

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIA DA COMPUTAÇÃO – BACHARELADO

SISTEMA DE ALOCAÇÃO DE MÁQUINAS DE TINGIMENTO
UTILIZANDO A TÉCNICA CSP

RODRIGO NORBERTO LERMEN

BLUMENAU
2009

2009/2-20

RODRIGO NORBERTO LERMEN

**SISTEMA DE ALOCAÇÃO DE MÁQUINAS DE TINGIMENTO
UTILIZANDO A TÉCNICA CSP**

Trabalho de Conclusão de Curso submetido à
Universidade Regional de Blumenau para a
obtenção dos créditos na disciplina Trabalho
de Conclusão de Curso II do curso de Ciência
da Computação — Bacharelado.

Prof. Paulo César Rodacki Gomes , Dr. - Orientador

**BLUMENAU
2009**

2009/2-20

SISTEMA DE ALOCAÇÃO DE MÁQUINAS DE TINGIMENTO UTILIZANDO A TÉCNICA CSP

Por

RODRIGO NORBERTO LERMEN

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Paulo César Rodacki Gomes, Dr. – Orientador, FURB

Membro: _____
Prof. Oscar Dalfovo, Dr. – FURB

Membro: _____
Prof. Roberto Heinzle, Mestre – FURB

Blumenau, 9 de dezembro de 2009

Dedico este trabalho a Susan, minha esposa e a Gessi, minha mãe, que muito me motivaram para a sua conclusão.

AGRADECIMENTOS

À minha família, pela paciência e compreensão pela ausência em determinadas horas para poder se dedicar.

Ao meu orientador, Paulo César Rodacki Gomes, que nos momentos em que eu pensei em desistir, o mesmo disse “calma que muitos que passaram por isso estão formados”.

A empresa Rovitex Indústria e Comércio de Malhas Ltda, que disponibilizou o espaço de tempo e ferramentas necessárias para concluir este trabalho.

Ao Fernando dos Santos, que ao expor as minhas dificuldades no desenvolvimento deste trabalho, prontamente me ajudou sem pedir retorno algum, apenas pelo simples motivo de repassar o seu conhecimento.

E a Deus, que me dá forças todos os dias.

Obstáculos são aquelas coisas amedrontadoras que vemos quando tiramos os olhos do objetivo.

Anônimo

RESUMO

O principal objetivo do departamento de Planejamento e Controle da Produção (PCP) dentro de uma indústria é a entrega da mercadoria dentro do prazo que foi adquirida pelo cliente. Este trabalho apresenta o desenvolvimento de um software para alocação de Ordens de Tingimento, que é uma das áreas com o maior número de restrições dentro do processo produtivo de uma empresa têxtil. Foi utilizada a técnica de *Constraint Satisfaction Problem* (CSP) que é explorada na área de Inteligência Artificial desde 1970 e atualmente é encontrada em diversos programas na área industrial. Através dos testes foi possível concluir que a utilização de Inteligência Artificial através dos algoritmos CSP é uma solução viável para os problemas de alocação de máquinas na tinturaria.

Palavras-chave: Problema de satisfação de restrições. Planejamento e controle da produção.

ABSTRACT

The main objective of the department of Planning and Production Control (PPC) within an industry is the timely delivery of merchandise that was purchased by the customer. This paper presents the development of software for allocating orders Dyeing, which is one of the areas with the greatest number of restrictions within the production process of a textile company. Technique was used for Constraint Satisfaction Problem (CSP) that is operated in the area of Artificial Intelligence since 1970 and is now found in several programs in the industrial área. Through the tests it was concluded that the use of artificial intelligence algorithms through the CSP is a viable solution to the allocation problems in dyeing machines.

Key-words: Constraint satisfaction problem. Production control and planning.

LISTA DE ILUSTRAÇÕES

Quadro 1 – Tabela exemplo do Menor Custo de Mudança (MCM).....	15
Figura 1 - Um a possível solução para o problema com seis rainhas	18
Figura 2 - Matriz representando a restrição entre x e y	20
Figura 3 - Resultado para as variáveis X e Y	20
Figura 4 - Mapa dos principais estados e territórios da Austrália	22
Figura 5 - Grafo de restrições do problema do mapa.....	22
Figura 6 - Algoritmo simples com retrocesso para CSP	23
Quadro 2 - Restrição das cores por máquina.....	30
Quadro 3 - Restrição dos materiais por máquina.....	31
Quadro 4 - Restrição dos pesos por máquina	31
Figura 7 - Diagrama de casos de uso do software	33
Figura 8 - Diagrama das classes de importação dos dados	34
Quadro 5 - Responsabilidades das classes de importação.....	35
Figura 9 - Classes para processamento da alocação	36
Quadro 6 – Exemplo de valores do domínio das cores.....	37
Quadro 7 - Exemplo de valores do domínio dos materiais	37
Quadro 8 – Exemplo de valores do domínio de pesos.....	37
Quadro 9 – Exemplo de valores do domínio das OTs	38
Quadro 10 - Exemplo de restrições de materiais nas máquinas	38
Quadro 11 - Exemplo de restrições de cores nas máquinas	39
Quadro 12- Exemplo de restrições de pesos nas máquinas.....	39
Quadro 13 - Exemplo das n -rainhas utilizando a biblioteca CHOCO.....	41
Quadro 14 - <i>Layout</i> do arquivo texto dos materiais.....	42
Quadro 15 – <i>Layout</i> do arquivo texto das cores	42
Quadro 16 - <i>Layout</i> do arquivo texto dos pesos	43
Quadro 17 - <i>Layout</i> do arquivo texto das OTs	43
Quadro 18 - <i>Layout</i> do arquivo texto das restrições dos materiais por máquina.....	43
Quadro 19 - <i>Layout</i> do arquivo texto das restrições das cores por máquina.....	44
Quadro 20 - <i>Layout</i> do arquivo das restrições do peso por máquina.....	44
Quadro 21 - <i>Layout</i> do arquivo das restrições de OTs nulas.....	44
Figura 10- Tela de importação dos arquivos	45

