HARLEM MAURICIO MADRID VILLADIEGO

# HEURÍSTICAS PARA O PROBLEMA DE DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES EM UM AMBIENTE DE PRODUÇÃO FLOWSHOP

Dissertação apresentada á Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para a obtenção do título de *Magister Scientiae*

VIÇOSA
MINAS GERAIS – BRASIL
2014

**Ficha catalográfica preparada pela Biblioteca Central da Universidade Federal de Viçosa - Câmpus Viçosa**

T
M183h
2014

Madrid Villadiego, Harlem Mauricio, 2014-
Heurísticas para o problema de dimensionamento e sequenciamento de lotes em um ambiente de produção flowshop. / Harlem Mauricio Madrid Villadiego. – Viçosa, MG, 2014.
x, 126f. : il. (algumas color.) ; 29 cm.

Inclui apêndices.
Orientador: José Elias Claudio Arroyo.
Dissertação (mestrado) - Universidade Federal de Viçosa.
Inclui bibliografia.

1. Heurística. 2. Algorítmo. 3. Flowshop. 4. Scheduling. 5. Dimensionamento de lotes . I. Universidade Federal de Viçosa. Departamento de Informática. Programa de Pós-graduação em Ciência da Computação. II. Título.

CDD 22. ed. 518.1

**HARLEM MAURICIO MADRID VILLADIEGO**

# HEURÍSTICAS PARA O PROBLEMA DE DIMENSIONAMENTO E SEQUENCIAMENTO DE LOTES EM UM AMBIENTE DE PRODUÇÃO FLOWSHOP

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 25 de julho de 2014.

Marcone Jamilson Freitas Souza

André Gustavo dos Santos

José Elias Claudio Arroyo
Orientador

*"Only Time"*
(Enya)

# Sumário

# Lista de Figuras

# Lista de Tabelas

# Resumo

VILLADIEGO, Harlem Mauricio Madrid, M.Sc., Universidade Federal de Viçosa, julho de 2014. **Heurísticas para o problema de dimensionamento e sequenciamento de lotes em um ambiente de produção flowshop**. Orientador: José Elias Claudio Arroyo. Coorientador: André Gustavo Dos Santos.

Nesta dissertação é considerado o problema integrado de dimensionamento e sequenciamento de lotes em um ambiente de produção flow shop com máquinas que possuem diferentes capacidades de produção e com tempos de preparação dependentes da sequência. Neste trabalho se considera um horizonte de planejamento finito e dividido em períodos iguais. O problema consiste em determinar, para cada período, as dimensões dos lotes de produtos e o sequenciamento dos mesmos de tal maneira que as demandas dos clientes sejam atendidas e as capacidades das máquinas sejam respeitadas. O objetivo é minimizar a soma dos custos de processamento, preparação e estoques. Dada a complexidade do problema, neste trabalho propõe-se três algoritmos heurísticos, todos baseados na metaheurística Iterated Greedy *(IG)*. Nos algoritmos propostos, a melhor sequência de produção em cada período é determinada pelos procedimentos básicos (destruição-construção e busca local) do algoritmo *IG* e o melhor dimensionamento de lotes, para cada produto em cada período, é determinado utilizando diferentes métodos. No primeiro algoritmo, denominado *IG+HR*, é utilizada uma adaptação da heurística Horizonte Rolante *(HR)*. No segundo algoritmo, denominado *IG+MDL*, é utilizado um método de Melhoria de Dimensionamento de Lotes que consiste em adiantar e postergar a produção entre períodos. Já no terceiro algoritmo, denominado *IG+F&O*, é utilizada uma adaptação da heurística Fix and Otimize *(F&O)* proposto na literatura. Para analisar o desempenho dos algoritmos heurísticos

propostos, diferentes testes computacionais foram realizados utilizando um conjunto instâncias de pequeno, médio e grande porte. Os resultados obtidos são comparados com os resultados dos melhores métodos disponíveis na literatura. Esses resultados são validados através de testes estatísticos.

# Abstract

VILLADIEGO, Harlem Mauricio Madrid, M.Sc., Universidade Federal de Viçosa. July, 2014. **Heuristic to lot sizing and sequencing problem in a flow shop environment**. Adviser: José Elias Claudio Arroyo. Co-adviser: André Gustavo Dos Santos.

In this work, we consider the integrated lot sizing and sequencing problem in a permutation flow shop environment with machines that have different production capacities and with sequence-dependent setup times. In this work, we consider a finite planning horizon and divided into equal periods. The problem consists on determining the lot sizing and the production sequence in each period of a planning horizon so that the customer demands will be met and machines capacity respected. The objective is to minimize the sum of the setup, production and inventory costs. Given the complexity of the problem, this work proposes three heuristic algorithms, all them based on metaheuristic Iterated Greedy *(IG)*. In the proposed algorithms, the better production sequence, in each period, is determined by the basic procedures (destruction-construction and local search) of *IG* and the best lot sizing, for each product for each period, is determined using different methods.In the first algorithm, named *IG+HR*, an adaptations of Rolling Horizon *(HR)* heuristic is used. In the second algorithm, named *IG+MDL*, a improvement lot sizing method is used in which the production between the periods is advanced and postponed. In the third algorithm, named *IG+F&O*, an adaptations of the heuristic Fix and Optimize *(F&O)*, proposed in the literature, is used. To analysis the performance of the proposed heuristic algorithms, different computational tests were realized using a set of small, medium and large instances of the problem. The results obtained are compared against the results of the better methods

available in the literature. These results are validated by statistical test.

# Capítulo 1

# Introdução

Dado o fenômeno da globalização, as indústrias procuram melhorar seus planejamentos estratégicos e processos produtivos já que estão enfrentando, no dia a dia, uma competição com indústrias estrangeiras por um melhor posicionamento no mercado. Em vista disto, as indústrias devem se propor objetivos como: adaptar seus sistemas para a melhora contínua da produtividade, estabelecer sistemas flexíveis, desenvolver novos produtos, além de manter: tempos de espera, estoques e custos reduzidos, priorizando o atendimento das necessidades do cliente (Lustosa et al. [2008]). Tentar alcançar esses objetivos traz como consequência o aperfeiçoamento dos sistemas de produção. Assim, na busca da excelência, percebe-se a importância de um eficaz e eficiente Planejamento e Controle da Produção *(PCP)*. Ao gerenciar um processo produtivo o *PCP* é imprescindível para a sobrevivência da empresa em um mercado altamente competitivo (Lustosa et al. [2008]).

Em geral o *PCP* consiste, essencialmente, em um conjunto de funções inter-relacionadas que objetivam comandar o processo produtivo e coordená-lo com os demais setores administrativos da empresa. O *PCP* tem três características principais: conjunto de funções, comando do processo produtivo e núcleo de coordenação entre setores administrativos. Pode-se detalhar quais são estas funções, entretanto, a diversidade de tipos de indústria torna difícil especificar cada função, já que algumas podem ser muito importantes em uma indústria e sem importância ou inexistente em outras (Zaccarelli [1967]). Este trabalho se centra na função principal do *PCP*, o comando do processo produtivo e, em particular, nas indústrias de manufatura onde o *PCP* é determinado de maneira similar.

O *PCP* no setor produtivo é um sistema de transformação de informações.

Informações sobre estoque existentes, vendas previstas, linha de produtos, modo de produzir, capacidade produtiva e dentre outras informações são recebidas. O *PCP* tem como função transformar estas informações em ordens de produção (Zaccarelli [1967]). Referindo-se a um entorno completamente produtivo, as ordens de produção contêm decisões sobre o "que", "quanto" e "quando" produzir (Zaccarelli [1967]). Para a tomada destas decisões é imprescindível considerar três problemas principais do *PCP*, estes são: o carregamento, o sequenciamento e a programação da produção dos produtos.

Em um contexto geral, o carregamento é a quantidade de trabalho alocado a um centro de trabalho (Slack et al. [2009]). No contexto de uma empresa produtiva, pode-se definir o carregamento como a quantidade de um produto (tamanho de lotes de produção) a serem produzidos em uma determinada máquina, conhecido como problema de *dimensionamento de lotes*. O *sequenciamento* por outro lado consiste em estabelecer a ordem na qual os lotes de vários produtos serão produzidos. Na prática, frequentemente, são dadas prioridades a certos lotes de produtos, estabelecidas por um conjunto predefinidos de regras, como: prioridade a um cliente, data prometida, restrições físicas, LIFO, FIFO, dentre outras. Na literatura, estas regras são pouco consideradas. Ao determinar a sequência dos lotes de produção, algumas operações requerem um cronograma detalhado, mostrando em que instante de tempo os lotes devem iniciar e finalizar sua produção (Slack et al. [2009]). Este cronograma é conhecido como o problema de *programação da produção*.

A programação da produção depende do ambiente de produção, já que os tempos de cada produto de um determinado lote dependem da sua rota através das diferentes máquinas, da disponibilidade destas e dos tempos de outros produtos do mesmo ou de outro lote (Belo Filho et al. [2009]). Pinedo [2008] classifica os ambientes de produção em: única máquina, máquinas paralelas (idênticas, uniformes e não relacionadas), flowshop, jobshop e openshop. O ambiente de produção considerado neste trabalho é o flowshop, e neste os lotes de produção são processados por certo número de máquinas ligadas em série. Dessa forma, um fluxo unidirecional e ininterrupto de lotes de produção passa através da série de máquinas (Belo Filho et al. [2012]).

O problema tratado nesta dissertação é baseado nos trabalhos de Mohammadi et al. [2010c], Ramezanian et al. [2012] e Belo Filho et al. [2012]. O problema considera a integração dos problemas de dimensionamento e sequenciamento de lotes em um ambiente de produção tipo flowshop, onde as máquinas possuem capacidade limitada e estão dispostas em série considerando um horizonte de planejamento finito e dividido

em períodos. Portanto, deve-se determinar a quantidade a produzir de cada produto, assim como a ordem em que serão processados nas máquinas. O objetivo é determinar a programação da produção que minimize os custos totais de produção.

Para abordar o problema foram desenvolvidos três algoritmos heurísticos baseados na simples e eficiente heurística Iterated Greedy *(IG)*, a qual foi proposta por Ruiz & Stützle [2007] para o problema de flowshop scheduling. No primeiro algoritmo proposto, combina-se o *IG* com a heurística Horizonte Rolante *(HR)* proposta por Ramezanian et al. [2012], neste algoritmo, o qual é nomeado *IG+HR*, o *IG* procura a melhor sequência de produtos em cada período e a *HR* procura o melhor dimensionamento de lotes para cada produto em cada período. No segundo algoritmo proposto, o *IG* é utilizado para procurar as melhores sequências de lotes nos períodos, enquanto que, as decisões de dimensionamento de lotes são determinadas por alguns métodos propostos por Gopalakrishnan et al. [2001] denominados Métodos para melhorar o Dimensionamento de Lotes *(MDL)*, dado a utilização destes, temos nomeado o algoritmo como *IG+MDL*. No terceiro algoritmo proposto, o *IG* é combinado com a heurística Fix and Otimize *(F&O)* proposta por Pochet & Wolsey [2006], resultando no *IG+F&O*. Da mesma forma que nos dois algoritmos anteriores o *IG* tenta melhorar a sequência de produção em cada período, enquanto a heurística *F&O* é utilizada para procurar melhores dimensionamentos de lotes. As heurísticas propostas *IG+HR*, *IG+MDL* e *IG+F&O* são comparadas com os métodos que, do nosso conhecimento, são os melhores da literatura, estes são: o *A-Teams* proposto por Belo Filho et al. [2012] e a *HR* proposta por Ramezanian et al. [2012].

## 1.1 Objetivos

Nesta seção são apresentados os objetivos desta dissertação.

### 1.1.1 Objetivo geral

O objetivo principal desta dissertação é desenvolver um algoritmo heurístico híbrido eficaz, eficiente e competitivo com os métodos da literatura para encontrar soluções de alta qualidade em tempo computacional razoável para o problema de sequenciamento e dimensionamento de lotes em um ambiente de produção flowshop.

Para alcançar este objetivo, os objetivos específicos a seguir devem ser atingidos.

### 1.1.2 Objetivos específicos

- Analisar as pesquisas da literatura que tratam o problema abordado e problemas relacionados, assim como as estratégias utilizadas para resolvê-los.
- Desenvolver algoritmos heurísticos híbridos robustos para resolver de maneira eficiente o problema tratado.
- Realizar uma adequada calibração de parâmetros para todos os algoritmos
- Resolver as classes de problemas-testes (instâncias) encontrados na literatura a fim de validar os resultados obtidos pelos algoritmos desenvolvidos.

## 1.2 Estrutura do trabalho

O restante da dissertação é organizado da seguinte forma: o capítulo 2 apresenta uma revisão da literatura dos trabalhos mais representativos do problema tratado. No capítulo 3 mostra-se a definição e formulação matemática do problema considerado. Nos capítulos 4 e 5 são descritos os algoritmos da literatura e os algoritmos propostos, respectivamente. No capítulo 6, os experimentos computacionais, como a calibração dos algoritmos e os resultados, são apresentados. Finalmente, no capítulo 7 são apresentadas as considerações finais seguido das propostas de pesquisas futuras.

# Capítulo 2

# Revisão da literatura

Devido à competitividade causada pela globalização, as indústrias, em especial as manufatureiras, procuram melhorar seus processos produtivos a fim de obter um melhor posicionamento no mercado. Para isto, é de vital importância que o planejamento e controle da produção da empresa seja realizado da forma mais eficiente possível.

O planejamento e controle de produção *(PCP)* é a atividade que permite coordenar e dirigir todas as operações de um processo produtivo (Tubino [2009]). O *PCP* tem por objetivo a descrição das quantidades a produzir em cada um dos períodos de tempos previstos de forma que não se violem as limitações de capacidade das instalações e se disponha de suficientes produtos para satisfazer as demandas dos mesmos (Corrêa et al. [2001]). Trata-se de uma das etapas mais importantes dentro da estratégia produtiva da empresa.

Da definição exposta acima, pode-se deduzir o conjunto de elemento que compõem o *PCP*, entre estes, o horizonte temporal, a capacidade da produção instalada, a taxa de produção ou quantidade a produzir em cada instante de tempo e o estoque que se mantém de um período a outro ao longo do período de planejamento (da Rocha [2008]). No *PCP* deve ser realizado de forma sistemática e através do estudo do conjunto dos custos da empresa, entre estes: os custos de processamento, custos de preparação dos equipamentos, custes de estoque, custos da variação da capacidade produtiva, custos de variação da produção e custos de mão de obra (Antunes [2008]). O *PCP* deve satisfazer as condições da empresa, minimizando os custos totais da mesma.

Como se pode notar, o *PCP* encontra-se bem relacionado com a área de produção e, segundo Zaccarelli [1967], o planejamento e controle da produção envolve decisões

sobre: quê, quando e como produzir determinados produtos. Neste contexto, se encontram os problemas de dimensionamento e sequenciamento de lotes, os quais são o escopo deste trabalho e portanto serão descritos a seguir.

O problema de dimensionamento de lote *(PDL)* consiste em determinar a quantidades e o período no qual os lotes serão produzidos (Bahl et al. [1987]). A produção destes lotes é realizada em um horizonte de planejamento finito dividido em períodos (semanas, meses, anos, etc.) no qual existe uma demanda de produtos a ser atendida. Neste problema existem certas restrições como atendimento à demanda, limites de estoque e capacidade dos equipamentos (máquinas). Geralmente, o objetivo deste problema é a minimização dos custos totais de produção, representados pela somatória dos custos de processamentos, custos de preparação das máquinas e custos de estoque.

O *PDL*, segundo Bahl et al. [1987] pode ser classificado como mono-estágio ou múltiplo-estágio. O *PDL* múltiplo-estágio encontra-se nos processos produtivos em que produtos finais demandam a pré-produção de componentes da sua estrutura, ou seja, existe a necessidade de atender a demanda dos produtos componentes, conhecida como demanda interna. Em Kuik et al. [1994], encontra-se uma revisão e classificações para o PDL múltiplo-estágio. O PDL mono-estágio caracteriza-se pela independência de seus produtos finais durante a produção, ou seja, os produtos não demandam a pré-produção de componentes de sua estrutura. O PDL mono-estágio pode ser considerado tanto para um único ou vários produtos (múltiplo-produto) e com ou sem restrições de capacidade. Uma revisão bibliográfica completa pode ser encontrada em Pochet & Wolsey [2006]

O caso mais pesquisado na literatura é o PDL mono-estágio múltiplo-produto com restrições de capacidade. A motivação para estudar essa classe de problemas está no fato de que, além de sua potencialidade de aplicações, o problema mono-estágio aparece como um subproblema em diversos casos, de modo que, implementações eficientes dos bons algoritmos disponíveis melhoram o desempenho de algoritmos projetados para problemas mais gerais (Bahl et al. [1987]).

Quando o problema contém a característica de múltiplo-produto, fatores importantes como tempo de processamentos dos produtos e tempos de preparação das máquinas são considerados, já que estes impactam diretamente nas restrições de capacidades, tempo e custo. Na literatura, vários autores discutem a importância de

considerar os tempos de preparação. Billington et al. [1994] destacam que a utilização ou não dos tempos de preparação depende do sistema produtivo, sendo que em alguns casos os tempos de preparação estão incluídos nos tempos de processamento. Trigeiro et al. [1989] também se centram na importância dos tempos de preparação, mostrando uma lista de problemas nos quais a consideração dos tempos de preparação é fundamental. Florian et al. [1980] afirmam que para problemas com recursos limitados de produção (capacidade: tempo disponível) e considerando tempos e custos de preparação torna o PDL NP-Difícil.

Um aspecto fundamental que se apresenta no PDL mono-estágio múltiplo-produto é o ambiente de produção. Geralmente, o ambiente de produção é composto por máquinas e estas devem estar preparadas para processar determinado produto. Esta preparação incorre diretamente em um tempo e, portanto, na capacidade, que geralmente é um limitante de tempo. Cabe ressaltar que a preparação das máquinas pode ser dependente ou independente da sequência dos produtos. Dado à existência de mais de um produto e de uma ou várias máquinas, é necessária uma programação da produção. A programação da produção consiste na ordem (ou sequência) na qual os lotes serão processados, assim como a determinação do tempo inicial e final de processamento de cada lote em cada máquina (Pochet & Wolsey [2006]).

A programação da produção depende do ambiente de produção, já que os tempos de cada produto de um determinado lote dependem da sua rota através dos estágios (máquinas) de produção, da disponibilidade desses estágios e dos tempos de outros produtos do mesmo ou de outro lote (Belo Filho et al. [2009]). Existem vários tipos de ambiente de produção definidos de acordo à configuração das máquinas e a rota dos produtos através destas. Pinedo [2008] classifica os ambientes de produção em: única máquina, máquinas paralelas (idênticas, uniformes e não relacionadas), flowshop, jobshop e openshop. No ambiente de única máquina, todos produtos devem ser processados nesta única máquina e sem interrupção, a máquina só pode processar um produto por vez. Em máquinas paralelas idênticas, os produtos têm o mesmo tempo de processamento e tempo de preparação e pode ser executado em qualquer uma das máquinas. Em máquinas paralelas uniformes, o tempo de processamento de um produto e o tempo de preparação de algumas máquinas são proporcionais aos tempos correspondentes em outras máquinas. Em máquinas paralelas não relacionadas, não existe relação alguma entre os tempos de processamentos e preparação de máquinas distintas, ou seja, cada máquinas tem tempo de processamento e tempo de preparação distinto para cada produto. No ambiente de produção flowshop, os produtos são

processados por uma quantidade de máquinas ligadas em série, dessa forma, um fluxo unidirecional contínuo e ininterrupto de produtos passa através da série de máquinas. No jobshop, cada produto tem sua própria rota de produção preestabelecida, sendo possível um produto ser processados mais de uma vez em uma máquina. No Openshop não existe uma rota preestabelecida ou restrição para o processamento dos produtos.

O ambiente de produção pesquisado nesta dissertação é o flowshop, portanto, a seguir apresenta-se uma breve revisão deste problema.

Johnson [1954] foi o primeiro a resolver o problema de fowshop scheduling*(PFS)* otimamente para duas máquinas, tendo como objetivo o makespan (instante de término da produção). Segundo Garey et al. [1976], o *PFS* para mais de duas máquinas torna-se NP-Difícil.

No PFS cada máquina do processo produtivo pode ter uma sequência diferente de produção, caracterizando o flowshop puro. Porém, na literatura e na prática industrial considera-se o flowshop permutacional, o qual mantém a mesma sequência de produção para todas as máquinas. Segundo Belo Filho et al. [2012], esta consideração é justificada pela restrição do número de programações possível, já que no flowshop puro, com $M$ máquinas e $N$ produtos diferentes, existem $(N!)^m$ programações e para o caso permutacional existem $N!$.

O problema de flowshop permutacional consiste em um conjunto de $N$ produtos a serem processados em um conjunto de $M$ máquinas na mesma ordem (ou sequência) obedecendo as seguintes suposições:

- Cada um dos produtos tem a mesma sequência de produção nas máquinas.
- Cada produto só pode ser processado em uma única máquina ao mesmo tempo.
- Cada máquina pode processar só um produto ao mesmo tempo.
- Cada produto é processado só uma vez em cada máquina.
- Chama-se operação à execução de um produto em uma máquina, e estas não são interrompidas.

O problema consiste então em definir a sequência de processamentos dos produtos nas máquinas, com objetivos especificados e respeitando as suposições anteriormente mencionadas. Cheng et al. [2000] e Allahverdi et al. [1999] realizam uma revisão de literatura de *PFS* considerando principalmente tempos de preparação. Uma revisão

mais extensa é realizada por Gupta & Stafford Jr [2006] comemorando os 50 anos do flowshop, a partir do trabalho de Johnson [1954], onde são discutidos cronologicamente e com detalhes as pesquisas realizadas neste período. Pouco depois, Ruiz & Stützle [2007] foram os primeiros a implementar a metaheurística Iterated Greedy a um problema de *scheduling*, a qual tem sido de boa aceitação na literatura, especificamente em problemas de flowshop com várias características. Exemplos destes são: Ruiz & Stützle [2008], Mascia et al. [2013], Fatih Tasgetiren et al. [2013] e Lin et al. [2013]. Recentemente Pan & Ruiz [2013] realizaram uma extensa revisão da literatura sobre heurísticas para o problema em questão.

## 2.1 Problemas integrados de dimensionamento e sequenciamento de lotes

Karimi et al. [2003] afirmam que os problemas de dimensionamento de lotes e os problemas de sequenciamento, tanto na teoria como na prática, podem ser tratados de forma independente. Por outro lado, Toso & Morabito [2005] e Toledo et al. [2007] afirmam que em problemas práticos específicos, ao tratá-los de forma independente se apresentam dificuldades para que a produção se torne flexível às mudanças do mercado e para que se obtenham soluções boas e factíveis com base na capacidade disponível e nos prazos de entregas estabelecidos. Stadtler Hartmut [2008], Urrutia et al. [2012], Seeanner et al. [2013], Ramezanian & Saidi-Mehrabad [2013] e Smith-Daniels & Ritzman [1988] também discordam de Karimi et al. [2003] e afirmam que as decisões de planejamento, com relação ao dimensionamento e sequenciamento de lotes, têm que ser tomadas simultaneamente, já que estes problemas encontram-se inter-relacionados, ou seja, decisões no sequenciamento afeitam diretamente ao dimensionamento e vice-versa. Drexl & Kimms [1997], Staggemeier & Clark [2001] e Zhu & Wilhelm [2006] realizam uma revisão de literatura de dimensionamento de lotes e sequenciamento sobre alguns modelos que integram estes problemas.

Como o escopo deste trabalho é o dimensionamento e sequenciamento de lotes em ambiente de produção flowshop, os trabalhos que para nosso conhecimento são os mais representativos, são discutidos a seguir.

Smith-Daniels & Ritzman [1988] foram os primeiros a propor um modelo matemático para o problema integrado de dimensionamento e sequenciamento de lotes. Os autores consideram no problema um horizonte de planejamento finito,

famílias de produtos, estoque intermediário, tempos de preparação dependentes da sequência, e *setup carryover* (preparação das máquinas mantidas entre períodos). O modelo é formulado para uma indústria de alimentos específica. Dada a grande quantidade de variáveis que o modelo matemático apresenta, este só é viável para instâncias menores, portanto não foi muito considerado na literatura.

Sikora et al. [1996] tratam o problema integrado de dimensionamento e programação da produção num ambiente *flow line*. O problema foi observado em uma indústria de placas de circuito impresso, considerando restrições de capacidade nas máquinas, tempos de preparação dependente da sequência e estoque intermediário. O objetivo é minimizar o *makespan* de cada período e os custos de estoque. Os autores propõem uma heurística chamada "integrated approach" baseada em dois enfoques diferentes, a heurística de Silver & Meal [1973] para o dimensionamento e a heurística de Palmer [1965] para o sequenciamento de lotes. Para o mesmo problema, Sikora [1996] propõe um Algoritmo Genético o qual é comparado com o método heurístico "integrated approach" de Sikora et al. [1996] e conclui que o Algoritmo Genético apresenta melhores resultados. O algoritmo genético apresenta operadores de cruzamento e mutação tanto para o sequenciamento como para o dimensionamento de lotes.

Ponnambalam & Reddy [2003] tratam o mesmo problema que Sikora [1996] e ainda considera o conceito de *lot splitting* (divisão de um lote de produção). Os autores propõem um método heurístico híbrido combinando um algoritmo genético e um simulated annealing. O algoritmo genético tenta resolver o dimensionamento de lotes enquanto o simulated annealing trata o sequenciamento. Ponnambalam & Reddy [2003] comparam o método heurístico híbrido com a heurística proposta por Sikora [1996] e observam que o método heurístico híbrido obtém melhores resultados.

Belo Filho et al. [2009] consideram o problema de dimensionamento e sequenciamento de lotes em ambiente de produção flowshop com estoque intermediário e restrições de capacidade. Esses autores propõem um sistema multiagentes "Times Assíncronos" e comparam os resultados obtidos com os métodos propostos por Ponnambalam & Reddy [2003] e Sikora [1996]. Os autores afirmam que o sistema multiagentes apresenta melhor desempenho.

Mohammadi et al. [2010b] apresentam um modelo matemático para o problema integrado de dimensionamento e sequenciamento de lotes. A principal característica

do problema é a preparação nas máquinas dependente da sequência, e esta preparação é preservada entre os períodos. Outras restrições como capacidades nas máquinas, demanda externa e armazenamento não permitido são consideradas. Vale ressaltar que o ambiente de produção tratado é o flowshop puro e o permutacional. Já que o modelo não encontra soluções para instâncias grandes em tempo computacional razoável, os autores desenvolvem dois limitantes inferiores e quatro enfoques heurísticos, dois baseados na estratégia de "horizonte rolante" e dois na estratégia "relax and fix". Os autores afirmam que as heurísticas com a última estratégia apresentam melhores resultados.

Mohammadi et al. [2010a] trabalham sobre a pesquisa de Mohammadi et al. [2010b]. Usando o mesmo modelo matemático e os mesmos limitantes inferiores, os autores apresentam melhoras significativas nas heurísticas. Vale ressaltar que são usadas as mesmas estratégias de "horizonte rolante" e "relax and fix" para resolver o problema no ambiente flowshop permutacional.

Mohammadi et al. [2010c] consideram o mesmo modelo matemático e os limitantes inferiores de Mohammadi et al. [2010b]. Porém apresentam duas novas heurísticas baseadas na estratégia de horizonte rolante para o dimensionamento de lotes e para o sequenciamento, a clássica heurística para flowshop, NEH de Nawaz et al. [1983]. Nesse trabalho o ambiente de produção é também limitado ao flowshop permutacional.

Belo Filho et al. [2012] estendem o trabalho realizado em 2009 (Belo Filho et al. [2009]) propondo o mesmo sistema multiagentes "Times Assíncronos" mas com melhoras significativas nos agentes. Esses autores comparam os resultados com os obtidos pelos métodos propostos por Mohammadi et al. [2010a] e afirmam que o sistema multiagentes apresenta melhores resultados.

Ramezanian et al. [2012] propõem um novo e mais eficiente modelo matemático para o mesmo problema. Os autores ressaltam que a vantagem do novo modelo está nas poucas restrições e nas poucas variáveis contínuas e binárias. Além do novo modelo são desenvolvidas duas heurísticas baseadas na estratégia de horizonte rolante. Ramezanian et al. [2012] comparam todos seus métodos com os propostos por Mohammadi et al. [2010c] e demostram que os métodos propostos apresentam melhor rendimento em comparação aos encontrados na literatura.

Vale ressaltar que as pesquisas de Mohammadi et al. [2010b], Mohammadi et al. [2010a], Mohammadi et al. [2010c], Ramezanian et al. [2012] envolvem exatamente o mesmo problema e fazem a mesma consideração para trabalhos futuros: a implementação de meta-heurísticas para o problema tratado.

# Capítulo 3

# Definição e modelagem do problema

Nesta dissertação é considerado o Problema integrado de Dimensionamento e Sequenciamento de Lotes em um ambiente de produção Flow Shop *(PDSLFS)*. Cada lote representa uma quantidade específica de um produto, os quais devem ser processados em um conjunto de máquinas restritas em capacidade e dispostas em série em um horizonte de planejamento finito e dividido em períodos. Existe uma demanda para cada produto em cada período, a qual deve ser atendida sem atrasos. Portanto, deve-se determinar a quantidade a produzir (dimensão do lote) de cada produto, assim como a ordem na qual serão processados nas máquinas em cada um dos períodos. O objetivo consiste na minimização dos custos totais de produção *(f)*, os quais são compostos pelo somatório dos custos de processamento, preparação e estoque. Neste problema, assume-se que:

- Existe um horizonte de planejamento $T$ (número de períodos determinados para a produção), o qual é finito e dividido em $t$ períodos iguais.
- $N$ produtos independentes devem ser processados na mesma ordem em um conjunto de $M$ máquinas.
- As $M$ máquinas são dispostas em série.
- Cada máquina pode processar só um lote de algum produto por vez.
- Todas as $M$ máquinas têm restrições de capacidade.
- Exitem custos e tempos de preparação de máquinas dependentes da sequência ao fazer uma troca de produtos.

- A preparação da máquina deve iniciar e terminar no mesmo período.
- Não é necessária a preparação de máquinas (entre períodos) para processar o mesmo produto. Ou seja, as máquinas preservam sua preparação entre os períodos.
- No início do horizonte de planejamento, as máquinas estão preparadas para um determinado produto.
- Em cada período existe uma demanda de produtos acabados, a qual sempre deve ser atendida sem atrasos.
- Não existe estoque entre duas máquinas consecutivas.
- O tempo de transporte de produtos intermediários e outros processos entre as máquinas sucessivas é desprezível.
- O processamento de um lote em uma máquina só se inicia se todos os recursos necessários estão disponível para a produção. Ou seja, se todo este lote já foi processado pela máquina anterior.
- Não há prioridade de produção nas máquinas entre produtos.

## 3.1 Modelo matemático para o PDSLFS

Nesta seção são apresentadas a notação e definição dos parâmetros e variáveis utilizadas no modelo matemático, o qual foi proposto por Ramezanian et al. [2012].

**Parâmetros**

$N$: número de produtos diferentes.

$M$: número de máquinas.

$T$: horizonte de planejamento (número de períodos).

$b_{jm}$: tempo de produção do produto $j$ na máquina $m$.

$C_{mt}$: capacidade (tempo disponível) da máquina $m$ no período $t$.

$d_{jt}$: demanda externa do produto $j$ no período $t$.

$p_{jmt}$: custo de produção do produto $j$ na máquina $m$ no período $t$.

$h_{jm}$: custo de estoque do produto $j$ na máquina $m$.

$S_{ijm}$: tempo de preparação de máquina do produto $i$ ao produto $j$ na máquina $m$.

$W_{ijm}$: custo de preparação de máquina do produto $i$ ao produto $j$ na máquina $m$.

$S_{j0m}$: configuração inicial de preparação da máquina $m$.

$BigM$: um número grande.

**Variáveis de decisão**

$x_{jt}$: quantidade do produto $j$ produzido no período $t$.

$I_{jt}$: estoque do produto $j$ no período $t$.

$SO_{jmt}$: tempo de inicio do produto $j$ na máquina $m$ no período $t$.

$CO_{jmt}$: tempo de finalização do produto $j$ na máquina $m$ no período $t$.

$Y_{ijt}$: variável binária que é igual a 1 se o produto $j$ é processado imediatamente depois do produto $i$ no período $t$, 0 caso contrario.

$Z_{jt}$: variável binária que é igual a 1 se $x_{jt} > 0$, 0 caso contrário.

$\alpha_{jt}$: variável binária que é igual a 1 se o produto $j$ é produzido primeiro (inicia sua produção) no período $t$, 0 caso contrário.

$\beta_{jt}$: variável binária que é igual a 1 se o produto $j$ é produzido por último (finaliza a produção) no período $t$, 0 caso contrário.

$w_t$: variável binária que é igual a 1 se pelo menos um produto é produzido no período $t$, 0 caso contrário.

$\delta_t$: é 0 se exatamente um produto é produzido no período $t$, e um número não negativo diferente de 0 no caso contrário.

$Y'_{ijt}$: Variável continua, onde $0 \leq Y'_{ijt} \leq 1$. É estritamente positiva quando o produto $i$ é produzido por último no período $t-1$ e o produto $j$ é produzido de primeiro no período $t$ e zero em qualquer outro caso.

**Modelo Matemático**

$$Min \sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{t=1}^{T} p_{jmt} \times x_{jt} + \sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{t=1}^{T} h_{jm} \times I_{jt} + \sum_{i=1}^{J}\sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{t=1}^{T} w_{ijm} \times (Y_{ijt} + Y'_{ijt}) \tag{3.1}$$

**s.t.**

$$I_{j(t-1)} + x_{jt} = I_{jt} + d_{jt} \qquad \forall j \in N, t \in T \tag{3.2}$$

$$x_{jt} \leq BigM \times Z_{jt} \qquad \forall j \in N, t \in T \tag{3.3}$$

$$CO_{jmt} = SO_{jmt} + b_{jm} \times x_{jt} \qquad \forall j \in N, m \in M, t \in T \tag{3.4}$$

$$CO_{jmt} \leq C_{mt} \qquad \forall j \in N, m \in M, t \in T \tag{3.5}$$

$$SO_{jmt} \geq CO_{j(m-1)t} \qquad \forall j \in N, t \in Tm = 2, ..., M \tag{3.6}$$

$$SO_{jmt} \geq CO_{imt} + S_{imt} \times Y_{ijt} - BigM(1 - Y_{ijt}) \qquad \forall i, j \in N, i \neq j, m \in M, t \in T \tag{3.7}$$

$$\sum_{i \in J, i \neq j}\sum_{i \in J} Y_{ijt} \geq \sum_{j \in J} Z_{jt} - 1 \qquad \forall t \in T \tag{3.8}$$

$$\sum_{i \in J, i \neq j} Y_{ijt} \leq Z_{jt} \qquad \forall j \in N, t \in T \tag{3.9}$$

$$\sum_{i \in J, i \neq j} Y_{jit} \leq Z_{jt} \qquad \forall j \in N, t \in T \tag{3.10}$$

$$\sum_{j \in J} Z_{jt} \leq BigM \times w_t \qquad \forall t \in T \tag{3.11}$$

$$\sum_{j \in J} Z_{jt} \leq (N-1) \times \delta_t \qquad \forall t \in T \tag{3.12}$$

$$\sum_{j \in J} \alpha_{jt} = w_t \qquad \forall t \in T \tag{3.13}$$

$$\sum_{j \in J} \beta_{jt} = w_t \qquad \forall t \in T \tag{3.14}$$

$$\alpha_{jt} \leq (Z_{jt} - \sum_{i \in J} Y_{ijt}) \qquad \forall j \in N, t \in T \tag{3.15}$$

$$\beta_{jt} \leq (Z_{jt} - \sum_{i \in J} Y_{jit}) \qquad \forall j \in N, t \in T \tag{3.16}$$

$$\alpha_{jt} + \beta_{jt} \leq 2 - \delta_t \qquad \forall j \in N, t \in T \tag{3.17}$$

$$1 + Y'_{ijt} \geq \alpha_{jt} + \beta_{i(t-1)} \qquad \forall i, j \in N, t = 2, ..., T \tag{3.18}$$

$$SO_{jmt} \geq \sum_{i \in J} s_{ijm} \times Y'_{ijt} \qquad \forall j \in N, m \in Mt \in T \tag{3.19}$$

$$x_{jt}, I_{jt}, CO_{jmt}, SO_{jmt}, Y'_{ijt}, \delta_t \geq 0; I_{j0} = 0; \forall j \in N, m \in M, t \in T \tag{3.20}$$

$$Y_{ijt}, Z_{jt} = \{0, 1\}; \forall i, j \in N, t \in T \tag{3.21}$$

A função objetivo, Equação 3.1, é a minimização dos custos totais de produção *(f)*. Os custos envolvidos são: custos de processamento, custos de preparação das máquinas ($f_{cpm}$) e custos de estoque. As restrições 3.2 garantem o balanço do fluxo dos produtos em estoque e os produtos previstos a fabricar, para satisfazer a demanda em cada período. A restrição 3.3 está relacionada com os produtos previstos a fabricar, é por isto que esta restrição apresenta uma relação entre a variável $x_{jt}$ e a variável binária $Z_{jt}$, a qual indica se o produto $j$ será processado ou não no período $t$. As restrições 3.4 e 3.6 definem a relação entre o tempo de início e de finalização da produção prevista, ou seja, nestas restrições se determina o tempo de início e finalização de cada lote de produção em cada máquina em cada período. A restrição 3.5 não permite ultrapassar a

capacidade de cada máquina em cada período. A restrição 3.6 obriga a iniciar o processamento do $j-$ésimo lote de determinado produto só quando foi totalmente processado na máquina anterior. A restrição 3.7 permite iniciar o processamento de um lote de um produto só quando o lote do produto precedente finalizou seu processamento na mesma máquina, incluído o tempo necessário de preparação. As restrições 3.8 a 3.10 determinam a sequência dos lotes de produção, considerando às restrições 3.9 e 3.10, onde um produto $j$ é incluído na sequência só se é produzido no período $t$.

O conjunto de restrições 3.11 a 3.18 determinam qual produto é processado primeiro e por último em cada período.

