



Universidade Federal de Minas Gerais

Programa de Pós-Graduação em Engenharia de Produção

Departamento de Engenharia de Produção / Escola de Engenharia

Relaxação Lagrangeana com Fixação de Variáveis Aplicada ao Problema de Seqüenciamento em uma Máquina

Helton Cristiano Gomes

Orientador: Prof. Dr. Carlos Roberto Venâncio de Carvalho

Belo Horizonte

Agosto, 2008

UNIVERSIDADE FEDERAL DE MINAS GERAIS

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO / ESCOLA DE ENGENHARIA

Relaxação Lagrangeana com Fixação de Variáveis Aplicada ao Problema de Seqüenciamento em uma Máquina

Helton Cristiano Gomes

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais para a obtenção do título de Mestre em Engenharia de Produção.

Área de Concentração: Produção e Logística

Linha de Pesquisa: Otimização de Sistemas Produtivos

Orientador: Prof. Dr. Carlos Roberto Venâncio de Carvalho

Belo Horizonte

Agosto, 2008

*“Dedico este trabalho aos meus pais,
familiares e amigos pelo amor e com-
preensão.”*

Resumo

O presente trabalho trata do Problema de Seqüenciamento em uma Máquina. Este tipo de problema aparece em uma variedade de situações práticas, como é o caso dos Problemas de Planejamento de Operações em máquinas em uma indústria de manufatura. O Problema de Seqüenciamento em uma Máquina consiste em ordenar n *jobs* para serem processados em uma única máquina. Os *jobs* são independentes e a máquina só pode processar um *job* de cada vez. Um caso especial deste tipo de problema surge quando é associado a cada *job* seu respectivo peso e data de chegada, onde o objetivo é a minimização do custo ponderado das datas de início de processamento dos *jobs*. Este problema é de Otimização Combinatória e, como grande parte destes problemas, é de difícil resolução. O modelo de Programação Linear Inteira estudado aqui para representar o problema apresenta variáveis indexadas no tempo.

A formulação com indexação no tempo pode ser utilizada para representar vários tipos de Problemas de Seqüenciamento. A relaxação de problemas modelados com esta característica pode gerar bons limites para o valor da solução do problema original. Esta formulação apresenta, entretanto, uma importante desvantagem, a dimensão do modelo obtido. Modelos representativos de instâncias relativamente pequenas contêm um elevado número de restrições e variáveis.

Visando gerar bons limites para o valor da solução do problema estudado, é utilizada uma Relaxação Lagrangeana, onde o Problema Dual Lagrangeano adotado é otimizado através do Método do Subgradiente. Para reduzir o número de variáveis necessário para a definição do problema é utilizado um procedimento de fixação de variáveis baseado nos custos reduzidos de uma solução Lagrangeana viável. Finalmente, o Problema de Programação Inteira com as variáveis fixadas pela metodologia descrita acima é resolvido por um *software* de otimização.

Palavras-Chave: Seqüenciamento em uma máquina, Relaxação Lagrangeana, Método do Subgradiente, Fixação de Variáveis.

Abstract

This paper deals with the Single-machine Scheduling Problem. This kind of problem arises in several practical situations, such as the problems of planning operations on machines in a manufacturing industry. The Single-machine Scheduling Problem consists in sorting n jobs to be processed on a single machine. The jobs are independent and the machine can only execute one job at a time. A special case of this kind of problem arises when each job is associated with their weight and their release date, where the goal is to minimize weighted cost of time at the beginning of processing the jobs. This problem is a Combinatorial Optimization, and as many of these kinds of problems, it is difficult to be solved. The Integer Linear Programming model studied here to represent the problem, presents time-indexed variables.

The time-indexed formulation can be used to represent different types of Scheduling Problems. The relaxation of modelled problems with this feature can generate good limits for the value of the solution of the original problem. However, this formulation shows an important disadvantage, the dimension of the obtained model. Representative Models of small instances have a large number of constraints and variables.

Aimed at generating good limits on the value of the solution of the studied problem, it was used a Lagrangean Relaxation, where the Dual Lagrangean Problem adopted is maximized through the Subgradient method. To reduce the number of the necessary variables for the definition of the problem, there is a procedure used for fixing variables based on the reduced costs of a feasible Lagrangeana solution. Finally, the Integer Programming Problem with the fixed variables by the methodology described above can be solved by a optimization software.

Keywords: Single-machine scheduling, Lagrangean Relaxation, Subgradient Method, Fixing Variables.

Agradecimentos

Gostaria de agradecer, primeiramente, à Deus pelo dom da vida.

Ao meu PAI.

À minha família, em especial à minha mãe, pelo apoio, compreensão e incentivos constantes, mesmo nos momentos mais difíceis.

À Flávia, principal motivação para tudo na minha vida. E à sua família pela ajuda e pelo carinho que têm por mim.

Ao professor Carlos pela orientação e pelas preciosas dicas, imprescindíveis para a realização deste trabalho.

À UFMG pela oportunidade e por me dar condições para vencer mais essa etapa.

Aos professores do DEP-UFMG pelos conselhos e ensinamentos que de uma forma ou de outra contribuíram para a realização deste trabalho.

Aos meus velhos amigos pelo companheirismo e força e, aos amigos que fiz durante estes 2 anos de UFMG, principalmente ao Magno e ao Moreira, pessoas muito especiais.

À CAPES pelo investimento em minha formação.

Sumário

| | |
|---|-----------|
| Lista de Figuras | ix |
| Lista de Tabelas | x |
| Lista de Siglas | 1 |
| 1 Introdução | 1 |
| 1.1 Objetivos | 2 |
| 1.2 Estrutura do Trabalho | 3 |
| 2 Revisão Bibliográfica | 4 |
| 2.1 O Problema de Seqüenciamento | 4 |
| 2.2 O Problema de Seqüenciamento em uma Máquina | 6 |
| 2.3 O Problema Estudado - SMSP | 7 |
| 2.3.1 Diferentes Formulações para o SMSP | 8 |
| 2.4 Métodos de Resolução | 10 |
| 2.5 Relaxação Lagrangeana | 11 |
| 2.5.1 Definição do Problema Lagrangeano | 12 |
| 2.5.2 Método do Subgradiente | 15 |
| 3 Uma Relaxação Lagrangeana para o SMSP Formulado com Indexação no Tempo | 17 |
| 3.1 SMSP Formulado com Indexação no tempo | 17 |

| | | |
|----------|--|-----------|
| 3.2 | Relaxação Lagrangeana Aplicada ao SMSP Formulado com Indexação no Tempo | 19 |
| 3.2.1 | Heurística Utilizada para a Obtenção de Limites Superiores . . . | 21 |
| 3.2.1.1 | Primeira Fase | 21 |
| 3.2.1.2 | Segunda Fase | 23 |
| 3.2.2 | Resolução do Problema Lagrangeano | 24 |
| 3.2.2.1 | Algoritmo de Programação Dinâmica para a Resolução do Problema Lagrangeano | 26 |
| 3.2.3 | Otimização via Método do Subgradiente | 27 |
| 3.3 | Fixação de Variáveis | 29 |
| 3.4 | Comparação com a Relaxação Lagrangeana Proposta por Avella <i>et al.</i> [5] | 30 |
| 4 | Resultados e Análise | 32 |
| 4.1 | Geração dos Problemas-Teste | 32 |
| 4.2 | Apresentação e Análise dos Resultados | 32 |
| 5 | Conclusão e Trabalhos Futuros | 39 |
| | Referências Bibliográficas | 41 |

Lista de Figuras

| | | |
|-----|--|----|
| 2.1 | Esquema para o Problema de Seqüenciamento em uma Máquina | 7 |
| 3.1 | Estrutura de vizinhança com troca na ordem de processamento do <i>job 2</i> | 23 |
| 3.2 | Grafo para o exemplo. Fonte: Adaptado de van den Akker <i>et al.</i> [2] . . . | 25 |
| 3.3 | Caminho no Grafo. Fonte: Próprio autor | 26 |

Lista de Tabelas

| | | |
|-----|--|----|
| 3.1 | Heurística Construtiva Gulosa | 22 |
| 3.2 | Dados do Exemplo | 22 |
| 4.1 | Resultados obtidos pela Relaxação Lagrangeana proposta para os problemas- teste com as mesmas dimensões utilizados por Avella <i>et al.</i> [5] | 33 |
| 4.2 | Características dos problema-teste | 34 |
| 4.3 | Resultados computacionais | 35 |
| 4.4 | Comparação entre os tempos de resolução sem e com fixação de variáveis | 36 |
| 4.5 | Percentual dos Limites Inferiores e Superiores em Relação ao Ótimo | 37 |
| 4.6 | Tempo total da metodologia e redução no tempo comparado ao CPLEX | 38 |

Capítulo 1

Introdução

Determinar a melhor Programação da Produção é um objetivo a ser alcançado por qualquer empresa, em vista da redução de custos de produção e/ou estocagem que ela pode proporcionar. Considerando que cada empresa possui um cenário diferente, a utilização de sistemas computacionais genéricos, nem sempre compatíveis com a realidade da empresa, se torna inadequada. Portanto, o desenvolvimento de sistemas específicos voltados para a realidade da empresa, utilizando técnicas adequadas, se torna útil para determinar a melhor Programação da Produção.

Em meio aos Problemas de Programação da Produção, os de Seqüenciamento têm recebido atenção especial da literatura, dentre os quais encontram-se os Problemas de Seqüenciamento em uma Máquina onde são associados pesos aos *jobs* e, estes apresentam datas de chegada. Um objetivo, bastante estudado para este tipo de problema, é o de minimizar a soma ponderada dos custos associados às datas de início de processamento dos *jobs*. Na Teoria de Seqüenciamento (*Scheduling Theory*), este é um problema básico, conhecido como *Single-Machine Scheduling Problem* - SMSP. Vários Problemas de Seqüenciamento, mais complexos e realistas, podem ser decompostos em problemas mais simples como o SMSP.

O SMSP é um problema de Otimização Combinatória e, como grande parte destes problemas, é de difícil resolução. O SMSP foi demonstrado ser fortemente *NP*-Difícil, ou seja, a resolução deste problema requer um elevado esforço computacional, tanto em relação ao tempo de resolução quanto em relação à memória necessária.

Entre as diferentes formulações encontradas na literatura, que podem ser utilizadas para modelar matematicamente o SMSP, destaca-se aqui a que utiliza variáveis indexadas no tempo. Uma variedade de problemas envolvendo seqüenciamento podem

ser formulados através de modelos de Programação Inteira os quais empregam variáveis indexadas por (j, t) , onde j denota o *job* a ser processado e t um período de tempo. Uma das desvantagens associadas a este tipo de formulação, quando comparada às outras formulações encontradas, é o elevado número de variáveis e restrições envolvidas.

Esta dissertação utiliza o SMSP formulado com indexação no tempo. Utiliza uma Relaxação Lagrangeana para gerar bons limites para o valor da solução do problema em questão, onde o Problema Dual Lagrangeano adotado é otimizado pelo Método do Subgradiente. Para fixar variáveis, reduzindo, dessa forma, o elevado número de variáveis presentes no modelo, é utilizado um procedimento baseado nos custos reduzidos de uma solução Lagrangeana viável. A fixação das variáveis é utilizada como dado de entrada para o Problema de Programação Inteira, visando a uma redução no esforço computacional para resolvê-lo. Para a validação desta metodologia foram realizados testes com instâncias fictícias de várias dimensões, geradas como proposto na literatura. É apresentada uma comparação entre os resultados obtidos pela metodologia proposta e a utilização, somente, de um *software* comercial de otimização.

Para a elaboração da metodologia foi realizada, primeiramente, uma revisão bibliográfica para que tivéssemos acesso às várias abordagens a respeito do problema estudado. O problema, portanto, foi definido e caracterizado de acordo com a visão dos autores pesquisados. Foi obtido, também, um amplo conhecimento acerca das ferramentas e técnicas utilizadas na metodologia.

1.1 Objetivos

O objetivo geral desta dissertação é desenvolver uma metodologia baseada na Relaxação Lagrangeana e fixação de variáveis visando a redução do esforço computacional necessário para a resolução do Problema de Seqüenciamento em uma Máquina com Data de Chegada para os *Jobs*, formulado com variáveis indexadas no tempo.

Os objetivos específicos desta dissertação são os seguintes:

- Fazer um levantamento bibliográfico sobre o SMSP e problemas afins, bem como de diversas metodologias utilizadas para resolvê-los;
- Definir um modelo matemático, dentre os encontrados na literatura, que melhor represente o SMSP para nossos fins;

- Utilizar uma Relaxação Lagrangeana para gerar limites para o SMSP, que permita a caracterização de variáveis de decisão sub-ótimas;
- Propor, no contexto da Relaxação Lagrangeana, um procedimento para a resolução do SMSP;
- Implementar um sistema computacional em Linguagem C++ baseado na metodologia proposta para resolver o problema;
- Criar um conjunto de problemas-teste do problema em questão;
- Validar o sistema computacional desenvolvido exclusivamente comparando suas soluções com aquelas geradas por um *software* de otimização.

