

Tiago Carvalho da Costa

*Heurística Simulated Annealing aplicado a
um problema de Job Shop Scheduling em
um ambiente industrial*

Campos dos Goytacazes/RJ

2014

Tiago Carvalho da Costa

Heurística Simulated Annealing aplicado a um problema de Job Shop Scheduling em um ambiente industrial

Monografia apresentada ao Curso de Graduação em Ciência da Computação da Universidade Estadual do Norte Fluminense Darcy Ribeiro como requisito para obtenção do título de Bacharel em Ciência da Computação, sob orientação do Prof. Fermín Alfredo Tang Montané, DrSc.

Orientador: Fermín Alfredo Tang Montané, DrSc.

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO

Campos dos Goytacazes/RJ

2014

Tiago Carvalho da Costa

Heurística Simulated Annealing aplicado a um problema de Job Shop Scheduling em um ambiente industrial

Monografia apresentada junto ao Curso de Ciência da Computação, da Universidade Estadual do Norte Fluminense Darcy Ribeiro – Campos / RJ, como requisito para obtenção do título de Bacharel em Ciência da Computação. Orientadora: Prof. Dr. Fermín Alfredo Tang Montané.

Aprovado em 30/01/2014.

COMISSÃO EXAMINADORA

Prof. Dr. Fermín Alfredo Tang Montané
Orientador - Universidade Estadual do
Norte Fluminense Darcy Ribeiro

Prof. Dr. Luis Antônio Rivera Escriba
Universidade Estadual do Norte Fluminense
Darcy Ribeiro

Prof^ª. Dr^ª. Gudelia Guillermina Morales de
Arica
Universidade Estadual do Norte Fluminense
Darcy Ribeiro

“Para tudo há um tempo, para cada coisa há um momento debaixo dos céus”

Eclesiastes 3:1

AGRADECIMENTOS

Agradeço primeiramente à Deus por ter me dado a sabedoria e discernimento nas minhas escolhas. Aos meus pais, Eliane e Manoel por todo incentivo e carinho, sem o apoio deles não conseguiria chegar onde cheguei, ao meu irmão Gustavo pelo apoio de disponibilidade em sempre me ajudar. Agradeço a todos os professores que contribuíram de alguma forma para a minha formação, em especial ao professor Fermín Alfredo Tang Montané pela paciência e por todo o conhecimento passado desde a iniciação científica até a conclusão desde trabalho. A todos amigos de república, companheiros de turma, todos aqueles que estiveram presentes nos momentos bons e no momentos difíceis dessa caminhada. Enfim, agradeço a todos que direta ou indiretamente contribuíram para essa vitória.

Lista de Figuras

1	Representação pelo Grafico de Gantt (CUNHA, 2010)	15
2	Representação do problema usando grafo disjuntivo (CUNHA, 2010)	16
3	Representação da solução usando grafo disjuntivo orientado (CUNHA, 2010)	17
4	Pseudo-código da heurística <i>Simulated Annealing</i>	21
5	Variação da temperatura	22
6	Probabilidade de aceitação de uma solução pior	23
7	Ilustração básica dos processos de uma siderúrgica	25
8	Estrutura do problema abordado	32
9	Etapas da perturbação da solução	37
10	Grafico de Gant da Solução (LISA, 2010)	39
11	Avaliação do Makespan	41
12	Avaliação do tempo de execução do algoritmo	41

Lista de Tabelas

1	Capacidade dos convertedores	28
2	Capacidade dos fornos panela	28
3	Características de produção dos tipos de aço	30
4	Parâmetros de roteirização na produção do aço	31
5	Arquivo de Dados de Entrada	38
6	Tabela de resultados	39
7	Medida de Qualidade da Solução	40
8	Avaliação dos parâmetros da heurística SA	45

Abstract

A Job Shop Scheduling problem is a classical optimization problem which consists to organize the processing of a number of tasks on a number of machines, in this work an algorithm was developed and implemented using Java language to model a data structure for the problem and search a good quality solution using Simulated Annealing heuristic.

In the steel industry an important sector in the production of steel is the steel plant, is where the cast iron is turned into steel. This environment has several features, such as converters, ovens pots, bridges, cars and other transports. According to the type of steel to be produced, the resource sequence to be used and the processing time of each resource may vary.

The objective of this work is to improve the use of these resources, it is necessary to plan and control optimally all resources involved in the production. In this work we propose a solution using a heuristic method for finding a good quality solution. This work is limited to minimize the makespan, Scheduling problems are classified with complexity NP-Hard, which implies high computational effort even for small problems considered.

Resumo

Um problema de *Job Shop Scheduling* é um problema clássico de otimização que consiste em organizar o processamento de certo número de tarefas em um certo número de máquinas, neste trabalho foi desenvolvido um algoritmo na linguagem Java para modelar uma estrutura de dados para o problema e buscar uma solução de boa qualidade utilizando a heurística *Simulated Annealing*.

Em uma siderurgia um importante setor na produção do aço é a aciaria, é o lugar onde o ferro-gusa (matéria-prima) é transformado em aço. Este ambiente possui diversos recursos, tais como, convertedores, fornos panelas, pontes, carros de transporte entre outros. De acordo com o tipo de aço a ser produzido, a sequência de recursos a ser utilizada, bem como os tempos de processamento de cada recurso podem variar.

O objetivo deste trabalho é aprimorar o uso desses recursos, para isso é necessário planejar e controlar de maneira ótima todos os recursos envolvidos na produção. Neste trabalho é proposta uma solução utilizando um método heurístico para encontrar uma solução de boa qualidade. Este trabalho se limita a minimização do *makespan*, problemas de *Scheduling* são classificados com complexidade *NP-Hard*, o que implica em um alto esforço computacional mesmo para problemas considerados pequenos.

Palavras Chaves: *Job Shop Scheduling*; Sequenciamento; Heurística; *Simulated Annealing*.

Sumário

Lista de Figuras	2
Lista de Tabelas	3
Abstract	4
Resumo	5
1 Introdução	9
1.1 Introdução	9
1.2 Objetivo e Justificativa	9
1.3 Estrutura do Trabalho	10
2 Fundamentos teóricos	11
2.1 Introdução ao Sequenciamento de Tarefas	11
2.2 Classificação das Máquinas	12
2.3 Variantes do Problema de <i>Scheduling</i>	12
2.3.1 <i>Open-Shop Scheduling</i>	12
2.3.2 <i>Flow-Shop Scheduling</i>	13
2.3.3 <i>Job-Shop Scheduling</i>	13
2.4 Notação	13
2.5 Representação	15
2.5.1 Gráfico de Gantt	15
2.5.2 Grafos Disjuntivos	15

	7
2.6	Métodos de Solução 17
2.7	O problema de Sequenciamento de Tarefas e suas Aplicações 18
3	<i>Simulated Annealing</i> 19
3.1	Introdução 19
3.2	O Algoritmo de Metrópolis 19
3.3	O Algoritmo <i>Simulated Annealing</i> 20
4	O processo de fabricação do aço 24
4.1	O Processo 24
4.2	O Ambiente Aciaria 26
5	Modelagem do problema 29
5.1	Componentes do problema de <i>Job-Shop Scheduling</i> 29
5.2	O Problema Abordado 31
5.3	Modelagem do Algoritmo - Classes Java 32
5.3.1	Classe Operação 32
5.3.2	Classe Máquina 33
5.3.3	Classe Tarefa 33
5.3.4	Classe Utils 34
5.3.5	Classe Simulated Annealing 34
5.3.6	Classe Processamento 34
5.3.7	Classes Auxiliares 35
6	Resultados e Discussões 36
6.1	A Heurística 36
6.1.1	Solução Inicial 36
6.1.2	Perturbação da Solução 36

	8
6.2 Entrada de dados	37
6.3 Resultado Final	38
6.4 Qualidade da Solução	40
7 Conclusão	42
7.1 Trabalhos Futuros	42
Referências Bibliográficas	43
ANEXO I	45

1 *Introdução*

1.1 *Introdução*

Durante a Revolução Industrial ocorrida no século XVIII, um assunto que começou a se destacar e se tornar muito importante no ambiente dos processos industriais de produção foi o melhor aproveitamento do tempo e dos recursos (mão de obra, matéria-prima, máquinas), reconheceram que somente uma exploração eficiente dos fatores de produção iria garantir o máximo retorno do capital investido. A necessidade de técnicas avançadas de gestão de tarefas nas indústrias é cada vez mais crítica. O objetivo do planejamento das operações de produção de uma fábrica, por exemplo, é gerar um movimento coordenado onde a procura é satisfeita a tempo e com baixos custos. O *Scheduling* tem como objetivo o melhor aproveitamento dos recursos no tempo necessário para executar um conjunto de tarefas, por isso têm sido desenvolvidas modernas técnicas e ferramentas computadorizadas para um uso otimizado dos recursos e do tempo. (BAKER; TRIESCH, 1974)

Embora esse campo de pesquisa tenha obtido muitos avanços nos últimos anos, o *Job Shop Scheduling* continua a ser um problema complexo pelo fato de não se enquadrar num modelo genérico, suas características e particularidades variam de caso para caso, e em um mesmo caso pode ocorrer um grande número de variações.

1.2 *Objetivo e Justificativa*

O objetivo deste trabalho é propor uma modelagem usando a linguagem Java para obter uma solução para um problema de planejamento de tarefas em um ambiente industrial utilizando a heurística *Simulated Annealing*. Um ambiente industrial que depende de máquinas para executar uma série de operações pode fazer com que durante a produção muito tempo seja desperdiçado, para diminuir esse desperdício de tempo e melhorar o aproveitamento das máquinas é necessário realizar um complexo planejamento de como

as máquinas serão utilizadas.

1.3 Estrutura do Trabalho

O presente trabalho tem a seguinte estrutura.

No capítulo 2, é feito uma revisão bibliográfica referente aos problemas de sequenciamento de tarefas, incluindo a sua classificação. É mostrado como um problema é modelado e como sua solução pode ser representada, ainda são descritos os métodos de solução existentes e algumas aplicações para o problema.

No capítulo 3 é feita uma revisão bibliográfica do método *Simulated Annealing*, que é um método heurístico para encontrar “boas” soluções aproximadas para problemas de otimização combinatória.

No capítulo 4 é descrito de forma resumida como é o processo de fabricação do aço, em especial o ambiente aciaria, e ainda é especificados os seus parâmetros para o problema de *Job Shop Scheduling*.

No capítulo 5 são definidos os dados que envolvem o problema de *Job-Shop Scheduling*, ou seja, a descrição das máquinas e das tarefas. Ainda são definidas nesse capítulo as classes Java que foram modeladas como estrutura de dados para um problema de *Job Shop Scheduling*.