Conforme a restrição 3.11, a variável $w_t$ será 1 se pelo menos um produto é produzido no período $t$. As restrições 3.13 e 3.14 garantem que $\alpha$ e $\beta$ serão iguais a 1 quando exatamente um produto é produzido em cada período, obviamente, se tem produção neste período. Se o produto $j$ não é o primeiro produzido no período $t$, a restrição 3.15 força à variável $\alpha$ a ser zero. Do mesmo modo, na restrição 3.16, a variável $\beta$ será zero se um produto $j$ não é produzido de último no período $t$. A restrição 3.17 força a variável $\delta$ a zero se exatamente um produto é produzido no período $t$. $\delta$ torna-se positiva pela restrição 3.12 se mais de um produto é produzido no período $t$. Se $\delta > 0$ para um período dado, então os valores de $\alpha$ e $\beta$ não poderão ser iguais a 1 para o mesmo produto, isto é lógico já que quando mais de um produto é produzido em um período, o mesmo produto não pode ser o primeiro produzido e ao mesmo tempo o último em um mesmo período. A restrição 3.18 assegura que a variável $Y'_{ijt}$ será 1 quando o produto $i$ é o último produzido no período $t-1$ e o produto $j$ é o primeiro produzido no período $t$, 0 caso contrário. A restrição 3.19 permite iniciar o processamento de um produto (primeiro produto) só depois de preparar a máquina no início de cada período. As restrições 3.20 e 3.21, indicam o tipo de cada variável.

## 3.2 Exemplo numérico

Para uma melhor compreensão do problema, um exemplo numérico é apresentado. Este exemplo contém três produtos, três máquinas e três períodos. Os dados são mostrados nas Tabelas 3.1 a Tabela 3.7

Tabela 3.1: Tempo de Produção ($b_{jm}$) do produto $j$ na máquina $m$

| $b_{jm}$ | $m_1$ | $m_2$ | $m_3$ |
|---|---|---|---|
| $j_1$ | 1.9 | 1.9 | 1.7 |
| $j_2$ | 1.5 | 1.8 | 1.8 |
| $j_3$ | 1.8 | 1.5 | 1.9 |

Tabela 3.2: Capacidade ($C_{mt}$) da máquina $m$ no Período $t$

| $C_{mt}$ | $t_1$ | $t_2$ | $t_3$ |
|---|---|---|---|
| $m_1$ | 1349 | 1409 | 1367 |
| $m_2$ | 1389 | 1360 | 1346 |
| $m_3$ | 1355 | 1317 | 1447 |

Tabela 3.3: Demanda ($d_{jt}$) do produto $j$ no período $t$

| $d_{jt}$ | $t_1$ | $t_2$ | $t_3$ |
|---|---|---|---|
| $j_1$ | 100 | 120 | 98 |
| $j_2$ | 105 | 150 | 142 |
| $j_3$ | 110 | 114 | 123 |

Tabela 3.4: Custo de estoque ($h_{jm}$) do produto $j$ na máquina $m$

| $h_{jm}$ | $m_1$ | $m_2$ | $m_3$ |
|---|---|---|---|
| $j_1$ | 0.3 | 0.4 | 0.3 |
| $j_2$ | 0.3 | 0.2 | 0.2 |
| $j_3$ | 0.2 | 0.3 | 0.3 |

Tabela 3.5: Custo de produção ($p_{jmt}$) do produto $j$ na máquina $m$ no período $t$

| | $t_1$ | | | | $t_2$ | | | | $t_3$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $m_1$ | $m_2$ | $m_3$ | | $m_1$ | $m_2$ | $m_3$ | | $m_1$ | $m_2$ | $m_3$ |
| $j_1$ | 0.1 | 0.7 | 0.9 | $j_1$ | 1.7 | 1.8 | 1.7 | $j_1$ | 4 | 1.8 | 1.7 |
| $j_2$ | 0.2 | 0.9 | 0.8 | $j_2$ | 1.7 | 2 | 1.8 | $j_1$ | 4.6 | 1.6 | 1.9 |
| $j_3$ | 0.3 | 0.9 | 0.6 | $j_3$ | 1.9 | 1.5 | 1.6 | $j_1$ | 4 | 2 | 1.8 |

Tabela 3.6: Tempo de preparação ($S_{ijm}$) do produto $i$ ao produto $j$ na máquina $m$

| | $m_1$ | | | | $m_2$ | | | | $m_3$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $j_1$ | $j_2$ | $j_3$ | | $j_1$ | $j_2$ | $j_3$ | | $j_1$ | $j_2$ | $j_3$ |
| $i_1$ | 0 | 41 | 41 | $i_1$ | 0 | 65 | 43 | $i_1$ | 0 | 56 | 44 |
| $i_2$ | 37 | 0 | 63 | $i_2$ | 49 | 0 | 66 | $i_2$ | 48 | 0 | 58 |
| $i_3$ | 37 | 39 | 0 | $i_3$ | 47 | 63 | 0 | $i_3$ | 44 | 53 | 0 |

Tabela 3.7: Custo de preparação ($W_{ijm}$) do produto $i$ ao produto $j$ na máquina $m$

| | $m_1$ | | | | $m_2$ | | | | $m_3$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $j_1$ | $j_2$ | $j_3$ | | $j_1$ | $j_2$ | $j_3$ | | $j_1$ | $j_2$ | $j_3$ |
| $i_1$ | 0 | 41 | 41 | $i_1$ | 0 | 65 | 43 | $i_1$ | 0 | 56 | 44 |
| $i_2$ | 37 | 0 | 63 | $i_2$ | 49 | 0 | 66 | $i_2$ | 48 | 0 | 58 |
| $i_3$ | 37 | 39 | 0 | $i_3$ | 47 | 63 | 0 | $i_3$ | 44 | 53 | 0 |

O exemplo numérico foi resolvido pelo software de otimização *ILOG IBM CPLEX 12.4* por meio da biblioteca *Concert Tecnologic*. A solução obtida é ilustrada na Figura 3.1 (Diagrama de Gantt). Nesta figura, os intervalos em cor roxa, azul e vermelha, representam os processamentos dos lotes dos produtos 1, 2 e 3 respectivamente. Os intervalos hachurados representam os tempos de preparação realizados entre os lotes de diferentes produtos para cada máquina. A sequência de produção no primeiro período inicia com o processamento do lote do produto 2 (azul), depois do tempo de preparação necessário é processado o lote do produto 1 (roxo) e por último neste

período o lote do produto 3 (vermelho) é processado. No segundo período o primeiro lote a processar é o correspondente ao produto 3, já que este foi o último processado no primeiro período. Depois dos tempos de preparação necessários, são processados os lotes dos produtos 2 e 1. Da mesma forma, do segundo período para o terceiro as preparações das máquinas são mantidas, portanto, o primeiro lote a ser processado no terceiro período é o correspondente ao produto 1, logo o lote do produto 3 e por último o lote do produto 2. Obviamente foram realizadas as preparações das máquinas entre os processamentos de cada lote de produção. As dimensões dos lotes e os estoques são mostrados nas Tabelas 3.8 e 3.9 respectivamente. O valor da função objetivo obtida para este exemplo foi $6209,96$ unidades monetárias.

Tabela 3.8: Quantidades do produto $i$ no período $t$

| $x_{jt}$ | $t_1$ | $t_2$ | $t_3$ |
|---|---|---|---|
| $j_1$ | 100 | 131.647 | 86.3529 |
| $j_2$ | 105 | 150 | 142 |
| $j_3$ | 110 | 134 | 103 |

Tabela 3.9: Estoque do produto $i$ no período $t$

| $I_{jt}$ | $t_1$ | $t_2$ | $t_3$ |
|---|---|---|---|
| $j_1$ | 0 | 11.647 | 0 |
| $j_2$ | 0 | 0 | 0 |
| $j_3$ | 0 | 20 | 0 |

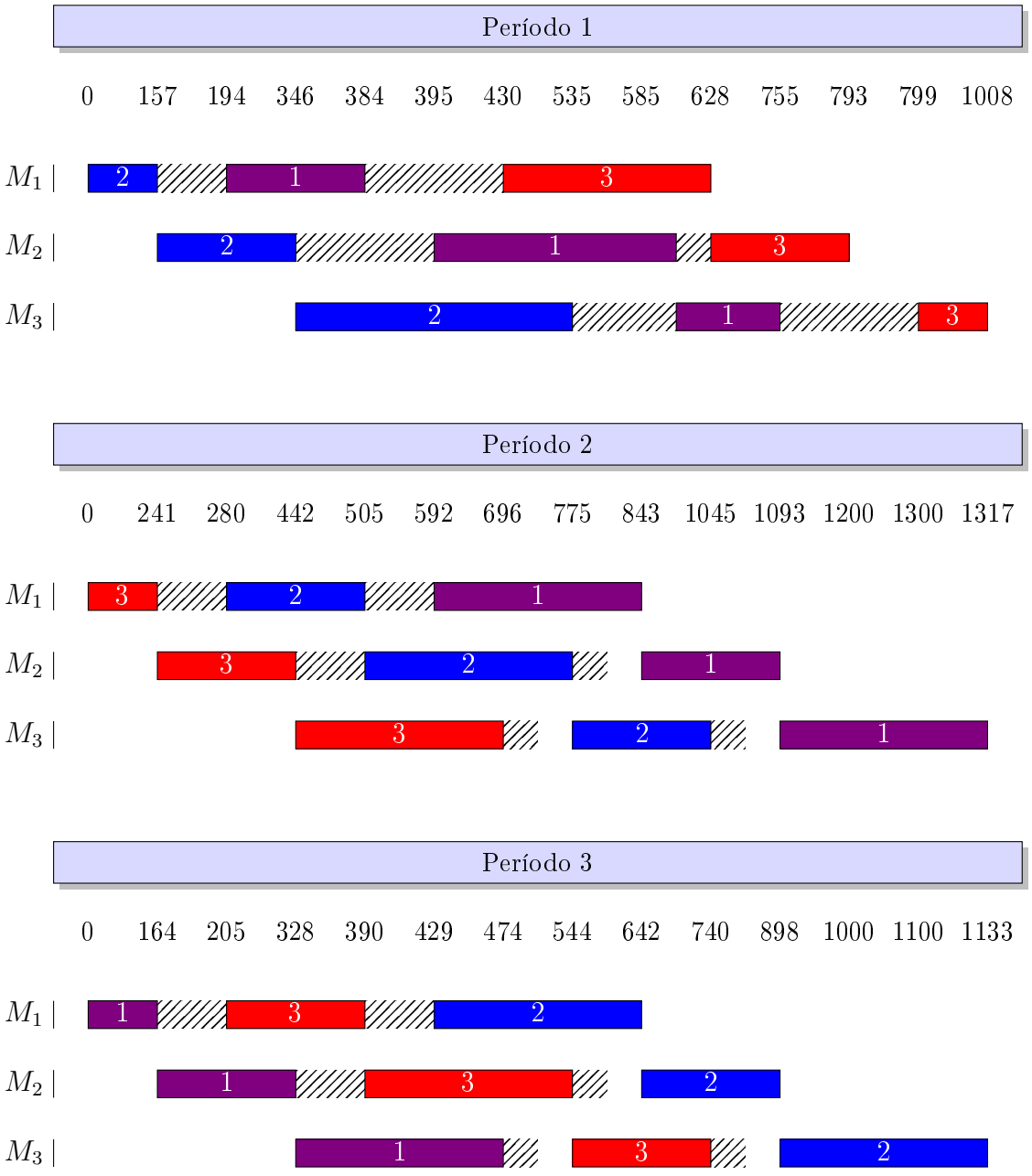



Figura 3.1: Diagrama de Gantt para o exemplo numérico

# Capítulo 4

# Procedimentos de solução da literatura

O modelo matemático adotado nesta dissertação é, na prática, ineficiente quando aplicado a problemas-teste de médio e grande porte, pois não encontrou soluções factíveis em tempo computacional razoável para este tipo de problemas. Dado isto, foram implementados procedimentos heurísticos para tentar encontrar "boas" soluções em tempo razoável.

Este capítulo está dividido em duas seções. Na primeira se descreve uma heurística baseada na estratégia de **Horizonte Rolante** a qual tem sido bastante aplicada a problemas de dimensionamento de lotes. Na segunda seção é descrita a metaheurística **Times Assíncronos** que tem sido aplicada com sucesso em vários problemas de otimização combinatória.

As heurísticas propostas neste trabalho são comparadas com essas heurísticas que, para nosso conhecimento, são as melhores na literatura aplicadas ao problema tratado.

## 4.1 Heurística horizonte rolante

A heurística horizonte rolante *(HR)* é usualmente aplicada a problemas de dimensionamento de lotes nos quais o ambiente é dinâmico. Porém, também tem sido aplicada em vários problemas nos quais o ambiente é estático, onde todos os parâmetros são conhecidos e tem obtido bons resultados (de Araujo et al. [2007]).

Usando a heurística *HR* o esforço computacional é consideravelmente reduzido, já que é utilizada uma estratégia de relaxação de variáveis. Assim, as variáveis binárias são

substituídas por variáveis contínuas em períodos que não são considerados diretamente em determinados passos da heurística (Mohammadi et al. [2010c]).

O procedimento heurístico consta de iterações, onde em cada iteração tenta-se encontrar os valores ótimos para as variáveis de decisão de um determinado período. O procedimento começa pelo primeiro período e avança, de um em um, até chegar ao último.

De acordo com Mercé & Fontan [2003], em cada passo do processo iterativo, que consta de $k$ iterações ($k = 1, ..., T$), o horizonte de planejamento é dividido em três seções: inicial, central e final.

- Seção inicial: consiste de $k-1$ períodos. Nesta seção, as variáveis de decisão correspondentes a esses períodos são completamente congeladas (fixas), de acordo com os valores encontrados em iterações anteriores.

- Seção central: só contém o $k-$ésimo período de planejamento. Para este período o problema é considerado como um todo. Nesta seção todas as variáveis associadas ao período de planejamento são consideradas como binárias. No período de planejamento tratado nesta seção a estratégia de relaxação não é aplicada.

- Seção Final: Esta seção contém os períodos restantes, os quais não são inclusos nas seções anteriores. Do período $k+1$ ao período $T$ é aplicada a estratégia de simplificação, ou seja, todas as variáveis de decisão associadas a estes períodos são relaxadas.

No final de cada iteração $k$, um período é "rolado" para cada seção, assim uma nova iteração pode ser iniciada. O procedimento heurístico é finalizado na iteração $T$, ou seja, quando se chega ao final do horizonte de planejamento.

Neste trabalho, se re-implementou a heurística horizonte rolante proposta por Ramezanian et al. [2012] para o *PDSLFS*, esta heurística é descrita a seguir.

### 4.1.1 Algoritmo heurístico horizonte rolante

Na heurística proposta por Ramezanian et al. [2012], as variáveis de decisão binárias congeladas (fixas) na primeira seção são só as binárias $w$, $\alpha$, $\beta$ e $Y$. A variável $x_{jt}$ correspondente ao dimensionamento de lote estará livre e será determinada através do algoritmo, mas só a associada ao período sobre a seção central. Em cada iteração $k$, as variáveis $x_{jt}$ e $Y$ são determinadas do período 1 até o período $k$ (período que pertence à seção central). Note-se que as variáveis de decisão binárias relacionadas aos períodos 1 até $k-1$ são determinadas nas iterações anteriores. As variáveis correspondentes à

seção final, variáveis associadas aos períodos $k+1$ até $T$, são relaxadas (estratégia de simplificação). Assim, a estratégia de simplificação ignora as variáveis correspondentes à preparação das máquinas (dentre outras) do período $k+1$ até o período $T$. Em cada iteração, as únicas variáveis binárias envolvidas no modelo matemático são as relacionadas com a seção central. Desta forma, o esforço computacional requerido para resolver o problema é consideravelmente reduzido. O procedimento heurístico proposto por Ramezanian et al. [2012] é apresentado no Algoritmo 1.

**Algorithm 1** : Horizonte Rolante

1: $k = 1$;
2: **while** $k < T$ **do**
3: Resolver o modelo com as seguintes suposições:
4: Seção inicial (só fixa as variáveis binárias)
5: $Y_{ijt} = Y^*_{ijt} \quad i, j = 1..N, t = 1..k-1$
6: $w_t = w^*_t \quad t = 1..k-1$
7: $\alpha_{jt} = \alpha^*_{jt} \quad i, j = 1..N, t = 1..k-1$
8: $\beta_{jt} = \beta^*_{jt} \quad i, j = 1..N, t = 1..k-1$
9: Seção final (estratégia de simplificação: todas as variáveis binárias são relaxadas)
10: $0 \leq Y_{ijt} \leq 1 \quad i, j = 1..N, t = k+1..T$
11: $0 \leq w_t \leq 1 \quad t = k+1..T$
12: $0 \leq \alpha_{jt} \leq 1 \quad i, j = 1..N, t = k+1..T$
13: $0 \leq \beta_{jt} \leq 1 \quad i, j = 1..N, t = k+1..T$
14: $k = k + 1$
15: **end while**
16: Mostrar os resultados obtidos pela heurística

## 4.2 Heurística times assíncronos

Tines assíncronos (A-Teams) foi proposto por de Souza & Talukdar [1993]. Esta heurística consiste de agentes que compartilham memória e tem três características fundamentais, as quais são:

- Agentes autônomos: os agentes tomam suas próprias decisões com respeito às soluções: seleção, modificação, destruição, dentre outras.
- Comunicação assíncrona: os agentes podem modificar informação com respeito a soluções nas memórias compartilhadas sem qualquer sincronização entre eles.
- Fluxo de dados cíclico: os agentes tomam, modificam e salvam informação continuamente nas memórias compartilhadas.

Os agentes são normalmente heurísticas ou procedimentos algorítmicos simples, mas metaheurísticas como Algoritmos Genéticos, Busca Tabu, dentre outros também podem ser usados (Belo Filho et al. [2009]).

Neste trabalho, se re-implementa o *A-Teams* proposto por Belo Filho et al. [2012] para o *PDSLFS*, o qual é chamado também de Times Assíncronos Híbrido *(TAH)*.

Os agentes implementados por Belo Filho et al. [2012] estão divididos em agentes construtores, de melhoria e destrutores. Agentes construtores (AgC) são os responsáveis por criar as soluções e inseri-las na memória compartilhada (MC). Os agentes de melhoria (AgM) selecionam soluções da MC e realizam procedimentos para melhorar essas soluções, logo são retornadas aos agentes destrutores (AgD) que analisam e comparam as soluções. Se a solução retornada por um AgM é melhor que alguma solução da MC, esta solução é eliminada e a solução retornada pelo AgM é inserida na MC, este processo é realizado pelo AgD.

Os agentes implementados por Belo Filho et al. [2012] são apresentados a seguir:

### 4.2.1 Agentes construtores

São apresentados dois agentes construtores *(AgC1 e AgC2)*. Ambos determinam a sequência de produção por métodos aproximados enquanto as decisões do dimensionamento de lote são baseadas em uma formulação matemática, a qual será detalhada posteriormente.

**Agente Construtor 1 (AgC1)**: Neste agente construtor, as sequências de produção para todos os períodos são obtidas aleatoriamente. A partir do segundo período até o último, o produto que estiver na última posição na sequência de produção no período anterior é colocado na primeira posição na sequência de produção do período atual (por inserção simples). O propósito deste procedimento é preservar a preparação das máquinas entre os períodos (setup carryover). Fixadas as sequências de produção para todos os períodos, as decisões de dimensionamento de lotes são determinadas por meio da resolução exata do modelo matemático proposto por Ramezanian et al. [2012], mas com algumas modificações as quais serão detalhadas adiante.

**Agente Construtor 2 (AgC2)**: Neste agente, as sequências de produção para todos os períodos são determinadas por meio da heurística *NEH*, considerando alguns critérios, como o *setup carryover*. A heurística *NEH* é um procedimento construtivo de inserção no qual duas sequências são utilizadas, uma sequência de inserção que contêm todos os produtos, e a outra sequência de produção que inicialmente está vazia e logo será preenchida, produto a produto, com os itens da sequência de

inserção, a cada iteração da heurística. Neste procedimento heurístico, a sequência de inserção utilizada é a ordem decrescente dos produtos de acordo aos custos totais de preparação das máquinas. O processo construtivo começa inserindo o primeiro produto da sequência de inserção na sequência de produção. O processo continua inserindo produto a produto da sequência de inserção na sequência de produção, de maneira gulosa, na melhor posição possível, ou seja na posição onde seja gerado o menor custo de preparação. Note que da mesma forma que no AgC1 o *setup carryover* é considerado, portanto, a partir do segundo período, o primeiro produto na sequência de produção será o último produto da sequência de produção do período anterior. Em seguida, o processo continua selecionando os produtos em ordem decrescente da sequência de inserção como foi descrito anteriormente.

A formulação matemática usada pelos agentes construtores para determinar a dimensão dos lotes é derivada do modelo matemático proposto por Ramezanian et al. [2012] (equações: 3.1 - 3.21). Como os $AgC$ determinam as sequências de produção, as quais fazem referência à variável $Y_{ijt}$ ,por meio de métodos heurísticos, esta variável se torna um parâmetro. Os AgC também garantem a preservação da preparação das máquinas entre os períodos, assim, as variáveis $(\alpha, \beta, w, \delta, Y')$ e as restrições correspondentes ao *setup carryover* são desnecessárias. As outras variáveis e restrições são análogas ao modelo matemático original. Assim, a formulação matemática resultante é:

Formulação matemática para o dimensionamento de lotes (note que a variável $Y_{ijt}$ é um parâmetro, e as demais restrições são análogas ao modelo matemático original).

$$Min \sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{t=1}^{T} p_{jmt} \times x_{jt} + \sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{t=1}^{T} h_{jm} \times I_{jt} + \sum_{i=1}^{J}\sum_{j=1}^{J}\sum_{m=1}^{M}\sum_{t=1}^{T} w_{ijm} \times Y_{ijt} \quad (4.1)$$

**s.t.**

$$I_{j(t-1)} + x_{jt} = I_{jt} + d_{jt} \qquad \forall j \in N, t \in T \quad (4.2)$$

$$x_{jt} \leq BigM \times Z_{jt} \qquad \forall j \in N, t \in T \quad (4.3)$$

$$CO_{jmt} = SO_{jmt} + b_{jm} \times x_{jt} \qquad \forall j \in N, m \in M, t \in T \quad (4.4)$$

$$CO_{jmt} \leq C_{mt} \qquad \forall j \in N, m \in M, t \in T \quad (4.5)$$

$$SO_{jmt} \geq CO_{j(m-1)t} \qquad \forall j \in N, t \in Tm = 2, ..., M \quad (4.6)$$

$$SO_{jmt} \geq CO_{imt} + S_{imt} \times Y_{ijt} - BigM(1 - Y_{ijt}) \qquad \forall i, j \in N, i \neq j, m \in M, t \in T \quad (4.7)$$

$$\sum_{i \in J, i \neq j} \sum_{i \in J} Y_{ijt} \geq \sum_{j \in J} Z_{jt} - 1 \qquad \forall t \in T \quad (4.8)$$

$$\sum_{i \in J, i \neq j} Y_{ijt} \leq Z_{jt} \qquad \forall j \in N, t \in T \quad (4.9)$$

$$\sum_{i \in J, i \neq j} Y_{jit} \leq Z_{jt} \qquad \forall j \in N, t \in T \quad (4.10)$$

$$x_{jt}, I_{jt}, CO_{jmt}, SO_{jmt} \geq 0; I_{j0} = 0; \forall j \in N, m \in M, t \in T \quad (4.11)$$

$$Z_{jt} = \{0, 1\}; \forall i, j \in N, t \in T \quad (4.12)$$

### 4.2.2 Agentes de Melhoria

Todos os agentes de melhoria escolhem uma solução da memória compartilhada utilizando a seleção por torneio de tamanho 2, ou seja, são escolhidas duas soluções aleatórias e a melhor delas considerando o valor da função objetivo será a selecionada para ser modificada ou melhorada pelo agente de melhoria.

**Agente de Melhoria 1 (AgSw):** O agente AgSw explora uma vizinhança por troca simples de dois produtos em uma sequência de produção de um dado período. O período e os produtos são selecionados de maneira aleatória. Após a solução e o período serem selecionados, o agente analisa certa quantidade de soluções vizinhas, mas só realizando movimentações na sequência de produção do período previamente selecionado. O agente retornará a melhor solução encontrada. O tipo de movimentação utilizada por este agente é também chamada de vizinhança *swap*.

**Agente de Melhoria 2 (AgIn):** Este agente adota a vizinhança por inserção, da mesma forma que o agente anterior. Após a solução ser selecionada por torneio e o período aleatoriamente, um produto é selecionado e inserido em outra posição na sequência de produção. Tanto o produto como a nova posição são determinados também aleatoriamente. Os produtos entre a atual e nova posição são movimentados a sua posição adjacente em direção à posição atual.

**Agente de melhoria 3 (AgBI):** Similar ao agente anterior, este agente adota a estratégia de vizinhança por inserção, o período e o produto são selecionados aleato-

riamente, porém, neste agente, o produto selecionado é inserido em todas as posições possíveis da sequência de produção. A solução final é dada pela inserção do produto na posição da sequência onde gerou menor custo de preparação das máquinas.

**Agente de melhoria 4 (AgCO):** O último agente que trata sobre o sequenciamento de produtos, procura melhorar uma solução alterando dois períodos consecutivos, onde o primeiro período $t$, sendo $(t < T - 1)$ é determinado aleatoriamente. Logo, é selecionado um produto também de maneira aleatória. Este produto, é colocado na última posição da sequência de produção do período $t$, e colocado na primeira posição da sequência de produção do período $t + 1$. O objetivo principal deste agente é melhorar a solução mantendo a preservação da preparação da máquina entre períodos.

**Agente de melhoria 5 (AgAP):** da mesma forma que os agentes de sequenciamento, os agentes de dimensionamento selecionam uma solução por torneio. Neste agente, um período $t$, onde $(t < T - 1)$, com produção excedente é selecionado aleatoriamente. Um produto entre os que têm estoque positivo é também selecionado de maneira aleatória. Logo, a produção excedente do produto selecionado é postergada ao período imediatamente seguinte com capacidade disponível. Note que as restrições de capacidade devem ser respeitadas, portanto, a quantidade a postergar será o máximo possível, mas dependendo diretamente da capacidade disponível no período que receberá as quantidades. Desta forma, o agente procura reduzir parte ou todo o estoque de um produto em um período determinado.

**Agente de melhoria 6 (AgEP):** Neste agente é selecionado um período $t$ $(t > 1)$ aleatoriamente. É também selecionado um produto qualquer para que sua produção seja adiantada para o período imediatamente anterior. A quantidade de produção a serem adiantada depende diretamente da capacidade disponível no período anterior. Se a capacidade for suficiente, toda a produção do produto será adiantada, se a capacidade não é suficiente, será adiantada uma parte da produção, tal que respeite a restrição da capacidade no período imediatamente anterior.

Baseado nos agentes descritos acima, o Algoritmo 2 apresenta o procedimento do A-Teams. Primeiramente os agentes construtores*(AgC)* preenchem a memória compartilhada *(MC)* com *TamMem* (tamanho da memória) soluções (linhas $2 - 7$). O *AgC1* só constrói uma solução, enquanto que o *AgC2* construirá as $TamMem - 1$ soluções restantes. O agente destruitor *(AgD)* permite inserir só soluções distintas na *(MC)* (linhas $4 - 6$). Em um número máximo de iterações $IterMax$, agentes de melhoria

*(AgM)* manipulam soluções procurando melhorá-las (linhas $8-16$). Os $AgM$ usam o parâmetro $TamViz$ (Tamanho da vizinhança) para limitar o número de soluções analisadas (linha 11). Se uma solução melhor é encontrada por um $AgM$, o $AgD$ substitui a pior solução na $MC$ pela nova solução (linhas $12-15$).

**Algorithm 2** : A_Team

```
1: Criar uma memória compartilhada vazia;
2: while Tamanho_Memoria < TamMem do
3:    Criar uma solução Sol = AgC;
4:    if AgD(Sol) then
5:       Inserir Sol em MC;
6:    end if
7: end while
8: while Iteracoes ≤ IterMax do
9:    Escolha uma solução Sol da MC por torneio de tamanho 2;
10:   Escolha aleatoriamente um AgM;
11:   Obter uma nova solução NSol = AgM(Sol, TamViz);
12:   if AgD(NSol) then
13:      Eliminar a pior solução da MC;
14:      Inserir NSol em MC;
15:   end if
16: end while
17: Retorne a melhor solução da MC;
```

# Capítulo 5

# Procedimentos de solução propostos

Neste capítulo são apresentados os métodos de solução propostos, os quais são baseados na metaheurística Iterated Greedy *(IG)* de Ruiz & Stützle [2007], na heurística Horizonte Rodante *(HR)* de Ramezanian & Saidi-Mehrabad [2013] e na heurística Fix and Otimize *(F&O)* de Pochet & Wolsey [2006], cujos detalhes serão descritos nas seções seguintes.

## 5.1 Heurística horizonte rolante + Iterated Greedy (IG+HR)

Inicialmente a heurística Horizonte Rolante foi usada por muitos autores para superar a intratabilidade computacional em relação ao tempo para problemas de grande tamanho. Como foi dito anteriormente, a estratégia utilizada por este método consiste na substituição da maior parte de variáveis de decisão binária por variáveis contínuas. Isto é feito decompondo o modelo matemático em vários submodelos com um número menor de variáveis de decisão binárias. Porém, se espera que para problemas grandes exista uma intratabilidade computacional. Para tentar superar esta intratabilidade, a heurística Horizonte Rolante *(HR)* tem sido combinada com o Iterated Greedy *(IG)* resultando no *IG+HR* para facilitar a notação. O problema continua sendo resolvido com a estratégia da *HR*, em $T$ iterações e, em uma iteração específica $k$ o valor da variável binária $Y$ correspondente ao sequenciamento dos produtos é determinada com o *IG*. Como na versão inicial da $HR$, o $IG + HR$ é dividido em três seções: inicial, central e final.

- Seção inicial: contém $k-1$ períodos. Nesta seção, as variáveis binárias $w$, $\alpha$, $\beta$ e $Y$ correspondentes a estes períodos (1 a $k-1$) são completamente congeladas (fixadas) de acordo com os valores encontrados em iterações anteriores.
- Seção Central: só contém o $k-$ésimo período do planejamento. As variáveis $w$, $\alpha$, $\beta$ associadas a este período são consideradas como binárias e determinadas otimamente, exceto a variável $Y$ que corresponde ao sequenciamento de produtos sendo determinada pelo $IG$.
- Seção final: esta seção contém os períodos que não foram incluso nas seções anteriores, para estes períodos é aplicada a estratégia de simplificação, ou seja, as variáveis binárias $w$, $\alpha$, $\beta$, $Y$ associadas a estes períodos são relaxadas.

Note que a única seção diferente da HR é a seção central, na qual o IG é executado para determinar o sequenciamento de produtos para o período tratado nesta seção. As seções inicial e final são mantidas. A seguir, o Algoritmo 3 *(IG+HR)* é apresentado e posteriormente o IG usado neste procedimento é detalhado.

**Algorithm 3** : Horizonte Rolante + Iterated Greedy

1: $k = 1$;
2: **while** $k < T$ **do**
3: Resolver o modelo com as seguintes suposições;
4: Iterated_Greedy($Y$); //$Y$ é fixa neste passo, mas só a sequência de produção correspondente ao período $k$;
5: Seção inicial (só fixa as variáveis binárias)
6: $Y_{ijt} = Y^*_{ijt} \quad i,j = 1..N, t = 1..k-1$
7: $w_t = w^*_t \quad t = 1..k-1$
8: $\alpha_{jt} = \alpha^*_{jt} \quad i,j = 1..N, t = 1..k-1$
9: $\beta_{jt} = \beta^*_{jt} \quad i,j = 1..N, t = 1..k-1$
10: Seção final (estratégia de simplificação: todas as variáveis binárias são relaxadas)
11: $0 \leq Y_{ijt} \leq 1 \quad i,j = 1..N, t = k+1..T$
12: $0 \leq w_t \leq 1 \quad t = k+1..T$
13: $0 \leq \alpha_{jt} \leq 1 \quad i,j = 1..N, t = k+1..T$
14: $0 \leq \beta_{jt} \leq 1 \quad i,j = 1..N, t = k+1..T$
15: $k = k+1$;
16: **end while**
17: Mostrar o resultados obtido pela heurística

### 5.1.1 Iterated Greedy

A metaheurística IG desde 1995 já tinha sido aplicada com sucesso por Jacobs & Brusco [1995], Vasko [1984] e Marchiori & Steenbeek [2000] no Set Covering Problem. Mais

recentemente, Ruiz & Stützle [2007] aplicaram o IG ao *permutation flowshop scheduling problem (PFSP)*. Esses autores foram os primeiros a aplicar este método a um problema de *scheduling*, obtendo bons resultados.

Em síntese, o procedimento do *IG* proposto por Ruiz & Stützle [2007] para o *PFSP* é da seguinte forma: gera-se uma solução (sequência de produtos) iterativamente por meio de uma estratégia construtiva de duas fases: destruição e construção. Na fase de destruição, alguns componentes (produtos) de uma solução são removidos, resultando em uma solução incompleta. A fase construtiva aplica um procedimento construtivo guloso para reconstruir a solução incompleta resultante na fase anterior. Uma vez que a solução tem sido completada, um critério de aceitação decide se a solução construída irá substituir a solução atual. O processo construtivo do *IG* é similar ao processo construtivo guloso da heurística NEH (Nawaz et al. [1983]). O processo geral do *IG* é também relacionado ao processo da *Iterated Local Search (ILS)*, só que em lugar de iterar sobre a busca local como é feito no *ILS*, o *IG* itera de maneira análoga em um processo construtivo (Ruiz & Stützle [2007]). O *IG* utilizado neste trabalho é o proposto por Ruiz & Stützle [2007], o qual é apresentado no Algoritmo 4 recebendo como parâmetros: número de iterações do algoritmo ($Iter_{totais}$), parâmetro de destruição ($d$) e iterações da busca local ($Iter_{BL}$). Os procedimentos deste algoritmo, serão descritos nas seguintes subseções.

**Algorithm 4** : IG($Iter_{totais}$, $d$, $Iter_{BL}$)

1: $\pi$ := Construção_Sequencia_Inicial();
2: $\pi^*$ := $\pi$;
3: iterações:=0;
4: **while** iterações $\leq Iter_{totais}$ **do**
5: $\pi$ := Destruição_Construção($\pi, d$);
6: $\pi$ := Busca_Local($\pi, Iter_{BL}$);
7: **if** $f_{cpm}(\pi) < f_{cpm}(\pi^*)$ **then**
8: $\pi^*$ := $\pi$;
9: **else**
10: $\pi$ := $\pi^*$;
11: **end if**
12: iterações:= iterações +1;
13: **end while**
14: **return** $\pi^*$;

#### 5.1.1.1 Inicialização das sequências iniciais de produção

Na construção das sequências de produção é adotada a ideia de Mohammadi et al. [2010c] que propõe uma heurística baseada na estratégia de horizonte rolante junto com a heurística NEH, a qual tem sido bastante utilizada na literatura para problemas de sequenciamento, por ser fácil de implementar e produzir bons resultados. A heurística NEH utiliza duas sequências, a de inserção e a de produção. O procedimento consiste em construir a sequência de produção tendo como base a sequência de inserção onde várias soluções parciais são testadas.

Na heurística NEH é importante a definição de uma "boa" sequência de inserção, pois desta depende, em grande parte, o sucesso do procedimento. Neste trabalho, a sequência de inserção é definida como a ordem decrescente dos produtos de acordo ao custo de preparação das máquinas.

O procedimento heurístico é aplicado para todos os períodos, e, dado que o critério a minimizar é o custo de preparação das máquinas, incluso entre períodos, a partir do segundo período, o primeiro produto na sequência de inserção será o último produto da sequência de produção do período anterior.

Os seguintes passos correspondem ao processo construtivo da heurística:

Seja $\pi_{Ins}$ a sequência de inserção dada por $\pi_{Ins} = \pi_1, \pi_2, ..., \pi_N$. E $\pi_{Prod}$ a sequência de produção inicialmente vazia.

- Atualizar $\pi_{Ins}$ considerando a $\pi_{Prod}$ do período anterior.
- Para cada produto $\pi_i$ que pertence à sequência de inserção, onde $i = 1, 2, ..., N$.
  - Inserir o produto $\pi_i$ em todas as posições possíveis da sequência de produção $\pi_{Prod}$.
    - Calcular o custo de preparação das máquinas $f_{cpm}$ (critério a minimizar) para cada sequência parcial $\pi_{Prod}$.
- Inserir $\pi_i$ na posição da sequência de produção que gerou menor custo de preparação.
- retornar ao ponto 2 até que a sequência de inserção esteja vazia.

Finalizado o procedimento heurístico, teremos uma sequência de produção para cada período, preservando a preparação das máquinas entre estes. Um exemplo de construção de uma solução é mostrado na Subseção 5.3.2.1.

#### 5.1.1.2 Procedimento de destruição - construção

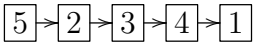
O procedimento de destruição é aplicado a cada sequência de produção $\pi$ de $N$ produtos. Este procedimento remove aleatoriamente $d$ diferentes produtos de $\pi$. Isto produz duas subsequências, a primeira ($\pi_D$) com $N-d$ produtos e a segunda com os produtos removidos, a qual é definida como $\pi_R$. $\pi_R$ contém os produtos que serão reinseridos em $\pi_D$ para completar a sequência de produção. O procedimento de construção inicia com a subsequência $\pi_D$ e realiza $d$ iterações, sendo que a cada iteração um produto de $\pi_R$ é reinserido em $\pi_D$. Este processo inicia inserindo o primeiro produto de $\pi_R$ em todas as $N-d+1$ possíveis posições de $\pi_D$. A melhor posição para este produto na sequência $\pi_D$ é aquela que gera o menor custos de preparação das máquinas ($f_{cpm}$).

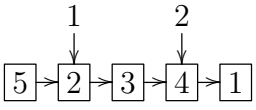
Este procedimento é repetido até que a subsequência $\pi_R$ esteja vazia. O procedimento de destruição - construção é aplicado à sequência de produção de cada período. O Algoritmo 5 apresenta este procedimento, o qual recebe como parâmetros uma sequência de produtos $\pi$ e o parâmetro de destruição $d$.