1.2 Estrutura do Trabalho

O restante do texto desta dissertação está estruturado da seguinte forma: No Capítulo 2 é apresentada uma revisão bibliográfica relacionada ao problema estudado, bem como sua definição e modelos matemáticos propostos para representá-lo. São apresentados, também, métodos utilizados para a resolução de problemas como o SMSP, com maior ênfase para o método adotado. São apresentados alguns dos trabalhos mais relevantes da literatura sobre o problema estudado e a técnica de resolução adotada.

O Capítulo 3 mostra uma descrição detalhada do modelo utilizado e da metodologia desenvolvida. O Capítulo 4 descreve como os problemas-teste foram gerados, apresenta e analisa os resultados computacionais obtidos através da metodologia proposta e apresenta, também, uma comparação entre o tempo computacional gasto pela metodologia e o tempo gasto utilizando somente um *software* de otimização. O Capítulo 5 conclui o trabalho e aponta sugestões para trabalhos futuros relacionados ao problema estudado.

Capítulo 2

Revisão Bibliográfica

2.1 O Problema de Seqüenciamento

De acordo com Baker [6], um Problema de Seqüenciamento (*Scheduling Problem*) consiste na alocação de recursos ao longo do tempo para executar um conjunto de tarefas. Recursos são bens ou serviços - cuja disponibilidade é limitada ou não - e podem ser classificados em recuperáveis e não-recuperáveis. Os exemplos mais comuns de recursos são: máquinas, matérias-primas e mão-de-obra. Tarefas ou operações são trabalhos elementares cuja realização necessita de um certo número de unidades de tempo e/ou recursos. Ainda de acordo com o autor, para classificar a maioria dos Problemas de Seqüenciamento é necessário caracterizar a configuração de seus dados (recursos e tarefas).

Quando não são conhecidos *a priori* todos os seus dados, os Problemas de Seqüenciamento são classificados como Dinâmicos e, quando todos os dados são conhecidos com antecedência, como Estáticos. Os Problemas de Seqüenciamento Estáticos podem, ainda, ser classificados como Estocásticos, quando alguns de seus dados forem expressos por uma distribuição de probabilidade e, como Determinísticos, quando não há distribuição de probabilidade associada a seus dados (Błazewicz *et al.* [8]).

Um Problema de Seqüenciamento pode conter um ou mais tipos de recursos e, na maioria dos problemas, estes recursos são máquinas que podem ser classificadas de duas formas:

- Paralelas: um conjunto de máquinas pode executar o mesmo tipo de tarefa. As máquinas paralelas podem ser:

- Idênticas: todas as máquinas possuem a mesma velocidade;
- Diferentes: cada máquina possui sua velocidade.
- Especializadas: cada máquina executa um único tipo de tarefa.

Em relação às tarefas, estas podem ser *preemptivas* (quando sua execução pode ser interrompida e ser reiniciada algum tempo mais tarde) ou *não preemptivas* (uma vez iniciada, sua execução não pode ser interrompida).

Alguns Problemas de Seqüenciamento podem apresentar, também, tempo de preparação de máquina. Esse tempo, também denominado tempo de *setup*, inclui o tempo de preparar a máquina, o processo ou a oficina para a execução de determinados tipos de tarefas. Isto inclui o tempo para obtenção das ferramentas, posicionamento dos materiais a serem usados no trabalho, processos de limpeza, preparação e ajuste das ferramentas, inspeção de materiais, etc. Os tempos de preparação de máquina podem ser caracterizados de duas formas: independentes da seqüência de produção, os quais dependem somente da tarefa a ser processada, podendo, então, ser adicionado ao tempo de processamento da tarefa e, dependentes da seqüência de produção, os quais dependem tanto da tarefa a ser processada quanto daquela que foi processada imediatamente antes na mesma máquina (Allahverdi *et al.* [3]).

Dentre os Problemas de Seqüenciamento, Baker [6] e Błazewicz *et al.* [8] destacam os Estáticos Determinísticos, com máquinas especializadas, tais como o Problema com uma Máquina, o Problema tipo *Job Shop* e o Problema tipo *Flow Shop*.

No Problema com uma Máquina, um conjunto de tarefas deve ser executado em uma única máquina especializada. Em um Problema de *Job Shop*, um conjunto de máquinas especializadas executa um determinado número de operações sobre um conjunto de *jobs*, onde cada *job* possui sua própria seqüência tecnológica de produção. O Problema de *Flow Shop* é um caso particular do Problema de *Job Shop* onde todos os *jobs* possuem a mesma seqüência tecnológica de produção (Baker [6] e Błazewicz *et al.* [8]).

Em um Problema de Seqüenciamento com Máquinas Paralelas, um determinado número de tarefas pode ser realizado por qualquer máquina pertencente ao conjunto de máquinas disponíveis.

Na Teoria de Seqüenciamento (Graham *et al.* [16]), um *job* é um conjunto ordenado de tarefas e representa a seqüência tecnológica de fabricação de um produto.

Uma operação ou tarefa, quando se trata de problemas com uma máquina, se confunde com o conceito de *job*.

Devido à grande variedade de Problemas de Seqüenciamento, tornou-se necessária a utilização de uma notação sistemática para que tais problemas sejam classificados. A notação mais utilizada, proposta por Graham *et al.* [16] e Błazewicz *et al.* [9], facilitou muito a apresentação dos Problemas de Seqüenciamento. Tal notação é composta por três campos $\alpha|\beta|\gamma$, onde cada um tem o seguinte significado: o campo α descreve o tipo de problema e o número de máquinas presentes nele; o campo β descreve as características das tarefas e dos recursos; o campo γ denota o critério de otimização. Como exemplo, tem-se $J3|n = 5|C_{max}$, onde $J3$ representa um problema do tipo *Job Shop* com três máquinas, $n = 5$ indica que o problema contém cinco *jobs* e o critério de otimização é de minimizar C_{max} (maior data de término de processamento de um *job*).

Vários critérios de otimização são encontrados na literatura, dentre os quais destacam-se os seguintes: minimização da maior data de término de processamento ($\text{Min } C_{max}$), minimização do custo ponderado de início de processamento dos *jobs* ($\text{Min } \sum_j w_j t_j$), minimização do tempo de fluxo médio ($\text{Min } \bar{F}$), minimização do atraso médio ($\text{Min } \bar{D}$) e minimização do tempo de fluxo máximo ($\text{Min } F_{max}$).

Dentre os vários Problemas de Seqüenciamento, esta dissertação tem como foco o Problema de Seqüenciamento em uma Máquina.

2.2 O Problema de Seqüenciamento em uma Máquina

O Problema de Seqüenciamento em uma Máquina consiste em determinar uma seqüência de produção, ordenando-se um conjunto de *jobs* para serem executados em uma única máquina especializada, com base em algum critério pré-estabelecido. O Problema de Seqüenciamento em uma Máquina surge em uma variedade de situações práticas, como é o caso dos Problemas de Planejamento de Operações em máquinas em uma indústria de manufatura.

Além dos sistemas produtivos que operam com uma única máquina, os que as estações de trabalho são dependentes umas das outras também podem ser representados por este tipo de problema.

Segundo Baker [6], o Problema básico de Seqüenciamento com uma máquina é caracterizado pelas seguintes condições:

- um conjunto de n jobs deve ser processado por uma única máquina;
- o processamento de cada job indisponibiliza a máquina por um intervalo de tempo conhecido *a priori*;
- os tempos de preparação da máquina são independentes da seqüência e podem ser incorporados ao tempo de processamento;
- a máquina está continuamente disponível e nunca é mantida inativa enquanto houver algum job esperando;
- uma vez iniciado, o processamento de um job é realizado até sua completa fabricação, não sendo admitidas interrupções.

Uma representação esquemática para o Problema de Seqüenciamento em uma Máquina pode ser visto na Figura 2.1. Neste exemplo, os $jobs$ de 1 a n são processados em uma única máquina, dando origem a n produtos.

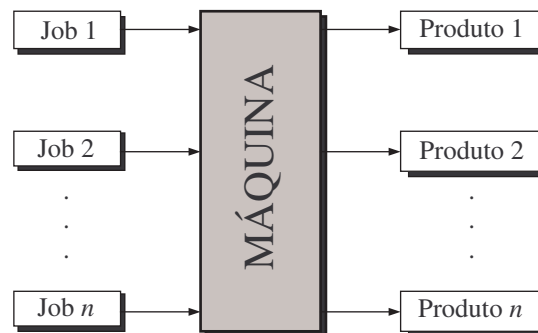


Figura 2.1: Esquema para o Problema de Seqüenciamento em uma Máquina

O Problema de Seqüenciamento em uma Máquina consiste então em seqüenciar n jobs para serem processados em uma única máquina especializada, que só pode processar um único job por vez. Os $jobs$ são independentes uns dos outros e estão disponíveis para processamento no instante inicial. Cada job j , $j = 1, 2, \dots, n$, possui um tempo de processamento p_j e a interrupção do processamento (*preempção*) de qualquer job não é permitida. O critério de otimização empregado varia de acordo com os objetivos do caso estudado.

2.3 O Problema Estudado - SMSP

Um caso especial para o Problema de Seqüenciamento em uma Máquina surge quando são associados pesos (w_j) aos $jobs$ e estes apresentam datas de chegada (r_j)

distintas. Isto implica que o processamento de um *job* só poderá ser iniciado a partir de sua data de chegada. Na Teoria de Seqüenciamento (*Scheduling Theory*), Graham *et al.* [16], este problema é conhecido como *Single-Machine Scheduling Problem* - SMSP e, é denotado por $1|r_j|\Sigma w_j C_j$.

Trata-se de um Problema de Otimização Combinatória e, como grande parte destes problemas, é de difícil resolução (Garey e Johnson [14]). Diferentes trabalhos, como por exemplo, os de Lenstra *et al.* [24] e Crama e Spieksma [11], relatam que o SMSP é fortemente *NP*-Difícil.

Diferentes formulações para o Problema de Seqüenciamento em uma Máquina são apresentados na literatura, onde destacam-se aqui os modelos de Programação Linear Inteira Mista propostos por Manne [27] e Wagner [33] e o modelo de Programação Linear Inteira, com variáveis indexadas no tempo, proposto por Sousa e Wolsey [31].

2.3.1 Diferentes Formulações para o SMSP

No modelo proposto por Manne [27], utiliza-se um par de restrições dicotômicas para controlar a ordem relativa dos *jobs* dentro da seqüência de produção. O modelo de Manne [27] procura satisfazer duas restrições básicas: seqüência e não interferência.

A restrição de seqüência garante que o processamento do *job* j seja iniciado no mínimo p_i unidades de tempo, após o início do processamento do *job* i , se o *job* j for processado após o *job* i . A restrição de não interferência procura garantir que, se um *job* j é processado imediatamente após um *job* i em uma mesma máquina, o *job* i deve preceder o *job* j de um tempo suficiente, de modo que i possa ser completado antes de j começar.

Um modelo para o SMSP, baseado em Manne [27], é descrito a seguir.

Os parâmetros de entrada do modelo são:

- J : conjunto de *jobs* a serem processados, $J = \{1, \dots, n\}$;
- p_j : tempo de processamento do *job* j ;
- r_j : data de chegada do *job* j ;
- w_j : peso associado ao *job* j ;
- M : valor positivo muito grande.

E as variáveis de decisão são:

- t_j : data de início do processamento do *job* j ;
- y_{ij} : variável que determina a seqüência de produção.

$$y_{ij} = \begin{cases} 1, & \text{se o } \textit{job } j \text{ é processado após o } \textit{job } i; \\ 0, & \text{caso contrário.} \end{cases}$$

Desta forma tem-se o seguinte modelo de Programação Linear Inteira Mista:

$$\min Z = \sum_{j=1}^n w_j t_j \quad (2.1)$$

$$\text{s. a:} \quad t_j \geq t_i + p_i - M(1 - y_{ij}) \quad \forall i, j = 1, 2, \dots, n \text{ e } i \neq j \quad (2.2)$$

$$y_{ij} + y_{ji} = 1 \quad \forall i, j = 1, 2, \dots, n \text{ e } i \neq j \quad (2.3)$$

$$t_j \geq r_j \quad \forall j = 1, 2, \dots, n \quad (2.4)$$

$$y_{ij} \in \{0, 1\} \quad \forall i, j = 1, \dots, n \quad (2.5)$$

A função objetivo (2.1) consiste em minimizar a soma ponderada das datas de início de processamento dos *jobs*. As restrições (2.2) definem a seqüência de processamento dos *jobs* sobre a máquina utilizada. As restrições (2.3) garantem que se o *job* j é processado depois do *job* i ($y_{ij} = 1$) o valor de y_{ji} será 0, e vice-versa. As restrições (2.4) garantem que o início do processamento de um *job* só ocorra quando ele estiver disponível. As restrições (2.5) definem, por sua vez, o domínio das variáveis y_{ij} .

Já o modelo de Wagner [33] utiliza o problema clássico de alocação de tarefas para assinalar *jobs* a posições na seqüência de produção.