No capítulo 6 são discutidos os aspectos do desenvolvimento do algoritmo e a avaliação da solução encontrada a partir de um método heurístico.

2 *Fundamentos teóricos*

Este capítulo descreve os principais elementos teóricos utilizados no desenvolvimento desta pesquisa, é descrito neste capítulo um pouco da introdução histórica do problema de sequenciamento de tarefas.

2.1 Introdução ao Sequenciamento de Tarefas

A necessidade de técnicas de gestão das tarefas, tanto nos serviços quanto nas indústrias é cada vez mais crítica. Segundo ROLDÃO (1995), o planejamento de operações é uma área em que se têm obtido mais progresso e cuja utilização traz mais vantagens ao gestor.

O Sequenciamento de Tarefas, ou *Scheduling*, também chamado por alguns autores de Calendarização, pode ser definido como um processo de decisão para uma distribuição de recursos ao longo do tempo para realizar um conjunto de tarefas, sujeito a restrições (datas limites, duração e pendência das operações, tempos de movimentação e preparação, disponibilidade e partilha de recursos) e preferências. (PAPE, 1994).

Nos problemas de *Scheduling*, de um modo geral, assume-se a necessidade de executar certo número de tarefas, onde cada tarefa possui um conjunto de operações (na literatura a operação é chamada de *job*, então cada tarefa possui um conjunto de *jobs*), que são executadas em uma sequência determinada, e usa certo número de máquinas. O processamento de uma operação requer o uso de uma máquina particular durante certo intervalo de tempo e cada máquina só pode processar uma operação de cada vez, a soma dos tempos de processamento de todas as operações é o tempo total de processamento de todas as tarefas, esse valor é chamado de *makespan*.

Os dados para um problema de *Scheduling* de forma geral pode ser posto da seguinte forma:

- Existe um conjunto de k operações, $O = \{O_1, O_2, \dots, O_k\}$

- Existem um conjunto de m máquinas, $P = \{ P_1, P_2, \dots, P_m \}$
- Pode haver um conjunto de s Recursos, $R = \{ R_1, R_2, \dots, R_s \}$

2.2 Classificação das Máquinas

Segundo BLAZEWICZ et al. (1994) as máquinas podem ser classificadas de acordo com as funções que desempenham, sendo essa classificação feita do seguinte modo:

- **Máquinas Paralelas:** Se todas as máquinas do conjunto P definido na seção 2.1 desempenham as mesmas funções. Elas podem ser distinguidas pelas suas velocidades de processamento.
- **Máquinas Dedicadas:** Cada máquina do conjunto P definido na seção 2.1 é especializada na execução e uma operação específica.

2.3 Variantes do Problema de *Scheduling*

Conforme explica CUNHA (2010), dependendo da classificação de um dado problema, existe também uma classificação dos *Jobs Schedules*, que é a forma como será tratado esse problema, ou seja, é a forma como os *Jobs* serão escalonados nas máquinas que os executam.

Existem basicamente três tipos de escalonamento: *Open-Shop*, *Flow-Shop* e *Job-Shop*. Uma vez identificado o tipo de problema, pode-se escolher uma destas estratégias para abordá-lo, abaixo segue uma rápida definição de cada um desses três tipos básicos de *Scheduling*.

2.3.1 *Open-Shop Scheduling*

Nesse tipo de problema as operações de cada uma tarefa não possuem uma sequência para serem executadas. Dessa forma, as operações são totalmente independentes, não existindo restrições quanto à ordem de execução. Através desse tipo de escalonamento, as tarefas precisam ser executadas por todas as máquinas e isso pode ocorrer em qualquer ordem, objetivando, também, minimizar o tempo de execução da tarefa mais demorada, o que é chamado *makespan*.

2.3.2 *Flow-Shop Scheduling*

Neste tipo de problema existe uma ordem pré-estabelecida para execução das operações de cada uma tarefa, ou seja, existe uma sequência de execução que deve ser respeitada. Neste caso que ao invés de se projetar o escalonamento de operações, faz-se o planejamento das tarefas em relação ao uso das máquinas.

2.3.3 *Job-Shop Scheduling*

No *Job-Shop Scheduling* cada operação é processada por uma máquina específica, isso restringe a execução de uma dada operação a um só tipo de máquina, e pode existir um caso de ter mais de uma máquina do mesmo tipo (PINEDO, 2002). Esse tipo de escalonamento permite que se estabeleçam diversas escalas, vez que pode haver flexibilidade quanto à ordem de processamento das operações, bem como, quanto à ordem de utilização das máquinas, que deve objetivar o tempo mínimo de ociosidade e o mínimo tempo total da execução (*makespan*). Essa medida de qualidade, conhecida como *makespan*, não é a única existente, porém é o critério mais simples e mais utilizado. O objetivo implica em buscar arranjos de operações cuja ordem de execução em cada máquina leve o menor tempo possível.

O *Job Shop Scheduling* é composto por um conjunto de Tarefas ($T = \{t_1, t_2, \dots, t_p\}$), onde cada tarefa é composto por um conjunto de operações ($j_i = \{o_{i1}, o_{i2}, \dots, o_{ip}\}$) e cada operação tem seu tempo de processamento, todos os tempos de processamento de cada operação formam um conjunto de tempos de processamento ($t_i = \{t_{i1}, t_{i2}, \dots, t_{ip}\}$), onde cada tempo pode ser identificado em relação as respectivas operações, assim, t_{ip} é o tempo de processamento da p -ésima operação do i -ésimo *job*. Em termos de complexidade de processamento o *Job Shop* corresponde ao problema de *Scheduling* mais genérico e mais complexo.

2.4 Notação

Job-Shop (J_m) : O J_m representa um *Job Shop Scheduling* com m máquinas.

Tempo de Processamento (P_{ij}) : O tempo de processamento P_{ij} representa o tempo de processamento da operação j na máquina i . O subscrito i pode ser omitido quando o tempo de processamento do i não depende da máquina ou se ele é processado por apenas uma máquina.

Tempo de início (r_j) : O tempo de início do processamento da operação, representado pela letra r_j , é o tempo (data) que a operação chega no sistema, indica o tempo que a operação pode começar a ser processado por uma máquina específica.

Tempo de fim (d_j) : O tempo de fim d_j da operação j representa o tempo de finalização do processamento da operação j , o tempo de fim é chamado em algumas literaturas como tempo de pronto.

Makespan (C_{max}) : O *Makespan* pode ser chamado de tempo de fabricação, é o tempo final de processamento da última operação, e é definido como $\max(C_1, \dots, C_n)$, onde C_1 é o tempo de fim da operação 1 e C_n é o tempo de fim da n -ésima operação, sendo assim, o *makespan* é o valor máximo do conjunto (C_1, \dots, C_n) , é o tempo de fim da operação mais demorada. O critério adotado para avaliar uma solução para o problema consiste em minimizar o *makespan*.

Para o caso de Makespan mínimo ($J_m \parallel C_{max}$) a seguinte programação matemática pode ser utilizada.

Minimize C_{max}

sujeito a

$$y_{kj} - y_{ij} \geq p_{ij} \quad \forall (i, j) \rightarrow (k, j) \in A \quad (1)$$

$$C_{max} - y_{ij} \geq p_{ij} \quad \forall (i, j) \in N \quad (2)$$

$$y_{ij} - y_{i1} \geq p_{i1} \quad \text{ou} \quad y_{i1} - y_{ij} \geq p_{ij} \quad \forall (i, 1) \in (i, j) = 1, \dots, m \quad (3)$$

$$y_{ij} \geq 0 \quad (4)$$

onde,

y_{ij} é o tempo inicial de processamento do *Job* j na máquina i ;

p_{ij} é a duração do *Job* j na máquina i

N é o conjunto de todas as operações (i, j)

A é conjunto de todas as restrições de rotas $(i, j) \rightarrow (k, j)$

m é o número total de máquinas

2.5 Representação

A solução para o problema de *Job-Shop Scheduling* pode ser representada de duas formas: Gráfico de Gantt e Grafos Disjuntivos.

2.5.1 Gráfico de Gantt

Desenvolvido em 1917 pelo engenheiro mecânico Henry Gantt, o Gráfico de Gantt representado na figura 1 é usado para representar uma solução para o problema de *Job Shop Scheduling*, ele exibe um possível sequenciamento de execução das operações por parte das máquinas.

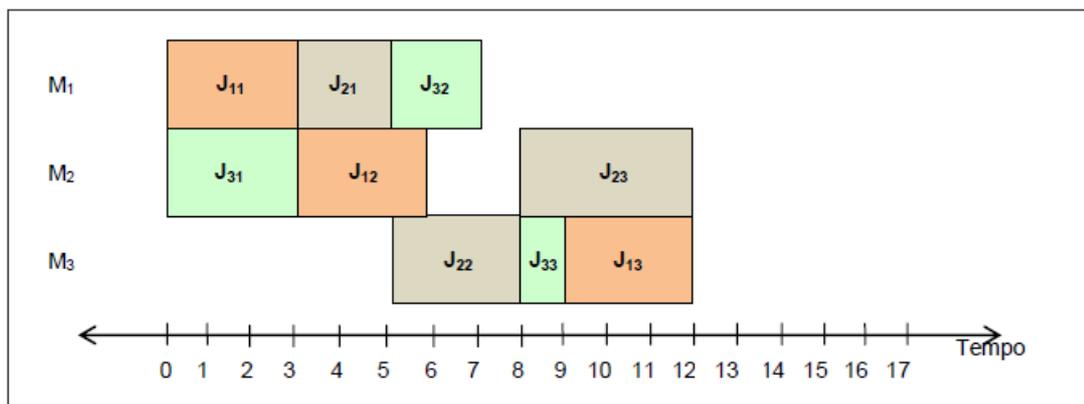


Figura 1: Representação pelo Gráfico de Gantt (CUNHA, 2010)

Nesse gráfico o eixo vertical representa as máquinas, o eixo horizontal representa a linha do tempo, e os retângulos representam as operações, eles são encaixados no gráfico de modo que o seu tempo de execução fique marcado na linha do tempo. As tarefas são identificadas pela cor do retângulo, por exemplo, a tarefa 1 está representado pela cor laranja, onde J_{11} representa a operação 1 da tarefa 1, o J_{12} representa a operação 2 da tarefa 1 e o J_{13} representa a operação 3 da tarefa 1. Com base na representação da solução da figura 1 o *makespan*, ou seja, o tempo total de processamento das operações é igual a 12 unidades de tempo.