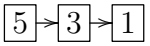
**Algorithm 5** : Destruição_Construção($\pi, d$);

1: $\pi$ := sequência de produção inicial.
2: $\pi_D := \pi$;
3: $\pi_R := \emptyset$;
4: **for** $i := 1$ **to** $d$ **do**
5: $\quad \pi_D$ := remova um produto aleatoriamente de $\pi_D$ e insira este em $\pi_R$;
6: **end for**
7: //os tamanhos de $\pi_D$ e $\pi_R$ são $N-d$ e $d$ respectivamente.
8: **for** $i := 1$ **to** $d$ **do**
9: $\quad \pi_D$ := melhor sequência obtida após inserir o produto $\pi_R(i)$ em todas as posições possíveis de $\pi_D$;
10: **end for**
11: **if** $f_{cpm}(\pi_D) < f_{cpm}(\pi)$ **then**
12: $\quad \pi := \pi_D$;
13: **end if**
14: **return** $\pi$;

A fim de facilitar o entendimento do procedimento Destruição-Construção, na Figura 5.1 mostra-se um exemplo ilustrativo. Os dados necessários envolvidos neste exemplo se encontram no apêndice A.

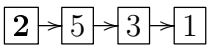

Sequência de produção inicial

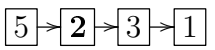

Selecionando $d = 2$ produtos aleatoriamente

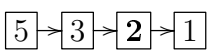

Sequência $\pi_D$ a ser reconstruida

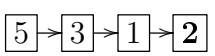

Sequência $\pi_R$, produtos a ser reinseridos em $\pi_D$

Após inserir o produto $\pi_R(2)$ em todas as posições possíveis de $\pi_D$ obtemos as sequências:

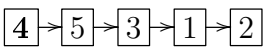

$CPM = 462$

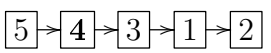

$CPM = 462$

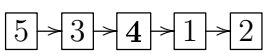

**CPM = 446***

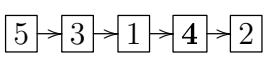

$CPM = 462$

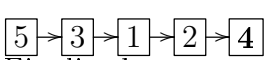
Após inserir o produto $\pi_R(4)$ em todas as posições possíveis de $\pi_D$ obtemos as sequências:


$CPM = 603$


$CPM = 613$


$CPM = 601$


$CPM =$ **578***


$CPM = 624$

Finalizado o processo temos encontrado a melhor sequencia de produção, sendo aquela que obteve o menor custo de preparação das máquinas (**578**)

Figura 5.1: Procedimento de Destruição - Construição

#### 5.1.1.3 Busca local

A sequência retornada pelo procedimento de destruição - construção é melhorada pelo procedimento de busca local *(BL)*. Este procedimento é baseado nos movimentos de troca simples e inserção. Esta *BL* é uma variante do método de *Variable Neighborhood Search (VNS)* denominada como *Reduced VNS (RVNS)* proposta por Hansen et al. [2010] para resolver o *PFSP*. O movimento de troca simples seleciona aleatoriamente dois produtos na sequência de produção e troca suas posições. O movimento de inserção remove aleatoriamente um produto de sua posição inicial e o insere em todas as possíveis posições. A função a minimizar na *BL* é também o custo de preparação das máquinas ($f_{cpm}$). A *BL* é detalhada no Algoritmo 6, o qual recebe como parâmetros: uma solução *SOL* e o máximo número de iterações ($Iter_{BL}$) usado como critério de parada.

Note-se que neste procedimento é recebida como parâmetro uma solução *SOL* e

esta é um objeto o qual contém uma sequencia de produção $\pi$ para cada período e os respectivos tamanhos de lotes de cada produtos.

**Algorithm 6** : Busca_Local($SOL, Iter_{BL}$)

1: **for** $l := 1$ **to** $Iter_{BL}$ **do**
2: Loop := 1, Tipo_Viz := 1
3: **while** Loop := 1 **do**
4: $\phi := \pi$
5: **if** Tipo_Viz := 1 **then**
6: $\phi$ := Sequência obtida pela troca de dois produtos selecionados aleatoriamente;
7: **else**
8: $\phi$ := Melhor sequência obtida após inserir um produto aleatoriamente selecionado em todas as posições possíveis;
9: Loop := 0;
10: **end if**
11: **if** $f_{cpm}(\phi) < f_{cpm}(\pi)$ **then**
12: $\pi := \phi$; //$SOL$ é atualizada neste passo.
13: Loop := 1, Tipo_Viz := 1
14: **else**
15: Tipo_Viz := 0
16: **end if**
17: **end while**
18: **end for**
19: **return** $SOL$;

Para um melhor entendimento da $BL$, as Figuras 5.2 e 5.3 mostram exemplos dos movimentos de troca simples e inserção, respectivamente.

Na Figura 5.2, a qual representa uma troca simples, foram selecionados aleatoriamente os produtos 1 e 4, primeiro e quarto na sequência de produção. Posteriormente, estes produtos trocam suas posições, ficando o produto 4 e o produto 1 na primeira e na quarta posição, respectivamente. Na Figura, a sequência do lado esquerdo da seta representa a sequência original onde os produtos (em cor vermelha) foram selecionados e a sequência do lado direito da seta é a resultante depois da troca.

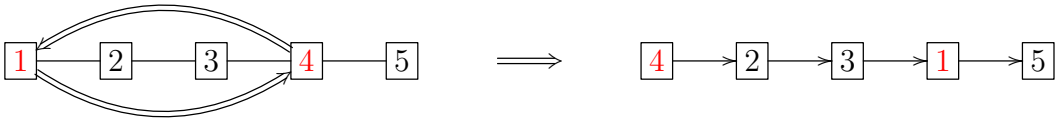



Figura 5.2: Busca Local:Troca simples

A Figura 5.3 correspondente a uma busca local por inserção, dado que a sequência de produção contém 5 produtos, são realizadas quatro inserções. Note que nesta figura as sequências à esquerda da seta representam as sequências iniciais, enquanto as sequências do lado direito da seta representam a sequência resultante depois da inserção. Os produtos em cor vermelha são os produtos envolvidos na movimentação e as setas sobre estes produtos mostram a posição na qual serão situados. Para o exemplo, se selecionou aleatoriamente o produto 3, o qual está situado na terceira posição da sequência de produção. Este produto deverá ser inserido em todas as posições possíveis da sequência.

Na primeira inserção, o produto 3 é inserido na quarta posição da sequência de produção, nesta posição encontra-se o produto 4, o qual é movimentado uma posição adjacente ao produto 3, dado que os produtos 3 e 4 estão situados um do lado do outro esta movimentação é uma troca simples. Na segunda inserção, o produto 3 é inserido na quinta posição da sequência de produção, nesta posição encontra-se o produto 5, assim, o produto 3 é inserido na quinta posição e os produtos 4 e 5 são movimentados uma posição à esquerda (em direção á anterior posição do produto 3). Na terceira inserção, o produto 3 é inserido na segunda posição da sequência, dado que o produto 3 esta situado na terceira posição, este movimento é uma troca simples. Como última inserção, o produto 3 será inserido na primeira posição da sequência de produção, neste caso, os produtos 1 e o 2 situados na primeira e segunda posição da sequência respectivamente, serão movimentados uma posição adjacente (em direção à posição anterior do produto 3).

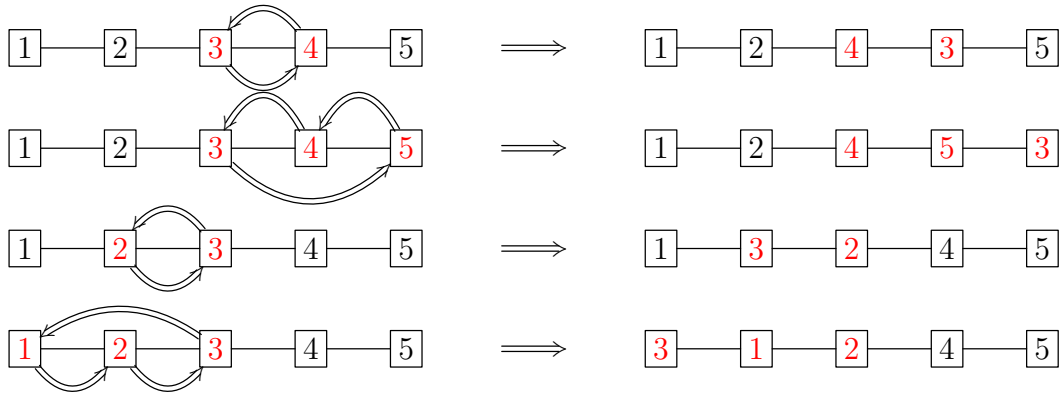



Figura 5.3: Busca Local: Inserção

Note que os procedimentos de Destruição-Construção e Busca Local são executados para cada sequência de produção,ou seja, estes procedimentos são realizados para todos os períodos.

## 5.2 Heurística Iterated Greedy com melhoria do Dimensionamento de Lotes (IG+MDL)

Como se pode notar, o IG proposto por Ruiz & Stützle [2007] procura melhorar o sequenciamento dos produtos. Dado que no *PDSLFS* é necessário melhorar também o dimensionamento de lotes, neste trabalho foram adotadas algumas estratégias para este propósito; no caso as propostas por Gopalakrishnan et al. [2001], as quais consistem em adiantar e postergar produções entre períodos. Nas seções seguintes a integração desses métodos ao *IG* é detalhada.

## 5.3 IG+MDL proposto para o PDSLFS

O *IG+MDL* proposto tenta melhorar, iterativamente, uma solução utilizando quatro diferentes procedimentos: destruição - construção, busca local, melhoria do dimensionamento de lotes *MDL* e o critério de aceitação. O *IG+MDL* gera uma solução inicial por meio de um procedimento construtivo. Logo, aplica iterativamente os quatro procedimentos citados anteriormente até que a condição de parada seja satisfeita. Os métodos destruição - construção e busca local são usados para gerar uma boa sequência de produção. O *MDL* é aplicado à melhor sequência de produção obtida pela busca local a fim de melhorar as quantidades a serem produzidas. Em cada iteração, e por meio do critério de aceitação, a solução a ser considerada na próxima iteração é eleita entre a solução atual e a solução obtida pelo *MDL*.

O Algoritmo 7 apresenta o funcionamento da heurística *IG+MDL* e seus componentes. Ela recebe como parâmetros: Número máximo de iterações do algoritmo ($It\_Max$), parâmetro de destruição ($d$) , número de iterações da busca local ($It\_Bl$) e o número máximo de iterações para o método de melhoria do dimensionamento de lotes ($It\_Dl$). A heurística inicia gerando uma solução inicial (linha 1). Se é construída uma solução infactível, é necessário usar a formulação matemática proposta por Ramezanian et al. [2012] para obter uma solução factível (linhas 2 - 4), da mesma forma que nos agentes construtores do *A-Teams*. As iterações do IG são executadas nas linhas 7 a 17, até que o critério de parada seja satisfeito. Nas linhas 8 e 9 são executados os procedimentos de destruição - construção e busca local respectivamente. O $MDL$ é executado na linha 10. Nas linhas 11 até 15 é verificado o critério de aceitação. Finalmente, a melhor solução obtida pelo algoritmo é retornada, linha 18.

**Algorithm 7** : IG+MDL($It\_Max$, $d$, $It\_Bl$, $It\_Dl$)

```
1:  SOL := Construção_Solução_Inicial();
2:  if Solução_Infatível(SOL) then
3:     SOL := Formulação_Matemática(SOL);
4:  end if
5:  SOL* := SOL;
6:  iterações:=0;
7:  while iterações ≤ It_Max do
8:     SOL := Destruição - Construção(SOL, d);
9:     SOL := Busca_Local(SOL, It_Bl);
10:    SOL := Melhoria_Dimensionamento_Lotes(SOL, It_Dl);
11:    if f(SOL) < f(SOL*) then
12:       SOL* := SOL;
13:    else
14:       SOL := SOL*;
15:    end if
16:    iterações:= iterações +1;
17: end while
18: return SOL*;
```

Para facilitar a explicação dos métodos utilizados pelos algoritmos, é necessário definir o formato de representação de uma solução e a correspondente estrutura de dados utilizada, assim, na seção seguinte (5.3.1) mostra-se a representação de uma solução usada pelos algoritmos implementados nesta dissertação. Na seção 5.3.2 mostra-se como uma solução inicial é construída. Nas seções 5.3.3, 5.3.4 e 5.3.5, mostra-se como uma solução é melhorada aplicando os procedimentos de destruição-construção, busca local e melhoria do dimensionamento de lotes, respectivamente.

### 5.3.1 Representação da solução

Uma solução é representada por $T$ (números de períodos) matrizes de ordem $3 \times N$ (onde $N$ é o número de produtos). A primeira linha de cada matriz representa a sequência de produção, a segunda e terceira linha representam as quantidades produzidas e o estoque de cada produto respectivamente. Na Figura 5.4 mostra-se um exemplo de uma solução para dois períodos ($t_1$ e $t_2$) e três produtos ($j_1, j_2$, e $j_3$). A sequência de produção no primeiro período é $(j_3, j_2, j_1)$ e no segundo é $(j_1, j_2, j_3)$. $DL_{jt}$ e $Est_{jt}$ representam, respectivamente, o dimensionamento de lote e o estoque para cada produto $j$ no período $t$.

| | $t_1$ | | | $t_2$ | | |
|---|---|---|---|---|---|---|
| Sequência de produção | $j_3$ | $j_2$ | $j_1$ | $j_1$ | $j_2$ | $j_3$ |
| Dimensão do Lote | $DL_{31}$ | $DL_{21}$ | $DL_{11}$ | $DL_{12}$ | $DL_{22}$ | $DL_{32}$ |
| Estoque | $Est_{31}$ | $Est_{21}$ | $Est_{11}$ | $Est_{12}$ | $Est_{22}$ | $Est_{32}$ |

Figura 5.4: Representação da Solução

Dada a representação de uma solução, os procedimentos da heurística $IG+MDL$ proposta são descritos a seguir.

### 5.3.2 Construção da solução inicial

A construção de uma solução inicial é feita em duas fases. Na primeira fase é construída a sequência de produção para cada período. Na segunda fase, considerando a sequência de produção obtida na primeira fase, é determinado o dimensionamento de lotes.

#### 5.3.2.1 Inicialização das sequências de produção

A inicialização da sequência de produção é obtida da mesma maneira que o *IG*, fazendo uso da heurística *NEH*, como na Subseção 5.1.1.1. Para facilitar o entendimento, um exemplo é mostrado a seguir. Os dados para este exemplo se encontram no Apêndice A.

A Tabela A.7 (no Apêndice A) mostra os custos de preparação das máquinas. Baseada nestes a sequência de inserção é estabelecida.

Para determinar a sequência de inserção $\pi_{Ins}$ calcula-se os custos de preparação de cada produto:

$W_j = \sum_{i=1}^{N} \sum_{m=1}^{M} w_{ijm} \quad \forall j = 1, 2, ...N$

$W_{j_1} = (0+26+38+25+13)+(0+21+58+32+54)+(0+45+23+45+10) = 390$

$W_{j_2} = (42+0+17+22+25)+(42+0+26+56+49)+(25+0+29+54+26) = 413$

$W_{j_3} = (21+42+0+55+47)+(41+10+0+27+41)+(47+32+0+51+23) = 437$

$W_{j_4} = (25+52+64+0+61)+(51+47+23+0+61)+(29+51+34+0+41) = 539$

$W_{j_5} = (13+58+41+37+0)+(26+26+19+27+0)+(15+37+13+61+0) = 373$

Desta forma, a sequência de inserção é : $\pi_{Ins} = (j_4, j_3, j_2, j_1, j_5)$.

Inicialmente a sequência de produção $\pi_{Prod}$ tem custo 0 já que não contém nenhum produto.

Seja $f_{cpm}$ o custo de preparação da máquina resultante da sequência de produção $\pi_{Prod}$. Inicialmente o primeiro produto da sequência de inserção $(j_4)$ é inserido em $\pi_{Prod}$ obtendo $\pi_{Prod} = (j_4)$ com custo $f_{cpm} = 0$, já que a máquina está previamente preparada para este produto. Continuando com o processo, o segundo produto na sequência de inserção $(j_3)$ é inserido em todas as posições possíveis da sequência de produção. O $f_{cpm}$ é calculado para cada sequência parcial resultante como segue:

$f_{cpm:43} = j_4 \rightarrow j_3 = 55 + 27 + 51 = 133.$
$f_{cpm:34} = j_3 \rightarrow j_4 = 64 + 23 + 34 = 121.$

A última sequência é selecionada ($\pi_{Prod} = j_3 \rightarrow j_4$) já que obteve o menor custo $f_{cpm:34}$. O próximo produto a inserir é o $j_2$, e de igual maneira calcula-se o custo de cada sequência parcial resultante. Ao inserir este produto em todas as posições possíveis da sequência de produção temos:

$f_{cpm:234} = j_2 \rightarrow j_3 \rightarrow j_4 = 42 + 10 + 32 + f_{cpm:34} = 205.$
$f_{cpm:324} = j_3 \rightarrow j_2 \rightarrow j_4 = 17 + 26 + 29 + 52 + 47 + 51 = 222.$
$f_{cpm:342} = j_3 \rightarrow j_4 \rightarrow j_2 = f_{cpm:34} + 22 + 56 + 54 = 253.$

È selecionada a sequência $\pi_{Prod} = j_2 \rightarrow j_3 \rightarrow j_4$ a qual apresenta um menor custo. O processo continua e o produto a inserir é o $j_1$.

$f_{cpm:1234} = j_1 \rightarrow j_2 \rightarrow j_3 \rightarrow j_4 = 42 + 42 + 25 + f_{cpm:234} = 314.$

$f_{cpm:2134} = j_2 \rightarrow j_1 \rightarrow j_3 \rightarrow j_4 = 26 + 21 + 45 + 21 + 41 + 47 + f_{cpm:34} = 322.$

$f_{cpm:2314} = j_2 \rightarrow j_3 \rightarrow j_1 \rightarrow j_4 = 42 + 10 + 32 + 38 + 58 + 23 + 25 + 51 + 29 = 308.$

$f_{cpm:2341} = j_2 \rightarrow j_3 \rightarrow j_4 \rightarrow j_1 = f_{cpm:234} + 25 + 32 + 45 = 282.$

A sequência $\pi_{Prod} = j_2 \rightarrow j_3 \rightarrow j_4 \rightarrow j_1$ gerou menor custo ($f_{cpm:2341} = 282$), portanto essa é selecionada para inserir o próximo produto, $j_5$:

$f_{cpm:52341} = j_5 \rightarrow j_2 \rightarrow j_3 \rightarrow j_4 \rightarrow j_1 = (13 + 54 + 10) + f_{cpm:2341} = 359.$

$f_{cpm:25341} = j_2 \rightarrow j_5 \rightarrow j_3 \rightarrow j_4 \rightarrow j_1 = (58 + 26 + 37) + (47 + 41 + 23) + (64 + 23 + 34) + (25 + 32 + 45) = 455$

$f_{cpm:23541} = j_2 \rightarrow j_3 \rightarrow j_5 \rightarrow j_4 \rightarrow j_1 = (42 + 10 + 32) + (41 + 19 + 13) + (61 + 61 + 41) + (25 + 32 + 45) = 455.$

$f_{cpm:23451} = j_2 \rightarrow j_3 \rightarrow j_4 \rightarrow j_5 \rightarrow j_1 = (42 + 10 + 32) + (64 + 23 + 34) + (37 + 27 + 61) + (13 + 54 + 10) = 407$

$f_{cpm:23415} = j_2 \rightarrow j_3 \rightarrow j_4 \rightarrow j_1 \rightarrow j_5 = f_{cpm:234} + (25 + 32 + 45) + (13 + 26 + 15) = 361.$

Desta forma a sequência de produção tem sido determinada para o primeiro período, sendo: $f_{cpm:52341} = j_5 \rightarrow j_2 \rightarrow j_3 \rightarrow j_4 \rightarrow j_1$.

Para manter a preservação da preparação das máquinas entre os períodos, a partir do segundo período, o primeiro produto na sequência de inserção é o último produto na sequência de produção do período anterior, desta forma a sequência de inserção para o segundo período é: $\pi_{Ins} = (j_1, j_4, j_3, j_2, j_5)$ e a sequência de produção obtida após o processo heurístico é: $\pi_{prod} = j_1 \rightarrow j_5 \rightarrow j_2 \rightarrow j_3 \rightarrow j_4$, coincidentemente o último produto desta sequência de produção é o produto inicial da sequência de inserção. Assim, esta não necessita ser alterada, portanto para o terceiro período a sequência de produção obtida é: $\pi_{prod} = j_4 \rightarrow j_1 \rightarrow j_3 \rightarrow j_5 \rightarrow j_2$. Finalizado o processo heurístico para todos os períodos, a inicialização do dimensionamento de lotes pode ser iniciada, isto é explicado na seguinte subseção.

#### 5.3.2.2 Inicialização do dimensionamento de lotes

Considerando a sequência de produção obtida na fase anterior, o dimensionamento de lotes é realizado em ordem decrescente (dimensionamento regressivo), começando pelo último período e terminando no primeiro considerando dois casos: se a capacidade de um período $t$ é suficiente para produzir todas as quantidades equivalentes à demanda de todos os produtos deste mesmo período $t$, a dimensão do lote de cada produto é igual à demanda do mesmo, esta é a política lote por lote. Por outro lado, se a capacidade de um período $t$ não é suficiente para produzir as quantidades de todos os produtos correspondentes à demanda do mesmo período $t$, é realizada a política lote por lote iniciando no primeiro produto na sequência de produção e avançando progressivamente (produto a produto) nesta sequência, até que a capacidade o permita. Os produtos não produzidos no período $t$ serão produzidos no período imediatamente anterior que possua capacidade disponível. Este procedimento é similar ao utilizado por Shim et al. [2011]. Em alguns casos (poucos), com este procedimento não se obtém uma solução factível, dado que as capacidades dos primeiros períodos podem ser insuficientes. Quando isto ocorre, utiliza-se a formulação matemática proposta por Ramezanian et al. [2012] (equações 4.1 - 4.12) para determinar o dimensionamento de lotes, já que são usadas as sequências de produção obtidas pela heurística NEH, sendo o procedimento similar ao realizado pelo agente construtor $AgC2$ do *A-Teams*.

O Algoritmo 8 apresenta a inicialização do dimensionamento de lotes (IDL), o qual recebe como parâmetro as sequências de produção de todos os períodos. Neste algoritmo, as linhas 1, 2 e 18, mostram que o dimensionamento de lotes é realizado de maneira regressiva, iniciando com o último período de planejamento e terminando no primeiro. Após definido o período ao qual se estabelecerão os dimensionamentos de lotes, as linhas 3, 4 e 12 indicam a ordem dos produtos para estabelecer a dimensão de seus lotes. Como se observa nesta linhas, o dimensionando inicia no primeiro produto na sequência de produção e continua dimensionando progressivamente até chegar ao último produto desta sequência. A linha 5 testa se existe capacidade suficiente para produzir as quantidades equivalentes demandadas do produto $j$ no período $t$, se existe tal capacidade, na linha 6 se define a dimensão do lote do produto $j$ no período $t$ ($DL_{jt}$) igual à demanda do mesmo produto $j$ no mesmo período $t$ ($d_{jt}$). Na linha 7, atualiza-se a capacidade, note que após realizado um dimensionamento, a capacidade no período $t$ é reduzida. Se no teste da linha 5 verifica-se que existe uma capacidade insuficiente para dimensionar o lote do produto $j$ no período $t$, na linha 9 identifica-se esse produto ($i$) e na linha 10 se interrompe referente ao dimensionamento, levando

o procedimento à modificação das demandas. Dado que é obrigatório atender toda a demanda, a partir do produto $i$ até o último produto na sequência de produção, os quais não foram produzidos no período $t$ por capacidade insuficiente, estes deverão ser processados no período imediatamente anterior a $t$ com capacidade disponível. Para isto, a demanda de cada produto $i$ no período $t$ ($d_{it}$) é transferida para o período anterior $t-1$. Este procedimento de transferência de demanda a um período anterior está representado nas linhas 14 a 17. Finalmente na linha 20 retorna-se uma solução.

**Algorithm 8** : $IDL(SOL)$;

```
1:  t := T;
2:  while t ≥ 1 do
3:     j := 1, i := N + 1;
4:     while j ≤ N do
5:        if Capacidade_Suficiente(j, t) then
6:           DL_jt = d_jt; //SOL é atualizada neste passo.
7:           Atualizar_Capacidade(t);
8:        else
9:           i := j;
10:          break;
11:       end if
12:       j := j + 1;
13:    end while
14:    while i ≤ N do
15:       d_i(t−1) := d_i(t−1) + d_it
16:       i := i + 1;
17:    end while
18:    t := t − 1;
19: end while
20: return SOL;
```

### 5.3.3 Procedimento de destruição - construção

O procedimento de destruição - construção é realizado da mesma forma que o algoritmo *IG+HR*, como se descreve na subseção 5.1.1.2.

### 5.3.4 Busca local

A busca local (BL) é a mesma utilizada no algorítimo *IG+HR*, a qual é detalhada no Algoritmo 6 da subseção 5.1.1.3.

Note que os procedimentos de Destruição-Construção e Busca Local são executados para todas as sequências de produção pertencentes à solução.

### 5.3.5 Melhoria do dimensionamento de lotes

A Melhoria do Dimensionamento de lotes *(MDL)* consiste de dois métodos: Método para Adiantar a Produção *MAP* e Método para Posterga a Produção *(MPP)*. Com estes métodos se procura melhorar o dimensionamento de lotes adiantando e postergando a produção entre períodos, respeitando as capacidades de produção destes. A seguir se descrevem esses métodos.

O $MAP$ adianta a produção de um produto $j$ de um período $t$ para o período imediatamente anterior $t-1$ se este tem capacidade disponível e o produto $j$ está sendo produzido. Com este método se procura adiantar toda a produção (lote inteiro) do produto $j$, assim os custos de processamento e de preparação no período $t$ que incorrem neste produto, serão consideravelmente reduzidos. Dado que $j$ já está sendo produzido no período $t-1$, não será necessário incorrer em mais custos de preparação neste período. O produto $j$ e o período $t$ são aleatoriamente selecionados. Note que neste método a divisão de um lote de produção não é considerada *(lot-splitting)*.

O $MAP$ é descrito no Algoritmo 9 e recebe como parâmetros uma solução $SOL$ e o número máximo de iterações $Iter_{DL}$. Estes parâmetros são inicializados nas linhas 1 e 2 respectivamente. Na linha 3 observa-se o número máximo de soluções a serem visitadas ($Iter_{DL}$). Primeiramente é selecionado de maneira aleatória um período $t-1$ (com capacidade de produção disponível) o qual deve ser menor que $T$, na linha 4 é realizada essa eleição. Na linha 5 é selecionado um produto aleatório $j$ do período $t$, este deve ser imediatamente posterior a $t-1$. Na linha 6 testa-se se existe capacidade suficiente no período $t-1$ para produzir todo o lote do produto $j$ que está sendo produzido no período $t$. Se existe capacidade disponível, o lote do produto $j$ no período $t$ ($DL_{jt}$) será todo produzido no período anterior $t-1$. Assim, na linha 7, é adicionado ao dimensionamento do lote do produto $j$ no período $t-1$ ($DL_{j(t-1)}$) a dimensão do lote do mesmo produto $j$ mais o correspondente ao período $t$ ($DL_{jt}$), portanto, a dimensão do lote do produto $j$ no período $t$ será igual a zero (linha 8) e o estoque do produto $j$ no período $t-1$ deverá ser ajustado conforme o lote produzido e a demanda deste

período (linha 9). Desde a linha 12 até a linha 16 o critério de aceitação é aplicado. Na linha 19, finalmente, uma solução é retornada.

**Algorithm 9** : $MAP(SOL, Iter_{DL})$;

1: $SOL^* := SOL$;
2: $iter := 1$;
3: **while** $iter < Iter_{DL}$ **do**
4: $t-1$ := período selecionado aleatoriamente com capacidade disponível menor que $T$.
5: $j$ := produto selecionado aleatoriamente do período $t$.
6: **if** Capacidade_Suficiente$(t-1)$ **then**
7: $DL_{j(t-1)} := DL_{j(t-1)} + DL_{jt}$;
8: $DL_{jt} := 0$;
9: $Est_{j(t-1)} := DL_{j(t-1)} - d_{j(t-1)}$;
10: $//SOL$ é atulizada nos três passos anteriores.
11: **end if**
12: **if** $f(SOL) < f(SOL^*)$ **then**
13: $SOL^* := SOL$;
14: **else**
15: $SOL := SOL^*$;
16: **end if**
17: $iter := iter + 1$;
18: **end while**
19: **return** SOL;

Baseado no exemplo numérico da Subseção 3.2, na Figura 5.5 mostra-se uma solução que contém três períodos ($t_1, t_2$ e $t_3$) e três produtos (1, 2 e 3). A solução foi construída pelo procedimento de construção da solução inicial, na inicialização da sequência de produção e do dimensionamento de lotes foram usada a heurística NEH e a política lote por lote, respectivamente.

Para ilustrar um movimento do $MAP$, na Figura 5.5, o produto 3 e o período $t_3$ foram ambos selecionados de maneira aleatória. Posteriormente, no período $t_2$ se verifica se existe capacidade disponível para produzir todo o lote do produto 3. Dado que a capacidade existe, todo este lote é adiantado para o período anterior $t_2$. Consequentemente, é gerado um estoque do produto 3 no período $t_2$ e o produto 3 no período $t_3$ deve ser excluído, já que as quantidades equivalentes à demanda deste produto estão

sendo produzidas no período $t_2$.

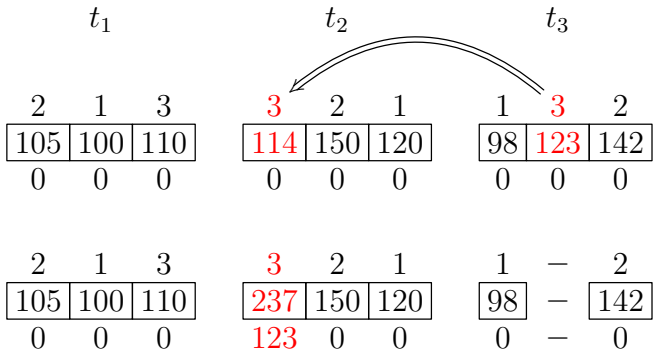



Figura 5.5: Método para adiantar a produção

No $MPP$, a produção excedente de um produto $j$ de um período $t$ é postergada ao período imediatamente posterior $t+1$, se o produto $j$ estiver sendo produzido neste período. A produção excedente a ser postergada depende da capacidade disponível no período $t+1$. Desta forma, se for possível, todo o estoque do produto $j$ será postergado. Se não for possível, será postergado a parte do estoque que seja permitido pela capacidade disponível no período $t+1$. Um dos principais propósito deste método é diminuir o custo do estoque; portanto, o produto $j$ selecionado aleatoriamente do período $t$, deverá ter produção excedente.

O $MPP$ é descrito no Algoritmo 10 e recebe como parâmetros uma solução $SOL$ e o número máximo de iterações $Iter_{DL}$. Desde a linha 3 até a linha 17 encontra-se o loop principal do algoritmo. Na linha 4 se seleciona um período ($t$) de maneira aleatória dentre todos os períodos como produção excedente. Na linha 5 é selecionado aleatoriamente um produto $j$ com estoque positivo que esteja sendo produzido no período $t$. Na linha 6 testa-se se no período $t+1$ existe capacidade disponível para produzir todo ou parte do estoque do produto $j$ do período $t$ ($Est_{jt}$). Se existe a capacidade disponível, na linha 7 se posterga o estoque permitido (segundo a capacidade) do produto $j$ do período $t$ ($Est\_P_{jt}$) para o período $t+1$. Em outras palavras, o ($Est\_P_{jt}$) será adicionado à dimensão do lote do produto $j$ no período $t+1$ ($DL_{j(t+1)}$). Dada uma postergação, o estoque do produto $j$ no período $t$ deverá ser atualizado, sendo o estoque inicial ($Est_{jt}$) menos o estoque postergado ($Est\_P_{jt}$), na linha 8 observa-se esta operação. A atualização do estoque também impacta na dimensão do lote, assim, a dimensão do lote para o produto $j$ no período $t$ ($DL_{jt}$) será igual à demanda ($d_{jt}$) do mesmo produto $j$ do mesmo período $t$ mais o estoque do produto $j$ que não foi postergado (linha 9). Da linha 12 a linha 16 é estabelecido o critério de aceitação

para avaliar as novas soluções geradas. Finalmente, na linha 18, é retornada a melhor solução encontrada pelo $MPP$.

**Algorithm 10** : $MPP(SOL, Iter_{DL})$;

1: $SOL^* := SOL$;
2: $Iter := 1$;
3: **while** $Iter < Iter_{DL}$ **do**
4: $t :=$ período selecionado aleatoriamente com produção excedente de pelo menos um produto.
5: $j :=$ produto com produção excedente selecionado aleatoriamente do período $t$.
6: **if** Capacidade_Disponível$(t+1)$ **then**
7: $DL_{j(t+1)} := DL_{j(t+1)} + Est\_P_{jt}$;
8: $Est_{jt} := Est_{jt} - Est\_P_{jt}$;
9: $DL_{jt} := d_{jt} + Est_{jt}$;
10: $//SOL$ é atulizada nos três passos anteriores.
11: **end if**
12: **if** $f(SOL) < f(SOL^*)$ **then**
13: $SOL^* := SOL$;
14: **else**
15: $SOL := SOL^*$;
16: **end if**
17: **end while**
18: **return** SOL;

Usando o mesmo problema do exemplo numérico da Subseção 3.2, na Figura 5.6 ilustra um movimento do $MPP$. Primeiramente é selecionado aleatoriamente um período com produção excedente de algum produto. No exemplo é selecionado o período $t_2$, este tem os produtos 1 e 3 com estoques. Entre estes produtos, o 1 é aleatoriamente selecionado. Seguidamente, no período $t_3$ verifica-se a existência da capacidade disponível para produzir todo o estoque do produto 1 do período $t_2$. Dado que a capacidade é suficiente, todo o estoque do produto 1 no período $t_2$ é postergado para o período $t_3$. Note que o estoque do produto 1 no período $t_2$ foi reduzido a zero, isto leva a um aumento na dimensão do lote deste mesmo produto no período $t_3$.

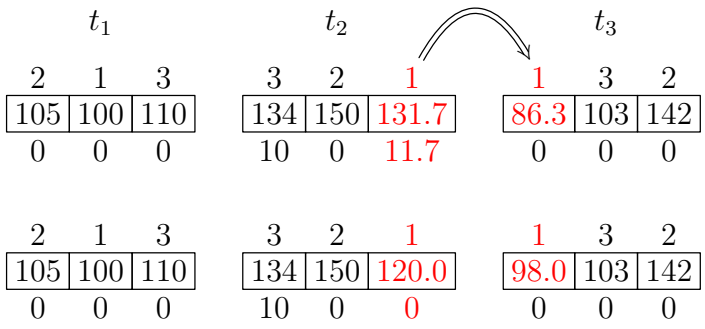



Figura 5.6: Método para postergar a produção

## 5.4 Heurística Iterated Greedy + Fix and Optimize (IG+F&O)

Além do algoritmo IG apresentado anteriormente, é também proposto um híbrido entre o IG e a heurística Fix & Optimize, resultando no algoritmo híbrido $IG + F\&O$.

A heurística Fix & Optimize, anteriormente denominada "exchange", proposta por Pochet & Wolsey [2006], foi uma versão melhorada da heurística Relax and Fix, também proposta neste mesmo trabalho. No trabalho de Sahling et al. [2009] foi introduzida uma nova estratégia, combinada com a heurística exchange, resultando na heurística Fix and Optimize ($F\&O$).

F&O é baseada na formulação matemática do problema. F&O consiste em dividir algumas variáveis de decisão do problema em dois conjuntos: conjunto de variáveis a serem fixadas e conjunto de variáveis a serem otimizadas. Este último conjunto recebe o nome de subproblema. A ideia da heurística é resolver o subproblema de maneira ótima, já que o número de variáveis a serem otimizadas é menor, dada a fixação de um número considerado de variáveis de decisão da formulação matemática original. O processo da heurística é iterativo e, em cada iteração, novos conjuntos de variáveis para fixar e otimizar são considerados. Note que os subproblemas também são problemas inteiros mistos, mas com um número menor de variáveis inteiras a otimizar e, portanto, mais fáceis de resolver.

O $IG+F\&O$ é apresentado no Algoritmo 11. Os procedimentos de destruição - construção e busca local não sofrem alterações, enquanto à construção da solução inicial é adicionado mais um procedimento. Em relação às decisões do dimensionamento de lotes, estes são definidos nos procedimentos exatos da heurística $F\&O$.

**Algorithm 11** : IG+F&O($d$, $It\_Bl$, $It\_IG$)

```
1: SOL := Construção_Solução_Inicial();
2: if Solução_Infactível(SOL); then
3:    Formulação_Matemática(SOL);
4: end if
5: SOL* := SOL;
6: iterações:=0;
7: while iterações ≤ It_IG do
8:    SOL := Destruição-Construção(SOL, d);
9:    SOL := Busca_Local(SOL, It_Bl);
10:   SOL := Fix_Optimize(SOL);
11:   if f(SOL) < SOL* then
12:      SOL* := SOL;
13:   else
14:      SOL := SOL*;
15:   end if
16:   iterações := iterações +1;
17: end while
18: return SOL*;
```

### 5.4.1 Construção da solução inicial

A obtenção da solução inicial é realizada de forma similar ao procedimento *IDL* do Algoritmo 7 (*IG+MDL*) e continua sendo de duas fases. O sequenciamento é determinado da mesma maneira, enquanto para o dimensionamento de lotes, é adicionado mais um procedimento, o qual será descrito a seguir.

Após o sequenciamento dos produtos ter sido determinado, o dimensionamento de lotes é realizado de forma regressiva, período a período, iniciando pelo último e terminando no primeiro período. Este procedimento é similar ao ($IDL$) do Algoritmos 7 anteriormente descrito. Em alguns casos, este procedimento não gera uma solução viável. Quando isto acontece, é iniciado um novo procedimento, dimensionamento progressivo, o qual é contrário ao dimensionamento regressivo. O procedimento progressivo inicia pelo primeiro período, produzindo as quantidades necessárias para cumprir a demanda, se este período ainda tem capacidade disponível, esta é usada para produzir quantidades do produto que mais consome tempo de produção no período seguinte. Este processo é repetido até chegar ao último período. Geralmente, o dimensionamento de lotes progressivo gera um estoque considerável. Para tentar minimizar estes, é aplicado o método para postergar a produção ($MPP$), com algumas modificações. Nesta vez, o produto selecionando será aquele que possua mais estoque para postergar parte da sua produção para o período seguinte. O MPP modificado

($MPP\_M$) é aplicado em todos os períodos da seguinte forma: $t_1 \rightarrow t_2$, $t_2 \rightarrow t_3$,..., $t_{T-1} \rightarrow t_T$. O Algoritmo 12 descreve a inicialização do novo método de dimensionamento de lotes, o qual tem sido chamado $IDL2$. É possível que após a execução deste método ainda resulte uma solução inviável. Neste caso, e como último recurso, se faz necessária a execução da formulação matemática como é feito no $IG + MDL$ e nos agentes construtores do $A - Teams$.