Para tratar o Problema de Seqüenciamento como um Problema de Alocação, segundo a proposta de Wagner [33], parte-se de um conjunto K de posições k possíveis dentro de uma seqüência de produção, onde cada *job* $j \in J$ deve ser alocado.

O SMSP não pode ser modelado como um Problema de Programação Linear Inteira Mista, baseado em Wagner [33], devido à impossibilidade de se representar linearmente sua função objetivo. Como a função objetivo consiste em minimizar a soma ponderada das datas de início de processamento dos *jobs*, para relacionar a variável s_k (data de início do processamento da posição k na seqüência de produção) com o peso (w_j) do *job* j alocado na posição k , ela deve ser multiplicada pela variável binária z_{jk} (variável que determina se o *job* j foi alocado na posição k da seqüência de produção).

Vários modelos baseados em Wagner [33] e Manne [27] são encontrados na literatura para os diversos Problemas de Programação da Produção.

Como é de interesse desta dissertação gerar bons limites para o valor da solução do SMSP, o modelo adotado foi o com variáveis indexadas no tempo proposto por Sousa e Wolsey [31], que será apresentado na seção 3.1. Tal modelo foi adotado devido ao fato de gerar limites de Relaxação Linear mais fortes, além de ser um bom guia para algoritmos de aproximação, como descrevem van den Akker *et al.* [2] e Sousa e Wolsey [31].

O modelo matemático para o SMSP, com a indexação de variáveis no tempo, foi originalmente proposto por Sousa e Wolsey [31], baseado no problema geral de escolha de projetos a serem realizados ao longo de um horizonte de tempo, sujeito a restrições de recursos. Tal formulação foi, também, utilizada por van den Akker *et al.* [1] [2] e Avella *et al.* [5].

Segundo van den Akker *et al.* [2], para os Problemas de Seqüenciamento, a formulação com a indexação no tempo é utilizada tanto na busca de uma solução ótima quanto de uma solução aproximada. Mas, como descrevem Sousa e Wolsey [31], o número elevado de variáveis e restrições necessário para modelar um problema através da indexação das variáveis no tempo é, geralmente, uma considerável desvantagem deste tipo de formulação.

2.4 Métodos de Resolução

Minoux [28] e Lasdon [22] descrevem os principais métodos de resolução para problemas com um grande número de variáveis e/ou restrições, dentre os quais destacam os métodos de Decomposição de Dantzig-Wolf e de Benders, o de Geração de Colunas e o de Relaxação Lagrangeana. Segundo Minoux [28], tais métodos permitem a redução de problemas de grande escala em uma seqüência de problemas de pequena dimensão.

O método de Decomposição de Dantzig-Wolfe aplica-se a Problemas de Programação Linear de grandes dimensões, em que a matriz dos coeficientes tecnológicos tem estrutura bloco-angular. O método procura aproveitar esta estrutura especial da matriz para reduzir a dimensão da base, criando um problema equivalente com grande redução no número de restrições (e portanto na dimensão da base). No entanto, a sua característica mais relevante é um esquema de geração implícita de colunas não-básicas

no problema equivalente, indispensável para tornar possível a resolução.

O método de Decomposição de Benders inicia dividindo as variáveis de um problema em dois vetores, x e y . Fixando valores para y , obtém-se um sub-problema somente com as variáveis x . Se a solução deste sub-problema revela que os valores fixados para y são inviáveis, a solução do dual do sub-problema é utilizado para identificar novos valores para y . O elemento central da Decomposição de Benders é a derivação dos cortes de Benders (restrições adicionais que são incorporadas ao sub-problema visando restringi-lo ainda mais). Cortes clássicos de Benders são formulados resolvendo o dual do sub-problema.

De acordo com Lorena e Senne [25], o método de Geração de Colunas pode ser aplicado a Problemas Lineares de grandes dimensões, no caso de não se dispor de todas as colunas *a priori*, ou quando se pretende resolver um problema utilizando a Decomposição de Dantzig-Wolfe, onde as colunas correspondem aos pontos extremos do conjunto convexo de soluções factíveis do problema. Entretanto, a aplicação direta do método de Geração de Colunas freqüentemente produz um número muito grande de colunas que não são relevantes para a solução final, comprometendo assim a convergência para a solução do problema. Abordagens baseadas no Método de Geração de Colunas têm aparecido em um grande número de trabalhos recentes como alternativa aos Métodos não lineares baseados em Relaxação Lagrangeana para resolver Problemas de Programação Linear Inteira de grande porte.

Um método de Relaxação Lagrangeana pode ser utilizado quando o elevado número de restrições é um dificultador para a resolução de um problema. A estratégia utilizada na Relaxação Lagrangeana é relaxar algumas destas restrições e, então, resolver o problema resultante. A resolução deste problema resultante, denominado Problema Lagrangeano, provê limites para o valor da solução ótima dos problemas.

Dentre os métodos descritos acima, utilizou-se, nesta dissertação, a Relaxação Lagrangeana.

2.5 Relaxação Lagrangeana

Uma das observações mais úteis dos anos 70 é a de que muitos problemas difíceis podem ser vistos como problemas fáceis, complicados por um certo número de restrições. Um procedimento muito utilizado para lidar com restrições difíceis consiste em retirá-las do conjunto de restrições do problema e acrescentá-las à função

objetivo utilizando um vetor de multiplicadores, denominados multiplicadores de Lagrange. Através deste procedimento, chamado Relaxação Lagrangeana, pode-se obter um problema de fácil resolução, denominado Problema Lagrangeano, cujo valor da solução é um limitante para o valor da solução do problema original. Em problemas de minimização, a resolução do Problema Lagrangeano gera limites inferiores e, para problemas de maximização, limites superiores para o valor da solução do problema original. Com base no limite gerado pela Relaxação Lagrangeana é possível estimar quão próxima está a solução disponível da solução ótima. Os limites gerados pela Relaxação Lagrangeana junto com valores obtidos para soluções viáveis do problema original, calculados, por exemplo, através de heurísticas, criam intervalos (*gaps*) para o valor da solução ótima do problema original.

A Relaxação Lagrangeana foi e tem sido aplicada na resolução de um amplo número de Problemas de Otimização Combinatória. Geoffrion [15] e Guignard [17] relatam a aplicação da Relaxação Lagrangeana na resolução de vários problemas como, por exemplo, o do Caixeiro Viajante, o de Localização, o de Recobrimento e Particionamento de Conjuntos, o de Atribuição Generalizado e o de Roteamento. Held e Karp [19], que utilizaram um método Lagrangeano para resolver o Problema do Caixeiro Viajante, foram precursores no uso deste método. Geoffrion [15] deu ao mesmo o nome de Relaxação Lagrangeana.

O procedimento para a obtenção do Problema Lagrangeano, para um Problema de Programação Inteira, é apresentado na seção 2.5.1 a seguir.

2.5.1 Definição do Problema Lagrangeano

Esta seção apresenta de forma sucinta a Relaxação Lagrangeana, suficiente para o entendimento do trabalho realizado nesta dissertação. Para melhor compreensão, veja a literatura disponível como, por exemplo, Fisher [12], Geoffrion [15], Guignard [17], Lasdon [22] e Minoux [28].

Considere o seguinte Problema de Programação Linear Inteira (P):

$$\text{minimizar } Z = \sum_{j=1}^n c_j x_j \quad (2.6)$$

$$\text{sujeito a: } \sum_{j=1}^n a_{ij} x_j \leq b_i \quad \forall i = 1, 2, \dots, m \quad (P) \quad (2.7)$$

$$x_j \in \mathbb{Z}_+^n \quad \forall j = 1, 2, \dots, n \quad (2.8)$$

Suponha que (P) é um problema difícil de se resolver devido ao conjunto de restrições (2.7), de forma que se esse conjunto for removido, o problema resultante se torna de fácil resolução. Isto pode ser feito associando um multiplicador de Lagrange $\lambda_i \geq 0$ a cada uma das m restrições do tipo (2.7), retirando-as do conjunto de restrições e acrescentando-as à função objetivo (2.6), por meio de seu produto pelo vetor de multiplicadores. Dessa forma obtemos a seguinte Problema Lagrangeano (PRL):

$$\text{minimizar } Z_\lambda = \sum_{j=1}^n c_j x_j + \sum_{i=1}^m \lambda_i \left(\sum_{j=1}^n a_{ij} x_j - b_i \right) \quad (PRL) \quad (2.9)$$

$$\text{sujeito a: } x_j \in \mathbb{Z}_+^n \quad \forall j = 1, 2, \dots, n \quad (2.10)$$

Simplificando, temos

$$\text{minimizar } Z_\lambda = \sum_{j=1}^n (c_j + \sum_{i=1}^m \lambda_i a_{ij}) x_j - \sum_{i=1}^m \lambda_i b_i \quad (PRL) \quad (2.11)$$

$$\text{sujeito a: } x_j \in \mathbb{Z}_+^n \quad \forall j = 1, 2, \dots, n \quad (2.12)$$

onde $Z_\lambda = \sum_{j=1}^n (c_j + \sum_{i=1}^m \lambda_i a_{ij}) x_j - \sum_{i=1}^m \lambda_i b_i$ é chamada Função Lagrangeana.

Como no caso descrito acima as restrições relaxadas são do tipo menor ou igual, os multiplicadores de Lagrange associados a estas são, então, definidos como valores reais não-negativos ($\lambda_i \geq 0, i = 1, 2, \dots, m$). Caso as restrições fossem de igualdade, os multiplicadores de Lagrange seriam definidos como reais e irrestritos em sinal e, como reais não-positivos em caso de restrições do tipo maior ou igual.

Considerando $x^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ uma solução viável de (P), observa-se que x^* é também viável para o Problema Lagrangeano e que $\sum_{j=1}^n a_{ij} x_j^* - b_i \leq 0, \forall i = 1, 2, \dots, m$. Assim, para qualquer valor para $\lambda_i \geq 0, \forall i = 1, 2, \dots, m$, temos que

$$Z_\lambda \leq \sum_{j=1}^n c_j x_j^* + \sum_{i=1}^m \lambda_i \left(\sum_{j=1}^n a_{ij} x_j^* - b_i \right)$$

Logo

$$Z_\lambda \leq \sum_{j=1}^n c_j x_j^* + \sum_{i=1}^m \lambda_i \left(\sum_{j=1}^n a_{ij} x_j^* - b_i \right) \leq \sum_{j=1}^n c_j x_j^* = Z$$

Portanto, o valor da Função Lagrangeana (Z_λ) define um limite inferior para o valor da solução do problema original (Z).

Note que para cada vetor de multiplicadores λ há um problema (PRL) associado, o qual o valor da sua solução (Z_λ) fornece um limitante inferior para o valor da solução do problema (P). A importância da Relaxação Lagrangeana nesse sentido é determinar o vetor de multiplicadores de Lagrange que gere um limitante inferior mais próximo possível do valor da solução de (P).

Para determinar o vetor de multiplicadores de Lagrange que gera o limite inferior mais próximo do valor da solução de (P), é necessário resolver o Problema Dual Lagrangeano (PDL), que por definição, é dado por:

$$Z_d = \max_{\lambda \geq 0} \{ \min Z_\lambda \} \quad (PDL)$$

O PDL é um problema no espaço dual dos multiplicadores de Lagrange, considerando que PRL é um problema em x .

A literatura apresenta alguns métodos para a resolução do Problema Dual Lagrangeano, dentre os quais destacam-se o Método do Subgradiente proposto por Held *et al.* [20], o Método de Feixes proposto por Lemarechal [23] e o Método do Volume proposto por Barahona e Anibal [7].

O Método do Subgradiente proposto por Held *et al.* [20], aqui tratado somente como Método do Subgradiente, consiste na resolução iterativa de Problemas Lagrangeanos, onde os multiplicadores de Lagrange são atualizados buscando a melhora do limite Lagrangeano. O Método de Feixes e o Método do Volume são extensões do Método do Subgradiente. No Método de Feixes, proposto por Lemarechal [23], são

utilizadas informações passadas para prover uma melhor aproximação da Função Lagrangeana. O Método do Volume, proposto por Barahona e Anibal [7], quando lidando com variáveis duais, é semelhante ao Método do Subgradiente, mas apresenta uma nova forma de lidar com as variáveis primais. Tal método tenta estimar o volume abaixo das facetas ativas em uma solução dual, onde este volume provê a solução primal.

Por funcionar bem para muitos problemas práticos, como descreve Fisher [12], o método utilizado nesta dissertação para a resolução do Problema Dual Lagrangeano foi baseado no Método do Subgradiente proposto por Held *et al.* [20].

2.5.2 Método do Subgradiente

Para a descrição do Método do Subgradiente, considera-se, nesta seção, o Problema de Programação Inteira descrito na seção 2.5.1, bem como seus relativos Problemas Lagrangeano e Dual Lagrangeano.

O Método do Subgradiente, proposto por Held *et al.* [20], é um método iterativo no qual, em uma iteração k , dado o corrente vetor de multiplicadores de Lagrange λ^k , um passo é dado em direção a um subgradiente da Função Lagrangeana.