Quadro 22 - Exemplo de arquivos textos	46
Figura 11- Tela de processamento da alocação	47
Figura 12 - Lista das OTs que foram importadas do arquivo texto	48
Quadro 23 - Criação do problema e das variáveis utilizando a biblioteca CHOCO	49
Quadro 24 - Criação da restrição para as cores	50
Quadro 25 - Restrições das OTs	50
Quadro 26 - Restrição das OTs que não podem estar em mais de uma máquina.....	51
Quadro 27 - Execução do algoritmo CSP e retorno da solução	51
Quadro 28 - Cálculo da melhor solução.....	52
Figura 13 - Resultado do algoritmo e a melhor solução encontrada	53
Quadro 29 - Tempo de resposta para encontrar todas as soluções	54
Figura 14- Gráfico de tingimento	62

SUMÁRIO

1 INTRODUÇÃO	12
1.1 OBJETIVOS DO TRABALHO.....	12
1.2 ESTRUTURA DO TRABALHO	13
2 FUNDAMENTAÇÃO TEÓRICA.....	14
2.1 PLANEJAMENTO E CONTROLE DA PRODUÇÃO.....	14
2.1.1 Controle de prioridade	16
2.1.2 Controle de capacidade	17
2.2 PROBLEMA DE SATISFAÇÃO DE RESTRIÇÕES (CSP)	17
2.2.1 Representação de uma restrição	19
2.2.2 Conceitos sobre Teoria dos Grafos.....	20
2.2.3 Alocação de recursos utilizando inteligência artificial para agendar e planejar	21
2.2.4 Busca com retrocesso (<i>backtracking</i>)	21
2.3 OTIMIZAÇÃO EM CSP.....	23
2.4 TRABALHOS CORRELATOS	24
3 ESPECIFICAÇÃO DO PROBLEMA	26
3.1 DIFICULDADES DA ÁREA DE TINGIMENTO	27
4 MODELAGEM DO PROBLEMA	29
5 DESENVOLVIMENTO DO SOFTWARE	32
5.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO	32
5.2 ESPECIFICAÇÃO.....	32
5.2.1 Casos de uso	33
5.2.2 Classes.....	33
5.2.3 Variáveis e domínios do CSP	36
5.2.4 Restrições	38
5.3 IMPLEMENTAÇÃO	40
5.3.1 CHOCO.....	40
5.3.2 Operacionalidade da implementação	42
5.3.2.1 <i>Layout</i> dos arquivos textos.....	42
5.3.2.2 Importação dos arquivos textos.....	44
5.3.2.3 Exemplo de arquivos textos	45
5.3.2.4 Processar o CSP	48

5.3.2.5 Encontrar a melhor solução.....	52
5.3.2.6 Resultado da execução do algoritmo CSP na base de dados	52
5.4 RESULTADOS E DISCUSSÃO.....	53
6 CONCLUSÕES	55
6.1 EXTENSÕES	55
REFERÊNCIAS BIBLIOGRÁFICAS	57
ANEXO A – Ordem de Tingimento	59
ANEXO B – Alocação das máquinas manualmente	60
ANEXO C – Gráfico de tingimento.....	61

1 INTRODUÇÃO

A entrega dos produtos na data prevista é o objetivo mais importante da área industrial de uma empresa têxtil. Deve-se planejar a produção de tal forma que as máquinas não fiquem sem alimentação aguardando a matéria-prima para o início de um processo. Para que este objetivo seja alcançado, é necessário que os materiais que fazem parte do processo de fabricação do produto estejam prontos na data prevista, visando assim o bom atendimento da produção na entrega do produto final ao cliente (REZENDE, 1992, p. 56).

Nas empresas têxteis que possuem o processo de tingimento dentro das suas atividades de produção, o aumento das necessidades do mercado (tipos de malhas, cores, processos de acabamento) combinado com a variedade de restrições das máquinas (capacidade por tipo de malha, tipo de malha que pode ser tingida na máquina, cores que a máquina pode tingir) torna demorado o processo de alocação manual das Ordens de Tingimento (OT). O esquecimento de uma simples restrição pode ocasionar a perda de mais de 500 quilos de malha numa única OT.

Surge então a necessidade de criar um sistema que utilize os algoritmos de Problemas de Satisfação de Restrições (*Constraint Satisfaction Problem – CSP*), onde o operador possa criar um conjunto de restrições de tingimento. Este sistema deverá analisar as OTs que estão disponíveis para alocação e determinar quais podem ser tingidas em cada máquina, de acordo com o conjunto de restrições pré-estabelecidos.

Na área de inteligência artificial, os algoritmos de CSP obedecem a uma representação padrão do problema, estruturado e muito simples, permitindo que a escrita da função sucessora e teste de objetivo sejam escritos de um modo genérico. A estrutura do grafo de restrições pode ser utilizada para simplificar o problema, proporcionando em alguns casos uma redução significativa da sua complexidade (RUSSEL; NORVIG, 2004, p. 134).

1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é desenvolver um sistema que apresente um mapa de alocação das máquinas de tingimento de uma empresa têxtil, com o foco na redução dos atrasos em decorrência de erros de análise das restrições.

Os objetivos específicos do trabalho são:

- a) modelar os processos de tingimento como CSP;
- b) definir as restrições das máquinas e da programação das malhas;
- c) carregar a programação das malhas definidas pelo departamento de Planejamento e Controle da Produção (PCP);
- d) disponibilizar um algoritmo baseado em CSP para escalonamento das tarefas de tingimento;
- e) apresentar o seqüenciamento da programação das OT's por máquina de tingimento, obtido através de CSP.