2.5.2 Grafos Disjuntivos

Grafos Disjuntivos podem ser usados tanto para modelar um problema de *Scheduling*, quanto para representar sua solução. A idéia principal dele consiste em mostrar a solução do problema destacando o uso das máquinas, e respeitando a precedência das operações.

Representação do Problema

Um grafo parcialmente orientado, constituído por outros grafos pode ser usado como uma representação que reúne as características mais relevantes para o projeto de escalonamento de um problema de *Job Shop Scheduling*. Uma representação do problema de *Job Shop Scheduling* pode ser visualizada pelo Grafo Disjuntivo ilustrado na figura 2

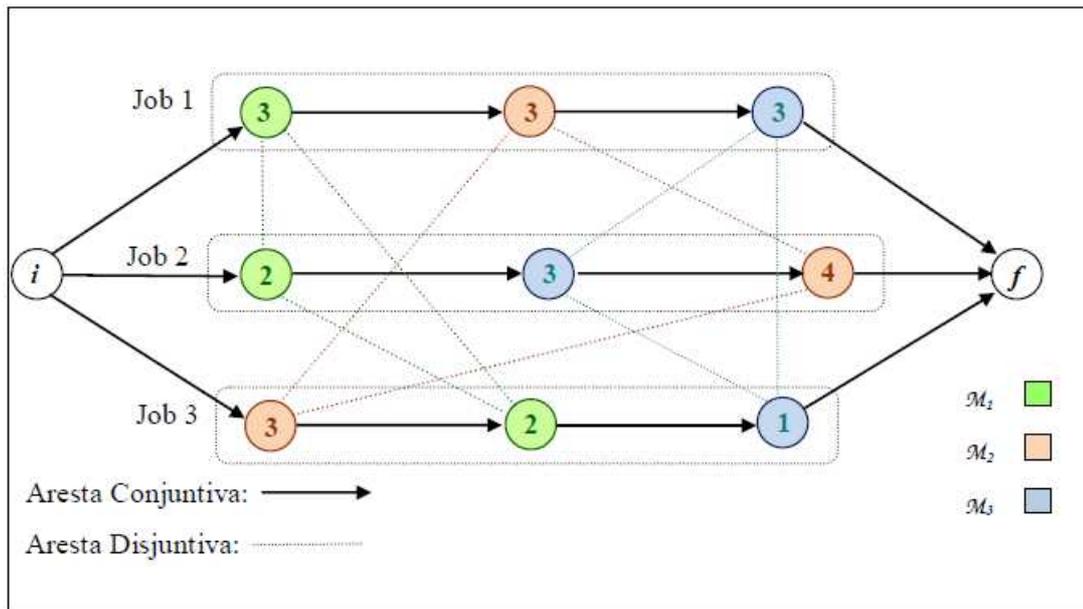


Figura 2: Representação do problema usando grafo disjuntivo (CUNHA, 2010)

As operações de uma mesma tarefa estão interligadas por uma relação de dependência de prioridade de execução. Esta ligação é representada por arestas horizontais orientadas em um único sentido, as quais formam o grafo conjuntivo e elas não podem sofrer modificações de sentido. Cada operação deve ser executada por uma máquina diferente, as cores de cada operação indica em qual máquina ela é executada, a interconexão não orientada forma o grafo disjuntivo, e relaciona as operações que são executados pela mesma máquina, o número representado em cada operação na figura acima indica o tempo de processamento da operação.

Representação da Solução

Uma possível solução para o problema de *Job Shop Scheduling* pode ser representada por uma simples orientação do grafo disjuntivo, como mostra a Figura 3, lembrando que, somente as arestas disjuntivas podem sofrer modificações, pois elas indicam o escalonamento da máquina, e as arestas conjuntivas representam a restrição de precedência de operações.

Realizar um escalonamento das operações constituintes significa buscar uma solução para o problema, esse escalonamento das operações possui uma certa flexibilidade, pode se encontrar diversas possibilidades de arranjo das operações, mas o grafo disjuntivo deve se manter acíclico, pois uma operação só pode ser executada por uma máquina apenas uma vez.

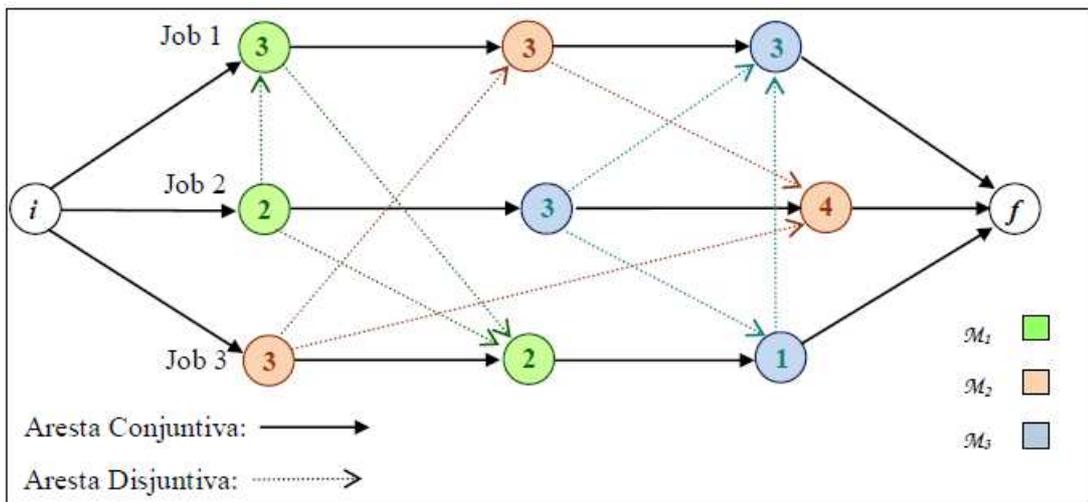


Figura 3: Representação da solução usando grafo disjuntivo orientado (CUNHA, 2010)

2.6 Métodos de Solução

O problema de *Job Shop Scheduling* é classificado como NP-difícil, ou seja, é um problema para o qual não existem algoritmos que o resolva em tempo polinomial. Trata-se de um “Problema de Otimização Combinatória” (SOUZA, 2001). Os algoritmos de solução podem ser classificados em duas categorias: MStemáticos e heurísticos.

- **Modelos Matemáticos:** Trata-se de modelos de programação inteira mista, resolvidos pelos métodos *Branch and Bound* ou *Branch and Cut* (LAND; DOIG, 1960). Modelos Matemáticos enfatizam a obtenção de resultados ótimos em função de um critério de desempenho. Este pode ser, por exemplo, a minimização dos tempos de produção. Dependendo da complexidade do problema a ser resolvido, o cálculo da solução ótima pode ser computacionalmente inviável.
- **Modelos Heurísticos:** Trata-se de modelos baseados em regras práticas de escalonamento que enfatiza a obtenção de “boas” soluções aproximada da solução ótima. Os modelos heurísticos são caracterizados por obter uma solução aproximada em tempos de computação viáveis.

2.7 O problema de Sequenciamento de Tarefas e suas Aplicações

Embora o nome sequenciamento de tarefas parece sugerir que o problema seja aplicado no ramo de produção industrial, ele ocorre em variados contextos, pode ser um ambiente de aplicação do problema, por exemplo, a distribuição de médicos e enfermeiros de um hospital, turmas e professores em uma escola, navios em um porto, cidades e caixeiros viajantes, etc. (REIS, 1996)

3 *Simulated Annealing*

Neste capítulo é feita uma revisão bibliográfica do método heurístico *Simulated Annealing*, esse método permite encontrar “boas” soluções aproximadas para problemas de otimização combinatória em tempos computacionais viáveis.

3.1 Introdução

Simulated Annealing é um algoritmo heurístico de busca local proposto inicialmente por KirkPatrick et al. (1983) que faz analogia à um processo de arrefecimento termodinâmico para evitar e escapar de mínimos locais. Os estados de um sólido são vistos como sendo análogas às soluções, visto que a energia do sólido refere-se a uma função objetivo. Apesar da analogia ao processo físico ser complicado, a implementação de algoritmos baseados neste método é relativamente simples. Eles mostraram que conceitos da mecânica estática para simular processos de recozimento, proposto por Metropolis et al. (1953), podiam ser estendidos para resolver problemas de otimização em geral, especialmente problemas de otimização combinatória.

3.2 O Algoritmo de Metrópolis

Na física, o recozimento, também chamado de *annealing*, é um tratamento térmico para obter estados de baixa energia. Para isso, primeiramente é feito o aquecimento do material até uma temperatura máxima T_0 , próximo ao ponto de fusão do sólido (estado onde as partículas estão arranjadas de forma praticamente aleatória), em seguida, reduz-se lentamente a temperatura do material (*annealing*) até que as partículas se reorganizem de forma estruturada proporcionando um estado próximo ao de mínima energia. O estado fundamental do sólido somente é obtido se a temperatura máxima for suficientemente alta e o resfriamento for suficientemente lento.

Em 1953, Metrópolis et al. estudaram certos problemas da mecânica estatística dos materiais sólidos. Este estudo ficou conhecido como Processos Físicos de Resfriamento destes materiais. Eles observaram que alguns materiais sólidos submetidos à alta temperatura inicial e depois tendo esta temperatura lentamente reduzida, sofriam mudanças em seu estado de energia, até que estes materiais atingissem o equilíbrio térmico, isto é, o estado de energia globalmente mínimo. Segundo Metrópolis, uma imperfeição neste princípio ocasionaria a obtenção de um estado de energia “localmente mínimo”. (FIGUEIRA, 2002)

3.3 O Algoritmo *Simulated Annealing*

O método *Simulated Annealing* (SA) apresenta uma estratégia muito similar ao princípio de melhoramento iterativo utilizado por muitas heurísticas para solução de problemas de otimização combinatória. Porém esse método possui uma característica diferenciada bem significativa que é permitir perturbações estocasticamente controladas, ocasionando o aumento do valor da função objetivo em qualquer iteração, entretanto esse aumento indesejável do valor da função objetivo pode proporcionar soluções melhores nas próximas iterações.

A ideia do melhoramento iterativo do método *Simulated Annealing* está sujeita a perturbações aleatórias, semelhantes às mudanças de localizações verificadas em uma partícula no processo de mecânica estatística, e que resulta em um aumento de energia nos processos físicos de resfriamento de materiais. Essa filosofia do *Simulated Annealing* evita que a busca fique bloqueado em um ótimo local.(FIGUEIRA, 2002)

Como o algoritmo *Simulated Annealing* permite aceitar soluções piores que a solução corrente, algumas regras deverão ser respeitadas:

- Dada uma temperatura fixa, durante um ciclo de execução do algoritmo, a magnitude da piora de uma solução influencia diretamente na sua aceitação. Quanto maior o grau de piora da solução, menores são as chances do algoritmo aceitar essa solução.
- A medida que o parâmetro de temperatura diminui, o algoritmo torna-se menos tolerante a aceitar soluções de pior qualidade. Ou seja, a probabilidade do algoritmo aceitar um solução ruim, diminui a medida que a temperatura cai.