Partindo de um vetor de multiplicadores de Lagrange inicial arbitrário (λ^0), os multiplicadores, a cada iteração, são atualizados pela expressão:

$$\lambda_i^{k+1} \equiv \max\{0, \lambda_i^k + \Theta^k g_i^k\} \quad \forall i = 1, 2, \dots, m, \quad (2.13)$$

onde Θ^k é o tamanho do passo (escalar positivo) e g_i^k , $i = 1, 2, \dots, m$, são os subgradientes da Função Lagrangeana.

Os subgradientes, na k -ésima iteração, são calculados da seguinte forma:

$$g_i^k = \sum_{j=1}^n a_{ij} x_j^k - b_i \quad \forall i = 1, 2, \dots, m$$

onde $x_j^k = [x_1^k, x_2^k, \dots, x_n^k]^T$ é a solução do Problema Lagrangeano na iteração k .

Segundo Florentino [13], a dificuldade dos métodos que utilizam (2.13) decorre do fato de que o subgradiente g^k nem sempre é uma direção de subida, o que faz com

que a taxa de convergência de tais algoritmos esteja estritamente ligada à determinação do tamanho do passo na direção do subgradiente g^k .

De acordo com Held *et al.* [20], alguns resultados teóricos relataram que uma escolha apropriada do tamanho do passo é importante para a convergência do método. As propriedades de convergência do Método do Subgradiente são descritas por este autor e o resultado teórico fundamental com relação ao passo é que $Z_\lambda \rightarrow Z_d$ se: $\Theta^k \rightarrow 0$ e $\sum_{i=0}^k \Theta_i \rightarrow \infty$ quando $k \rightarrow \infty$.

Dentre as propostas encontradas na literatura que relatam a forma como é calculado o tamanho do passo, com o objetivo de tentar obter uma taxa de convergência mais rápida, a mais utilizada na prática é:

$$\Theta^k = \alpha \frac{Z_{ub} - Z_\lambda}{\|g^k\|^2}$$

onde α é um parâmetro com valor tal que $0 < \alpha \leq 2$, Z_{ub} é um limite superior para o valor da solução de (P), Z_λ é o valor da atual solução de (PRL) e $\|g^k\|^2 = \sum_{i=1}^m (g_i^k)^2$.

Lucena [26] afirma que uma dada heurística deve estar disponível para gerar soluções viáveis para o Problema de Programação Linear Inteira (P). O objetivo desta heurística é definir bons limites superiores (Z_{ub}) para serem utilizados no Método do Subgradiente.

Capítulo 3

Uma Relaxação Lagrangeana para o SMSP Formulado com Indexação no Tempo

3.1 SMSP Formulado com Indexação no tempo

Em um modelo de Programação Linear Inteira com variáveis indexadas no tempo, o horizonte de planejamento T é dividido em intervalos de tempo unitários t , onde cada intervalo de tempo t tem seu início na data $t - 1$ e, seu término, na data t . Segundo van den Akker *et al.* [2], a formulação com indexação no tempo é poderosa o suficiente para tratar vários tipos de Problemas de Seqüenciamento, com diferentes funções objetivo e restrições bastante variadas.

O modelo com indexação no tempo adotado para representar o SMSP, proposto por Sousa e Wolsey [31], é descrito a seguir.

Os parâmetros de entrada do modelo são:

- J : conjunto de *jobs* a serem processados, $J = \{1, \dots, n\}$;
- p_j : tempo de processamento do *job* j ;
- r_j : data de chegada do *job* j ;
- w_j : peso associado ao *job* j ;
- c_{jt} : custo por iniciar o processamento do *job* j no início do intervalo de tempo t ,

$$c_{jt} = w_j(t - 1)$$

As variáveis de decisão são:

- x_{jt} : variável que determina em qual intervalo de tempo t o processamento do *job* j vai ser iniciado.

$$x_{jt} = \begin{cases} 1, & \text{se o } job \ j \text{ começa a ser processado no início do intervalo de tempo } t; \\ 0, & \text{caso contrário.} \end{cases}$$

Desta forma tem-se o seguinte modelo de Programação Linear Inteira:

$$\min Z = \sum_{j=1}^n \sum_{t=r_j+1}^{T-p_j+1} c_{jt} x_{jt} \quad (3.1)$$

$$\text{s. a:} \quad \sum_{t=r_j+1}^{T-p_j+1} x_{jt} = 1 \quad \forall j = 1, 2, \dots, n \quad (3.2)$$

$$\sum_{j=1}^n \sum_{s=\gamma(j,t)}^t x_{js} \leq 1 \quad \forall t = 1, \dots, T \quad (3.3)$$

$$x_{jt} \in \{0, 1\} \quad \forall j = 1, \dots, n; \forall t = r_j + 1, \dots, T - p_j + 1 \quad (3.4)$$

onde, para qualquer $j \in J$ e $t \in [r_j + 1, T - p_j + 1]$, $\gamma(j, t) = \max\{r_j + 1, t - p_j + 1\}$.

A função objetivo (3.1) consiste em minimizar o custo ponderado das datas de início de processamento dos *jobs*. As restrições de alocação (3.2) garantem que o processamento de cada *job* só seja iniciado exatamente uma vez. As restrições de capacidade (3.3) garantem que a máquina processe no máximo um *job* a cada intervalo de tempo t . As restrições (3.4), por sua vez, garantem a integralidade das variáveis de decisão.

Considerando que o problema estudado nesta dissertação apresenta datas de chegada para os *jobs*, como observa também van den Akker *et al.* [1], as variáveis x_{jt} para $t = 1, 2, \dots, r_j$ não precisam ser definidas. Portanto, as variáveis são definidas, para cada *job* j , apenas no intervalo $[r_j + 1, T - p_j + 1]$, que são as possíveis datas de início de seu processamento. Considerando, também, que todos os custos na função objetivo são não decrescentes, a viabilidade do problema é garantida então se

$$T \geq \sum_{j=1}^n p_j + \max_{j \in J} r_j.$$

Por outro lado, verifica-se que, em uma solução ótima,

$$t^* \leq \sum_{j=1}^n p_j + \max_{j \in J} r_j,$$

onde t^* é o intervalo de tempo quando é finalizada a execução do último *job* de uma seqüência ótima. Assim, para o caso geral, definindo

$$T = \sum_{j=1}^n p_j + \max_{j \in J} r_j,$$

gera-se o problema de menor dimensão com sua viabilidade garantida.

3.2 Relaxação Lagrangeana Aplicada ao SMSP Formulado com Indexação no Tempo

Ao aplicar a técnica de Relaxação Lagrangeana para resolver um Problema de Programação Linear Inteira, deve-se, primeiro, identificar um conjunto de restrições convenientes para serem relaxadas. Idealmente, o Problema Lagrangeano obtido deve ser de fácil resolução. Para o caso específico do SMSP formulado com indexação no tempo, serão apresentados dois possíveis Problemas Lagrangeanos. O primeiro refere-se à relaxação das restrições de capacidade da máquina (3.3) e, o segundo, das restrições de alocação (3.2).

Associando um multiplicador de Lagrange $\beta_t \geq 0$, a cada uma das T restrições de capacidade de máquina, retirando-as do conjunto de restrições e acrescentando-as à Função Objetivo através de seu produto por β_t , obtemos o primeiro Problema Lagrangeano:

$$\min \quad Z_\beta = \sum_{j=1}^n \sum_{t=r_j+1}^{T-p_j+1} c_{jt} x_{jt} + \sum_{t=1}^T \sum_{j=1}^n \sum_{s=\gamma(j,t)}^t x_{js} \beta_t - \sum_{t=1}^T \beta_t \quad (3.5)$$

$$\text{s. a:} \quad \sum_{t=r_j+1}^{T-p_j+1} x_{jt} = 1 \quad \forall j = 1, 2, \dots, n \quad (3.6)$$

$$x_{jt} \in \{0, 1\} \quad \forall j = 1, 2, \dots, n; \forall t = r_j + 1, \dots, T - p_j + 1 \quad (3.7)$$

Até o momento, de nosso conhecimento, nenhuma referência foi encontrada na literatura utilizando este primeiro Problema Lagrangeano. Acredita-se que a dificuldade associada a esta relaxação decorre do fato de que, na resolução do Problema Dual Lagrangeano referente a este problema, $\|g\| \rightarrow \infty$ e, portanto, $\Theta \rightarrow 0$. Esta dificuldade pode ser contornada, por exemplo, com o desenvolvimento de Algoritmos *Relax-and-Cut*, onde apenas algumas das restrições de capacidade de máquina seriam efetivamente dualizadas.

O segundo Problema Lagrangeano, apresentado a seguir, refere-se à relaxação das restrições de alocação (3.2). Esta relaxação corresponde à mesma estudada por Avella *et al.* [5].

Associando um multiplicador de Lagrange λ_j , real e irrestrito em sinal, a cada uma das n restrições de alocação, retirando-as do conjunto de restrições e acrescentando-as à Função Objetivo por meio de seu produto por λ_j , obtemos o segundo Problema Lagrangeano:

$$\min \quad Z_\lambda = \sum_{j=1}^n \sum_{t=r_j+1}^{T-p_j+1} c_{jt} x_{jt} + \sum_{j=1}^n \lambda_j \left(\sum_{t=r_j+1}^{T-p_j+1} x_{jt} - 1 \right) \quad (3.8)$$

$$\text{s. a:} \quad \sum_{j=1}^n \sum_{s=\gamma(j,t)}^t x_{js} \leq 1 \quad \forall t = 1, 2, \dots, T \quad (3.9)$$

$$x_{jt} \in \{0, 1\} \quad \forall j = 1, 2, \dots, n; \forall t = r_j + 1, \dots, T - p_j + 1 \quad (3.10)$$

Simplificando, temos

$$\min \quad Z_\lambda = \sum_{j=1}^n \sum_{t=r_j+1}^{T-p_j+1} (c_{jt} + \lambda_j) x_{jt} - \sum_{j=1}^n \lambda_j \quad (3.11)$$

$$\text{s. a:} \quad \sum_{j=1}^n \sum_{s=\gamma(j,t)}^t x_{js} \leq 1 \quad \forall t = 1, 2, \dots, T \quad (3.12)$$

$$x_{jt} \in \{0, 1\} \quad \forall j = 1, 2, \dots, n; \forall t = r_j + 1, \dots, T - p_j + 1 \quad (3.13)$$

Nesta dissertação, opta-se pela utilização do segundo Problema Lagrangeano. Os principais fatos que motivaram esta escolha foram a facilidade de resolução deste problema, através de um algoritmo baseado em Programação Dinâmica e, por ele não

apresentar a dificuldade associada ao primeiro Problema Lagrangeano apresentado.

Conforme exposto no Capítulo anterior, por definição, o Problema Dual Lagrangeano referente ao Problema Lagrangeano adotado é:

$$Z_d = \max_{\lambda} \{ \min Z_{\lambda} \} \quad (3.14)$$

Para a resolução do Problema Dual Lagrangeano (3.14), buscando o limite inferior mais próximo do valor da solução do problema original, é proposta a utilização do Método do Subgradiente, baseado em Held *et al.* [20], descrito na seção 3.2.3. Primeiramente, serão descritos a heurística proposta para a obtenção do limite superior utilizado no método, seção 3.2.1, e, a forma como é resolvido o Problema Lagrangeano adotado, seção 3.2.2.

3.2.1 Heurística Utilizada para a Obtenção de Limites Superiores

A heurística desenvolvida, nesta dissertação, para obter uma solução viável para o SMSP, utilizada como limite superior (Z_{ub}) no Método do Subgradiente, possui duas fases. A primeira gera uma solução inicial viável para o problema (Heurística Construtiva) e foi baseada nas heurísticas descritas por Jouglet *et al.* [21] (seção 3.2.1.1). A segunda fase consiste em uma Busca Local, que por sua vez, foi baseada nos procedimentos descritos por Gupta e Smith [18] (seção 3.2.1.2).

3.2.1.1 Primeira Fase

A primeira fase da heurística proposta consiste em gerar uma solução inicial viável através do seqüenciamento dos *jobs* pela sua data de chegada (r_j) e pelo seu custo. São seqüenciados primeiro aqueles *jobs* que estão disponíveis mais cedo e, em caso de empate, decide-se pelo menor custo. Este procedimento é representado pela Tabela 3.1.

Tabela 3.1: Heurística Construtiva Gulosa

Algoritmo Guloso: crie uma seqüência $\sigma = (J_{[1]}, \dots, J_{[n]})$

- 1: $NS \leftarrow \{J_1, \dots, J_n\}$,
 - 2: $i \leftarrow 0$,
 - 3: **While** $NS \neq \emptyset$ **do**
 - 4: Escolha $J_x \in NS$ tal que $r_x = \min_{J_l \in NS} \{r_l\}$,
 - 5: $i \leftarrow i + 1$,
 - 6: $J_{[i]} \leftarrow J_x$,
 - 7: $NS \leftarrow NS \setminus \{J_x\}$
 - 8: **End While**
-

No algoritmo da Tabela 3.1, NS denota o conjunto de *jobs* não seqüenciados e, $J_{[i]}$, o *job* seqüenciado na i -ésima posição. O processo de escolha de J_x adotado, denominado *Earliest Start Time* - EST, consiste em construir seqüenciamentos nos quais a máquina não se mantém inativa se houver algum *job* disponível para ser processado. A cada iteração, o *job* escolhido para ser processado será o que estiver disponível mais cedo (menor data de chegada) e pertencer ao conjunto NS . Em caso de empate, isto é, dois ou mais *jobs* estarem disponíveis para processamento no mesmo instante de tempo t , a regra de prioridade adotada consiste em seqüenciar primeiro aquele que possuir o menor custo no período de tempo avaliado. A primeira fase da Heurística pára quando todos os *jobs* estiverem seqüenciados, ou seja, quando o conjunto NS estiver vazio. Segundo Jouglet *et al.* [21], este procedimento apresenta complexidade $O(n^2)$.