**Algorithm 12** : $IDL2(SOL)$;

```
1: IDL(SOL);
2: if Solucao_Infactivel(SOL) then
3:   for t = 1 ≤ T do
4:     for j = 1 ≤ N do
5:       Dimensao_Lote_jt = Demanda_jt;
6:       if Capacidade_Insuficiente() then
7:         Sair deste procedimento e executar a formulação matemática;
8:       end if
9:     end for
10:    while Capacidade_Disponivel(t) do
11:      Selecione o produto i que requer mais tempo de processamento no período
         t + 1.
12:      Produzir o permitido do produto i no período t.
13:    end while
14:  end for
15:  MPP_M();
16: end if
```

### 5.4.2 Melhoria do dimensionamento de lotes

Após terminado o procedimento de busca local, um procedimento heurístico é iniciado com o objetivo de melhorar o dimensionamento de lotes fazendo uso da heurística $F\&O$

Dado que os procedimentos de destruição - construção e busca local garantem a preservação da preparação das máquinas entre os períodos, é usada a formulação matemática reduzida (equações 4.1 - 4.12).

Em cada iteração da heurística, um conjunto de $n$ produtos são selecionados de maneira aleatória. As variáveis correspondentes ao dimensionamento de lote, relacionadas a esse conjunto de produtos são as que compõem o denominado subproblema, o qual será otimizado. As variáveis restantes, correspondentes ao dimensionamento de

lote, as quais não estão no subproblema, e as variáveis correspondentes ao sequenciamento dos produtos, são fixadas conforme os valores da solução atual.

Neste procedimento, as variáveis de decisão consideradas são $SO_{jmt}$, $CO_{jmt}$, $Z_{jt}$, $x_{jt}$ e $I_{jt}$. O resto das variáveis consideradas pelo modelo matemático e a variável $Y_{ijt}$ são fixadas com valores encontrados em iterações anteriores. Note que as variáveis relacionadas ao conjunto de produtos selecionados (subproblema) são otimizadas sobre todo o horizonte de planejamento.

A Figura 5.7 mostra um exemplo do funcionamento da heurística F&O. O horizonte de planejamento consta de três períodos ($t_1$, $t_2$ e $t_3$) e três produtos ($j_1$, $j_2$ e $j_3$). As notações $Seq$, $DL$ e $Est$, representam o ordem de produção, a dimensão do lote ($x$) e o estoque ($I$) de cada produto ($j$) em cada período ($t$). As variáveis $SO_{jmt}$ e $CO_{jmt}$ determinam o tempo de início e finalização de um lote do produto $j$ em uma máquina $m$ no período $t$.

No exemplo, foi selecionado aleatoriamente um produto ($j_1$), o qual será otimizado ao longo do horizonte de planejamento. As variáveis relacionadas à dimensão de lote ($x_1$) e ao estoque ($I_1$) do produto $j_1$ serão otimizadas (subproblema: variáveis em verde), enquanto as variáveis correspondente aos produtos não selecionados ($x_2$, $x_3$, $I_2$ e $I_3$) são fixadas (variáveis em vermelho). Com respeito às variáveis que determinam o tempo de início e finalização de um lote de produção ($SO$ e $CO$), são apenas fixas as variáveis correspondes aos produtos que são processados antes que o produto selecionado ($j_1$). Note que no período $t_1$ o lote do produto $j_2$ é processado antes do produto $j_1$, portanto, as variáveis $SO_{J_2m1}$ e $CO_{J_2m1}$ são inalteradas. Dado que a produção do lote do produto $j_1$ pode ser alterado, a variável $CO_{J_1m1}$ poderá ser modificada, o que tem influencia direta nas varáveis $SO_{J_3m1}$ e $CO_{J_3m1}$ do lote do produto $j_3$, já que a produção do lote deste produto vai a iniciar depois da produção do lote do produto $j_1$ ter terminado mais o tempo de preparação necessário. No período $t_2$, o produto $j_1$ está situado na última posição da sequência de produção, assim, as variáveis correspondentes aos produtos diferentes de $j_1$ são fixadas ($SO_{J_2m2}$, $CO_{J_2m2}$, $SO_{J_3m2}$ e $CO_{J_3m2}$). Neste período só o lote de produção do produto $j_1$ pode ser alterado, portanto só as variáveis $SO_{J_1m2}$ e $CO_{J_1m2}$ serão modificadas. No último período, o produto a ser otimizado está situado na primeiro posição na sequência de produção. Se o lote deste produto é alterado, necessariamente as variáveis $SO_{jmt}$ e $CO_{jmt}$ de todos os produtos serão também alteradas, pois o tempo de início para processar um lote, depende do tempo de finalização do lote anterior, mais o tempo de preparação das máquinas.

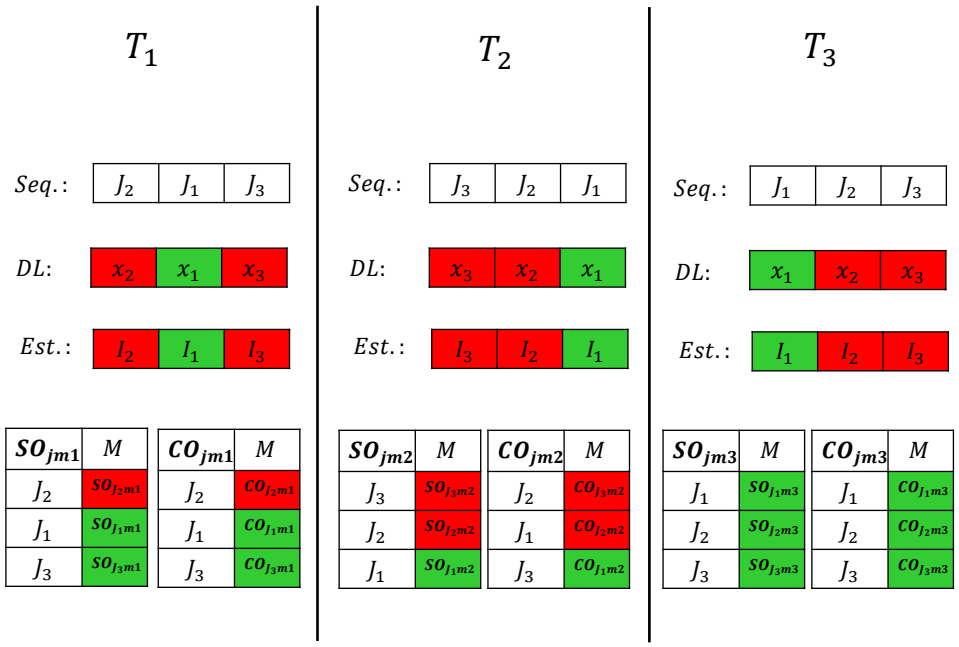



Figura 5.7: Fix and Optimize

# Capítulo 6

# Experimentos computacionais

Neste capítulo apresentam-se os testes computacionais realizados para analisar o desempenho dos algoritmos propostos *(IG+HR, IG+MDL, IG+F&O)* em relação á qualidade da solução e tempo computacional em comparação com os algoritmos da literatura *(HR A-Teams)*, esses foram reimplementadas conforme os artigos originais.

## 6.1 Ambiente de teste

Os testes foram realizados usando um computador com processador *Intel(R) Xeon(R) CPU X5650 2,67GHz com 48GB de RAM*. Todos os algoritmos foram implementados em C++. Para a resolução dos problemas de programação linear e programação linear inteira mista foi utilizado o software IBM ILOG CPLEX 12.4 por meio da biblioteca CPLEX Concert Tecnologic.

## 6.2 Geração de problemas-teste

Os problemas-teste (ou instâncias) foram gerados conforme os experimentos realizados por Mohammadi et al. [2010b]. As instâncias de problemas consideradas nesses experimentos são classificadas de acordo com o número de produtos, máquinas e períodos *(N, M, T)*. As 20 classes de instâncias geradas por esses autores são mostradas na Tabela 6.1. Note que os tamanhos de instâncias variam de $(3, 3, 3)$ a $(15, 15, 15)$.

Tabela 6.1: Classes de instâncias da literatura

| **(N,M,T)** | **(N,M,T)** | **(N,M,T)** | **(N,M,T)** |
|---|---|---|---|
| (3,3,3) | (7,5,5) | (5,10,5) | (10,10,10) |
| (5,3,3) | (5,7,5) | (7,7,10) | (10,10,15) |
| (3,5,3) | (5,5,7) | (7,10,7) | (10,15,10) |
| (3,3,5) | (7,7,7) | (10,5,5) | (15,10,10) |
| (5,5,5) | (5,5,10) | (10,7,7) | (15,15,15) |

Neste trabalho, foram geradas instâncias cujas dimensões variam de $(3, 3, 3)$ a $(15, 15, 15)$ como em Mohammadi et al. [2010b], e instâncias maiores no qual as dimensões $(N, M, T)$ variam entre $(20, 10, 10)$ a $(60, 20, 20)$, veja Tabela 6.2. Desta forma, um total de 101 classes de instâncias foram geradas. Para cada classe de instâncias foram gerados 5 exemplares; assim, no total temos 505 problemas para testar todos os algoritmos.

Tabela 6.2: Novas classes de instâncias geradas

| **(N,M,T)** | **(N,M,T)** | **(N,M,T)** | **(N,M,T)** | **(N,M,T)** | **(N,M,T)** |
|---|---|---|---|---|---|
| (20,10,10) | (25,15,20) | (35,10,15) | (40,20,10) | (50,10,20) | (55,20,15) |
| (20,10,15) | (25,20,10) | (35,10,20) | (40,20,15) | (50,15,10) | (55,20,20) |
| (20,10,20) | (25,20,15) | (35,15,10) | (40,20,20) | (50,15,15) | (60,10,10) |
| (20,15,10) | (25,20,20) | (35,15,15) | (45,10,10) | (50,15,20) | (60,10,15) |
| (20,15,15) | (30,10,10) | (35,15,20) | (45,10,15) | (50,20,10) | (60,10,20) |
| (20,15,20) | (30,10,15) | (35,20,10) | (45,10,20) | (50,20,15) | (60,15,10) |
| (20,20,10) | (30,10,20) | (35,20,15) | (45,15,10) | (50,20,20) | (60,15,15) |
| (20,20,15) | (30,15,10) | (35,20,20) | (45,15,15) | (55,10,10) | (60,15,20) |
| (20,20,20) | (30,15,15) | (40,10,10) | (45,15,20) | (55,10,15) | (60,20,10) |
| (25,10,10) | (30,15,20) | (40,10,15) | (45,20,10) | (55,10,20) | (60,20,15) |
| (25,10,15) | (30,20,10) | (40,10,20) | (45,20,15) | (55,15,10) | (60,20,20) |
| (25,10,20) | (30,20,15) | (40,15,10) | (45,20,20) | (55,15,15) | – |
| (25,15,10) | (30,20,20) | (40,15,15) | (50,10,10) | (55,15,20) | – |
| (25,15,15) | (35,10,10) | (40,15,20) | (50,10,15) | (55,20,10) | – |

Os parâmetros para cada classe de instâncias são gerados por meio de uma distribuição de probabilidade uniforme conforme aos valores e intervalos estabelecidos por Mohammadi et al. [2010b]. A seguir mostram-se os valores e intervalos de cada um dos parâmetros utilizados:

- Demanda: $d_{jt} = U(0; 180)$
- Custo de estoque: $h_{jm} = U(0,2; 0,4)$
- Tempos de processamentos: $b_{jm} = (1,5; 2)$
- Custos de processamentos: $p_{jmt} = U(1,5; 2)$
- Tempos de preparação: $S_{ijm} = (35; 70)$
- Custos de preparação: $W_{ijm} = (35; 70)$
- Capacidade: $C_{mt} = (x, y)$ onde, $x = 200N + 100(M-1)$ e $y = 200N + 200(M-1)$

## 6.3 Métrica de avaliação dos algoritmos

Para avaliar o desempenho de todos os algoritmos em relação à qualidade da solução e tempo computacional, foi necessário padronizar o conjunto de dados obtidos por cada algoritmo, já que se apresentam muitas instâncias de diferentes classes, contendo, um conjunto de valores sem relação alguma.

Neste trabalho os dados obtidos por cada algoritmo são padronizados utilizando a média do desvio relativo porcentual (Relative Percentual Deviation, RPD). Esta medida é comumente utilizada na literatura (Vallada & Ruiz [2011]).

O *RPD* é calculado pela seguinte equação:

$$RPD\% = 100 \times \frac{Metodo_{sol} - Melhor_{sol}}{Melhor_{sol}} \tag{6.1}$$

Onde $Metodo_{sol}$ é o valor da função objetivo de uma solução obtida por um dos algoritmos e $Melhor_{sol}$ é o valor da função objetivo da melhor solução encontrada entre todos os algoritmos comparados. Desta forma, o *RPD* define a qualidade de uma solução em relação à melhor solução obtida. Quanto menor é o valor do *RPD*, melhor será a qualidade da solução do algoritmo. O *RPD* de um algoritmo é calculado utilizando a média do valor da função objetivo de 10 execuções desse algoritmo em cada uma das instâncias.

## 6.4 Calibração de parâmetros dos algoritmos

Para a determinação da configuração ótima dos parâmetros dos algoritmos propostos *(IG+HR, IG+MDL, IG+F&O)* e da literatura (TAH), foi necessário realizar um

desenho de experimento *(design of experiments, DOE)*. Nesta dissertação temos optado pelo desenho experimental fatorial multi-nível Montgomery [2001].

Os desenhos fatoriais multi-nível são usados para estudar efeitos de $q$ fatores (parâmetros) quantitativos. Neste tipo de desenho experimental deve-se especificar a faixa de cobertura e o número de diferentes níveis para cada fator nos quais se realiza o experimento. Logo, constrói-se o banco de dados que contêm todas as configurações dos diferentes níveis de todos os fatores, chamadas de condições experimentais *(CEX)*.

Para testar as condições experimentais é necessário definir um conjunto de classes de instâncias a utilizar no experimento. Foram utilizadas as instâncias de médio porte, as instâncias de tamanho entre $(5, 5, 10)$ e $(15, 15, 15)$. Assim, tem-se 11 classes, de instâncias. Para cada classe 10 exemplares foram gerados. Desta forma tem-se um conjunto de 110 instâncias. Nos experimentos de calibração de parâmetros é possível utilizar todas as classes de instâncias só que o esforço computacional seria demasiado alto, por isso, temos selecionado uma "amostra" e esta contém características de todas as classes de instâncias.

A seguir descrevem-se os detalhes da calibração dos fatores (ou parâmetros) de cada um dos algoritmos.

### 6.4.1 Calibração do times assíncronos híbridos (TAH)

O *TAH* tem quatro parâmetros: Tamanho da memória compartilhada ($T\_Mc$), iterações dos agentes ($Iter\_Ag$), iterações sem melhora ($Iter\_Sm$) e número de iterações totais do algoritmo ($Iter\_Alg$). A Tabela 6.3 apresenta o desenho experimental fatorial multi-nível, esta tabela contém cada um dos fatores (parâmetros) com sua respectiva faixa de cobertura e o número de níveis estabelecidos. Os valores mínimos e máximos da faixa de cobertura (colunas 2 e 3) e o número de nível de cada fator (coluna 3) foram estabelecidos com base na literatura e a experimentos prévios dos algoritmos.

Tabela 6.3: Desenho experimental fatorial multi-nível para o *TAH*

| **Fatores** | **Mínimo** | **Máximo** | **Níveis** |
|---|---|---|---|
| $T\_Mc$ | 5 | 20 | 3 |
| $Iter\_Ag$ | 4 | 9 | 3 |
| $Iter\_Sm$ | 20 | 60 | 3 |
| $Iter\_Alg$ | 50000 | 100000 | 3 |

O desenho experimental, o qual foi realizado no software *Statgraphics Centurion XV* encontrou 81 condições experimentais (combinações de parâmetros), estas são apresentadas na Tabela B.1 (ver apêndice B).

Após testar todas as condições experimentais em cada uma das instâncias foram calculados os valores do *RPD* conforme a equação 6.1.

Geralmente os desenhos experimentais estão baseados na análise de variância *ANOVA*(Montgomery [2001]), porém em alguns casos não é possível se basear nesta análise, dado que os resíduos da variável resposta do estudo experimental, em nosso caso o *RPD*, não cumpre com as suposições da *ANOVA* (normalidade, independência e homogeneidade de variância). Nosso caso, a variável resposta (resíduos do *RPD*) não cumpre com as suposições. Dado isto, o desenhos experimental foi baseado no teste de Kruskal-Wallis (Montgomery [2001]).

O teste de Kruskal-Wallis é uma extensão do teste de Wilcoxon-Mann-Whitney (Montgomery [2001]) e é análogo ao teste $F$ utilizado na *ANOVA*. Enquanto a *ANOVA* está sujeita ao cumprimento obrigatório das suposições, o teste de Kruskal-Wallis não está sujeita a nenhuma suposição para que a comparação possa ser realizada com sucesso.

O teste Kruskal-Wallis é usado para testar a hipótese nula ($H_0$) de que "todas as medianas ($Me$) das amostras $(1, .., n)$ são iguais" contra a hipótese alternativa ($H_1$) de que "ao menos uma das medianas é diferente". Nosso caso, uma amostra é o conjunto de dados já padronizados obtidos por uma condição experimental.

As hipóteses ($H_0$) e ($H_1$) são escritas formalmente da seguinte forma:

$H_0 : Me_1 = Me_2 = ... = Me_n$.
$H_1 : Me_i \neq Me_j$ para pelo menos um par $i, j$.
Onde
$H_0$ : Hipótese nula.
$H_1$ : Hipótese alternativa.
$Me_1$ : Mediana dos resultados obtidos pela amostra 1.
$Me_2$ : Mediana dos resultados obtidos pela amostra 2.
$Me_n$ : Mediana dos resultados obtidos pela amostra $n$.
$Me_i$ e $Me_j$ são duas medianas quaisquer (diferentes) das $n$ amostras.

Nos testes estatísticos como o Kruskal-Wallis, a informação mais importante é o

*valor-P*. Este valor encontra-se associado com as hipóteses estabelecidas. Se o *valor-P* é menor ou igual $0,05$ a hipóteses nula estabelecida para o teste é rejeitada. Isto indica uma diferença estatística significativa entre as amostras. No caso de o *valor-P* ser maior que $0,05$ a hipótese nula é aceita, isto afirma que não existe uma diferença estatística significativa entre as amostras. Dado que o *valor-P* está diretamente relacionado com as provas de hipóteses, só será necessário a explicação do *valor-P* ou das hipóteses estabelecidas.

O teste Kruskal-Wallis aplicado às condições experimentais do Algoritmo *TAH* é mostrado na Tabela 6.4. Nesta tabela mostra-se as 40 configurações que obtiveram os melhores resultados com relação à variável *RPD*, as 41 condições experimentais restantes não foram consideradas dado que obtiveram valores *(RPD)* excessivamente altos. O *Valor-P=0,0* do teste garante uma diferença estatística significativa entre as medianas das amostras, rejeitando $H_0$ e segundo a coluna do ranking médio a melhor condição experimental é a $CEX\_67$ com ranking de 66.

Tabela 6.4: Teste de Kruskall-Wallis para as CEX do *TAH*

| **Condição Exp.** | **Tamanho da amostra** | **Ranking médio** |
|---|---|---|
| $CEX\_1$ | 110 | 77,55 |
| $CEX\_3$ | 110 | 284,6 |
| $CEX\_5$ | 110 | 379,6 |
| $CEX\_6$ | 110 | 383,5 |
| $CEX\_7$ | 110 | 125,5 |
| $CEX\_8$ | 110 | 147,6 |
| $CEX\_10$ | 110 | 227,7 |
| $CEX\_11$ | 110 | 104,5 |
| $CEX\_13$ | 110 | 211,1 |
| $CEX\_14$ | 110 | 72 |
| $CEX\_17$ | 110 | 320,4 |
| $CEX\_18$ | 110 | 344,8 |
| $CEX\_19$ | 110 | 292,9 |
| $CEX\_23$ | 110 | 300,5 |
| $CEX\_24$ | 110 | 379,1 |
| $CEX\_25$ | 110 | 226,9 |
| $CEX\_26$ | 110 | 203,1 |
| $CEX\_27$ | 110 | 381,6 |

*Continua na próxima página*

Tabela 6.4 – *Continuação da página anterior*

| **Condição Exp.** | **Tamanho da amostra** | **Ranking médio** |
|---|---|---|
| $CEX\_29$ | 110 | 77,27 |
| $CEX\_32$ | 110 | 82,36 |
| $CEX\_39$ | 110 | 140,1 |
| $CEX\_43$ | 110 | 138,4 |
| $CEX\_44$ | 110 | 296,3 |
| $CEX\_49$ | 110 | 229,1 |
| $CEX\_51$ | 110 | 238,3 |
| $CEX\_52$ | 110 | 321,5 |
| $CEX\_55$ | 110 | 344,9 |
| $CEX\_57$ | 110 | 293 |
| $CEX\_60$ | 110 | 301 |
| $CEX\_65$ | 110 | 379,5 |
| $CEX\_66$ | 110 | 73 |
| $CEX\_67$ | 110 | **66** |
| $CEX\_68$ | 110 | 194,7 |
| $CEX\_69$ | 110 | 217,2 |
| $CEX\_70$ | 110 | 249,5 |
| $CEX\_72$ | 110 | 117 |
| $CEX\_77$ | 110 | 137,1 |
| $CEX\_78$ | 110 | 176,5 |
| $CEX\_80$ | 110 | 78,55 |
| $CEX\_81$ | 110 | 205,8 |
| Valor-P = 0,0 | | |

A fim de ter uma melhor apreciação das 40 melhores condições experimentais, o gráfico de intervalos de confiança S agrupada é mostrado na Figura 6.1. Neste gráfico pode-se observar a diferença estatística significativa que existe entre algumas condições experimentais concordando com o *valor-P* do teste Kruskal-Wallis. No gráfico também se aprecia que são 8 as condições experimentais que apresentam os menores valores da variável em estudo. Aplicando conceitos estatísticos seria indiferente selecionar qualquer uma desta 8 condições, porém para o Algoritmo *TAH* selecionamos a que apresenta o menor ranking médio (66) sendo a condição experimental $CEX\_67$: $T\_Mc = 20$, $Iter\_Ag = 9$, $Iter\_Sm = 40$, $Iter\_Alg = 50000$.

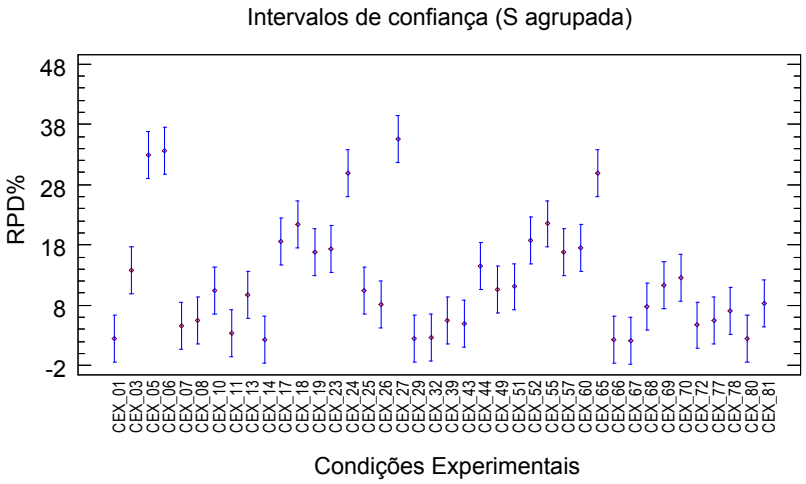



Figura 6.1: Intervalos de confiança S agrupada para às condições experimentais do *TAH*

### 6.4.2 Calibração dos parâmetros do algoritmo IG+HR

Na calibração deste algoritmo híbrido só é necessário calibrar o *IG*, pois a *HR* não tem parâmetros. Porém na execução dos problemas a heurística *HR* é considerada, decisões desta podem afetar diretamente no *IG*.

O IG tem 3 parâmetros a serem calibrados: parâmetro de destruição ($d$), números iterações da busca local ($Iter\_Bl$) e número de iterações totais do algoritmo ($Iter\_Alg$). Foi realizado um desenho experimental fatorial multi-nível e os dados de entrada requeridos por este, são mostrados na Tabela 6.5. Na primeira coluna encontra-se o nome de cada fator, a segunda e terceira coluna, representam o valor mínimo e máximo da faixa de cobertura de cada fator e a última coluna mostra o número de níveis estabelecidos por fator.

Tabela 6.5: Desenho experimental fatorial multi-nível para o *IG+HR*

| **Fatores** | **Mínimo** | **Máximo** | **Níveis** |
|---|---|---|---|
| $d$ | 0,3N | 0,7N | 3 |
| $Iter\_Bl$ | 4 | 9 | 3 |
| $Iter\_Alg$ | 20 | 60 | 3 |

O desenho experimental fatorial multi-nível achou 27 condições experimentais, estas são mostradas na Tabela B.2 (ver Apêndice B)

Cada condição experimental do *IG+HR* foi executada para cada classe de instância, os valores da função objetivo obtidos foram padronizados conforme o *RPD*, equação 6.1.

Esta calibração também é baseada no teste de Kruskal-Wallis, cujo resultado é mostrado na Tabela 6.6. O teste mostra um *Valor-P* igual à zero indicando uma diferença estatística significativa entre as medianas das amostras, sendo as amostras os dados obtidos por cada condição experimental. Segundo o ranking médio (coluna três da Tabela 6.6), três condições experimentais apresentam menores valores que as restantes, estas condições são $0,5N-6-80$, $0,5N-6-20$ e $0,3N-4-20$ com rankings médios de 47.5, 47.5 e 59 respectivamente.

Tabela 6.6: Teste de Kruskall-Wallis para as CEX do *IG+HR*

| **Condição Exp.** | **Tamanho da amostra** | **Ranking médio** |
|---|---|---|
| $0,3N-8-50$ | 110 | 79,3 |
| $0,7N-8-80$ | 110 | 228,5 |
| $0,3N-6-20$ | 110 | 92,5 |
| $0,3N-4-80$ | 110 | 86,15 |
| $0,3N-4-20$ | 110 | 59 |
| $0,5N-8-20$ | 110 | 221 |
| $0,3N-8-20$ | 110 | 202,3 |
| $0,7N-4-50$ | 110 | 90,4 |
| $0,5N-4-80$ | 110 | 90,4 |
| $0,3N-4-50$ | 110 | 160,1 |
| $0,7N-6-20$ | 110 | 236,9 |
| $0,7N-4-20$ | 110 | 78,5 |
| $0,7N-8-50$ | 110 | 109,4 |
| $0,7N-6-80$ | 110 | 109,4 |
| $0,5N-4-20$ | 110 | 207,7 |
| $0,7N-4-80$ | 110 | 124,3 |
| $0,5N-4-50$ | 110 | 77,7 |
| $0,5N-6-20$ | 110 | **47,5** |
| $0,5N-6-80$ | 110 | **47,5** |
| $0,5N-6-50$ | 110 | 135 |
| $0,3N-8-80$ | 110 | 226,9 |
| $0,7N-8-20$ | 110 | 99,5 |

*Continua na próxima página*

Tabela 6.6 – *Continuação da página anterior*

| **Condição Exp.** | **Tamanho da amostra** | **Ranking médio** |
|---|---|---|
| $0,3N-6-50$ | 110 | 231,5 |
| $0,3N-6-80$ | 110 | 137 |
| $0,7N-6-50$ | 110 | 137 |
| $0,5N-8-50$ | 110 | 132,1 |
| $0,5N-8-80$ | 110 | 211 |
| Valor-P = 0,0 | | |

Para apreciar visualmente todos os dados obtidos pelas condições experimentais, na Figura 6.2 mostra-se o gráfico de intervalos de confiança S agrupada por condição experimental, este gráfico mostra que definitivamente as três condições mencionadas acima são as que obtiveram os menores *RPDs*. Para o algoritmo *IG+HR* selecionamos a condição experimental que apresenta o menor ranking médio, sendo a $d = 0,5N$, $Iter\_Bl = 6$, $Iter\_Alg = 80$.

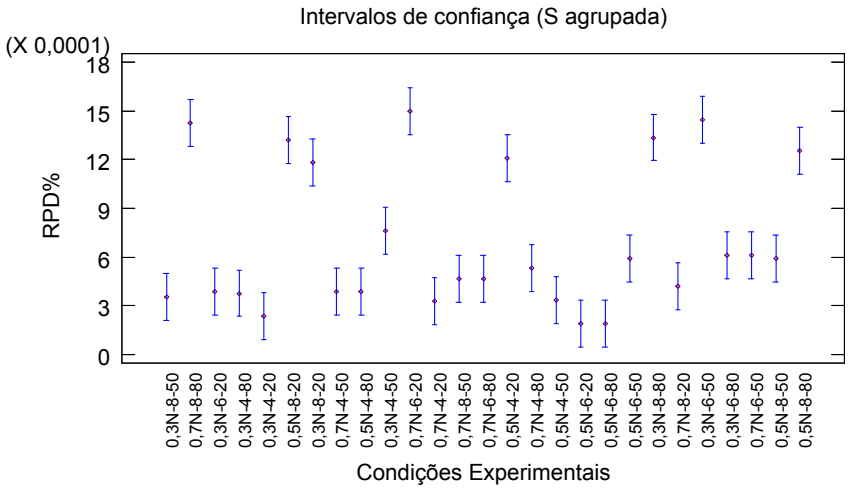



Figura 6.2: Intervalos de confiança S agrupada para às condições experimentais do *IG+HR*

### 6.4.3 Calibração dos parâmetros da heurística IG+MDL

O *IG+MDL* tem 4 parâmetros: os básicos do *IG*, parâmetro de destruição ($d$), número iterações da busca local ($Iter\_Bl$) e número de iterações totais do algoritmo ($Iter\_Alg$), além destes, é necessário a calibração do parâmetros das iterações do *MDL* ($Iter\_MDL$). Realizou-se também um desenho experimental fatorial multi-nível, os dados de entrada para realizar este experimento são mostrados na Tabela 6.7.

Tabela 6.7: Desenho experimental fatorial multi-nível para o algoritmo *IG+MDL*

| **Fatores** | **Mínimo** | **Máximo** | **Níveis** |
|---|---|---|---|
| $d$ | 0,2N | 0,4N | 3 |
| $Iter\_Bl$ | 3 | 5 | 3 |
| $Iter\_MDL$ | 4 | 6 | 3 |
| $Iter\_Alg$ | 100 | 200 | 3 |

Baseados nos dados da tabela 6.7, foram achadas 81 condições experimentais pelo desenho experimental fatorial multi-nível, estas são mostradas na Tabela B.3 (ver Apêndice B).

Após testar todas as condições experimentais em cada uma das classes de instâncias, a padronização dos valores obtidos por cada condição foi realizada conforme o *RPD*. Logo, é realizado o teste de Kruskal-Wallis, o qual é mostrado na Tabela 6.8, nesta tabela são mostradas as 40 condições experimentais que obtiveram os melhores resultados, as 41 restantes não foram consideradas já que os valores obtidos por estas foram excessivamente altos.

Tabela 6.8: Teste de Kruskall-Wallis para as CEX do *IG+MDL*

| **Condição Exp.** | **Tamanho da amostra** | **Ranking médio** |
|---|---|---|
| $CEX\_1$ | 10 | 265,1 |
| $CEX\_2$ | 10 | 60,3 |
| $CEX\_3$ | 10 | 335,6 |
| $CEX\_5$ | 10 | 267,9 |
| $CEX\_6$ | 10 | 251,5 |
| $CEX\_8$ | 10 | 47,8 |
| $CEX\_10$ | 110 | 320,5 |
| $CEX\_12$ | 110 | 293,3 |
| $CEX\_14$ | 110 | 277,7 |
| $CEX\_16$ | 110 | **13,95** |
| $CEX\_17$ | 110 | 267,6 |
| $CEX\_19$ | 110 | 307 |
| $CEX\_20$ | 110 | 15,8 |
| $CEX\_22$ | 110 | 199,3 |
| $CEX\_24$ | 110 | 361,6 |
| $CEX\_26$ | 110 | 95,6 |
| $CEX\_28$ | 110 | 303,9 |

*Continua na próxima página*

Tabela 6.8 – *Continuação da página anterior*

| **Condição Exp.** | **Tamanho da amostra** | **Ranking médio** |
|---|---|---|
| $CEX\_30$ | 110 | 182,4 |
| $CEX\_35$ | 110 | 21,65 |
| $CEX\_36$ | 110 | 317,1 |
| $CEX\_37$ | 110 | 100 |
| $CEX\_42$ | 110 | 250,1 |
| $CEX\_46$ | 110 | 316,1 |
| $CEX\_51$ | 110 | 90,9 |
| $CEX\_53$ | 110 | 145,3 |
| $CEX\_55$ | 110 | 51,5 |
| $CEX\_56$ | 110 | 169,8 |
| $CEX\_57$ | 110 | 323,9 |
| $CEX\_58$ | 110 | 378,3 |
| $CEX\_59$ | 110 | 145,7 |
| $CEX\_61$ | 110 | 310 |
| $CEX\_62$ | 110 | 59,3 |
| $CEX\_64$ | 110 | 198,8 |
| $CEX\_65$ | 110 | 217,8 |
| $CEX\_68$ | 110 | 342,1 |
| $CEX\_71$ | 110 | 154,4 |
| $CEX\_73$ | 110 | 174,3 |
| $CEX\_77$ | 110 | 149,3 |
| $CEX\_79$ | 110 | 53,7 |
| $CEX\_81$ | 110 | 183 |
| Valor-P = 0,0 | | |

Segundo a terceira coluna (ranking médio) da tabela do teste Kruskall-Wallis existem 2 condições experimentais que obtêm valores muito menores com relação às outras condições. Estas condições experimentais são: $CEX\_16$ e $CEX\_20$ com ranking médios de $13,95$ e $15,8$ respectivamente. Para ajudar a selecionar a melhor condição, o gráfico 6.3 mostra os intervalos de confiança S agrupada de cada condição experimental. Neste gráfico se apreciam 11 condições que apresentam valores relativamente menores e próximos entre si, assim, entre estas condições não existe uma diferença significativa, portanto, para o procedimento $IG + MDL$ selecionamos a condição experimental que apresentou o menor ranking médio, sendo a $CEX\_16$ com: $d = 0,2$, $Iter\_Bl = 5$, $Iter\_MDL = 4$, $Iter\_Alg = 200$.

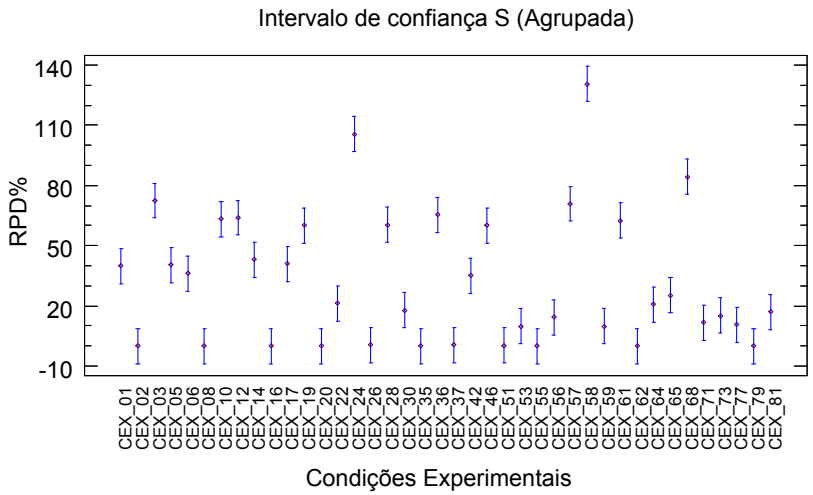



Figura 6.3: Intervalos de confiança S agrupada para às condições experimentais do *IG+MDL*

### 6.4.4 Calibração dos parâmetros do algoritmo $IG + F\&O$

A calibração deste procedimento é similar às realizadas anteriormente. O *IG+F&O*, como o algoritmo *IG+MDL*, contém os parâmetros básicos do *IG*: parâmetros de destruição ($d$), número de iterações da busca local ($Iter\_Bl$), número de iterações totais do algoritmo ($Iter\_Alg$), além destes parâmetros, o *IG+F&O* contém o número de produtos ($n\_Sub$) a formar o subconjunto de produtos a serem otimizados. Realizou-se também um desenho experimental fatorial multi-nível. Os dados de entrada para este experimento são mostrados na Tabela 6.9.

Tabela 6.9: Desenho experimental multi-nível para o $IG + F\&O$

| **Fatores** | **Mínimo** | **Máximo** | **Níveis** |
|---|---|---|---|
| $d$ | 0,2N | 0,4 N | 3 |
| $Iter\_Bl$ | 3 | 7 | 3 |
| $n\_Sub$ | 0,2N | 0,4N | 3 |
| $Iter\_Alg$ | 100 | 200 | 3 |

Com os dados da Tabela 6.9, foram obtidas 81 condições experimentais, as quais são mostradas na Tabela B.4 (ver Apêndice B).

Da mesma forma que nos experimentos anteriores foi realizado o teste de Kruskal-Wallis para as 40 melhores condições experimentais. O teste é mostrado na Tabela 6.10, na última coluna desta tabela, duas condições experimentais apresentam os menores

ranking médios, relativamente próximos entre si, e bem distantes das outras condições experimentais. Para ajudar a selecionar a melhor condição experimental é mostrado o gráfico de intervalos de confiança S agrupada (Figura 6.4). No gráfico se observam 10 condições experimentais que apresentam valores relativamente menores, porém, não existe uma diferença significativa entre estas, assim, é indiferente selecionar qualquer uma destas. Para este algoritmo é selecionada a condição experimental com o menor ranking médio, sendo a $CEX\_26$ com: $d = 0,2N$, $Iter\_Bl = 3$, $n\_Sub = 0,3N$, $Iter\_Alg = 200$.