As duas regras são implementadas mediante a seguinte formula:

$$r < e^{-\Delta/T}$$

Onde:

r é um número aleatório $\in [0,1]$;

Δ é o grau de piora da solução;

T é a temperatura

Na figura 4 encontra-se o algoritmo da heurística *Simulated Annealing*.

```

1  procedimento simulated_annealing (f(.), N(.),  $\alpha$ ,  $T_0$ ,  $Iter_{max}$ , s)
2   $s^* \leftarrow s$ 
3   $Iter \leftarrow 0$ 
4   $T \leftarrow T_0$ 
5   $T_{min} \leftarrow valor$ 
6  enquanto ( $T > T_{min}$ ) faça
7      enquanto ( $Iter < Iter_{max}$ ) faça
8           $Iter = Iter + 1$ 
9          Gerar  $s' \in N(s)$ 
10          $\Delta = f(s') - f(s)$ 
11         se ( $\Delta < 0$ ) entao
12              $s \leftarrow s'$ 
13             se ( $f(s') < f(s^*)$ ) entao  $s^* \leftarrow s'$ 
14         senao
15             Gerar  $r \in [0,1]$ 
16             se ( $r < e^{-\Delta/T}$ ) entao  $s \leftarrow s'$ 
17         fim se
18     fim enquanto
19      $T = \alpha * T$ 
20      $Iter \leftarrow 0$ 
21 fim enquanto
22  $s \leftarrow s^*$ ;
23 Retornar s;
24 fim simulated_annealing

```

Figura 4: Pseudo-código da heurística *Simulated Annealing*

Os parâmetros considerados são:

s : (solução inicial) solução corrente

s' : solução vizinha

s^* : melhor solução encontrada

$f(\cdot)$: função de avaliação

$f(s)$: função de avaliação da solução s

$N(\cdot)$: vizinhança

$N(s)$: conjunto de soluções vizinhas a s , vizinhança de s

T : temperatura corrente no ciclo de resfriamento

T_0 : temperatura inicial

α : constante de resfriamento

$Iter$: iteração corrente

$Iter_{max}$: máximo de iterações no ciclo de resfriamento

O algoritmo simula ciclos de resfriamento, a temperatura cai de acordo com um fator de resfriamento pré-definido. Como podemos ver na figura 5.

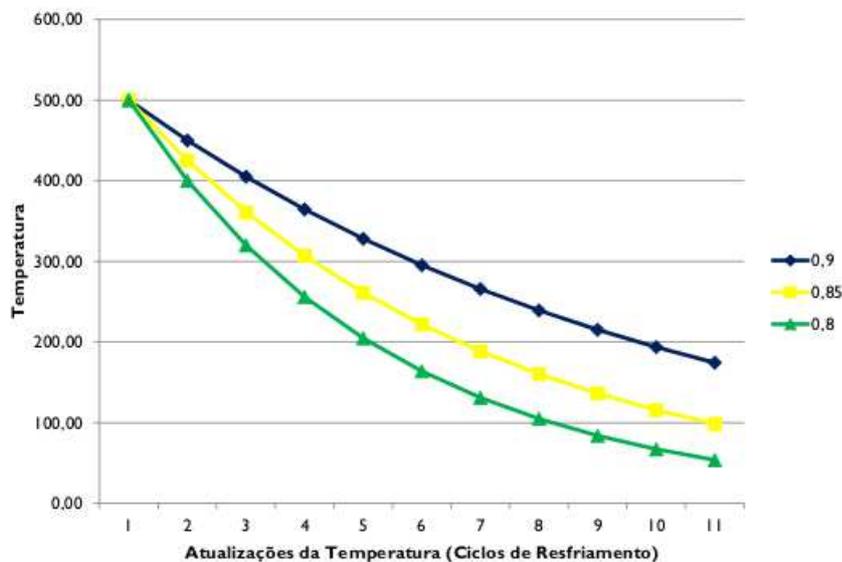


Figura 5: Variação da temperatura

A probabilidade de aceitação de uma solução pior diminui com a temperatura e com o grau de piora da nova solução. Como é mostrado na figura 6.

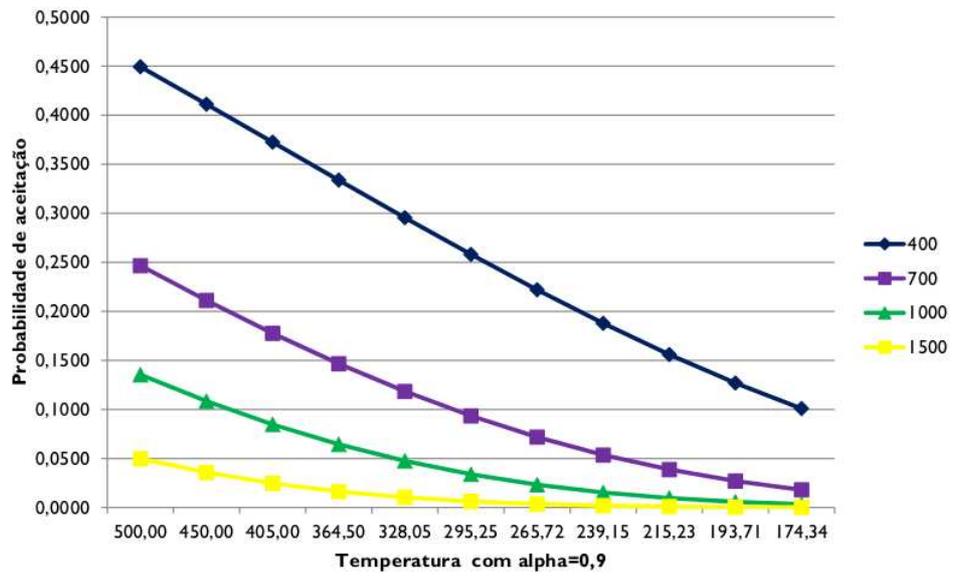


Figura 6: Probabilidade de aceitação de uma solução pior

4 *O processo de fabricação do aço*

Neste capítulo é descrito de forma resumida como é o processo de fabricação do aço, em especial o ambiente aciaria, e ainda é especificados os seus parâmetros para o problema de *Job-Shop Scheduling*.

4.1 O Processo

Com base no relatório encontrado em (UFPR, 2012), o aço é um produto siderúrgico definido como liga metálica composta principalmente de ferro e pequenas quantidades de carbono. O processo siderúrgico pode ser dividido em 4 grandes partes, que são descritas abaixo:

1. Preparo das matérias-primas (coqueira e sintetização).
2. Produção de gusa (alto-forno).
3. Produção de aço (aciaria).
4. conformação mecânica (laminação)

A figura 7 mostra uma visão geral das etapas do processo de fabricação do aço, começando pelo alto forno até chegar ao processo de laminação.

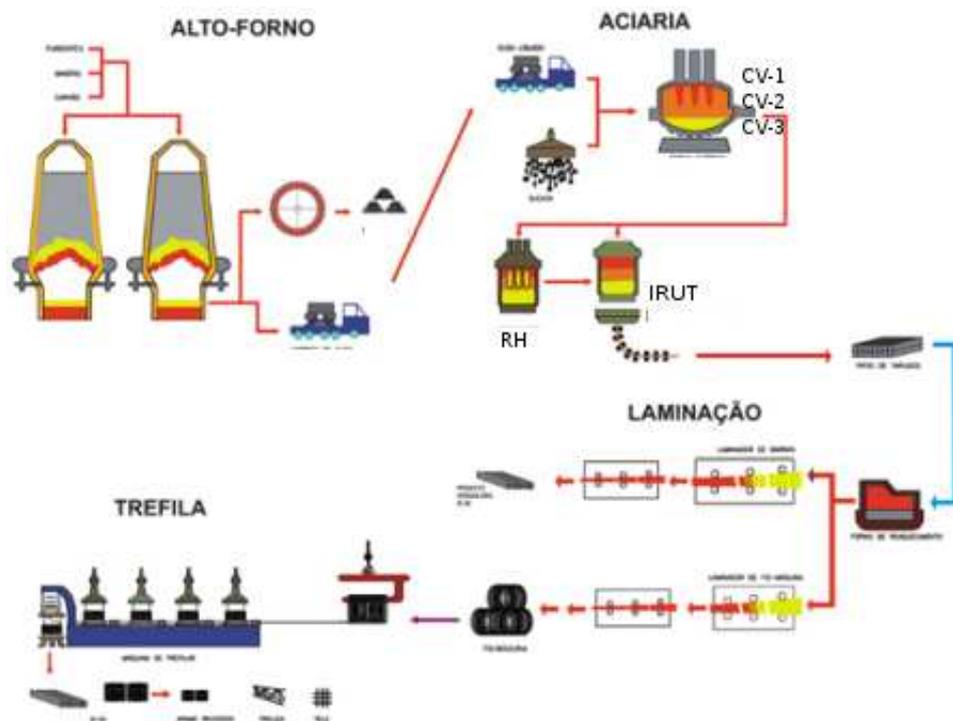


Figura 7: Ilustração básica dos processos de uma siderúrgica

Preparo das Matérias-primas

As matérias-primas necessárias para a obtenção do aço são as seguintes: minério de ferro e o carvão mineral. Ambos não são encontrados puros na natureza, sendo necessário então um preparo nas matérias-primas de modo a reduzir o consumo de energia e aumentar a eficiência no processo.

Coqueira e Sintetização

A coqueificação ocorre a uma temperatura de 1300°C em ausência de ar durante um período de 18 horas, onde ocorre a liberação de substâncias voláteis. O produto resultante desta etapa, o coque, é um material poroso com elevada resistência mecânica, alto ponto de fusão e grande quantidade de carbono.

Na sinterização, a preparação do minério de ferro é feita cuidando-se da granulometria, visto que os grãos mais finos são indesejáveis pois diminuem a permeabilidade do ar na combustão, comprometendo a queima. Para solucionar o problema, adicionam-se materiais fundentes (calcário, areia de sílica ou o próprio sinter) aos grãos mais finos. Com a composição correta, estes elementos são levados ao forno onde a mistura é fundida. Em seguida, o material resultante é resfriado e britado até atingir a granulometria desejada

(diâmetro médio de 5mm). O produto final deste processo é denominado de sinter.