Simplificadamente, o exemplo abaixo ilustra a solução viável obtida pelo Algoritmo Guloso aplicado para resolver uma instância com 4 *jobs*:

Tabela 3.2: Dados do Exemplo

| <i>job</i> | 1 | 2 | 3 | 4 |
|------------|---|---|----|---|
| p_j | 1 | 6 | 10 | 1 |
| r_j | 1 | 0 | 7 | 2 |
| w_j | 4 | 8 | 1 | 5 |

Como $r_2 = 0 < r_1 = 1 < r_4 = 2 < r_3 = 7$, a solução inicial proposta pelo Algoritmo Guloso é $v = (2, 1, 4, 3)$.

Na solução apresentada para o exemplo, o *job* 2 é seqüenciado primeiro, iniciando sua execução no período de tempo $t = 1$ ($x_{21} = 1$). Em seguida, são seqüenciados os *jobs* 1, iniciando sua execução no período de tempo $t = 7$ ($x_{17} = 1$), 4, iniciando sua execução no período de tempo $t = 8$ ($x_{48} = 1$) e por fim o *job* 3, iniciando sua execução no período de tempo $t = 9$ ($x_{39} = 1$). O término do processamento dos *jobs* ocorre no período de tempo $t = 19$, com $Z = 67$.

3.2.1.2 Segunda Fase

A segunda fase da heurística desenvolvida consiste em uma Busca Local. Em linhas gerais, essa classe de heurísticas parte de uma solução inicial qualquer e caminha, a cada iteração, de vizinho para vizinho de acordo com a definição de vizinhança adotada.

Na Busca Local proposta, uma nova solução viável para o SMSP é obtida, a partir da solução encontrada pela primeira fase da heurística, deslocando-se o *job* j alocado na posição k da seqüência de produção para a posição $k + 1$. Se esta solução for de melhora, ela passa a ser a solução corrente e o procedimento é reiniciado. Caso contrário, o *job* j retorna à sua posição de origem (k) e é, em seguida, deslocado para a posição $k + 2$, gerando uma nova solução. Este processo é realizado, sucessivamente, até que o *job* j seja deslocado para a última posição da seqüência de produção, caso nenhuma melhora seja encontrada com seus possíveis deslocamentos. Este procedimento é feito, iniciando pelo *job* alocado na primeira posição ($k = 1$) e, terminando naquele alocado na penúltima posição. O procedimento pára quando o *job* alocado na penúltima posição é deslocado para a última e não há melhora na solução. Este procedimento possui complexidade $O(n^2)$.

Ilustrando este procedimento, se o seqüenciamento s encontrado pela primeira fase da heurística for $(2, 1, 4, 3)$, o *job* 2, que se encontra na primeira posição, pode ser deslocado para as posições 2, 3 e 4, o *job* 1, que se encontra na segunda posição, pode ser deslocado para as posições 3 e 4 e o *job* 4, que se encontra na posição 3, pode ser deslocado, somente, para a posição 4. Se, dentre estas novas soluções obtidas, uma delas for melhor que a solução s , ela assume o lugar de s e o procedimento é reiniciado. Se não for encontrada nenhuma solução melhor do que s , o procedimento pára quando o *job* 4 for deslocado para a posição 4.

Portanto, quando o *job* 2 for deslocado para a posição 3, o novo seqüenciamento s' encontrado será $(1, 4, 2, 3)$. A representação deste exemplo pode ser vista na Figura 3.1.

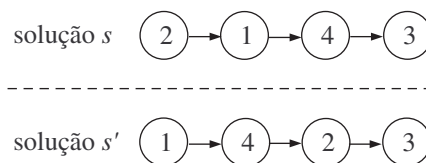


Figura 3.1: Estrutura de vizinhança com troca na ordem de processamento do *job* 2

Então, sendo o valor da solução inicial s igual a 67 e o valor da solução s' encontrada pelo procedimento de Busca Local igual a 47, s' assume o lugar de s e o procedimento é reiniciado deslocando o *job* 1 para a posição 2 e assim sucessivamente.

3.2.2 Resolução do Problema Lagrangeano

A matriz de restrições (3.12) que define o Problema Lagrangeano adotado é uma matriz intervalo [30] e, assim sendo, é totalmente unimodular. Embora possa ser resolvido através de Programação Linear, adota-se nesta dissertação uma estratégia de solução diferente para sua resolução. O fato de as restrições (3.12) definirem uma matriz intervalo permite que o Problema Lagrangeano seja resolvido em tempo pseudo-polinomial através de um algoritmo baseado em Programação Dinâmica que, implicitamente, lida com a determinação de um caminho mínimo em um grafo convenientemente escolhido.

Os vértices deste grafo representam instantes de tempo: para cada instante t no horizonte de planejamento T , associamos um vértice t no grafo. Emprega-se também um vértice artificial, de índice $T + 1$ no grafo em questão. Na construção do grafo, utiliza-se dois tipos de arcos: os arcos de processo e os arcos que representam ociosidade da máquina. Os arcos de processo são obtidos considerando-se todas as possibilidades de execução de cada um dos *jobs* ao longo do horizonte de tempo. Assim sendo, para cada *job* j e cada instante de tempo $t \geq r_j + 1$, inseri-se um arco de processo que inicia em t e termina em $t + p_j$. Observe que este arco só é inserido caso a data de disponibilidade do *job* j seja satisfeita em t . Observe também que podem existir dois (ou mais) arcos de processo distintos que possuem as mesmas extremidades inicial e final. Para compreender como esta situação pode ocorrer, considere dois *jobs* i e j tais que $p_j = p_i$ e que $t = 1 + \max\{r_i, r_j\}$. Claramente, dois arcos que partem de t e chegam a $t + p_i$ seriam empregados, cada um representando a possibilidade de a máquina iniciar o processamento de cada um dos *jobs* no instante t . Os arcos associados à ociosidade, por sua vez, representam a possibilidade de a máquina não processar *job* algum entre intervalos consecutivos de tempo. Assim sendo, para cada $t \geq 1$, um arco de ociosidade que inicia em t e termina em $t + 1$ é introduzido no grafo.

A título de ilustração, considere o exemplo mostrado na Figura 3.2, no qual dois *jobs* devem ser executados pela máquina. Assuma que ambos estão disponíveis para processamento no instante inicial ($r_1 = 0$ e $r_2 = 0$), que seus tempos de processamento sejam, respectivamente, $p_1 = 2$ e $p_2 = 3$ e que $T = 6$. Na Figura 3.2, os arcos

associados à ociosidade da máquina são representados através de linha tracejada. Os arcos de processo associados à execução do *job* 1 encontram-se sempre acima dos arcos de ociosidade. Já os arcos de processo associados ao *job* 2 estão representados abaixo dos arcos de ociosidade.

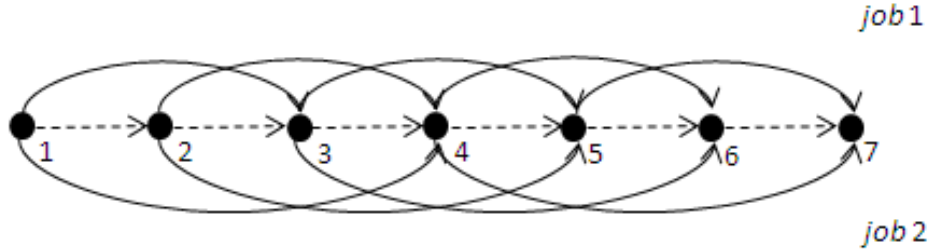


Figura 3.2: Grafo para o exemplo. Fonte: Adaptado de van den Akker *et al.*[2]

Considerando que, ao percorrer o arco de processo que inicia em t e termina em $t + p_j$, paga-se o custo $c_{jt} + \lambda_j$ e que, ao percorrer um arco de ociosidade, nenhum custo é incorrido, uma solução viável para o Problema Lagrangeano pode ser obtida através de um caminho de custo mínimo, que inicia em 1 e termina em $T + 1$, no grafo descrito acima. Possivelmente, este caminho mínimo poderá envolver mais de um arco de processo associado a um *job*, ou mesmo, deixar de incluir um arco de processo associado a algum *job*. Nestes casos, a restrição de alocação (3.2) associada ao *job* que foi executado mais de uma vez (ou que deixou de ser executado) foi violada pela solução Lagrangeana.

Note que o caminho mínimo só vai passar por algum arco de processo quando seu custo for nulo ou negativo ($c_{jt} + \lambda_j \leq 0$), caso contrário ele sempre passará pelos arcos de ociosidade com custo zero. Devido ao fato de os multiplicadores de Lagrange (λ_j) serem irrestritos em sinal, isso ocorrerá quando algum λ_j for negativo e $|\lambda_j| \geq c_{jt}$, pois os valores de c_{jt} são sempre positivos e crescentes no tempo. Como os multiplicadores de Lagrange são calculados nesta dissertação por $\lambda_j^{k+1} = \lambda_j^k + \Theta^k g_j^k$, $j = 1, 2, \dots, n$, considerando que Θ é sempre positivo e que a solução do Problema Lagrangeano pode violar as restrições de alocação, λ_j^{k+1} pode ser negativo se:

- $\lambda_j^k < 0$, $\Theta^k g_j^k \geq 0$, ou seja, $g_j^k \geq 0$, e $|\lambda_j^k| > \Theta^k g_j^k$;
 $g_j^k \geq 0$ quando o *job* j é alocado uma ($g_j^k = 0$) ou mais ($g_j^k > 0$) vezes.
- $\lambda_j^k \geq 0$, $\Theta^k g_j^k < 0$, ou seja, $g_j^k < 0$, e $|\Theta^k g_j^k| > \lambda_j^k$;
 $g_j^k < 0$ quando o *job* j não é alocado nenhuma vez ($g_j^k = -1$).

- $\lambda_j^k \leq 0$ e $\Theta^k g_j^k < 0$ ou $\lambda_j^k < 0$ e $\Theta^k g_j^k \leq 0$.

Portanto, o caminho mínimo no grafo descrito acima vai ter sempre custo zero ou negativo. A parcela $\sum_{j=1}^n \lambda_j$, da Função Lagrangeana, é que garante valores positivos para a solução do Problema Lagrangeano.

Na Figura 3.3, ilustramos uma solução Lagrangeana que é viável para o problema ilustrado na Figura 3.2. Na solução apresentada, o processamento do *job 1* é iniciado em $t = 1$ ($x_{11} = 1$) e o processamento do *job 2* é iniciado no período $t = 3$ ($x_{23} = 1$). Para chegar ao destino $T + 1$, emprega-se o arco de ociosidade que inicia em T e termina em $T + 1$.

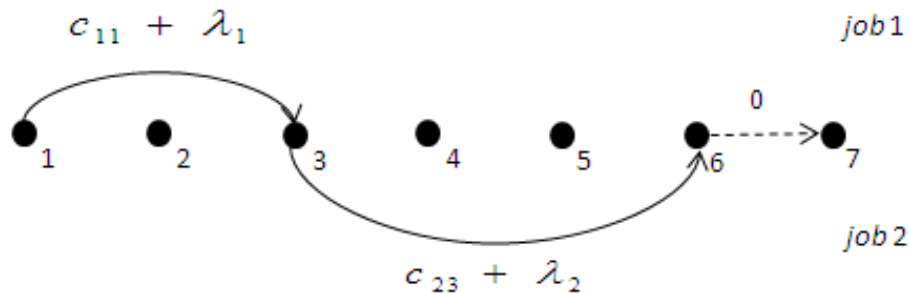


Figura 3.3: Caminho no Grafo. Fonte: Próprio autor

Observe que o grafo associado à resolução do Problema Lagrangeano é acíclico e que todos os arcos (t, t') a ele pertencentes são tais que $t < t'$. Devido a estas características, o Problema de Caminho com Custo Mínimo associado pode ser resolvido através de um algoritmo simples, baseado em Programação Dinâmica.

3.2.2.1 Algoritmo de Programação Dinâmica para a Resolução do Problema Lagrangeano

De acordo com Arenales *et al.* [4], a Programação Dinâmica é uma das técnicas que podem ser utilizadas para a resolução de Problemas de Otimização. Tal técnica consiste na decomposição do problema original em uma seqüência de problemas menores e mais simples de serem resolvidos.