1.2 ESTRUTURA DO TRABALHO

O trabalho está organizado em seis capítulos. No capítulo 2 é apresentada a fundamentação teórica sobre Planejamento e Controle da Produção (PCP), seus conceitos de prioridade e capacidade e dos conceitos de *Constraint Satisfaction Problem* (CSP). Sobre CSP são demonstrados exemplos do algoritmo *backtracking*, uma breve explanação sobre otimização e alguns trabalhos correlatos. No capítulo 3 são especificadas as dificuldades de alocação das máquinas de tingimento e no capítulo 4 a modelagem deste problema. É detalhado o desenvolvimento do software no capítulo 5 e por fim são apresentadas as conclusões e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Nas seções seguintes são detalhados os conceitos de planejamento e controle de produção, em especial sobre o seqüenciamento das ordens de tingimento, os conceitos relativos à CSP e, por fim, serão descritos os trabalhos correlatos.

2.1 PLANEJAMENTO E CONTROLE DA PRODUÇÃO

O sistema de Planejamento, Programação e Controle da Produção (PPCP) é uma área de decisão da manufatura, cujo objetivo corresponde tanto ao planejamento como ao controle dos recursos do processo produtivo a fim de gerar bens e serviços. Este sistema também transforma informações, pois recebe dados sobre estoques existentes, vendas previstas, linha de produtos, modo de produzir e capacidade produtiva, resultando em Ordens de Produção (OP) (MARTINS; LAUGENI, 2005, p. 213).

O sistema de PPCP deve informar corretamente a situação corrente dos recursos, como pessoas, equipamentos, instalações e materiais, além de ser capaz de reagir de forma eficaz. A informação deve estar disponível e atualizada para que se possa oferecer aos clientes uma ampla variedade de serviços, melhorar o planejamento, a programação e o controle, fazendo com que a habilidade da empresa possa ser o diferencial para que a mesma seja de classe mundial, acrescentando que a informação deve estar disponível também para todos que fazem parte do processo produtivo.

Dentro deste processo produtivo, enquanto as OPs controlam todo o fluxo do processo, as Ordens de Tingimento (OT) controlam apenas o fluxo de produção da tinturaria.

Um vez definidas as OTs, essas devem ser seqüenciadas na fábrica. Este processo é conhecido como seqüenciamento ou programação de chão-de-fábrica. Uma vez seqüenciadas, deve-se acompanhar o desenvolvimento do programa e estabelecer instrumentos e variáveis de medida que possibilitem responder a questões relevantes, tais como: em que fase a ordem de produção de encontra?; quando o produto será terminado?; quanto tempo de máquina é necessário?, entre outras questões (MARTINS; LAUGENI, 2005, p. 219).

O seqüenciamento das OTs nas máquinas afeta diretamente o custo da produção, pois existem seqüências melhores e piores. Assim, quando se passa de um material a outro

semelhante em termos de necessidade de processamento, o custo de *setup* é relativamente menor que se passasse a outro material com características muito diferentes. Na prática, o custo de *setup* pode obrigar que se respeitem estas seqüências mais favoráveis de programação. O prazo de entrega do material é uma medida de urgência com que deve ser fabricado: quanto mais curto for este prazo, maior será a prioridade da fila de seqüenciamento (MOREIRA, 1998, p. 242).

O método de Menor Custo de Mudança (MCM) que foi descrito por Laugeni e Martins (2006, p. 440), é indicado para as tinturarias justamente por causa do tempo que é desperdiçado na lavagem da máquina. Inicialmente é criada uma matriz onde é relacionado o custo de mudança na execução das OTs, pois dependendo da cor que se vai pintar o tecido, é necessário lavar o equipamento.

DE (PRECEDÊNCIA)						
PARA		A	B	C	D	E
	A	-	75	90	60	42
	B	85	-	79	97	45
	C	62	91	-	87	75
	D	95	85	55	-	65
	E	55	85	65	95	-

Fonte: Martins e Laugeni (2005, p. 440).

Quadro 1 – Tabela exemplo do Menor Custo de Mudança (MCM)

No Quadro 1, quando se passa de C para D o custo é 55. Quando se passa de D para C o custo é 87 e assim por diante. A solução para este exemplo seria o seguinte: procura-se o menor custo de mudança, que é de E para A e custa 42. Desta forma, E é a primeira OT a ser executada e A, a segunda. Estando em A, deve-se ir para C, que é a de menor custo de mudança, igual a 62. Estando-se em C, deve-se ir para D, com o menor custo de 55. Estando-se em D só resta B para ser executada, a um custo de mudança de 97. Assim, a seqüência de execução deve ser: $E \rightarrow A \rightarrow C \rightarrow D \rightarrow B$, totalizando o custo de mudança em 256 (MARTINS e LAUGENI, 2005, p. 441).

Os Controles da Atividade de Produção (CAP) são técnicas de administração de capacidade e prioridade usadas para programar e controlar as operações de produção. O controle de prioridade assegura que as atividades de produção sigam o plano prioritário definidos pela necessidade de entrega das OTs e o controle da capacidade ajuda a controlar as máquinas para garantir que estejam provendo o volume de malha tingida e o tempo de

equipamento que é necessário para realizar a tarefa que foi programada. (MONKS, 1987, p. 347).

Existem restrições de decisão de prioridade para determinar a seqüência na qual as OTs devem ser executadas. Algumas restrições são simples, atribuindo as tarefas de acordo com um critério único, como: chegado primeiro, servido primeiro, data mais anterior de vencimento, menor tempo devido, menor tempo de processamento, maior tempo de processamento, pedido preferencial do cliente. Outras restrições são mais complexas, envolvendo cálculos de mais de uma variável, como é o caso da regra de Johnson, que fornece um tempo mínimo de processamento para seqüenciamento de n tarefas em 2 máquinas ou centros de trabalho em que a mesma seqüência de processamento deve ser seguida para todas as tarefas. Esse procedimento maximiza o tempo de operação simultâneo de ambos os centros de trabalho.

Outro exemplo é o cálculo da Relação Crítica (RC), que é uma regra dinâmica de programação, que classifica as tarefas de acordo com as que são mais urgentes quanto ao tempo de trabalho, de modo que as OTs possam ser entregues de acordo com o programa. Por ser uma regra dinâmica, é possível executar novamente o cálculo a cada dia para que o sistema forneça um controle rígido da produção (MONKS, 1987, p.357)

2.1.1 Controle de prioridade

As principais atividades do controle de prioridade são a liberação da OT, despacho e controle de posição. A liberação converte uma necessidade do planejamento para um pedido real na tinturaria (OT) e a programação é implementada via lista de despacho. Ela relaciona todas as tarefas disponíveis para um centro de trabalho e as hierarquiza por uma prioridade relativa (por exemplo, a relação crítica).