Tabela 6.10: Teste de Kruskall-Wallis para as CEX do *IG+F&O*

| **Condição Exp.** | **Tamanho da amostra** | **Ranking médio** |
|---|---|---|
| $CEX\_2$ | 110 | 338,8 |
| $CEX\_3$ | 110 | 147,7 |
| $CEX\_7$ | 110 | 168,7 |
| $CEX\_8$ | 110 | 142,3 |
| $CEX\_9$ | 110 | 48 |
| $CEX\_13$ | 110 | 178,1 |
| $CEX\_15$ | 110 | 318,6 |
| $CEX\_19$ | 110 | 310,8 |
| $CEX\_21$ | 110 | 275,9 |
| $CEX\_25$ | 110 | 13,45 |
| $CEX\_26$ | 110 | **15** |
| $CEX\_27$ | 110 | 195,1 |
| $CEX\_35$ | 110 | 138,2 |
| $CEX\_37$ | 110 | 266,7 |
| $CEX\_39$ | 110 | 360,7 |
| $CEX\_41$ | 110 | 45,8 |
| $CEX\_42$ | 110 | 305,4 |
| $CEX\_43$ | 110 | 85,6 |
| $CEX\_44$ | 110 | 322,8 |
| $CEX\_45$ | 110 | 301,5 |
| $CEX\_47$ | 110 | 177,2 |
| $CEX\_51$ | 110 | 377,6 |
| $CEX\_52$ | 110 | 20,55 |
| $CEX\_56$ | 110 | 249,3 |
| $CEX\_57$ | 110 | 194,6 |

*Continua na próxima página*

Tabela 6.10 – *Continuação da página anterior*

| Condição Exp. | Tamanho da amostra | Ranking médio |
|---|---|---|
| $CEX\_58$ | 110 | 90 |
| $CEX\_59$ | 110 | 163,9 |
| $CEX\_60$ | 110 | 315 |
| $CEX\_61$ | 110 | 80,9 |
| $CEX\_63$ | 110 | 138,6 |
| $CEX\_64$ | 110 | 307,9 |
| $CEX\_65$ | 110 | 52,4 |
| $CEX\_68$ | 110 | 315,4 |
| $CEX\_69$ | 110 | 215,2 |
| $CEX\_70$ | 110 | 263,6 |
| $CEX\_71$ | 110 | 53,3 |
| $CEX\_74$ | 110 | 334,6 |
| $CEX\_75$ | 110 | 266,4 |
| $CEX\_76$ | 110 | 250 |
| $CEX\_77$ | 110 | 174,4 |
| Valor-P = 0,0 | | |

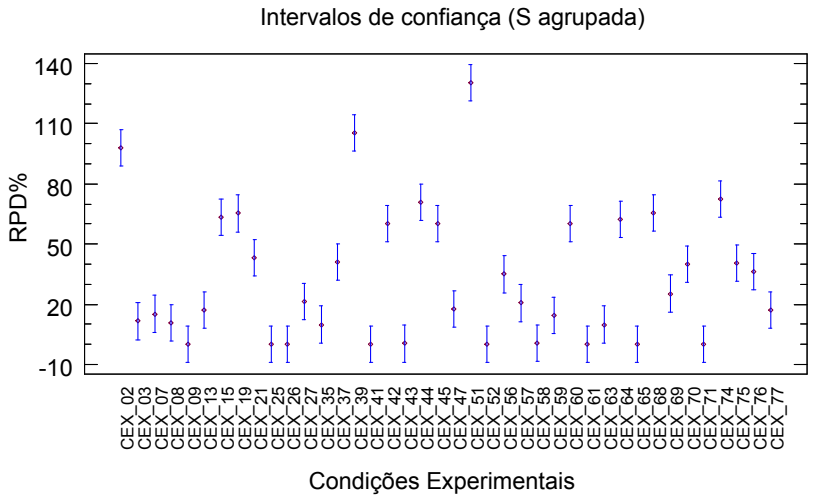



Figura 6.4: Intervalos de confiança S agrupada para às condições experimentais do *IG+F&O*

## 6.5 Resultados computacionais

Nesta seção são descritos os principais resultados computacionais em relação à qualidade da solução e os tempos computacionais obtidos por todos os algoritmos. Os resultados completos, sendo o valor absoluto da função objetivo do modelo matemático, os valores médios da função objetivo obtidos pelas heurísticas da literatura *(TAH, HR)* e dos algoritmos propostos *(IG+HR, IG+MDL, IG+F&O)*, são mostrados no Apêndice C. Os resultados computacionais são divididos em três análises estatísticas. Análise para as 20 classes de instâncias geradas conforme a literatura, instâncias que variam entre as dimensões $(3, 3, 3)$ a $(15, 15, 15)$. Estas instâncias são divididas em classes de pequeno e médio porte. As classe de instâncias de pequeno porte, variam nos tamanhos de $(3, 3, 3)$ a $(7, 7, 7)$. As classes de médio porte variam nos tamanhos de $(5, 5, 10)$ a $(15, 15, 15)$. A segunda análise considera as 81 classes de instâncias propostas, consideradas instâncias de grande porte, e iniciam com a classe $(20, 10, 10)$ e finalizam na classe $(60, 20, 20)$. Finalmente é realizado mais uma análise considerando todas as 101 classes de instâncias.

### 6.5.1 Análises dos resultados das instâncias de pequeno e médio porte

Como foi dito anteriormente, Ramezanian & Saidi-Mehrabad [2013] formularam o modelo matemático para o *PDSLFS*. Porém este só resolve otimamente as instâncias de pequeno porte em um tempo computacional razoável. Para todos os procedimentos, incluso o modelo matemático, foi imposto um tempo máximo de 3600 segundos, usado também como tempo máximo em Ramezanian & Saidi-Mehrabad [2013].

Para as classes de instâncias de pequeno porte, a Tabela 6.11 mostra a média dos valores ótimos da função objetivo encontrado pelo modelo matemático (coluna MIP). Além disso, nesta tabela mostra-se o valor médio da função objetivo por classe de instância tanto para os algoritmos da literatura como para os algoritmos propostos.

Tabela 6.11: Média da função objetivo para instâncias de pequeno porte

| C. de Problema | MIP | HR | TAH | IG+HR | IG | IG+F&O |
|---|---|---|---|---|---|---|
| **(3,3,3)** | 4347,18 | 4370,76 | 4470,42 | 4487,24 | 4461,02 | 4367,16 |
| **(5,3,3)** | 8449,02 | 8493,16 | 8721,90 | 8722,74 | 8709,26 | 8452,62 |
| **(3,5,3)** | 8975,48 | 9003,84 | 9121,62 | 9120,82 | 9128,60 | 8993,51 |
| **(3,3,5)** | 7936,60 | 8045,04 | 8218,64 | 8257,00 | 8216,68 | 7979,28 |
| **(5,5,5)** | 23890,80 | 24005,26 | 24895,00 | 24858,62 | 24757,65 | 23960,65 |
| **(7,5,5)** | 32506,44 | 32818,14 | 34097,34 | 34365,28 | 33775,90 | 32647,32 |
| **(5,7,5)** | 33279,76 | 33611,70 | 34211,22 | 34211,22 | 34491,84 | 33391,03 |
| **(5,5,7)** | 31265,2 | 31435,18 | 32761,592 | 32732,78 | 32802,39 | 31335,16 |
| **(7,7,7)** | 62195,70 | 62770,04 | 66003,16 | 66194,09 | 65220,75 | 62720,63 |

Para uma melhor interpretação dos resultados, na Tabela 6.12 se mostram os *RPD%* médios por classe de instâncias de pequeno porte. Nesta tabela pode ser observado que o algoritmo *IG+F&O* nas nove classes de problemas obtém um erro relativo mais aproximado ao valor exato (valor obtido pelo modelo matemático). A heurística *HR* também obtém valores próximos ao valor exato, porém não tão próximos quanto o *IG+F&O*. Já as heurísticas *TAH, IG+HR* e *IG+MDL*, obtêm valores um pouco distante ao valor ótimo. Com referências a estes três algoritmos, o *IG+MDL* obtém melhores valores médios em seis classes de problemas, seguido pelo *IG+HR* que obtém três valores mais próximos ao valor ótimo e por último o *TAH* que obteve os valores mais distantes para todas as classes de problemas de pequeno porte.

Tabela 6.12: RPD% médio por clase de instância de pequeno porte

| (N,M,T) | HR | TAH | IG+HR | IG | IG+F&O |
|---|---|---|---|---|---|
| **(3,3,3)** | 0,5424% | 2,8349% | 3,2219% | 2,6187% | **0,4596%** |
| **(5,3,3)** | 0,5224% | 3,2297% | 3,2397% | 3,0801% | **0,0427%** |
| **(3,5,3)** | 0,3160% | 1,6282% | 1,6193% | 1,7060% | **0,2008%** |
| **(3,3,5)** | 1,3663% | 3,5537% | 4,0370% | 3,5290% | **0,5378%** |
| **(5,5,5)** | 0,4791% | 4,2033% | 4,0510% | 3,6284% | **0,2924%** |
| **(7,5,5)** | 0,9589% | 4,8941% | 5,7184% | 3,9053% | **0,4334%** |
| **(5,7,5)** | 0,9974% | 2,7989% | 2,7989% | 3,6421% | **0,3344%** |
| **(5,5,7)** | 0,5437% | 4,7861% | 4,6940% | 4,9166% | **0,2238%** |
| **(7,7,7)** | 0,9234% | 6,1217% | 6,4287% | 4,8638% | **0,8440%** |

Como foi dito anteriormente, o modelo matemático só resolve as instâncias de

pequeno porte, já para os problemas de médio porte uma solução em um tempo computacional estabelecido, não é garantida. Por essa razão o modelo matemático para as instâncias de médio e grande porte não é considerado. A seguir mostram-se as análises para as instâncias de médio porte. Na Tabela 6.13, apresenta-se a média da função objetivo por classe de instância de médio porte, obtidos pelos algoritmos comparados.

Tabela 6.13: Média da função objetivo para classe de instâncias de médio porte

| (N,M,T) | HR | TAH | IG+HR | IG+MDL | IG+F&O |
|---|---|---|---|---|---|
| **(5,5,10)** | 48565,1 | 49056,3 | 48591,476 | 48324,608 | 48857,5 |
| **(5,10,5)** | 36095,48 | 38329,994 | 37119,686 | 37129,064 | 35659,8 |
| **(7,7,10)** | 75962,52 | 74755,924 | 73828,672 | 73997,794 | 72321,434 |
| **(7,10,7)** | 83378,14 | 84567,538 | 80156,532 | 80605,326 | 77898,586 |
| **(10,5,5)** | 47018,9 | 45298,5 | 45179,644 | 45233,69 | 41667,02 |
| **(10,7,7)** | 89946,18 | 89341,08 | 89226,74 | 89357,08 | 82925,7 |
| **(10,10,10)** | 164564,05 | 135309,32 | 134311,52 | 134443,5 | 135875,96 |
| **(10,10,15)** | 183800,4 | 197236,6 | 196532,4 | 196825,36 | 161138,14 |
| **(10,15,10)** | 210690,5 | 192939,88 | 185725,04 | 186836,84 | 192939,88 |
| **(15,10,10)** | 181037 | 169965,74 | 167388,42 | 166019,22 | 165795,92 |
| **(15,15,15)** | 664939,4 | 632993,5 | 626931,72 | 628681,22 | 6385885,8 |

Para a classe de instância de médio porte, um valor exato ou ótimo não é conhecido, portanto, o valor referência para o cálculo do *RPD* será a menor média da função objetivo por classe de instâncias entre todos os algoritmos. Definido este valor, na Tabela 6.14 mostra-se os *RPD%* médios por classe de problemas (médio porte). Nesta tabela, o algoritmo *IG+F&O* novamente apresenta superioridade entre os demais algoritmos, desta vez, em sete casos de onze possíveis ele consegue o menor valor, sendo estes os valores referência para o grupo de problemas de médio porte. O segundo algoritmo com melhor desempenho é o *IG+HR* encontrando três vezes o menor valor dos onze possíveis. O *IG+MDL* obtém valores um pouco altos em comparação com o *IG+F&O*; porém, os valores obtidos por este algoritmo estão relativamente próximos ao *IG+HR*. Segue o algoritmo *TAH* que nessas classes de problemas alcança seis melhores valores que a heurística *HR*, este último obteve valores bem distantes dos outros algoritmos.

Tabela 6.14: RPD% médio para as classes de instâncias de médio porte

| (N,M,T) | HR | TAH | IG+HR | IG+MDL | IG+F&O |
|---|---|---|---|---|---|
| **(5,5,10)** | 0,4977 % | 1,5141 % | 0,5522 % | **0,0000%** | 1,1027% |
| **(5,10,5)** | 1,2218 % | 7,4880 % | 4,0939 % | 4,1202% | **0,0000%** |
| **(7,7,10)** | 5,0346 % | 3,3662 % | 2,0841 % | 2,3179% | **0,0000%** |
| **(7,10,7)** | 7,0342 % | 8,5611 % | 2,8986 % | 3,4747% | **0,0000%** |
| **(10,5,5)** | 12,8444 % | 8,7155 % | 8,4302 % | 8,5599% | **0,0000%** |
| **(10,7,7)** | 8,4660 % | 7,7363 % | 7,5984 % | 7,7556% | **0,0000%** |
| **(10,10,10)** | 22,5242 % | 0,7429 % | **0,0000%** | 0,0983% | 1,1648% |
| **(10,10,15)** | 14,0639 % | 22,4022 % | 21,9652% | 22,1470% | **0,0000%** |
| **(10,15,10)** | 13,4422 % | 3,8142 % | **0,0000%** | 0,5986% | 3,8847% |
| **(15,15,15)** | 9,1927 % | 2,5150 % | 0,9605 % | 0,1347% | **0,0000%** |
| **(15,15,15)** | 6,0625 % | 0,9667 % | **0,0000%** | 0,2791% | 1,4282% |

A fim de avaliar o desempenho dos algoritmos propostos em comparação com os algoritmos da literatura e determinar se existem diferenças estatísticas significativas entre as soluções *(RPD%)* obtidas por estes algoritmos, uma análise estatística é realizada. Inicialmente foi realizada uma análise de variância (ANOVA), mas as suposições desta não foram satisfeitas. No caso de alguma suposição da ANOVA não ser satisfeita o teste de Kruskal-Wallis é uma alternativa não paramétrica que é usada para provar se existe ou não diferenças estatísticas significativas entre várias amostras de resultados. Entenda-se por amostra o conjunto de dados obtido por um algoritmo.

O teste Kruskal-Wallis é usado para testar a hipótese nula $H_0$ de que "todas as medianas das amostras são iguais" contra a hipótese alternativa $H_1$ de que "ao menos uma das medianas é diferente". As hipóteses são escritas formalmente da seguinte forma:

$H_0 : Me_1 = Me_2 = Me_3 = Me_4 = Me_5$.
$H_1 : Me_i \neq Me_j$ para pelo menos um par $i, j$.
Onde;
$Me_1$ : Mediana dos resultados obtidos por *HR*.
$Me_2$ : Mediana dos resultados obtidos por *TAH*.
$Me_3$ : Mediana dos resultados obtidos por *IG+HR*.
$Me_4$ : Mediana dos resultados obtidos por *IG+MDL*.
$Me_5$ : Mediana dos resultados obtidos por $IG + F\&O$.

$Me_i$ : e $Me_j$ são duas medianas quaiquer dentre as 5 amostras.

A seguir mostra-se o teste de Kruskal-Wallis a fim de determinar se existe ou não diferenças estatísticas significativas entre as amostras dos algoritmos. As amostras contêm os resultados das instâncias de pequeno e médio porte. Na Tabela 6.15 mostra-se o resultado do teste de Kruskal-Wallis. A coluna "Amostra" corresponde aos nomes dos algoritmos, a segunda coluna contém o "Tamanho da amostra" por cada algoritmo. Note que o tamanho da amostra é 100 para cada algoritmo, pois são 20 classes de instâncias com 5 exemplares para cada. Na última coluna da tabela mostra-se o "Ranking médio" por algoritmos. Note que nesta coluna o algoritmo com o menor ranking médio é o $IG+F\&O$ com 148.8, indicando que este obteve menores $RPD's$ para as classes de instâncias de pequeno e médio porte, seguido pelos algoritmos *IG+MDL* (com 251, 6), $IG + HR$ (com 272, 1), *HR* (com 275, 2) e finalmente o *TAH* que obteve um ranking médio de 304, 8. O teste de Kruskal-Wallis foi realizado com um nível de confiança do 95%, pelo que o *valor-P = 0,0* indica que existe uma diferença estatística significativa entre as amostras, assim, a hipótese nula ($H_0$) é rejeitada.

Tabela 6.15: Teste Kruskal-Wallis para instâncias de pequeno e médio porte

| **Amostra** | **Tamnho da amostra** | **Ranking médio** |
|---|---|---|
| *RH* | 100 | 275,2 |
| *TAH* | 100 | 304,8 |
| *IG+HR* | 100 | 272,1 |
| *IG+MDL* | 100 | 251,6 |
| *IG+F&O* | 100 | 148,8 |
| *Valor-P* = 0,0 | | |

Sendo $H_0$ rejeitada, o *valor-P=0,0* indicando uma diferença estatística significativa entre as amostras e o menor valor de ranking médio, não é informação suficiente para concluir qual algoritmo obtém melhores soluções para as instâncias tratadas. Por esta razão é realizada uma análise estatística final baseada no "gráfico de caixa".

O gráfico de caixa, inventado por John Tukey (Montgomery [2001]), oferece o resumo dos dados de cada amostra. A caixa central contém a metade dos dados, se estendendo desde o quartil inferior até o quartil superior. As linhas que se estendem nos extremos inferiores e superiores da caixa (bigodes) mostram o valor mais baixo e mais alto da amostra, exceto por aqueles valores que estejam extremadamente

longe das caixas. A linha horizontal no interior da caixa corresponde ao valor da mediana amostral, enquanto que o sinal (+) representa a média amostral. Os cortes incidentes na linha que representa a mediana amostral são intervalos de confiança aproximados para a mediana populacional. Para facilitar a notação o gráfico de caixa e bigode será nomeado simplesmente como gráfico de "caixa" ou "boxplot".

Na Figura 6.5 é apresentado o gráfico de caixa para comparar os 5 algoritmos. Este gráfico está diretamente relacionado como a Tabela 6.15 do teste de Kruskal Wallis. A diferença estatística significativa demonstrada pelo *Valor-P* entre as amostras, é também visível no gráfico de caixa. Note que a caixa de cada amostra apresenta cortes que representam intervalos de confiança da mediana. Se cortes de caixas diferentes se sobrepõem, o caso dos cortes das caixas dos algoritmos $TAH$ e $IG{+}HR$ significa que não existe diferenças estatísticas significativas, caso contrário, quando os cortes de caixas diferentes não se sobrepõem, situação que apresenta o $IG{+}F\&O$ em relação a qualquer outro algoritmo, desta forma e por meio do gráfico pode-se concluir que existe uma diferença estatística significativa com um nível de confiança do 95% entre o $IG{+}F\&O$ e os demais algoritmos. Outro fator importante neste gráfico é que também se aprecia qual algoritmo obtém menores valores, é o caso do $IG{+}F\&O$ já que a caixa que o representa encontra-se próxima ao eixo horizontal, indicando que encontra valores mais próximos a zero que os outros algoritmos. Um gráfico de caixa ampliado para uma melhor visualização é mostrado na Figura 6.6.

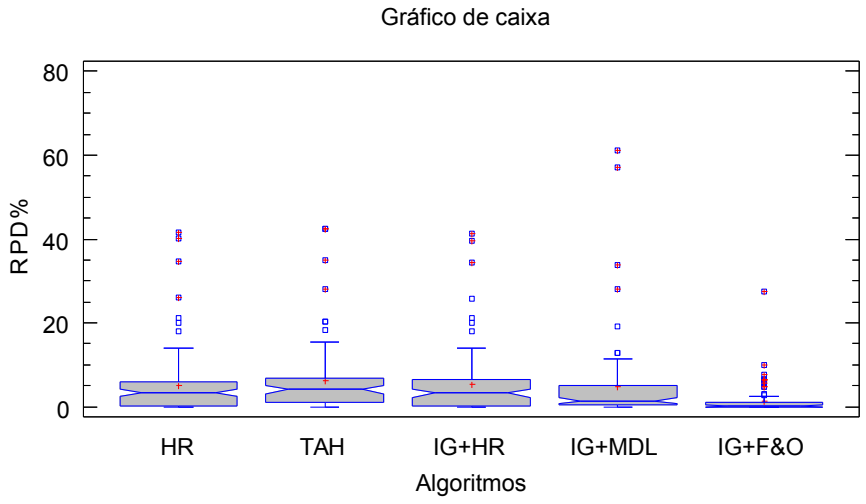



Figura 6.5: Gráfico de caixa dos RPD% para instâncias de pequeno e médio porte.

Além de avaliar os algoritmos com relação à qualidade da solução obtida é também necessária a comparação dos tempos de execução dos mesmos. Primeiramente

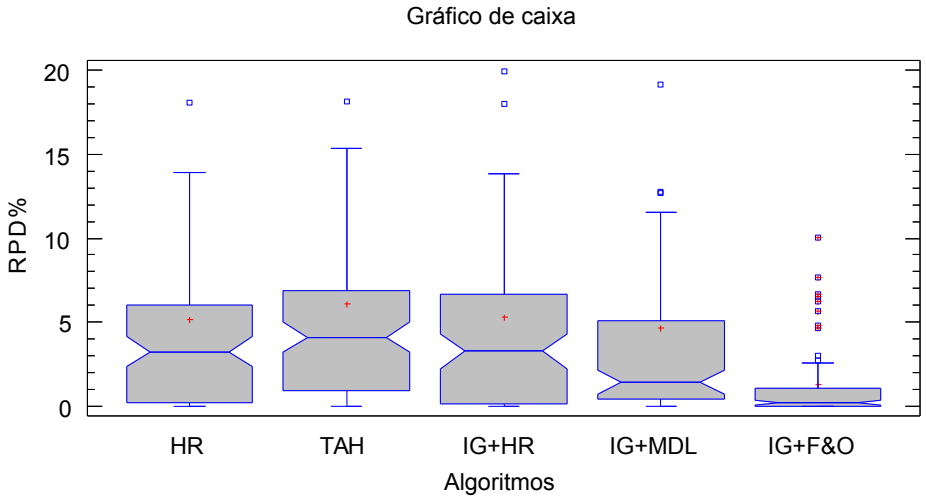



Figura 6.6: Gráfico de caixa ampliado dos RPD% para instâncias de pequeno e médio porte.

são discutidos os tempos médios por classe de instâncias e posteriormente uma análise estatística, é realizada, considerando cada uma das instâncias.

Na Tabela 6.16 mostra-se o tempo computacional médio por classe de instâncias de pequeno porte. O algoritmo *IG+MDL*, para todas as nove classes de instâncias, obteve os menores tempos computacionais. O algoritmo *IG+F&O* foi o segundo que resolveu os problemas em um menor tempo computacional, ele obteve nas cinco últimas classes de instâncias melhores tempos que os algoritmos *TAH*, *IG+HR*, e *HR*. O terceiro algoritmo em resolver as classes de instância em um menor tempo é o *TAH*. Ele obtém menores tempos de execução que as heurísticas *IG+HR* e *HR*, isto é explicado pelo fato que estas duas heurísticas utilizam a execução do modelo matemático. Vale ressaltar que a heurística *HR* apresenta tempos muitos altos em comparação com os outros algoritmos, por exemplo: na última classe de instância pertencente às de pequeno porte, com 7 tarefas, 7 máquinas e 7 períodos, o *IG+HR* demora mais que o *IG+MDL*, *IG+F&O* e *TAH* com um tempo de 0.19 segundos, e a heurística *HR* demorou aproximadamente 30 segundos.

Na Tabela 6.17, mostra-se o tempo médio computacional por classe de instância de médio porte. O algoritmo *IG+MDL* continua obtendo os menores tempos de execução em todas as classes de problemas em relação a todos os algoritmos. Os outro algoritmos que obtiveram os menores tempos são o *IG+F&O* e o *TAH*, sendo o *IG+F&O* mais eficiente em todas as classes de instâncias que o *TAH*. Nos algoritmos

Tabela 6.16: Tempos computacionais médios para classe de instâncias de pequeno porte

| (N,M,T) | MIP | HR | TAH | IG+HR | IG+MDL | IG+F&O |
|---|---|---|---|---|---|---|
| **(3,3,3)** | 0,1020 | 0,0400 | 0,0178 | 0,0128 | **0,0128** | 0,0226 |
| **(5,3,3)** | 1,1700 | 0,1600 | 0,0242 | 0,0156 | **0,0140** | 0,0244 |
| **(3,5,3)** | 0,028 | 0,048 | 0,0232 | 0,0144 | **0,0138** | 0,0234 |
| **(3,3,5)** | 0,7975 | 0,2725 | 0,0236 | 0,0250 | **0,0210** | 0,0384 |
| **(5,5,5)** | 7,015 | 1,2875 | 0,049 | 0,0518 | **0,0272** | 0,047 |
| **(7,5,5)** | 67,71 | 2,864 | 0,0686 | 0,075 | **0,0244** | 0,057 |
| **(5,7,5)** | 86,315 | 2,176 | 0,0646 | 0,0678 | **0,0296** | 0,0506 |
| **(5,5,7)** | 45,992 | 2,676 | 0,0704 | 0,101 | **0,0400** | 0,0672 |
| **(7,7,7)** | 3603,952 | 30,4100 | 0,1270 | 0,1900 | **0,0404** | 0,1246 |

*HR* e *IG+HR* note-se o consumo de tempo pela execução da formulação matemática, mesmo que o *IG+HR* apresenta menores tempos de execução que a heurística *HR*, estes tempos estão bastante distantes dos tempos obtidos pelo *IG+MDL* o qual obteve os menores tempos de execução. Os tempos alcançados pela heurística *HR* são excessivamente altos em comparação com todos os outros algoritmos.

Tabela 6.17: Tempo computacional médio por classe de instâncias de médio porte

| (N,M,T) | HR | TAH | IG+HR | IG+MDL | IG+F&O |
|---|---|---|---|---|---|
| **(5,5,10)** | 11,8600 | 0,2664 | 0,4268 | **0,1070** | 0,1320 |
| **(5,10,5)** | 2,4620 | 0,1156 | 0,3426 | **0,0746** | 0,0898 |
| **(7,7,10)** | 73,4540 | 0,5004 | 0,926 | **0,0960** | 0,2238 |
| **(7,10,7)** | 322,9320 | 0,3486 | 0,7504 | **0,0968** | 0,1608 |
| **(10,5,5)** | 273,3952 | 0,1514 | 0,5452 | **0,0550** | 0,1218 |
| **(10,7,7)** | 424,8400 | 0,3966 | 0,8752 | **0,0960** | 0,2042 |
| **(10,10,10)** | 119,4900 | 1,1736 | 2,0682 | **0,1832** | 0,4744 |
| **(10,10,15)** | 378,4360 | 2,7184 | 2,5070 | **0,2728** | 0,7224 |
| **(10,15,10)** | 160,4950 | 1,8676 | 2,4415 | **0,2652** | 0,6356 |
| **(15,10,10)** | 301,4096 | 2,1818 | 2,5268 | **0,2894** | 0,826 |
| **(15,15,15)** | 249,6120 | 16,4634 | 28,2512 | **0,6068** | 3,4986 |

Dado que as análises realizadas anteriormente foram baseadas só nos tempos médios por classe de instâncias, é considerada uma análise estatística para comprovar a diferença estatística significativa que existe entre todos os tempos de execução obtidos pelos algoritmos. Nesta análise são considerados os tempos de execução dos algoritmos para cada uma das instâncias pertencentes às classes de pequeno e médio porte. O teste de Kruskal - Wallis é realizado para comprovar se existe diferença estatística significativa entre as medianas das amostras, neste caso, a amostra corresponde aos tempos de execução obtidos por cada algoritmo em cada instância. Na Tabela 6.18

mostra-se o resultado do teste de Kruskal-Wallis para comparar os tempos de execução dos algoritmo. Dado que o teste foi realizado com um nível de confiança do 95%, o *Valor-P = 0,0* indica que existe uma diferença estatística significativa entre as amostras, rejeitando a hipótese nula. O ranking médio do *IG+MDL* (55, 73) sendo bem menor em relação ao outros, indica que obteve os menores tempos de execução, seguido na ordem pelos algoritmos *IG+F&O, TAH e IG+HR* com ranking 211, 0, 250, 9 e 284, 4 respectivamente.

Tabela 6.18: Teste Kruskal-Wallis para instâncias de pequeno e médio porte

| **Amostra** | **Tamnho da amostra** | **Ranking médio** |
|---|---|---|
| *TAH* | 100 | 250,90 |
| *IG+HR* | 100 | 284,40 |
| *IG+MDL* | 100 | 55,73 |
| *IG+F&O* | 100 | 211,00 |
| *Valor-P* = 0,0 | | |

Para comprovar a diferença estatística encontrada pelo *valor-P*, é utilizado o gráfico de caixa, mostrado na Figura 6.7, porém, Neste gráfico não se aprecia bem a diferenças que existem entre as amostras dado que os tempos de execução da heurística *HR* para estas classes de instâncias foram muitos altos em comparação com os outros algoritmos, por isso, foi necessário não considerar esta heurística. Um novo gráfico de caixas foi necessário e este é apresentado na Figura 6.8. Neste último gráfico se observa a diferença estadística significativa desde que os cortes das caixas de cada amostra não se sobrepõem. No gráfico também se observa que o algoritmo *IG+MDL* obtém os valores mais próximos a zero. O fato que o desenho da caixa do algoritmo *IG+MDL* é só uma "linha" significa que os valores obtidos por este são muito parecidos ou muito próximos, em outras palavras indica a robustez deste algoritmo em relação aos tempos de execução.

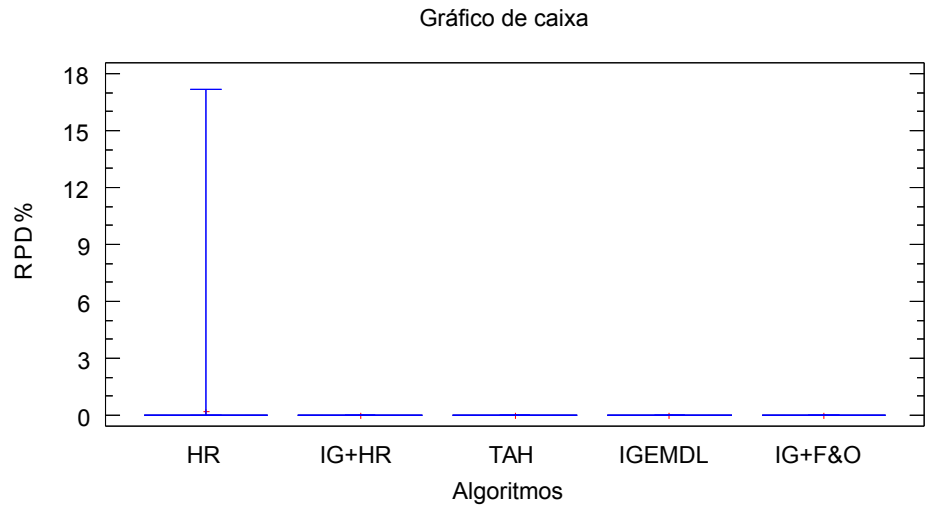



Figura 6.7: Gráfico de caixa dos RPD% (tempos) para instâncias de pequeno e médio porte.

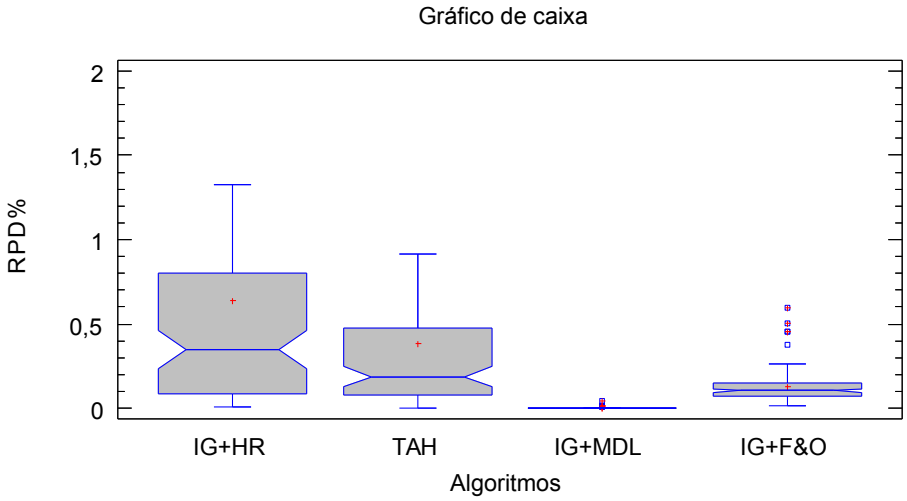



Figura 6.8: Gráfico de caixa ampliado dos RPD% (tempos) para instâncias de pequeno e médio porte.

### 6.5.2 Análises de resultados para classes de instâncias de grande porte

Para as classes de instâncias de grande porte, as quais são compreendidas entre os tamanhos de $(20, 10, 10)$ até $(60, 20, 20)$ é também realizado uma análise sobre os *RPDs* médios. Posteriormente é realizado um análise estatística para verificar se os resultados são significativamente diferentes.

Para as classes de instâncias de grande porte é também realizado o teste de Kruskal-Wallis, o qual é apresentado na Tabela 6.19. O *valor-P* da tabela indica uma diferença estatística significativa entre as amostras, o tamanho da amostra é de 405 já que são 81 classes de problemas e 5 exemplares por cada uma. Na última coluna pode-se notar que existe bastante diferença entre o ranking médio do algoritmo *IG+F&O* e dos algoritmos *IG+MDL* e *IG+HR* e ainda mais do *TAH*. Entre os algoritmos *IG+MDL* e *IG+HR* não existe muita diferença e enquanto o *TAH* permaneceu com o ranking médio bem mais distante destes.

Tabela 6.19: Teste Kruskal-Wallis para as instâncias de grande porte

| **Amostra** | **Tamnho da amostra** | **Ranking médio** |
|---|---|---|
| *TAH* | 405 | 1285,0 |
| *IG+HR* | 405 | 893,0 |
| *IG+MDL* | 405 | 830,9 |
| *IG+F&O* | 405 | 233,4 |
| Valor-P = 0,0 | | |

A diferença estatística significativa demonstrada pelo *Valor-P* e o melhor desempenho obtido pelo algoritmo *IG+F&O* comprovada pelo ranking médio, pode-se observar também na Figura 6.9 que mostra gráfico de caixa para as classes de instância de grande porte. Nesta figura, observa-se que os algoritmos *IG+HR* e *IG+MDL* não apresentam diferença estatística significativa entre si, já que os cortes incidentes na mediana não se sobrepõem. Já os algoritmos *TAH* e *IG+F&O* tem diferença estatística significativa em comparação com todos os algoritmos, vale notar que o algoritmo *TAH* apresenta diferenças porque obteve valores muito altos e o *IG+F&O* porque todos os valores alcançados por este são os menores entre todas as amostras e estão bem próximos à zero.

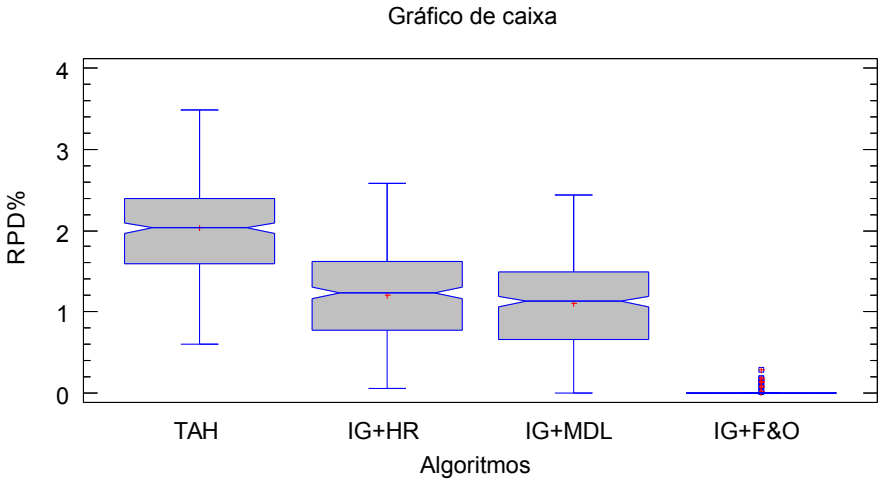



Figura 6.9: Gráfico de caixa dos RPD% para instâncias de grande porte.

Uma análises estatística é também realizada para comparar os tempos de execução dos algoritmos para as classes de instâncias de grande porte. O resultado de teste de Kruskal - Wallis é mostrado na Tabela 6.20. Novamente o *Valor-P* do teste indica a diferença estatística significativa entre os tempos de execução e o ranking médio indica que o algoritmo que apresenta menores tempos é o *IG+MDL* e pouco mais distante o *IG+F&O* seguindo o *TAH* e finalmente o *IG+HR*. Os dados obtidos neste teste são similares aos testes realizados para a classe de problemas de pequeno e médio porte. Note-se que nas duas análises realizadas os algoritmos ocupam a mesma posição quanto ao tempo de execução. Os algoritmos ordenados pelo tempo de execução de menor a maior são: $IG+MDL \rightarrow IG+F\&O \rightarrow TAH \rightarrow IG+HR$.

Tabela 6.20: Teste Kruskal-Wallis para problema de grande porte

| **Amostra** | **Tamnho da amostra** | **Ranking médio** |
|---|---|---|
| *TAH* | 405 | 932,0 |
| *IG+HR* | 405 | 1238,0 |
| *IG+MDL* | 405 | 210,7 |
| *IG+F&O* | 405 | 861,6 |
| *Valor-P* = 0,0 | | |

Os dados obtidos no teste de Kruskal-Wallis são relacionados com o gráfico de caixas para as mesmas classes de instâncias. Na Figura 6.10 (gráfico de caixa) pode-se observar que não existe nenhuma relação entre os rankings médios dos algoritmos, já que cada ranking está distante de outro, isto é apreciado no gráfico, desde que os cortes

de cada caixa não se sobrepõem com nenhum outro corte de outra caixa. O significado disto é uma diferença estatística significativa das amostras entre si. Neste gráfico também se observa novamente a robustez do algoritmo *IG+MDL* em relação aos tempos de execução, sendo sua caixa de dados uma "linha". O algoritmo *IG+F&O* com valores pouco mais altos que o *IG+MDL*, também representa certa robustez em comparação com os algoritmos *TAH* e *IG+HR*, tendo este último os valores mais distantes e altos em relação aos outros algoritmos. Realizado este teste, pode-se concluir que o *IG+HR* para classes de problemas de grande porte não obtém soluções em um tempo computacional razoável. Como foi realizado anteriormente, uma ampliação do gráfico de caixa é necessária para uma melhor visualização e esta é apresentada na Figura 6.11.

