

Tiago Carvalho da Costa

*Uma solução para o problema de  
sequenciamento de tarefas em um ambiente  
industrial utilizando a heurística GRASP*

Campos dos Goytacazes/RJ

2012

Tiago Carvalho da Costa

*Uma solução para o problema de  
sequenciamento de tarefas em um ambiente  
industrial utilizando a heurística GRASP*

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<sup>a</sup>. Fermín Alfredo Tang Montané, DrSc.

Tutor: Fermín Alfredo Tang Montané, DrSc.

UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE DARCY RIBEIRO

Campos dos Goytacazes/RJ

2012

## *Lista de Figuras*

1	Representação pelo Grafico de Gantt . . . . .	11
2	Representação do problema usando grafo disjuntivo . . . . .	12
3	Representação da solução usando grafo disjuntivo orientado . . . . .	13
4	Pseudo-código da heurística GRASP . . . . .	16
5	pseudo-código da fase de construção da heurística GRASP . . . . .	17
6	pseudo-código da fase de busca local da heurística GRASP . . . . .	18

## *Lista de Tabelas*

1	Capacidade dos convertedores . . . . .	22
2	Capacidade dos fornos panela . . . . .	22
3	Características de produção dos tipos de aço . . . . .	24
4	Parâmetros de roterização na produção do aço . . . . .	24

# *Sumário*

<b>Lista de Figuras</b>	<b>1</b>
<b>Lista de Tabelas</b>	<b>2</b>
<b>1 Introdução</b>	<b>5</b>
1.1 Objetivo e Justificativa . . . . .	5
1.2 Estrutura do Trabalho . . . . .	6
<b>2 Fundamentos teóricos</b>	<b>7</b>
2.1 Introdução ao Sequenciamento de Tarefas . . . . .	7
2.2 Classificação das Máquinas . . . . .	8
2.3 Variantes do Problema de Scheduling . . . . .	8
2.3.1 Open-Shop Scheduling . . . . .	8
2.3.2 Flow-Shop Scheduling . . . . .	9
2.3.3 Job-Shop Scheduling . . . . .	9
2.4 Notação . . . . .	10
2.5 Representação do Problema . . . . .	11
2.5.1 Gráfico de Gantt . . . . .	11
2.5.2 Grafos Disjuntivos . . . . .	12
2.6 Métodos de Solução . . . . .	13
2.7 O problema de Sequenciamento de Tarefas e suas Aplicações . . . . .	14
<b>3 Heurística GRASP</b>	<b>15</b>

	4
3.1 Introdução . . . . .	15
3.2 Fase Construtiva . . . . .	16
3.3 Fase de Busca Local . . . . .	17
<b>4 O processo de fabricação do aço</b>	<b>19</b>
4.1 O Processo . . . . .	19
4.2 O Ambiente Aciaria . . . . .	21
<b>5 Modelagem do problema</b>	<b>23</b>
5.1 Parâmetros para o problema de Job-Shop Scheduling . . . . .	23
<b>Referências Bibliográficas</b>	<b>26</b>

# ***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, reconheceram que somente uma exploração eficiente dos fatores de produção irá 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 numa 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 necessários para executar um conjunto de tarefas (BAKER; TRIESCH, 1974), por isso têm sido desenvolvidas modernas técnicas e ferramentas computadorizadas para um uso otimizado dos recursos e do tempo.

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

## **1.1 Objetivo e Justificativa**

O objetivo geral deste trabalho é propor uma solução para um problema de planejamento de tarefas em um ambiente industrial utilizando Heurística GRASP

Em um ambiente industrial, um processo que depende de máquinas para executar uma série de operações pode fazer com que durante a produção muito tempo seja desperdiçado, e diminuir esse desperdício de tempo e melhorar o aproveitamento dos recursos, e consequentemente maximizar o lucro, é preciso planejar o processamento de cada operação. Esse planejamento não é simples de realizar, e com base nessa complexidade, uma heurística

é proposta para buscar uma solução para o problema de sequenciamento de tarefas em uma indústria.

## 1.2 Estrutura do Trabalho

O presente trabalho tem a seguinte estrutura. No capítulo 2, apresenta-se 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 é descrito os métodos de solução existentes e algumas aplicações para o problema.

No capítulo 3 é apresentada a heurística GRASP, é descrito em detalhes suas principais rotinas, a fase de construção de uma solução e a rotina de busca local.