O controle de posição (relatórios) mede o andamento das OTs e mostra problemas e mudanças, tais como falhas internas, defeitos, erros no tamanho do lote e atrasos previstos. Com a transmissão dos dados a alta velocidade, os ajustes na lista de prioridades e no sistema podem ser baseados em prioridades válidas. (MONKS, 1987, p. 358 - 360).

2.1.2 Controle de capacidade

As principais funções do controle de capacidade são o controle de tempo de atendimento, equilíbrio e controle de entrada e saída. O tempo de atendimento total consiste no tempo que os pedidos esperam em atraso não-liberado mais o tempo de atendimento de fabricação. Do tempo de atendimento de fabricação, somente cerca de 10 por cento (ou menos) é em geral o tempo de funcionamento ou trabalho. A maior parte dele é tempo de espera ou de fila. Para a maioria dos centros de trabalho (máquinas), a capacidade é relativamente fixa e muito parecida com o fluxo através de um funil.

Se o trabalho for liberado para a produção em uma taxa mais rápida que sua taxa de produção, tanto o tempo de atendimento de fabricação como o trabalho em andamento aumentarão. Porém a produção não aumentará. O único resultado é um estoque se acumulando, uma maior falta de espaço, pedidos perdidos, expedição desnecessária e maiores custos de manutenção de estoque.

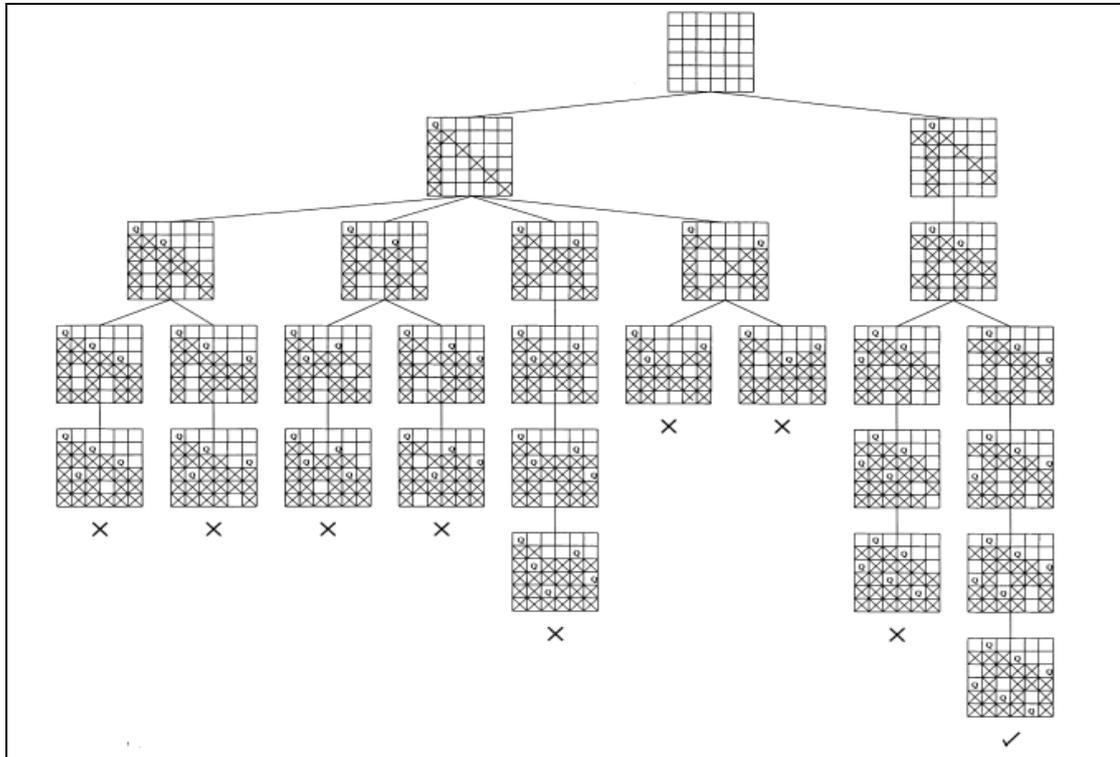
Uma segunda função do controle de capacidade é a de manter um equilíbrio efetivo da carga de trabalho em relação à capacidade das máquinas. O equilíbrio pode, em geral, ser melhorado mudando-se o trabalho de um período para outro. Se a carga total estiver desequilibrada, podem ser necessários equipamentos adicionais ou vias alternadas. (MONKS, 1987, p. 361).

2.2 PROBLEMA DE SATISFAÇÃO DE RESTRIÇÕES (CSP)

Basicamente, um problema CSP é composto por um número finito de variáveis, cada uma associada a um domínio finito, e a um número de restrições que as variáveis podem ter simultaneamente. A tarefa é atribuir um valor para cada variável que satisfaça todas as restrições (TSANG, 1993, p. 1).

O problema das n -rainhas é um quebra-cabeças muito conhecido pelos cientistas e normalmente utilizado para ilustrar como os algoritmos CSP resolvem este problema. Dado um número n inteiro qualquer, o problema é colocar n rainhas em n quadrados distintos num tabuleiro de xadrez com dimensões $n \times n$, satisfazendo as restrições que duas rainhas não podem estar se ameaçando. A restrição que rege este quebra-cabeças diz que uma rainha pode

ameaçar outras peças que estejam na mesma linha, coluna ou diagonal. A Figura 1 mostra a execução do algoritmo e uma possível solução para um problema com 6 rainhas.



Fonte: BRAILSFORD, POTTS e SMITH (1999, p. 1).