Alto Forno

Esta parte do processo de fabricação do aço consiste na redução do minério de ferro, utilizando o coque metalúrgico e outros fundentes, que misturados com o minério de ferro são transformados em ferro gusa. A reação ocorre no equipamento denominado Alto Forno, e constitui uma reação exotérmica. Após a reação, o ferro gusa na forma líquida é transportado nos carros-torpedos (vagões revestidos com elemento refratário) para uma estação de dessulfuração, onde são reduzidos os teores de enxofre a níveis aceitáveis. Também são feitas análises da composição química da liga (carbono, silício, manganês, fósforo, enxofre) e a seguir o carro torpedo transporta o ferro gusa para a aciaria, onde será transformado em aço.

Aciaria

Na aciaria, o ferro gusa é transformado em aço através da injeção de oxigênio puro sob pressão no banho de gusa líquido, dentro de um convertedor. A reação, constitui na redução da gusa através da combinação dos elementos de liga existentes (silício, manganês) com o oxigênio soprado, o que provoca uma grande elevação na temperatura, atingindo aproximadamente 1700°C. Os gases resultantes do processo são queimados logo na saída do equipamento e os demais resíduos indesejáveis são eliminados pela escória, que fica a superfície do metal. Após outros ajustes finos na composição do aço, este é transferido para a próxima etapa que constitui o lingotamento contínuo.

Laminação

Posteriormente, os lingotes devem passar pelo processo de laminação, podendo ser a quente ou a frio, onde se transformarão em chapas através da diminuição da área da seção transversal.

4.2 O Ambiente Aciaria

A Aciaria é local dentro de uma usina siderúrgica onde existem máquinas e equipamentos voltados para o processo de transformação do ferro-gusa em diferentes tipos de aço. (SIDERURGIA, 2006)

A transformação do ferro-gusa em aço envolve:

- A diminuição dos teores de carbono, silício, fósforo, enxofre e nitrogênio a níveis bastante baixos.
- A adição de sucata ou minério de ferro para ajustar a temperatura do aço bruto.
- O ajustes dos teores de carbono, manganês, elementos de liga e da temperatura no forno ou na panela de vazamento.

O setor de aciaria compreende os convertedores (refino primário) e o refino secundário, em seguida o aço entra na fase de lingotamento. Este trabalho se limita ao sequenciamento e *Scheduling* das corridas de aço apenas no setor de aciaria, ou seja, entre os processos de dessulfuração no carro torpedo e o lingotamento contínuo.

O aço (neste caso, será tarefa) chega na aciaria através dos carros torpedos vindo do setor do alto-forno e o destino é o setor de conformação mecânica, mais precisamente o lingotamento contínuo. O algoritmo irá encontrar a melhor arranjo das operações em cada “máquina” com o objetivo de minimizar o tempo de fabricação de todos os aços a serem produzidos.

O ambiente de aciaria usado nesse trabalho é apenas ilustrativo, a finalidade é fornecer o cenário para o sequenciamento da produção, ou seja, os parâmetros necessários para a resolução Problema de *Job-Shop Scheduling*.

A aciaria proposta é composta pelos seguintes recursos: Converteedores e Refino Secundário

Converteedores

O convertedor é um tipo de forno revestido com tijolos refratários que transforma o ferro usa e sucata em aço. Uma lança sopra oxigênio em alta pressão para o interior do forno produzindo reações químicas que separam as impurezas, como os gases e a escória. No convertedor ocorre o processo chamado refino primário.

Convertedores	Capacidade de produção (toneladas/corrída)	Tempo médio por corrída (minutos)
CV-1	300	42
CV-2	400	40
CV-3	500	38

Tabela 1: Capacidade dos convertedores

Refino Secundário

Para alcançar determinadas propriedades, o aço passa por uma etapa chamada refino secundário, onde são feitas correções em sua composição química e sua temperatura, geralmente essas correções são feitas em um forno panela.

Fornos Panelas	Capacidade de produção (toneladas/corrída)	Tempo médio por corrída (minutos)
RH	500	12 a 50
IRUT	500	12 a 50

Tabela 2: Capacidade dos fornos panela

5 *Modelagem do problema*

Neste capítulo serão definidos os componentes que envolvem o problema de *Job-Shop Scheduling*, ou seja, a descrição das máquinas e dos *jobs*, definição de suas rotas e seus tempo de processamento em cada máquina. Neste capítulo é descrito como a modelagem das classes que envolvem o problema foi realizada.

5.1 Componentes do problema de *Job-Shop Scheduling*

Os componentes que definem o problema do *Job-Shop Scheduling* são:

- Jobs : Os *jobs* nesse problema em particular são os tipos de aço que serão produzidos.
- Recursos : Descrição das características e quantidades de maquinas no sistema.
- Sequência das máquinas: A rota de produção é definida pela sequência das máquinas no sistema, isto é, a ordem das máquinas nas quais os aços irão ser processados. Uma característica do problema de *Job-shop Scheduling* é que cada tipo de aço possua sua rota de produção definida.
- Tempo de processamento : Cada aço possui um determinado tempo de fabricação predeterminado, esse tempo se dá em função do tipo de máquina em que é processado, características físicas e químicas do aço, volume de produção, entre outras restrições.

Os dados para estes componentes são definidos com base no estudo de caso de uma indústria de fabricação de aço, a tabela 3 contém informações relevantes dos tipos de aço produzidos neste ambiente fictício.

Na tabela 3 abaixo estão definidos os tipos de aço produzidos neste estudo de caso, cada tipo de aço faz analogia a uma tarefa.

Tipos de aço (Tarefas)	Elementos (%)		
	c	Mn	Si
A-A	0,89	2,74	0,45
A-B	0,87	2,7	0,4
A-C	0,85	2,72	0,43
A-D	0,95	2,76	0,47
A-E	0,92	2,75	0,49
A-F	0,88	2,8	0,5
A-G	0,83	2,78	0,39
A-H	0,86	2,85	0,42
A-I	0,9	2,68	0,52
A-J	0,84	2,65	0,55

Tabela 3: Características de produção dos tipos de aço

A tabela 4 mostra os dados para estas componentes são definidos com base no estudo de caso da indústria de fabricação de aço que serão utilizados para resolução do problema de *Job-Shop Scheduling* em um ambiente de aciaria. Para resolução desse problema será considerado somente as rotas 2 e 3 da tabela acima, ou seja, será resolvido o *Scheduling* das operações dos aços entre os convertedores e os fornos RH e IRUT.

Tipos de aço	Rotas de produção				Tempo total	
	1º	2º	3º	4º	de processamento	
					2º	3º
A-A	CT	CV-1	RH	MLC	279	200
A-B	CT	CV-2	IRUT	MLC	275	256
A-C	CT	CV-3	RH	MLC	348	120
A-D	CT	CV-1	IRUT	MLC	384	358
A-E	CT	CV-2	RH	MLC	430	300
A-F	CT	CV-3	IRUT	MLC	458	473
A-G	CT	CV-1	RH	MLC	372	116
A-H	CT	CV-2	IRUT	MLC	502	516
A-I	CT	CV-3	RH	MLC	463	360
A-J	CT	CV-1	IRUT	MLC	361	297

Tabela 4: Parâmetros de roteirização na produção do aço

Legenda:

CT - Carro Torpedo

IRUT – Forno IRUT

CV – Convertedores

MLC – Máquina de Lingotamento Contínuo

RH – Forno RH

5.2 O Problema Abordado

No problema abordado os convertedores e os fornos fazem analogia às máquinas, enquanto os dez tipos de aço (veja a tabela 4) fazem analogia às tarefas, os convertedores CV-1, CV-2 e CV-3 são análogos as máquinas 1,2 e 3 respectivamente, enquanto os fornos RH e IRUT são análogos às máquinas 4 e 5 respectivamente. Além disso, os sistema modelado possui dois níveis, no primeiro nível se encontram os convertedores (CV-1, CV-2 e CV-3) e no segundo os fornos (RH e IRUT). Cada tarefa (cada aço) deve ser processada em alguns dos convertedores do primeiro nível para depois ser processada em algum forno do segundo nível, ou seja, um aço possui uma operação processada no primeiro nível e outra operação processada no segundo nível, veja a figura 8.

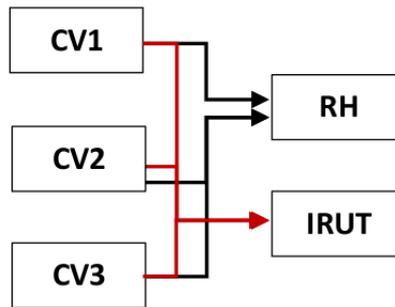


Figura 8: Estrutura do problema abordado

5.3 Modelagem do Algoritmo - Classes Java

Nessa seção serão apresentadas as classes que foram implementadas na linguagem Java para processar o problema.

5.3.1 Classe Operação

A operação é parte da tarefa que será de fato processada nas máquinas, a classe foi modelada como mostra o código abaixo. Na classe Operacao são definidos os atributos numeroDaTarefa e numeroDaMaquina que são respectivamente, um identificador da tarefa a qual a operação pertence e em qual máquina irá processar a operação. O cálculo dos tempos de início e fim são calculados pela classe Máquina à medida as operações forem alocadas para serem processadas.

```

1 public class Operacao {
2
3     private int numeroDaTarefa;
4     private int numeroDaMaquina;
5     private int tempoDeProcessamento;
6     private int inicioDoProcessamento;
7     private int terminoDoProcessamento;
8
9     public Operacao(int numeroDaMaquina, int tempoDeProcessamento, int numeroDaTarefa) {}
10    public int getNumeroDaMaquina() {}
11    public void setNumeroDaMaquina(int numeroDaMaquina) {}
12    public int getTempoDeProcessamento() {}
13
14
15 }
```

5.3.2 Classe Máquina

A classe Máquina foi modelada como mostra o código abaixo. Cada máquina é responsável por processar um certo número de operações, sendo assim ela possui um como atributo um ArrayList, que no Java é uma lista (um conjunto) de operações que será processadas pela máquina, uma função importante que é definida na classe Maquina é a função que calcula os tempos de inicio e fim de cada operação.

```

1  public class Maquina {
2
3      private int numeroDaMaquina;
4      private ArrayList<Operacao> operacoesExecutadas;
5
6      public Maquina(int numeroDaMaquina){}
7      public int getNumeroDaMaquina() {}
8      public void setNumeroDaMaquina(int numeroDaMaquina){}
9      public void addOperacoes(Operacao op) {}
10     public ArrayList<Operacao> getOperacoesExecutadas()
11     public void setOperacoesExecutadas(ArrayList<Operacao> operacoesExecutadas){}
12     public void calcularTemposDosJobsParaOPromeiroSetDeMaquinas(){}
13     public void calcularTempoDasOpsDoSegundoSetDeMaqs(ArrayList<Tarefa> tarefas) {}
14 }