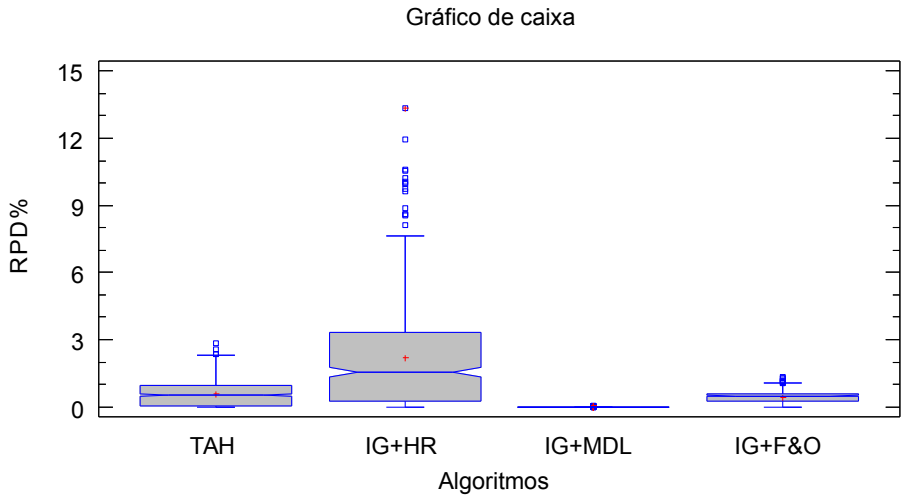



Figura 6.10: Gráfico de caixa dos RPD% (tempo) para instâncias de grande porte.

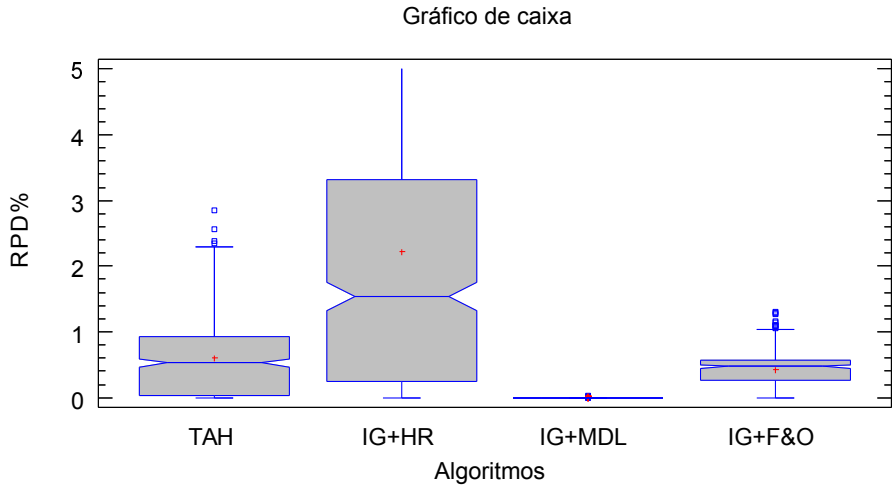



Figura 6.11: Gráfico de caixa ampliada dos RPD% (tempo) para instâncias de grande porte.

### 6.5.3 Análises para todas as classes de instâncias

Nesta seção os resultados são analisados considerando todas as classes de problemas. Na Tabela 6.21 mostra-se o teste de Kruskal - Wallis para todas as instâncias de problemas. Na coluna "tamanho da amostra" mostra 505 para cada algoritmo já que são 101 classes de problemas e 5 exemplares por classe. O *valor-P=0,0*, resultado do teste prova uma diferença estatística significativa entre as medianas das amostras, rejeitando $H_0$. O ranking médio do *IG+F&O* é o menor e com ampla diferença dentre todas as amostras. Os rankings dos algoritmos *IG+MDL* e *IG+HR* estão relativamente próximos, enquanto o ranking do *TAH* encontra-se bem distantes dos demais.

Tabela 6.21: Teste Kruskal-Wallis para todas as instâncias

| **Amostra** | **Tamanho da amostra** | **Ranking médio** |
|---|---|---|
| *TAH* | 505 | 1499,0 |
| *IG+HR* | 505 | 1128,0 |
| *IG+MDL* | 505 | 1069,0 |
| *IG+F&O* | 505 | 346,9 |
| *Valor-P*= 0,0 | | |

Sendo o *Valor-P* igual a zero e o ranking médio para o algoritmo *IG+F&O* o menor entre todos, pode-se concluir que este algoritmo é o que apresenta um melhor desempenho para todas as classes de problemas. Esta afirmação pode-se comprovar

na Figura 6.12, gráfico de caixa relacionado ao teste de Kruskall-Wallis. No gráfico é como se espera, o *IG+F&O* obtém os menores valores, de fato bem próximos a zero. Entre os algoritmos *IG+HR* e *IG+MDL* não existe diferença estatística significativa e o *TAH* apresenta uma diferença estatística significativa em relação a todos os algoritmos, obtendo valores muito distantes às demais amostras. Para uma melhor visualização, o gráfico é ampliado e é mostrado na Figura 6.13.

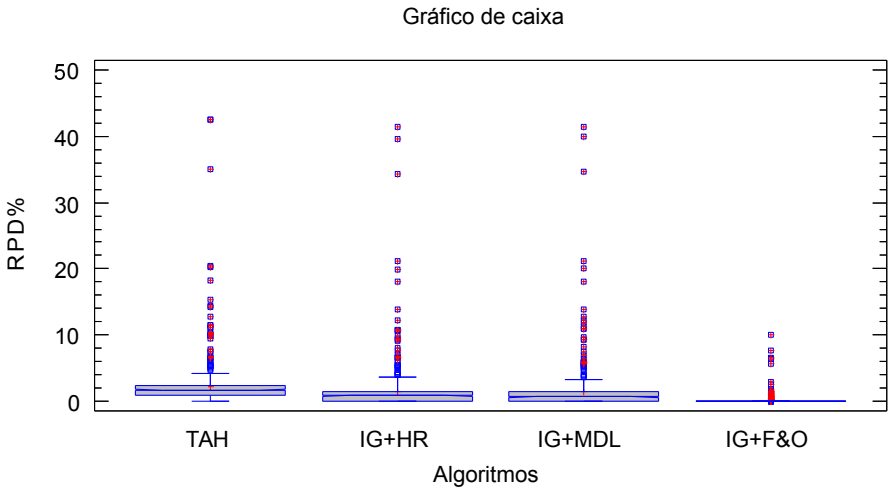



Figura 6.12: Gráfico de caixa dos RPD% para todas as instâncias

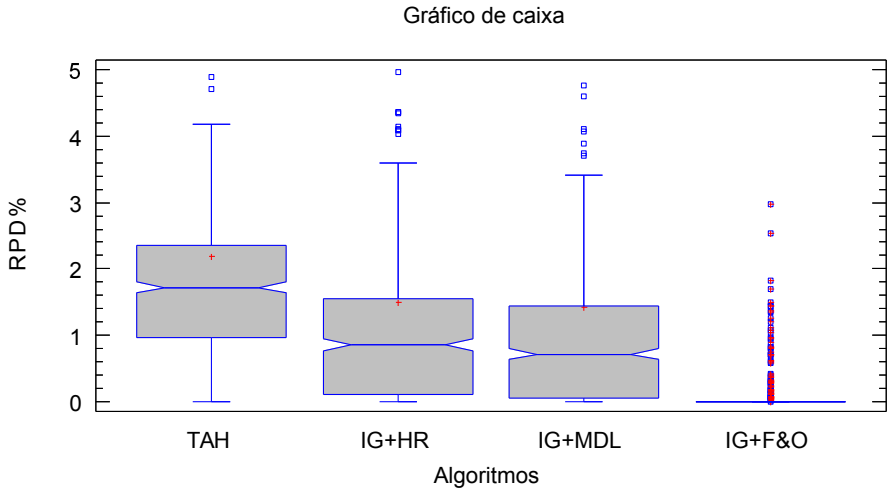



Figura 6.13: Gráfico de caixa ampliado dos RPD% para todas as instâncias

A análise para os tempos de execução dos algoritmos é realizada através do teste de Kruskal-Wallis o qual é mostrado na Tabela 6.22. Nesta Tabela o *Valor-P* formalmente indica que pelo menos uma mediana das amostras é significativamente diferentes

das medianas das outras amostras. Como o ranking médio do *IG+MDL* é o menor de todos espera-se que este algoritmo seja o que apresenta a diferença significativa. Dado que os rankings médios dos outros algoritmos estão relativamente próximos dificulta uma conclusão só considerando este teste. Para esclarecer realmente qual algoritmo é melhor com relação aos tempos de execução, mostras-se a Figura 6.14, gráfico de caixa relacionado ao teste de Kruskall - Wallis para todas as classes de instância.

Tabela 6.22: Teste Kruskal-Wallis para problema de grande porte

| **Amostra** | **Tamnho da amostra** | **Ranking médio** |
|---|---|---|
| *TAH* | 505 | 1182,0 |
| *IG+HR* | 505 | 1496,0 |
| *IG+MDL* | 505 | 265,8 |
| *IG+F&O* | 505 | 1099,0 |
| *Valor-P* = 0,0 | | |

Note que no gráfico de caixa apresentado na Figura 6.14 existem diferenças estatísticas significativa entre todos os algoritmos, já que nenhum corte de cada caixa se sobrepõem sobre outro corte de outra caixa. Neste gráfico também verifica-se o resultado do teste de Kruskall-WAllis que mostra que o algoritmo *IG+MDL* possui o menor ranking médio (265, 8), seguido dos algoritmos *IG+F&O* com (1099), *TAH* com (1182) e *IG+HR* com (1496), nesta mesma ordem. Para apreciar melhor as diferenças em relação aos tempos de execução, o gráfico é estendido, para uma melhor visualização, e este é mostrado na Figura 6.15.

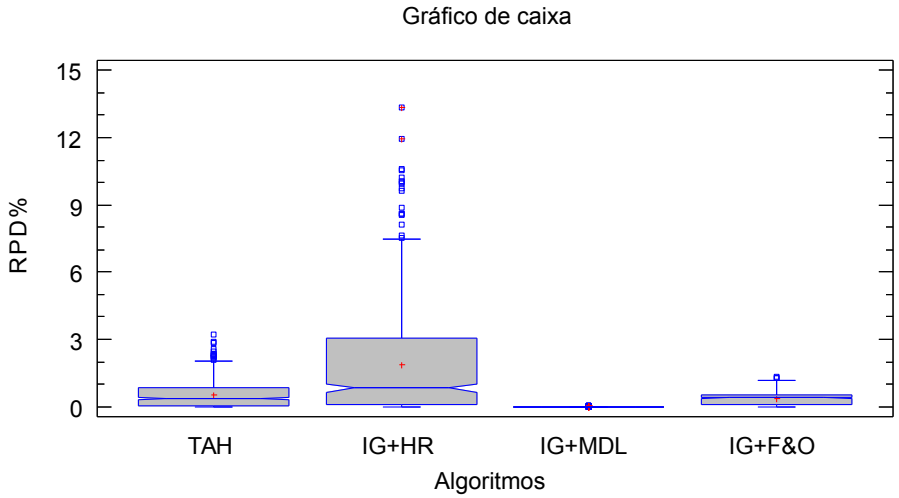



Figura 6.14: Gráfico de caixa dos RPD% (tempo) para todas as classes de instâncias

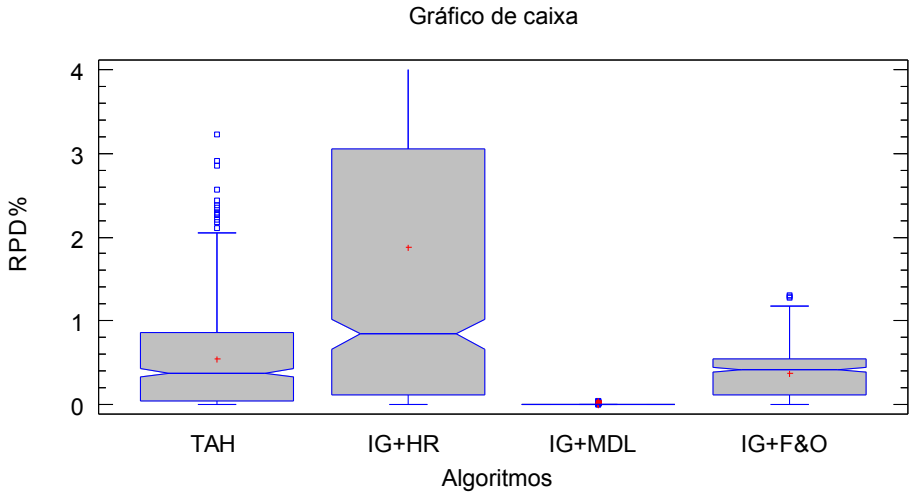



Figura 6.15: Gráfico de caixa ampliado dos RPD% (tempo) para todas as classes de instâncias

## 6.6 Conclusões dos experimentos computacionais

Na seção dos resultados experimentais, primeiramente foi realizada a calibração de cada um dos algoritmos. Com respeito à calibração do *TAH*, o parâmetro número máximo de iterações do algoritmo (Iter_Alg) foi estabelecido por Belo Filho et al. [2012] em 100000, sendo que uma iteração foi definida como o processo de cálculo da função objetivo e da capacidade, dada uma solução. Este tipo de definição de iterações é pouco usual na literatura, porém, para este procedimento temos mantido o conceito de iteração definido pelo autor. Na literatura não existe informação sobre parâmetros do *TAH* para problemas similares, mesmo assim, 100000 como número máximo de iterações parece ser muito alto, por esta razão, na calibração deste procedimento a faixa de cobertura para este parâmetro foi: mínimo de 50000 e máximo de 100000. Não satisfeito com os valores mínimo e máximo estabelecidos para o parâmetro de número de iterações, foi proposto um novo parâmetro que pode favorecer a eficiência do *TAH*, a saber, o número máximo de iterações sem melhora $Iter\_Sm$. Ao estabelecer novas faixas de cobertura para os parâmetros do *TAH* e a adição de um novo parâmetro, os tempos de execução foram consideravelmente reduzidos e a qualidade da solução foi mantida conforme os resultados apresentados por Belo Filho et al. [2012].

Quanto à calibração dos algoritmos propostos (*IG+HR, IG+MDL e IG+F&O*), se tomou como base as pesquisas nas quais o *IG* foi proposto para tratar o flowshop. Acreditamos que neste trabalho é a primeira vez que o *IG* é proposto para o *PDSLFS*.

Portanto, as faixas de cobertura para cada um dos parâmetros do *IG* foram também selecionados em base na experiência na fase de programação do mesmo.

Referindo-nos à qualidade de solução produzida pelos algoritmos, a heurística *HR* é a melhor opção para classes de problemas de pequeno porte. Entretanto, para a classe de problemas de médio e grande porte esta heurística não deve ser considerada já que os tempos de execução são excessivamente altos. Geralmente, com esta heurística, não se obteve uma solução para as classes de problemas de grande porte no tempo de execução estabelecido.

A combinação das heurísticas *IG+HR* reduziu os tempos de execução, porém, a qualidade da solução da heurística *HR* é melhor, isto é esperado, já que é substituído o procedimento exato por um procedimento aproximado para o sequenciamento dos produtos. Note-se que os tempos de execução ao integrar o *IG* à *HR* foram consideravelmente reduzidos permitindo encontrar soluções no tempo limite estabelecido, incluso para instâncias de problemas de grande porte.

Em termos gerais o algoritmo *TAH* obtêm boas soluções em tempo computacional razoável. As soluções iniciais construídas por este algoritmo são de boa qualidade, embora seja custosa computacionalmente já que a formulação matemática é utilizada. Os agentes de melhoria que tratam o sequenciamento de lotes de produção são os bons métodos tradicionais já testados na literatura, portanto, estes agentes geralmente melhoram a sequência de produção. Em relação aos agentes que tratam o dimensionamento de lotes, a contribuição destes é pouca, notou-se que para determinados problemas só os agentes de sequenciamentos melhoram a solução. O *TAH* supera em tempo computacional à heurística *IG+HR*, sendo que a diferença que existe entre os tempos de execução desses algoritmos ao finalizar o processo de busca de uma solução, não são significativos. Enquanto a qualidade se refere, a heurística *IG+HR* na maioria dos casos de instâncias obtém melhores resultados que o *TAH*, existindo uma diferença estatística significativa em relação à qualidade da solução.

Situação análoga ao *TAH* acontece com o *IG+MDL*, as sequências de produção são constantemente melhoradas, mas decisões em relação ao dimensionamento são poucas vezes tomadas. A diferença entre o *TAH* e o *IG+MDL* encontra-se em que o *IG+MDL* poucas vezes faz uso da formulação matemática, isto reduz o tempo computacional consideravelmente. Por outro lado, note que o procedimento do *TAH* para melhorar uma solução é selecionar um agente e se o selecionado trata o sequenciamento de lotes, este seleciona só um período aleatoriamente, ou seja, o agente melhora

só uma sequência de produção em toda uma solução, enquanto o *IG+MDL* procura melhorar a sequência de lotes de produção em todos os períodos, o que melhora a solução significativamente. Mesmo que uma iteração do *TAH* fosse considerada como a execução do corpo principal do algoritmo (da linha 8 até a linha 16 do algoritmo *A-Teams*), o número máximo de iterações do *IG+MDL* seria ainda menor que o número máximo de iterações do *TAH*. Por causa das características e funcionalidades da heurística *IG+MDL* expostas anteriormente, supera significativamente tanto em tempo computacional como em qualidade da solução à heurística *TAH*.

A heurística *IG+F&O* da mesma forma que o *IG+MDL* obtém boas sequências de lotes de produção. Este procedimento trata o dimensionamento de lotes por meio da heurística *F&O*, a qual faz uso da formulação matemática, tornando o algoritmo pouco mais demorado que o algoritmo *IG+MDL*, porém, a diferença entre tempos de execução não é considerável. Vale destacar também que os tempos de execução obtidos pelo *IG+F&O* são, na maioria dos casos, menores que os tempos de execução obtidos pelo *TAH*. A qualidade da solução obtida pela *IG+F&O*, é a melhor em comparação com todos os algoritmos em todas as classes de instâncias, isto, foi comprovado estatisticamente na seção anterior.

# Capítulo 7

# Conclusões

Neste trabalho foi estudado o problema de dimensionamento e sequenciamento de lotes em um ambiente de produção do tipo flowshop *(PDSLFS)*. O horizonte de planejamento considerado é finito e dividido em períodos. Para cada período existe uma demanda determinística e deve ser atendida sem atrasos. As máquinas são restritas em capacidade e as preparações destas inclusa entre períodos são dependentes da sequência de produção. O objetivo do problema é determinar uma programação da produção eficiente que satisfaça os clientes da maneira mais econômica possível, tentando minimizar os custos totais de produção, os quais são representados pelos custos de processamento, custos de estocagem dos produtos acabados e custos de preparação das máquinas.

Na literatura se encontrou o modelo matemático para o *PDSLFS* proposto por Ramezanian et al. [2012], que apresenta soluções com características permutacionais. O autor deste modelo afirma que tal característica é bem considerada tanto na prática industrial como na literatura. Dado que o modelo matemático só resolve problemas de pequeno porte, o mesmo autor propus a heurística Horizonte Rolante, com ela se consegue resolver problemas de porte mais elevados, porém, por ser baseada no modelo matemático requer um alto esforço computacional. Antes de Ramezanian et al. [2012], Belo Filho et al. [2012] propuseram o *A-Teams* para este mesmo problema. Este é um procedimento heurístico híbrido já que suas soluções são construídas baseadas no modelo matemático de Mohammadi et al. [2010b]. Vale ressaltar que o modelo matemático proposto por Ramezanian et al. [2012] é muito mais eficiente que o de Mohammadi et al. [2010b] e as heurísticas *HR* e *A-Teams* não foram anteriormente comparadas.

Neste trabalho, para tratar o *PDSLFS* são propostas heurísticas baseadas no algoritmo *IG*. Primeiramente o *IG* foi combinado com a *HR*, resultando o híbrido *IG+HR*. A combinação destes métodos (*IG+HR*) acelerou o processo de busca de solução em aproximadamente um 60% em relação à heurística *HR*. Embora a qualidade da solução do *HR* seja piorada em um 15% aproximadamente, é recomendável usar este método para instâncias de pequeno e médio porte, a perda em qualidade esta compensada com o pouco tempo gasto usado pelo procedimento para achar uma boa solução. A *IG+HR* ainda é fortemente dependente de muitas variáveis a serem otimizadas na formulação matemática, desta forma, para instâncias de grande porte o esforço computacional é demasiado.

Dado que instâncias de grande porte são difíceis de resolvê-las com a heurística *IG+HR* em tempo computacional razoável, foi necessário propor outra heurística, desta vez, o *IG* foi combinado com métodos totalmente aproximados para melhorar o dimensionamento de lotes *(MDL)*, a qual é chamada *IG+MDL*. Nesta heurística *(IG+MDL)* notou-se que o *IG* no processo de busca heurístico melhorava a sequência de produção consideravelmente, enquanto, o *MDL* faz pouca contribuição às decisões de dimensionamento de lotes, porém, as soluções obtidas pelo *IG+MDL* são de qualidade aceitável dado o pouco tempo computacional gasto para encontrar uma solução. A pouca utilização de métodos exatos no algoritmo *IG+MDL* o faz 69% aproximadamente mais rápido que o segundo algoritmo em gastar menos tempo computacional em encontrar uma boa solução *(IG+F&O)*.

Com a implementação da heurística *IG+F&O*, se buscou reduzir o tempo computacional gasto em relação à heurística *IG+HR* (mantendo a eficácia) e melhorar a qualidade da solução obtida pela heurística *IG+MDL* (mantendo a eficiência). Esses dois objetivos foram conseguidos, já que a heurística *IG+F&O* encontra melhores soluções que todos os algoritmos testados neste trabalho em todas as classes de instâncias, desta forma, enquanto a qualidade se refere a heurística *IG+F&O* é 100% melhor em todos os casos de problemas. Enquanto a tempo computacional, só o *IG+MDL* é mais rápido, más os tempos computacionais gastos pelo *IG+F&O* em todos os casos de instâncias são razoável e aceitável.

Em pesquisas futuras, recomenda-se tentar acelerar o processo de busca do *IG+F&O*. Também, propor novas estratégias baseadas em métodos aproximados que melhorem o dimensionamento de lotes, dado que as propostas por Gopalakrishnan et al. [2001] são as únicas utilizadas na literatura e estas apresentam pouca contribuição. Em relação ao problema, é interessante introduzir novas características ou restrições como: prioridades a certos produtos, diminuição do tempo ocioso, bloqueios de máquinas em

instantes de tempos, dentre outras, essas características tornam o problema mais interessante. Além disso, recomenda-se a consideração de outros ambientes de produção como o jobshop, flowshop flexível e jobshop flexível.

# Referências Bibliográficas


Allahverdi, A.; Gupta, J. & Aldowaisan, T. (1999). A review of scheduling research involving setup considerations. *Omega*, 27(2):219--239.

Antunes, J. (2008). *Sistemas de produção: conceitos e práticas para projetos e gestão da produção enxuta*. Bookman.

Bahl, H. C.; Ritzman, L. P. & Gupta, J. N. (1987). Or practice determining lot sizes and resource requirements: A review. *Operations Research*, 35(3):329--345.

Belo Filho, M.; Santos, M. O. & Meneses, C. N. (2012). Asynchronous teams for joint lot-sizing and scheduling problem in flow shops. *International Journal of Production Research*, 50(20):5809--5822.

Belo Filho, M. A. F.; dos Santos, M. O. & de Meneses, C. N. (2009). Programação de produção e dimensionamento de lotes para flowshop.

Billington, P.; Blackburn, J.; Maes, J.; Millen, R. & Van Wassenhove, L. N. (1994). Multi-item lotsizing in capacitated multi-stage serial systems. *IIE transactions*, 26(2):12--18.

Cheng, T. E.; Gupta, J. N. & Wang, G. (2000). A review of flowshop scheduling research with setup times. *Production and Operations Management*, 9(3):262--282.

Corrêa, H. L.; Gianesi, I. G. & Caon, M. (2001). Planejamento, programação e controle da produção. *São Paulo: Atlas*, 1.

da Rocha, D. R. (2008). *Gestão da produção e operações*. Ciência Moderna.

de Araujo, S. A.; Arenales, M. N. & Clark, A. R. (2007). Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *Journal of Heuristics*, 13(4):337--358.

de Souza, P. S. & Talukdar, S. N. (1993). Asynchronous organizations for multi-algorithm problems. Em *Proceedings of the 1993 ACM/SIGAPP symposium on Applied computing: states of the art and practice*, pp. 286--293. ACM.

Drexl, A. & Kimms, A. (1997). Lot sizing and scheduling?survey and extensions. *European Journal of Operational Research*, 99(2):221--235.

Fatih Tasgetiren, M.; Pan, Q.-K.; Suganthan, P. N. & Buyukdagli, O. (2013). A variable iterated greedy algorithm with differential evolution for the no-idle permutation flowshop scheduling problem. *Computers & Operations Research*, 40(7):1729--1743.

Florian, M.; Lenstra, J. K. & Rinnooy Kan, A. (1980). Deterministic production planning: Algorithms and complexity. *Management science*, 26(7):669--679.

Furlan, M. M. & dos Santos, M. O. (2011). *Uma abordagem híbrida de algoritmo de abelhas e x-and-optimize para o problema de dimensionamento de lotes multiestágio com limitação de capacidade e preservação da preparação*. Bookman.

Garey, M. R.; Johnson, D. S. & Sethi, R. (1976). The complexity of flowshop and jobshop scheduling. *Mathematics of operations research*, 1(2):117--129.

Gopalakrishnan, M.; Ding, K.; Bourjolly, J.-M. & Mohan, S. (2001). A tabu-search heuristic for the capacitated lot-sizing problem with set-up carryover. *Management Science*, 47(6):851--863.

Gupta, J. N. & Stafford Jr, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169(3):699--711.

Hansen, P.; Mladenović, N. & Pérez, J. A. M. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, 175(1):367--407.

Jacobs, L. W. & Brusco, M. J. (1995). Note: A local-search heuristic for large set-covering problems. *Naval Research Logistics (NRL)*, 42(7):1129--1140.

Johnson, S. M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1(1):61--68. ISSN 1931-9193.

Karimi, B.; Fatemi Ghomi, S. & Wilson, J. (2003). The capacitated lot sizing problem: a review of models and algorithms. *Omega*, 31(5):365--378.

Kuik, R.; Salomon, M. & Van Wassenhove, L. N. (1994). Batching decisions: structure and models. *European Journal of Operational Research*, 75(2):243--263.

Lin, S.-W.; Ying, K.-C. & Huang, C.-Y. (2013). Minimising makespan in distributed permutation flowshops using a modified iterated greedy algorithm. *International Journal of Production Research*, 51(16):5029--5038.

Lustosa, L. J.; de Mesquita, M. A. & OLIVEIRA, R. J. (2008). *Planejamento e controle da produção*. Elsevier Brasil.

Marchiori, E. & Steenbeek, A. (2000). An evolutionary algorithm for large scale set covering problems with application to airline crew scheduling. Em *Real-World Applications of Evolutionary Computing*, pp. 370--384. Springer.

Mascia, F.; López-Ibáñez, M.; Dubois-Lacoste, J. & Stützle, T. (2013). From grammars to parameters: Automatic iterated greedy design for the permutation flow-shop problem with weighted tardiness. Em *Learning and Intelligent Optimization*, pp. 321--334. Springer.

Mercé, C. & Fontan, G. (2003). Mip-based heuristics for capacitated lotsizing problems. *International Journal of Production Economics*, 85(1):97 – 111. ISSN 0925-5273. <ce:title>Planning and Control of Productive Systems</ce:title>.

Mohammadi, M.; Fatemi Ghomi, S.; Karimi, B. & Torabi, S. (2010a). Mip-based heuristics for lotsizing in capacitated pure flow shop with sequence-dependent setups. *International Journal of Production Research*, 48(10):2957--2973.

Mohammadi, M.; Ghomi, S. F.; Karimi, B. & Torabi, S. (2010b). Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups. *Journal of Intelligent Manufacturing*, 21(4):501--510.

Mohammadi, M.; Torabi, S.; Ghomi, S. F. & Karimi, B. (2010c). A new algorithmic approach for capacitated lot-sizing problem in flow shops with sequence-dependent setups. *The International Journal of Advanced Manufacturing Technology*, 49(1-4):201--211.

Montgomery, D. (2001). Analysis of experiments.

Nawaz, M.; Enscore, E. E. & Ham, I. (1983). A heuristic algorithm for the< i> m</i>-machine,< i> n</i>-job flow-shop sequencing problem. *Omega*, 11(1):91--95.

Pan, Q.-K. & Ruiz, R. (2013). A comprehensive review and evaluation of permutation flowshop heuristics to minimize flowtime. *Computers & Operations Research*, 40(1):117--128.

Pinedo, M. (2008). *Scheduling: theory, algorithms, and systems.* Springer Science+ Business Media.

Pochet, Y. & Wolsey, L. A. (2006). *Production planning by mixed integer programming.* Springer Verlag.

Ponnambalam, S. & Reddy, M. (2003). A ga-sa multiobjective hybrid search algorithm for integrating lot sizing and sequencing in flow-line scheduling. *The International Journal of Advanced Manufacturing Technology*, 21(2):126--137.

Ramezanian, R. & Saidi-Mehrabad, M. (2013). Hybrid simulated annealing and mip-based heuristics for stochastic lot-sizing and scheduling problem in capacitated multi-stage production system. *Applied Mathematical Modelling*, 37(7):5134 – 5147. ISSN 0307-904X.

Ramezanian, R.; Saidi-Mehrabad, M. & Teimoury, E. (2012). A mathematical model for integrating lot-sizing and scheduling problem in capacitated flow shop environments. *The International Journal of Advanced Manufacturing Technology*, pp. 1--15.

Ruiz, R. & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, 177(3):2033--2049.

Ruiz, R. & Stützle, T. (2008). An iterated greedy heuristic for the sequence dependent setup times flowshop problem with makespan and weighted tardiness objectives. *European Journal of Operational Research*, 187(3):1143 – 1159. ISSN 0377-2217.

Sahling, F.; Buschkühl, L.; Tempelmeier, H. & Helber, S. (2009). Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic. *Computers & Operations Research*, 36(9):2546--2553.

Seeanner, F.; Almada-Lobo, B. & Meyr, H. (2013). Combining the principles of variable neighborhood decomposition search and the fix amp;optimize heuristic to solve multi-level lotsizing and scheduling problems. *Computers & Operations Research*, 40(1):303 – 317. ISSN 0305-0548.

Shim, I.-S.; Kim, H.-C.; Doh, H.-H. & Lee, D.-H. (2011). A two-stage heuristic for single machine capacitated lot-sizing and scheduling with sequence-dependent setup costs. *Computers & Industrial Engineering*, 61(4):920 – 929. ISSN 0360-8352.

Sikora, R. (1996). A genetic algorithm for integrating lot-sizing and sequencing in scheduling a capacitated flow line. *Computers & Industrial Engineering*, 30(4):969--981.

Sikora, R.; Chhajed, D. & Shaw, M. J. (1996). Integrating the lot-sizing and sequencing decisions for scheduling a capacitated flow line. *Computers & Industrial Engineering*, 30(4):659 – 679. ISSN 0360-8352.

Silver, E. A. & Meal, H. (1973). A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Production and Inventory Management*, 14(2):64--74.

Slack, N.; Chambers, S. & Johnston, R. (2009). *Administração da produção*. Atlas.

Smith-Daniels, V. & Ritzman, L. P. (1988). A model for lot sizing and sequencing in process industries. *The International Journal Of Production Research*, 26(4):647--674.

Stadtler Hartmut, K. C. (2008). Supply chain management and advanced planning: Concepts,models software and case studies. *Springer-Verlag*, 9(4).

Staggemeier, A. T. & Clark, A. R. (2001). A survey of lot-sizing and scheduling models. Em *23rd Annual Symposium of the Brazilian Operational Research Society (SOBRAPO)*, pp. 938--947. Citeseer.

Toledo, C. F.; França, P. M.; Morabito, R. & Kimms, A. (2007). Um modelo de otimização para o problema integrado de dimensionamento de lotes e programação da produção em fábricas de refrigerantes. *Pesquisa Operacional*, 27(1):155--186.

Toso, E. A. V. & Morabito, R. (2005). Otimização no dimensionamento e sequenciamento de lotes de produção: estudo de caso numa fábrica de rações. *Gestão & Produção*, 12(2):203--217.

Trigeiro, W. W.; Thomas, L. J. & McClain, J. O. (1989). Capacitated lot sizing with setup times. *Management science*, 35(3):353--366.

Tubino, D. F. (2009). *Planejamento e controle da produção: teoria e prática*. Atlas.

Urrutia, E. D. G.; Aggoune, R.; Dauzère-Pérès, S. et al. (2012). New integrated approach for solving multi-level lot sizing and scheduling problems. *Proceedings of MOSIM'12*.

Vallada, E. & Ruiz, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research*, 211(3):612--622.

Vasko, F. J. (1984). An efficient heuristic for large set covering problems. *Naval Research Logistics Quarterly*, 31(1):163--171.

Zaccarelli, S. B. (1967). *Programação e controle da produção*. Livraria Pioneira Editôra.

Zhu, X. & Wilhelm, W. E. (2006). Scheduling and lot sizing with sequence-dependent setup: A literature review. *IIE transactions*, 38(11):987--1007.

# Apêndice A

Instância utiliza nos procedimentos: construção da solução inicial, procedimento de destruição-construção e na busca local dos algoritmos *IG+RH, IG+MDL, e IG+F&O*.

Tabela A.1: Tempo de Produção ($b$) do produto $J$ na máquina $M$

| $b_{jm}$ | $M_1$ | $M_2$ | $M_3$ |
|---|---|---|---|
| $J_1$ | 1.8 | 2 | 1.9 |
| $J_2$ | 1.6 | 1.7 | 1.6 |
| $J_3$ | 1.9 | 1.6 | 1.5 |
| $J_4$ | 1.5 | 1.7 | 1.9 |
| $J_5$ | 2.0 | 1.7 | 2.0 |

Tabela A.2: Capacidade ($C$) da máquina $M$ no Período $T$

| $C_{mt}$ | $T_1$ | $T_2$ | $T_3$ |
|---|---|---|---|
| $M_1$ | 1977 | 2020 | 1990 |
| $M_2$ | 2061 | 1925 | 2075 |
| $M_3$ | 2006 | 1988 | 2074 |

Tabela A.3: Demanda ($d$) do produto $J$ no período $T$

| $d_{jt}$ | $T_1$ | $T_2$ | $T_3$ |
|---|---|---|---|
| $J_1$ | 127 | 111 | 143 |
| $J_2$ | 172 | 114 | 100 |
| $J_3$ | 194 | 125 | 103 |
| $J_4$ | 120 | 99 | 146 |
| $J_5$ | 88 | 66 | 119 |

Tabela A.4: Custo de estoque ($h$) do produto $J$ na máquina $M$

| $h_{jm}$ | $M_1$ | $M_2$ | $M_3$ |
|---|---|---|---|
| $J_1$ | 0.3 | 0.3 | 0.4 |
| $J_2$ | 0.2 | 0.4 | 0.4 |
| $J_3$ | 0.2 | 0.3 | 0.2 |
| $J_4$ | 0.3 | 0.4 | 0.4 |
| $J_5$ | 0.4 | 0.2 | 0.4 |

Tabela A.5: Custo de produção ($p$) do produto $j$ na máquina $m$ no período $t$

| | $T_1$ | | | | $T_2$ | | | | $T_3$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | $M_1$ | $M_2$ | $M_3$ | | $M_1$ | $M_2$ | $M_3$ | | $M_1$ | $M_2$ | $M_3$ |
| $J_1$ | 1.5 | 1.8 | 1.9 | $J_1$ | 1.9 | 1.5 | 2.0 | $J_1$ | 1.8 | 1.8 | 2.0 |
| $J_2$ | 1.7 | 2.0 | 2.0 | $J_2$ | 1.9 | 1.9 | 1.7 | $J_2$ | 1.6 | 2.0 | 1.8 |
| $J_3$ | 1.9 | 1.7 | 1.8 | $J_3$ | 1.5 | 1.7 | 1.5 | $J_3$ | 1.9 | 1.6 | 2.0 |
| $J_4$ | 1.5 | 1.9 | 1.7 | $J_4$ | 1.6 | 1.8 | 2.0 | $J_4$ | 1.6 | 1.6 | 1.6 |
| $J_5$ | 1.8 | 2.0 | 1.8 | $J_5$ | 1.9 | 1.8 | 1.9 | $J_5$ | 1.8 | 1.8 | 1.5 |

Tabela A.6: Tempo de preparação ($S$) do produto $i$ ao produto $j$ na máquina $m$

| | $M_1$ | | | | | | $M_2$ | | | | | | $M_3$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | |
| $i_1$ | 0 | 52 | 60 | 48 | 43 | $i_1$ | 0 | 44 | 61 | 61 | 46 | $i_1$ | 0 | 35 | 48 | 69 | 35 | |
| $i_2$ | 56 | 0 | 46 | 57 | 48 | $i_2$ | 41 | 0 | 50 | 57 | 36 | $i_2$ | 55 | 0 | 37 | 55 | 47 | |
| $i_3$ | 58 | 37 | 0 | 68 | 43 | $i_3$ | 58 | 46 | 0 | 53 | 39 | $i_3$ | 53 | 49 | 0 | 38 | 63 | |
| $i_4$ | 35 | 42 | 58 | 0 | 39 | $i_4$ | 36 | 66 | 47 | 0 | 57 | $i_4$ | 55 | 64 | 54 | 0 | 61 | |
| $i_5$ | 63 | 35 | 48 | 62 | 0 | $i_5$ | 57 | 69 | 51 | 61 | 0 | $i_5$ | 60 | 56 | 63 | 47 | 0 | |

Tabela A.7: Custo de preparação ($W$) do produto $i$ ao produto $j$ na máquina $m$

| | $M_1$ | | | | | | $M_2$ | | | | | | $M_3$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ | |
| $i_1$ | 0 | 42 | 21 | 25 | 13 | $i_1$ | 0 | 42 | 41 | 51 | 26 | $i_1$ | 0 | 25 | 47 | 29 | 15 | |
| $i_2$ | 26 | 0 | 42 | 52 | 58 | $i_2$ | 21 | 0 | 10 | 47 | 26 | $i_2$ | 45 | 0 | 32 | 51 | 37 | |
| $i_3$ | 38 | 17 | 0 | 64 | 41 | $i_3$ | 58 | 26 | 0 | 23 | 19 | $i_3$ | 23 | 29 | 0 | 34 | 13 | |
| $i_4$ | 25 | 22 | 55 | 0 | 37 | $i_4$ | 32 | 56 | 27 | 0 | 27 | $i_4$ | 45 | 54 | 51 | 0 | 61 | |
| $i_5$ | 13 | 25 | 47 | 61 | 0 | $i_5$ | 54 | 49 | 41 | 61 | 0 | $i_5$ | 10 | 26 | 23 | 41 | 0 | |

# Apêndice B

Tabelas correspondentes às condições experimentais encontradas pelo desenho experimental fatorial multinível para cada um dos algoritmos.

