + 901\lambda_{m_2,1} + 500\lambda_{m_3,2} \\ & \lambda_{m_1,0} + \lambda_{m_1,1} + \lambda_{m_3,1} = 1 \\ & \lambda_{m_4,0} + \lambda_{m_3,2} = 1 \\ & \lambda_{m_1,0} + \lambda_{m_2,1} = 1 \\ & \lambda_{m_1,0} + \lambda_{m_1,1} = 1 \\ & \lambda_{m_2,0} + \lambda_{m_2,1} = 1 \\ & \lambda_{m_3,0} + \lambda_{m_3,1} + \lambda_{m_3,2} = 1 \\ & \lambda_{m_4,0} = 1 \end{aligned}$$

Este novo problema tem como solução $\lambda_{m_1,1} = \lambda_{m_2,1} = \lambda_{m_3,0} = \lambda_{m_4,0} = 1$, com valor de 3101. As duais resultam em $\pi_{p_1} = -600$, $\pi_{p_2} = -1500$, $\pi_{p_3} = 2000$, $\mu_{m_1} = 600$, $\mu_{m_2} = 0$, $\mu_{m_3} = 2000$ e $\mu_{m_4} = 1700$. Resultam somente em uma nova configuração para m_4 , sem processos e custo 1000. A iteração termina com um custo reduzido total de -1799 . Segue o problema mestre resultante:

$$\begin{aligned} \min \quad & 2000\lambda_{m_1,0} + 2000\lambda_{m_3,0} + 200\lambda_{m_4,0} \\ & + 1400\lambda_{m_3,1} + 901\lambda_{m_2,1} + 500\lambda_{m_3,2} + 1000\lambda_{m_4,1} \\ & \lambda_{m_1,0} + \lambda_{m_1,1} + \lambda_{m_3,1} = 1 \\ & \lambda_{m_4,0} + \lambda_{m_3,2} = 1 \\ & \lambda_{m_1,0} + \lambda_{m_2,1} = 1 \\ & \lambda_{m_1,0} + \lambda_{m_1,1} = 1 \\ & \lambda_{m_2,0} + \lambda_{m_2,1} = 1 \\ & \lambda_{m_3,0} + \lambda_{m_3,1} + \lambda_{m_3,2} = 1 \\ & \lambda_{m_4,0} + \lambda_{m_4,1} = 1 \end{aligned}$$

Este tem como solução $\lambda_{m_1,1} = \lambda_{m_2,1} = \lambda_{m_3,2} = \lambda_{m_4,1} = 1$, com valor de

2401. As duais resultam em $\pi_{p_1} = -600$, $\pi_{p_2} = -1500$, $\pi_{p_3} = 2000$, $\mu_{m_1} = 600$, $\mu_{m_2} = 0$, $\mu_{m_3} = 2000$ e $\mu_{m_4} = 1700$. Os sub problemas de geração de colunas não retornam mais colunas novas, terminando a fase de geração de coluna e o custo reduzido total da iteração é -1799 . As variáveis são convertidas para inteiras, resultando na resposta $\lambda_{m_1,1} = \lambda_{m_2,1} = \lambda_{m_3,2} = \lambda_{m_4,1} = 1$, que é a configuração ótima conhecida. Vale comentar que este problema resultou em um valor de objetivo menor, pois não está presente o custo de movimentação de serviço.

A Tabela 4.1 mostra os resultados da geração de colunas desta versão simplificada nas instâncias **a1**. A coluna *Objetivo* contém o custo final validado pelo programa **checker** fornecido pelos proponentes e onde há o valor *ST* significa que a solução fornecida foi a *Solução Trivial* (presente na Tabela 2.4). A *Solução Trivial* é a solução na qual não há movimentação de processos, ou seja, está na origem que é a configuração inicial. A coluna *Tempo* informa o tempo total, em segundos que a execução utilizou. Este tempo é desde a invocação do programa até o término. A coluna *iterações* contém o número de iterações na etapa de geração de colunas. A coluna *Melhorias* informa quantas vezes o custo reduzido total desta iteração foi melhor que o melhor até então estabelecido. A coluna *MCRT* informa o último melhor custo reduzido total. A coluna *Motivo* é o motivo de término do ensaio, tendo um destes valores:

Concluído Informa que a execução foi concluída. A geração de colunas foi interrompida pois o custo reduzido total atingiu o critério de parada da Equação (4-44).

TCG A execução foi interrompida por limite de tempo durante a etapa de geração de colunas;

TMIP A execução foi interrompida por limite de tempo durante o estágio de resolução do MIP resultante

CGRep A execução foi concluída, porém o ciclo de geração de colunas foi interrompido pois todas as colunas geradas na última iteração eram repetidas.

Inf O resolvidor identificou que o modelo era inviável.

Unb O resolvidor identificou que o modelo era ilimitado.

Como mencionado nos ensaios para o MIP, quando uma execução termina como *concluída*, a resposta é considerada como ótima quando a diferença entre

os limites inferior e superior do objetivo, está na tolerância de 1.0×10^{-4} em relação ao valor absoluto do limite superior.

São utilizados os parâmetros `NumericFocus=2` e `Quad=1`, de forma a instruir o resolvidor para lidar com possíveis problemas numéricos no mestre reduzido. Nos problemas auxiliares são utilizados os parâmetros `Quad=1`, `NumericFocus=3` e `MIPGap=0,01`.

Com base nos dados da Tabela 4.1, somente a instância `a1_1` concluiu passando por todas as etapas. Para as demais instâncias, a execução é interrompida por limite de tempo durante a etapa de geração de colunas.

Tabela 4.1: Resultado parcial por geração de colunas, CG1

Instância	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
<code>a1_1</code>	44.306.501	523,820	Concluído	2071	100	0.00
<code>a1_2</code>	ST	21600,006	TCG	258	41	-614.226.567,775
<code>a1_3</code>	ST	21600,012	TCG	235	56	-53.550.603,682
<code>a1_4</code>	ST	21600,013	TCG	185	21	-5.320.982.128,463
<code>a1_5</code>	ST	21600,033	TCG	307	10	-5.332.429.863,654

4.14.2

CG1 - Instância `a1_1`

O gráfico da Figura 4.1 contém a evolução do valor do objetivo a cada iteração. Contém também a densidade do problema nas iterações, onde pode-se notar que a instância em questão é densa, pois mais de 20% dos coeficientes são não-nulos.

Vale notar que a densidade inicial é de 25%, pois existem P restrições de processos e M restrições de máquina, portanto $P + M$ restrições. Inicialmente, o problema possui M variáveis, que resultam num total de $(P + M)M$ coeficientes, dos quais os P processos estão alocados nas M máquinas, portanto existem $(P + M)$ coeficientes com valor 1. Resultando em uma densidade inicial de $\frac{P + M}{(P + M)M} = \frac{1}{M}$. Como $M = 4$ para a instância em questão, logo a densidade é de $\frac{1}{4} = 0.25$.

Em relação ao tempo utilizado por iteração, utiliza-se menos de 1 segundo, 0,25 segundos. O gráfico da Figura 4.2 detalha o tempo que cada problema, mestre e auxiliares, utilizaram, assim como o tempo total da iteração. Pode-se notar que o problema mestre é o problema mais demorado, se comparado aos auxiliares da mesma iteração. Também pode-se notar que

os tempos utilizados pelo resolvidor MIP (Gurobi) nos problemas são os principais componentes de tempo utilizado. O mesmo gráfico contém também a quantidade de iterações necessárias pelo resolvidor para obter uma solução ótima do problema mestre.

O gráfico da Figura 4.3 contém a evolução do valor absoluto do custo reduzido total (CRT), que é a soma de todos os custos reduzidos gerados pelos problemas auxiliares de precificação, assim como o melhor custo reduzido total (MCRT), ou seja o melhor valor de CRT das iterações anteriores. Por melhor, entende-se que é o maior valor relativo, ou o menor em valor absoluto. O mesmo gráfico contém também a quantidade acumulada de melhorias, ou seja, a quantidade acumulada de vezes que o CRT resultante da iteração foi melhor que o MCRT. Nota-se que nas 1000 iterações iniciais há a maioria das melhorias, mais de 80%, assim como o MCRT reduz da ordem de grandeza inicial de 10^8 para 10^2 .

4.14.3

CG1 - Instância a1_2

Em relação à instância a1_2, a evolução do objetivo é apresentada no gráfico da Figura 4.4. Nota-se que somente nas iterações finais há redução do valor do objetivo. O mesmo gráfico também expõe a densidade do problema a cada iteração, onde a densidade é superior ao dobro da densidade inicial de $\frac{1}{M} = \frac{1}{100} = 0,01$.

Em relação ao tempo por iteração, as iterações resolvem em uma média de 83,72 segundos. O gráfico da Figura 4.5 contém o tempo por iteração, onde notoriamente o tempo do problema mestre foi muito superior ao tempo dos problemas auxiliares. O mesmo gráfico também contém a quantidade de iterações, onde observa-se que a quantidade de iterações é proporcional com o tempo gasto.

O gráfico 4.6 contém o custo reduzido total por iteração, o melhor custo reduzido total e a quantidade acumulada de melhorias. Nota-se que nas iterações finais há metade das melhorias e a redução de uma ordem de grandeza do valor de CRT.

Ao correlacionar os gráficos, observa-se que próximo do início da redução do objetivo, há um maior tempo de resolução dos problemas e também

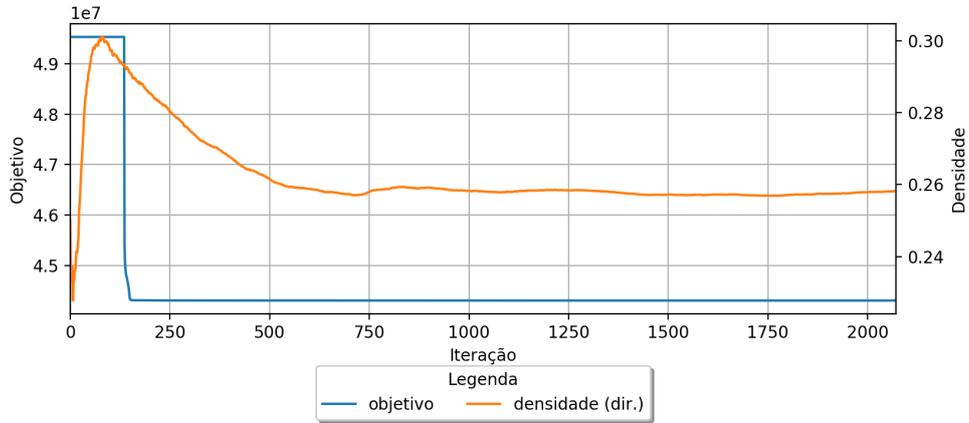


Figura 4.1: Objetivo e densidade de CG1, a1_1

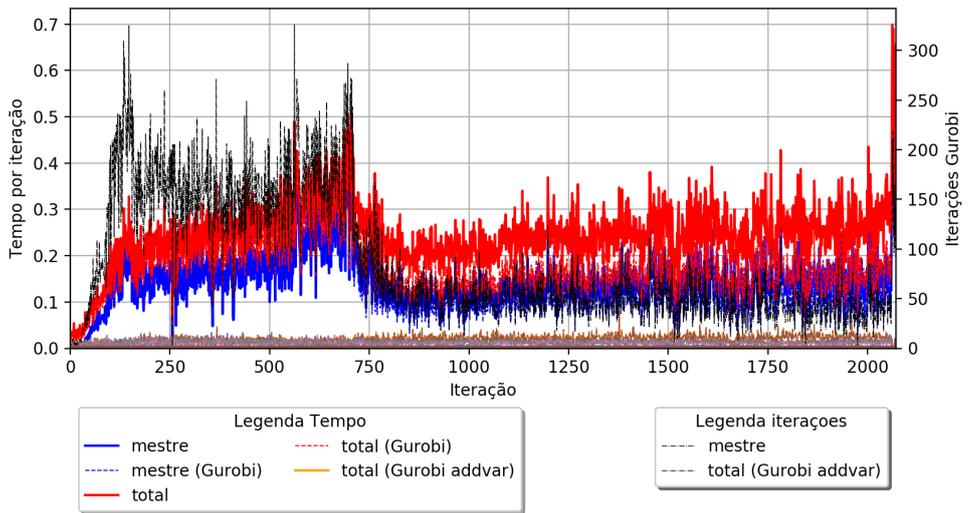


Figura 4.2: Tempos e iterações de CG1, a1_1

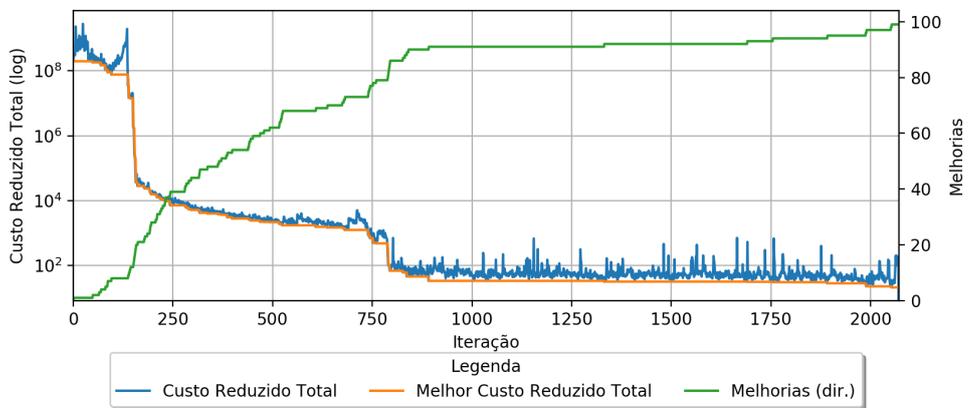


Figura 4.3: Custo reduzido de CG1, a1_1

ocorreu o pior CRT (maior valor absoluto). Após o início da redução, o tempo de resolução reduz quase à metade, assim como o CRT também decresce rapidamente.

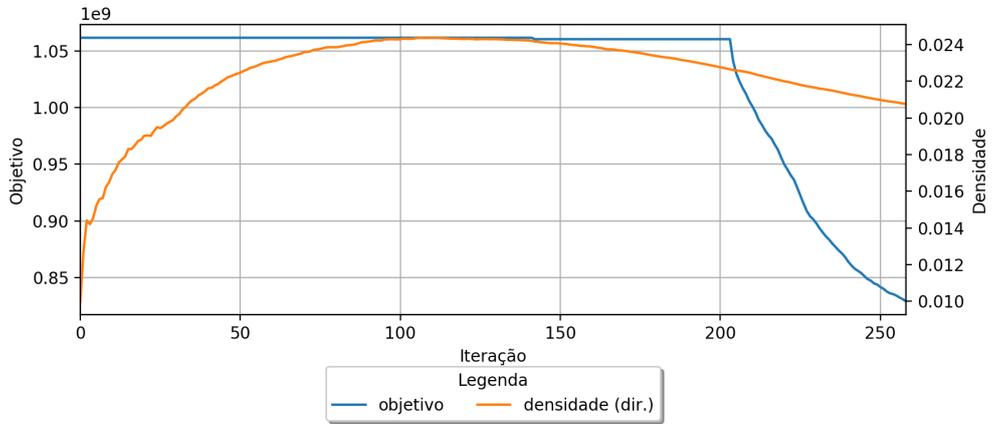


Figura 4.4: Objetivo e densidade de CG1, a1_2

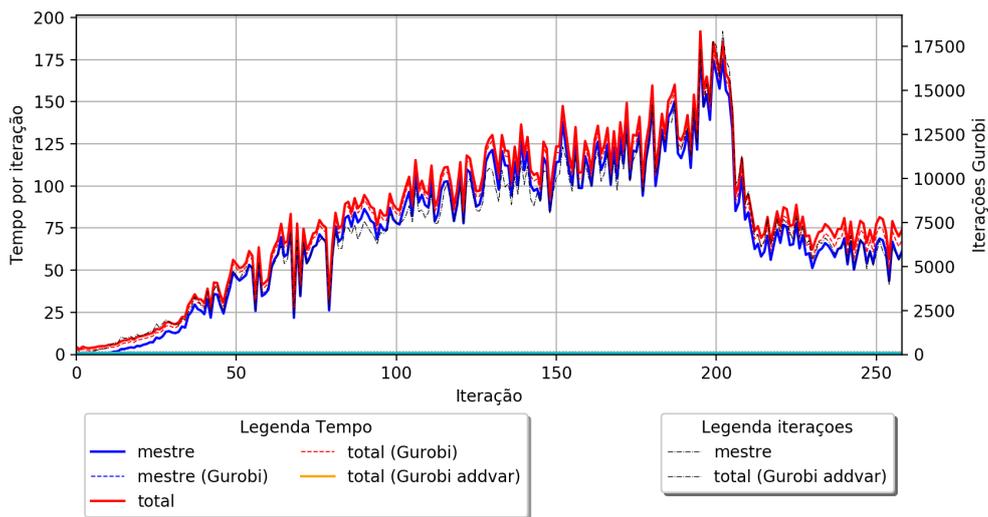


Figura 4.5: Tempos e iterações de CG1, a1_2

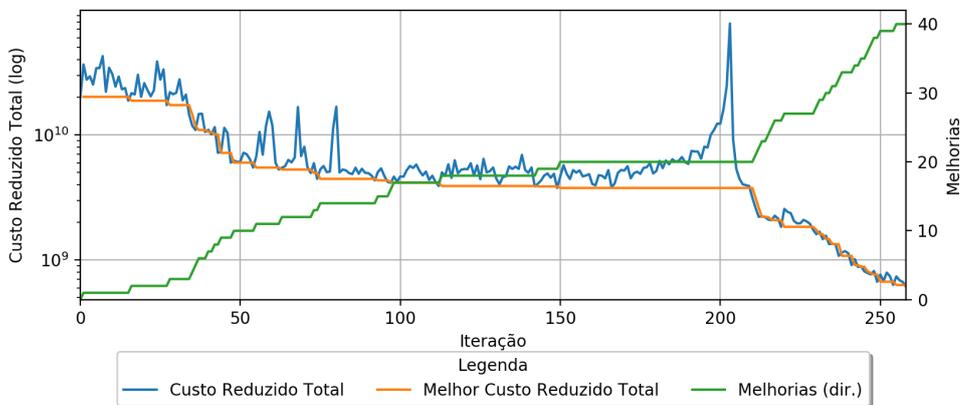


Figura 4.6: Custo reduzido de CG1, a1_2

4.14.4

CG1 - Instância a1_3

Para a instância a1_3, a evolução do objetivo é exposta no gráfico da Figura 4.7. Nota-se que não há alteração no valor do objetivo. O mesmo gráfico também contém a densidade do problema a cada iteração, onde a densidade é superior ao dobro da densidade inicial de $\frac{1}{M} = \frac{1}{100} = 0,01$.

Em relação ao tempo por iteração, a média de resolução da iteração é de 91,91 segundos. O gráfico da Figura 4.8 contém o tempo por iteração, onde notoriamente o tempo do problema mestre é muito superior ao tempo dos problemas auxiliares. O mesmo gráfico também contém a quantidade de iterações, onde observa-se que a quantidade de iterações é proporcional ao tempo gasto.

O gráfico da Figura 4.9 contém o custo reduzido total por iteração. Contém também o MCRT e a quantidade acumulada de melhorias. Nota-se que nas iterações finais há poucas melhorias e redução do CRT.

Ao correlacionar os gráficos, observa-se que por volta da iteração 150, há um aumento de tempo necessário para resolver o problema mestre, assim como a redução na melhoria do CRT.

4.14.5

CG1 - Instância a1_4

Em relação à instância a1_4, a evolução do objetivo é apresentada no gráfico da Figura 4.10. Nota-se que não há alteração no valor do objetivo. O mesmo gráfico também expõe a densidade do problema a cada iteração, onde a densidade é superior ao triplo da densidade inicial de $\frac{1}{M} = \frac{1}{50} = 0,02$.

O tempo médio de resolução de cada iteração é de 116,75 segundos. O gráfico da Figura 4.11 contém o tempo por iteração, onde também o tempo do problema mestre foi muito superior ao tempo dos problemas auxiliares. O mesmo gráfico também contém a quantidade de iterações, onde observa-se que a quantidade de iterações é proporcional ao tempo gasto.

No gráfico da Figura 4.12 está apresentado o custo reduzido total por iteração. Contém também o MCRT e a quantidade acumulada de melhorias.

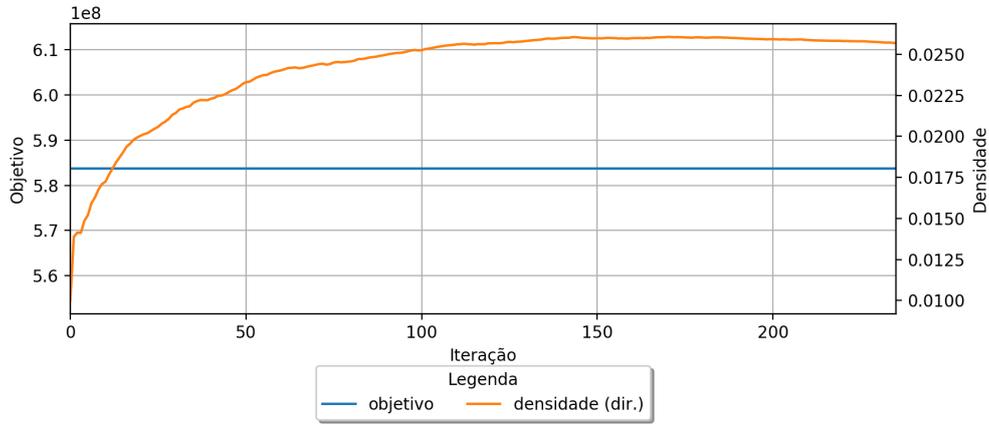


Figura 4.7: Objetivo e densidade de CG1, a1_3

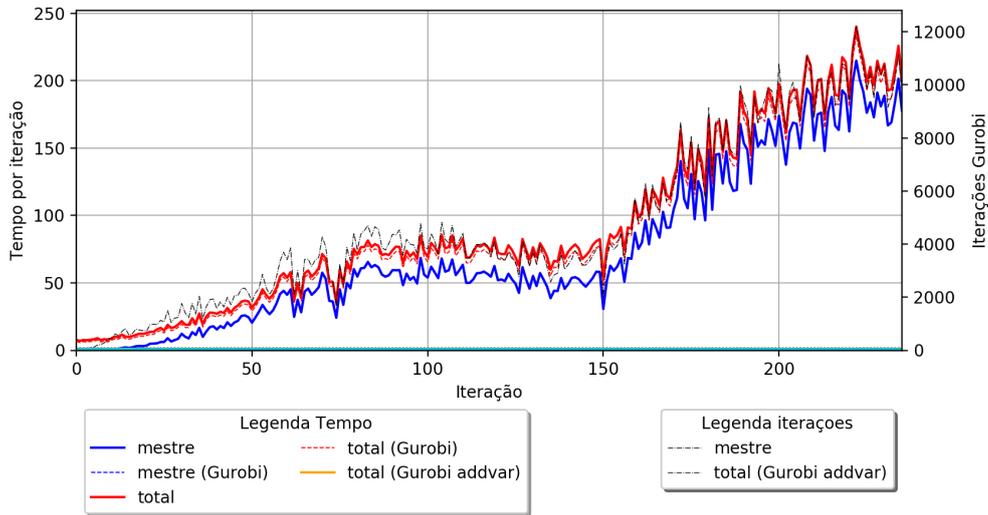


Figura 4.8: Tempos e iterações de CG1, a1_3

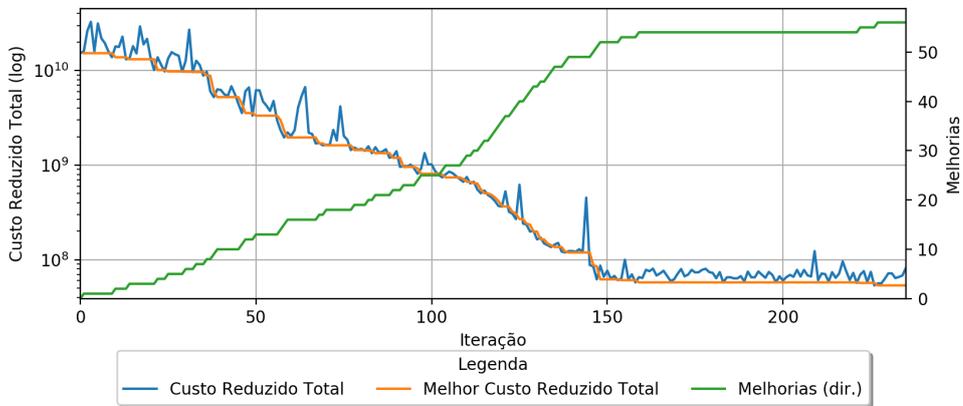


Figura 4.9: Custo reduzido de CG1, a1_3

Nota-se que nas primeiras 100 iterações há quase 75% das melhorias.