Figura 1 - Um a possível solução para o problema com seis rainhas

Um domínio de variáveis é um conjunto de todos os valores possíveis que podem ser atribuídos a uma variável. Se x é uma variável, então deve-se usar D_x para denotar o seu domínio.

Quando o domínio contém apenas números, as variáveis são chamadas de variáveis numéricas. O domínio de uma variável numérica pode ser números inteiros, racionais ou reais. Quando o domínio possui valores booleanos, as variáveis são chamadas de variáveis booleanas. Quando o domínio conter um tipo enumerado de objetos, as variáveis são chamadas de variáveis simbólicas. Por exemplo, uma variável que representa os dias de uma semana é chamada variável simbólica no qual o domínio finito é o conjunto {Domingo, Segunda, Terça, Quarta, Quinta, Sexta, Sábado} (TSANG, 1993, p. 6).

Uma **legenda** é um par variável-valor que representa a atribuição do valor para a variável. É utilizado $\langle x, y \rangle$ para indicar que a legenda tem a atribuição do valor v para a variável x . $\langle x, v \rangle$ é significativo apenas se v é um domínio de x ($v \in D_x$).

Uma **legenda composta** é a atribuição simultânea de valores para as variáveis. É utilizado $\langle x_1, v_1 \rangle \langle x_2, v_2 \rangle \dots \langle x_n, v_n \rangle$ para indicar a atribuição das variáveis v_1, v_2, \dots, v_n para x_1, x_2, \dots, x_n respectivamente.

Num conjunto de variáveis, a restrição (*constraint*) limita os valores que as variáveis podem ter simultaneamente. Uma restrição pode ser representada de diferentes formas, por exemplo: funções, matrizes, igualdades, entre outros.

A satisfação do problema é a relação binária entre o valor da variável atual e a restrição que foi definida.

A definição formal para o CSP é o tupla: (Z, D, C) , onde Z é o conjunto finito de variáveis (x_1, x_2, \dots, x_n) ; D é a função que relaciona cada variável em Z com um conjunto de objetos, sendo que estes objetos podem ser de qualquer tipo (numérico, booleano, entre outros). C é um número finito de restrição num subconjunto arbitrário de variáveis em Z (TSANG, 1993, p. 9).

Uma tarefa no CSP é atribuir um valor para cada variável onde todas as restrições sejam satisfeitas simultaneamente.

Um CSP é referido em alguns momentos como um Problema de Satisfação de Restrições Finitos (PSRF), pois na maioria dos problemas são utilizados números finitos de variáveis. O termo Problema de Satisfação de Restrições é frequentemente utilizado para problemas finitos, sendo que na sua definição as variáveis podem ter domínios infinitos, como por exemplo, um conjunto de números reais. Estes problemas são importantes e pertencem a uma outra classe de problemas que demandam um conjunto diferente de técnicas especializadas para serem resolvidos (TSANG, 1993, p. 10).

2.2.1 Representação de uma restrição

Uma restrição pode ser representada de diferentes formas. Restrições de variáveis numéricas podem ser representadas por equações ou inequações, como por exemplo, uma restrição binária $C_{x,y}$ que pode ser $x + y < 10$. Um jeito de representar uma restrição binária é através de matrizes ou valores booleanos. No exemplo da figura 2, é definido que a variável x terá valores inteiros entre 1 e 3, enquanto a variável y terá valores inteiros entre 4 e 7. A restrição diz que o “ $x + y$ deve ser ímpar” pode ser representada pela matriz da Figura 2.

Valores de X	C_{xy}	Valores de Y			
		4	5	6	7
1		1	0	1	0
2		0	1	0	1
3		1	0	1	0

Fonte: TSANG (1993, p. 11).

Figura 2 - Matriz representando a restrição entre x e y

Os resultados das variáveis X e Y são mostrados na :

X	Y
1	4
1	6
2	5
2	7
3	4
3	6

Figura 3 - Resultado para as variáveis X e Y

Um CSP binário é um CSP apenas com restrições binárias ou unárias. Um CSP com restrições não limitadas a unárias e binárias são referidas como um CSP genérico.

2.2.2 Conceitos sobre Teoria dos Grafos

A Teoria dos Grafos tornou-se uma importante parte das pesquisas sobre CSP, onde o grafo é uma tupla (V, U) onde V é um conjunto de nós e U é o conjunto de arcos. Um nó pode ser um objeto de qualquer tipo e um arco é um par de nós. Um grafo unidirecional é uma tupla (V, E) onde V é um conjunto de nós e E é um conjunto de bordas, cada uma com a sua coleção de exatamente dois elementos em V (TSANG, 1993, p. 14).