```

5.3.3 Classe Tarefa

A classe possui um conjunto de operações, para isso também foi usado a estrutura ArrayList, na classe Tarefa existe um método importante que faz o cálculo do tempo total de processamento de uma tarefa, ou seja, a soma do tempo de processamento de todas as operações de uma determinada tarefa.

```

1  public class Tarefa {
2
3      private int numeroDaTarefa;
4      private int tempoTotalDeProcessamento;
5      private ArrayList<Operacao> operacoes;
6
7      private int calcularTempoTotalDeProcessamento(){}
8      public void verificarOperacoes() {}
9
10 }

```

5.3.4 Classe Utils

Foi criada uma classe denominada Utils para definir alguns métodos importantes para a solução do problema. Nesta classe está definido o método que calcula o *makespan*. Nessa classe também é definido o método de construção da solução inicial, essa solução inicial é gerada escolhendo aleatoriamente uma tarefa e alocando as suas operações nas respectivas máquinas que irão processá-las, as funções que auxiliam nessa escolha aleatória também estão definidas nessa classe.

```

1 public class Utils {
2
3     public static int escolherTarefaAleatoriamente(){}
4     public static ArrayList<Integer> gerarListaDasTarefasAleatoriamente(){ }
5     public static void exibirSolucao(ArrayList<Maquina> maquinas) {}
6     public static int calcularMakespan(ArrayList<Maquina> maquinas){}
7     public static void construcao(ArrayList<Maquina> maquinas, ArrayList<Tarefa> tarefas) {}
8
9 }

```

5.3.5 Classe Simulated Annealing

Nessa classe esta codificada a heurística que ira solucionar o problema, a solução inicial que é gerada por um método da classe Utils é passado como solução inicial para a heurística iniciar o procedimento de busca de uma solução ótima.

```

1 public class SimulatedAnnealing {
2
3     public static void procedimentoSimulatedAnnealing(ArrayList<Maquina> solucao,
4                                                         ArrayList<Tarefa> tarefas,
5                                                         double tempInicial,
6                                                         int numIteracao,
7                                                         double numAlpha) {}
8     public static ArrayList<Maquina> perturbarSolucao(ArrayList<Maquina> solucaoProvisoria) {}
9 }

```

5.3.6 Classe Processamento

A classe processamento é a classe principal do programa, nessa classe são criadas as instâncias das máquinas em forma de ArrayList pois o problema possui cinco máquinas, e as tarefas são instanciadas também na forma de ArrayList pois no problema tratado possui

dez tarefas, após a criação das instâncias a solução inicial é gerada e depois essa solução juntamente com os parâmetros temp, alpha e iterMax (Temperatura inicial, alfa, número máximo de iteração) são passados para o método procedimentoSimulatedAnnealing da classe SimulatedAnnealing.

```

1  public class Processamento {
2
3      public static void main(String[] args) throws IOException {
4
5          long tempoInicial = System.currentTimeMillis();
6          ArrayList<Tarefa> tarefas = ArquivoUtils.lerArquivoTxt();
7          ArrayList<Maquina> maquinas = new ArrayList<Maquina>();
8          ArrayList<Integer> ordemAleatoriaDasTarefas = new ArrayList<Integer>();
9          ordemAleatoriaDasTarefas = Utils.gerarListaDasTarefasAleatoriamente();
10
11         for (int i = 0; i < Constants.NUMERO_DE_MAQUINAS; i++) {
12             maquinas.add(new Maquina(i + 1));
13         }
14
15         Utils.construcao(maquinas, tarefas);
16
17         double temp = 8433.05;
18         double alpha = 0.93;
19         int iterMax = 30;
20
21         SimulatedAnnealing.procedimentoSimulatedAnnealing(maquinas,
22                                                         tarefas,
23                                                         temp,
24                                                         iterMax,
25                                                         alpha);
26     }

```

5.3.7 Classes Auxiliares

Foi definida uma classe auxiliar chamada ArquivoUtils que cuida de toda a manipulação do arquivo de entrada de dados, essa classe é responsável por fazer a leitura dos dados de entrada do problema e gravar em um arquivo a solução gerada. Uma classe chamada Constants foi criada para armazenar algumas constantes usadas no código fonte.

6 *Resultados e Discussões*

Neste capítulo serão discutidos os aspectos do desenvolvimento do algoritmo e a avaliação da solução encontrada a partir de um método heurístico, este se mostra mais eficaz em tempos computacionais quando comparado com um método exato, mas por outro lado, o método heurístico encontra uma solução de qualidade inferior ao uma solução obtida por um método exato.

6.1 A Heurística

6.1.1 Solução Inicial

Neste trabalho, a solução inicial foi construída selecionando aleatoriamente uma tarefa e alocando seus *jobs* nas máquinas na qual irão ser processados em ordem decrescente do tempo de processamento, como essa solução inicial foi construída aleatoriamente não há garantia que o seu *makespan* seja “bom”, essa solução inicial é passada como parâmetro para o procedimento *Simulated Annealing* para que ela seja perturbada e geradas novas soluções até se chegar na solução final, que é considerada a solução ótima para o problema em questão.

6.1.2 Perturbação da Solução

Nesta implementação o critério adotado para perturbar a solução foi o seguinte:

Ao escolher aleatoriamente uma máquina é escolhido também de forma aleatória uma operação que é executada pela máquina escolhida, a operação escolhida é trocada de posição (ordem de execução) com a operação vizinha, essa troca influencia diretamente no tempo de execução total.

Como se pode ver na figura 9 um exemplo de como a solução é perturbada, neste caso a perturbação acontece na máquina 2, em um primeiro momento a solução da tarefa

Maquina: 1	4/0-384	7/384-756	10/756-1117	1/1117-1396		
Maquina: 2	2/0-275	5/275-705	8/705-1207			
Maquina: 3	6/0-458	3/458-806	9/806-1269			
Maquina: 4	9/1269-1629	5/1629-1929	7/1929-2045	1/2045-2245	3/2245-2365	
Maquina: 5	2/275-531	4/531-889	10/1117-1414	6/1414-1887	8/1887-2403	
Maquina: 1	4/0-384	7/384-756	10/756-1117	1/1117-1396		
Maquina: 2	2/0-275	8/275-777	5/777-1207			
Maquina: 3	6/0-458	3/458-806	9/806-1269			
Maquina: 4	9/1269-1629	5/1629-1929	7/1929-2045	1/2045-2245	3/2245-2365	
Maquina: 5	2/275-531	4/531-889	10/1117-1414	6/1414-1887	8/1887-2403	
Maquina: 1	4/0-384	7/384-756	10/756-1117	1/1117-1396		
Maquina: 2	8/0-502	2/502-777	5/777-1207			
Maquina: 3	6/0-458	3/458-806	9/806-1269			
Maquina: 4	9/1269-1629	5/1629-1929	7/1929-2045	1/2045-2245	3/2245-2365	
Maquina: 5	2/777-1033	4/1033-1391	10/1391-1688	6/1688-2161	8/2161-2677	

Figura 9: Etapas da perturbação da solução

5 que iniciava em 275 e terminava em 705 troca de lugar com a tarefa 8 e passa a começar em 705 e terminar 1207, com essa mudança o valor do *makespan* não sofre alteração e permanece com valor 2403. E uma segunda perturbação a tarefa 8 que iniciava em 275 e terminava em 777 troca de lugar com a tarefa 2 e passa a começar em 0 e terminar em 502, com isso o valor do *makespan* que antes era 2403 sobe para 2677, e esse procedimento é repetido n vezes (valor que é definido no parâmetro número de iteração)

6.2 Entrada de dados

Os dados de entrada para o problema vem de um arquivo no formato txt como o definido na figura 5, neste arquivo estão definidos em forma matricial os tempos de processamento de cada *job*.

- Na primeira linha da tabela estão definidos o número de linhas e o número de colunas da tabela, que representam, respectivamente, o número de tarefas (linhas) e o número de máquinas (colunas);
- Cada linha representa uma máquina, e cada elemento T_{ij} representa o tempo de processamento de uma operação da tarefa i na máquina j , se este elemento possuir o valor 0 (zero) significa que a tarefa não é executada na máquina em questão.

	10	5			
	M1	M2	M3	M4	M5
T1	279	0	0	200	0
T2	0	275	0	0	256
T3	0	0	348	120	0
T4	384	0	0	0	358
T5	0	430	0	300	0
T6	0	0	458	0	473
T7	372	0	0	116	0
T8	0	502	0	0	516
T9	0	0	463	360	0
T10	361	0	0	0	297

Tabela 5: Arquivo de Dados de Entrada

6.3 Resultado Final

O resultado final do algoritmo é a sequência de produção ótima que minimiza o tempo de fabricação, o *makespan*. Os resultados estão dispostos na tabela 6. Esta tabela nos mostra os tempos, inicial e final das operações de cada tarefa (tipo de aço) nas respectivas máquinas.

Para se obter o *schedule* das máquinas, solução para o problema, foi necessário realizar testes variando-se os parâmetros da heurística *Simulated Annealing* para ajustá-los de forma a obter uma solução de boa qualidade. Foi implementado uma função para variar esses parâmetros, para cada variação foram geradas dez soluções e calculada a média aritmética dessas soluções geradas, a solução com menor média foi escolhida e o seus parâmetros foram utilizados para gerar a solução em questão. No Anexo I encontra-se a tabela contendo as cem variações de parâmetros e suas respectivas soluções.

Os parâmetros escolhidos foram os seguintes:

Temperatura inicial = 10000,00; Temperatura final = 1000,00;

Alpha = 0.8; Iterações por ciclo = 200

Obs.: A temperatura final foi definida como sendo 10% da temperatura inicial, isso vale para cada variação da temperatura inicial.