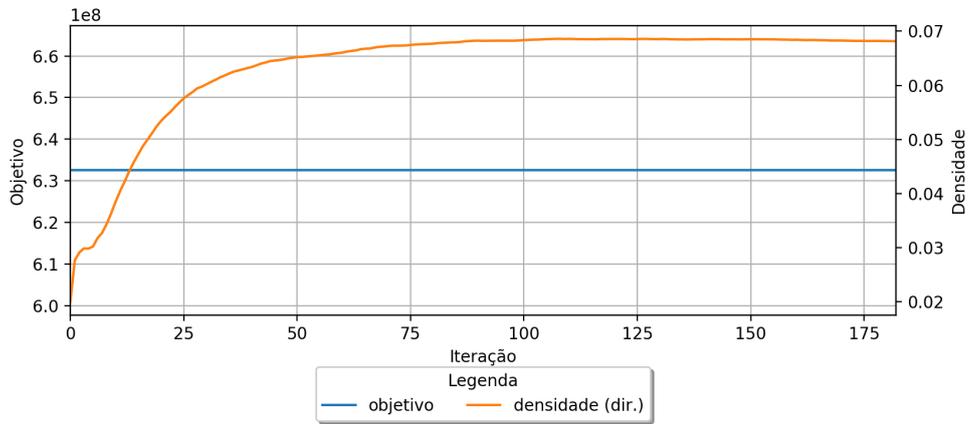


Figura 4.10: Objetivo e densidade de CG1, a1_4

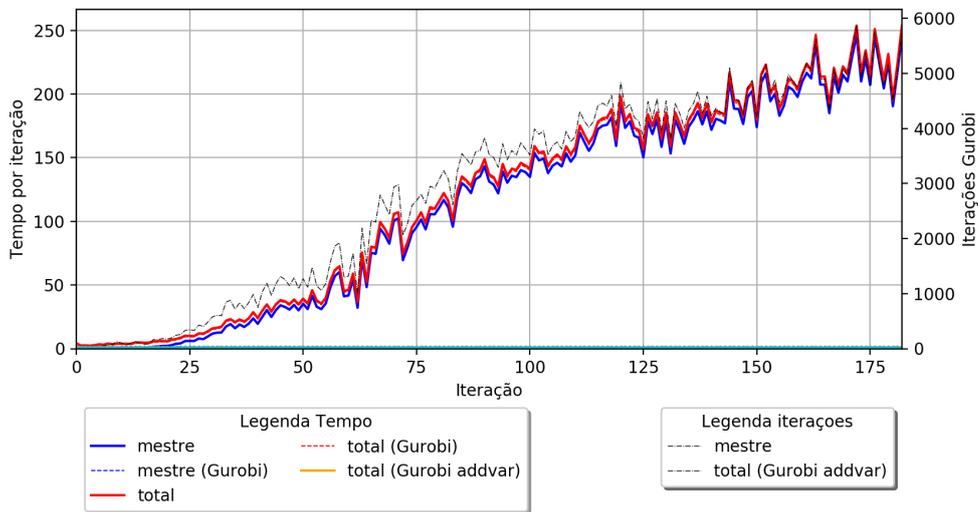


Figura 4.11: Tempos e iterações de CG1, a1_4

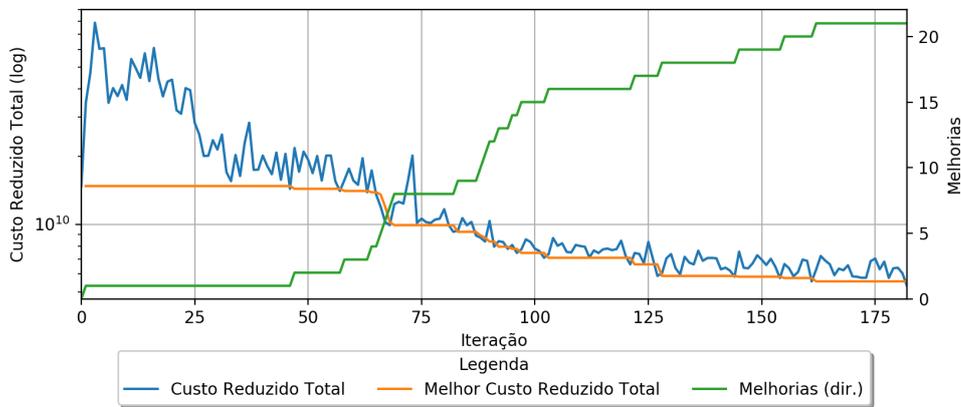


Figura 4.12: Custo reduzido de CG1, a1_4

4.14.6

CG1 - Instância a1_5

Em relação à instância a1_5, a evolução do valor do objetivo é apresentada no gráfico da Figura 4.13. Nota-se que não há alteração no valor do objetivo. O mesmo gráfico também expõe a densidade do problema a cada iteração, que aumenta a cada iteração, até atingir um valor próximo do dobro da densidade inicial de $\frac{1}{M} = \frac{1}{12} = 0,083\bar{3}$.

Em relação ao tempo por iteração, as iterações resolvem em uma média de 91,91 segundos. O gráfico da Figura 4.14 contém o tempo por iteração, onde notoriamente o tempo do problema mestre foi muito superior ao tempo dos problemas auxiliares. O mesmo gráfico também contém a quantidade de iterações, onde observa-se que a quantidade de iterações é proporcional com o tempo gasto.

O gráfico da Figura 4.15 contém o custo reduzido total por iteração. Assim como o MCRT e a quantidade acumulada de melhorias. Nota-se que nas primeiras 150 iterações há poucas melhorias e o valor de CRT tem variações de uma ordem de grandeza. Após a iteração 150, a oscilação do CRT reduz, e há várias melhorias, indicando que diferença entre o CRT gerado e o MCRT até então são de pequena variação, se comparado com a ordem de grandeza.

4.14.7

Modelo completo - CG2

Com base nos ensaios do modelo reduzido *CG1*, são realizados novos ensaios com um modelo completo, denominado de *CG2*, contendo todas as restrições previamente elencadas.

Os resultados dos ensaios do modelo completo nas instâncias a1 estão presentes na Tabela 4.2. Lembrando que a coluna *Objetivo* contém o custo final validado pelo programa *checker* fornecido pelos proponentes e onde há o valor *ST* cuja a solução fornecida foi a *Solução Trivial*, ou seja, não há mudança em relação à configuração inicial. A coluna *Tempo* informa o tempo, em segundos que a execução utiliza. A coluna *iterações* contém o número de iterações na etapa de geração de colunas. A coluna *Melhorias* informa quantas vezes o custo reduzido total desta iteração é melhor que o melhor até então estabelecido. A coluna *MCRT* informa o último melhor custo reduzido total.

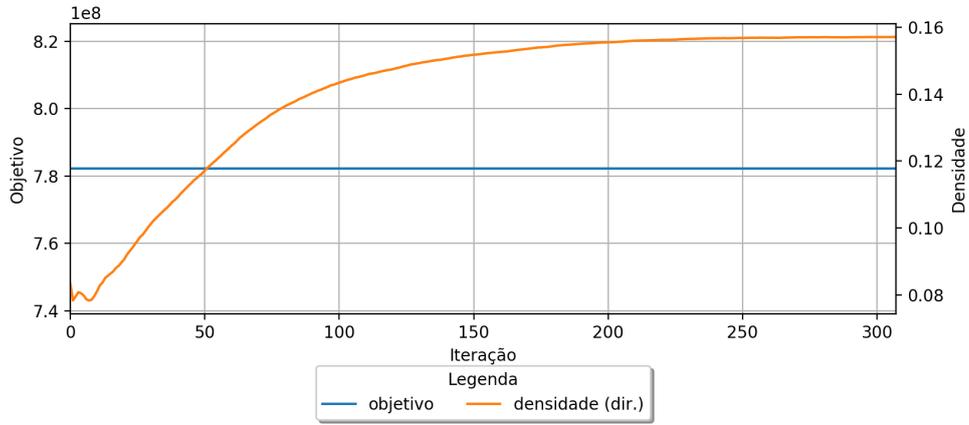


Figura 4.13: Objetivo e densidade de CG1, a1_5

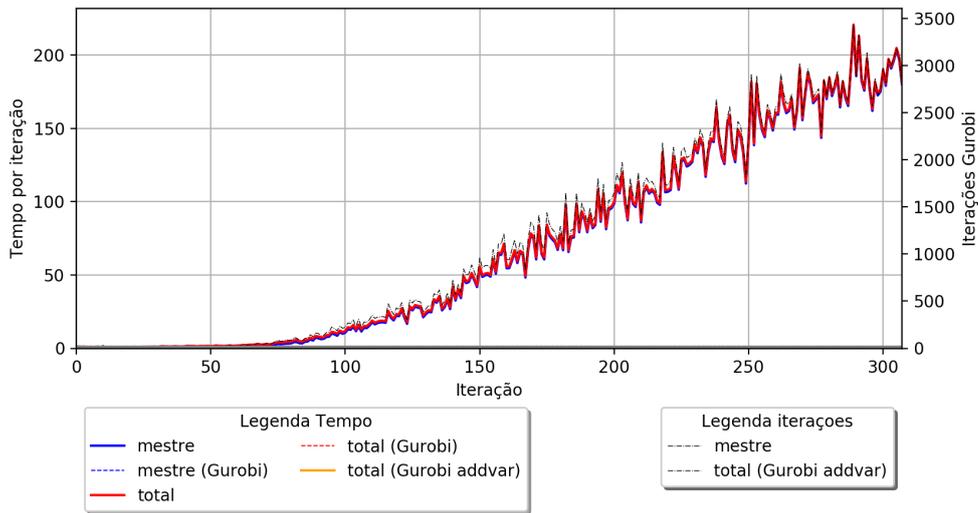


Figura 4.14: Tempos e iterações de CG1, a1_5

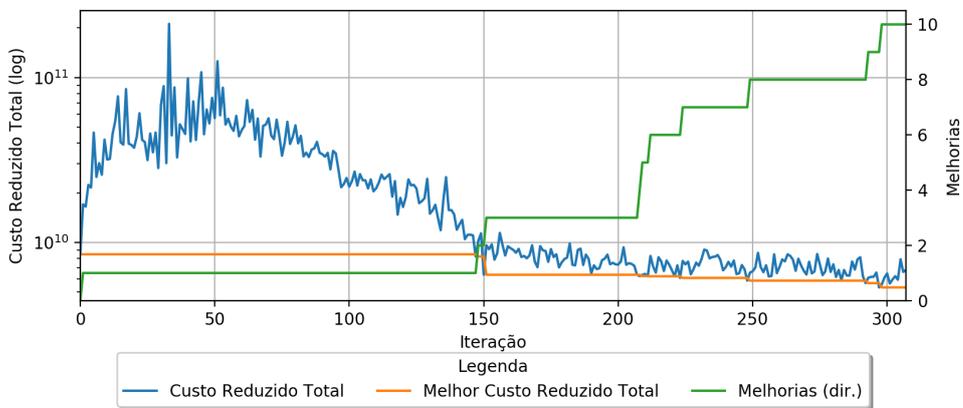


Figura 4.15: Custo reduzido de CG1, a1_5

A coluna *Motivo* é o motivo de término do ensaio. São utilizados os mesmos parâmetros do resolvidor adotados em *CG1*.

Tabela 4.2: Resultado parcial por geração de colunas, CG2

Instância	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
a1_1	ST	3822,117	CGRep	2066	92	-812,091
a1_2	ST	21600,063	TCG	69	3	-11.846.713.877,557
a1_3	ST	21600,102	TCG	58	8	-7.538.322.348,657
a1_4	ST	21600,091	TCG	68	50	-11.202.433.750,248
a1_5	ST	21600.037	TCG	163	1	-8.475.723.770,000

Com base na Tabela 4.2, pode-se perceber que, usando todas as restrições do problema há um aumento de dificuldade, pois nenhum obteve solução distinta da trivial. Para as instâncias que são terminadas devido ao tempo limite, a quantidade de iterações realizadas é menor quando comparada com a versão simplificada. Os melhores custos reduzidos finais são ordens de grandeza superiores as execuções da versão simplificada.

4.14.8

CG2 - Instância a1_1

A instância a1_1 termina a etapa de geração de colunas, pois na última iteração de geração de colunas, todos os problemas auxiliares retornaram variáveis já definidas. O gráfico da Figura 4.16 expõe a evolução do objetivo ao longo das iterações. Pode-se notar que o valor reduz, entretanto, ao resolver o MIP resultante, a solução obtida é a solução trivial. Apesar da redução do objetivo, as colunas geradas não permitem a existência de uma solução inteira, distinta da solução trivial.

Vale notar que para esta mesma instância, os ensaios de *CG1* e *CG2* têm uma quantidade de iterações semelhantes, 2071 e 2066 respectivamente, ou seja uma diferença de 0,24%. Entretanto, o tempo consumido por *CG2* é muito superior que em *CG1*. Na versão simplificada o tempo foi de 523,820s contra 3822,117s do modelo completo, ou seja, um aumento de 629%. Em relação ao número de melhorias no custo reduzido total foi menor. Este índice menor pode estar relacionado à parada por repetição das colunas geradas. Porém, como ambos ensaios executaram quase a mesma quantidade de iterações, a quantidade de melhorias por iteração no modelo completo é inferior em relação ao modelo simplificado, na ordem de 51%.

O gráfico da Figura 4.17 expõe o tempo por iteração, onde notoriamente

o tempo do problema mestre foi muito superior ao tempo dos problemas auxiliares. O mesmo gráfico também contém a quantidade de iterações, que excetuando-se as primeiras iterações, o resolvidor do problema mestre utiliza muitas iterações.

Em relação ao custo reduzido, cuja evolução é apresentada em escala logarítmica do valor absoluto no gráfico da Figura 4.18, observa-se que nas primeiras 250 iterações há uma redução de quase metade na ordem de grandeza. Nas 1000 iterações seguintes há a redução de apenas uma ordem de grandeza. Em relação às melhorias, observa-se que quase metade das melhorias ocorre nas primeiras 250 e uma quantidade semelhante ocorre nas 1000 seguintes.

4.14.9

CG2 - Instância a1_2

Em relação à instância a1_2, a evolução do objetivo ao longo das iterações é apresentado no gráfico da Figura 4.19. Observa-se que não ocorre alteração no valor ao longo das iterações. O mesmo gráfico expõe a densidade do problema por iteração, e a densidade do problema aumenta com as iterações, terminando com um valor superior ao dobro do inicial.

O tempo por iteração é apresentado no gráfico da Figura 4.20, e percebe-se que o tempo necessário aumenta com o passar das iterações, atingindo a ordem de 20 minutos nas últimas iterações, cujo principal componente é problema mestre. Observa-se também que a quantidade de iterações no mestre varia conforme o tempo necessário.

O gráfico da Figura 4.21 contém a evolução do custo reduzido, onde observa-se que as reduções no MCRT foram de pequena magnitude, se comparado com a ordem de grandeza do valor.

4.14.10

CG2 - Instância a1_3

Para a instância a1_3, a evolução do objetivo ao longo das iterações é apresentado no gráfico da Figura 4.22. Observa-se que não há alteração no valor ao longo das iterações. O mesmo gráfico expõe a densidade do problema

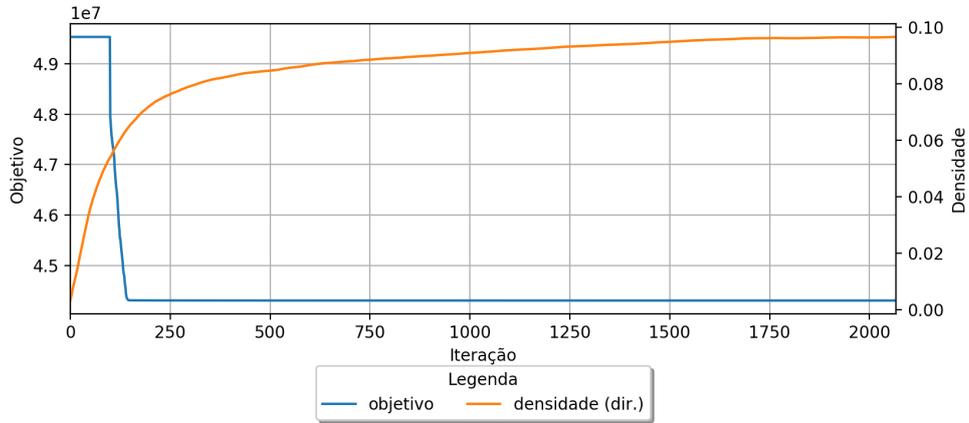


Figura 4.16: Objetivo e densidade de CG2, a1_1

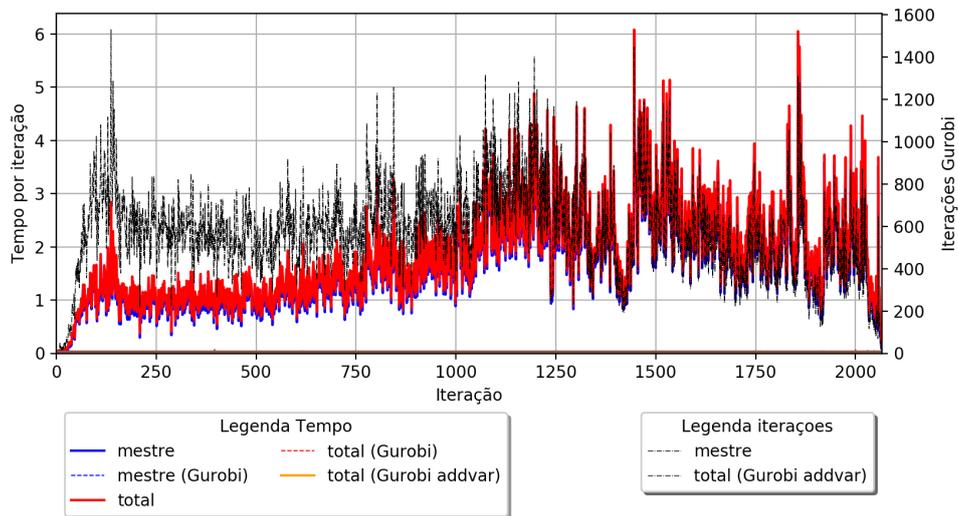


Figura 4.17: Tempos e iterações de CG2, a1_1

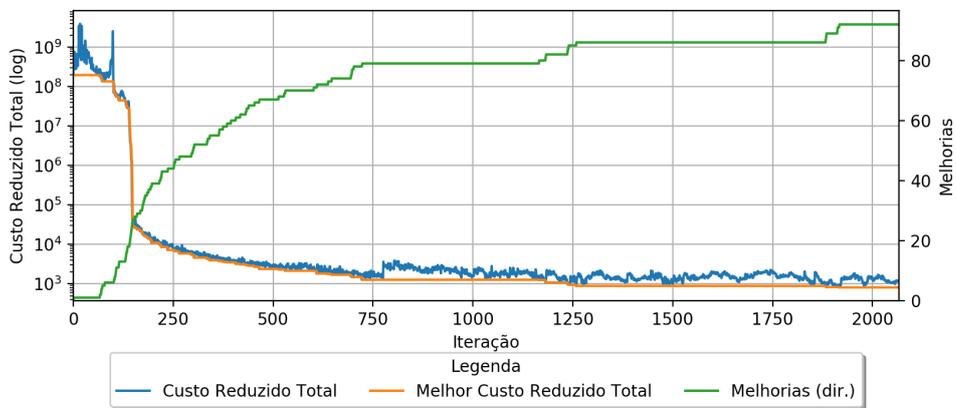


Figura 4.18: Custo reduzido de CG2, a1_1

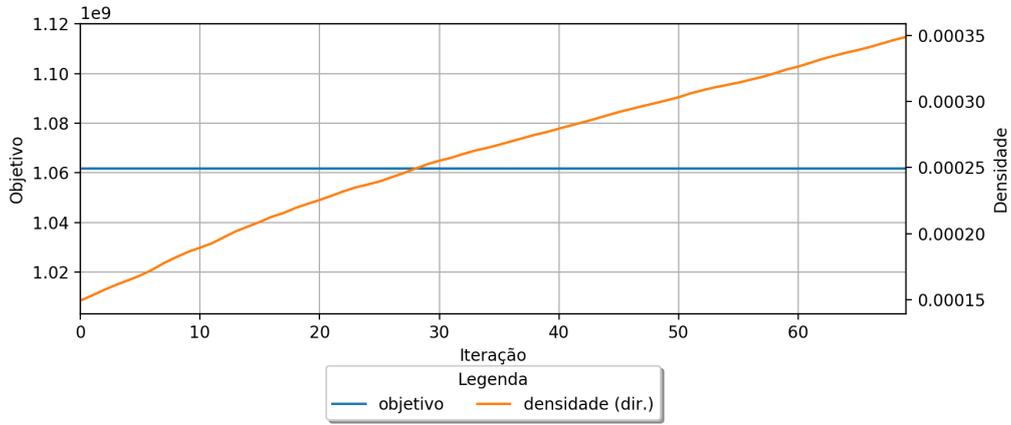


Figura 4.19: Objetivo e densidade de CG2, a1_2

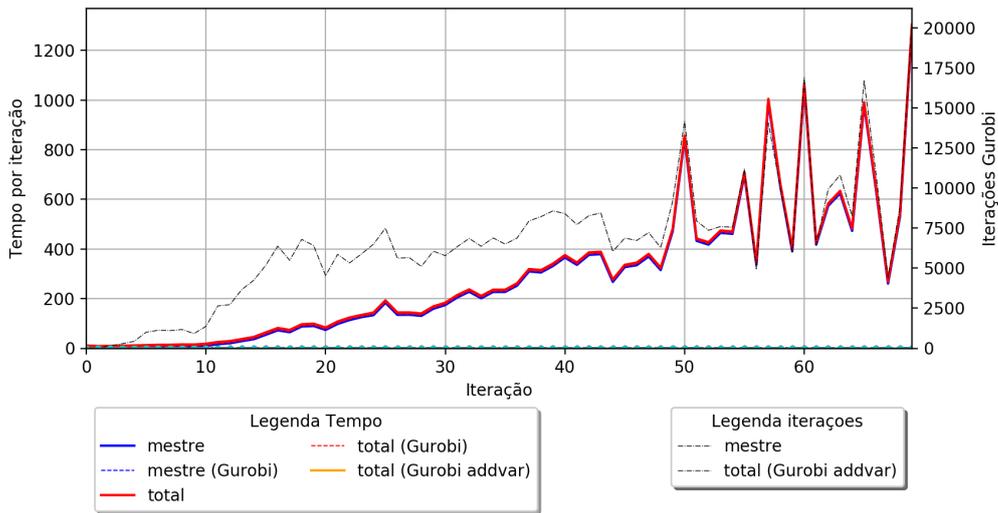


Figura 4.20: Tempos e iterações de CG2, a1_2

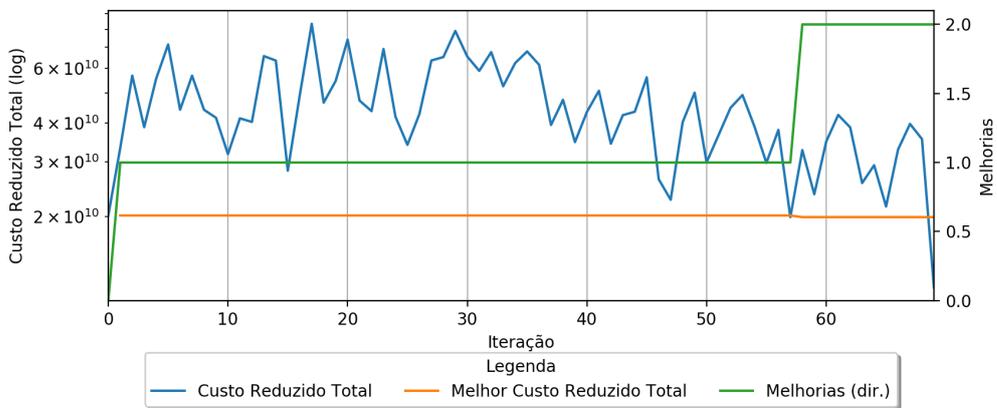


Figura 4.21: Custo reduzido de CG2, a1_2

por iteração, e nota-se que a densidade do problema aumenta com as iterações, terminando com um valor superior a 6 vezes do inicial.

O tempo por iteração é apresentado no gráfico da Figura 4.23, que de forma análoga a instância a1_3, o tempo necessário aumenta com o passar das iterações, atingindo na ordem de 25 minutos nas últimas iterações, cujo principal componente é problema mestre. Observa-se também que a quantidade de iterações no mestre varia conforme o tempo necessário.

O gráfico da Figura 4.24 contém a evolução do custo reduzido, onde pode-se notar que há a variação de mais de uma ordem de grandeza no valor do CRT. Também observa-se que quase todas as melhorias ocorrem nas últimas 20 iterações

4.14.11

CG2 - Instância a1_4

A evolução do objetivo para a instância a1_4 ao longo das interações é apresentada no gráfico da Figura 4.25. Não há alteração no valor ao longo das iterações. O mesmo gráfico expõe a densidade do problema por iteração e a densidade do problema foi aumentando com as iterações, terminando com um valor superior à seis vezes do inicial.

O tempo por iteração é apresentado no gráfico da Figura 4.26, que de forma análoga às instâncias a1_2 e a1_3, o tempo necessário foi aumentando com o passar das iterações, atingindo a ordem de 25 minutos nas últimas iterações, cujo principal componente é problema mestre. Observa-se também que a quantidade de iterações no mestre varia conforme o tempo necessário.

O gráfico da Figura 4.24 contém a evolução do custo reduzido, onde pode-se notar que há a variação de mais de um uma ordem de grandeza no valor do CRT. Também observa-se que quase todas as melhorias ocorrem nas últimas 20 iterações

4.14.12

CG2 - Instância a1_5

Com relação à instância a1_5, não ocorre alteração no valor do objetivo, conforme apresentado gráfico da Figura 4.28. Observa-se que não há alteração