No capítulo 4 é descrito de forma resumida o problema que será resolvido, neste caso o processo de fabricação do aço, bem como cada etapa desse procedimento.



## 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 – constrangimentos relaxáveis (relacionadas com datas limite, produtividade, frequência de troca de ferramentas e estabilidade da fábrica) (PAPE, 1994).

Nos problemas de Scheduling, de um modo geral, assume-se a necessidade de executar certo número de tarefas ou processos, cada um dos quais consiste numa dada sequência de operações, a executar pela ordem especificada na sequência, 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. É dada uma função que mede o custo de cada solução possível e pretende-se obter uma solução em que o custo seja mínimo.

O parâmetros 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 adicionais,  $R = \{ R_1, R_2, \dots, R_s \}$

## 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  $P$  desempenham as mesmas funções. Elas podem ser distinguidas pelas suas velocidades de processamento.
- **Máquinas Dedicadas:** Cada máquina do conjunto  $P$  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 um dos Jobs 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 total de execução que é chamado *makespan* e o tempo total de fluxo denominado *flowtime*.

### 2.3.2 Flow-Shop Scheduling

Neste tipo de problema existe uma ordem pré-estabelecida para execução das operações de cada um dos Job, ou seja, existe uma sequência de execução que deve ser respeitada. Neste caso, o processamento das tarefas ocorre em linha, ou seja, as tarefas passam por todas as máquinas que compõem o sistema, exatamente na mesma ordem, assim a  $i$ -ésima operação de cada uma das tarefas deverá ser sempre feita na máquina  $i$ .

Percebe-se 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

O Job-Shop Scheduling caracteriza-se por diferenciar as operações de cada um dos Jobs, direcionando sua execução para máquinas específicas, o que restringe a execução de uma dada operação a uma só 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, ou seja, a melhor solução para o problema.

O Job Shop Scheduling é composto por um conjunto de Jobs ( $J = \{j_1, j_2, \dots, j_p\}$ ), onde cada job 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 com  $m$  máquinas, onde cada job tem sua rota predeterminada a seguir.

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

**Data de início ( $r_j$ )** : A data de início do Job  $j$ , representado pela letra  $r_j$ , é o tempo (data) que o Job chega no sistema, indica o tempo que o job  $j$  pode começar a ser processado

**Data de vencimento ( $d_j$ )** : A data de vencimento  $d_j$  do Job  $j$  representa a data de finalização do processamento do Job  $j$ , é o quem é chamado de tempo de pronto.

**Makespan ( $C_{max}$ )** : O Makespan pode ser chamado de tempo de fabricação, é tempo final de processamento de todos os jobs, e é definido como  $\max (C_1, \dots, C_n)$ .

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

*Minimize*  $C_{max}$

*sujeito a*

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

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

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

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

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 do Problema

Um *Job-Shop Scheduling* é a representação de uma programação dos Jobs, compostos por operações interdependentes, onde cada Job deve ser processado por uma máquina diferente. 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 seqüenciamento de execução das operações por parte das maquinas.