2.2.3 Alocação de recursos utilizando inteligência artificial para agendar e planejar

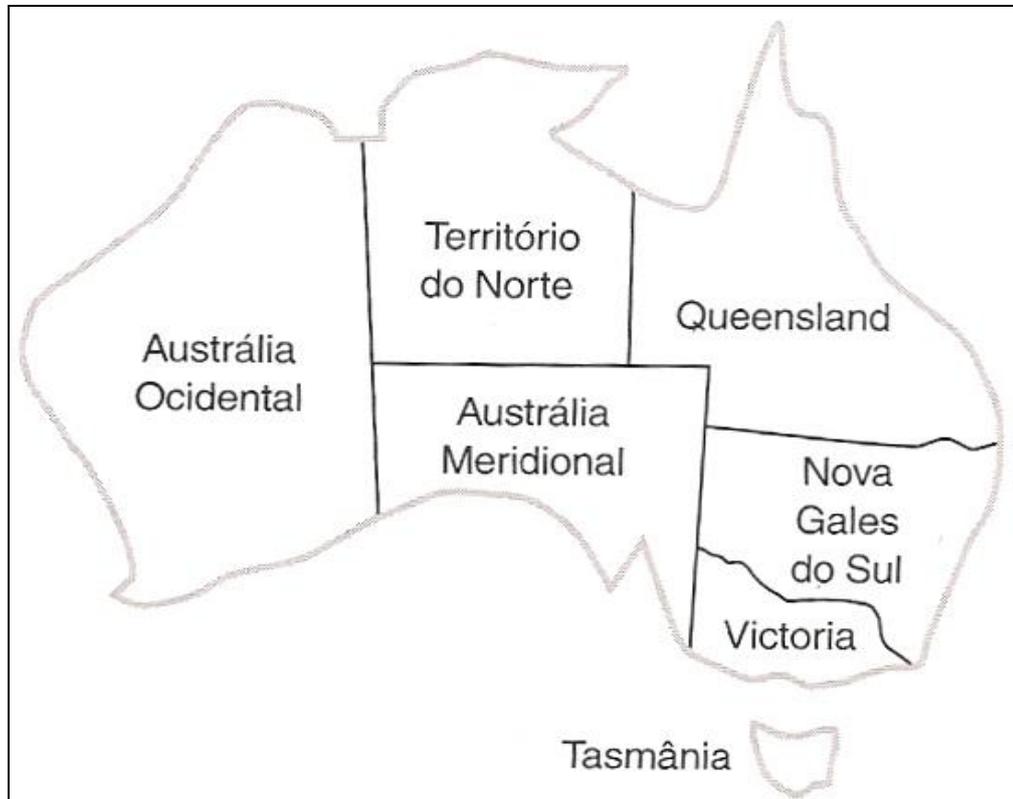
Alocação de recursos e agendamento são as aplicações mais conhecidas com o uso de CSP. O problema de seqüenciamento de carros é um exemplo de problema de agendamento que as técnicas de solução de CSP são aplicadas com sucesso. Um problema típico de agendamento é um problema onde é dado um conjunto de tarefas e perguntado para alocar recursos para eles. Cada tarefa pode solicitar um número de recursos, que inclui tempo, máquinas, ferramentas, mão-de-obra, entre outros.

A alocação de recursos, especialmente quando tempo de recursos compartilhados são envolvidos, é basicamente um CSP. Cada variável de um CSP representa uma requisição de recurso compartilhado. Por exemplo, a variável X pode representar uma requisição de uma máquina para um trabalho. O domínio da variável é o conjunto de possíveis valores que esta variável pode ter. O domínio de X no exemplo do seqüenciamento dos carros pode ser o conjunto de máquinas disponíveis na fábrica que possuem capacidade de executar o trabalho. Atribuir um valor para a variável representa a alocação de um recurso para o trabalho. A alocação de recursos é normalmente restrita de diversas formas. Por exemplo, entre as M máquinas disponíveis para o trabalho J , apenas as máquinas P , Q e R possuem capacidade para executar o trabalho J . Muitas vezes, uma máquina pode apenas processar um trabalho de cada vez. Algumas vezes, se o trabalho J utilizar a máquina M_1 , então deve utilizar certas ferramentas e certos engenheiros. A tarefa é alocar para cada variável o valor na qual todas as restrições sejam satisfeitas (TSANG, 1993, p. 25).

2.2.4 Busca com retrocesso (*backtracking*)

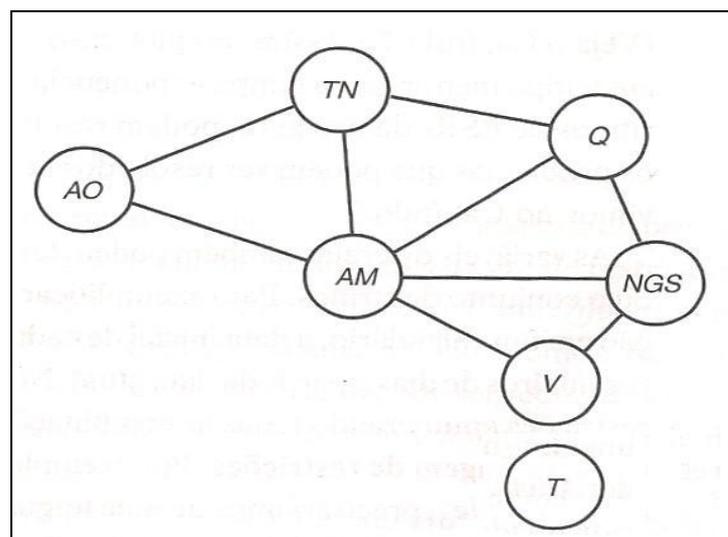
Dependendo da quantidade de variáveis que o problema possui, a árvore de busca gerada pelo algoritmo CSP pode criar uma ramificação muito grande, fazendo com que se torne um problema demorado para resolver. Isso ocorre, pois todos os algoritmos de busca de CSPs geram sucessores considerando atribuições possíveis apenas para uma única variável em cada nó da árvore de busca. Num dos exemplos mais conhecidos para demonstrar um CSP, chamado de coloração de mapas, o nó raiz de uma árvore de busca para colocar o mapa da Austrália poderia ser escolhido entre AM = vermelho, AM = verde e AM = azul. Este exemplo de CSP consiste em atribuir uma cor para cada região de modo que não haja regiões

vizinhas com a mesma cor. Na Figura 4 é possível demonstrar o mapa e na Figura 5 como fica definido o grafo de restrições.



Fonte: RUSSEL e NORVIG (2004, p. 135).

Figura 4 - Mapa dos principais estados e territórios da Austrália



Fonte: RUSSEL e NORVIG (2004, p. 135).

Figura 5 - Grafo de restrições do problema do mapa

A busca com retrocesso (*backtraking*) é uma busca em profundidade que escolhe valores para uma variável de cada vez e que efetua o retrocesso quando uma variável não tem valores válidos restantes a serem atribuídos. Os valores são testados no final de cada atribuição e caso o exista uma restrição que invalide o estado atual, é executado o retrocesso, não havendo assim a criação dos ramos do nó atual (RUSSEL e NORVIG, 2004, p. 138).