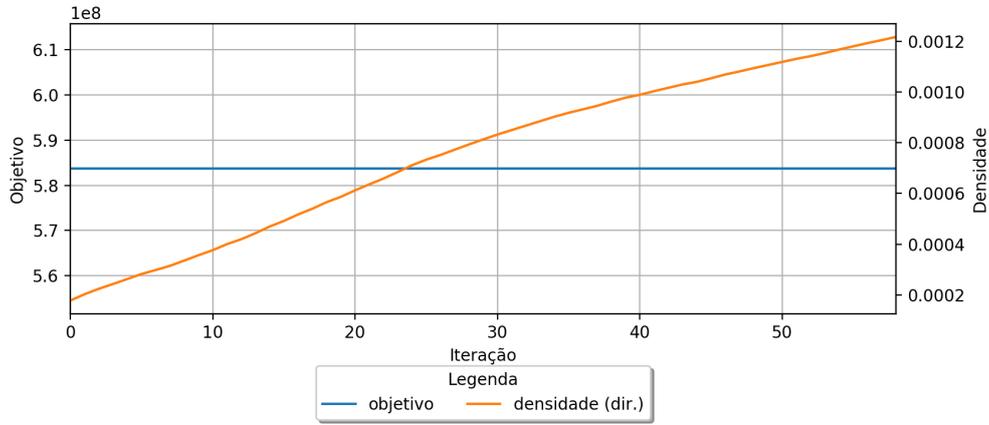


Figura 4.22: Objetivo e densidade de CG2, a1_3

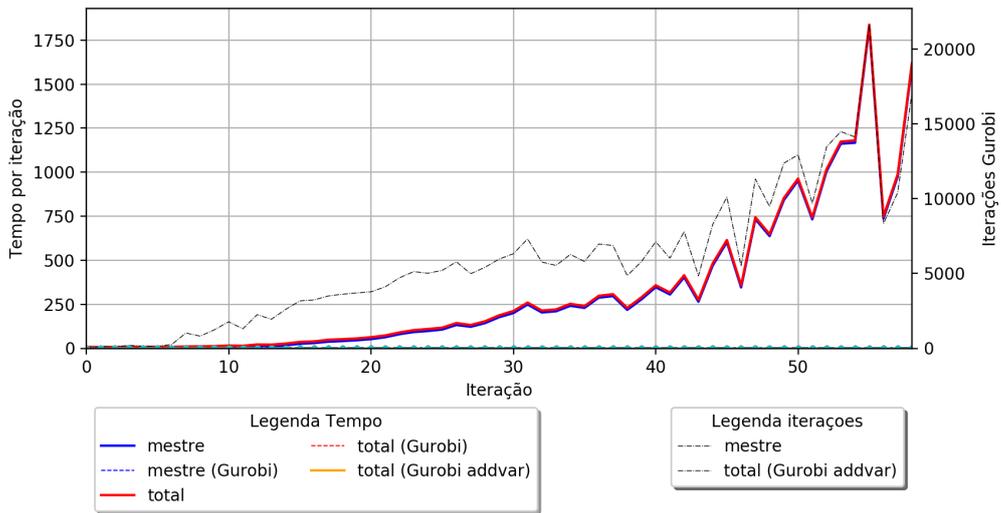


Figura 4.23: Tempos e iterações de CG2, a1_3

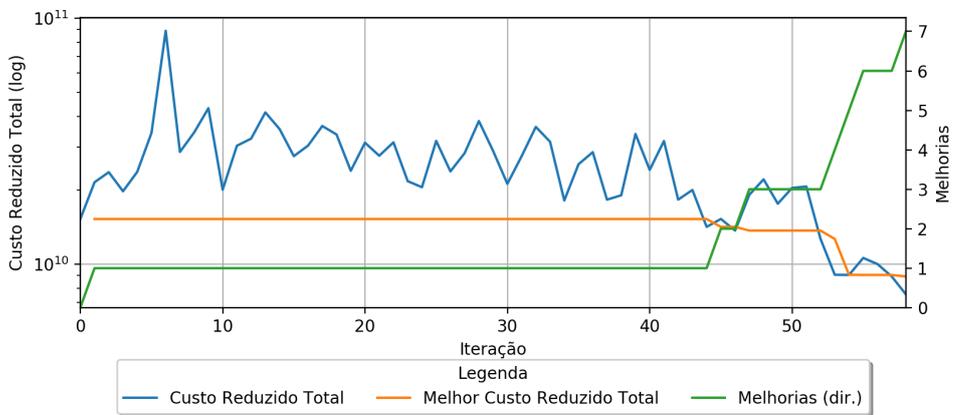


Figura 4.24: Custo reduzido de CG2, a1_3

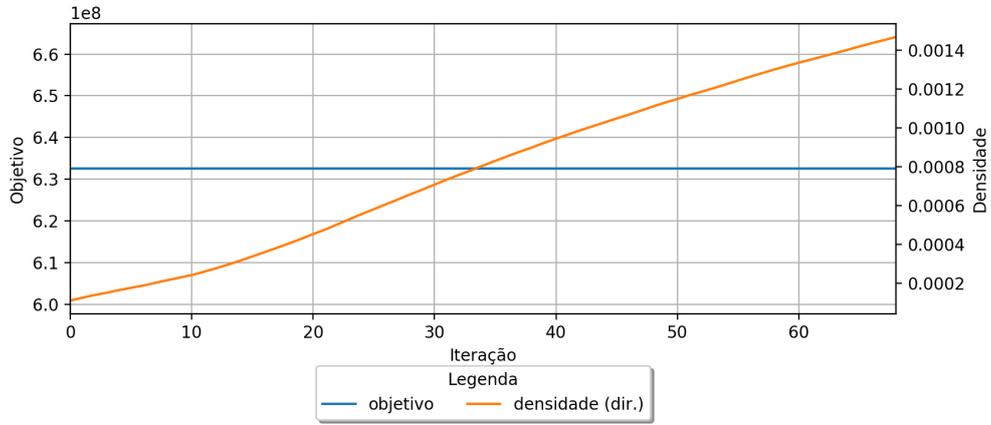


Figura 4.25: Objetivo e densidade de CG2, a1_4

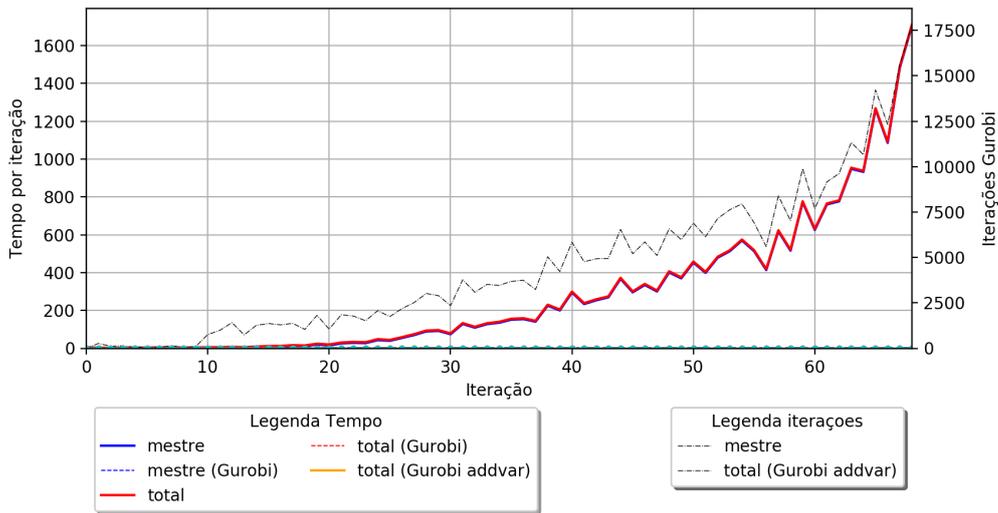


Figura 4.26: Tempos e iterações de CG2, a1_4

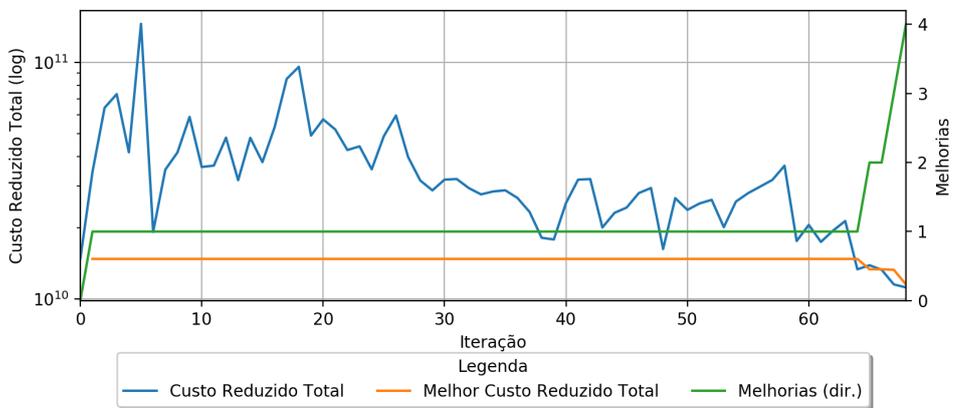


Figura 4.27: Custo reduzido de CG2, a1_4

no valor ao longo das iterações. A densidade do problema também aumenta com as iterações, terminando com um valor superior a 10 vezes do inicial. A densidade inicial desta instância é de 0,018%, composta por 33.449 restrições e 6.880 variáveis contendo 41437 não zeros. Na última iteração engloba 8.836 variáveis com 1.506.132 de elementos não nulos. Portanto, há um aumento de mais de 50 vezes.

Em relação ao tempo, o problema mestre foi o principal componente de tempo a cada iteração, como apresentado no gráfico da Figura 4.29. Observa-se que o tempo e as quantidades de iterações do resolvidor do problema mestre aumentam com o passar das iterações. Nas últimas iterações, é necessária a ordem de 10 minutos para resolver o problema mestre.

Na evolução do custo reduzido, exposto no gráfico da Figura 4.30, pode-se notar que há uma oscilação de uma ordem de grandeza nas primeiras 20 iterações. Observa-se também que após a iteração 40 a oscilação do valor do CRT reduz.

4.14.13

Comentários

Com base nos ensaios anteriores, pode-se concluir que os problemas auxiliares possuem rápida execução, ao contrário do que ocorre com os problemas mestres. Também tem-se que as restrições relacionadas com serviço resultam em um aumento de complexidade no problema. Vale notar que o espaço de soluções da versão simplificada *CG1* contém o espaço de soluções do modelo completo *CG2*. Com base nesta característica de continência¹ e o fato do modelo simplificado necessitar de menos tempo por ciclo de geração de coluna, pode-se então realizar a construção de um primeiro conjunto de colunas por meio de *CG1*. Ao terminar a geração de colunas em *CG1*, pode-se completar o modelo com as restrições relacionadas a serviços e resolver o modelo completo.

Dos mesmos ensaios, o tempo despendido pelo problema mestre é elevado, assim como o custo reduzido total entre iterações pode ter uma variação em ordens de grandeza, tornando a convergência lenta. Deve-se então buscar mecanismos para acelerar a resolução do problema mestre e reduzir a amplitude de oscilação do custo reduzido total.

¹soluções de *CG1* contém as soluções de *CG2*

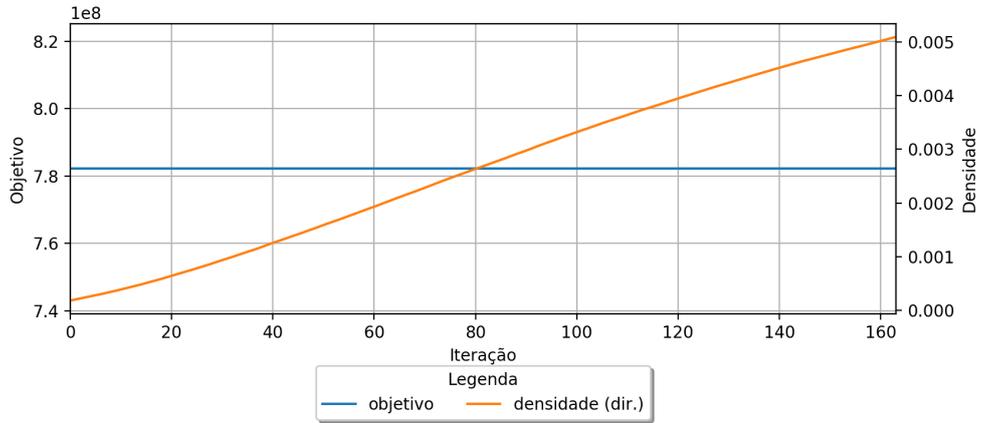


Figura 4.28: Objetivo e densidade de CG2, a1_5

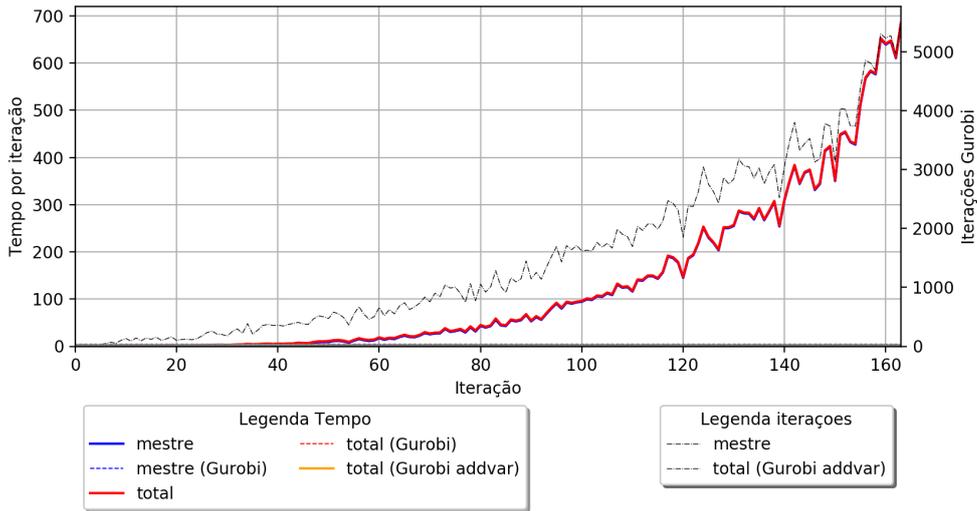


Figura 4.29: Tempos e iterações de CG2, a1_5

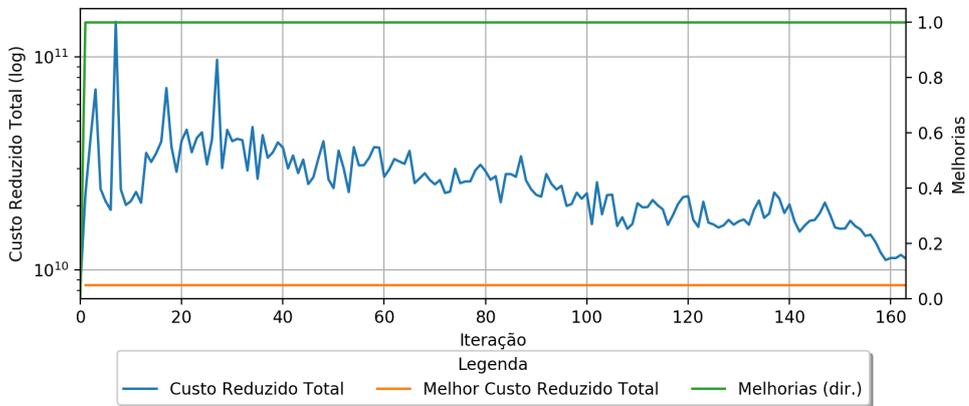


Figura 4.30: Custo reduzido de CG2, a1_5

Os resultados previamente apresentados de CG1 servirão de controle para os novos ensaios de geração de coluna, que visam melhorar tanto o resultado, como o tempo.

4.15

Otimização após adição de coluna

A partir dos resultados da seção anterior, observa-se que a densidade do problema aumenta com o passar das iterações. Com base nesse fato, forma-se a hipótese que cada nova variável pode formar muitas arestas com as variáveis já existentes e, ao realizar a otimização da nova iteração com todas as novas variáveis, o resolvidor requer muitas iterações para determinar a nova base do problema.

Uma possibilidade a ser explorada é a otimização do mestre após a adição de cada nova variável gerada. Como o resolvidor é parametrizado para usar uma partida a quente, então após a adição de uma variável pode-se realizar a otimização com um espaço de soluções ligeiramente maior.

A Tabela 4.3 contém resultados de realizar a otimização após a adição de cada nova variável no modelo simplificado *CG1* com as instância **a1**. Somente a instância **a1_1** concluiu a execução, as demais são interrompidas por tempo. Nestas instâncias interrompidas, o tempo de execução é substancialmente superior ao limite estipulado. Isto se deve pelo fato do controle de tempo para interromper o programa somente é realizado após o mestre reduzido ou do último problema auxiliar de precificação.

Para as instâncias **a1_2**, **a1_3** e **a1_4**, os valores absolutos do melhor custo reduzido total são superiores do que no ensaio de controle. Somente para a instância **a1_5** o valor de MCRT é melhor. Em termos de iterações, somente as instâncias **a1_4** e **a1_5** possuem mais iterações com a otimização após a adição do que no controle.

Tabela 4.3: Resultado parcial por geração de colunas, CG1, com otimização após cada nova coluna

Instância	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
a1_1	44.306.501	628,460	Concluído	1741	140	-0,085
a1_2	ST	21784,699	TCG	183	13	-6.695.745.720,815
a1_3	ST	21622,581	TCG	224	49	-176.015.323,131
a1_4	ST	21752,684	TCG	194	16	-6.605.570.634,171
a1_5	ST	21698,931	TCG	346	14	-4.376.081.256,537

4.15.1

Instância a1_1

Para a instância a1_1, a evolução do custo é apresentada no gráfico da Figura 4.31, onde o valor ótimo é atingido nas primeiras 200 iterações. A etapa de geração de colunas é interrompida, pois o valor do custo reduzido total de $-0,085$ é maior que a tolerância de $-\epsilon = -0,1$. Em relação à densidade do problema, que é apresentada no mesmo gráfico, varia de 23% até 29% nas mesmas primeiras 200 iterações. Comparando com a evolução do ensaio de controle, pode-se notar que ambos apresentaram um comportamento semelhante em relação à objetivo e densidade.

Em relação a tempo por iteração, apresentado no gráfico da Figura 4.32, pode-se notar que o tempo total das otimizações após a adição de variáveis é o principal componente do tempo total da iteração. Pode-se notar que o tempo do problema mestre é praticamente constante, que consiste basicamente no pós processamento da otimização como, por exemplo, a coleta dos valores das variáveis duais. O tempo gasto de otimização pelo Gurobi é nulo, pois não há alteração do modelo após otimização da adição da última variável. Comparando os tempos totais de otimização após a adição de variáveis com os tempos do resolvidor do ensaio de controle, nota-se que são semelhantes, observa-se que estão na mesmas ordens de grandeza. Comparando os tempos totais das iterações, pode-se notar que o ensaio de controle é melhor, ou seja, o tempo médio das iterações é menor do que o tempo total deste ensaio.

Em relação à quantidade de iterações do Gurobi, a quantidade de iterações do mestre, por aspectos do resolvidor, é idêntica à quantidade de iterações de otimização do último problema auxiliar do ciclo anterior. Comparando a quantidade total de iterações do Gurobi dos problemas auxiliares deste ensaio, com a quantidade do problema mestre do ensaio de controle, as iterações do resolvidor estão na mesma ordem de grandeza.

O gráfico da Figura 4.33, contém a evolução do custo reduzido total. Pode-se observar que há muita oscilação nas iterações iniciais, variando em até 2 ordens de grandeza, de 10^8 até 10^{10} . Após a iteração 1600, nota-se uma convergência a 0. Em relação à quantidade de melhorias, pode-se notar que estas foram mais frequentes, pois nas 1741 existem 140 melhorias, resultando em média de 1 melhoria a cada 12 ou 13 iterações. O ensaio de controle tem as 100 melhorias em 2071 iterações, resultando em uma média de 20.71. Comparando com o ensaio de controle, pode-se notar que o custo reduzido, em

geral, teve oscilação menor.

Em resumo, a otimização após a adição de variável resulta em menos iterações totais, com uma maior quantidade de melhorias. Entretanto, é necessário uma quantidade de tempo maior.

4.15.2

Instância a1_2

A evolução do valor da função objetivo da instância a1_2 é apresentada no gráfico da Figura 4.34, onde é mostrado uma pequena redução relativa no objetivo próximo da iteração 175. O mesmo gráfico contém a densidade do problema mestre, onde se percebe que a densidade do problema é superior a 2%.

Em relação ao tempo de cada iteração, presente no gráfico da Figura 4.35, nota-se que o tempo de otimização após otimizações é o principal componente do tempo total da iteração. O tempo total do resolvidor nos problemas mestre e auxiliares é praticamente constante. Comparando com o ensaio de controle, é observável que os tempos de cada iteração deste ensaio é, em geral, superior ao controle. Outro detalhe contido no gráfico é que os tempos do problema mestre são virtualmente nulos, isto é, o tempo desprendido com a otimização do problema mestre acrescido de pós processamento possui ordem de grandeza pequena o bastante para não aparecer no gráfico. Em relação às iterações do resolvidor, a cada ciclo de geração de colunas, presente no mesmo gráfico, pode-se notar que a quantidade total de iterações após a adição de variável é superior ao ensaio de controle.

A evolução do custo reduzido total deste ensaio está no gráfico da Figura 4.36, onde nas 100 primeiras iterações ocorre praticamente todas as melhorias. Em relação ao desempenho, em média há uma melhoria a cada 14 ciclos, pior em relação ao ensaio de controle, que tem uma média de 6 iterações.

Com base nos dados deste ensaio, para a instância a1_2, a otimização após a adição não produz ganhos.

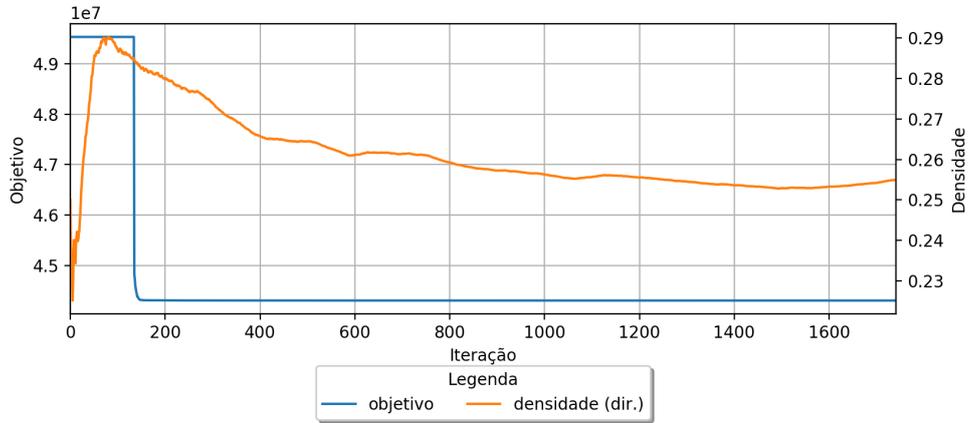


Figura 4.31: Objetivo e densidade de CG1, a1_1, com otimização após cada nova coluna

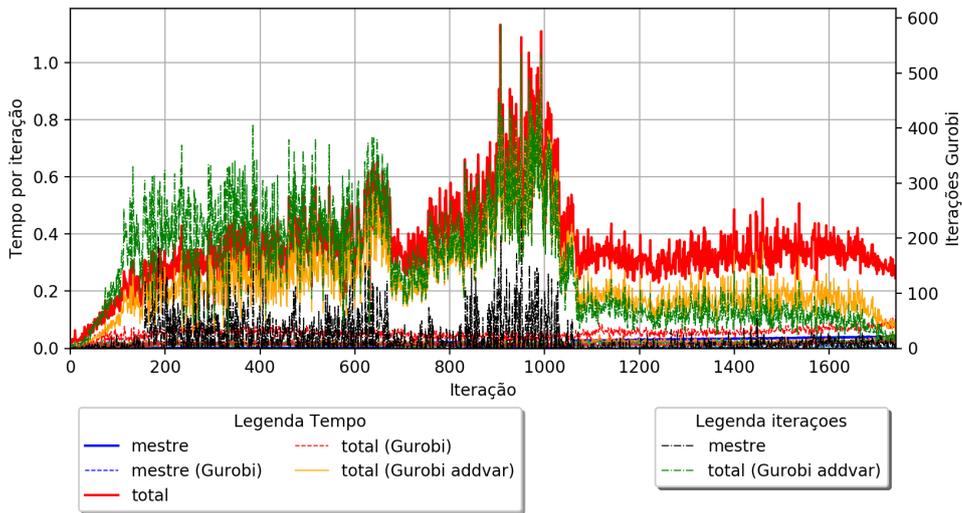


Figura 4.32: Tempos e iterações de CG1, a1_1, com otimização após cada nova coluna

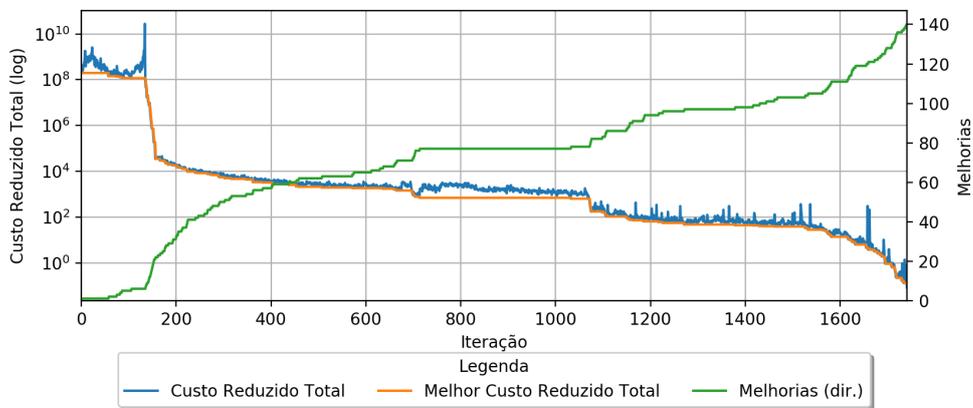


Figura 4.33: Custo reduzido de CG1, a1_1, com otimização após cada nova coluna

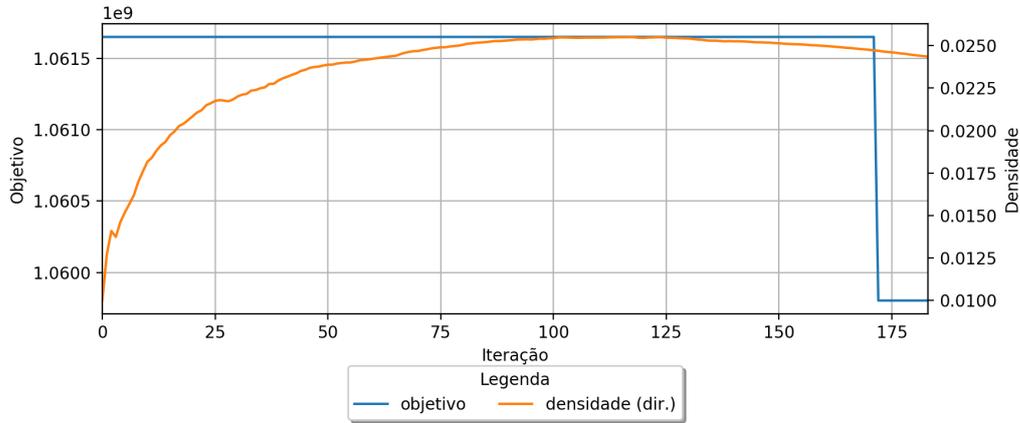


Figura 4.34: Objetivo e densidade de CG1, a1_2, com otimização após cada nova coluna

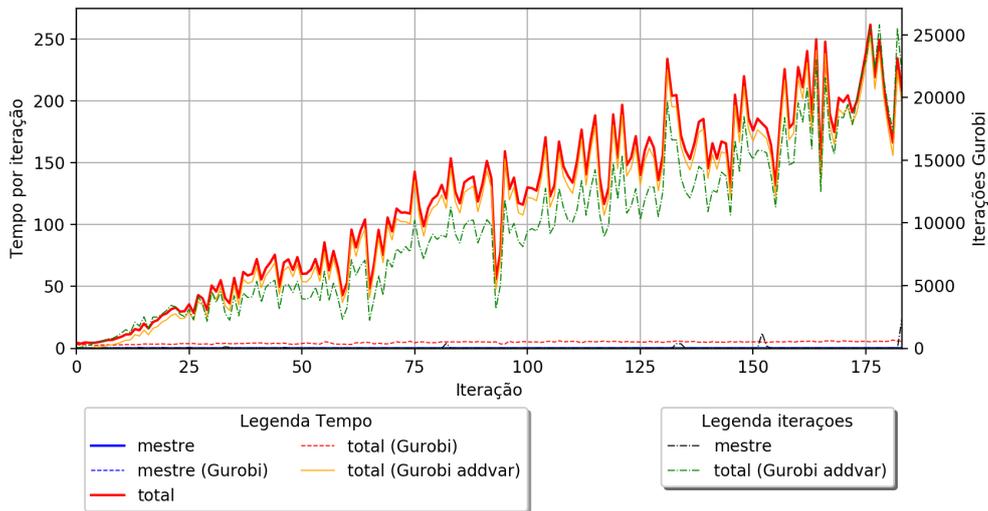


Figura 4.35: Tempos e iterações de CG1, a1_2, com otimização após cada nova coluna

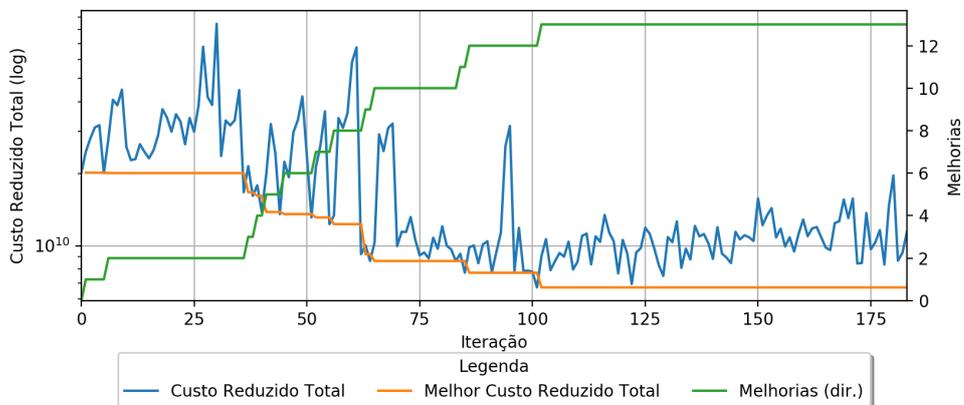


Figura 4.36: Custo reduzido de CG1, a1_2, com otimização após cada nova coluna

4.15.3

Instância a1_3

No caso da instância a1_3, o gráfico da Figura 4.37 mostra que não ocorre alteração no valor do objetivo. O mesmo gráfico também exhibe que a densidade tende a ser superior a 2% e atinge um valor próximo de 2,5%. Comparando com o ensaio de controle, pode-se perceber que a densidade deste ensaio tende a ser menor em relação à mesma iteração no ensaio de controle.