A figura 10 mostra a solução representada sob a forma de gráfico de Gantt. Esta solução foi gerada pelo software LISA.(LISA, 2010)

TAREFAS	Solução ótima para o problema de <i>Job-Shop Scheduling</i>									
	CV-1		CV-2		CV-3		RH		IRUT	
	<i>Início</i>	<i>Fim</i>	<i>Início</i>	<i>Fim</i>	<i>Início</i>	<i>Fim</i>	<i>Início</i>	<i>Fim</i>	<i>Início</i>	<i>Fim</i>
(1) A-A	756	1035					1035	1235		
(2) A-B			0	275					931	1187
(3) A-C					458	806	1235	1355		
(4) A-D	372	756							1187	1545
(5) A-E			777	1207			1715	2015		
(6) A-F					0	458			458	931
(7) A-G	0	372					372	488		
(8) A-H			275	777					1842	2358
(9) A-I					806	1269	1355	1715		
(10) A-J	1035	1396							1545	1842

Tabela 6: Tabela de resultados

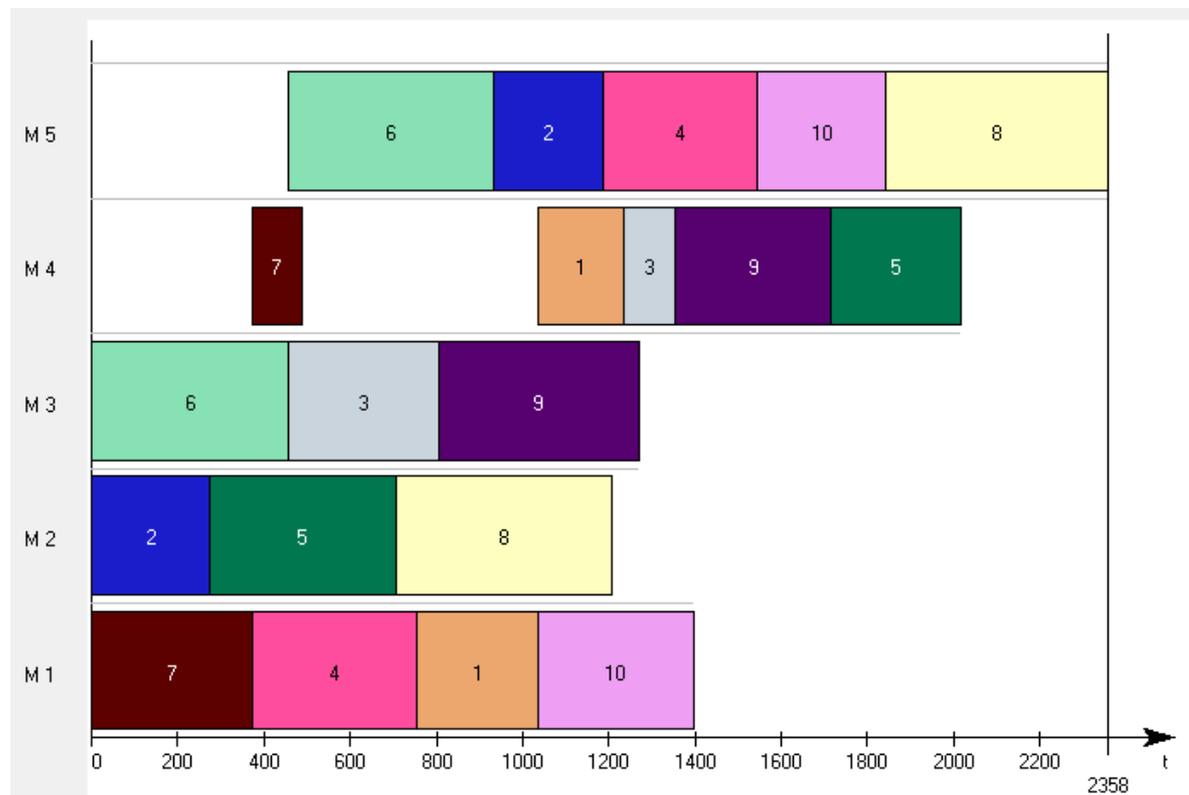


Figura 10: Grafico de Gant da Solução (LISA, 2010)

6.4 Qualidade da Solução

Embora a solução encontrada por um método heurístico seja retornada rápida, a solução em questão foi gerada em aproximadamente 217 milissegundos, não há garantia que essa solução seja ótima, embora seja uma solução boa.

Com o objetivo de avaliar o grau de aceitação da solução encontrada, foi feito um comparativo entre a heurística *Simulated Annealing* usada nesse trabalho com uma solução gerada pelo método exato *Branch & Bound* implementado no trabalho de Portilho (2007). No referido trabalho foi resolvido a mesma instância do problema tratado neste trabalho. Na tabela 7 está a avaliação feita.

Apesar de estar nesta tabela o tempo de execução dos métodos, estes parâmetros não são completamente comparáveis pois não é conhecido o ambiente que o método *Branch & Bound* do trabalho de Portilho (2007) foi executado.

Métodos	Medidas de Qualidade		
	Makespan	Tempo de processamento	Aumento do makespan
Simulated Annealing	2358	0,21 segundos	8,41%
Branch & Bound	2175	14,59 segundos	

Tabela 7: Medida de Qualidade da Solução

Observa-se que o método heurístico obteve um tempo de processamento 97% melhor quando comparado ao método exato, mas o *makespan* gerado resultou em um piora de 8,41%. Na figura 11 abaixo, está o graficamente o comparativo mostrando a diferença do *makespan* gerado pelos métodos Exato e Heurístico.

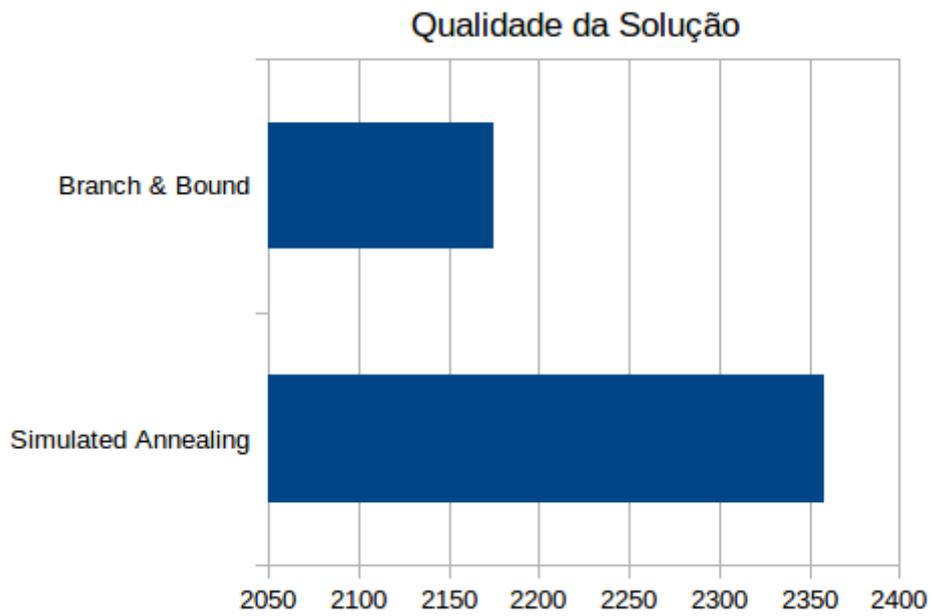


Figura 11: Avaliação do Makespan

Na figura 12 abaixo, está graficamente a diferença entre os tempos de execução dos algoritmos Exato e Heurístico.

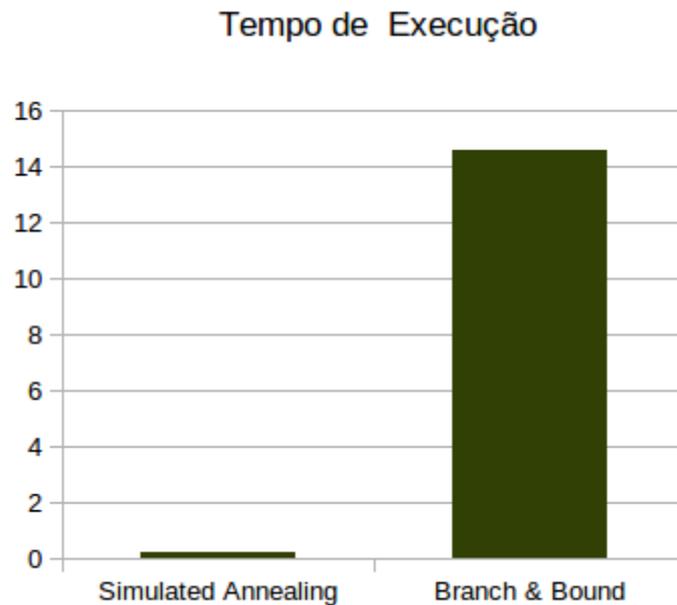


Figura 12: Avaliação do tempo de execução do algoritmo

7 Conclusão

O método heurístico *Simulated Annealing* é um método robusto para buscar boas soluções para problemas de *scheduling* apesar de métodos heurísticos serem aplicados para problemas bem maiores, nesse trabalho a heurística *Simulated Annealing* foi aplicado à um problema relativamente pequeno, mas é um problema com um grau de complexidade NP-Difícil. Apesar desse ser uma instância pequena, o método heurístico apresentou um bom desempenho computacional, por outro lado, um método exato tem um desempenho computacional pior, se aplicado à um problema de grande porte essa diferença de tempos computacionais entre as heurísticas se torna ainda maior.

Embora o problema tratado nesse problema é relativamente pequeno, o método obteve um bom desempenho, a solução teve uma piora de aproximadamente 8,5% quando comparada com uma solução gerada por um método exato, veja a figura 7, em relação ao tempo de execução do algoritmo, a performance do método Heurístico foi mais eficiente do que o método Exato para a mesma instância do problema, com a heurística a redução do tempo de execução foi de aproximadamente 97%.

7.1 Trabalhos Futuros

Como trabalho futuro, poderia ser feito um estudo para aprimorar a escolha dos parâmetros da heurística *Simulated Annealing*, O método usado para gerar a solução inicial poderia ser modificado na tentativa de uma otimização, a etapa de perturbação da solução poderia ser modificada, poderia propor uma nova formula para se realizar essa perturbação e com isso melhorar a convergência da solução.