O Algoritmo de Programação Dinâmica adotado para a resolução do Problema Lagrangeano consiste em percorrer todos os nós t do grafo associado a ele, de 1 até $T + 1$, calculando-se o caminho mínimo entre 1 e t . O problema é resolvido quando o custo do caminho mínimo de 1 até $T + 1$ for determinado. Para estabelecer a recursão

de Programação Dinâmica que define o algoritmo, será empregada a função $H(t)$, que define o custo do caminho mínimo entre 1 e t :

$$H(t) = \min\{H(t-1), \min_{j \in J} \{H(t-p_j+1) + \lambda_j + c_{j,t-p_j+1} : \forall j \text{ tal que } r_j \leq t-p_j+1\}\}. \quad (3.15)$$

Inicializa-se $H(1) = -\sum_{j=1}^n \lambda_j$. Assim sendo, ao final do algoritmo, tem-se que:

$$Z_\lambda = H(T+1)$$

3.2.3 Otimização via Método do Subgradiente

Um limitante válido para o valor da solução do problema original é determinado pelo valor da solução do Problema Dual Lagrangeano (Z_d). Z_d pode, então, ser otimizado através do Método do Subgradiente obtendo, dessa forma, o limitante mais próximo do valor da solução do problema original.

Para inicializar as iterações do Método do Subgradiente utilizado nesta dissertação, utiliza-se da heurística descrita na seção 3.2.1 para obter uma solução viável para o problema original, determinando, assim, um limite superior (Z_{ub}). Atribui-se, inicialmente, valor zero para os multiplicadores de Lagrange ($\lambda_j^0 = 0, \forall j = 1, 2, \dots, n$). Em seguida, resolve-se pela primeira vez o Problema Lagrangeano, com os multiplicadores de Lagrange nulos, através do procedimento apresentado na seção 3.2.2, determinando o primeiro limite inferior (Z_λ).

Com base na atual solução do Problema Lagrangeano ($x_{jt}^k, j = 1, \dots, n$ e $t = r_j + 1, \dots, T - p_j + 1$), calcula-se os subgradientes, através de:

$$g_j^k = \sum_{t=r_j+1}^{T-p_j+1} x_{jt}^k - 1 \quad \forall j = 1, 2, \dots, n.$$

É calculado, em seguida, o tamanho do passo (Θ^k). Para isso utiliza-se a seguinte fórmula:

$$\Theta^k = \alpha \frac{Z_{ub} - Z_\lambda}{\|g_j^k\|^2}$$

Para o cálculo do tamanho do passo, foram testadas várias políticas para determinar o valor inicial e a forma de atualização de α . A que apresentou o melhor

resultado foi a adotada, onde o valor de α é iniciado com valor 2 e reduzido à metade a cada 50 iterações consecutivas do Método do Subgradiente sem melhora no valor da solução do Problema Lagrangeano (Z_λ).

Utilizando os valores obtidos para os subgradientes ($g_j^k, \forall j = 1, 2, \dots, n$) e para o tamanho do passo (Θ^k), atualiza-se, dando início a uma nova iteração, os multiplicadores de Lagrange através da expressão:

$$\lambda_j^{k+1} = \lambda_j^k + \Theta^k g_j^k \quad \forall j = 1, 2, \dots, n$$

onde λ_j é irrestrito em sinal devido às restrições relaxadas (3.2) serem de igualdade.

Um novo valor para Z_λ é determinado, resolvendo novamente o Problema Lagrangeano, utilizando desta vez os novos valores obtidos para os multiplicadores de Lagrange. Calcula-se agora, novos valores para os subgradientes, baseados na nova solução do Problema Lagrangeano, e o novo tamanho do passo. Tais valores são utilizados para dar início a mais uma iteração, atualizando os multiplicadores de Lagrange. Este processo é realizado, continuamente, até que o critério de parada adotado para o método seja satisfeito.

A menos que se obtenha $Z_{ub} = Z_\lambda$, não há nenhuma outra indicação clara de quando terminar as iterações do Método do Subgradiente. Por este motivo, a estratégia adotada como critério de parada é aquela geralmente utilizada pela literatura: limitar o número de iterações do método. O número máximo de iterações adotado foi 900 ($k = 1, 2, \dots, 900$).

Visando ainda conferir melhores propriedades de convergência ao Método do Subgradiente descrito acima, como apresentado por Avella *et al.* [5], baseado em Camerini *et al.* [10], foi utilizada nesta dissertação a deflexão do subgradiente (d) no lugar do subgradiente (g), onde a direção de busca utilizada na iteração k é a média ponderada do subgradiente da iteração k (g_j^k) e das direções obtidas nas duas iterações anteriores (d_j^{k-1} e d_j^{k-2}) e é dada pela fórmula:

$$d_j^k = \frac{g_j^k + 0,3d_j^{k-1} + 0,1d_j^{k-2}}{1,4} \quad \forall j = 1, 2, \dots, n$$

onde $d_j^0 = 0$ e $d_j^1 = g_j^1$.

Portanto, o tamanho do passo passa a ser calculado pela seguinte fórmula:

$$\Theta^k = \alpha \frac{Z_{ub} - Z_{\lambda^k}}{\|d_j^k\|^2}$$

e, os multiplicadores de Lagrange atualizados por:

$$\lambda_j^{k+1} = \lambda_j^k + \Theta^k d_j^k \quad \forall j = 1, 2, \dots, n$$

Ao final de cada iteração do Método do Subgradiente, as variáveis nulas da solução atual do Problema Lagrangeano ($x_{jt}^k = 0$) são testadas para serem fixadas, e, conseqüentemente, eliminadas do problema. O procedimento para fixação de variáveis é descrito na seção 3.3.

3.3 Fixação de Variáveis

Devido ao elevado número de variáveis existente na formulação do SMSP pela indexação no tempo, Sousa e Wolsey [31] destacam a importância da disponibilidade de um bom procedimento para a redução deste número. Um procedimento bastante utilizado na literatura, aplicado também a outros modelos de Programação Linear Inteira, é o que avalia os custos reduzidos de soluções viáveis, tanto da Relaxação Lagrangeana como da Relaxação Linear. A avaliação dos custos reduzidos pode permitir a fixação de algumas variáveis nos valores 0 ou 1, reduzindo, assim, o espaço de soluções.

De acordo com Mitchell [29], para se fixar variáveis em Problemas de Programação Inteira (PPI) é necessário conhecer, primeiramente, uma solução viável para o PPI. O valor desta solução provê um limite superior (Z_{ub}) para o valor ótimo do PPI. Quando se resolve otimamente a Relaxação de Programação Linear do PPI, gera-se uma solução com valor Z_{PL} . Se uma variável x_i tem valor zero na solução da Relaxação de Programação Linear do PPI e, se o custo reduzido c_i desta variável for maior que $Z_{ub} - Z_{PL}$, então a variável x_i tem que ter valor zero em qualquer solução ótima para o PPI. Isso porque qualquer solução viável para a Relaxação de Programação Linear do PPI com $x_i = 1$ tem valor, pelo menos, $c_i + Z_{PL}$. Assim, se o seu custo reduzido for suficientemente grande, a variável pode ser fixada como zero.

O custo reduzido c_i de uma variável x_i corresponde ao custo por aumento de uma unidade no valor da função objetivo se x_i assumir um valor diferente de zero na solução do problema. Para problemas de minimização, uma solução básica é ótima se todos os custos reduzidos das variáveis não-básicas forem positivos. Isso indica que o valor da função objetivo aumentará se uma variável não-básica assumir um valor não-nulo (Thorsteinsson e Ottosson [32]).

O procedimento utilizado nesta dissertação para fixar variáveis, reduzindo, desta forma, o elevado número de variáveis presentes no modelo adotado, é baseado nos custos reduzidos das variáveis com valor nulo em uma solução Lagrangeana.

Ao final de cada iteração do Método do Subgradiente, são testadas todas as variáveis com valor zero na solução Lagrangeana corrente. O procedimento consiste em tomar uma variável x_{jt} que não foi selecionada na solução Lagrangeana corrente ($x_{jt} = 0$) e impor que ela seja necessariamente selecionada ($x_{jt} = 1$). Avalia-se, então, qual é a mínima variação no custo Lagrangeano impondo $x_{jt} = 1$. Para isto, é necessário calcular o custo reduzido associado à variável candidata a entrar. Este cálculo é feito somando-se ao custo Lagrangeano de x_{jt} , o custo de dois caminhos mínimos no grafo que foi utilizado para resolver o Problema Lagrangeano. Considerando uma variável $x_{jt} = 0$, o primeiro caminho tem como origem o vértice 1 e, como destino, o vértice t ; o segundo, tem como origem o vértice $t + p_j$ e, como destino, o vértice $T + 1$. Os vértices t e $t + p_j$ representam, respectivamente, os vértices origem e destino do arco que representa a máquina processamento o *job* j , com custo $c_{jt} + \lambda_j$, referente à variável $x_{jt} = 0$ considerada. Denotando como c_1 e c_2 os custos destes dois caminhos mínimos, tem-se que: se $c_1 + c_2 +$ custo Lagrangeano de x_{jt} for maior que o valor do limite superior ($c_1 + c_2 + (c_{jt} + \lambda_j) > Z_{ub}$), pode-se fixar $x_{jt} = 0$, garantindo, assim, que o *job* j não seja alocado no instante t .

3.4 Comparação com a Relaxação Lagrangeana Proposta por Avella *et al.*[5]

Avella *et al.*[5] utilizam, em seu trabalho, o modelo com variáveis indexadas no tempo para representar o SMSP e, propõem uma Relaxação Lagrangeana para a sua resolução. O modelo utilizado nesta dissertação difere do utilizado por este autor na definição das variáveis de decisão. Já a Relaxação Lagrangeana utilizada difere na forma como são determinados os limites inferior (Z_λ) e superior (Z_{ub}) e, na forma de atualização do parâmetro α .

No modelo utilizado por Avella *et al.* [5], as variáveis x_{jt} determinam em qual intervalo de tempo o processamento do *job* j termina, sendo, portanto, definidas para cada *job* j no intervalo $[r_j + p_j - 1, T]$. Onde c_{jt} é o custo por completar o processamento do *job* j no instante de tempo t e, é calculado por: $c_{jt} = w_{jt}$.

Na Relaxação Lagrangeana proposta por Avella *et al.* [5], a determinação dos limites inferiores, ou seja, a resolução do Problema Lagrangeano, é feita associando a ele um grafo interseção $G(V, E)$, contendo um vértice jt , com custo $c_{jt} + \lambda_j$, para cada variável x_{jt} . O Problema Lagrangeano é resolvido, polinomialmente por um algoritmo de Programação Dinâmica, como um *Weighted Stable Set Problem* no grafo G .

Baseado no fato de que as soluções do Problema Lagrangeano não são necessariamente viáveis para o problema original, pois elas podem violar as restrições relaxadas, Avella *et al.* [5] utilizam uma heurística para gerar limites superiores, viabilizando estas soluções para o problema original. Na solução do Problema Lagrangeano obtido por esses autores, pode ocorrer que alguns *jobs* sejam seqüenciados mais de uma vez e outros não sejam seqüenciados. Assim, os autores propõem em sua heurística uma maneira de eliminação dos *jobs* seqüenciados mais de uma vez e a inserção daqueles não seqüenciados.

Em relação à atualização do parâmetro α , Avella *et al.* [5] adotam a que consideram a mais eficiente estratégia: dividi-lo por 1,01 a cada iteração, independente do comportamento da solução Lagrangeana (Z_λ).

Uma comparação dos resultados obtidos pela Relaxação Lagrangeana utilizada nesta dissertação com os resultados obtidos por Avella *et al.* [5] é apresentada na seção 4.2.

Capítulo 4

Resultados e Análise

4.1 Geração dos Problemas-Teste

Para a geração dos problemas-teste utilizados nesta dissertação, foi criado um procedimento, baseado em Avella *et al.* [5], onde eles são gerados randomicamente, de acordo com as seguintes características:

- os pesos w_j são gerados de $U[1, 20]$, isto é, são uniformemente distribuídos no intervalo $[1, 20]$;
- os tempos de processamento p_j são gerados de $U[1, p_{max}]$, sendo p_{max} o tempo de processamento máximo adotado para os *jobs*.
- as datas de chegada r_j são geradas de $U[0, 1/2 \sum_{j=1}^n p_j]$;

Avella *et al.* [5] geraram problemas-teste com 75, 100, 200, 300 e 400 *jobs*, com tempos de processamento máximo iguais a 10, 20, 30, 40 e 50.

4.2 Apresentação e Análise dos Resultados

A metodologia proposta foi desenvolvida na Linguagem de Programação *C++* e executada nos programas *Bloodshed Dev C++* e *ILOG CPLEX 9.1*. A máquina utilizada foi um *Centrino Duo* 1,83 Ghz e 2 GB de memória RAM.

Nas tabelas apresentadas a seguir, cada problema-teste é representado da seguinte forma: xx_pp , onde xx é o número de *jobs* e pp é o tempo de processamento

máximo para os *jobs*.