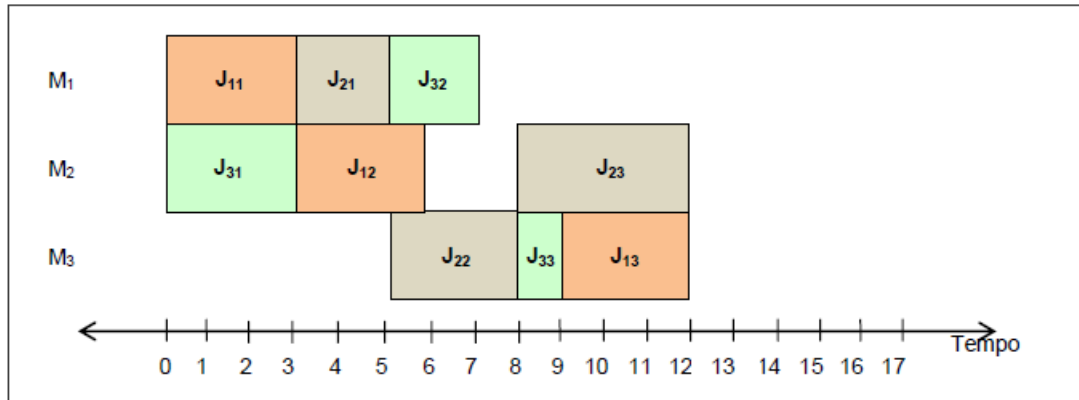


Figura 1: Representação pelo Grafico de Gantt

Nesse gráfico o eixo vertical representa as maquinas, o eixo horizontal representa a linha do tempo, e os retângulos representam as operações que compõem os Jobs, eles são encaixados no gráfico de modo que o seu tempo de execução fique marcado na linha do tempo. Os Jobs são identificados facilmente pela cor do retângulo, por exemplo, o Job 1 está representado pela cor laranja, onde  $J_{11}$  representa a operação 1 do Job 1, o  $J_{12}$  representa a operação 2 do Job 1 e o  $J_{13}$  representa a operação 3 do Job 1. Com base na representação da solução da figura 1 o *makespan*, ou seja, o tempo total de conclusão 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 de operação dos Jobs.

### 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

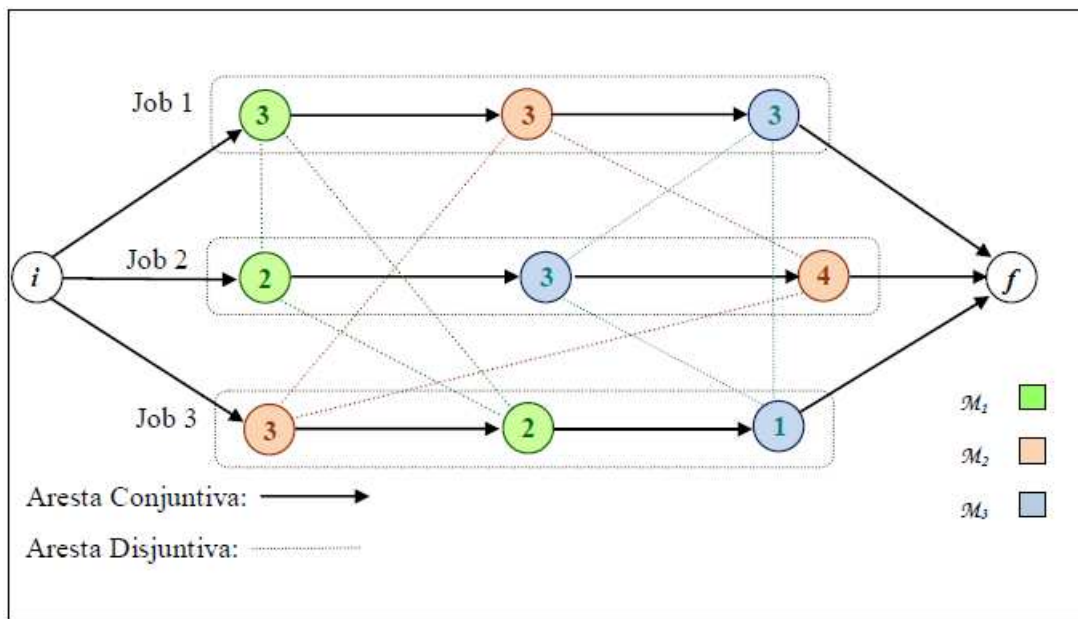


Figura 2: Representação do problema usando grafo disjuntivo

As operações de um mesmo Job 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 job indica em qual máquina ele é executado, a interconexão não orientada forma o grafo disjuntivo, e relaciona os jobs 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 ser alcançar diversas possibilidades de arranjo das operações, mas o grafo disjuntivo deve ser manter acíclico, pois uma operação só pode ser executada por uma máquina apenas uma vez.