Em relação ao tempo por iteração, apresentado no gráfico da Figura 4.38, o tempo total de otimização após as adições de variáveis é o principal componente do tempo total. Também pode-se notar um aumento repentino próximo da iteração 100. Em relação à quantidade de total de iterações do resolvidor da otimização após cada adição de variáveis, observa-se que nas iterações finais é menor se comparado ao ensaio de controle.

A evolução do custo reduzido total é apresentada no gráfico da Figura 4.39. É mostrada uma maior oscilação até a iteração 100 do que nas iterações restantes e comparando com o ensaio de controle, o decréscimo do valor é menor. Neste ensaio, em média, ocorre uma melhoria a cada 4.57 iterações, uma média pior comparando com o ensaio de controle, que é de 3.98.

Assim como na instância a1_3, a otimização após cada adição de variável não produz ganhos.

4.15.4

Instância a1_4

Com respeito com a instância a1_4, não há alteração significativa no objetivo, como apresentado no gráfico da Figura 4.40, e a densidade do problema converge para 7%. Comparando com o controle, pode-se notar que este ensaio possui densidade maior.

Em relação ao tempo, os tempos por iteração deste ensaio, expostos no gráfico da Figura 4.41, são semelhantes aos tempos por iteração do ensaio de controle. Assim como nas instâncias anteriores, o tempo total de otimização após a adição de variáveis é o principal componente do tempo total da iteração. Quanto a quantidade de iterações, para esta instância, a quantidade total de iterações tende a ser inferior do que na instância de controle.

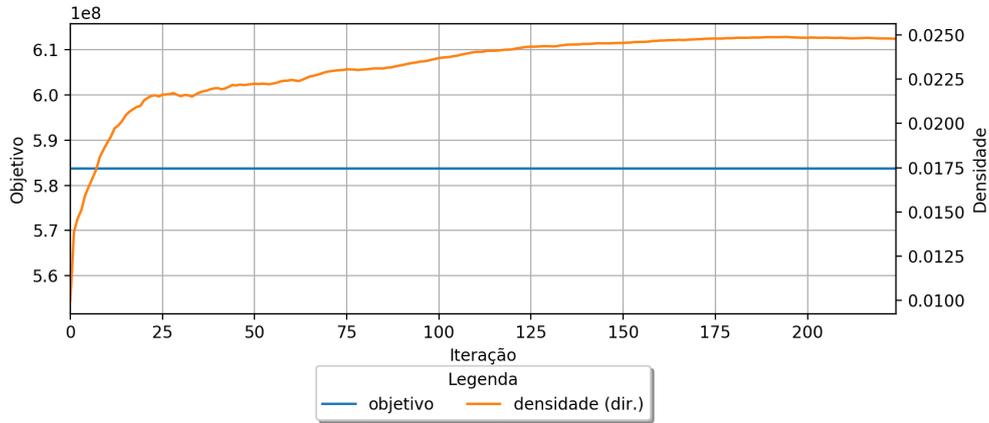


Figura 4.37: Objetivo e densidade de CG1, a1_3, com otimização após cada nova coluna

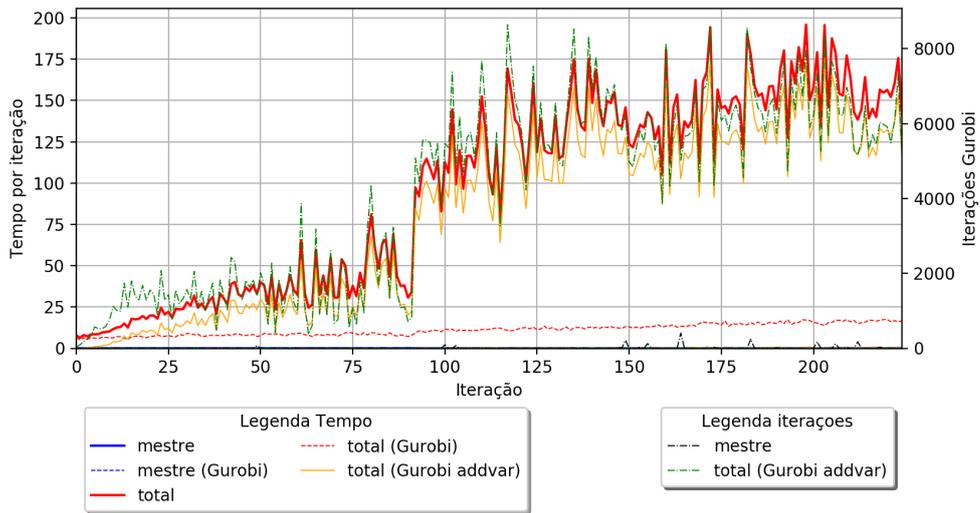


Figura 4.38: Tempos e iterações de CG1, a1_3, com otimização após cada nova coluna

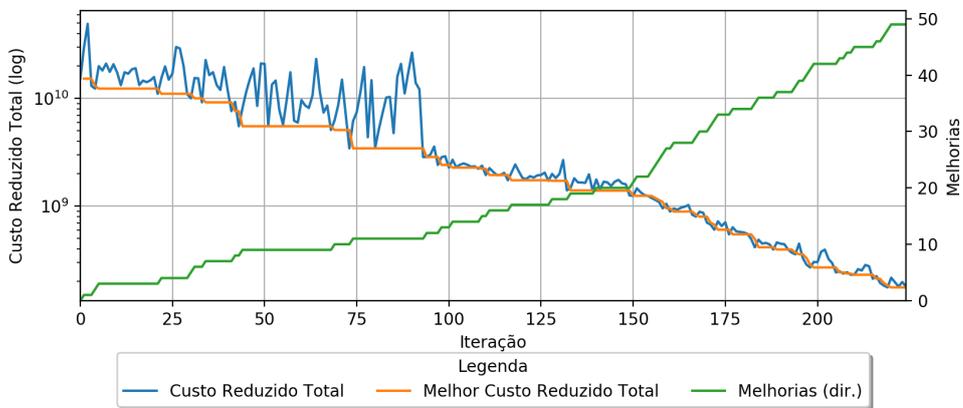


Figura 4.39: Custo reduzido de CG1, a1_3, com otimização após cada nova coluna

O gráfico da Figura 4.42 contém a evolução do custo reduzido total, onde nota-se a variação de amplitude é maior que no ensaio de controle.

Com base nos dados deste ensaio, a otimização após a adição de cada variável permitiu que para a instância a1_4 tivesse um número maior de iterações, entretanto, o custo reduzido total é pior para o mesmo tempo despendido.

4.15.5

Instância a1_5

De forma análoga a instância anterior, não há alteração significativa no objetivo da instância a1_5, como presente no gráfico da Figura 4.43. Comparando com o ensaio de controle, pode-se notar que a densidade deste ensaio é superior, mais próximo de 16%.

Em relação à tempo por iteração, como apresentado no gráfico da Figura 4.44, assim como nas demais instância, o tempo total de otimização após as adições de variáveis é a principal parcela de tempo despendido. Comparando com o ensaio de controle, pode-se perceber que o tempo total de cada iteração é menor, assim como a quantidade de iterações.

O gráfico da Figura 4.45 contém a evolução do custo reduzido total, onde percebe-se uma grande variação de valor nas 150 primeiras iterações. Após a iteração 150, além de reduzir a oscilação, as melhorias são mais frequentes. O valor final do custo reduzido também é melhor.

Após análise dos dados desde ensaio, conclui-se que a otimização após cada adição de variável permitiu resultados melhores para a instância a1_5.

4.15.6

Comentários sobre otimização após adição de variáveis

Com base nos resultados destes ensaios e também do ensaio de controle, pode-se notar comportamentos distintos em relação à tempo por iteração e iterações do Gurobi. Um motivo para tal característica pode ser resultante do fato dos subproblemas serem executados na mesma ordem e as novas variáveis, conseqüentemente, são analisadas na mesma ordem.

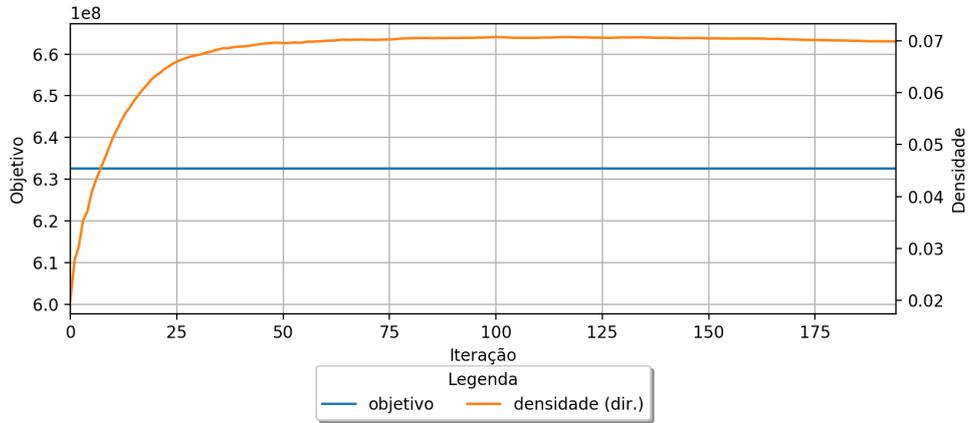


Figura 4.40: Objetivo e densidade de CG1, a1_4, com otimização após cada nova coluna

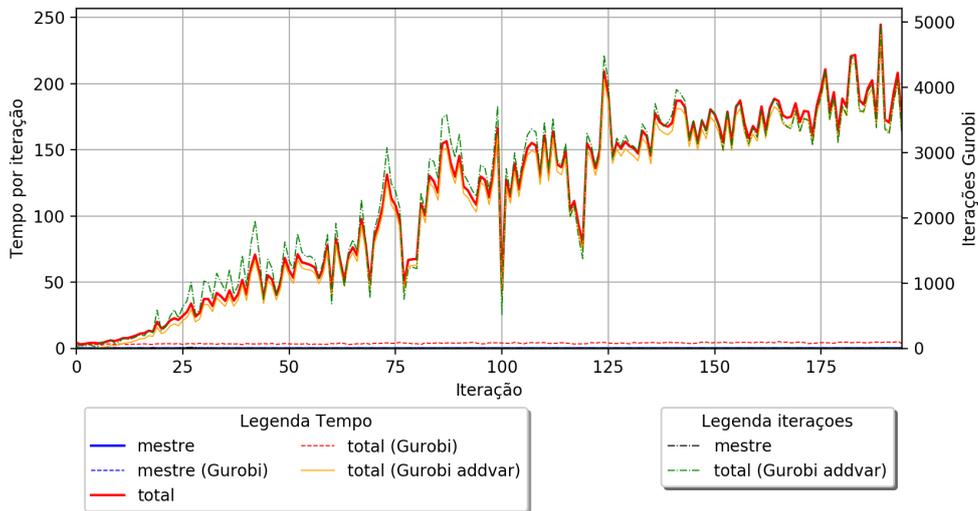


Figura 4.41: Tempos e iterações de CG1, a1_4, com otimização após cada nova coluna

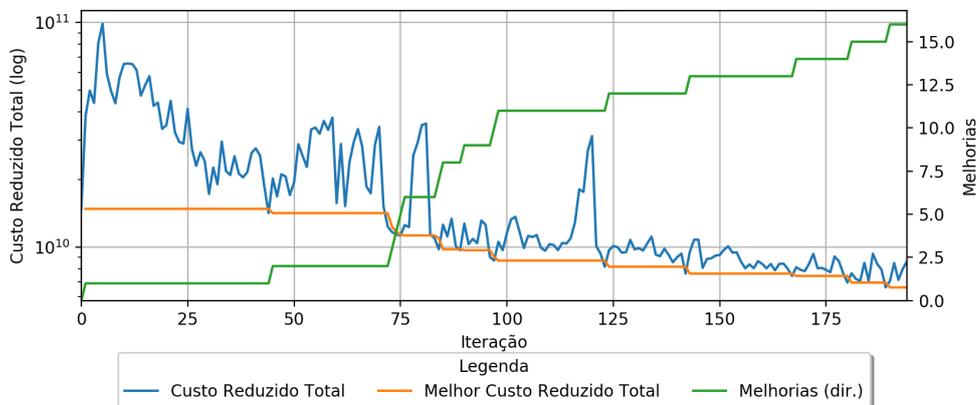


Figura 4.42: Custo reduzido de CG1, a1_4, com otimização após cada nova coluna

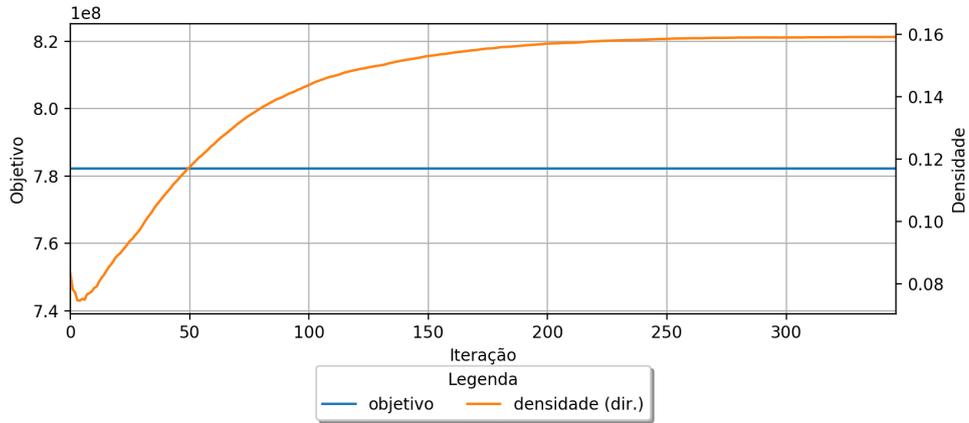


Figura 4.43: Objetivo e densidade de CG1, a1_5, com otimização após cada nova coluna

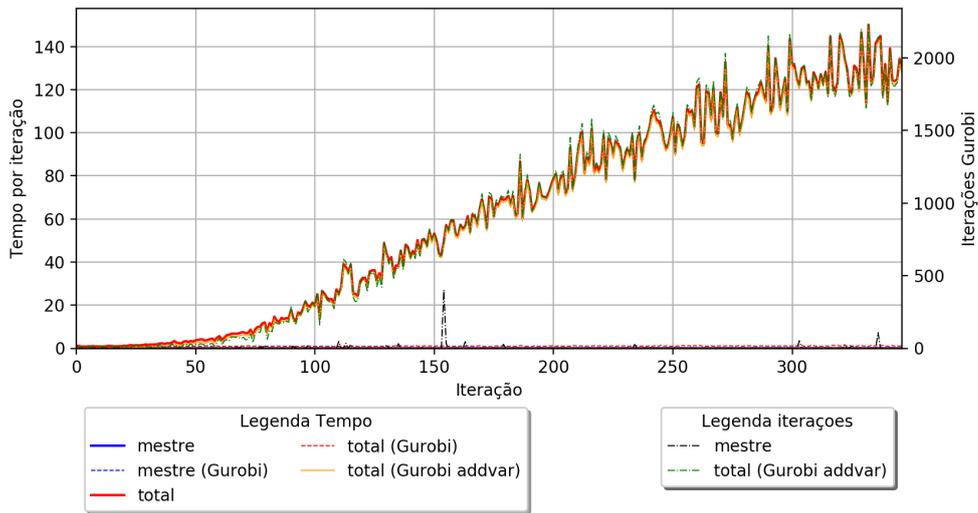


Figura 4.44: Tempos e iterações de CG1, a1_5, com otimização após cada nova coluna

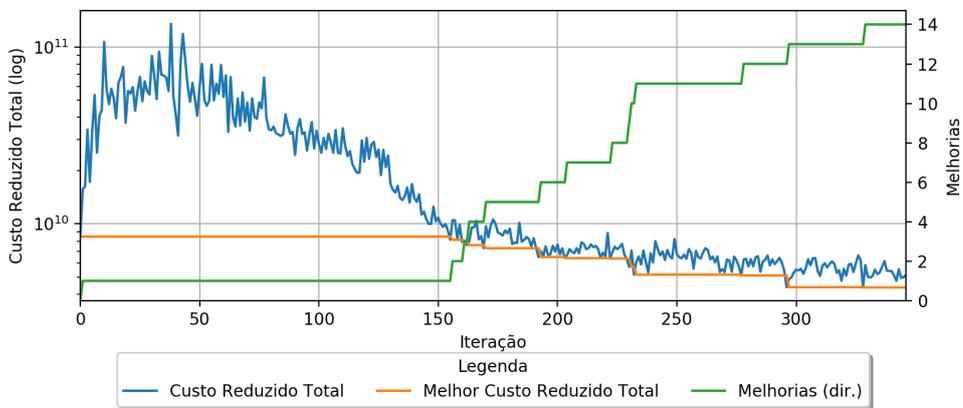


Figura 4.45: Custo reduzido de CG1, a1_5, com otimização após cada nova coluna

Também pode-se concluir que esta abordagem não é eficiente de forma geral, pois somente uma instância obteve ganhos, para as demais os resultados foram piores.

4.16

Geração prévia de colunas

Uma característica comum nas evoluções do custo reduzido total dos ensaios anteriores é a grande variação de amplitude nas iterações iniciais. Nos ensaios anteriores, cada máquina começava com somente uma possibilidade de alocação de processos, ou seja, a superfície inicial do poliedro é bastante reduzida. Uma outra abordagem visando melhores resultados é criar novas configurações e adicionar ao problema.

Uma forma direta de criação de novas colunas consiste em dada a configuração inicial, para cada processo $p \in \mathcal{P}$, se o processo p está alocado, então gera uma nova configuração sem este processo. Caso o processo não esteja inicialmente alocado, então adiciona uma configuração válida contendo este processo, ou seja, caso o processo p entre em conflito com algum outro do serviço a qual pertence, não adiciona esta configuração inválida.

A Tabela 4.4 contém os resultados dos ensaios do modelo simplificado *CG1* com a pré geração nas instâncias **a1**. As instâncias **a1_1** e **a1_3** terminam com uma solução distinta da solução trivial. Todas as instâncias terminam com o melhor custo reduzido total melhor, isto é, absolutamente menor. Exceto na instância **a1_5**, todas terminam com uma quantidade maior de iterações.

Tabela 4.4: Resultado parcial por geração de colunas, CG1, com geração de colunas prévia

Instância	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
a1_1	44.306.501	89,364	CGRep	665	98	-0,255
a1_2	ST	21600,006	TCG	443	183	-63,891
a1_3	583.005.717	13689,631	CGRep	260	107	-16,195
a1_4	ST	21600,003	TCG	208	5	-1.508.971.879,349
a1_5	ST	21600,022	TCG	211	20	-907.900.503,482

Apesar de várias serem realizadas mais iterações, cada instância tem um tempo disponível menor para o ciclo de geração de colunas e para o MIP resultante, pois parte do tempo disponível é consumido com a geração prévia. A Tabela 4.5 contém algumas informações sobre o estado inicial de cada instância na etapa de geração de colunas, onde a *Tempo* contém o tempo em segundos utilizado para a criação dos modelos, incluindo a geração prévia. A coluna

Variáveis informa a quantidade de variáveis iniciais do mestre reduzido. A coluna *Razão* informa a razão entre as variáveis iniciais e um máximo teórico de $M(P + 1)$, pois para cada máquina, além da coluna inicial podem existir P outras variáveis iniciais. A coluna *Densidade Inicial* é a densidade, em valores percentuais do mestre reduzido neste ensaio. A coluna *Densidade Esperada* é a densidade percentual, caso todas as colunas mantivessem o a densidade inicial do ensaio de controle. A coluna *Variação* informa a razão da diferença entre as densidades, também em valores percentuais.

Na etapa de geração prévia, a instância **a1_3** apresenta o maior tempo para criar os modelos, com 1702,907 segundos. Em relação a variáveis, proporcionalmente a instância **a1_2** resulta no menor número de variáveis iniciais, enquanto a instância **a1_5** resulta no maior número. Estas mesmas instâncias resultam na maior e na menor variação de densidade respectivamente.

Tabela 4.5: Tempo de geração dos modelos, variáveis e densidade em CG1

Instância	Tempo	Variáveis	Razão (%)	Densidade Inicial (%)	Densidade Esperada (%)	Variação (%)
a1_1	0,549	233	57,67	26,221	25,00	4,8
a1_2	1364,548	19991	19,97	1,093	1,00	9,3
a1_3	1702,907	29268	29,23	1,058	1,00	5,8
a1_4	886,124	21428	42,81	2,103	2,00	5,1
a1_5	333,595	10091	84,00	8,392	8,33	0,7

4.16.1

Instância a1_1

Na instância **a1_1**, não há alteração no objetivo ao longo das iterações, com exibido no gráfico da Figura 4.46. Nesta instância a solução ótima consiste na troca de máquina de um único processo, ou seja, a geração prévia é capaz de produzir as colunas da solução. Também pode-se observar que variação da densidade é menor, se comparado com o ensaio de controle.

Em relação à tempo por iteração, quase todas apresentam um tempo total inferior a 0,5 segundos. Somente nas últimas iterações ocorre um aumento no tempo necessário por iteração, como pode ser visto no gráfico da Figura 4.47. Em relação à quantidade de iterações do resolvidor no problema mestre, o problema não utiliza mais de que 200 iterações. Nos últimos ciclos, a quantidade de iterações reduz a quase 0, comparando com o ensaio de controle tanto os tempos, quanto as iterações deste ensaio apresentaram de forma geral melhores resultados.

A evolução do custo reduzido total, apresentado no gráfico da Figura 4.48, tem oscilações menores. Apesar deste ensaio utilizar uma quantidade menor de iterações, a quantidade de melhorias é quase a mesma: 98 neste ensaio e 100 no controle.

Para a instância **a1_1** a geração prévia de colunas gera a solução ótima conhecida em um espaço de tempo menor.

4.16.2

Instância **a1_2**

Para a instância **a1_2**, o objetivo começa a reduzir logo nas primeiras iterações, conforme gráfico da Figura 4.49. Em relação à densidade, que não é superior a 1,35%, o aumento da densidade neste ensaio é menor quando comparado ao ensaio de controle.

Os tempo por iteração estão apresentados no gráfico da Figura 4.50, onde se nota que o problema mestre foi a principal parcela de tempo despendido nas primeiras 250 iterações. Após a iteração 250, o tempo do problema mestre começa a reduzir, porém o tempo total não reduz da mesma forma, pois os tempos dos sub problemas auxiliares passam a compor a principal parcela. Comparado com o ensaio de controle, os tempos das 250 primeiras iterações se assemelham com os tempos das 50 últimas iterações do controle. Em relação à quantidade de iterações do resolvedor, pode-se notar que para este ensaio a quantidade máxima de iterações utilizada é em torno da metade da quantidade máxima do ensaio de controle.

Quanto a convergência do custo reduzido total, exposto no gráfico da Figura 4.51, há uma redução na ordem de 7 ordens de grandeza, com valor inicial na ordem de 10^9 e terminando com ordem de 10^2 . Observa-se também que apresentam variações no valor de grande amplitude, na ordem de 3 ordens de grandeza. Em relação à quantidade de melhorias, em média ocorre uma melhoria a cada 2.4 iterações. Comparando com o ensaio de controle, a convergência este ensaio é melhor do que a convergência do controle.

Com base nos resultados deste ensaio, para a instância **a1_2** a geração prévia de colunas produz resultados melhores, apesar da etapa de geração de colunas ser interrompida por limite de tempo de execução.