Na Tabela 4.1 são apresentados os resultados obtidos pela Relaxação Lagrangeana utilizada nesta dissertação, para problemas-teste com as mesmas dimensões utilizadas por Avella *et al.* [5]. Nesta tabela são apresentados os limites inferior (Z_d) e superior (Z_{UB}), o *gap* Lagrangeano e o tempo, dado em segundos, gasto para a obtenção dos limites. O Gap Lagrangeano é calculado por $((Z_{UB} - Z_d)/Z_{UB}) \times 100$.

| Instância | Z_d | Z_{UB} | Gap(%) | t |
|-----------|----------|----------|--------|--------|
| 75_10 | 97173,1 | 97682 | 0,5 | 2,05 |
| 75_20 | 182044 | 183313 | 0,7 | 4,44 |
| 75_30 | 243287 | 244597 | 0,5 | 6,06 |
| 75_40 | 347687 | 349528 | 0,5 | 7,42 |
| 75_50 | 375479 | 377690 | 0,6 | 10,31 |
| 100_10 | 188846 | 189793 | 0,5 | 4,13 |
| 100_20 | 312588 | 313740 | 0,4 | 7,70 |
| 100_30 | 467835 | 470505 | 0,6 | 10,20 |
| 100_40 | 765550 | 769778 | 0,5 | 17,31 |
| 100_50 | 605295 | 609038 | 0,6 | 16,16 |
| 200_10 | 678697 | 682765 | 0,6 | 18,22 |
| 200_20 | 1149850 | 1155410 | 0,5 | 32,06 |
| 200_30 | 1796260 | 1806850 | 0,6 | 46,51 |
| 200_40 | 2321640 | 2335890 | 0,6 | 63,82 |
| 200_50 | 3237680 | 3256610 | 0,6 | 86,82 |
| 300_10 | 1390420 | 1398060 | 0,5 | 41,82 |
| 300_20 | 2779460 | 2797030 | 0,6 | 76,20 |
| 300_30 | 3981640 | 4005560 | 0,6 | 113,24 |
| 300_40 | 5737660 | 5765220 | 0,5 | 147,30 |
| 300_50 | 6994600 | 7040020 | 0,6 | 194,34 |
| 400_10 | 2665670 | 2680100 | 0,5 | 76,47 |
| 400_20 | 4827020 | 4849160 | 0,5 | 137,44 |
| 400_30 | 7754460 | 7796500 | 0,5 | 223,58 |
| 400_40 | 10016200 | 10081500 | 0,6 | 265,97 |
| 400_50 | 12976100 | 13031400 | 0,4 | 355,03 |

Tabela 4.1: Resultados obtidos pela Relaxação Lagrangeana proposta para os problemas-teste com as mesmas dimensões utilizados por Avella *et al.* [5]

A Relaxação Lagrangeana, utilizada nesta dissertação, obteve um *gap* médio igual a 0,5%. Já a Relaxação Lagrangena proposta Avella *et al.* [5] obteve, para problemas-teste de mesmas dimensões, um *gap* médio de 0,76%.

Para a metodologia proposta (Relaxação Lagrangeana com fixação de variáveis + *software* de otimização) foram gerados problemas-teste com o número de *jobs* igual

a 20, 30, 40, 50, 60, 70, 75, 100 e 200. Para o tempo máximo de processamento, os valores utilizados foram 10, 20 e 30. Problemas-teste com mais de 200 *jobs* e tempo de processamento máximo superior a 30 não puderam ser resolvidos pelo *software* CPLEX devido à elevada necessidade de memória computacional.

A Tabela 4.2 apresenta o número de variáveis, o tempo médio de processamento e o peso médio dos *jobs* em cada problema-teste. Os resultados numéricos obtidos pela metodologia, com os tempos dados em segundos, podem ser visualizados nas Tabelas 4.3, 4.4 e 4.6.

| Instância | Número de Variáveis | p_j (Médio) | w_j (Médio) |
|-----------|---------------------|---------------|---------------|
| 20_10 | 14822 | 5 | 9 |
| 20_20 | 15409 | 10 | 10 |
| 20_30 | 17917 | 15 | 9 |
| 30_10 | 4731 | 4 | 9 |
| 30_20 | 10092 | 9 | 11 |
| 30_30 | 17196 | 15 | 8 |
| 40_10 | 12336 | 6 | 10 |
| 40_20 | 21608 | 11 | 9 |
| 40_30 | 33095 | 16 | 10 |
| 50_10 | 18108 | 5 | 9 |
| 50_20 | 33791 | 10 | 9 |
| 50_30 | 49507 | 15 | 10 |
| 60_10 | 25908 | 5 | 9 |
| 60_20 | 43152 | 9 | 9 |
| 60_30 | 73623 | 16 | 10 |
| 70_10 | 28879 | 4 | 9 |
| 70_20 | 60985 | 10 | 9 |
| 70_30 | 91039 | 14 | 10 |
| 75_10 | 40925 | 4 | 9 |
| 75_20 | 75936 | 10 | 10 |
| 75_30 | 93682 | 15 | 9 |
| 100_10 | 67544 | 5 | 10 |
| 100_20 | 129879 | 11 | 10 |
| 100_30 | 180563 | 16 | 9 |
| 200_10 | 275488 | 6 | 9 |
| 200_20 | 480591 | 10 | 9 |
| 200_30 | 729512 | 15 | 10 |

Tabela 4.2: Características dos problema-teste

Na Tabela 4.3, são apresentados o valor da Relaxação de Programação Linear (Z_{PL}) e o tempo computacional gasto para obtê-lo, o limite inferior (Z_d), o limite superior (Z_{UB}), o *gap* e o tempo computacional relativos à Relaxação Lagrangeana,

bem como a porcentagem de variáveis fixadas.

| Instância | Z_{PL} | t | Relaxação Lagrangeana | | | | Fixação (%) |
|-----------|-----------|--------|-----------------------|----------|--------|-------|-------------|
| | | | Z_d | Z_{UB} | Gap(%) | t | |
| 20_10 | 38135,1 | 3,6 | 38129,8 | 38186 | 0,15 | 0,87 | 85,03 |
| 20_20 | 51918,8 | 4,2 | 51908,5 | 52240 | 0,63 | 0,87 | 72,61 |
| 20_30 | 51334,4 | 5,1 | 51304,6 | 51528 | 0,36 | 1,03 | 78,72 |
| 30_10 | 9798,0 | 0,2 | 9795,95 | 9845 | 0,50 | 0,28 | 73,81 |
| 30_20 | 32660,4 | 1,3 | 32654,9 | 33093 | 1,32 | 0,59 | 55,65 |
| 30_30 | 38878,7 | 5,1 | 38874,5 | 38916 | 0,11 | 1,00 | 85,11 |
| 40_10 | 37097,4 | 1,5 | 37094,9 | 37202 | 0,29 | 0,72 | 75,40 |
| 40_20 | 67636,6 | 6,8 | 67622,7 | 67882 | 0,38 | 1,25 | 62,02 |
| 40_30 | 84743,2 | 16,3 | 84735,4 | 85088 | 0,41 | 1,89 | 71,27 |
| 50_10 | 44993,8 | 2,9 | 44987,5 | 45256 | 0,59 | 1,05 | 62,45 |
| 50_20 | 80439,8 | 11,9 | 80428,9 | 80908 | 0,59 | 1,95 | 64,99 |
| 50_30 | 123675,2 | 31,3 | 123667 | 124309 | 0,52 | 2,84 | 69,94 |
| 60_10 | 58140,2 | 5,5 | 58130,6 | 58539 | 0,70 | 1,50 | 58,14 |
| 60_20 | 108750,6 | 18,6 | 108722 | 109595 | 0,80 | 2,52 | 54,18 |
| 60_30 | 215241,6 | 63,1 | 215198 | 215926 | 0,34 | 4,28 | 72,37 |
| 70_10 | 76248,1 | 6,1 | 76172 | 76710 | 0,70 | 1,70 | 51,53 |
| 70_20 | 157133,7 | 35,6 | 157102 | 158052 | 0,60 | 3,61 | 64,22 |
| 70_30 | 271299,5 | 106,0 | 271073 | 272668 | 0,58 | 5,31 | 56,21 |
| 75_10 | 109600,4 | 11,5 | 109014 | 110133 | 1,02 | 2,00 | 46,51 |
| 75_20 | 205260,5 | 50,3 | 205192 | 207115 | 0,93 | 3,22 | 44,58 |
| 75_30 | 233455,6 | 57,1 | 233369 | 234693 | 0,56 | 6,19 | 57,36 |
| 100_10 | 207093,9 | 27,5 | 206801 | 208429 | 0,78 | 5,53 | 39,57 |
| 100_20 | 312777,7 | 40,9 | 312512 | 313740 | 0,39 | 8,75 | 58,22 |
| 100_30 | 421249,6 | 218,1 | 420935 | 423762 | 0,67 | 12,13 | 49,05 |
| 200_10 | 682391,8 | 266,5 | 681409 | 686725 | 0,77 | 21,27 | 30,25 |
| 200_20 | 1151598,1 | 836,7 | 1150220 | 1155410 | 0,45 | 37,07 | 43,95 |
| 200_30 | 1871729,6 | 4309,3 | 1870790 | 1882920 | 0,64 | 55,51 | 35,89 |

Tabela 4.3: Resultados computacionais

Na Tabela 4.4 são apresentados o valor da solução ótima obtida para os problemas-teste e os tempos computacionais gastos no *software* CPLEX sem e com a fixação das variáveis para a obtenção das soluções.

| Instância | Ótimo | t(CPLEX sem Fixação) | t(CPLEX com Fixação) |
|-----------|-----------|----------------------|----------------------|
| 20_10 | 38186,0 | 4,38 | 0,83 |
| 20_20 | 52057,0 | 5,31 | 1,89 |
| 20_30 | 51528,0 | 4,76 | 1,39 |
| 30_10 | 9845,0 | 0,38 | 0,09 |
| 30_20 | 32801,0 | 4,22 | 2,78 |
| 30_30 | 38916,0 | 6,00 | 1,86 |
| 40_10 | 37180,0 | 2,92 | 0,47 |
| 40_20 | 67751,0 | 28,97 | 17,63 |
| 40_30 | 85086,0 | 176,74 | 23,75 |
| 50_10 | 45156,0 | 30,47 | 16,09 |
| 50_20 | 80827,0 | 53,81 | 42,58 |
| 50_30 | 123676,0 | 763,88 | 422,56 |
| 60_10 | 58433,0 | 51,76 | 14,88 |
| 60_20 | 109088,0 | 62,55 | 32,47 |
| 60_30 | 215926,0 | 734,70 | 385,77 |
| 70_10 | 76476,0 | 105,30 | 86,8 |
| 70_20 | 157636,0 | 416,91 | 339,05 |
| 70_30 | 272167,0 | 557,53 | 220,12 |
| 75_10 | 109951,0 | 378,64 | 167,89 |
| 75_20 | 205616,0 | 403,69 | 127,50 |
| 75_30 | 234202,0 | 460,81 | 379,14 |
| 100_10 | 207756,0 | 133,46 | 76,53 |
| 100_20 | 313559,0 | 585,32 | 220,77 |
| 100_30 | 422513,0 | 1866,29 | 1332,03 |
| 200_10 | 684379,0 | 1081,72 | 483,56 |
| 200_20 | 1154822,0 | 2499,15 | 1602,03 |
| 200_30 | 1877719,0 | 9711,10 | 8944,32 |

Tabela 4.4: Comparação entre os tempos de resolução sem e com fixação de variáveis

Na Tabela 4.5 são descritos os percentuais de diferença dos limites inferiores (Z_d) e superiores (Z_{UB}) obtidos pela Relaxação Lagrangeana em relação à solução ótima dos problemas-teste.

| Instância | $((\text{Ótimo} - Z_d)/\text{Ótimo}) \times 100$ | $((Z_{ub} - \text{Ótimo})/Z_{ub}) \times 100$ |
|-----------|--|---|
| 20_10 | 0,15 | 0,00 |
| 20_20 | 0,29 | 0,35 |
| 20_30 | 0,36 | 0,00 |
| 30_10 | 0,50 | 0,00 |
| 30_20 | 0,45 | 0,88 |
| 30_30 | 0,11 | 0,00 |
| 40_10 | 0,23 | 0,06 |
| 40_20 | 0,19 | 0,19 |
| 40_30 | 0,41 | 0,00 |
| 50_10 | 0,37 | 0,22 |
| 50_20 | 0,49 | 0,10 |
| 50_30 | 0,08 | 0,51 |
| 60_10 | 0,52 | 0,18 |
| 60_20 | 0,34 | 0,46 |
| 60_30 | 0,34 | 0,00 |
| 70_10 | 0,40 | 0,31 |
| 70_20 | 0,34 | 0,26 |
| 70_30 | 0,40 | 0,18 |
| 75_10 | 0,85 | 0,17 |
| 75_20 | 0,21 | 0,72 |
| 75_30 | 0,46 | 0,10 |
| 100_10 | 0,46 | 0,32 |
| 100_20 | 0,33 | 0,06 |
| 100_30 | 0,37 | 0,29 |
| 200_10 | 0,41 | 0,37 |
| 200_20 | 0,40 | 0,05 |
| 200_30 | 0,37 | 0,28 |

Tabela 4.5: Percentual dos Limites Inferiores e Superiores em Relação ao Ótimo

Como pode ser observado na Tabela 4.5, os limites inferiores ficaram, em média, a 0,41% do valor da solução ótima dos problemas-teste e, os limites superiores, a 0,25%. Para os problemas-teste 20_10, 20_30, 30_10, 30_30 e 60_30 a heurística proposta obteve a solução ótima dos problemas.