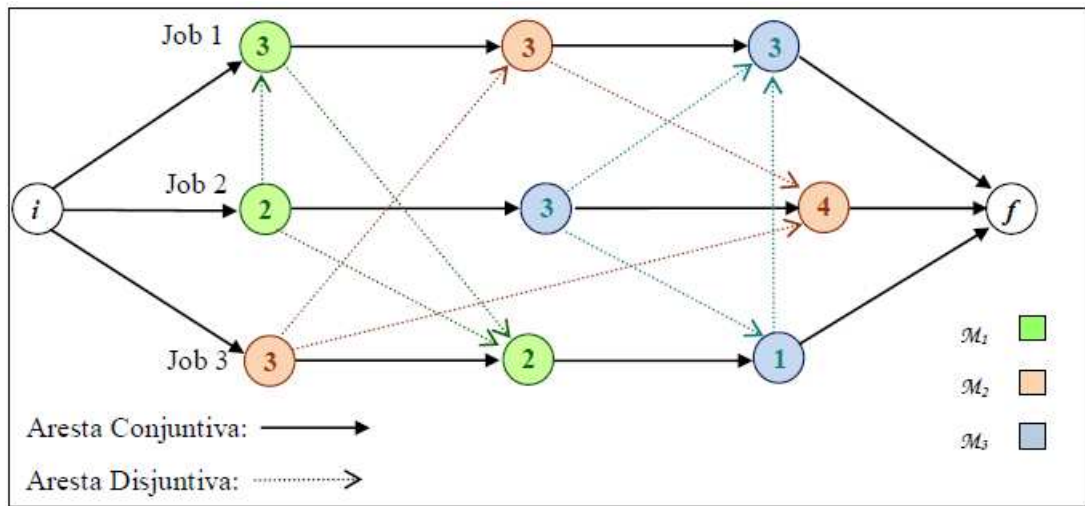


Figura 3: Representação da solução usando grafo disjuntivo orientado

A solução encontrada através do escalonamento das operações no grafo disjuntivo necessita ser avaliada e comparada com outras soluções, uma forma de avaliação é analisar o tempo do caminho crítico, isto é, analisar o tempo de execução das operações mais demoradas **assim a solução encontrada deve ser menor que a análise do caminho crítico.**

## 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 classes: Modelos matemáticos e Modelos heurísticos.

- **Modelos Matemáticos:** Trata-se de modelos de programação inteira mista, resolvidos pelos métodos branch and bound ou branch and cut. Modelos Matemáticos enfatizam a obtenção de resultados ótimos em função de algum parâmetro de desempenho. Este pode ser, por exemplo, a minimização dos tempos de produção ou a maximização do uso dos recursos. Dependendo da complexidade do problema tratado, o cálculo da solução ótima pode ser computacionalmente inviável.
- **Modelos Heurísticos:** Trata-se de modelos baseados em regras prática de escalonamento que enfatiza a obtenção de “boas” soluções, próxima 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

A programação de tarefas constitui-se num alvo de pesquisa a ser buscado por fábricas de diversas designações, as quais disponham de máquinas destinadas a papéis específicos com o fim de produzir, comumente, mais de um tipo de produto e, às vezes, utilizando uma mesma máquina em alguma etapa do processo; com o objetivo de minimizar o tempo ocioso e maximizar a produtividade.

Embora o nome seqüenciamento 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 sala de aula, navios em um porto, cidades e caixeiros viajantes, etc. (REIS, 1996)



## 3 *Heurística GRASP*

Neste capítulo é feita uma revisão bibliográfica da heurística GRASP, bem como sua fase de construção e de busca local.

### 3.1 Introdução

GRASP - *Greedy Randomized Adaptive Search Procedure*, ou procedimento de busca adaptativa gulosa e aleatória, é uma técnica iterativa proposta por Feo e Resende (1995) que consiste numa fase de construção de uma solução viável, na qual uma solução é gerada, elemento a elemento, e de uma fase de busca local, na qual se procura melhorar de forma iterativa a qualidade da solução encontrada anteriormente, a fase de busca local trabalha de forma iterativa através de sucessivas substituições da solução atual por uma melhor de sua vizinhança, quando não há mais soluções melhores essa fase é terminada.

Na figura 4, encontra-se um pseudocódigo genérico da heurística GRASP, os parâmetros que precisam ser definidos na heurística são o parâmetro de aleatoriedade  $\alpha$  e o critério de parada (GRASPmax) que geralmente é o número de iterações da heurística. De acordo com a figura 4, a heurística GRASP constrói repetidamente uma solução (passo 4) e esta é melhorada por uma busca local (passo 5) e a melhor solução encontrada até o momento é armazenada (passo 7 e 8).