O fraco desempenho de um algoritmo de busca em profundidade pode ser melhorado se forem utilizadas funções heurísticas específicas do domínio, baseado no conhecimento que se tem sobre o problema. Desta forma, pode ser criada uma função para selecionar uma variável que ainda não foi atribuída, melhorando consideravelmente o desempenho do algoritmo. Na Figura 6 é possível analisarmos um algoritmo simples com retrocesso, que foi incluído algumas funções heurísticas para melhorar a velocidade de retorno da solução.

```

função PESQUISA-COM-RETROCESSO(psr) retorna uma solução ou falha
retornar RETROCESSO-RECURSIVO({ }, psr)
função RETROCESSO-RECURSIVO(atribuição, psr) retorna uma solução ou falha
se atribuição é completa então retornar atribuição
var ← SELECIONAR-VARIÁVEL-NÃO-ATRIBUÍDA(VARIÁVEIS[psr], atribuição, psr)
para cada valor em VALORES-DE-ORDEM-NO-DOMÍNIO (var, atribuição, psr) faça
  se valor é consistente com atribuição de acordo com RESTRICÇÕES[psr] então
    adicionar { var = valor } a atribuição
    resultado ← RETROCESSO-RECURSIVO (atribuição, psr)
    se resultado != falha então retornar resultado
  remover { var = valor } de atribuição
retornar falha
  
```

Fonte: NORVIG e RUSSEL (2004, p. 139).

Figura 6 - Algoritmo simples com retrocesso para CSP

2.3 OTIMIZAÇÃO EM CSP

Em aplicação de seqüenciamento na indústria, algumas soluções encontradas pelo CSP são melhores que outras. A com a atribuição de valores diferentes para a mesma variável, é possível encontrar soluções com custos diferentes. A tarefa em alguns casos é encontrar a melhor solução e por este motivo este problema é chamado de *Constraint Satisfaction Optimization Problem* (CSOP). (TSANG, 1993, p. 299).

Por definição, um CSOP é um CSP com uma função de otimização que mapeia todas as possíveis soluções para um valor numérico: (Z, D, C, f) onde (Z, D, C) é um CSP, e se S é um conjunto de soluções da tupla de (Z, D, C) , então $f: S \rightarrow$ valor numérico.

Sendo assim, a tarefa de um CSOP é encontrar a solução com o melhor (maior ou menor) valor de f (TSANG, 1993, p. 300).

O problema de agendamento de tarefas não é um CSP e sim um CSOP, onde apenas encontrar uma solução não é o suficiente.

Para encontrar a melhor solução, é necessário encontrar todas as soluções possíveis e então comparar os seus valores numéricos encontrados pela função f .

Tsang (1993, p. 301) descreve que existem dois importantes métodos para resolver um CSOP:

- a) algoritmo *Branch and Bound* (B&B);
- b) algoritmos genéticos ou *Genetic Algorithms* (GAs).

2.4 TRABALHOS CORRELATOS

Pires (2006) desenvolve em seu TCC um algoritmo de *backtracking* integrando com a ferramenta chamada *CSP && COP Builder* para solucionar CSPs. Esta ferramenta, desenvolvida pelo Grupo de Inteligência Artificial (GIA) da Fundação Universidade Regional de Blumenau (FURB), é utilizada em conjunto com o *dynDCSP*, que também foi desenvolvido pelo GIA e possui algoritmos CSPs de forma distribuída.

Santos (2005) utiliza o mesmo *framework* *dynDCSP* para implementar o algoritmo de *Constraint Optimization Problem* (COP) em ambiente distribuído chamado *Distributed Constraint Optimization Problem* (DCOP) para então poder especificar um problema de modelagem molecular e verificar a sua viabilidade. O que difere um COP de um CSP é que além de encontrar a solução, o COP procura pela solução de custo ótimo.

Bruns (2005) aplicou a técnica CSP de forma distribuída, chamada *Distributed Constraint Satisfaction Problem* (DCSP) para solucionar o problema de sincronismo de semáforos em uma malha viária, onde o mesmo afirma que “os benefícios destes sistemas são da ordem de 10% a 15% em relação à semáforos de plano fixo bem ajustados”. Nas grandes cidades, a possibilidade de fazer esta análise de forma centralizada tornar-se-ia de difícil realização e por este motivo, o *framework* *dynDCSP* foi utilizado para desenvolver esta solução.

Kammer (2006) apresentou no seu TCC o desenvolvimento de um sistema de

informação para logística de transporte utilizando a técnica CSP. Este sistema analisava as notas fiscais emitidas e distribuía entre os recursos de transportes disponíveis e suas restrições eram: destino, peso e custo de viagem do recurso de transporte. Foram utilizadas as bibliotecas do *framework* Choco para a modelagem do problema e concluiu-se que a utilização da técnica é bem empregada na solução de problemas ligados a logística.

3 ESPECIFICAÇÃO DO PROBLEMA

Numa empresa têxtil, o problema da tinturaria inicia-se no momento que o departamento comercial da empresa indica a quantidade de peças de cada produto que devem ser produzidas para entrega numa determinada semana. Fala-se em semana, pois na confecção o número de peças que podem ser produzidas na semana é normalmente calculado pelo número de funcionários disponíveis da costura, o tempo disponível que cada funcionário tem para trabalhar, o tempo que cada peça leva para ser costurada e o percentual de eficiência dos funcionários da costura.

Outro motivo por se falar de peças por semana é a dificuldade em determinar a data exata que cada produto será entregue, pois a quantidade de variáveis que podem atrapalhar o processo é muito grande. Normalmente acontecem atrasos na entrega de insumos como botões, linha ou então problemas nas máquinas de corte, falta de pessoal e assim por diante.

Tendo-se o número de peças que o departamento comercial solicitou para poder vender, estes números são repassados para o departamento de Planejamento e Controle de Produção (PCP) que se encarrega de analisar as quantidades solicitadas e inicia o processo de emissão das Ordens de Produção (OP). É a partir desse momento que o processo produtivo começa realmente a contar.

O departamento de PCP, tendo as OPs em mãos, passa para a fase de cálculo dos materiais que serão necessários para a produção e separa entre os que serão produzidos dentro da empresa e os que devem ser comprados. Estes dados são adquiridos através da ficha técnica de cada produto, permitindo saber a quantidade exata para a produção de cada peça e então é feita a multiplicação pelo número de peças que deverão ser produzidas. O resultado é uma lista de todos os materiais necessários para que os produtos sejam produzidos e entregues na semana definida.