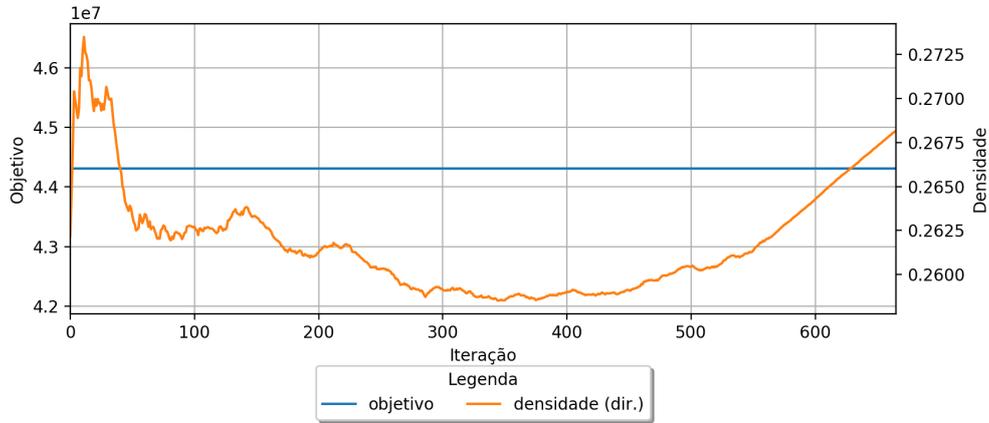


Figura 4.46: Objetivo e densidade de CG1, a1_1, com geração de colunas prévia

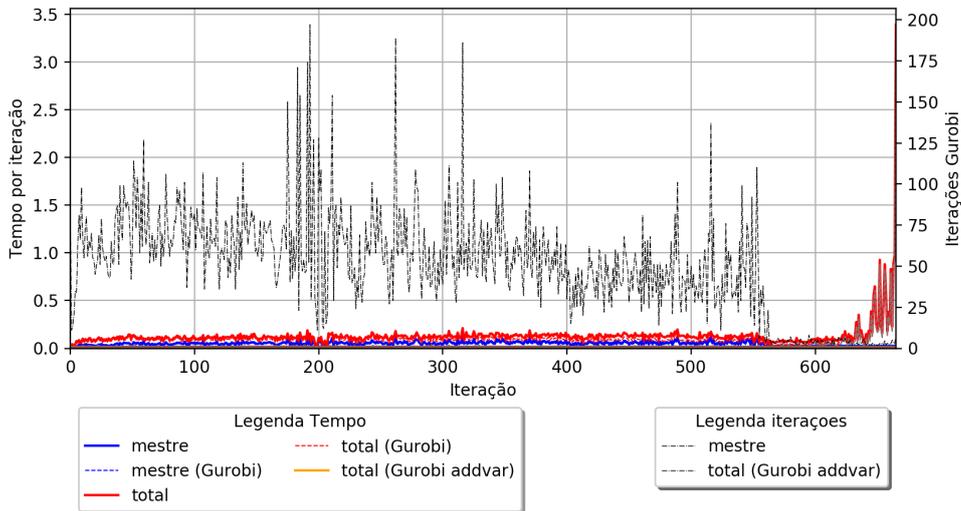


Figura 4.47: Tempos e iterações de CG1, a1_1, com geração de colunas prévia

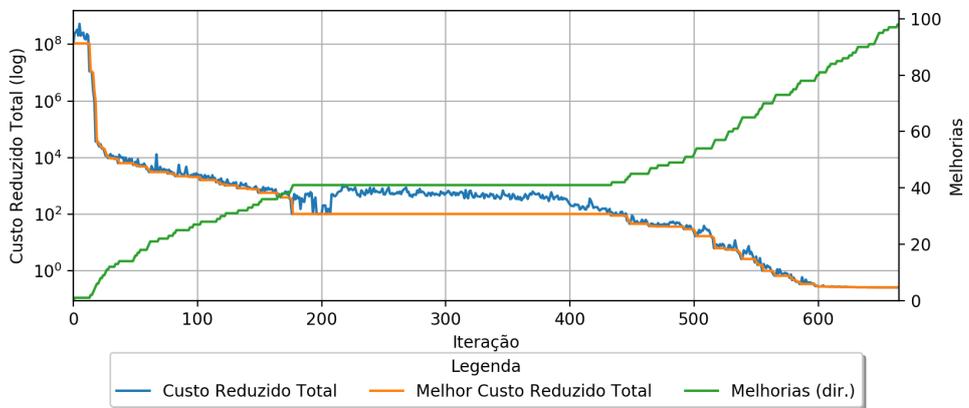


Figura 4.48: Custo reduzido de CG1, a1_1, com geração de colunas prévia

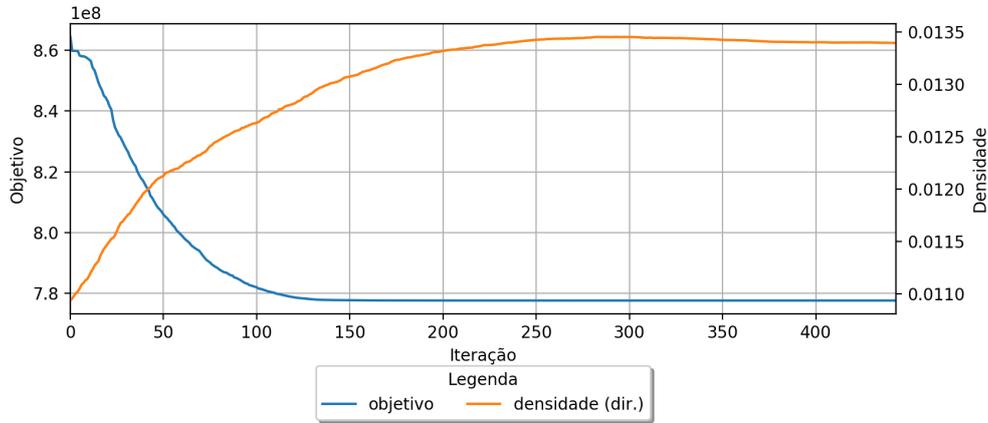


Figura 4.49: Objetivo e densidade de CG1, a1_2, com geração de colunas prévia

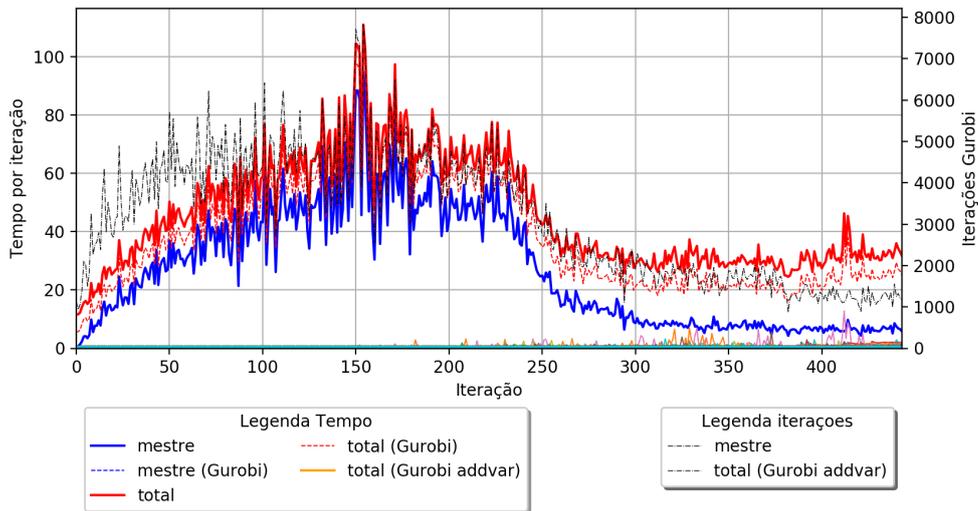


Figura 4.50: Tempos e iterações de CG1, a1_2, com geração de colunas prévia

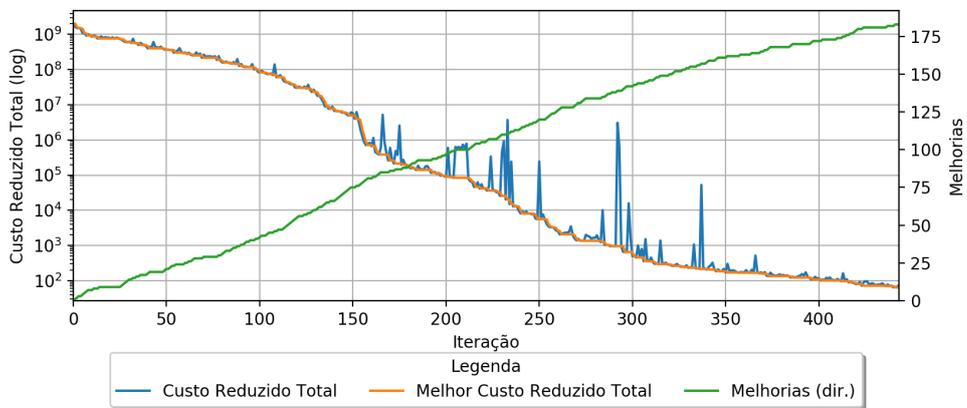


Figura 4.51: Custo reduzido de CG1, a1_2, com geração de colunas prévia

4.16.3

Instância a1_3

No caso da instância a1_3, o valor da função objetivo também começa a reduzir logo nas primeiras iterações, conforme gráfico da Figura 4.52. Em relação à densidade, que não é superior a 1,09%, o aumento da densidade neste ensaio é menor quando comparado ao ensaio de controle.

Em relação aos tempos de cada iteração, expostos no gráfico da Figura 4.53, pode-se perceber que o tempo necessário para resolver o problema mestre tem pouca influência no tempo total e os tempos dos problemas auxiliares se tornam a principal parcela de tempo de cada iteração. Nota-se também que há iterações que algum problema auxiliar requer mais tempo do que os demais. Em relação à quantidades de iterações do resolvedor, nas primeiras iterações há muitas iterações que convergem para 0 rapidamente. Comparando com o ensaio de controle, o tempo máximo de iteração deste ensaio é na ordem de 1/3 do máximo do ensaio de controle: 70 segundos neste ensaio contra 240 no controle.

A evolução do custo reduzido total, apresentado no gráfico da Figura 4.54, inicia com um valor na ordem de 10^9 e converge para um valor na ordem de 10^1 . Pode-se notar que o valor do CRT pode apresentar uma variação de amplitude de 5 ordens de grandeza. Em relação as melhorias, em média ocorre uma melhoria a cada 2,4 iterações.

Com base nos resultados deste ensaio, para a instância a1_3 a geração prévia de colunas produziu resultados melhores, de forma de retorna uma solução em tempo hábil.

4.16.4

Instância a1_4

O valor do objetivo da instância a1_4 começa a reduzir logo nas primeiras iterações, conforme gráfico da Figura 4.55. Comparando com o ensaio de controle, nota-se que o valor inicial deste ensaio é inferior. Observa-se também que há reduções no valor em várias iterações. Em relação à densidade, há um aumento na densidade, terminando em 3,4%. Comparando com o ensaio anterior, a densidade deste ensaio é em torno de 50% da densidade do ensaio de controle.

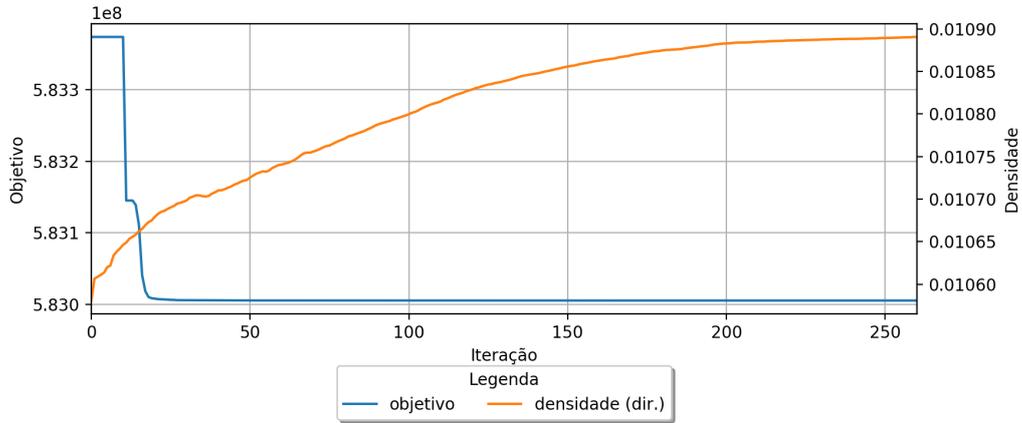


Figura 4.52: Objetivo e densidade de CG1, a1_3, com geração de colunas prévia

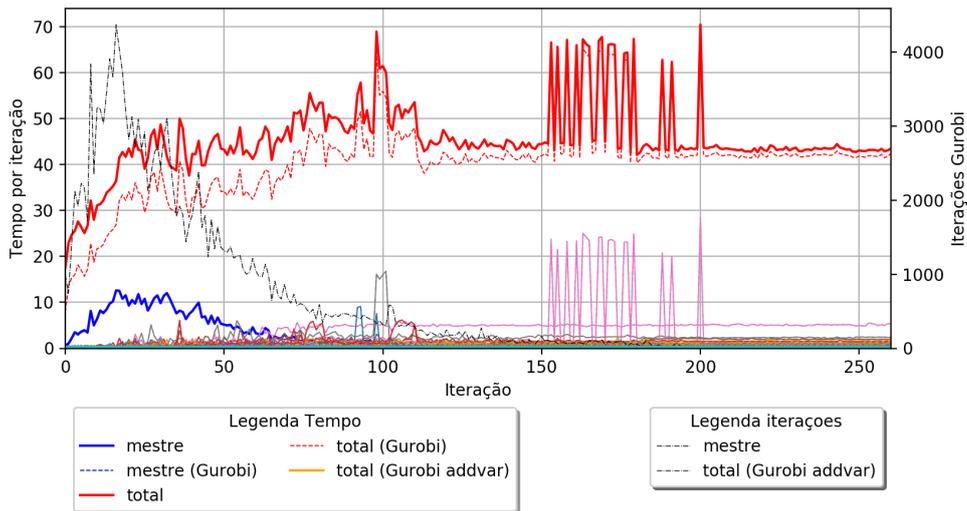


Figura 4.53: Tempos e iterações de CG1, a1_3, com geração de colunas prévia

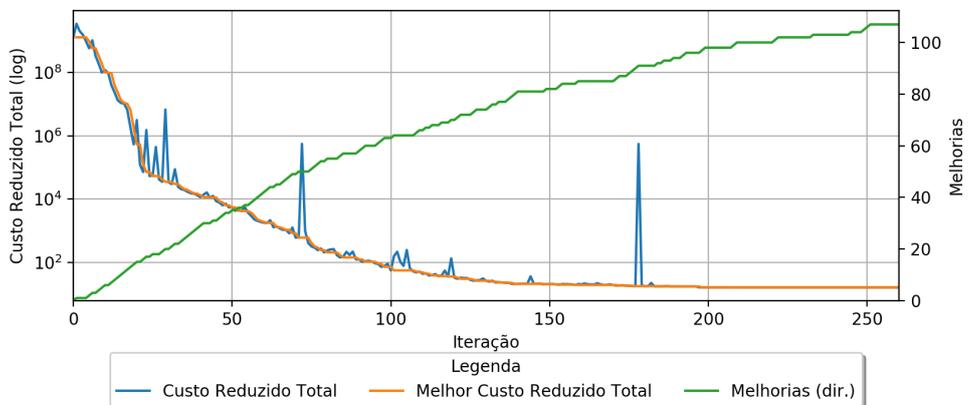


Figura 4.54: Custo reduzido de CG1, a1_3, com geração de colunas prévia

Quanto aos tempos de cada iteração, dispostos no gráfico da Figura 4.56, o tempo de resolução do problema mestre é a principal parcela de tempo da iteração. Comparando com o ensaio de controle, os tempos de resolução são semelhantes, assim como as iterações do resolvidor no problema mestre.

Pelo gráfico da Figura 4.57, nota-se que o custo reduzido total apresenta variações de amplitude na mesma ordem de grandeza. Comparando com o ensaio de controle, observa-se que o valor inicial do custo reduzido é uma ordem de grandeza menor neste ensaio.

Com base nos resultados deste ensaio, para a instância **a1_4** a geração prévia de colunas resultou em resultados melhores, apesar do valor final do custo terminar com uma amplitude de 10^9 .

4.16.5

Instância **a1_5**

Para a última instância **a1_5**, não ocorre alteração no valor do objetivo, como pode ser visto no gráfico da Figura 4.58. Em relação à densidade, esta aumenta para 10,5%. Comparando com o ensaio de controle, pode perceber que o valor inicial do objetivo é inferior, assim como a densidade final também é menor.

Em relação à tempo por ciclo de geração de colunas e iterações no problema mestre, dispostas no gráfico da Figura 4.59, pode-se notar que não há mudanças significativas quando comparado com o ensaio de controle.

A evolução do custo reduzido total é exibido no gráfico da Figura 4.60. Pode-se notar que a variação de amplitude permanece na mesma ordem de grandeza. Comparando com o ensaio de controle, pode-se perceber que o valor inicial do CRT neste ensaio é uma ordem de grandeza menor.

Com base nos resultados deste ensaio, para a instância **a1_5** a geração prévia de colunas resulta em resultados melhores, apesar do valor final do custo terminar com uma amplitude de 10^9 .

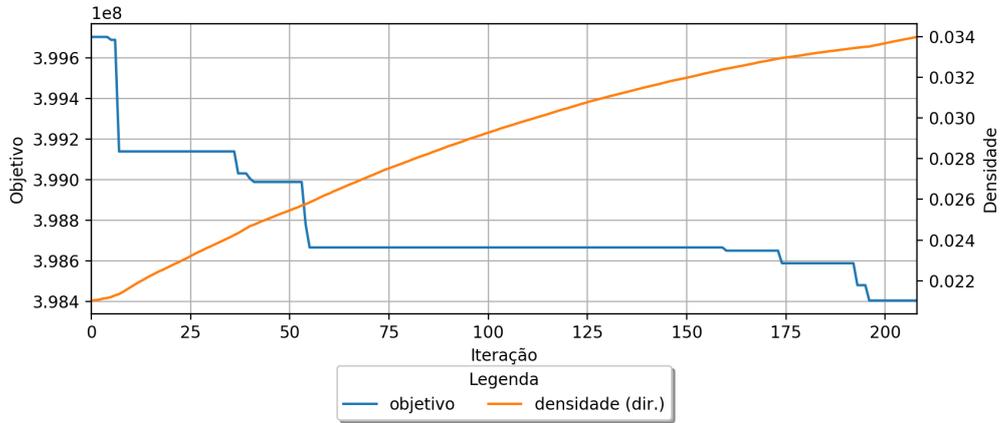


Figura 4.55: Objetivo e densidade de CG1, a1_4, com geração de colunas prévia

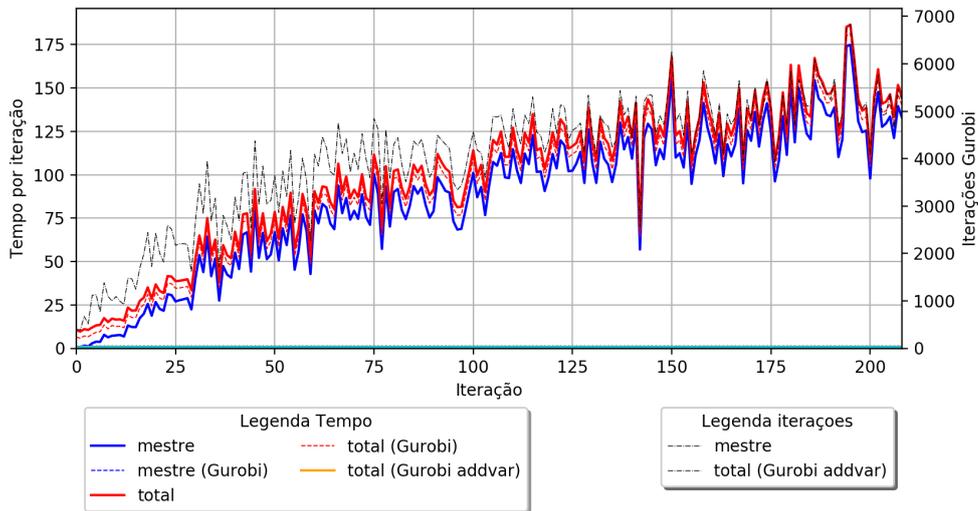


Figura 4.56: Tempos e iterações de CG1, a1_4, com geração de colunas prévia

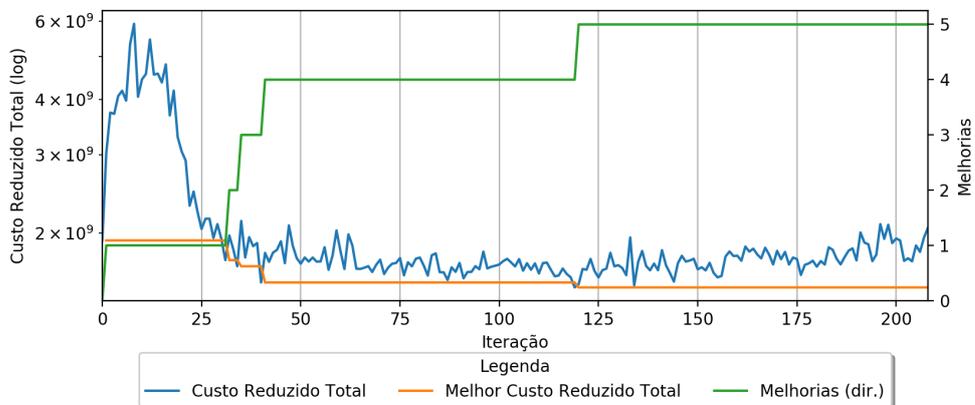


Figura 4.57: Custo reduzido de CG1, a1_4, com geração de colunas prévia

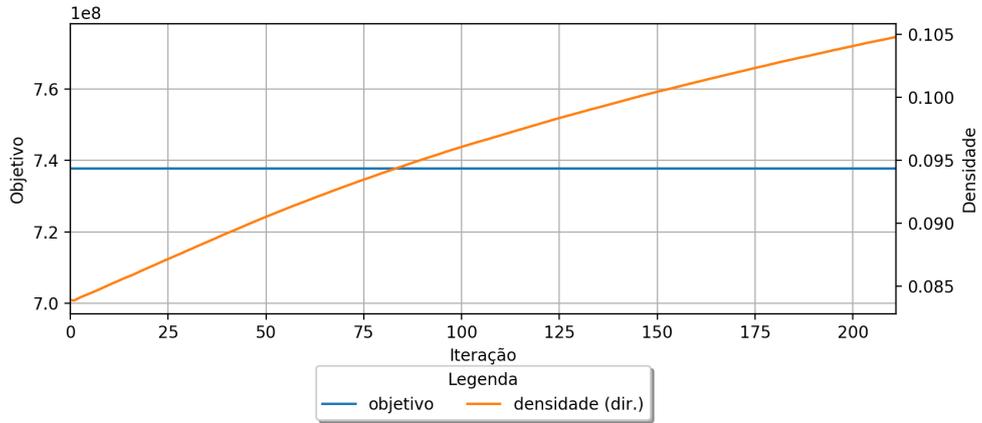


Figura 4.58: Objetivo e densidade de CG1, a1_5, com geração de colunas prévia

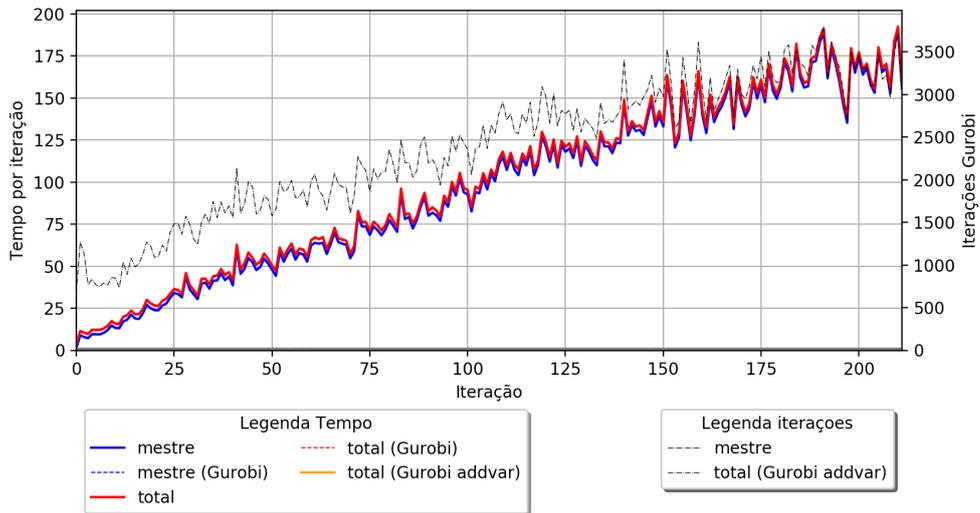


Figura 4.59: Tempos e iterações de CG1, a1_5, com geração de colunas prévia

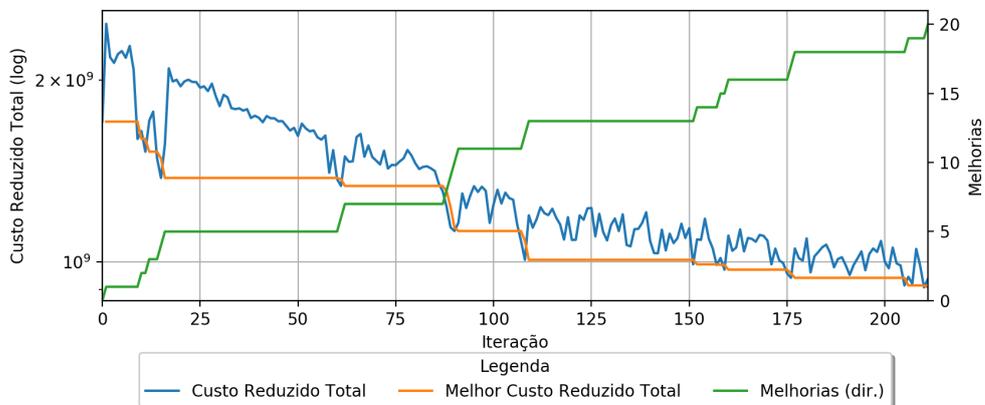


Figura 4.60: Custo reduzido de CG1, a1_5, com geração de colunas prévia

4.16.6

Comentários

Com base nos resultados deste ensaio, pode-se concluir que um geração inicial de colunas produz ganhos em todas as instâncias. Para as instâncias a1_2 e a1_3 os benefícios são mais evidentes, pois o custo reduzido total final está mais próximo do critério de parada do que nos ensaios anteriores. Para a instância a1_3, em especial, a geração prévia foi capaz de produzir uma solução distinta da solução trivial. Para as instâncias a1_4 e a1_5, a geração prévia possibilita em iniciar com um valor de objetivo menor, assim como um valor de custo reduzido total melhor.

4.17

Estabilização das Variáveis duais

Em virtude da grande amplitude na oscilação do custo reduzido total, busca-se usar a decomposição ponderada de Dantzig-Wolfe (Wentges, 1997, Pessoa et al., 2010) para reduzir a amplitude. Nesta decomposição ponderada, os valores das variáveis duais compõe uma média ponderada com um conjunto de duais quasi-fixo. Este ponto quasi-fixo é alterado sempre que o conjunto produzir um resultado melhor. Esta ponderação não altera o problema mestre adicionando variáveis ou restrições, como por exemplo o método da caixa de (Pigattu et al., 2005), que adiciona grupos de variáveis adicionais ao problema principal para estabilizar cada dual, por meio de funções lineares por parte. Também não adiciona novas restrições.

Seja ω^b o valor do melhor custo reduzido total de uma iteração. Seja também β^b as duais que resultaram nesse custo reduzido ω^b . Para cada iteração, o problema mestre produz as duais com valores β . Com os valores β e β^b , pode-se deduzir valores das duais estabilizadas β^{st} por meio de:

$$\beta^{st} = \beta^b \alpha + (1 - \alpha) \beta \quad (4-46)$$

Sendo α um valor escalar na faixa $0 \leq \alpha < 1$.

Os valores β^{st} são então utilizados nos problemas auxiliares de precifica-

ção. O custo reduzido total da iteração resulta em um valor ω . Em um problema de minimização, caso este novo valor for maior que ω^b , ou seja ocorre uma melhoria, então os valor ω passar ser o melhor custo reduzido total ω^b e as duais β^{st} são os novos valores das duas β^b .

Para valores iniciais de ω^b e β^b são adotados $-\infty$ e 0 respectivamente, pois qualquer ω resultante na primeira iteração é sempre maior que ω^b .

Vale notar que usar uma estabilização com $\alpha = 0$ produz os mesmos resultados de uma execução sem estabilização, pois os valores das variáveis duais utilizados nos problemas auxiliares são iguais aos valores das variáveis duais fornecidas pelo problema mestre.

Com base no resultados dos ensaios anteriores, observa-se que para algumas instâncias a oscilação do custo pode apresentar uma grande amplitude: uma variação em ordens de grandeza. Com o objetivo de reduzir as grandes variações, usa-se um valor de α próximo de 1. Porém deve ainda assim permitir variações. O objetivo da estabilização é reduzir a amplitude de oscilação e não anular. A variação é necessária para a convergência. Tendo em mente o requisito de permitir uma variação, porém reduzindo a amplitude da mesma, foram realizados ensaios iniciais com α constante de 0,9. Esta escolha resulta que as duais fornecidas pelo problema mestre é reduzido em uma ordem de grandeza, pois $(1 - \alpha) = 0,1$.

A Tabela 4.6 contém o resultado das execuções do modelo simplificado *CG1* usando a estabilização com α constante de 0,9. Este ensaio resulta que todas as instâncias terminassem na solução trivial, porém o melhor custo reduzido total final das instâncias **a1_2** e **a1_3** terminam próximo de zero, quando comparados as valores iniciais.

Tabela 4.6: Resultado parcial por geração de colunas, *CG1*, com estabilização constante=0,9

Instância	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
a1_1	ST	0,158	CGRep	2	2	-177.929.165,200
a1_2	ST	21600,106	TCG	544	279	-4,965
a1_3	ST	21622,589	TCG	407	280	-99,181
a1_4	ST	21600,029	TCG	210	37	-2.765.695.632,598
a1_5	ST	21600,006	TCG	338	69	-1.533.807.826,658

4.17.1

Instância a1_1

Na instância a1_1 a estabilização conduz a uma repetição de colunas geradas nas primeiras iterações, resultando em uma abordagem pior, quando comparadas aos resultados dos demais ensaios. A Tabela 4.7 contém resultados de ensaios usando vários parâmetros para α . Observa-se uma certa instabilidade, pois para $\alpha = 0,20$ e $\alpha = 0,30$ são resolvidos porém para $\alpha = 0,25$ a execução é terminada por repetição de colunas em algumas iterações. A mesma falta de consistência ocorre para $\alpha = 0,70$ e $\alpha = 0,80$, onde são terminados por repetição de colunas com poucas iterações, porém resolve para $\alpha = 0,75$. Uma possível hipótese para essa instabilidade pode estar relacionada com a pouca quantidade de variáveis iniciais.