Tabela B.1: Condições experimentais para o *TAH*

| **Condição Exp.** | $T\_Mc$ | $Iter\_Ag$ | $Iter\_Sm$ | $Iter\_Alg$ |
|---|---|---|---|---|
| $CEX\_1$ | 13 | 4 | 40 | 1,00E+005 |
| $CEX\_2$ | 5 | 7 | 40 | 1,00E+005 |
| $CEX\_3$ | 13 | 9 | 60 | 5,00E+004 |
| $CEX\_4$ | 5 | 9 | 40 | 5,00E+004 |
| $CEX\_5$ | 20 | 4 | 60 | 1,00E+005 |
| $CEX\_6$ | 20 | 4 | 60 | 5,00E+004 |
| $CEX\_7$ | 20 | 7 | 20 | 5,00E+004 |
| $CEX\_8$ | 20 | 4 | 40 | 7,50E+004 |
| $CEX\_9$ | 5 | 7 | 60 | 1,00E+005 |
| $CEX\_10$ | 20 | 4 | 20 | 1,00E+005 |
| $CEX\_11$ | 20 | 7 | 20 | 1,00E+005 |
| $CEX\_12$ | 5 | 7 | 20 | 7,50E+004 |
| $CEX\_13$ | 20 | 4 | 20 | 5,00E+004 |
| $CEX\_14$ | 13 | 7 | 60 | 1,00E+005 |
| $CEX\_15$ | 13 | 4 | 20 | 7,50E+004 |
| $CEX\_16$ | 13 | 9 | 20 | 5,00E+004 |
| $CEX\_17$ | 20 | 9 | 20 | 7,50E+004 |
| $CEX\_18$ | 13 | 9 | 60 | 1,00E+005 |
| $CEX\_19$ | 13 | 9 | 20 | 1,00E+005 |
| $CEX\_20$ | 5 | 4 | 40 | 7,50E+004 |
| $CEX\_21$ | 13 | 7 | 20 | 5,00E+004 |

*Continua na próxima página*

Tabela B.1 – *Continuação da página anterior*

| **Condição Exp.** | *T_Mc* | *Iter_Ag* | *Iter_Sm* | *Iter_Alg* |
|---|---|---|---|---|
| *CEX_22* | 5 | 9 | 40 | 7,50E+004 |
| *CEX_23* | 20 | 7 | 40 | 7,50E+004 |
| *CEX_24* | 13 | 9 | 60 | 7,50E+004 |
| *CEX_25* | 20 | 4 | 40 | 1,00E+005 |
| *CEX_26* | 20 | 9 | 60 | 1,00E+005 |
| *CEX_27* | 5 | 9 | 60 | 1,00E+005 |
| *CEX_28* | 13 | 4 | 20 | 5,00E+004 |
| *CEX_29* | 13 | 9 | 40 | 1,00E+005 |
| *CEX_30* | 13 | 7 | 20 | 1,00E+005 |
| *CEX_31* | 5 | 7 | 40 | 5,00E+004 |
| *CEX_32* | 13 | 4 | 60 | 1,00E+005 |
| *CEX_33* | 13 | 9 | 40 | 5,00E+004 |
| *CEX_34* | 13 | 9 | 20 | 7,50E+004 |
| *CEX_35* | 5 | 7 | 20 | 5,00E+004 |
| *CEX_36* | 5 | 4 | 60 | 5,00E+004 |
| *CEX_37* | 5 | 9 | 60 | 5,00E+004 |
| *CEX_38* | 5 | 7 | 20 | 1,00E+005 |
| *CEX_39* | 20 | 4 | 60 | 7,50E+004 |
| *CEX_40* | 5 | 4 | 40 | 5,00E+004 |
| *CEX_41* | 13 | 7 | 40 | 5,00E+004 |
| *CEX_42* | 5 | 4 | 20 | 1,00E+005 |
| *CEX_43* | 5 | 9 | 60 | 7,50E+004 |
| *CEX_44* | 20 | 9 | 20 | 5,00E+004 |
| *CEX_45* | 13 | 4 | 40 | 7,50E+004 |
| *CEX_46* | 5 | 4 | 40 | 1,00E+005 |
| *CEX_47* | 5 | 4 | 20 | 7,50E+004 |
| *CEX_48* | 5 | 9 | 20 | 1,00E+005 |
| *CEX_49* | 20 | 7 | 60 | 5,00E+004 |
| *CEX_50* | 5 | 9 | 40 | 1,00E+005 |
| *CEX_51* | 20 | 9 | 40 | 1,00E+005 |
| *CEX_52* | 20 | 7 | 40 | 5,00E+004 |
| *CEX_53* | 13 | 4 | 20 | 1,00E+005 |
| *CEX_54* | 5 | 7 | 40 | 7,50E+004 |
| *CEX_55* | 20 | 4 | 40 | 5,00E+004 |

*Continua na próxima página*

Tabela B.1 – *Continuação da página anterior*

| **Condição Exp.** | $T\_Mc$ | $Iter\_Ag$ | $Iter\_Sm$ | $Iter\_Alg$ |
|---|---|---|---|---|
| $CEX\_56$ | 13 | 7 | 60 | 5,00E+004 |
| $CEX\_57$ | 20 | 7 | 20 | 7,50E+004 |
| $CEX\_58$ | 5 | 4 | 20 | 5,00E+004 |
| $CEX\_59$ | 5 | 9 | 20 | 5,00E+004 |
| $CEX\_60$ | 13 | 7 | 40 | 1,00E+005 |
| $CEX\_61$ | 13 | 4 | 60 | 7,50E+004 |
| $CEX\_62$ | 5 | 7 | 60 | 5,00E+004 |
| $CEX\_63$ | 5 | 9 | 20 | 7,50E+004 |
| $CEX\_64$ | 13 | 4 | 60 | 5,00E+004 |
| $CEX\_65$ | 20 | 9 | 60 | 5,00E+004 |
| $CEX\_66$ | 13 | 9 | 40 | 7,50E+004 |
| $CEX\_67$ | 20 | 9 | 40 | 5,00E+004 |
| $CEX\_68$ | 20 | 7 | 60 | 7,50E+004 |
| $CEX\_69$ | 20 | 7 | 60 | 1,00E+005 |
| $CEX\_70$ | 20 | 9 | 20 | 1,00E+005 |
| $CEX\_71$ | 13 | 4 | 40 | 5,00E+004 |
| $CEX\_72$ | 20 | 4 | 20 | 7,50E+004 |
| $CEX\_73$ | 5 | 4 | 60 | 7,50E+004 |
| $CEX\_74$ | 13 | 7 | 20 | 7,50E+004 |
| $CEX\_75$ | 13 | 7 | 60 | 7,50E+004 |
| $CEX\_76$ | 5 | 7 | 60 | 7,50E+004 |
| $CEX\_77$ | 20 | 9 | 60 | 7,50E+004 |
| $CEX\_78$ | 20 | 9 | 40 | 7,50E+004 |
| $CEX\_79$ | 5 | 4 | 60 | 1,00E+005 |
| $CEX\_80$ | 13 | 7 | 40 | 7,50E+004 |
| $CEX\_81$ | 20 | 7 | 40 | 1,00E+005 |

Tabela B.2: Condições experimentais para o *IG+HR*

| $Dest-Cons\ (d)$ | $Iter\_Bl$ | $Iter\_Alg$ |
|---|---|---|
| $0,3N$ | 8 | 50 |
| $0,7N$ | 8 | 80 |
| $0,3N$ | 6 | 20 |
| $0,3N$ | 4 | 80 |

*Continua na próxima página*

Tabela B.2 – *Continuação da página anterior*

| $Dest-Cons\ (d)$ | $Iter\_Bl$ | $Iter\_Alg$ |
|---|---|---|
| $0,3N$ | 4 | 20 |
| $0,5N$ | 8 | 20 |
| $0,3N$ | 8 | 20 |
| $0,7N$ | 4 | 50 |
| $0,5N$ | 4 | 80 |
| $0,3N$ | 4 | 50 |
| $0,7N$ | 6 | 20 |
| $0,7N$ | 4 | 20 |
| $0,7N$ | 8 | 50 |
| $0,7N$ | 6 | 80 |
| $0,5N$ | 4 | 20 |
| $0,7N$ | 4 | 80 |
| $0,5N$ | 4 | 50 |
| $0,5N$ | 6 | 20 |
| $0,5N$ | 6 | 80 |
| $0,5N$ | 6 | 50 |
| $0,3N$ | 8 | 80 |
| $0,7N$ | 8 | 20 |
| $0,3N$ | 6 | 50 |
| $0,3N$ | 6 | 80 |
| $0,7N$ | 6 | 50 |
| $0,5N$ | 8 | 50 |
| $0,5N$ | 8 | 80 |

Tabela B.3: Condições experimentais para o *IG+MDL*

| **Condição Exp.** | $d$ | $Iter\_Bl$ | $Iter\_MDL$ | $Iter\_Alg$ |
|---|---|---|---|---|
| $CEX\_1$ | 0,6 | 7 | 6 | 200 |
| $CEX\_2$ | 0,2 | 7 | 6 | 100 |
| $CEX\_3$ | 0,2 | 3 | 6 | 200 |
| $CEX\_4$ | 0,6 | 5 | 8 | 100 |
| $CEX\_5$ | 0,6 | 7 | 8 | 150 |
| $CEX\_6$ | 0,6 | 3 | 6 | 200 |
| $CEX\_7$ | 0,6 | 5 | 8 | 150 |

*Continua na próxima página*

Tabela B.3 – *Continuação da página anterior*

| **Condição Exp.** | $d$ | $Iter\_Bl$ | $Iter\_MDL$ | $Iter\_Alg$ |
|---|---|---|---|---|
| $CEX\_8$ | 0,4 | 7 | 6 | 150 |
| $CEX\_9$ | 0,6 | 3 | 8 | 150 |
| $CEX\_10$ | 0,4 | 3 | 6 | 200 |
| $CEX\_11$ | 0,2 | 5 | 6 | 150 |
| $CEX\_12$ | 0,4 | 5 | 6 | 200 |
| $CEX\_13$ | 0,2 | 5 | 8 | 150 |
| $CEX\_14$ | 0,4 | 5 | 8 | 200 |
| $CEX\_15$ | 0,4 | 5 | 4 | 150 |
| $CEX\_16$ | 0,2 | 7 | 4 | 200 |
| $CEX\_17$ | 0,4 | 7 | 6 | 100 |
| $CEX\_18$ | 0,4 | 5 | 8 | 150 |
| $CEX\_19$ | 0,2 | 5 | 4 | 200 |
| $CEX\_20$ | 0,6 | 3 | 4 | 200 |
| $CEX\_21$ | 0,2 | 7 | 4 | 150 |
| $CEX\_22$ | 0,4 | 3 | 4 | 200 |
| $CEX\_23$ | 0,4 | 3 | 6 | 150 |
| $CEX\_24$ | 0,4 | 7 | 8 | 150 |
| $CEX\_25$ | 0,4 | 7 | 4 | 100 |
| $CEX\_26$ | 0,2 | 5 | 6 | 200 |
| $CEX\_27$ | 0,2 | 5 | 8 | 100 |
| $CEX\_28$ | 0,6 | 7 | 8 | 200 |
| $CEX\_29$ | 0,2 | 3 | 8 | 150 |
| $CEX\_30$ | 0,2 | 7 | 6 | 150 |
| $CEX\_31$ | 0,4 | 5 | 8 | 100 |
| $CEX\_32$ | 0,4 | 3 | 6 | 100 |
| $CEX\_33$ | 0,2 | 7 | 4 | 100 |
| $CEX\_34$ | 0,6 | 3 | 4 | 150 |
| $CEX\_35$ | 0,4 | 5 | 4 | 200 |
| $CEX\_36$ | 0,6 | 7 | 4 | 200 |
| $CEX\_37$ | 0,4 | 7 | 4 | 200 |
| $CEX\_38$ | 0,2 | 3 | 4 | 100 |
| $CEX\_39$ | 0,2 | 3 | 6 | 100 |
| $CEX\_40$ | 0,2 | 5 | 4 | 150 |
| $CEX\_41$ | 0,4 | 3 | 4 | 150 |

*Continua na próxima página*

Tabela B.3 – *Continuação da página anterior*

| **Condição Exp.** | $d$ | $Iter\_Bl$ | $Iter\_MDL$ | $Iter\_Alg$ |
|---|---|---|---|---|
| $CEX\_42$ | 0,6 | 3 | 8 | 200 |
| $CEX\_43$ | 0,6 | 3 | 4 | 100 |
| $CEX\_44$ | 0,6 | 7 | 6 | 100 |
| $CEX\_45$ | 0,4 | 5 | 4 | 100 |
| $CEX\_46$ | 0,2 | 7 | 8 | 100 |
| $CEX\_47$ | 0,6 | 3 | 6 | 100 |
| $CEX\_48$ | 0,2 | 3 | 4 | 150 |
| $CEX\_49$ | 0,4 | 3 | 8 | 150 |
| $CEX\_50$ | 0,6 | 5 | 6 | 100 |
| $CEX\_51$ | 0,4 | 7 | 8 | 200 |
| $CEX\_52$ | 0,6 | 7 | 4 | 150 |
| $CEX\_53$ | 0,2 | 3 | 8 | 200 |
| $CEX\_54$ | 0,6 | 5 | 4 | 100 |
| $CEX\_55$ | 0,6 | 7 | 4 | 100 |
| $CEX\_56$ | 0,6 | 7 | 6 | 150 |
| $CEX\_57$ | 0,2 | 7 | 8 | 200 |
| $CEX\_58$ | 0,6 | 5 | 8 | 200 |
| $CEX\_59$ | 0,6 | 5 | 4 | 200 |
| $CEX\_60$ | 0,2 | 5 | 6 | 100 |
| $CEX\_61$ | 0,6 | 7 | 8 | 100 |
| $CEX\_62$ | 0,2 | 3 | 4 | 200 |
| $CEX\_63$ | 0,4 | 3 | 4 | 100 |
| $CEX\_64$ | 0,4 | 3 | 8 | 200 |
| $CEX\_65$ | 0,2 | 5 | 8 | 200 |
| $CEX\_66$ | 0,4 | 5 | 6 | 150 |
| $CEX\_67$ | 0,6 | 5 | 4 | 150 |
| $CEX\_68$ | 0,4 | 7 | 4 | 150 |
| $CEX\_69$ | 0,6 | 3 | 8 | 100 |
| $CEX\_70$ | 0,2 | 5 | 4 | 100 |
| $CEX\_71$ | 0,4 | 7 | 6 | 200 |
| $CEX\_72$ | 0,2 | 7 | 8 | 150 |
| $CEX\_73$ | 0,6 | 5 | 6 | 200 |
| $CEX\_74$ | 0,2 | 7 | 6 | 200 |
| $CEX\_75$ | 0,4 | 5 | 6 | 100 |

*Continua na próxima página*

Tabela B.3 – *Continuação da página anterior*

| **Condição Exp.** | *d* | *Iter_Bl* | *Iter_MDL* | *Iter_Alg* |
|---|---|---|---|---|
| *CEX_76* | 0,2 | 3 | 8 | 100 |
| *CEX_77* | 0,6 | 5 | 6 | 150 |
| *CEX_78* | 0,2 | 3 | 6 | 150 |
| *CEX_79* | 0,6 | 3 | 6 | 150 |
| *CEX_80* | 0,4 | 3 | 8 | 100 |
| *CEX_81* | 0,4 | 7 | 8 | 100 |

Tabela B.4: Condições experimentais para o *IG+F&O*

| **Condição Exp.** | *d* | *Iter_Bl* | *n_Sub* | *Iter_Alg* |
|---|---|---|---|---|
| *CEX_1* | 0,4 | 5 | 0,4 | 150 |
| *CEX_2* | 0,4 | 3 | 0,3 | 200 |
| *CEX_3* | 0,2 | 7 | 0,3 | 200 |
| *CEX_4* | 0,4 | 5 | 0,4 | 100 |
| *CEX_5* | 0,3 | 5 | 0,2 | 150 |
| *CEX_6* | 0,3 | 3 | 0,2 | 150 |
| *CEX_7* | 0,4 | 3 | 0,2 | 200 |
| *CEX_8* | 0,3 | 5 | 0,4 | 200 |
| *CEX_9* | 0,2 | 5 | 0,4 | 200 |
| *CEX_10* | 0,4 | 5 | 0,2 | 100 |
| *CEX_11* | 0,4 | 3 | 0,2 | 100 |
| *CEX_12* | 0,3 | 7 | 0,3 | 150 |
| *CEX_13* | 0,3 | 5 | 0,3 | 200 |
| *CEX_14* | 0,4 | 5 | 0,3 | 150 |
| *CEX_15* | 0,2 | 7 | 0,4 | 200 |
| *CEX_16* | 0,3 | 5 | 0,2 | 100 |
| *CEX_17* | 0,2 | 3 | 0,3 | 150 |
| *CEX_18* | 0,2 | 7 | 0,4 | 150 |
| *CEX_19* | 0,2 | 3 | 0,4 | 200 |
| *CEX_20* | 0,4 | 3 | 0,3 | 150 |
| *CEX_21* | 0,2 | 5 | 0,2 | 200 |
| *CEX_22* | 0,2 | 5 | 0,3 | 150 |
| *CEX_23* | 0,2 | 7 | 0,4 | 100 |
| *CEX_24* | 0,3 | 3 | 0,3 | 150 |

*Continua na próxima página*

Tabela B.4 – *Continuação da página anterior*

| **Condição Exp.** | *d* | *Iter_Bl* | *n_Sub* | *Iter_Alg* |
|---|---|---|---|---|
| *CEX_25* | 0,4 | 7 | 0,4 | 200 |
| *CEX_26* | 0,2 | 3 | 0,3 | 200 |
| *CEX_27* | 0,2 | 7 | 0,2 | 100 |
| *CEX_28* | 0,3 | 5 | 0,4 | 150 |
| *CEX_29* | 0,2 | 5 | 0,2 | 100 |
| *CEX_30* | 0,4 | 5 | 0,3 | 100 |
| *CEX_31* | 0,2 | 5 | 0,4 | 150 |
| *CEX_32* | 0,2 | 3 | 0,2 | 150 |
| *CEX_33* | 0,3 | 3 | 0,2 | 100 |
| *CEX_34* | 0,4 | 3 | 0,3 | 100 |
| *CEX_35* | 0,4 | 7 | 0,4 | 100 |
| *CEX_36* | 0,4 | 3 | 0,4 | 150 |
| *CEX_37* | 0,3 | 7 | 0,3 | 200 |
| *CEX_38* | 0,3 | 3 | 0,3 | 100 |
| *CEX_39* | 0,4 | 3 | 0,4 | 200 |
| *CEX_40* | 0,2 | 5 | 0,2 | 150 |
| *CEX_41* | 0,2 | 7 | 0,2 | 150 |
| *CEX_42* | 0,3 | 7 | 0,2 | 200 |
| *CEX_43* | 0,3 | 3 | 0,3 | 200 |
| *CEX_44* | 0,4 | 5 | 0,4 | 200 |
| *CEX_45* | 0,3 | 7 | 0,2 | 150 |
| *CEX_46* | 0,3 | 3 | 0,4 | 150 |
| *CEX_47* | 0,4 | 5 | 0,2 | 200 |
| *CEX_48* | 0,3 | 5 | 0,3 | 150 |
| *CEX_49* | 0,4 | 5 | 0,2 | 150 |
| *CEX_50* | 0,4 | 3 | 0,2 | 150 |
| *CEX_51* | 0,2 | 7 | 0,3 | 100 |
| *CEX_52* | 0,3 | 7 | 0,4 | 200 |
| *CEX_53* | 0,3 | 3 | 0,4 | 100 |
| *CEX_54* | 0,2 | 5 | 0,3 | 100 |
| *CEX_55* | 0,2 | 3 | 0,3 | 100 |
| *CEX_56* | 0,4 | 7 | 0,2 | 200 |
| *CEX_57* | 0,3 | 3 | 0,2 | 200 |
| *CEX_58* | 0,4 | 7 | 0,3 | 200 |

*Continua na próxima página*

Tabela B.4 – *Continuação da página anterior*

| **Condição Exp.** | $d$ | $Iter\_Bl$ | $n\_Sub$ | $Iter\_Alg$ |
|---|---|---|---|---|
| $CEX\_59$ | 0,4 | 5 | 0,3 | 200 |
| $CEX\_60$ | 0,2 | 3 | 0,2 | 200 |
| $CEX\_61$ | 0,3 | 7 | 0,3 | 100 |
| $CEX\_62$ | 0,2 | 5 | 0,4 | 100 |
| $CEX\_63$ | 0,2 | 5 | 0,3 | 200 |
| $CEX\_64$ | 0,2 | 7 | 0,2 | 200 |
| $CEX\_65$ | 0,4 | 7 | 0,3 | 150 |
| $CEX\_66$ | 0,2 | 3 | 0,2 | 100 |
| $CEX\_67$ | 0,2 | 3 | 0,4 | 150 |
| $CEX\_68$ | 0,4 | 7 | 0,4 | 150 |
| $CEX\_69$ | 0,3 | 3 | 0,4 | 200 |
| $CEX\_70$ | 0,3 | 7 | 0,2 | 100 |
| $CEX\_71$ | 0,3 | 7 | 0,4 | 150 |
| $CEX\_72$ | 0,3 | 5 | 0,3 | 100 |
| $CEX\_73$ | 0,2 | 3 | 0,4 | 100 |
| $CEX\_74$ | 0,4 | 7 | 0,2 | 150 |
| $CEX\_75$ | 0,3 | 7 | 0,4 | 100 |
| $CEX\_76$ | 0,3 | 5 | 0,2 | 200 |
| $CEX\_77$ | 0,2 | 7 | 0,3 | 150 |
| $CEX\_78$ | 0,4 | 3 | 0,4 | 100 |
| $CEX\_79$ | 0,3 | 5 | 0,4 | 100 |
| $CEX\_80$ | 0,4 | 7 | 0,2 | 100 |
| $CEX\_81$ | 0,4 | 7 | 0,3 | 100 |

# Apêndice C

Resultados completos obtidos pelo modelo matemático, os algoritmos da literatura e os algoritmos propostos. Na tabela mostra-se todas as classes de instâncias utilizadas e os valores médio da função objetivo obtido por cada algoritmo.

Tabela C.1: Resultados estendidos para classes de instância de pequeno porte

| **C. de Problema** | $MIP$ | $HR$ | $TAH$ | $IG+HR$ | $IG+MDL$ | $IG+F\&O$ |
|---|---|---|---|---|---|---|
| **3n3m3t_0** | 4050,6 | 4050,6 | 4176,6 | 4176,6 | 4186,6 | 4050,6 |
| **3n3m3t_1** | 4735,1 | 4735,1 | 4861,5 | 4861,5 | 4872 | 4735,1 |
| **3n3m3t_2** | 3714,3 | 3832,2 | 3828,2 | 3832,2 | 3814,2 | 3814,2 |
| **3n3m3t_3** | 5923 | 5923 | 5923 | 5923 | 5923 | 5923 |
| **3n3m3t_4** | 3312,9 | 3312,9 | 3562,8 | 3642,9 | 3509,3 | 3312,9 |
| **5n3m3t_0** | 9584 | 9717 | 9674,8 | 9821,8 | 9684,8 | 9584 |
| **5n3m3t_1** | 7305,1 | 7305,1 | 7600 | 7600 | 7604,3 | 7305,1 |
| **5n3m3t_2** | 8663,4 | 8700,4 | 8958,8 | 8958,8 | 8967,3 | 8672,03 |
| **5n3m3t_3** | 7858,5 | 7858,5 | 8347,7 | 8183,9 | 8218,9 | 7858,5 |
| **5n3m3t_4** | 8834,1 | 8884,8 | 9028,2 | 9049,2 | 9071 | 8843,5 |
| **3n5m3t_0** | 7292,2 | 7393,3 | 7564,6 | 7564,6 | 7581,7 | 7367,8 |
| **3n5m3t_1** | 7526,6 | 7526,6 | 7946,2 | 7946,2 | 7977,2 | 7526,6 |
| **3n5m3t_2** | 9965,3 | 9970,3 | 9968,3 | 9970,3 | 9965,3 | 9965,3 |
| **3n5m3t_3** | 9702,24 | 9731,94 | 9737,946 | 9731,94 | 9716,842 | 9710,756 |
| **3n5m3t_4** | 10391,1 | 10397,1 | 10391,1 | 10391,1 | 10402 | 10397,1 |
| **3n3m5t_0** | 8375,6 | 8525,9 | 8709,4 | 8709,4 | 8720,8 | 8452,8 |
| **3n3m5t_1** | 7461,8 | 7612,5 | 7931,5 | 7931,5 | 7943,5 | 7531,5 |
| **3n3m5t_2** | 7800,98 | 7911,38 | 8109,7 | 8210,5 | 8075,3 | 7867,5 |
| **3n3m5t_3** | 8079,7 | 8210,5 | 8119,5 | 8210,5 | 8081,3 | 8079,7 |
| **3n3m5t_4** | 7964,92 | 7964,92 | 8223,1 | 8223,1 | 8262,5 | 7964,92 |

*Continua na próxima página*

Tabela C.1 – *Continuação da página anterior*

| C. Problema | *MIP* | *HR* | *TAH* | *IG + HR* | *IG + MDL* | *IG + F&O* |
|---|---|---|---|---|---|---|
| **5n5m5t_0** | 28184,5 | 28206 | 28756,7 | 28756,7 | 28819,26 | 28298,06 |
| **5n5m5t_1** | 28756,7 | 28825,2 | 28756,7 | 28756,7 | 28816,7 | 28825,2 |
| **5n5m5t_2** | 20293,5 | 20446 | 21487,6 | 21487,6 | 21507,7 | 20354,91 |
| **5n5m5t_3** | 19254,1 | 19319,9 | 21256,2 | 21074,3 | 20365 | 19290,5 |
| **5n5m5t_4** | 22965,2 | 23229,2 | 24217,8 | 24217,8 | 24279,6 | 23034,6 |
| **7n5m5t_0** | 36037,4 | 36311,6 | 37376,1 | 37775,31 | 36571,42 | 36200,39 |
| **7n5m5t_1** | 29160,3 | 29461,3 | 31109 | 31062,6 | 30672,3 | 29160,3 |
| **7n5m5t_2** | 33277,8 | 33668 | 35247,4 | 35479,3 | 35244,5 | 33294,7 |
| **7n5m5t_3** | 32091,7 | 32196,6 | 33377,1 | 33774,3 | 33426 | 32178,11 |
| **7n5m5t_4** | 31965 | 32453,2 | 33377,1 | 33734,9 | 32965,29 | 32403,1 |
| **5n7m5t_0** | 30432,8 | 30441,2 | 32074,2 | 32074,2 | 32141,6 | 30432,8 |
| **5n7m5t_1** | 32038,9 | 32074,2 | 32074,2 | 32074,2 | 32038,9 | 32043,1 |
| **5n7m5t_2** | 34931,4 | 34931,4 | 35635,9 | 35635,9 | 35095,94 | 34931,8 |
| **5n7m5t_3** | 34985,4 | 36100,2 | 35635,9 | 35635,9 | 35600,94 | 35148,07 |
| **5n7m5t_4** | 34010,3 | 34511,5 | 35635,9 | 35635,9 | 37581,84 | 34399,4 |
| **5n5m7t_0** | 28166,2 | 28457,5 | 30405,5 | 30468,5 | 30403,3 | 28303,1 |
| **5n5m7t_1** | 31269,6 | 31571,1 | 31405,5 | 31468,5 | 32319,3 | 31333,7 |
| **5n5m7t_2** | 29489,9 | 29489,9 | 31089,3 | 31089,3 | 31267,5 | 29489,9 |
| **5n5m7t_3** | 30529,1 | 30786,2 | 32299,2 | 32564,2 | 32320,7 | 30677,9 |
| **5n5m7t_4** | 36871,2 | 36871,2 | 38608,46 | 38073,4 | 37701,17 | 36871,2 |
| **7n7m7t_0** | 60308,2 | 61153,2 | 64092,3 | 64395,9 | 63834,1 | 61868,3 |
| **7n7m7t_1** | 68530,5 | 69211,5 | 71403,4 | 71837,48 | 68843,99 | 68895,45 |
| **7n7m7t_2** | 58871,7 | 59471,3 | 63030,7 | 62837,1 | 62344,1 | 59368,9 |
| **7n7m7t_3** | 62003,1 | 62479,5 | 66127,2 | 66537 | 66070,5 | 62103,5 |
| **7n7m7t_4** | 61265 | 61534,7 | 65362,2 | 65363 | 65011,1 | 61367 |

Tabela C.2: Resultados estendidos para classes de instância de médio porte

| C. de Problema | *HR* | *TAH* | *IG + HR* | *IG + MDL* | *IG + F&O* |
|---|---|---|---|---|---|
| **5n5m10t_0** | 48308,5 | 47480,8 | 47051,14 | 47061,28 | 47753,8 |
| **5n5m10t_1** | 43048,4 | 45725,7 | 45908 | 45896,4 | 45725,7 |
| **5n5m10t_2** | 56698,6 | 56698,6 | 54427,54 | 53167,56 | 56698,6 |
| **5n5m10t_3** | 44926 | 45694,8 | 45741,1 | 45717,4 | 44400,8 |
| **5n5m10t_4** | 49844 | 49681,6 | 49829,6 | 49780,4 | 49708,6 |

*Continua na próxima página*

Tabela C.2 – *Continuação da página anterior*

| **C. Problema** | *HR* | *TAH* | *IG + HR* | *IG + MDL* | *IG + F&O* |
|---|---|---|---|---|---|
| **5n10m5t_0** | 39368,4 | 41624,9 | 41579,8 | 41548,3 | 41265,9 |
| **5n10m5t_1** | 37333,9 | 40066,2 | 39903,5 | 39922,52 | 35046,8 |
| **5n10m5t_2** | 33316,9 | 35649,4 | 35424,9 | 35468,7 | 31959,7 |
| **5n10m5t_3** | 41321,9 | 44331,07 | 38738,43 | 38755,9 | 41248,2 |
| **5n10m5t_4** | 29136,3 | 29978,4 | 29951,8 | 29949,9 | 28778,4 |
| **7n7m10t_0** | 99321 | 97009,32 | 96138,02 | 96221 | 94573,85 |
| **7n7m10t_1** | 61637,8 | 61044,9 | 60811,3 | 60871,9 | 61468,2 |
| **7n7m10t_2** | 98628,6 | 95595,7 | 93476,24 | 94003,87 | 90652,42 |
| **7n7m10t_3** | 65875,9 | 61476,7 | 60853,4 | 60891,9 | 61353,8 |
| **7n7m10t_4** | 54349,3 | 58653 | 57864,4 | 58000,3 | 53558,9 |
| **7n10m7t_0** | 99409,5 | 97989,19 | 88129,87 | 88740,41 | 93119,33 |
| **7n10m7t_1** | 108307 | 107468 | 99699,73 | 99888,41 | 107316,7 |
| **7n10m7t_2** | 59418,4 | 58943,3 | 58840,7 | 58870,9 | 49870,4 |
| **7n10m7t_3** | 98479 | 102715 | 98500,18 | 99679,91 | 89025,2 |
| **7n10m7t_4** | 51276,8 | 55722,2 | 55612,18 | 55847 | 50161,3 |
| **10n5m5t_0** | 49683,7 | 47038,5 | 47139,3 | 47192,3 | 47602,4 |
| **10n5m5t_1** | 47311,7 | 49758,5 | 49387,6 | 49489,6 | 45264,3 |
| **10n5m5t_2** | 59213,4 | 52430,8 | 51978,7 | 52028,3 | 36774,2 |
| **10n5m5t_3** | 49322 | 48536,3 | 48608,9 | 48681,8 | 49235,4 |
| **10n5m5t_4** | 29563,7 | 28728,4 | 28783,72 | 28776,45 | 29458,8 |
| **10n7m7t_0** | 102384 | 106768,3 | 106314,5 | 106398,6 | 107179,3 |
| **10n7m7t_1** | 59967,8 | 51520,5 | 51239,1 | 51393 | 38158 |
| **10n7m7t_2** | 99706,4 | 97144,2 | 97441,4 | 97505,2 | 92611,1 |
| **10n7m7t_3** | 90147,4 | 95227,4 | 95327,8 | 95486,9 | 89231,6 |
| **10n7m7t_4** | 97525,3 | 96045 | 95810,9 | 96001,7 | 87448,5 |
| **10n10m10t_0** | 222199 | 212654 | 211056,8 | 210957,8 | 212989,4 |
| **10n10m10t_1** | 114649,7 | 112977 | 112734,7 | 112880,8 | 114649,7 |
| **10n10m10t_2** | 120912 | 120912 | 119430,4 | 119715,3 | 120759,8 |
| **10n10m10t_3** | 125274 | 125146,6 | 123724,6 | 123931,7 | 125274 |
| **10n10m10t_4** | 106929,1 | 104857 | 104611,1 | 104731,9 | 105706,9 |
| **10n10m15t_0** | 165616 | 167688 | 168899,4 | 168744,2 | 150410,4 |
| **10n10m15t_1** | 175028 | 184545,5 | 185462,9 | 185813,6 | 167538,3 |
| **10n10m15t_2** | 145423 | 168856,5 | 168460,8 | 168615,8 | 140430,4 |
| **10n10m15t_3** | 149024 | 162490,5 | 163459,5 | 163654,8 | 134981,2 |

*Continua na próxima página*

Tabela C.2 – *Continuação da página anterior*

| C. Problema | $HR$ | $TAH$ | $IG + HR$ | $IG + MDL$ | $IG + F\&O$ |
|---|---|---|---|---|---|
| **10n10m15t_4** | 283911 | 302602,5 | 296379,4 | 297298,4 | 212330,4 |
| **10n15m10t_0** | 299107 | 297396,5 | 269269,8 | 272097 | 296244,9 |
| **10n15m10t_1** | 168956 | 157092,5 | 156819,3 | 157053,5 | 159086,9 |
| **10n15m10t_2** | 186326 | 176683,5 | 174749 | 176876 | 176320,5 |
| **10n15m10t_3** | 190230 | 176444,5 | 173925,7 | 174160,5 | 177087,8 |
| **10n15m10t_4** | 122274 | 156427,5 | 153861,4 | 153997,2 | 155959,3 |
| **15n10m10t_0** | 178502 | 170594,9 | 171371,1 | 169988,6 | 169708,4 |
| **15n10m10t_1** | 184539,5 | 184539,5 | 165544,2 | 163821,5 | 163721,8 |
| **15n10m10t_2** | 162126 | 160628,3 | 162126 | 160889,9 | 160754,3 |
| **15n10m10t_3** | 183572 | 176246,5 | 178953,3 | 177755,9 | 177472,9 |
| **15n10m10t_4** | 158947,5 | 157819,5 | 158947,5 | 157640,2 | 157322,2 |
| **15n15m15t_0** | 720361 | 700649,5 | 689130,8 | 694056,4 | 709674,3 |
| **15n15m15t_1** | 691507 | 683004,5 | 676709 | 677900,6 | 682309,5 |
| **15n15m15t_2** | 691422 | 685712,5 | 682302,7 | 683961,7 | 688752,6 |
| **15n15m15t_3** | 697469 | 683450,5 | 677662,2 | 677623,5 | 686802,6 |
| **15n15m15t_4** | 523938 | 412145,5 | 408853,9 | 409863,9 | 411890 |