Referências Bibliográficas

- BAKER, K. R.; TRIESCH, D. *Introduction to Sequencing and Scheduling*. New York: Wiley, 1974.
- BLAZEWICZ, J. et al. *Scheduling in Computer and Manufacturing Systems*. Verlag: Springer, 1994.
- CUNHA, C. R. *Metaheurística Híbrida GRASP e Busca Tabu aplicada ao problema de escalonamento de tarefas*. Dissertação (Mestrado) — Universidade Federal da Paraíba, 2010.
- FIGUEIRA, M. de A. *Identificação de células de produção em uma planta industrial utilizando uma heurística Simulated Annealing*. Dissertação (Mestrado) — Universidade Estadual do Norte Fluminense - UENF, 2002.
- KIRKPATRIC, S. et al. Optimisation by simulated annealing. *Science*, v. 220, p. 671–680, 1983.
- LAND, A. H.; DOIG, A. G. *An automatic method of solving discrete programming problems*. 1960.
- LISA, S. *Library of Scheduling Algorithms, LISA*. 2010. Disponível em: <<http://lisa.math.uni-magdeburg.de>>. Acesso em: 03/07/2013.
- METROPOLIS, N. et al. Equation of state calculations by fast computing machines. *J. Chem. Phys.*, v. 21, p. 1087–1092, 1953.
- PAPE, C. L. *Intelligent Scheduling: Scheduling as intelligent control of decision-making and constraint propagation*. Lisboa: Morgan Kaufman, 1994.
- PINEDO, M. *Theory, Algorithms and Systems*. 2^a. ed. New Jersey: Prentice-Hall, 2002.
- PORTILHO, L. M. P. *Aplicação do Problema de Job Shop Scheduling em aciarias*. Monografia (Graduação), Juiz de Fora - MG, 2007.
- REIS, J. *Uma Introdução ao Scheduling*. Lisboa, 1996. Departamento de Ciências e Tecnologias de Informação do ISCTE-DCTI.
- ROLDÃO, V. S. *Planejamento e Programação da Produção*. Lisboa: Monitor, 1995.
- SIDERURGIA, I. B. de. *Processo Siderúrgico - Fluxograma de Produção*. 2006. Disponível em: <www.ibs.org.br>. Acesso em: 19/06/2012.
- SOUZA, M. J. F. *Tópicos Especiais em Inteligência Artificial*. Ouro Preto, 2001. DECOM/ICEB/UFOP - Apostila.

UFPR. *O aço e sua obtenção*. 2012. Disponível em:
<<http://www.cesec.ufpr.br/metalia/01/01-texto.htm>>. Acesso em: 20/07/2012.

ANEXO I

Tabela 8: Avaliação dos parâmetros da heurística SA

Temperatura Inicial	Numero de Iterações	Alpha	MAKESPAN										Média
			V 1	V 2	V 3	V 4	V 5	V 6	V 7	V 8	V 9	V 10	
10000	200	0,9	2524	2409	2645	3296	3017	2912	2912	2872	2633	2924	2814,4
10000	200	0,8	2924	2645	2409	3296	2645	2261	2540	2540	2872	2645	2677,7
10000	200	0,7	2633	2540	2872	2633	2409	3296	2645	2912	2924	3296	2816
10000	200	0,6	3296	2645	3296	2524	3296	3296	2645	2540	2633	2924	2909,5
10000	150	0,9	2645	2912	3017	2924	2872	2409	2337	2261	2872	3296	2754,5
10000	150	0,8	3296	2409	2645	2872	2872	3296	2540	2872	2633	2540	2797,5
10000	150	0,7	3296	2337	2524	2409	3296	2924	2872	2872	2872	3296	2869,8
10000	150	0,6	2409	2912	2540	2337	2524	2872	2872	2912	2924	2524	2682,6
10000	100	0,9	2524	3017	2365	2924	2633	2645	3296	3296	2633	2912	2824,5
10000	100	0,8	2409	2912	2645	3296	2912	2633	2924	2524	3296	2645	2819,6
10000	100	0,7	2633	2912	2912	2540	2540	2261	2872	2924	2924	2337	2685,5
10000	100	0,6	2924	3296	2633	2924	3296	2872	2924	2924	2409	3296	2949,8
10000	50	0,9	2633	2924	3296	3296	3296	2633	3296	2540	2872	2261	2904,7

10000	50	0,8	2365	2365	3296	2924	3296	2872	3296	2645	2924	2540	2852,3
10000	50	0,7	2409	2645	3296	3017	2645	2409	3296	2540	2872	2645	2777,4
10000	50	0,6	2645	2912	3296	2633	3296	2540	2872	2872	2540	3296	2890,2
8000	200	0,9	2872	2924	3296	2645	3017	2337	3296	3017	2524	3296	2922,4
8000	200	0,8	2337	3296	2872	2872	2924	2365	2645	2261	2912	2924	2740,8
8000	200	0,7	2924	2633	2540	2409	2337	2633	2633	3296	3296	3296	2799,7
8000	200	0,6	3296	3296	2524	2924	3017	2633	3296	3296	2872	2270	2942,4
8000	150	0,9	2872	2872	2524	3296	3017	3296	3017	2645	2912	2337	2878,8
8000	150	0,8	2633	3296	2645	2912	2365	2524	3017	3017	2365	2409	2718,3
8000	150	0,7	2365	3296	2924	2261	3017	3296	2912	3296	3296	3296	2995,9
8000	150	0,6	3296	3017	3017	2912	2924	2924	2365	2924	3017	2872	2926,8
8000	100	0,9	3296	2872	2524	2270	2524	2924	2524	2633	3296	2540	2740,3
8000	100	0,8	2872	2872	2524	2540	2524	2872	2924	3017	3296	3017	2845,8
8000	100	0,7	2270	2872	2524	3296	3296	2872	3296	2633	3017	2540	2861,6
8000	100	0,6	3296	3296	2872	2365	3296	2872	2365	3017	2365	3296	2904
8000	50	0,9	2524	3296	2645	2872	2872	2912	3296	2524	2645	3296	2888,2
8000	50	0,8	2924	2912	2633	2633	3296	3017	2261	2872	3296	3296	2914
8000	50	0,7	2409	2872	2645	3017	2645	2645	2872	2872	3017	3296	2829
8000	50	0,6	2409	2645	2924	2524	3017	2365	2540	2872	2872	3296	2746,4
6000	200	0,9	3296	2924	2872	3296	3296	3017	2912	3296	2872	3296	3107,7

6000	200	0,8	2912	3296	3017	2524	2912	2912	3296	2645	3017	2645	2917,6
6000	200	0,7	3017	3296	2365	2872	2365	3017	2337	2633	3296	2645	2784,3
6000	200	0,6	2645	2872	3296	3296	3296	2365	3296	2872	3017	3017	2997,2
6000	150	0,9	2270	3296	3017	2365	3017	2872	2633	2912	2872	2924	2817,8
6000	150	0,8	2645	3296	2524	2633	2633	3296	2365	3296	3017	2633	2833,8
6000	150	0,7	2924	2524	2365	2645	2924	2633	2633	2872	2409	3296	2722,5
6000	150	0,6	3017	2540	2912	2337	2540	2365	3296	3017	2872	3017	2791,3
6000	100	0,9	2540	2912	2524	2872	2540	2912	3017	2912	2924	2872	2802,5
6000	100	0,8	3296	3296	2912	2872	2872	3296	2872	2924	2924	2872	3013,6
6000	100	0,7	2524	2912	2409	3017	3296	3296	2524	3296	2633	2924	2883,1
6000	100	0,6	3296	2912	3296	2924	2524	3296	2924	3017	2409	2409	2900,7
6000	50	0,9	2645	2633	3296	2409	2365	2924	2270	3296	3296	3296	2843
6000	50	0,8	3296	2872	2540	3296	2633	2872	2872	2872	2409	2365	2802,7
6000	50	0,7	2540	3296	2645	2912	2633	2365	3296	2524	3296	2645	2815,2
6000	50	0,6	2924	2409	3017	2540	2633	2633	3296	2645	3017	2645	2775,9
4000	200	0,9	2912	3017	2872	2872	2337	2912	3017	2633	2924	2912	2840,8
4000	200	0,8	3296	2645	3296	2633	2924	2409	2872	3017	2645	2409	2814,6
4000	200	0,7	2645	2872	3296	2872	3296	3296	2365	2872	2409	2540	2846,3
4000	200	0,6	2409	3017	2645	2924	2645	2912	2924	2912	2645	3296	2832,9
4000	150	0,9	3296	2872	2872	2524	2540	3296	2872	2524	2924	3296	2901,6
4000	150	0,8	2872	2365	2633	2912	3296	3296	2540	2365	3017	2872	2816,8

4000	150	0,7	2633	2633	2924	2365	2645	2872	3017	2540	3296	3296	2822,1
4000	150	0,6	2872	2261	3296	3296	2645	2365	3296	2524	2645	3296	2849,6
4000	100	0,9	2912	2633	2633	2924	3296	2645	2365	3296	2872	2337	2791,3
4000	100	0,8	2633	2912	3296	2633	2872	3296	3017	2365	3296	3296	2961,6
4000	100	0,7	3296	2872	3296	2633	3296	3296	2540	2872	2524	3296	2992,1
4000	100	0,6	2924	2633	3017	2645	2633	3017	2872	2924	3296	2633	2859,4
4000	50	0,9	2924	2633	2924	2633	3296	3296	3296	3296	2924	2540	2976,2
4000	50	0,8	3296	2261	3296	2365	2365	2912	2912	2924	3296	2337	2796,4
4000	50	0,7	3296	3296	2645	3296	2872	3296	2912	2924	2872	2924	3033,3
4000	50	0,6	2924	2540	2540	2633	3296	2540	2633	3296	2924	2645	2797,1
2000	200	0,9	2409	3296	3296	2409	2633	2924	2872	3017	2645	2524	2802,5
2000	200	0,8	2872	2912	2645	2924	3296	2409	2872	3017	3017	3296	2926
2000	200	0,7	2872	3296	2645	2872	2924	2540	2645	2261	3296	3296	2864,7
2000	200	0,6	3017	3296	2409	2924	2365	2645	2409	2633	2872	2912	2748,2
2000	150	0,9	3296	3296	2912	2872	2337	2365	3296	2912	2540	2912	2873,8
2000	150	0,8	2365	2924	2409	3296	2270	2924	3017	2365	2872	3296	2773,8
2000	150	0,7	2633	2872	3296	2924	2261	2872	3296	2261	2633	2409	2745,7
2000	150	0,6	3296	3296	3296	2872	2872	2524	2924	2872	2924	2524	2940
2000	100	0,9	2365	2270	3296	2872	3296	3296	2924	2409	2872	2924	2852,4
2000	100	0,8	2912	2633	2924	3296	2540	2365	3296	2633	3296	2912	2880,7
2000	100	0,7	2409	3296	2872	3296	2924	2409	2337	2645	2645	2365	2719,8

2000	100	0,6	3296	2872	2633	2365	2633	3296	2872	2524	3017	3017	2852,5
2000	50	0,9	2924	2912	3017	2633	3296	2872	2524	2365	2924	3296	2876,3
2000	50	0,8	3296	2872	2633	3296	2872	2912	2912	2645	3296	2924	2965,8
2000	50	0,7	3296	3017	3017	2337	2872	3017	2924	2872	2872	2409	2863,3
2000	50	0,6	2524	3296	2872	2645	2872	2645	2872	3296	2872	2924	2881,8