A melhor solução encontrada ao longo de todas as iterações GRASP realizadas é retornada como resultado do algoritmo de otimização GRASP (SOUZA, 2001)

---

```

1  procedimento GRASP (f(.), g(.), N(.), GRASPmax, s)
2   $f^* \leftarrow \infty$ ;
3  para (iter = 1, 2, ..., GRASPmax) faça
4      Construc ao(g(.),  $\alpha$ , s);
5      BuscaLocal(f(.), N(.), s);
6      se (f(s) <  $f^*$ ) ent o
7           $s^* \leftarrow s$ ;
8           $f^* \leftarrow f(s)$ ;
9      fimse
10     fimpara
11      $s \leftarrow s^*$ ;
12     retorne s;
13 fim GRASP

```

---

Figura 4: Pseudo-c digo da heur stica GRASP

Como o GRASP normalmente trata de problemas da ordem de complexidade NP-Dif cil, poderia ficar indefinidamente em busca de uma solu  o  tima, para evitar esse caso,   preciso adotar um crit rio de parada na fase de busca local, por exemplo, o n mero m ximo de itera  es.

## 3.2 Fase Construtiva

Na fase de constru  o, uma solu  o   iterativamente constru da, elemento por elemento. A cada itera  o dessa fase, os pr ximos elementos candidatos a serem inclu dos na solu  o s o colocados em uma lista  $C$  de candidatos, seguindo um crit rio de ordena  o pr -determinado.

O processo de sele  o   baseado em uma fun  o adaptativa gulosa  $g : C \rightarrow \mathbb{R}$ , que estima o benef cio da sele  o de cada um dos elementos. A heur stica   adaptativa porque os benef cios associados com a escolha de cada elemento s o atualizados em cada itera  o da fase de constru  o para refletir as mudan as oriundas da sele  o do elemento anterior. A componente probabil stica  $\alpha \in [0, 1]$ , denominada taxa gulosa, reside no fato de que cada elemento   selecionado de forma aleat ria a partir de um subconjunto restrito formado pelos melhores elementos que comp em a lista de candidatos. Este subconjunto recebe o nome de lista de candidatos restrita LCR. Essa t cnica de escolha permite que diferentes solu  es sejam geradas em cada itera  o GRASP.

O parâmetro  $\alpha$  controla o nível de gulosidade e aleatoriedade na geração das soluções durante as iterações GRASP. Um valor  $\alpha = 0$  faz gerar soluções puramente gulosas, enquanto  $\alpha = 1$  faz produzir soluções totalmente aleatórias. As soluções geradas pela fase de construção do GRASP provavelmente não são localmente ótimas com respeito à definição de vizinhança que adotaremos a seguir na fase de busca local. Daí a importância da fase de busca local, a qual objetiva melhorar a solução construída.

O pseudocódigo da figura 5 descreve a fase de construção da Solução GRASP.

---

```

1  procedimento Construcao (g(.),  $\alpha$ , s)
2  s  $\leftarrow$   $\alpha$ 
3  Inicialize o conjunto C de candidatos
4  enquanto C  $\neq$  { } faca
5       $t_{min} = \min\{g(t) \mid t \in C\}$ 
6       $t_{max} = \max\{g(t) \mid t \in C\}$ 
7      LCR = {t  $\in$  C |  $g(t) \leq t_{min} + \alpha(t_{max} - t_{min})$ }
8      Selecione aleatoriamente um elemento t  $\in$  LCR
9      s  $\leftarrow$  s  $\cup$  {t}
10  Atualize o conjunto C de candidatos
11  fimenquanto
12  retorne s
13  fim Construcao
```

---

Figura 5: pseudo-código da fase de construção da heurística GRASP

### 3.3 Fase de Busca Local

A fase de busca local está baseada na noção de vizinhança. A função  $N$ , a qual depende da estrutura do problema tratado, associa a cada solução viável  $s$  sua vizinhança  $N(s)$ . Cada solução  $s' \in N(s)$  é chamado de vizinho de  $s$ . É denominado movimento a modificação  $m$  que transforma uma solução  $s$  em outra  $s'$ , que esteja em sua vizinhança. Representa-se essa operação por  $s' \leftarrow s \in m$ . Em linhas gerais, a fase de busca local, começando de uma solução obtida pela fase de construção GRASP, navega pelo espaço de pesquisa passando de uma solução para outra, que seja sua vizinha, em busca de um ótimo local. A figura 6 descreve o pseudo-código de um algoritmo básico de busca local com respeito a uma certa vizinhança  $N(.)$  de  $s$ .