O uso de uma estabilização constante de 0,9 não é benéfico para a instância a1_1.

Tabela 4.7: Resultados de CG1, a1_1, com vários coeficientes de estabilização constante

coeficiente α	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
0,00	44.306.501	523,820	Concluído	2071	100	0.000
0,05	44.306.501	552,316	CGRep	2005	113	-4.847
0,10	44.306.501	353,180	CGRep	1632	97	-2,105
0,15	44.306.501	703,019	CGRep	2307	141	-5,082
0,20	44.306.501	753,156	CGRep	1536	117	-0,808
0,25	49.528.750	12,367	CGRep	119	10	-72.870.453,102
0,30	44.306.501	389,692	Concluído	1613	160	-0,091
0,35	44.306.501	351,305	CGRep	1665	176	-0,125
0,40	49.528.750	10,607	CGRep	108	8	-98.162.369,673
0,45	49.528.750	12,136	CGRep	119	8	-76.895.898,533
0,50	49.528.750	11,842	CGRep	117	16	-72.773.412,454
0,55	44.306.501	197,011	Concluído	1049	173	-0,083
0,60	44.306.501	184,217	CGRep	1079	139	-3,539
0,65	49.528.750	10,076	CGRep	106	15	-52.937.764,164
0,70	49.528.750	13,372	CGRep	132	40	-574.535,102
0,75	44.306.501	121,937	CGRep	845	149	-18,322
0,80	49.528.750	0,457	CGRep	12	2	-159.819.024,200
0,85	49.528.750	0,157	CGRep	2	2	-168.023.475,800
0,90	49.528.750	0,158	CGRep	2	2	-177.929.165,200

4.17.2

Instância a1_2

Para a instância a1_2, o gráfico da Figura 4.61 contém o valor do objetivo a cada iteração. Pode-se observar que há uma redução significativa no valor. Em relação à densidade, pode-se notar que esta atinge um valor na ordem de

2,75% e em seguida reduz para um valor menor que 2%. Comparando com o ensaio de controle, este ensaio começa a redução do valor do objetivo antes e a densidade na última iteração é menor.

Em relação aos tempos de execução das iterações, presente no gráfico da Figura 4.62, observa-se que nas primeiras 300 iterações o tempo do problema mestre é a principal parcela do total da iteração. Após a iteração 300, o tempo do mestre reduz e os problemas auxiliares passam a formar a principal parcela de tempo.

O custo reduzido, apresentado no gráfico da Figura 4.63, tem oscilação de amplitudes menores e redução frequentes. Há uma redução de um custo inicial de na ordem de 10^{10} para um valor na ordem de 10^1 . Portanto há uma redução muito mais significativa em comparação com o ensaio de controle.

Para a instância a1_2, a estabilização constante de 0,9 resulta em ganhos significativos.

4.17.3

Instância a1_3

Na instância a1_3, o objetivo, presente no gráfico da Figura 4.64, apresenta uma redução inicial significativa no valor. Quanto à densidade, atinge um valor na ordem de 2,4% e em seguida reduz para um valor menor que 1,8%. Em relação ao o ensaio de controle, este ensaio a densidade é menor, além de ocorrer a redução do objetivo.

Os tempos de execução das iterações, contido no gráfico da Figura 4.65, observa-se que nas primeiras 250 iterações o tempo do problema mestre é a principal parcela do total da iteração. Em seguida, o tempo do mestre reduz e os problemas auxiliares passam a formar a principal parcela de tempo.

Quanto ao custo reduzido, exibido no gráfico da Figura 4.66, ocorre oscilações de amplitudes menores. Comparando com o ensaio de controle, a estabilização permite reduções frequentes no valor, além de uma redução significativa de um valor inicial de 10^{10} para 10^2 .

Para a instância a1_3, a estabilização constante de 0,9 resulta em ganhos significativos

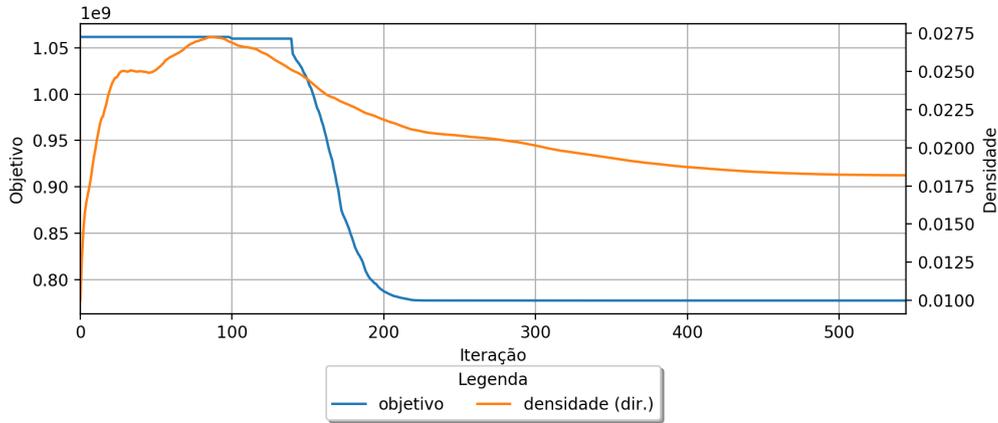


Figura 4.61: Objetivo e densidade de CG1, a1_2, com estabilização constante=0,9

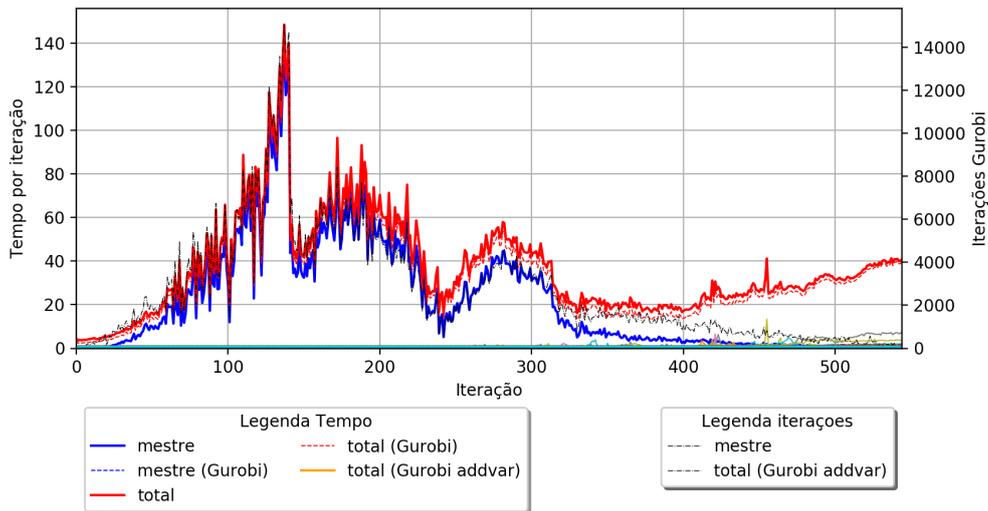


Figura 4.62: Tempos e iterações de CG1, a1_2, com estabilização constante=0,9

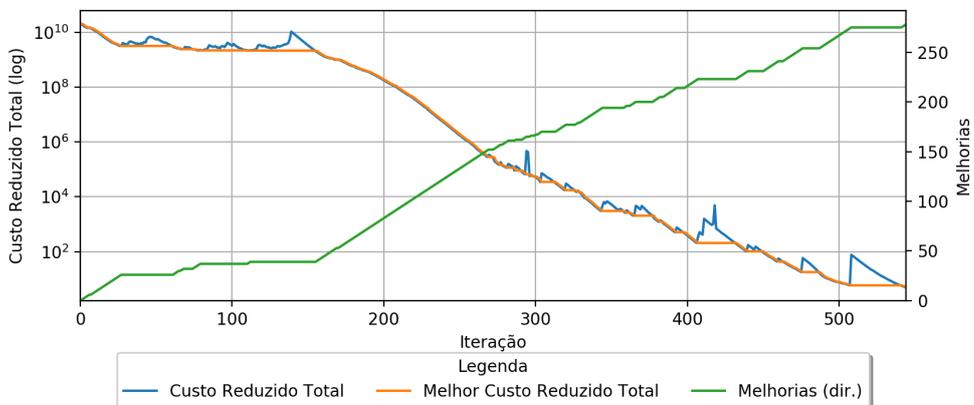


Figura 4.63: Custo reduzido de CG1, a1_2, com estabilização constante=0,9

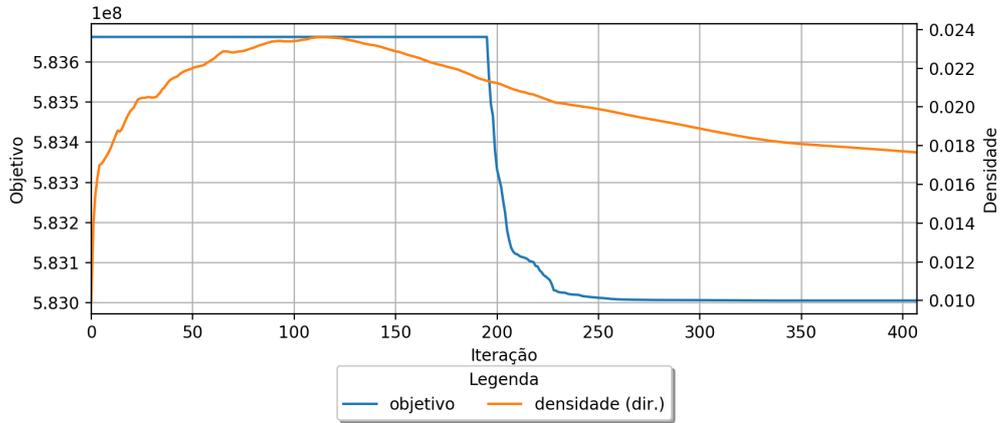


Figura 4.64: Objetivo e densidade de CG1, a1_3, com estabilização constante=0,9

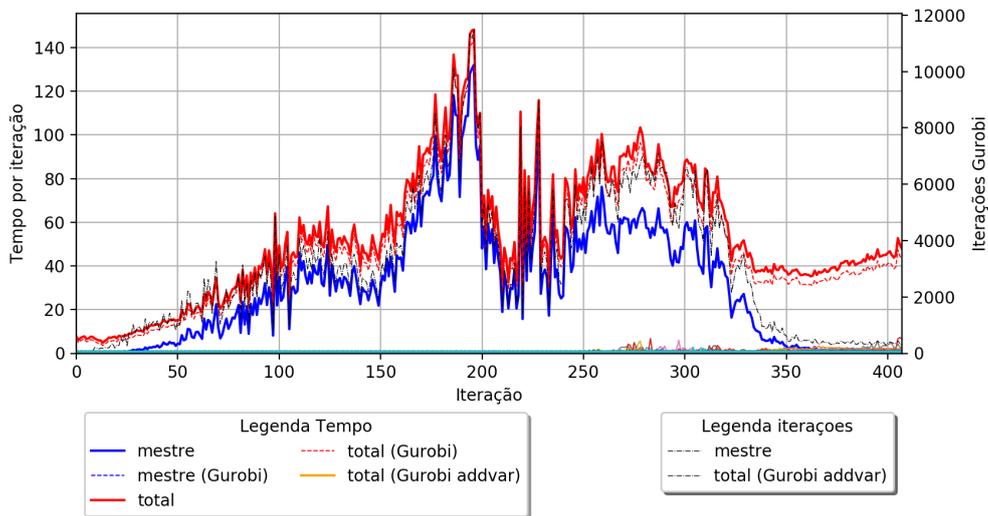


Figura 4.65: Tempos e iterações de CG1, a1_3, com estabilização constante=0,9

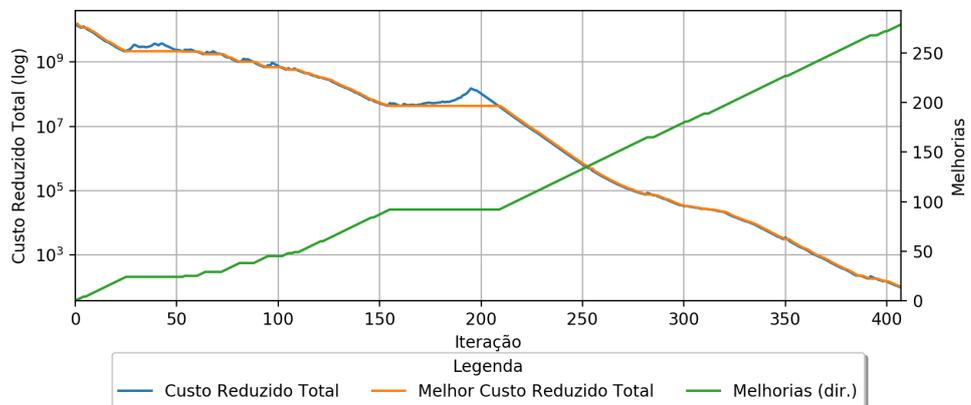


Figura 4.66: Custo reduzido de CG1, a1_3, com estabilização constante=0,9

4.17.4

Instância a1_4

Para a instância a1_4, não ocorre redução significativa no valor do objetivo, Figura 4.67, assim como no ensaio de controle. A estabilização resulta em um aumento menor na densidade, além de uma redução após algumas iterações. Quanto a tempo por iteração, exibido na Figura 4.68, pode-se perceber que há uma estabilização de tempo após a iteração 150.

O gráfico da Figura 4.69 contém a evolução do custo reduzido total, onde se nota que há uma redução significativa e constante nas primeiras 50 iterações. Após essa iteração, o valor do custo apresenta uma maior oscilação e poucas reduções. Apesar do custo reduzido ainda elevado, a estabilização permite uma redução maior.

Para a instância a1_4, a estabilização constante de 0,9 resulta em ganhos.

4.17.5

Instância a1_5

Na instância a1_5, com base no gráfico da Figura 4.70, observa-se que não há redução no valor do objetivo, da mesma forma que no ensaio de controle. Entretanto, a densidade do problema é menor, quando comparada ao mesmo ensaio de controle. Em relação aos tempos por iteração, Figura 4.71, estes são semelhantes aos tempos do ensaio de controle.

O custo reduzido total, presente no gráfico da Figura 4.72, a amplitude de oscilação é menor quando comparada ao ensaio de controle. Também pode-se perceber que após um conjuntos de colunas há uma redução significativa no valor do custo reduzido.

Para a instância a1_5, a estabilização constante de 0,9 resulta em ganhos.

4.17.6

Comentários

Com base nos resultados deste ensaio, pode-se perceber que a estabilização das variáveis duais por meio de um coeficiente constante de 0,9 resulta

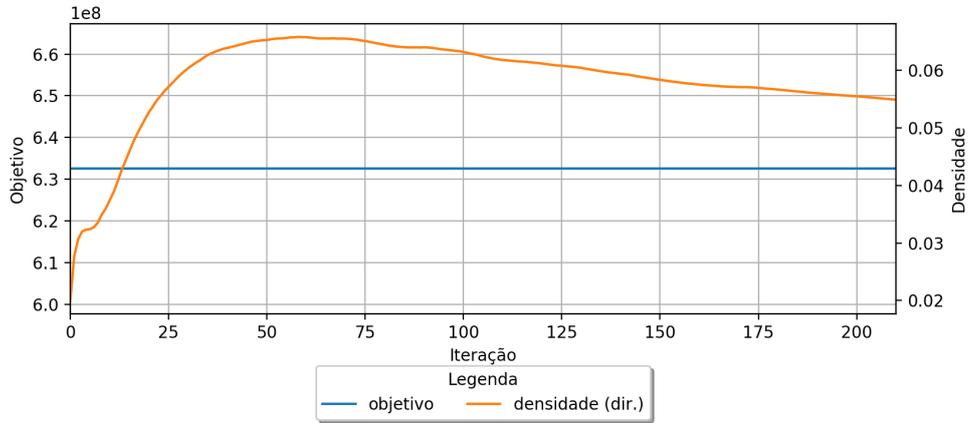


Figura 4.67: Objetivo e densidade de CG1, a1_4, com estabilização constante=0,9

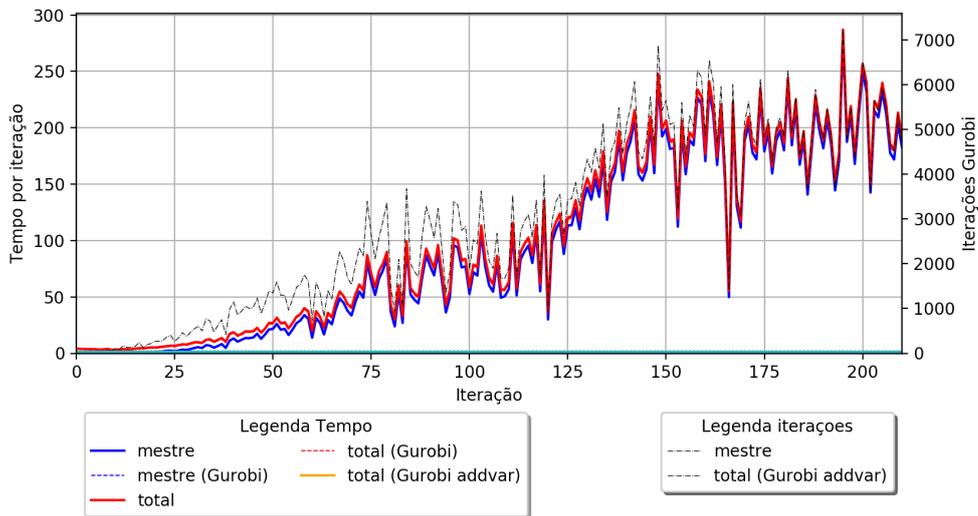


Figura 4.68: Tempos e iterações de CG1, a1_4, com estabilização constante=0,9

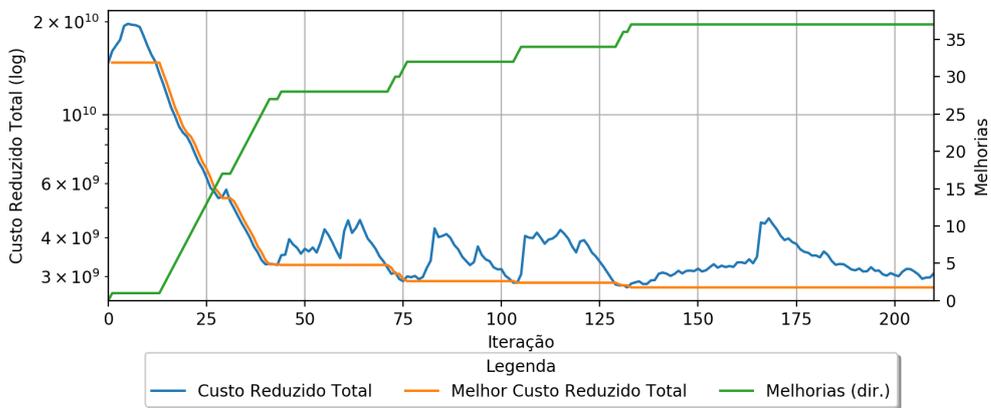


Figura 4.69: Custo reduzido de CG1, a1_4, com estabilização constante=0,9

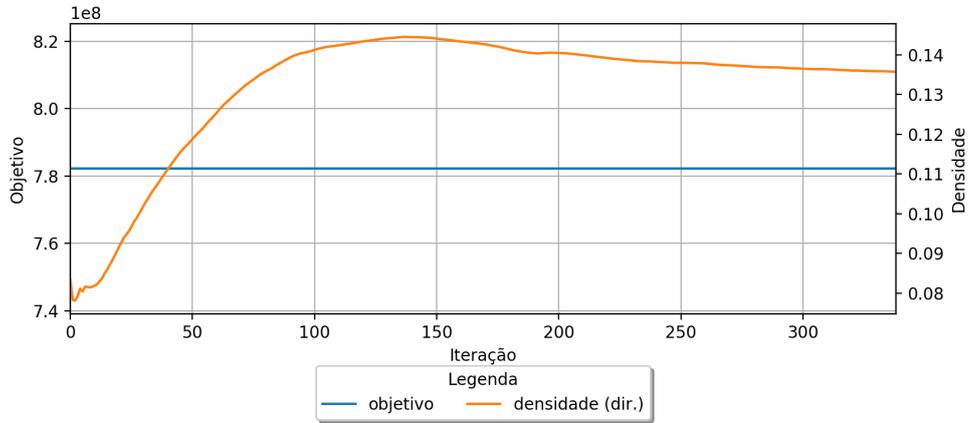


Figura 4.70: Objetivo e densidade de CG1, a1_5, com estabilização constante=0,9

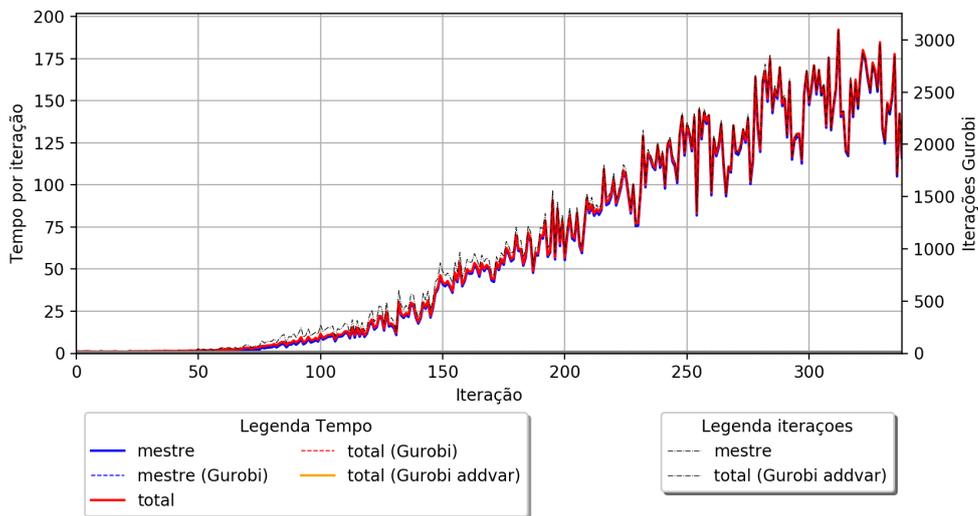


Figura 4.71: Tempos e iterações de CG1, a1_5, com estabilização constante=0,9

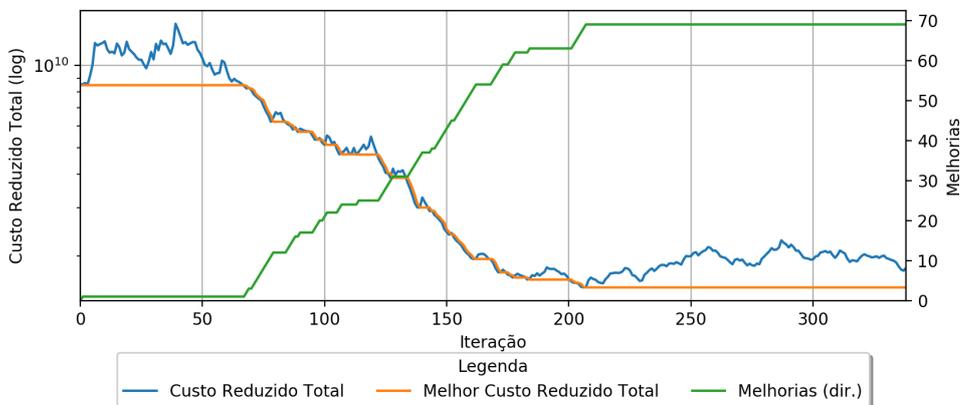


Figura 4.72: Custo reduzido de CG1, a1_5, com estabilização constante=0,9

em ganhos significativos em 2 das 5 instâncias, a1_2 e a1_3, onde a redução do custo reduzido é maior, terminando em valores próximo de zero, quando comparados com os valores iniciais. Para as instâncias a1_4 e a1_5 o ganho é menor, pois resulta em uma redução pequena no custo reduzido total final. Somente para a instância a1_1 a estabilização apresenta resultados piores, terminando por repetição de colunas nas iterações iniciais.

4.18

Ensaio combinados

Dos ensaios anteriores, tanto a geração prévia de colunas e quanto a estabilização apresentam ganhos para a maioria das instâncias. Alguns desses ganhos são significativos, enquanto outros são menores. A Tabela 4.8 contém os resultados de um ensaio combinando uma geração de colunas prévia com a estabilização constante de 0,9.

A combinação possibilita que 3 das 5 instâncias terminem a etapa de geração de colunas, sendo estas as instâncias a1_1, a1_2 e a1_3. Somente para a1_2 a execução termina durante o MIP final. Para as outras duas há conclusão do MIP final. Para as instâncias a1_4 e a1_5, o ensaio termina durante a etapa de geração de colunas.

Tabela 4.8: Resultado parcial por geração de colunas, CG1, com estabilização constante=0,9 e geração prévia de variáveis

Instância	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
a1_1	44.306.501	2,695	CGRep	33	27	-12.375.912,144
a1_2	881.988.071	21600,142	CGRep - TMIP	456	207	-64,441
a1_3	583.005.818	14544,467	CGRep	352	191	-59,473
a1_4	ST	21600,237	TCG	179	105	-137.863.104,649
a1_5	ST	21600,355	TCG	161	108	-140.133.763,740

4.18.1

Instância a1_1

Na instância a1_1 não há alteração no valor do objetivo, como exposto no gráfico da Figura 4.73. Porém como comentado anteriormente, a solução desta instância consiste em trocar um processo de máquina. Nota-se que a densidade do problema aumenta inicialmente e em seguida reduz. Comparando com o ensaio de controle, pode-se notar que a densidade deste ensaio varia menos.

Em relação à tempo por iteração, presente no gráfico da Figura 4.74, os problemas, de forma geral, resolvem rapidamente. Comparando com o ensaio de controle, o tempo total da iteração é maior, pois o tempo de cada problema auxiliar também é maior. Apesar do maior tempo total, são necessários um menor número de iterações para resolver.

Em relação ao custo reduzido, exibido no gráfico da Figura 4.75, há pouca redução no valor total, porém há reduções frequentes. Comparando com o controle, a redução total é menor. A falta de redução é compensada, para esta instância, com a geração prévia produzir a resposta final.

Para a instância **a1_1**, o uso combinado de geração prévia e estabilização resulta em resultados positivos.

4.18.2

Instância a1_2

Para a instância **a1_2**, o valor do objetivo reduz desde as primeiras iterações, como pode ser visto no gráfico da Figura 4.76. Após a iteração 100 não há redução significativa. Em relação densidade, observa-se que esta é inferior a 1,25%. Comparando com ensaio de controle, nota-se que tanto o valor do objetivo quanto a densidade é menor.