Por fim, na Tabela 4.6, são apresentados o tempo computacional total gasto pela metodologia proposta para cada problema-teste, a redução do tempo em relação ao CPLEX sem variáveis fixadas e a porcentagem de redução do tempo.

| Instância | t(RL + CPLEX) | Redução de Tempo | % de Melhora |
|-----------|---------------|------------------|--------------|
| 20_10 | 1,70 | 2,68 | 61,19 |
| 20_20 | 2,76 | 2,55 | 48,02 |
| 20_30 | 2,42 | 2,34 | 49,16 |
| 30_10 | 0,37 | 0,01 | 2,37 |
| 30_20 | 3,37 | 0,85 | 20,14 |
| 30_30 | 2,86 | 3,14 | 52,33 |
| 40_10 | 1,19 | 1,73 | 59,32 |
| 40_20 | 18,88 | 10,09 | 34,83 |
| 40_30 | 25,64 | 151,10 | 85,49 |
| 50_10 | 17,14 | 13,33 | 43,76 |
| 50_20 | 44,53 | 9,28 | 17,24 |
| 50_30 | 425,40 | 338,48 | 46,76 |
| 60_10 | 16,38 | 35,38 | 68,35 |
| 60_20 | 34,99 | 27,56 | 44,07 |
| 60_30 | 390,05 | 344,65 | 46,91 |
| 70_10 | 88,50 | 16,80 | 15,95 |
| 70_20 | 342,66 | 74,25 | 19,70 |
| 70_30 | 225,53 | 332,00 | 65,41 |
| 75_10 | 169,89 | 208,75 | 59,87 |
| 75_20 | 130,60 | 273,07 | 78,32 |
| 75_30 | 385,33 | 75,59 | 17,75 |
| 100_10 | 82,03 | 51,67 | 61,73 |
| 100_20 | 229,49 | 356,21 | 66,49 |
| 100_30 | 1344,16 | 522,72 | 41,26 |
| 200_10 | 504,83 | 576,51 | 57,57 |
| 200_20 | 1639,10 | 860,04 | 40,97 |
| 200_30 | 8999,93 | 711,16 | 7,81 |

Tabela 4.6: Tempo total da metodologia e redução no tempo comparado ao CPLEX

Com base nos resultados apresentados na Tabela 4.6, pode-se observar que a metodologia proposta gerou uma redução no tempo computacional necessário para a resolução de todos os problemas-teste. O melhor caso ocorreu para o problema-teste 40_30, onde houve uma redução de 151,10 segundos no tempo computacional gasto para sua resolução, melhora de 85,49%. Como pior caso tem-se o problema-teste 30_10 onde a melhora chegou a apenas 2,37%. A metodologia proposta gerou, em média, 44,88% de redução no tempo de resolução dos problemas-teste.

Capítulo 5

Conclusão e Trabalhos Futuros

Este trabalho teve como objetivo desenvolver uma metodologia baseada em Relaxação Lagrangeana e na fixação de variáveis para a redução do esforço computacional necessário para a resolução de Problemas de Seqüenciamento em uma Máquina onde são associados pesos aos *jobs* e, estes apresentam datas de chegada distintas (SMSP). Este problema apresenta várias aplicações práticas, como, por exemplo, é o caso das empresas de manufatura.

Definiu-se, inicialmente, dentre os modelos de Programação Linear Inteira e Inteira Mista apresentados na literatura, um modelo para representar o problema. O modelo adotado foi o formulado com variáveis indexadas no tempo. Este tipo de modelo pode ser utilizado para representar uma variedade de Problemas de Seqüenciamento, mas apresenta uma importante desvantagem, o elevado número de variáveis e restrições presentes no modelo.

Foram criados problemas-teste de várias dimensões, conforme proposto na literatura, e, para validar a metodologia desenvolvida foram feitas comparações entre suas soluções e aquelas geradas utilizando somente o *software* de otimização.

O algoritmo baseado em Relaxação Lagrangeana utilizado, permitiu a geração de bons limites para a solução dos problemas. Os limites inferiores gerados ficaram, em média, a 0,41% do ótimo e, os superiores, a 0,25%. Os limites gerados foram utilizados em um procedimento para fixação de variáveis. Este procedimento procurou fixar variáveis analisando-se os custos reduzidos destas em uma solução Lagrangeana. As variáveis fixadas foram, então, eliminadas do modelo de Programação Inteira, que, por fim, foi resolvido por um *software* de otimização.

Com base nas experiências computacionais feitas, concluímos que a combinação da Relaxação Lagrangeana com a fixação de variáveis provou ser uma ferramenta útil para a melhoria da resolução de Problemas de Seqüenciamento em uma Máquina formulados com indexação no tempo. Os resultados obtidos mostraram, para a maioria dos casos, uma redução significativa no esforço computacional necessário para a resolução dos problemas-teste em comparação com o tempo gasto com a utilização somente do *software* de otimização. A metodologia apresentou, em média, uma redução de 44,88% no tempo computacional necessário para a resolução dos problemas.

Como propostas para trabalhos futuros, outras funções objetivo poderiam ser utilizadas, como, por exemplo, a de minimizar a maior data de término de processamento de um *job* (*makespan*). Restrições como janelas de tempo e tempo de preparação de máquina poderiam ser incluídas na formulação. Pode ser relevante, também, a utilização de outro método de resolução para o Problema Dual Lagrangeano, ao invés do Método do Subgradiente. Poderiam ser utilizados, por exemplo, o Método do Volume ou o Método de Feixes.

Como continuação para este trabalho, serão introduzidas desigualdades válidas à formulação adotada para o SMSPP, baseadas em Sousa e Wolsey [31]. Em uma dada iteração do Método do Subgradiente, uma parte destas desigualdades que violarem a solução do Problema Lagrangeano seria dualizada. O objetivo, com a introdução destas desigualdades, é reduzir ainda mais o *gap* Lagrangeano, permitindo a fixação de um número maior de variáveis.

Esta dissertação deu origem ao artigo *Relaxação Lagrangeana com Fixação de Variáveis Aplicada ao Problema de Seqüenciamento em uma Máquina com Data de Chegada para os Jobs* apresentado no XL SBPO (Simpósio Brasileiro de Pesquisa Operacional) ocorrido do dia 02 a 05 de Setembro de 2008 em João Pessoa - PB.

Referências Bibliográficas

- [1] Akker, J. M. van den; Hoesel, C. P. M. van; Savelsbergh, M. W. P. “A Polyhedral Approach to Single-Machine Scheduling Problems”. *Mathematical Programming*, v. 85, p. 541 – 572, 1999.
- [2] Akker, J. M. van den; Hurkens, C. A. J.; Savelsbergh, M. W. P. “Time-Indexed Formulations for Machine Scheduling Problems: Column Generation”. *INFORMS Journal on Computing*, v. 12, n. 2, p. 111, 2000.
- [3] Allahverdi, A.; Gupta, J. N. D.; Aldowaisan, T. “A Review of Scheduling Research Involving Setup Considerations”. *Omega - The International Journal of Management Science*, v. 27, p. 219 – 239, 1999.
- [4] Arenales, M.; Armentano, V.; Morabito, R.; Yanasse, H. *Pesquisa Operacional para Cursos de Engenharia*. 1ª Edição, Editora Campus - Elsevier, Rio de Janeiro-RJ, 2006.
- [5] Avella, P.; Boccia, M.; D’Auria, B. “Near-Optimal Solutions of Large-Scale Single-Machine Scheduling Problems”. *INFORMS Journal on Computing*, v. 17, n. 2, p. 183 – 191, 2005.
- [6] Backer, K. R. *Introduction to Sequencing and Scheduling*. John Wiley, New York-NY, 1974.
- [7] Barahona, F.; Anibal, R. “The Volume Algorithm: Producing Primal Solutions with a Subgradient Method”. *Mathematical Programming*, v. 87, p. 385 – 399, 2000.
- [8] Błażewicz, J.; Ecker, K. H.; Pesch, E.; Schmidt, G.; Węglarz, J. *Scheduling Computer and Manufacturing Processes*. Springer-Verlag Berlin, 1996.

- [9] Błazewicz, J.; Lenstra, J. K.; Kan, A. H. G. Rinnooy. “Scheduling Subject to Resource Constraints: Classification and Complexit”. *Discrete Appl. Math.*, v. 5, p. 11 – 24, 1983.
- [10] Camerini, P. M.; Fratta, L.; Maffioli, F. “On Improving Relaxation Methods by Modified Gradient Technics”. *Math. Programming Study*, n. 3, p. 26–34, 1975.
- [11] Crama, Y.; Spieksma, F. C. R. “Scheduling *jobs* of Equal Length: Complexity and Facets”. *Mathematical Programming*, v. 72, p. 207 – 227, 1996.
- [12] Fisher, M. L. “The Lagrangian Relaxation Method of Solving Integer Programming Problems”. *Management Science*, v. 50, n. 12, p. 1861, 1981.
- [13] Florentino, H. O. *Relaxação Lagrangeana em Programação Inteira*. Master’s thesis, ICMC - USP, 1990.
- [14] Garey, M. R.; Johnson, D. S. *Computers and Intractability - A guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [15] Geoffrion, A. M. “Lagrangian Relaxation end its Uses in Integer Programming”. *Mathematical Programming Study*, 1974.
- [16] Graham, R. L.; Lawler, E. L.; Lenstra, J. K.; Kan, A. H. G. Rinnooy. “Optimization and approximation in determinstic sequencing and scheluling theory: a survey”. *Ann. Discrete Math.*, v. 5, p. 287–326, 1979.
- [17] Guignard, M. “Lagrangean Relaxation”. *TOP*, v. 11, n. 2, p. 151 – 228, 2003.
- [18] Gupta, S. R.; Smith, J. S. “Algoritms for Single Machine Total Tardiness Scheduling with Sequence Dependent Setups”. *European Journal of Operation Research*, v. 175, p. 722 – 739, 2006.
- [19] Held, M.; Karp, R. M. “The travelling salesman problem and minimum spanning trees”. *Mathematical Programming*, v. 1, p. 6 – 25, 1971.
- [20] Held, M.; Wolfe, P.; Crowder, H. P. “Validation of Subgradient Optimization”. *Mathematical Programming*, v. 6, p. 62 – 88, 1974.
- [21] Jouglet, A.; Savourey, D.; Carlier, J.; Baptiste, P. “Dominance-based Heuristics for One-machine Total Cost Scheduling Problems”. *European Journal of Operational Research*, v. 184, p. 879 – 899, 2006.

- [22] Lasdon, L. S. *Optimization Theory for Large Systems*. Dover Publications, INC, 2002.
- [23] Lemarechal, C. “An Algorithm for Minimizing Convex Functions”. *In IFIP’74 Congress*, p. 552 – 556, 1974.
- [24] Lenstra, J. K.; Kan, A. H. G. Rinnooy; Brucker, P. “Complexity of Machine Scheduling Problems”. *Annals of Discrete Mathematics*, v. 1, p. 343 – 362, 1977.
- [25] Lorena, L. A. N.; Senne, E. L. F. “A Column Generation Approach to Capacitated P-median Problems”. *Computers and Operations Research*, v. 31, p. 863 – 876, 2004.
- [26] Lucena, A. “Non Delayed Relax-and-Cut Algorithms”. *Annals of Operation Research*, v. 140, p. 375 – 410, 2005.
- [27] Manne, A. S. “On the Job-shop Scheduling Problem”. *Operations Research*, v. 8, p. 219 – 223, 1960.
- [28] Minoux, M. *Mathematical Programming: Theory and Algorithms*. Wiley, 1986.
- [29] Mitchell, J. E. “Fixing variables and generating classical cutting planes when using an interior point branch and cut method to solve integer programming problems”. *European Journal of Operation Research*, v. 97, p. 139 – 148, 1997.
- [30] Nemhauser, G.L.; Wolsey, L.A. *Integer and Combinatorial Optimization*. Wiley, 1988.
- [31] Sousa, J. P.; Wolsey, L. A. “A Time Indexed Formulation of Non-Preemptive Single Machine Problems”. *Mathematical Programming*, v. 54, p. 353 – 367, 1992.
- [32] Thorsteinsson, E. S.; Ottosson, G. “Linear Relaxations and Reduced-Cost Based Propagation of Continuous Variable Subscripts”. *Annals of Operation Research*, v. 115, p. 15 – 29, 2002.
- [33] Wagner, H. M. “An Integer Programming Model for Machine Scheduling”. *Naval Research Logistics Quarterly*, v. 6, p. 131 – 140, 1959.