---

```

1  procedimento BuscaLocal ( $f(\cdot)$ ,  $N(\cdot)$ ,  $s$ )
2   $s^* \leftarrow s$ 
3   $V = \{s' \in N(s) \mid f(s') < f(s)\}$ 
4  enquanto ( $V > 0$ ) faca
5      selecione  $s \in V$ 
6      se ( $f(s) < f(s^*)$ ) entao
7           $s^* \leftarrow s$ 
8      fimse
9       $V = \{s' \in N(s) \mid f(s') < f(s)\}$ 
10 finenquanto
11  $s \leftarrow s^*$ 
12 retorne  $s$ 
13 fim BuscaLocal

```

---

Figura 6: pseudo-código da fase de busca local da heurística GRASP

## *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**

(PARAÍBA, 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)

#### **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  $1300^{\circ}\text{C}$  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 siderurgica 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 (*jobs*) 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 seqüência do jobs em cada “máquina” em função do objetivo predefinido, ou seja, a seqüência que minimiza 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

## Convertedores

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.

<b>Convertedores</b>	<b>Capacidade de produção</b> (toneladas/corrída)	<b>Tempo médio por corrida</b> (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.

<b>Fornos Painelas</b>	<b>Capacidade de produção</b> (toneladas/corrída)	<b>Tempo médio por corrida</b> (minutos)
RH	500	12 a 50
IRUT	500	12 a 50

Tabela 2: Capacidade dos fornos panela



## 5 *Modelagem do problema*

Neste capítulo será definido os parâmetros 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. O parâmetro de otimização utilizado será a minimização do *makespan*.

### 5.1 Parâmetros para o problema de Job-Shop Scheduling

Os parâmetros necessários para a resolução do *Job-Shop Scheduling* são:

- Jobs : Os *jobs* nesse problema em particular é o aço a ser produzido. Nas siderurgia, de modo geral, é a capacidade de um convertedor que define a dimensão de um lote de produção em cada ciclo de tratamento de aços em aciaria.
- Recursos : Descrição das características e quantidades de maquinas no sistema.
- Sequencia das máquinas: A rota de produção é definida pela sequencia das máquinas no sistema, isto é, a ordem das máquinas nas quais os diversos tipos de aço irão ser processados. Uma caracterísitca do problema de *Job-shop Scheduling* é que cada tipo de aço possua sua rota de produção definida.
- Tempo de processamento : Cada *job* (aço) possui um determinado tempo de fabricação predeterminado, esse tempo se dá em função do tipo de máquina em que é processado, caracterísitcas físicas e químicas do aço, volume de produção, entre outras restrições.

Abaixo temos a tabela 3 que nos mostra as informações mais relevantes dos tipos de aço produzidos neste ambiente fictício.

Tipos de aço	Elementos (%)			Produção (t)	Nº de corridas
	c	Mn	Si		
A-A	0,89	2,74	0,45	2000	7
A-B	0,87	2,7	0,4	3100	9
A-C	0,85	2,72	0,43	5000	10
A-D	0,95	2,76	0,47	2500	8
A-E	0,92	2,75	0,49	3500	4
A-F	0,88	2,8	0,5	4000	6
A-G	0,83	2,78	0,39	1000	8
A-H	0,86	2,85	0,42	8000	9
A-I	0,9	2,68	0,52	5000	11
A-J	0,84	2,65	0,55	3000	7

Tabela 3: Características de produção dos tipos de aço

A tabela 4 abaixo nos mostra os parâmetros de roteirização que serão utilizados para resolução do problema de *Job-Shop Scheduling* em um ambiente de aciaria.

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

Para resolução desse problema será considerado somente as rotas 2 e 3 da tabela acima, ou seja, será resolvido o *Scheduling* do aço entre os convertedores e os fornos RH e IRUT.

## *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.
- FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. *Global Optimization*, v. 6, p. 109–133, 1995.
- PAPE, C. L. *Intelligent Scheduling: Scheduling as intelligent control of decision-making and constraint propagation*. Lisboa: Morgan Kaufman, 1994.
- PARAÍBA, U. F. da. *O aço e sua obtenção*. 2012. Disponível em: <<http://www.cesec.ufpr.br/metallica/01/01-texto.htm>>. Acesso em: 20/07/2012.
- PINEDO, M. *Theory, Algorithms and Systems*. 2<sup>a</sup>. ed. New Jersey: Prentice-Hall, 2002.
- 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](http://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.