Em relação à tempo por iteração, pelo gráfico da Figura 4.77, nas primeiras 200 iterações o problema mestre é a principal parcela de tempo total da iteração. Após a iteração 200, o tempo do problema mestre reduz e os tempos dos problema auxiliares se tornam a principal componente do tempo total. Comparando com o ensaio de controle, o tempo total é inferior aos tempos do controle.

A evolução do custo reduzido total é exposto no gráfico da Figura 4.78, onde nota-se uma grande redução nas primeiras 300 iterações. Após a iteração 300 não há redução no custo total observa-se um aumento na variação.

Para a instância **a1_2**, o uso combinado de geração prévia e estabilização resulta em resultados positivos.

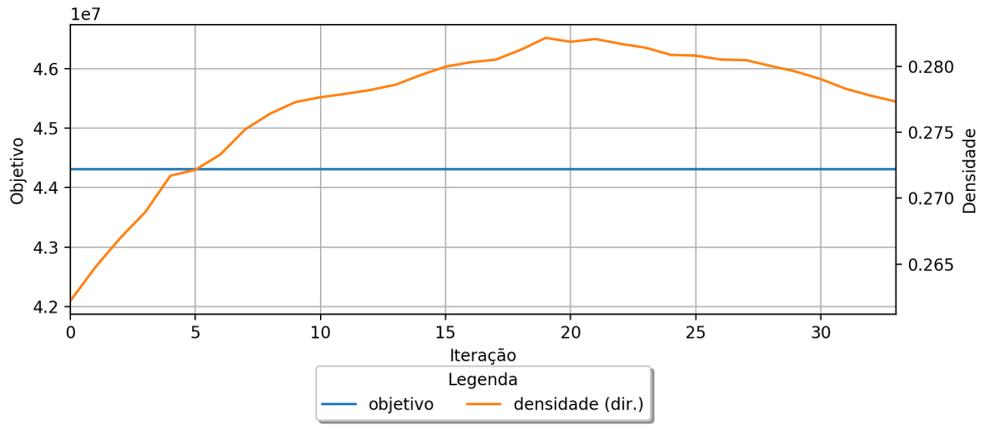


Figura 4.73: Objetivo e densidade de CG1, a1_1, com estabilização constante=0,9 e pré geração de colunas

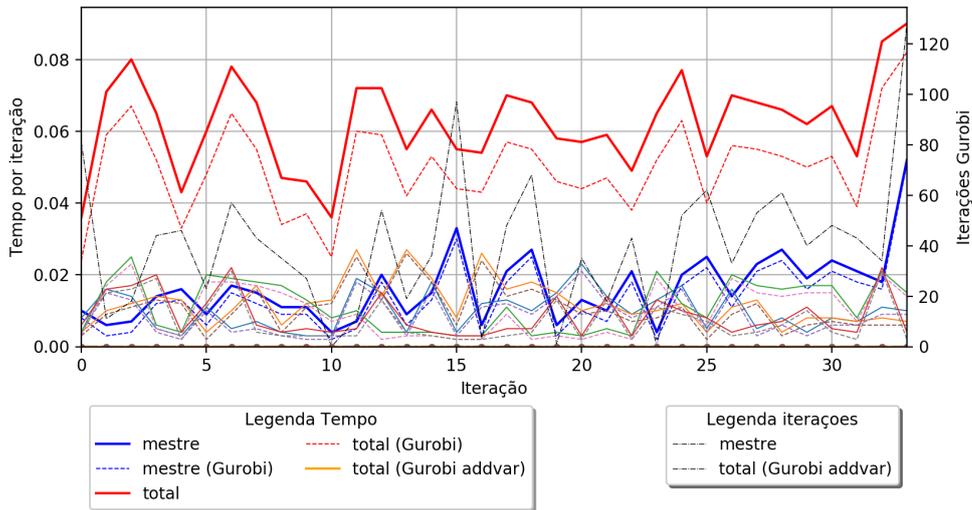


Figura 4.74: Tempos e iterações de CG1, a1_1, com estabilização constante=0,9 e pré geração de colunas

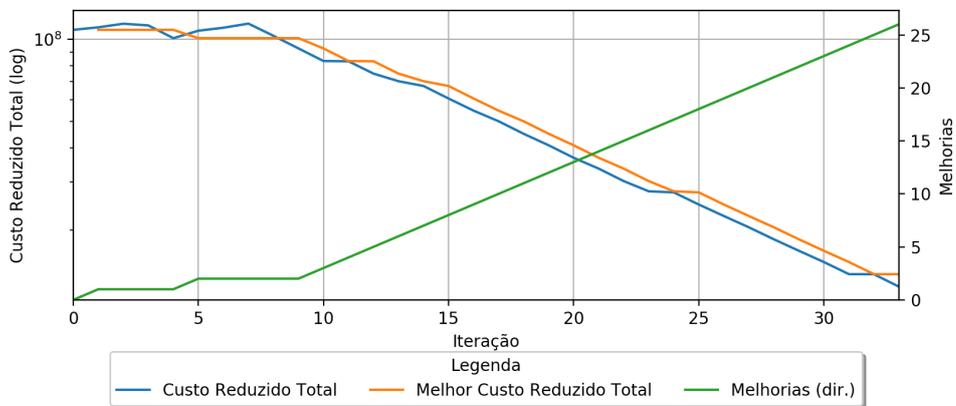


Figura 4.75: Custo reduzido de CG1, a1_1, com estabilização constante=0,9 e pré geração de colunas

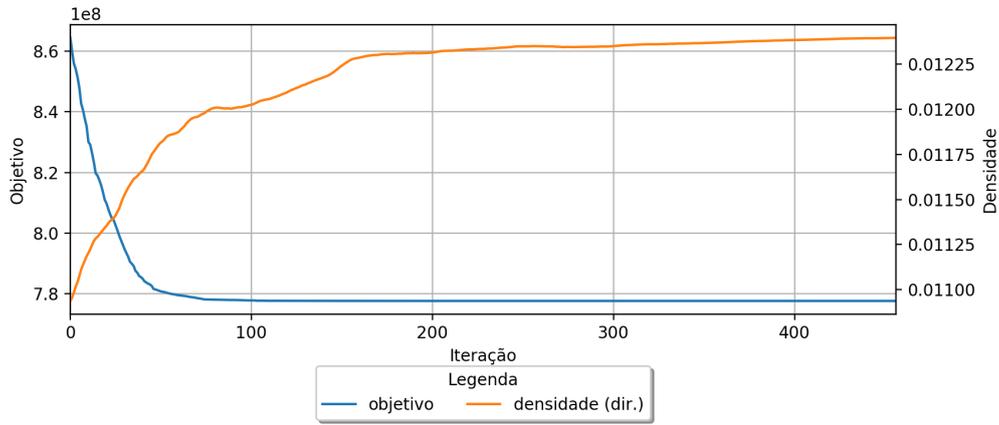


Figura 4.76: Objetivo e densidade de CG1, a1_2, com estabilização constante=0,9 e pré geração de colunas

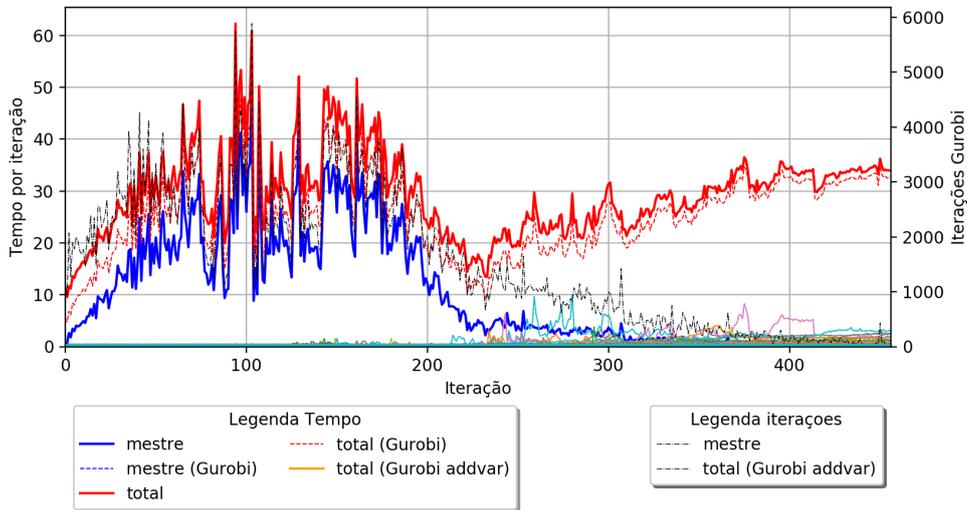


Figura 4.77: Tempos e iterações de CG1, a1_2, com estabilização constante=0,9 e pré geração de colunas

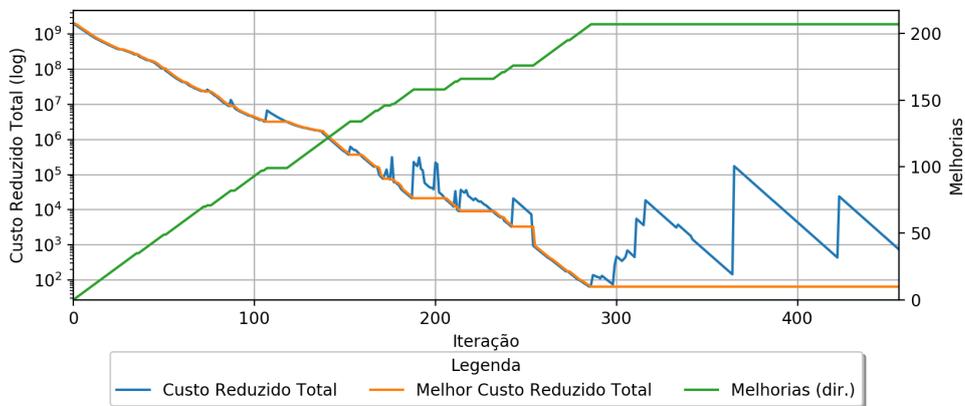


Figura 4.78: Custo reduzido de CG1, a1_2, com estabilização constante=0,9 e pré geração de colunas

4.18.3

Instância a1_3

Na instância a1_3, a evolução do objetivo é exposta no gráfico da Figura 4.79, onde observa-se uma redução significativa nas iterações iniciais. Em relação à densidade, há um pequeno aumento desde o valor inicial, terminando em um valor menor que 1,1%. Comparando com o ensaio de controle, este ensaio possui redução do objetivo e a densidade final é menor.

Em relação à tempo por iteração, apresentado no gráfico da Figura 4.80, nota-se que o tempo do problema principal não é a principal parcela de tempo. Também pode observar que em algumas iterações alguns problemas auxiliares requerem mais tempo de resolução do que os demais. Comparando com o ensaio de controle, o tempo máximo deste ensaio é na ordem da metade do tempo máximo do controle.

O gráfico da Figura 4.81 apresenta a evolução do custo reduzido total. Pode-se notar que há uma redução de várias ordens de grandeza no valor. Nota-se também que ocorre um aumento repentino em algumas iterações. Comparado ao ensaio de controle, a redução do custo reduzido é maior.

Para a instância a1_3, o uso combinado de geração prévia e estabilização resulta em resultados positivos.

4.18.4

Instância a1_4

Para a instância a1_4, a evolução do objetivo é exibido no gráfico da Figura 4.82, onde observa-se redução do valor ao longo das iterações. Em relação à densidade, nota-se que há um aumento de menos de 0,5%. Comparando com o ensaio de controle, o valor inicial do objetivo é menor e o aumento da densidade também é menor.

O gráfico da Figura 4.83 contém os tempos por iteração. Nota-se que o tempo de resolução do problema principal é a principal parcela de tempo do total. Comparando com o ensaio de controle, os tempos são semelhantes.

Quanto ao custo reduzido total por iteração, presente no gráfico da Figura 4.84, há uma redução total na ordem de uma ordem de grandeza, porém

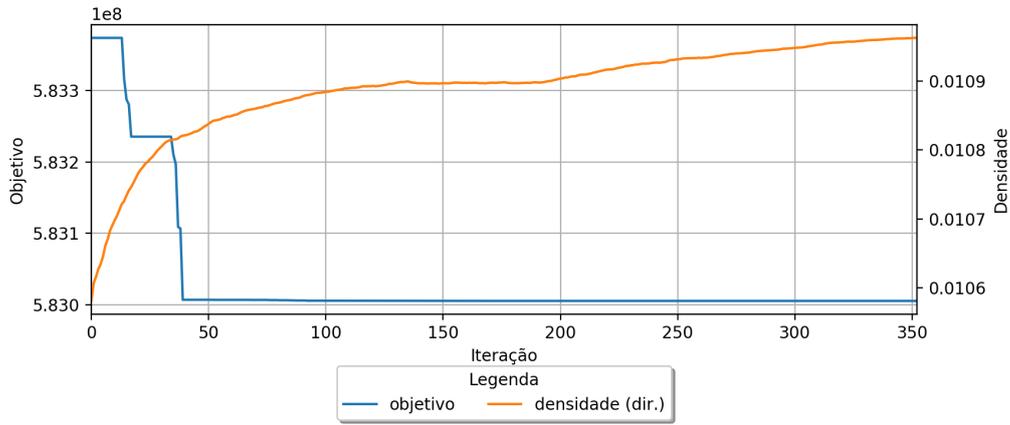


Figura 4.79: Objetivo e densidade de CG1, a1_3, com estabilização constante=0,9 e pré geração de colunas

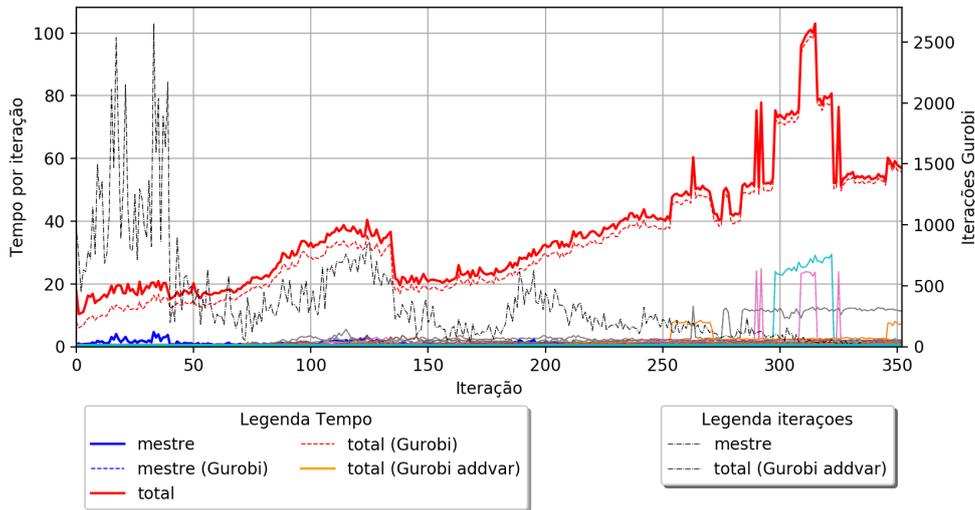


Figura 4.80: Tempos e iterações de CG1, a1_3, com estabilização constante=0,9 e pré geração de colunas

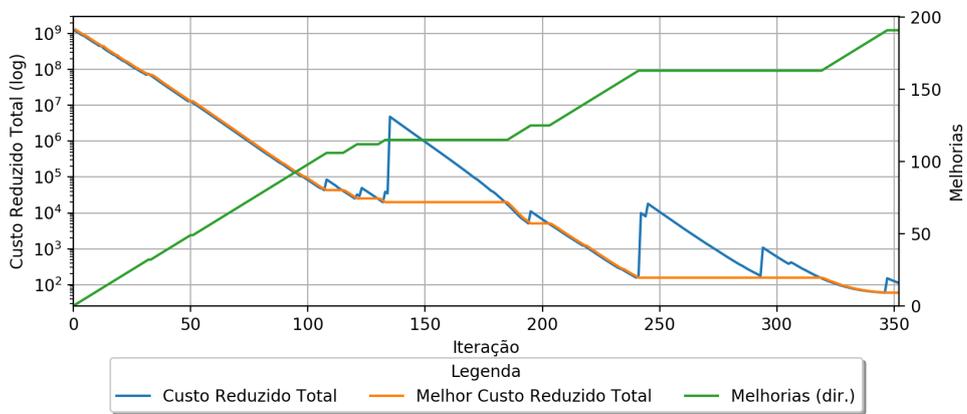


Figura 4.81: Custo reduzido de CG1, a1_3, com estabilização constante=0,9 e pré geração de colunas

são reduções frequentes. Comparando com o ensaio de controle, ambos reduzem uma ordem de grandeza, porém neste ensaio o custo reduzido total inicial é menor.

Para a instância **a1_4**, o uso combinado de geração prévia e estabilização resulta em resultados positivos.

4.18.5

Instância **a1_5**

Para última instância **a1_5**, o gráfico da Figura 4.85 expõe a evolução do objetivo, onde há uma redução no valor após a iteração 60. Em relação à densidade, a variação total é menor que 0,5%. Comparando com o ensaio de controle, para esta instância o valor inicial do objetivo é menor, além de apresentar uma redução. A densidade neste ensaio é na ordem da metade da densidade final do ensaio de controle.

Os tempos das iterações são apresentados no gráfico da Figura 4.86, onde nota-se que o problema principal detém a maior parcela de tempo do total da iteração. Comparando com o ensaio de controle, pode-se perceber que os tempos de resolução nas iterações finais é na ordem do dobro dos tempos finais do ensaio de controle.

A evolução do custo reduzido total está presente no gráfico da Figura 4.87, onde nota-se reduções constantes no valor do custo reduzido total. Comparando com o ensaio de controle, pode-se perceber que o valor inicial deste ensaio é menor do que o valor inicial do controle.

Para a instância **a1_5**, o uso combinado de geração prévia e estabilização resulta em resultados positivos, apesar do tempo de resolução mais alto.

4.18.6

Comentários

Com base nos resultados deste ensaio, pode-se perceber que uma geração de colunas prévia combinada com a estabilização das variáveis duais por meio de um coeficiente constante de 0,9 resulta em ganhos significativos em 3 das 5 instâncias, **a1_1**, **a1_2** e **a1_3**, pois há conclusão da etapa de geração de colunas. Para as instâncias **a1_1** e **a1_3** há término do MIP final. Para as

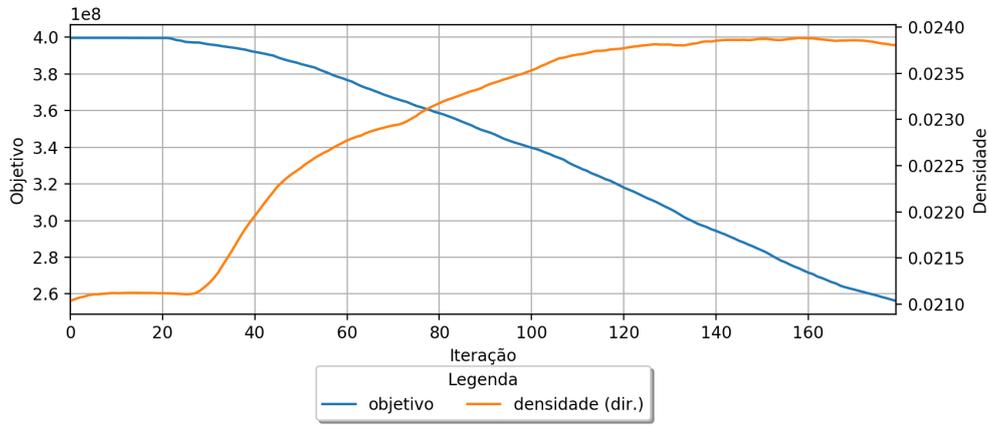


Figura 4.82: Objetivo e densidade de CG1, a1_4, com estabilização constante=0,9 e pré geração de colunas

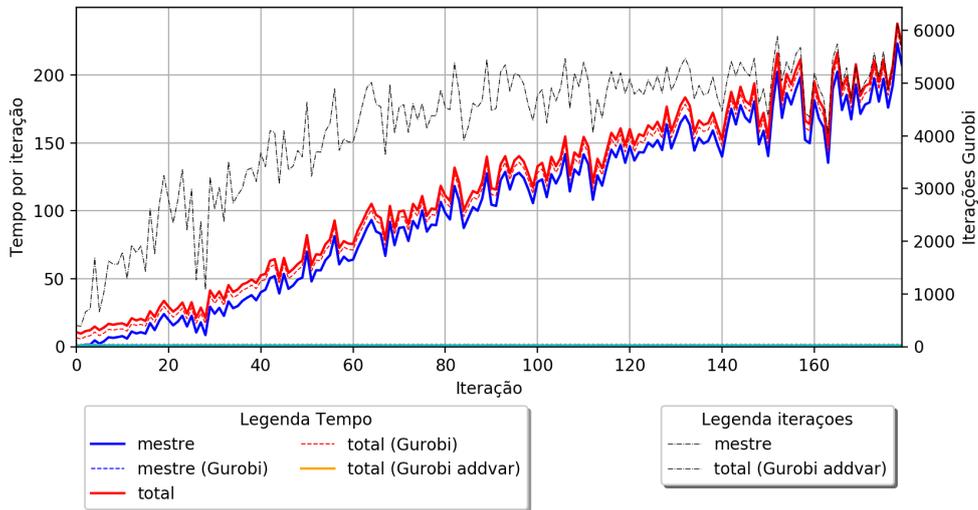


Figura 4.83: Tempos e iterações de CG1, a1_4, com estabilização constante=0,9 e pré geração de colunas

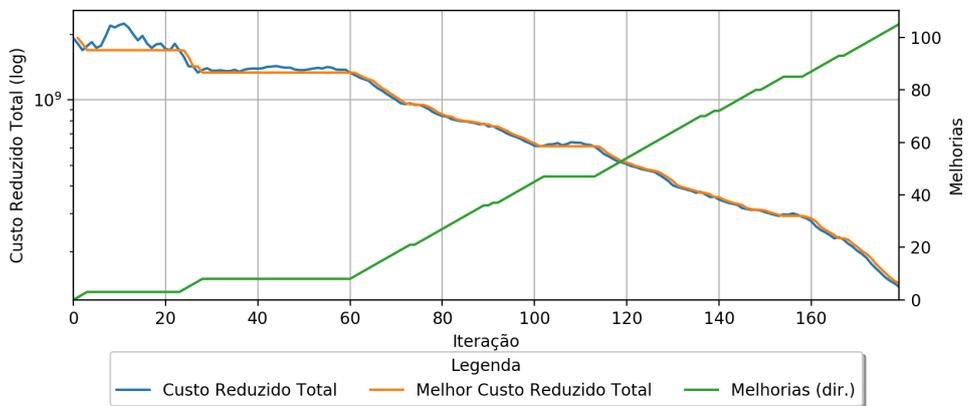


Figura 4.84: Custo reduzido de CG1, a1_4, com estabilização constante=0,9 e pré geração de colunas

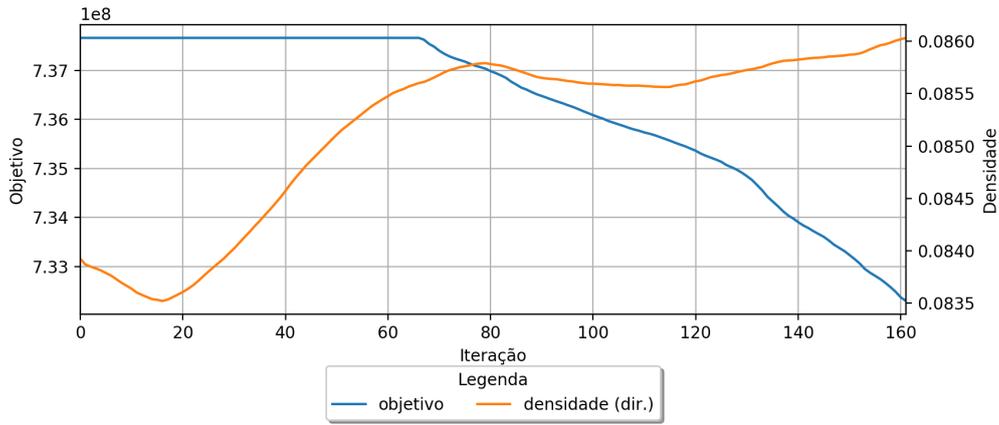


Figura 4.85: Objetivo e densidade de CG1, a1_5, com estabilização constante=0,9 e pré geração de colunas

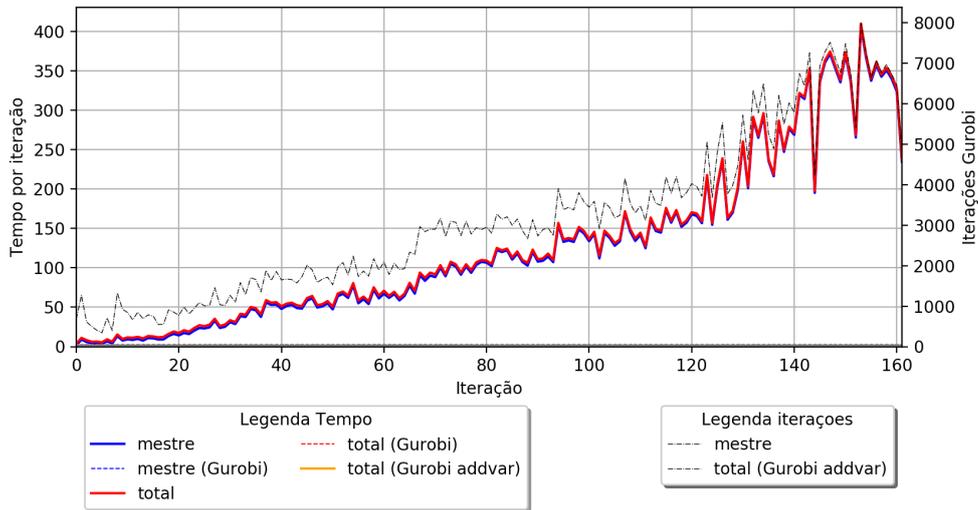


Figura 4.86: Tempos e iterações de CG1, a1_5, com estabilização constante=0,9 e pré geração de colunas

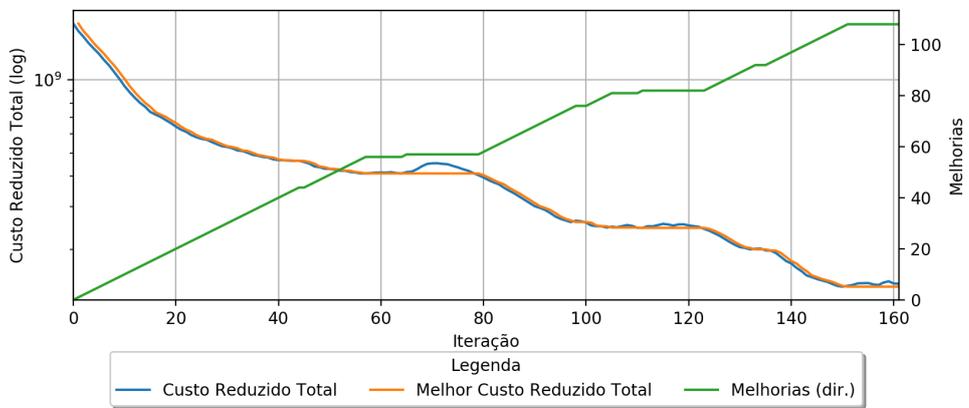


Figura 4.87: Custo reduzido de CG1, a1_5, com estabilização constante=0,9 e pré geração de colunas

instâncias a1_4 e a1_5 o ganho é menor, resulta em um custo reduzido inicial menor e estes apresentam reduções frequentes ao longo das iterações. Portanto a combinação da geração prévia e estabilização é uma combinação válida para acelerar a geração de colunas.