Tabela C.3: Resultados estendidos para classes de instância de grande porte

| C. de Problema | $TAH$ | $IG + HR$ | $IG + MDL$ | $IG + F\&O$ |
|---|---|---|---|---|
| **20n10m10t_0** | 285987,2 | 282872,7 | 282677,9 | 282391,7 |
| **20n10m10t_1** | 289799,5 | 286618,3 | 286457,8 | 285067,4 |
| **20n10m10t_2** | 293776,2 | 290285,3 | 289985,7 | 290298,9 |
| **20n10m10t_3** | 292701 | 288922,8 | 288654,7 | 290407,2 |
| **20n10m10t_4** | 281092,9 | 278085,6 | 312762,5 | 277092,6 |
| **20n10m15t_0** | 435457,2 | 431621 | 431090 | 429720,6 |
| **20n10m15t_1** | 432057,6 | 426420,5 | 426158,3 | 424590,2 |
| **20n10m15t_2** | 424635,5 | 419362,1 | 418867,1 | 417502,4 |
| **20n10m15t_3** | 422026,2 | 415024,2 | 414451,8 | 413708,3 |
| **20n10m15t_4** | 422030,8 | 416546 | 416185 | 413397,6 |
| **20n10m20t_0** | 586449,8 | 580987,6 | 580329,7 | 577768,3 |
| **20n10m20t_1** | 584507,3 | 578775,7 | 577986,1 | 577662,6 |
| **20n10m20t_2** | 588857,8 | 581731,3 | 607989,7 | 580357,6 |
| **20n10m20t_3** | 580176,7 | 573558,5 | 573056,5 | 570470,8 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| **C. Problema** | *TAH* | *IG + HR* | *IG + MDL* | *IG + F&O* |
|---|---|---|---|---|
| **20n10m20t_4** | 571108,2 | 567089,2 | 566481 | 564315 |
| **20n15m10t_0** | 444891,2 | 440896,5 | 440340,9 | 441741,3 |
| **20n15m10t_1** | 434310,6 | 429617,4 | 429309 | 428683,3 |
| **20n15m10t_2** | 425467,5 | 421203,8 | 420719,5 | 422055,6 |
| **20n15m10t_3** | 460773,3 | 457321,1 | 464980,9 | 458144,1 |
| **20n15m10t_4** | 437128,9 | 433503,5 | 433047,4 | 434124,3 |
| **20n15m15t_0** | 663197,2 | 656940,2 | 656389,8 | 655644,2 |
| **20n15m15t_1** | 649394,5 | 642512,4 | 641822,3 | 643930,4 |
| **20n15m15t_2** | 655718,9 | 649236,7 | 648585,4 | 647178,7 |
| **20n15m15t_3** | 670457,1 | 663469,8 | 662384,2 | 663440,4 |
| **20n15m15t_4** | 656483,5 | 649033 | 648414,9 | 648718,3 |
| **20n15m20t_0** | 883805,8 | 873901,4 | 873067,4 | 874760,5 |
| **20n15m20t_1** | 864765,6 | 858681,3 | 857754,7 | 861301 |
| **20n15m20t_2** | 863540,2 | 854330,3 | 853435,9 | 851720,1 |
| **20n15m20t_3** | 887894,8 | 880903,7 | 879997,6 | 881065,5 |
| **20n15m20t_4** | 883683,9 | 876597,9 | 875468,6 | 873567,3 |
| **20n20m10t_0** | 594442,1 | 591044,3 | 590623,2 | 592885,3 |
| **20n20m10t_1** | 589805,6 | 580292,2 | 579833,8 | 579339,1 |
| **20n20m10t_2** | 596022 | 590290 | 589686,9 | 593337,6 |
| **20n20m10t_3** | 601480,4 | 597266,4 | 596582,6 | 599084,1 |
| **20n20m10t_4** | 620340,5 | 612848,9 | 612296,6 | 616709,7 |
| **20n20m15t_0** | 896764,1 | 890582,2 | 889983,6 | 891129 |
| **20n20m15t_1** | 861255,3 | 855979,1 | 855322 | 855672,7 |
| **20n20m15t_2** | 859743,9 | 851597,1 | 850792,1 | 850506,7 |
| **20n20m15t_3** | 858337,3 | 846070,8 | 845142,3 | 851223,8 |
| **20n20m15t_4** | 892812,2 | 886129,1 | 885252,3 | 890384,4 |
| **20n20m20t_0** | 1,14E+006 | 1124308 | 1123621 | 1127201 |
| **20n20m20t_1** | 1,18E+006 | 1169229 | 1168209 | 1171709 |
| **20n20m20t_2** | 1,15E+006 | 1137267 | 1136363 | 1140483 |
| **20n20m20t_3** | 1,14E+006 | 1123220 | 1122031 | 1125169 |
| **20n20m20t_4** | 1,16E+006 | 1156707 | 1155880 | 1156690 |
| **25n10m10t_0** | 359130 | 354558,7 | 354001,4 | 352025,6 |
| **25n10m10t_1** | 359363,9 | 355698,1 | 355259,2 | 354585,9 |
| **25n10m10t_2** | 365546,2 | 362319,9 | 361648,8 | 359024,1 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| **C. Problema** | *TAH* | *IG + HR* | *IG + MDL* | *IG + F&O* |
|---|---|---|---|---|
| **25n10m10t_3** | 348988,4 | 343914,7 | 343126,1 | 342220,6 |
| **25n10m10t_4** | 372929,8 | 369132,3 | 368363,8 | 366199 |
| **25n10m15t_0** | 543138,8 | 536885,3 | 536195,6 | 530983,7 |
| **25n10m15t_1** | 557162,9 | 551662,5 | 550665,5 | 547262 |
| **25n10m15t_2** | 535257,2 | 532282,8 | 531187,4 | 528768,3 |
| **25n10m15t_3** | 565017,7 | 558535,3 | 557740 | 555443 |
| **25n10m15t_4** | 530792,5 | 524050,3 | 523177,1 | 520090,6 |
| **25n10m20t_0** | 690671,9 | 683051,6 | 682247,2 | 679841,3 |
| **25n10m20t_1** | 742978,3 | 737163,8 | 735179,6 | 732218,8 |
| **25n10m20t_2** | 706274,1 | 696894,7 | 694894,8 | 691187,2 |
| **25n10m20t_3** | 713530,5 | 703668,2 | 702435,2 | 702873,2 |
| **25n10m20t_4** | 734482,3 | 727286,3 | 726489 | 721645,6 |
| **25n15m10t_0** | 563383,6 | 559532,9 | 558624,6 | 557821 |
| **25n15m10t_1** | 531109,1 | 524359,1 | 523217,4 | 521240 |
| **25n15m10t_2** | 541641,6 | 537131,2 | 536233,2 | 535156,1 |
| **25n15m10t_3** | 548329 | 542472,8 | 541148,5 | 542306,8 |
| **25n15m10t_4** | 529997,7 | 526515,2 | 525492 | 522707,8 |
| **25n15m15t_0** | 832514,4 | 826045,1 | 825190 | 824466,4 |
| **25n15m15t_1** | 824397,5 | 817259,5 | 816443,7 | 816250,6 |
| **25n15m15t_2** | 849034,8 | 843081 | 842381,8 | 841524,3 |
| **25n15m15t_3** | 799432,6 | 794219,5 | 792347,4 | 790120,1 |
| **25n15m15t_4** | 801939,6 | 793903,6 | 792460,6 | 791647,3 |
| **25n15m20t_0** | 1,08E+006 | 1076288 | 1075560 | 1071566 |
| **25n15m20t_1** | 1,07E+006 | 1059159 | 1057880 | 1056562 |
| **25n15m20t_2** | 1,10E+006 | 1088594 | 1087146 | 1086057 |
| **25n15m20t_3** | 1,08E+006 | 1066250 | 1063610 | 1059557 |
| **25n15m20t_4** | 1,08E+006 | 1073952 | 1072375 | 1068555 |
| **25n20m10t_0** | 729645,9 | 723487,2 | 722463,9 | 723732,1 |
| **25n20m10t_1** | 720624,4 | 713772,9 | 713023,5 | 713951,4 |
| **25n20m10t_2** | 708765,9 | 702418,8 | 701611,3 | 701343,2 |
| **25n20m10t_3** | 709505,4 | 700340,2 | 699605,9 | 699616,4 |
| **25n20m10t_4** | 715922,5 | 709440,1 | 708316,5 | 707568,9 |
| **25n20m15t_0** | 1,07E+006 | 1060451 | 1059579 | 1056844 |
| **25n20m15t_1** | 1,11E+006 | 1097656 | 1097119 | 1097956 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| **C. Problema** | $TAH$ | $IG + HR$ | $IG + MDL$ | $IG + F\&O$ |
|---|---|---|---|---|
| **25n20m15t_2** | 1,09E+006 | 1073821 | 1073087 | 1071870 |
| **25n20m15t_3** | 1,09E+006 | 1076946 | 1075909 | 1072533 |
| **25n20m15t_4** | 1,07E+006 | 1064528 | 1062943 | 1064469 |
| **25n20m20t_0** | 1,47E+006 | 1455834 | 1454767 | 1449940 |
| **25n20m20t_1** | 1,48E+006 | 1464404 | 1463306 | 1463271 |
| **25n20m20t_2** | 1,46E+006 | 1448982 | 1447690 | 1446529 |
| **25n20m20t_3** | 1,42E+006 | 1410108 | 1407891 | 1408766 |
| **25n20m20t_4** | 1,43E+006 | 1414345 | 1411601 | 1415689 |
| **30n10m10t_0** | 435192,9 | 431578 | 431261,1 | 426798,7 |
| **30n10m10t_1** | 440123,4 | 434372,8 | 433479 | 430840 |
| **30n10m10t_2** | 451605,9 | 443652,9 | 442920,3 | 438530,8 |
| **30n10m10t_3** | 446498,2 | 443059,1 | 442052,8 | 439734,5 |
| **30n10m10t_4** | 441153,3 | 436351,5 | 435656,3 | 432550,2 |
| **30n10m15t_0** | 669892,3 | 664724,4 | 664342,4 | 659467,4 |
| **30n10m15t_1** | 659100 | 650694 | 650138,2 | 644142,1 |
| **30n10m15t_2** | 645069,1 | 638976,9 | 637797,4 | 632728,7 |
| **30n10m15t_3** | 652213,2 | 646598,1 | 644881,5 | 638696,1 |
| **30n10m15t_4** | 644701,5 | 638853,7 | 636558,4 | 630870,1 |
| **30n10m20t_0** | 890096 | 879246,6 | 878275,1 | 869978,5 |
| **30n10m20t_1** | 877587,6 | 868158,5 | 865963,6 | 859012,4 |
| **30n10m20t_2** | 873526 | 863915,5 | 862296,4 | 856470,7 |
| **30n10m20t_3** | 872000,9 | 864446,7 | 863313,1 | 857116,4 |
| **30n10m20t_4** | 869594,3 | 857357,1 | 855777,8 | 850535,8 |
| **30n15m10t_0** | 633409 | 626059,3 | 625440,6 | 621965,8 |
| **30n15m10t_1** | 659014,6 | 653820,9 | 653078,5 | 652252,2 |
| **30n15m10t_2** | 648642 | 641167,9 | 640090,4 | 638096,8 |
| **30n15m10t_3** | 635961,4 | 631241,8 | 630247,9 | 627542,8 |
| **30n15m10t_4** | 638645,3 | 631402,7 | 630488,1 | 628846 |
| **30n15m15t_0** | 990586,8 | 979875,1 | 978911,4 | 974452,3 |
| **30n15m15t_1** | 954281,9 | 943007,9 | 942150,8 | 939176,2 |
| **30n15m15t_2** | 994644,8 | 987384,4 | 986477,2 | 982733,8 |
| **30n15m15t_3** | 977913,9 | 969686,1 | 968812,4 | 964091 |
| **30n15m15t_4** | 997353,6 | 988820,1 | 987275,7 | 982962,9 |
| **30n15m20t_0** | 1,32E+006 | 1305714 | 1304934 | 1297782 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| C. Problema | *TAH* | *IG + HR* | *IG + MDL* | *IG + F&O* |
|---|---|---|---|---|
| **30n15m20t_1** | 1,31E+006 | 1300272 | 1299218 | 1294629 |
| **30n15m20t_2** | 1,29E+006 | 1280931 | 1279525 | 1270768 |
| **30n15m20t_3** | 1,32E+006 | 1311998 | 1309483 | 1303498 |
| **30n15m20t_4** | 1,29E+006 | 1279687 | 1277043 | 1269845 |
| **30n20m10t_0** | 842161,4 | 830500,6 | 829600,1 | 831018,7 |
| **30n20m10t_1** | 886670,1 | 878371,2 | 877347,6 | 876763,7 |
| **30n20m10t_2** | 873469 | 864706,2 | 863234,2 | 863366,1 |
| **30n20m10t_3** | 855375,3 | 848973,7 | 847950,2 | 845766,3 |
| **30n20m10t_4** | 856813,2 | 850976,9 | 849257,3 | 847244,5 |
| **30n20m15t_0** | 1,31E+006 | 1305057 | 1304022 | 1298156 |
| **30n20m15t_1** | 1,27E+006 | 1264503 | 1263246 | 1257471 |
| **30n20m15t_2** | 1,33E+006 | 1318385 | 1317535 | 1313531 |
| **30n20m15t_3** | 1,31E+006 | 1298834 | 1297679 | 1296798 |
| **30n20m15t_4** | 1,28E+006 | 1267538 | 1266649 | 1264447 |
| **30n20m20t_0** | 1,74E+006 | 1725720 | 1724214 | 1718120 |
| **30n20m20t_1** | 1,76E+006 | 1746955 | 1745315 | 1738693 |
| **30n20m20t_2** | 1,79E+006 | 1777557 | 1776208 | 1771947 |
| **30n20m20t_3** | 1,76E+006 | 1750877 | 1748631 | 1747546 |
| **30n20m20t_4** | 1,76E+006 | 1738150 | 1736196 | 1732996 |
| **35n10m10t_0** | 508674,7 | 503057,5 | 502136,9 | 498262,3 |
| **35n10m10t_1** | 521654 | 514644,5 | 513959,4 | 509442,6 |
| **35n10m10t_2** | 509509,6 | 504290,3 | 502712 | 499580,3 |
| **35n10m10t_3** | 511723,8 | 506383,6 | 505606,7 | 500526,5 |
| **35n10m10t_4** | 513092,9 | 508434,9 | 507290,5 | 503356 |
| **35n10m15t_0** | 765869,8 | 758755,5 | 758198,3 | 749164,6 |
| **35n10m15t_1** | 754549,8 | 746140,2 | 745308 | 739417,6 |
| **35n10m15t_2** | 775596,5 | 769320,9 | 767872,6 | 761319,9 |
| **35n10m15t_3** | 760496,8 | 752637,3 | 750704,7 | 743023,2 |
| **35n10m15t_4** | 773428,7 | 767946,9 | 766400,4 | 758505,7 |
| **35n10m20t_0** | 1,01E+006 | 996050,4 | 994630,2 | 983843,7 |
| **35n10m20t_1** | 1,00E+006 | 992616,9 | 991645,3 | 978334,6 |
| **35n10m20t_2** | 1,00E+006 | 991490,8 | 989125,8 | 978381,3 |
| **35n10m20t_3** | 1,02E+006 | 1017456 | 1016096 | 1003201 |
| **35n10m20t_4** | 1,02E+006 | 1018578 | 1016067 | 1003676 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| C. Problema | $TAH$ | $IG + HR$ | $IG + MDL$ | $IG + F\&O$ |
|---|---|---|---|---|
| **35n15m10t_0** | 769467,2 | 763204,3 | 762717,2 | 757794,7 |
| **35n15m10t_1** | 752777 | 744793 | 744285,3 | 741688,4 |
| **35n15m10t_2** | 783199,9 | 777234,6 | 776560,6 | 772080,3 |
| **35n15m10t_3** | 754159,3 | 746120,3 | 745422,3 | 741052,2 |
| **35n15m10t_4** | 791176,2 | 783310,7 | 782820,8 | 778059,9 |
| **35n15m15t_0** | 1,13E+006 | 1125126 | 1124237 | 1118706 |
| **35n15m15t_1** | 1,12E+006 | 1107124 | 1106457 | 1099067 |
| **35n15m15t_2** | 1,14E+006 | 1131389 | 1130514 | 1124557 |
| **35n15m15t_3** | 1,14E+006 | 1131565 | 1130345 | 1121129 |
| **35n15m15t_4** | 1,12E+006 | 1114974 | 1113793 | 1105414 |
| **35n15m20t_0** | 1,51E+006 | 1504252 | 1502668 | 1493299 |
| **35n15m20t_1** | 1,51E+006 | 1493588 | 1492607 | 1479645 |
| **35n15m20t_2** | 1,51E+006 | 1496918 | 1495711 | 1485236 |
| **35n15m20t_3** | 1,51E+006 | 1501356 | 1500183 | 1490880 |
| **35n15m20t_4** | 1,55E+006 | 1540913 | 1536778 | 1530106 |
| **35n20m10t_0** | 1,02E+006 | 1011552 | 1010255 | 1005377 |
| **35n20m10t_1** | 1,02E+006 | 1011963 | 1011009 | 1007927 |
| **35n20m10t_2** | 1,01E+006 | 1002079 | 1001151,8 | 997306,5 |
| **35n20m10t_3** | 995532,6 | 986425,6 | 985744,3 | 979086,3 |
| **35n20m10t_4** | 1,03E+006 | 1028226 | 1027111 | 1021787 |
| **35n20m15t_0** | 1,56E+006 | 1547810 | 1546488 | 1540871 |
| **35n20m15t_1** | 1,56E+006 | 1543319 | 1542404 | 1537300 |
| **35n20m15t_2** | 1,52E+006 | 1508378 | 1507409 | 1501626 |
| **35n20m15t_3** | 1,52E+006 | 1510407 | 1509170 | 1502041 |
| **35n20m15t_4** | 1,51E+006 | 1499046 | 1497773 | 1487398 |
| **35n20m20t_0** | 2,06E+006 | 2045759 | 2044264 | 2035935 |
| **35n20m20t_1** | 2,03E+006 | 2018275 | 2016649 | 2007840 |
| **35n20m20t_2** | 2,03E+006 | 2013537 | 2011997 | 2003850 |
| **35n20m20t_3** | 2,03E+006 | 2018725 | 2017470 | 2008667 |
| **35n20m20t_4** | 2,05E+006 | 2036528 | 2035185 | 2027432 |
| **40n10m10t_0** | 583014,5 | 578193,2 | 577474,2 | 570695,4 |
| **40n10m10t_1** | 594435,3 | 589353,7 | 588596,7 | 580755,3 |
| **40n10m10t_2** | 601116,2 | 595045,1 | 594396 | 587665,8 |
| **40n10m10t_3** | 581837,9 | 575121,3 | 573969 | 569276,9 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| **C. Problema** | *TAH* | *IG + HR* | *IG + MDL* | *IG + F&O* |
|---|---|---|---|---|
| **40n10m10t_4** | 602449,7 | 594957,1 | 593343,2 | 589313,4 |
| **40n10m15t_0** | 878734,7 | 871759,9 | 871066,5 | 857576,2 |
| **40n10m15t_1** | 872940,6 | 865682,7 | 864674,1 | 854960,2 |
| **40n10m15t_2** | 861737,5 | 853823,9 | 852928,8 | 840666,6 |
| **40n10m15t_3** | 872304,1 | 864242,1 | 862459,5 | 851580,6 |
| **40n10m15t_4** | 891126,7 | 883007,7 | 881916,5 | 869757,5 |
| **40n10m20t_0** | 1,19E+006 | 1173073 | 1171921 | 1154665 |
| **40n10m20t_1** | 1,19E+006 | 1176224 | 1174619 | 1161216 |
| **40n10m20t_2** | 1,15E+006 | 1144402 | 1142550 | 1126187 |
| **40n10m20t_3** | 1,17E+006 | 1158471 | 1156932 | 1142786 |
| **40n10m20t_4** | 1,15E+006 | 1136784 | 1134446 | 1119450 |
| **40n15m10t_0** | 863374 | 857114,9 | 855953,2 | 848596,3 |
| **40n15m10t_1** | 848754,6 | 843198 | 842546 | 834596,2 |
| **40n15m10t_2** | 867490,5 | 860110,7 | 859352,3 | 851530,4 |
| **40n15m10t_3** | 901323,2 | 894743,2 | 893973,8 | 885714,2 |
| **40n15m10t_4** | 885971,8 | 878695,2 | 877906,7 | 872499,3 |
| **40n15m15t_0** | 1,32E+006 | 1309006 | 1307978 | 1296281 |
| **40n15m15t_1** | 1,28E+006 | 1275011 | 1273712 | 1261669 |
| **40n15m15t_2** | 1,28E+006 | 1268178 | 1267050 | 1254055 |
| **40n15m15t_3** | 1,29E+006 | 1276279 | 1275039 | 1265725 |
| **40n15m15t_4** | 1,32E+006 | 1312718 | 1311391 | 1302012 |
| **40n15m20t_0** | 1,72E+006 | 1706765 | 1705512 | 1686240 |
| **40n15m20t_1** | 1,74E+006 | 1724835 | 1723665 | 1708491 |
| **40n15m20t_2** | 1,75E+006 | 1740629 | 1739033 | 1724653 |
| **40n15m20t_3** | 1,72E+006 | 1706415 | 1704957 | 1691470 |
| **40n15m20t_4** | 1,74E+006 | 1725881 | 1724366 | 1706463 |
| **40n20m10t_0** | 1,16E+006 | 1151845 | 1150990 | 1144702 |
| **40n20m10t_1** | 1,16E+006 | 1150134 | 1149071 | 1141857 |
| **40n20m10t_2** | 1,15E+006 | 1144185 | 1143254 | 1136588 |
| **40n20m10t_3** | 1,19E+006 | 1183100 | 1182140 | 1178683 |
| **40n20m10t_4** | 1,21E+006 | 1195495 | 1194571 | 1188354 |
| **40n20m15t_0** | 1,71E+006 | 1697526 | 1696358 | 1685578 |
| **40n20m15t_1** | 1,76E+006 | 1742696 | 1741338 | 1731257 |
| **40n20m15t_2** | 1,76E+006 | 1746831 | 1745650 | 1733230 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| C. Problema | $TAH$ | $IG + HR$ | $IG + MDL$ | $IG + F\&O$ |
|---|---|---|---|---|
| **40n20m15t_3** | 1,76E+006 | 1748782 | 1747206 | 1735356 |
| **40n20m15t_4** | 1,78E+006 | 1771450 | 1770278 | 1758354 |
| **40n20m20t_0** | 2,30E+006 | 2274109 | 2272451 | 2256670 |
| **40n20m20t_1** | 2,30E+006 | 2285580 | 2284093 | 2262810 |
| **40n20m20t_2** | 2,34E+006 | 2324309 | 2322545 | 2305540 |
| **40n20m20t_3** | 2,37E+006 | 2356210 | 2354429 | 2331830 |
| **40n20m20t_4** | 2,29E+006 | 2277663 | 2276161 | 2255660 |
| **45n10m10t_0** | 650787,7 | 643859 | 643329,1 | 632310 |
| **45n10m10t_1** | 664209,3 | 658322,8 | 657178,7 | 648366 |
| **45n10m10t_2** | 650488,5 | 644344,5 | 644518,3 | 632611 |
| **45n10m10t_3** | 626243,2 | 621589,1 | 620549,7 | 609401 |
| **45n10m10t_4** | 656238,1 | 649591,4 | 648804,8 | 638433 |
| **45n10m15t_0** | 977385 | 967840,8 | 966670 | 949893 |
| **45n10m15t_1** | 970463,8 | 964628,2 | 963595,6 | 946779 |
| **45n10m15t_2** | 978773,1 | 970325,7 | 968964,4 | 952372 |
| **45n10m15t_3** | 981886,6 | 974086,2 | 972874,9 | 957209 |
| **45n10m15t_4** | 984933,5 | 976409,4 | 974377 | 959510 |
| **45n10m20t_0** | 1,31E+006 | 1300434 | 1298547 | 1272060 |
| **45n10m20t_1** | 1,32E+006 | 1312262 | 1310378 | 1287030 |
| **45n10m20t_2** | 1,29E+006 | 1279540 | 1277641 | 1256210 |
| **45n10m20t_3** | 1,32E+006 | 1313805 | 1309868 | 1289480 |
| **45n10m20t_4** | 1,31E+006 | 1301565 | 1299991 | 1280300 |
| **45n15m10t_0** | 1,01E+006 | 1004753 | 1004056 | 991166 |
| **45n15m10t_1** | 993131,4 | 986329,2 | 985395,3 | 974409 |
| **45n15m10t_2** | 951628,3 | 942621,6 | 942139,1 | 930039 |
| **45n15m10t_3** | 978586,5 | 968700,4 | 967826,2 | 957374 |
| **45n15m10t_4** | 1,00E+006 | 991335,2 | 990625,6 | 980520 |
| **45n15m15t_0** | 1,48E+006 | 1470208 | 1468599 | 1450950 |
| **45n15m15t_1** | 1,47E+006 | 1453565 | 1451922 | 1435310 |
| **45n15m15t_2** | 1,46E+006 | 1450028 | 1448774 | 1430560 |
| **45n15m15t_3** | 1,47E+006 | 1454645 | 1453381 | 1434950 |
| **45n15m15t_4** | 1,50E+006 | 1486002 | 1484781 | 1465470 |
| **45n15m20t_0** | 1,95E+006 | 1935031 | 1933043 | 1911670 |
| **45n15m20t_1** | 1,95E+006 | 1939770 | 1937889 | 1913110 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| **C. Problema** | *TAH* | *IG + HR* | *IG + MDL* | *IG + F&O* |
|---|---|---|---|---|
| **45n15m20t_2** | 1,94E+006 | 1921278 | 1919581 | 1892040 |
| **45n15m20t_3** | 1,96E+006 | 1951267 | 1949663 | 1924330 |
| **45n15m20t_4** | 1,95E+006 | 1939293 | 1937435 | 1914480 |
| **45n20m10t_0** | 1,31E+006 | 1307120 | 1306402 | 1296750 |
| **45n20m10t_1** | 1,29E+006 | 1280152 | 1279094 | 1265660 |
| **45n20m10t_2** | 1,28E+006 | 1269990 | 1269702 | 1254080 |
| **45n20m10t_3** | 1,29E+006 | 1286060 | 1285008 | 1271900 |
| **45n20m10t_4** | 1,32E+006 | 1308738 | 1307847 | 1296820 |
| **45n20m15t_0** | 1,95E+006 | 1939064 | 1937804 | 1918400 |
| **45n20m15t_1** | 1,92E+006 | 1903081 | 1901606 | 1883070 |
| **45n20m15t_2** | 1,93E+006 | 1918392 | 1916636 | 1898910 |
| **45n20m15t_3** | 1,98E+006 | 1972250 | 1970664 | 1952050 |
| **45n20m15t_4** | 1,97E+006 | 1956253 | 1954821 | 1939300 |
| **45n20m20t_0** | 2,61E+006 | 2595226 | 2593422 | 2569070 |
| **45n20m20t_1** | 2,63E+006 | 2613720 | 2611730 | 2589570 |
| **45n20m20t_2** | 2,64E+006 | 2626333 | 2624500 | 2600520 |
| **45n20m20t_3** | 2,64E+006 | 2623828 | 2621598 | 2596780 |
| **45n20m20t_4** | 2,56E+006 | 2541839 | 2539807 | 2512790 |
| **50n10m10t_0** | 744350,6 | 739524,7 | 739091,9 | 724511 |
| **50n10m10t_1** | 717181,9 | 709760 | 708959,7 | 695527 |
| **50n10m10t_2** | 722845,2 | 714404,7 | 713333,4 | 700912 |
| **50n10m10t_3** | 737630,3 | 731794,4 | 730912 | 719663 |
| **50n10m10t_4** | 716592,8 | 710807 | 709732,6 | 699631 |
| **50n10m15t_0** | 1,08E+006 | 1074504 | 1073178 | 1055580 |
| **50n10m15t_1** | 1,08E+006 | 1065712 | 1065093 | 1041870 |
| **50n10m15t_2** | 1,08E+006 | 1067088 | 1066039 | 1045230 |
| **50n10m15t_3** | 1,10E+006 | 1090266 | 1089345 | 1067800 |
| **50n10m15t_4** | 1,09E+006 | 1083421 | 1081730 | 1061980 |
| **50n10m20t_0** | 1,47E+006 | 1461270 | 1459454 | 1433240 |
| **50n10m20t_1** | 1,47E+006 | 1460393 | 1458368 | 1433110 |
| **50n10m20t_2** | 1,47E+006 | 1456001 | 1453773 | 1427570 |
| **50n10m20t_3** | 1,48E+006 | 1473160 | 1470273 | 1444610 |
| **50n10m20t_4** | 1,44E+006 | 1431349 | 1428033 | 1401880 |
| **50n15m10t_0** | 1,13E+006 | 1124634 | 1123562 | 1110120 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| C. Problema | $TAH$ | $IG + HR$ | $IG + MDL$ | $IG + F\&O$ |
|---|---|---|---|---|
| **50n15m10t_1** | 1,12E+006 | 1111032 | 1110285 | 1092690 |
| **50n15m10t_2** | 1,06E+006 | 1054257 | 1053339 | 1039760 |
| **50n15m10t_3** | 1,11E+006 | 1097091 | 1096415 | 1082180 |
| **50n15m10t_4** | 1,08E+006 | 1072169 | 1071207 | 1057140 |
| **50n15m15t_0** | 1,69E+006 | 1675389 | 1674380 | 1651330 |
| **50n15m15t_1** | 1,64E+006 | 1624685 | 1623348 | 1603100 |
| **50n15m15t_2** | 1,59E+006 | 1582895 | 1581109 | 1558420 |
| **50n15m15t_3** | 1,67E+006 | 1657944 | 1656413 | 1631140 |
| **50n15m15t_4** | 1,63E+006 | 1623376 | 1622356 | 1599370 |
| **50n15m20t_0** | 2,19E+006 | 2180609 | 2178767 | 2147660 |
| **50n15m20t_1** | 2,19E+006 | 2168452 | 2166894 | 2133600 |
| **50n15m20t_2** | 2,19E+006 | 2173916 | 2171945 | 2138910 |
| **50n15m20t_3** | 2,19E+006 | 2166725 | 2164572 | 2133780 |
| **50n15m20t_4** | 2,16E+006 | 2146010 | 2143992 | 2112660 |
| **50n20m10t_0** | 1,46E+006 | 1447719 | 1446814 | 1432660 |
| **50n20m10t_1** | 1,45E+006 | 1438873 | 1438021 | 1420710 |
| **50n20m10t_2** | 1,45E+006 | 1434124 | 1432871 | 1419480 |
| **50n20m10t_3** | 1,49E+006 | 1480182 | 1479123 | 1464280 |
| **50n20m10t_4** | 1,42E+006 | 1410954 | 1409828 | 1395070 |
| **50n20m15t_0** | 2,15E+006 | 2135509 | 2133611 | 2111430 |
| **50n20m15t_1** | 2,20E+006 | 2176549 | 2174917 | 2151960 |
| **50n20m15t_2** | 2,22E+006 | 2201681 | 2200110 | 2181150 |
| **50n20m15t_3** | 2,18E+006 | 2163715 | 2161990 | 2137970 |
| **50n20m15t_4** | 2,16E+006 | 2148228 | 2146666 | 2125560 |
| **50n20m20t_0** | 2,92E+006 | 2894034 | 2891309 | 2858170 |
| **50n20m20t_1** | 2,88E+006 | 2861497 | 2859186 | 2827120 |
| **50n20m20t_2** | 2,89E+006 | 2863225 | 2861354 | 2829170 |
| **50n20m20t_3** | 2,89E+006 | 2874790 | 2872450 | 2842850 |
| **50n20m20t_4** | 2,99E+006 | 2966598 | 2964877 | 2928790 |
| **55n10m10t_0** | 804041,6 | 798656,4 | 797748,7 | 781048 |
| **55n10m10t_1** | 808409,3 | 801699 | 801261,7 | 783544 |
| **55n10m10t_2** | 830275,1 | 824286,6 | 823512,9 | 807499 |
| **55n10m10t_3** | 829872,9 | 823109,5 | 822304,2 | 805182 |
| **55n10m10t_4** | 803541,8 | 797528,9 | 796455 | 780364 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| C. Problema | $TAH$ | $IG + HR$ | $IG + MDL$ | $IG + F\&O$ |
|---|---|---|---|---|
| **55n10m15t_0** | 1,18E+006 | 1171643 | 1170070 | 1145710 |
| **55n10m15t_1** | 1,21E+006 | 1205836 | 1203938 | 1183160 |
| **55n10m15t_2** | 1,21E+006 | 1203314 | 1202074 | 1179540 |
| **55n10m15t_3** | 1,22E+006 | 1205102 | 1205227 | 1178210 |
| **55n10m15t_4** | 1,22E+006 | 1202694 | 1201128 | 1179330 |
| **55n10m20t_0** | 1,60E+006 | 1586086 | 1583521 | 1552670 |
| **55n10m20t_1** | 1,62E+006 | 1601545 | 1599413 | 1567480 |
| **55n10m20t_2** | 1,60E+006 | 1591615 | 1588684 | 1558330 |
| **55n10m20t_3** | 1,61E+006 | 1599883 | 1597195 | 1564800 |
| **55n10m20t_4** | 1,59E+006 | 1573803 | 1572165 | 1536050 |
| **55n15m10t_0** | 1,23E+006 | 1218272 | 1216671 | 1200720 |
| **55n15m10t_1** | 1,19E+006 | 1180060 | 1179194 | 1162530 |
| **55n15m10t_2** | 1,19E+006 | 1180057 | 1179358 | 1162920 |
| **55n15m10t_3** | 1,20E+006 | 1197373 | 1196447 | 1179430 |
| **55n15m10t_4** | 1,22E+006 | 1210196 | 1209670 | 1194110 |
| **55n15m15t_0** | 1,77E+006 | 1764960 | 1762888 | 1733720 |
| **55n15m15t_1** | 1,80E+006 | 1787576 | 1786362 | 1757780 |
| **55n15m15t_2** | 1,81E+006 | 1790254 | 1789077 | 1760730 |
| **55n15m15t_3** | 1,81E+006 | 1800526 | 1798445 | 1772720 |
| **55n15m15t_4** | 1,83E+006 | 1811097 | 1809632 | 1783390 |
| **55n15m20t_0** | 2,42E+006 | 2408891 | 2407451 | 2370270 |
| **55n15m20t_1** | 2,39E+006 | 2365761 | 2364034 | 2330010 |
| **55n15m20t_2** | 2,37E+006 | 2357657 | 2355570 | 2319980 |
| **55n15m20t_3** | 2,41E+006 | 2392706 | 2390761 | 2354720 |
| **55n15m20t_4** | 2,42E+006 | 2411033 | 2408796 | 2376090 |
| **55n20m10t_0** | 1,58E+006 | 1570803 | 1569508 | 1551470 |
| **55n20m10t_1** | 1,61E+006 | 1602712 | 1601984 | 1584450 |
| **55n20m10t_2** | 1,62E+006 | 1607346 | 1606472 | 1586320 |
| **55n20m10t_3** | 1,60E+006 | 1583887 | 1582966 | 1565670 |
| **55n20m10t_4** | 1,62E+006 | 1612097 | 1610801 | 1582780 |
| **55n20m15t_0** | 2,47E+006 | 2451126 | 2449349 | 2419580 |
| **55n20m15t_1** | 2,37E+006 | 2350288 | 2348475 | 2320830 |
| **55n20m15t_2** | 2,46E+006 | 2442500 | 2440770 | 2413050 |
| **55n20m15t_3** | 2,42E+006 | 2408026 | 2406549 | 2377230 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| **C. Problema** | *TAH* | $IG + HR$ | $IG + MDL$ | $IG + F\&O$ |
|---|---|---|---|---|
| **55n20m15t_4** | 2,38E+006 | 2367937 | 2366826 | 2336040 |
| **55n20m20t_0** | 3,21E+006 | 3194820 | 3192150 | 3154330 |
| **55n20m20t_1** | 3,23E+006 | 3207690 | 3205203 | 3167270 |
| **55n20m20t_2** | 3,25E+006 | 3224529 | 3222160 | 3181240 |
| **55n20m20t_3** | 3,21E+006 | 3191197 | 3188738 | 3150360 |
| **55n20m20t_4** | 3,29E+006 | 3267941 | 3265553 | 3226780 |
| **60n10m10t_0** | 882386,6 | 873817,5 | 872824,2 | 853421 |
| **60n10m10t_1** | 891766,6 | 883900,5 | 882622,8 | 865880 |
| **60n10m10t_2** | 886104,9 | 880247,4 | 878429,3 | 861645 |
| **60n10m10t_3** | 880190,6 | 872174,7 | 871076,8 | 856210 |
| **60n10m10t_4** | 877027,2 | 868284,8 | 868819,9 | 851383 |
| **60n10m15t_0** | 1,31E+006 | 1300805 | 1299042 | 1268100 |
| **60n10m15t_1** | 1,32E+006 | 1312522 | 1311064 | 1283470 |
| **60n10m15t_2** | 1,30E+006 | 1289131 | 1287949 | 1258700 |
| **60n10m15t_3** | 1,31E+006 | 1302274 | 1301818 | 1274350 |
| **60n10m15t_4** | 1,32E+006 | 1310740 | 1309072 | 1283230 |
| **60n10m20t_0** | 1,74E+006 | 1723120 | 1721697 | 1680810 |
| **60n10m20t_1** | 1,73E+006 | 1717723 | 1715163 | 1676690 |
| **60n10m20t_2** | 1,73E+006 | 1710451 | 1707449 | 1671810 |
| **60n10m20t_3** | 1,76E+006 | 1744168 | 1741419 | 1703550 |
| **60n10m20t_4** | 1,75E+006 | 1740186 | 1738229 | 1700920 |
| **60n15m10t_0** | 1,32E+006 | 1314995 | 1314216 | 1292640 |
| **60n15m10t_1** | 1,30E+006 | 1291988 | 1291374 | 1269090 |
| **60n15m10t_2** | 1,28E+006 | 1272765 | 1271498 | 1252920 |
| **60n15m10t_3** | 1,32E+006 | 1309493 | 1309219 | 1288700 |
| **60n15m10t_4** | 1,30E+006 | 1293499 | 1292445 | 1274980 |
| **60n15m15t_0** | 1,91E+006 | 1890554 | 1889217 | 1853020 |
| **60n15m15t_1** | 1,97E+006 | 1957217 | 1955769 | 1925020 |
| **60n15m15t_2** | 2,02E+006 | 2006722 | 2004638 | 1976790 |
| **60n15m15t_3** | 1,99E+006 | 1973540 | 1971502 | 1940380 |
| **60n15m15t_4** | 1,97E+006 | 1953164 | 1951372 | 1919510 |
| **60n15m20t_0** | 2,63E+006 | 2608796 | 2606871 | 2567320 |
| **60n15m20t_1** | 2,61E+006 | 2596537 | 2593422 | 2553450 |
| **60n15m20t_2** | 2,65E+006 | 2627334 | 2624295 | 2584900 |

*Continua na próxima página*

Tabela C.3 – *Continuação da página anterior*

| **C. Problema** | $TAH$ | $IG + HR$ | $IG + MDL$ | $IG + F\&O$ |
|---|---|---|---|---|
| **60n15m20t_3** | 2,66E+006 | 2641275 | 2638754 | 2598740 |
| **60n15m20t_4** | 2,60E+006 | 2584416 | 2581854 | 2538230 |
| **60n20m10t_0** | 1,70E+006 | 1689890 | 1688357 | 1668840 |
| **60n20m10t_1** | 1,79E+006 | 1775884 | 1775396 | 1753020 |
| **60n20m10t_2** | 1,74E+006 | 1732961 | 1732303 | 1703270 |
| **60n20m10t_3** | 1,75E+006 | 1728466 | 1727401 | 1708320 |
| **60n20m10t_4** | 1,70E+006 | 1689056 | 1687779 | 1667960 |
| **60n20m15t_0** | 2,60E+006 | 2572648 | 2570883 | 2536230 |
| **60n20m15t_1** | 2,65E+006 | 2628090 | 2625910 | 2594570 |
| **60n20m15t_2** | 2,58E+006 | 2551592 | 2549507 | 2514730 |
| **60n20m15t_3** | 2,63E+006 | 2612435 | 2610922 | 2576800 |
| **60n20m15t_4** | 2,65E+006 | 2632830 | 2631085 | 2592850 |
| **60n20m20t_0** | 3,51E+006 | 3491035 | 3487435 | 3442700 |
| **60n20m20t_1** | 3,57E+006 | 3549079 | 3546963 | 3508600 |
| **60n20m20t_2** | 3,46E+006 | 3432960 | 3430447 | 3386040 |
| **60n20m20t_3** | 3,43E+006 | 3404800 | 3402140 | 3356690 |
| **60n20m20t_4** | 3,55E+006 | 3526191 | 3523954 | 3479250 |