4.19

Modelo CG3

Os valores obtidos até então são resultantes do modelo *CG1*, que não considera as restrições de espalhamento ou de dependência. Também não considera o custo de movimentação de serviço. Pela ausência dessa componente de custo, o MIP resultante seleciona as colunas que mais reduzem o custo, porém pode haver uma grande movimentação de processos nos serviços, que ao validar com o programa *checker*, o custo real é maior que o custo calculado.

Como mencionado nos comentários dos ensaios iniciais de *CG1* e *CG2*, o espaço de soluções de *CG1* contém as soluções de *CG2*, portanto pode-se adicionar as restrições de serviço, pois os coeficientes dessas restrições são linearmente dependentes dos coeficientes de processos. O modelo *CG3* implementa essa expansão no MIP final.

A Tabela 4.9 contém os resultados de ensaios com *CG3*, com uma criação prévia de colunas e também com uma estabilização constante de $\alpha = 0,9$ das duais. Como o estágio de geração de colunas é o *CG1*, os valores de quantidades de iterações, melhorias e o melhor custo reduzido total final são idênticos. Na instância a1_3 há redução no objetivo e um aumento no tempo de execução. Para a instância a1_1 o objetivo se manteve o mesmo e somente há aumento de tempo. A instância a1_2 termina por tempo no MIP final e com valor mais alto, resultante do aumento de dificuldade por causa das restrições adicionadas. Para as demais instâncias, não ocorre alterações.

Tabela 4.9: Resultado parcial por geração de colunas, CG3, com estabilização constante=0,9 e geração prévia de variáveis

Instância	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
a1_1	44.306.501	3,916	CGRep	33	27	-12.375.912,144
a1_2	995.686.515	21601,794	CGRep - TMIP	456	207	-64,441
a1_3	583.005.757	15131,097	CGRep	352	191	-59,473
a1_4	ST	21600,018	TCG	179	106	-137.863.104,649
a1_5	ST	21600,020	TCG	161	108	-140.133.763,740

A adição das restrições inicialmente ignoradas resulta no aumento de tempo para resolver, porém o MIP final resulta no mesmo valor de objetivo

que o fornecido pelo `checker`.

4.20

Comentários

Dos resultados relacionados à geração de colunas, observa-se que o consumo de memória do computador utilizado é no máximo 3GB durante os ensaios.

A Tabela 4.10 contém resultados dos ensaios por geração de colunas para instância `a1_1`, com o modelo `CG1` como sendo o controle e este é usado como base, caso não seja informado outro modelo. Pode-se notar que a maioria das abordagens resultam em redução do custo final. Como comentado anteriormente, coeficientes distintos de estabilização resultam em situações finais diferentes. Um ensaio com um determinado coeficiente tem resolução porém os ensaios com coeficientes maior ou menor não conduzem resultados distintos da solução inicial. As colunas de motivo, quantidade de iterações e melhorias e o valor do melhor custo reduzido total dos ensaios de `CG1` com geração de colunas prévias e estabilização e os resultados de `CG3` são os mesmos, pois a etapa de geração de colunas é a mesma. Essas igualdades mostram uma repetibilidade no resultados em execuções com os mesmos parâmetros. O tempo mais alto de `CG3`, em relação à `CG1` com os mesmos parâmetros é devido ao tempo de expandir o modelo (adicionar as restrições que são inicialmente desconsideradas) e um tempo maior no MIP final pois há aumento de dificuldade do problema.

Para a instância `a1_2`, os resultados dos ensaios deste capítulo estão na Tabela 4.11, onde observa-se que a estabilização e a geração prévia de colunas resulta no término do estágio de geração de colunas. O resultado de `CG3` é pior devido ao aumento de complexidade do problema final, que resulta em uma execução pior, pois em ambos casos os ensaio são terminados por tempo.

Para a instância `a1_3`, os resultados dos ensaios deste capítulo estão na Tabela 4.12, onde observa-se que a geração prévia de colunas resulta no término do estágio de geração de colunas. Comparativamente a presença da estabilização resulta na adição de colunas que no modelo simplificado é melhor, porém ao incluir o custo de movimentação de serviço outras colunas são selecionadas. O tempo por `CG3` é maior resultante da adição das restrições de serviço e conseqüente aumento da dificuldade do problema.

Tabela 4.10: Resultado de geração de colunas, a1_1

Abordagem	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
controle	44.306.501	523,820	Concluído	2071	100	0.000
CG2	ST	3822,117	CGRep	2066	92	-812,091
otmizar após adição de variável	ST	628,460	Concluído	1741	140	-0,085
geração prévia	44.306.501	89,364	CGRep	665	98	-0,255
estabilização constante=0,05	44.306.501	552,316	CGRep	2005	113	-4,847
estabilização constante=0,10	44.306.501	353,180	CGRep	1632	97	-2,105
estabilização constante=0,15	44.306.501	703,019	CGRep	2307	141	-5,082
estabilização constante=0,20	44.306.501	753,156	CGRep	1536	117	-0,808
estabilização constante=0,25	ST	12,367	CGRep	119	10	-72.870.453,102
estabilização constante=0,30	44.306.501	389,692	Concluído	1613	160	-0,091
estabilização constante=0,35	44.306.501	351,305	CGRep	1665	176	-0,125
estabilização constante=0,40	ST	10,607	CGRep	108	8	-98.162.369,673
estabilização constante=0,45	ST	12,136	CGRep	119	8	-76.895.898,533
estabilização constante=0,50	ST	11,842	CGRep	117	16	-72.773.412,454
estabilização constante=0,55	44.306.501	197,011	Concluído	1049	173	-0,083
estabilização constante=0,60	44.306.501	184,217	CGRep	1079	139	-3,539
estabilização constante=0,65	ST	10,076	CGRep	106	15	-52.937.764,164
estabilização constante=0,70	ST	13,372	CGRep	132	40	-574.535,102
estabilização constante=0,75	44.306.501	121,937	CGRep	845	149	-18,322
estabilização constante=0,80	ST	0,457	CGRep	12	2	-159.819.024,200
estabilização constante=0,85	ST	0,157	CGRep	2	2	-168.023.475,800
estabilização constante=0,90	ST	0,158	CGRep	2	2	-177.929.165,200
estabilização constante=0,90 e geração prévia	44.306.501	2,695	CGRep	33	27	-12.375.912,144
CG3	44.306.501	3,916	CGRep	33	27	-12.375.912,144

Tabela 4.11: Resultado de geração de colunas de a1_2

Abordagem	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
controle	ST	21600,006	TCG	258	41	-614.226.567,775
CG2	ST	21600,063	TCG	69	3	-11.846.713.877,557
otmizar após adição de variável	ST	21784,699	TCG	183	13	-6.695.745.720,815
geração prévia	ST	21600,006	TCG	443	183	-63,891
estabilização constante=0,90	ST	21600,106	TCG	544	279	-4,965
estabilização constante=0,90 e geração prévia	881.988.071	21600,142	CGRep - TMIP	456	207	-64,441
CG3	995.686.515	21601,794	CGRep - TMIP	456	207	-64,441

Tabela 4.12: Resultado de geração de colunas a1_3

abordagem	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
controle	ST	21600,012	TCG	235	56	-53.550.603,682
CG2	ST	21600,102	TCG	58	8	-7.538.322.348,657
otmizar após adição de variável	ST	21622,581	TCG	224	49	-176.015.323,131
geração prévia	583.005.717	13689,631	CGRep	260	107	-16,195
estabilização constante=0,90	ST	21622,589	TCG	407	280	-99,181
estabilização constante=0,90 e geração prévia	583.005.818	14544,467	CGRep	352	191	-59,473
CG3	583.005.757	15131,097	CGRep	352	191	-59,473

Para as instâncias a1_4 e a1_5, os resultados deste capítulo estão nas Tabelas 4.13 e 4.14 respectivamente. Pode-se notar que os ganhos com a geração prévia e a estabilização são significativos, mesmo com um número de iterações menor. Em menos iterações há muito mais melhorias, que indica uma redução frequente no valor absoluto do custo reduzido total. Também termina com os custos reduzido em ordens de grandeza menor.

Tabela 4.13: Resultado de geração de colunas a1_4

abordagem	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
controle	ST	21600,013	TCG	185	21	-5.320.982.128,463
CG2	ST	21600,091	TCG	68	50	-11.202.433.750,248
otmizar após adição de variável	ST	21752,684	TCG	194	16	-6.605.570.634,171
geração prévia	ST	21600,003	TCG	208	5	-1.508.971.879,349
estabilização constante=0,90	ST	21600,029	TCG	210	37	-2.765.695.632,598
estabilização constante=0,90 e geração prévia	ST	21600,237	TCG	179	105	-137.863.104,649
CG3	ST	21600,018	TCG	179	106	-137.863.104,649

Tabela 4.14: Resultado de geração de colunas a1_5

abordagem	Objetivo	Tempo	Motivo	Iterações	melhorias	MCRT
controle	ST	21600,033	TCG	307	10	-5.332.429.863,654
CG2	ST	21600,037	TCG	163	1	-8.475.723.770,000
otmizar após adição de variável	ST	21698,931	TCG	346	14	-4.376.081.256,537
geração prévia	ST	21600,022	TCG	211	20	-907.900.503,482
estabilização constante=0,90	ST	21600,006	TCG	338	69	-1.533.807.826,658
estabilização constante=0,90 e geração prévia	ST	21600,355	TCG	161	108	-140.133.763,740
CG3	ST	21600,020	TCG	161	108	-140.133.763,740

Com base nos resultados conclui-se que, apesar de não haver resolução para todas as instâncias, a geração de colunas resulta em um uso menor de memória tendo como contrapartida um tempo maior. Portanto, é possível lidar com problemas maiores, usando um tempo maior.

Análise de Resultados

5.1

Programação inteira

No Capítulo 3, aborda-se o problema do desafio por programação inteira, cuja modelagem é baseada nas restrições presentes no documento de enunciado, transcrito no Capítulo 2, para restrições de programação inteira. A transcrição mais fidedigna, *MIP1*, não é capaz de resolver todas as instâncias devido à insuficiência de memória.

O modelo *MIP2* é criado para resolver o problema de memória, pois elimina um grupo de variáveis, rearranjando algumas definições do problema. Tanto *MIP1* quanto *MIP2* são modelos puramente inteiros, e visando resolver dificuldades numéricas é elaborado o modelo *MIP3* que altera algumas variáveis com grandes limites para contínuas. Um quarto modelo *MIP4* é criado visando reduzir os limites de algumas variáveis por meio do escalonamento de alguns coeficientes do problema.

A Tabela 5.1 contém um resumo dos resultados do capítulo. Os valores em negrito são as melhores soluções obtidas.

Tabela 5.1: Resultado de a1 por MIP

Instância	MIP1		MIP2		MIP3		MIP4	
	Objetivo	Tempo	Objetivo	Tempo	Objetivo	Tempo	Objetivo	Tempo
a1_1	44.306.501	0,706	44.306.501	0,295	44.306.501	0,293	44.306.501	0,288
a1_2	não resolveu	1398	804.180.142	3581,654	804.166.728	3564,085	804.713.563	4080,830
a1_3	não resolveu	1369	583.318.987	523,072	583.322.849	552,797	583.304.086	560,361
a1_4	não resolveu	3815	383.721.957	21600,421	381.150.275	21600,420	393.439.494	3566,204
a1_5	727.578.610	2424,695	727.578.312	18,337	727.578.310	18,408	727.603.785	14,530

Dos resultados apresentados, pode constatar que o modelo *MIP3* apresenta, em geral, melhores resultados. Somente para uma única instância, o modelo *MIP4* produz objetivo melhor.

Os modelos *MIP3* ou *MIP4* não são aptos para resolver instâncias

maiores, pois pode facilmente atingir o limite de tempo, como por exemplo para a instância `a1_4`. Outra desvantagem dos modelos MIPs é a multiplicidade de soluções retornadas, oriunda da manipulação de números com alta ordem de grandeza.

5.2

Geração de colunas

O Capítulo 4 aborda a técnica de geração de colunas, que tradicionalmente é usada para resolver problemas grandes com memória limitada. Para uso desta técnica, é necessário interpretar o problema como um problema de escolher uma solução dentro de um conjunto de soluções possíveis para cada máquina. Cria-se então um problema auxiliar de identificar que soluções devem pertencer a este conjunto.

Um primeiro modelo *CG1* é elaborado como uma versão simplificação do problema do desafio. Este modelo é utilizado para validar a técnica. Neste modelo somente uma instância fornece uma resposta distinta da configuração inicial. Ao trabalhar com o modelo completo, *CG2*, são obtidos resultados piores, pois para todas as instâncias a resposta fornecida é a configuração inicial. Da forma que foram modelados, o espaço de soluções de *CG2* está contido no espaços de *CG1*, portanto pode-se trabalhar na versão simplificada para formar um espaço de soluções maior para depois resolver o modelo completo.

Visando melhorar o processo de geração de colunas, são testados vários mecanismos, como otimizar após cada adição de variável ao problema. Este mecanismo produz ganhos na instância `a1_5`, para as demais instâncias não produz resultados satisfatórios.

Também utiliza-se de um conjunto adicional de variáveis, usando conhecimento do problema, resultando em uma melhor convergência para as instâncias `a1_2` e `a1_3`, que terminam com um custo reduzido na ordem de 10^1 e 10^2 respectivamente. Estes valores são próximos de zero, considerando que o valor inicial é da ordem de 10^9 . Para as demais há um ganho menor.

Outra abordagem adotada é o uso de estabilização das variáveis duais, que permitiu uma solução para a instância `a1_3`. Há ganhos menores para as demais instâncias.

Emprega-se também a combinação de métodos anteriores, geração prévia e estabilização, que é capaz de produzir soluções para as instâncias a1_2 e a1_3. Para as demais instâncias o ganho é mais significativo.

Por fim, o modelo *CG3* é elaborado para realizar a etapa de geração de colunas com a versão simplificada de *CG1*, usando uma geração prévia e estabilização. Ao término de descobrir o espaço de soluções, resolve o problema completo.

A Tabela 5.2 contém os resultados do capítulo. Os valores em negrito são os melhores valores obtidos.

Tabela 5.2: Resultado de a1 por geração de colunas

Abordagem	a1_1		a1_2	
	Objetivo	Tempo	Objetivo	Tempo
CG1	44.306.501	523,820	1.061.649.570	21600,006
CG2	49.528.750	3822,117	1.061.649.570	21600,063
CG1+otimização após adição	44.306.501	628,460	1.061.649.570	21784,699
CG1+geração prévia	44.306.501	89,364	1.061.649.570	21600,006
CG1+estabilização	49.528.750	0,158	1.061.649.570	21600,106
CG1+geração prévia+estabilização	44.306.501	2,695	881.988.071	21600,142
CG3+geração prévia+estabilização	44.306.501	4,001	890.029.619	21600,154

Abordagem	a1_3		a1_4	
	Objetivo	Tempo	Objetivo	Tempo
CG1	583.662.270	21600,012	632.499.600	21600,013
CG2	583.662.270	21600,102	632.499.600	21600,091
CG1+otimização após adição	583.662.270	21622,581	632.499.600	21752,684
CG1+geração prévia	583.005.717	13689,631	632.499.600	21600,003
CG1+estabilização	583.662.270	21622,589	632.499.600	21600,029
CG1+geração prévia+estabilização	583.005.818	14544,467	632.499.600	21600,237
CG3+geração prévia+estabilização	583.005.757	15131,097	632.499.600	21600,018

Abordagem	a1_5	
	Objetivo	Tempo
CG1	782.189.690	21600,033
CG2	782.189.690	21600,037
CG1+otimização após adição	782.189.690	21698,931
CG1+geração prévia	782.189.690	21600,022
CG1+estabilização	782.189.690	21600,006
CG1+geração prévia+estabilização	782.189.690	21600,355
CG3+geração prévia+estabilização	782.189.690	21600,020

Dos resultados apresentados, pode-se constatar que a instância a1_2 apresenta melhores resultados usando o modelo simplificado com uma geração prévia de colunas e estabilização. Na instância a1_3, o melhor resultado considera os custos de movimentação de serviço, presente no modelo *CG3*, além a geração prévia e estabilização. Para as instâncias a1_4 e a1_5, os ensaios terminam na etapa de geração de colunas.

5.3

Comparação com resultados oficiais

A Tabela 5.3 contém os melhores resultados obtidos. A coluna ROADEF tem o melhor valor que ocorreu durante o desafio¹. Vale ressaltar que o valor do desafio é obtido com o limite de tempo de 5 minutos.

Tabela 5.3: Resultado de a1

Instância	ROADEF	Melhor MIP		Melhor CG	
		Objetivo	Tempo	Objetivo	Tempo
a1_1	44.306.501	44.306.501	0,288	44.306.501	2,695
a1_2	777.532.896	804.166.728	3564.085	881.988.071	21600,142
a1_3	583.005.717	583.304.086	560,361	583.005.717	13689,631
a1_4	252.728.589	381.150.275	21600,420	632.499.600	21600,003
a1_5	727.578.309	727.578.310	18,408	782.189.690	21600,006

Com base nos resultados, pode-se notar o resultado a1_3 obtido por geração de colunas é igual ao melhor valor do desafio. Lembrando que a resolução destas instâncias tem um tempo limite maior de 6 horas. Para a instância a1_1 ambas metodologias resultam no mesmo valor do desafio. Para a instância a1_5, o valor obtido por programação inteira é praticamente igual ao do desafio, erro de 1 unidade de um número com ordem de grandeza 10^9 ($7,27 \times 10^8$).

¹<http://www.roadef.org/challenge/2012/en/qualif.php>

Conclusão e Trabalhos Futuros

O desafio proposto pela ROADEF/EURO, em parceria com a Google, insere-se em contexto atual de *data center*, que consiste em melhorar o uso dos recursos computacionais, distribuindo processos nos equipamentos disponíveis.

A limitação de tempo de 5 minutos do desafio levou as equipes participantes a trabalharem em heurísticas ou métodos probabilísticos, ou seja, baseando-se em métodos não determinísticos, em que não garantem a mesma resposta caso executado múltiplas vezes para o mesmo problema. Também não garantem a obtenção da solução ótima.

No presente trabalho, são utilizadas duas abordagens para tratar o tema: programação inteira e geração de colunas, métodos amplamente utilizados na resolução desta classe de problemas. As metodologias adotadas permitem a possibilidade de verificar que a solução obtida é ótima e podem apresentar resultados iguais ou melhores que os melhores resultados que ocorreram no desafio.

Outra característica é a repetibilidade, ou seja, execuções com os mesmos parâmetros no mesmo computador resultam praticamente nos mesmos resultados, podendo variar somente o tempo.

Entretanto, é necessário lidar com coeficientes com várias ordens de grandeza de variação, de 10^0 até 10^7 , que resulta em dificuldades numéricas. Os ensaios do MIP necessitam de uma grande quantidade de memória, quase todos os 16GB disponíveis no computador. Ao se resolver as instâncias a2 os programas são abortados por falta de memória. A metodologia por geração de colunas apresenta um consumo de memória muito menor, na ordem de 3GB. Apesar da vantagem em relação à memória, a geração de colunas requer mais tempo de processamento e também apresenta dificuldades para convergir enquanto exista um número pequeno de variáveis.

6.1

Sugestão para trabalhos futuros

Dada a experiência resultante deste trabalho, pode-se melhorar a geração de colunas de forma a garantir que forneça a solução ótima. Com a etapa de geração de colunas concluída, pode-se enumerar todas as colunas cujo custo reduzido é menor que a diferença da solução atual e da melhor solução inteira conhecida. As colunas que não foram selecionadas não pertencerão a solução inteira, ou seja, o valor da variável é nulo. Quando a diferença entre a solução atual e a melhor solução inteira for nula, então a resposta é ótima.

Uma segunda sugestão, é explorar uma geração de colunas com foco nos serviços e não nos processos, pois nos serviços estão definidas as restrições de espalhamento e dependência, que não realizam contribuição ao custo.

Outra possibilidade é melhorar a geração de colunas prévia, produzindo uma vizinhança maior. Desta forma pode-se iniciar a etapa de geração de colunas com uma convergência melhor.

Uma quarta sugestão é melhorar o tempo de resolução do MIP resultante, que pode ser por meio de realizar uma análise prévia por meio de *Constraint Programming*, para limitar as variáveis do problema final.

A

Referências bibliográficas

- [Afsar et al., 2016] AFSAR, H. M.; ARTIGUES, C.; BOURREAU, E.; KEDAD-SIDHOUM, S.. **Machine reassignment problem: the roaDef/euro challenge 2012**. Annals of Operations Research, p. 1–17, 2016.
- [Barnhart et al., 1998] BARNHART, C.; JOHNSON, E. L.; NEMHAUSER, G. L.; SAVELBERGH, M. W. P.; VANCE, P. H.. **Branch-and-price: Column generation for solving huge integer programs**. Operations research, 46(3):316–329, 1998.
- [Butelle et al., 2015] BUTELLE, F.; ALFANDARI, L.; COTI, C.; FINTA, L.; LETO-CART, L.; PLATEAU, G.; ROUPIN, F.; ROZENKNOPP, A.; CALVO, R. W.. **Fast machine reassignment**. Annals of Operations Research, p. 1–28, 2015.
- [Dantiz & Wolf, 1960] DANTZIG, G. B.; WOLFE, P.. **Decomposition principle for linear programs**. Operations research, 8(1):101–111, 1960.
- [Ford & Fulkerson, 1958] FORD JR, L. R.; FULKERSON, D. R.. **A suggested computation for maximal multi-commodity network flows**. Management Science, 5(1):97–101, 1958.
- [Gabay & Zaourar, 2012] GABAY, M.; ZAOURAR, S.. **A grasp approach for the machine reassignment problem**. In: EURO 2012-25TH EUROPEAN CONFERENCE ON OPERATIONAL RESEARCH, 2012.
- [Gabay & Zaourar, 2013] GABAY, M.; ZAOURAR, S.. **Variable size vector bin packing heuristics-application to the machine reassignment problem**. working paper or preprint, 2013.
- [Gavranovic et al., 2012] GAVRANOVIC, H.; BULJUBASIC, M.; DEMIROVIC, E.. **Variable neighborhood search for google machine reassignment problem**. Electronic Notes in Discrete Mathematics, 39:209–216, 2012.

- [Gilmore & Gomory, 1961] GILMORE, P. C.; GOMORY, R. E.. **A linear programming approach to the cutting-stock problem**. *Operations research*, 9(6):849–859, 1961.
- [Gilmore & Gomory, 1963] GILMORE, P. C.; GOMORY, R. E.. **A linear programming approach to the cutting stock problem—part ii**. *Operations research*, 11(6):863–888, 1963.
- [Hoffman et al., 2015] HOFFMANN, R.; RIFF, M. C.; MONTERO, E.; ROJAS, N.. **Google challenge: A hyperheuristic for the machine reassignment problem**. In: 2015 IEEE CONGRESS ON EVOLUTIONARY COMPUTATION (CEC), p. 846–853. IEEE, 2015.
- [Jaskowski et al., 2015] JASKOWSKI, W.; SZUBERT, M.; GAWRON, P.. **A hybrid mip-based large neighborhood search heuristic for solving the machine reassignment problem**. *Annals of Operations Research*, p. 1–30, 2015.
- [Legillon et al, 2015] LEGILLON, F.; MELAB, N.; RENARD, D.; TALBI, EL-G.. **A multi-objective evolutionary algorithm for cloud platform reconfiguration**. In: PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM WORKSHOP (IPDPSW), 2015 IEEE INTERNATIONAL, p. 286–291. IEEE, 2015.
- [Mars, 2016] MARS, S. **Modeling: Best practices and techniques**, 2016.
- [Masson et al., 2013] MASSON, R.; VIDAL, T.; MICHALLET, J.; PENNA, P. H. V.; PETRUCCI, V.; SUBRAMANIAN, A.; DUBEDOUT, H.. **An iterated local search heuristic for multi-capacity bin packing and machine reassignment problems**. *Expert Systems with Applications*, 40(13):5266–5275, 2013.
- [Mehta et al., 2012] MEHTA, D.; O’SULLIVAN, B.; SIMONIS, H.. **Comparing solution methods for the machine reassignment problem**. In: PRINCIPLES AND PRACTICE OF CONSTRAINT PROGRAMMING, p. 782–797. Springer, 2012.
- [Mrad et al., 2015] MRAD, M.; GHARBI, A.; HAOUARI, M.; KHARBECHÉ, M.. **An optimization-based heuristic for the machine reassignment problem**. *Annals of Operations Research*, p. 1–18, 2015.
- [Pessoa et al., 2010] PESSOA, A.; UCHOA, E.; DE ARAGÃO, M. P.; RODRIGUES, R.. **Exact algorithm over an arc-time-indexed formulation**

- for parallel machine scheduling problems. *Mathematical Programming Computation*, 2(3):259–290, 2010.
- [Pigattu et al.,2005] PIGATTI, A.; DE ARAGAO, M. P.; UCHOA, E.. **Stabilized branch-and-cut-and-price for the generalized assignment problem**. *Electronic Notes in Discrete Mathematics*, 19:389–395, 2005.
- [ROADEF, 2012] ROADEF; EURO. **ROADEF/EURO challenge 2012: Machine reassignment**. <http://challenge.roadef.org/2012/en/index.php>, 2012.
- [Ries et al., 2012] RIES, J.; BEULLENS, P.; SALT, D.. **Instance-specific multi-objective parameter tuning based on fuzzy logic**. *European Journal of Operational Research*, 218(2):305–315, 2012.
- [Saber et al., 2014] SABER, T.; VENTRESQUE, A.; GANDIBLEUX, X.; MURPHY, L.. **GeNePi: A multi-objective machine reassignment algorithm for data centres**. In: *INTERNATIONAL WORKSHOP ON HYBRID METAHEURISTICS*, p. 115–129. Springer, 2014.
- [Saber et al., 2015] SABER, T.; VENTRESQUE, A.; MARQUES-SILVA, J.; THORBURN, J.; MURPHY, L.. **Milp for the multi-objective vm reassignment problem**. In: *TOOLS WITH ARTIFICIAL INTELLIGENCE (ICTAI), 2015 IEEE 27TH INTERNATIONAL CONFERENCE ON*, p. 41–48. IEEE, 2015.
- [Wang et al., 2016] WANG, ZHUO; LU, ZHIPENG; YE, TAO. **Multi-neighborhood local search optimization for machine reassignment problem**. *Computers; Operations Research*, 68:16–29, 2016.
- [Wentges, 1997] WENTGES, P. **Weighted dantzig-wolfe decomposition for linear mixed-integer programming**. *International Transactions in Operational Research*, 4(2):151–162, 1997.
- [Xu & Foters, 2011] XU, J.; FORTES, J.. **A multi-objective approach to virtual machine management in datacenters**. In: *PROCEEDINGS OF THE 8TH ACM INTERNATIONAL CONFERENCE ON AUTONOMIC COMPUTING*, p. 225–234. ACM, 2011.