O próximo passo é enviar a necessidade de malha para a tinturaria através das Ordens de Tingimento (OT), que indicam as malhas, cores e quantidades que devem ser produzidas, bem como o prazo de entrega. Ao final do processo de tingimento, a malha é colocada no estoque específico e aguarda pelo próximo processo, que é o corte.

Com a malha tingida e aguardando no estoque, é feita a separação das malhas necessárias para produzir uma OP. Esta malha é enviada para a atividade de corte onde são criados os enfeitos e cortados. O processo de enfeitar uma malha consiste em pegar o rolo de malha, estender em camadas numa mesa comprida e então a máquina de corte puxa esta

malha e começa a cortar as partes, que então são enviadas para as próximas fases como bordado, estamparia, preparação, costura e por última a embalagem, quando finalmente chegam no estoque.

3.1 DIFICULDADES DA ÁREA DE TINGIMENTO

Numa empresa têxtil verticalizada, que compreende malharia, beneficiamento estamparia, bordado e confecção, dentro do mesmo parque fabril, o processo de tingimento pode chegar a consumir mais de 1/3 do tempo total de uma Ordem de Produção (OP). Dependendo do tipo e da composição da malha a ser tingida, o fluxo e o processo de tingimento podem variar bastante. Um exemplo de OT encontra-se no Quadro 30, onde pode ser observado no tópico “Roteiro” a quantidade de atividades necessárias para entregar a malha pronta no estoque.

Para iniciar o processo de tingimento da malha, é necessário emitir uma receita, que é muito parecido no conceito, com uma receita de bolo. A receita de tingimento é definida pela composição da malha e nesta receita são indicados os corantes, produtos químicos, a quantidade de água que será utilizada no processo e os passos que devem ser seguidos para que ao final tenhamos a tonalidade correta e dentro dos padrões de qualidade.

O processo de tingimento geralmente é dividido em três partes: preparação, tingimento e lavação. A preparação tem a finalidade de remover as impurezas da malha provenientes da malharia e do próprio fio. O tingimento dá cor a malha enquanto a lavação remove o corante que não reagiu com a fibra. Como pode ser observada no Anexo C, a receita de tingimento mostra a temperatura que o banho tem que estar e em qual ponto deve ser acrescentado o corante ou o produto químico. Cada fibra possui suas particularidades para fixar o corante, algumas precisam estar com a temperatura do banho mais alta no momento da fixação, outras precisam de mais produto químico para ter o mesmo efeito. As fibras mais comuns são Poliéster (PES), Algodão (CO) e Viscose (CV).

Devido às características das fibras, existem regras no processo de tingimento que devem ser seguidas para que a cor desejada seja obtida. As máquinas de tingimento podem ser de Baixa Temperatura (BT) ou de Alta Temperatura (HT). As BT são utilizadas para tingir fibras como algodão e viscose, aonde a temperatura máxima no processo de tingimento chega

a 95 graus, já as HT atingem temperaturas elevadas com 130 graus, necessária para tingir a fibra de poliéster.

A cor da receita também influencia no momento da alocação das máquinas, pois a seqüência que as OTs irão passar em cada máquina é determinada pelo degrade de cor, ou seja, a tonalidade dos lotes programados vai ficando mais escura ou mais clara e dessa forma, não precisamos lavar a máquina entre os tingimentos, pois quando estamos lavando a máquina não estamos produzindo. Ao alocar uma OT na cor branca e em seguida alocar uma da cor preta, não será necessário lavar a máquina por dentro, pois o resíduo que por ventura ficar na máquina não irá influenciar o tingimento da OT preta. O contrário já não é verdade, pois ao tingir o preto primeiro, o resíduo desse processo por ser mais escuro, irá influenciar no resultado da OT branca, ou seja, a ordem da alocação irá influenciar diretamente no tempo total de processamento das tarefas da tinturaria.

O peso da OT é restringido pela capacidade da máquina que será utilizada. A máquina possui a capacidade máxima e mínima para tingir determinada malha. Se exceder a quantidade determinada, existe o perigo de trancar a malha dentro do equipamento e ocasionar problemas como manchar a malha ou até danificar a máquina.

Após o tingimento, a malha segue para os processos de acabamento que serão realizados de acordo com o fluxo definido no desenvolvimento. Essa parte do processo define a largura, gramatura, toque, e estabilidade dimensional (encolhimento e torção) da malha.

4 MODELAGEM DO PROBLEMA

A modelagem do problema da tinturaria a ser resolvido pelo CSP consiste em identificar a tupla (variáveis, domínio e restrições) que será analisada pelo algoritmo. Inicialmente foram levantados os dados da tinturaria como máquinas, materiais, cores e pesos para se ter idéia das restrições que deveriam ser utilizadas na modelagem. As máquinas são separadas conforme as suas características e restrições como: tipos de materiais que podem ser tingidos; cores que podem ser tingidas e por fim a capacidade de tingimento da máquina.

A restrição do tipo de material consiste em identificar se o material que será tingido é composto de fibras de algodão, poliéster, viscose ou outra fibra.

A restrição da cor é utilizada para aumentar a produtividade da máquina, evitando assim as constantes limpezas internas. Um exemplo disso é que ao tingir uma cor escura e depois uma clara, é necessário fazer a lavagem da máquina para retirar os resíduos de corante escuro para então carregar a máquina com a malha que será tingida na cor clara.

A capacidade de tingimento da máquina restringe o limite máximo e mínimo de malha que a máquina consegue mover dentro dela, pois a malha fica em movimento durante o processo de tingimento. Se esta capacidade for ultrapassada, a máquina não conseguirá puxar a malha e conseqüentemente irá trancar.

Como o objetivo do CSP é resolver problemas analisando as restrições, foi necessário analisar cada um dos componentes do problema separadamente para então identificar quais seriam as variáveis, domínios e por último as suas restrições. Ao final do processamento do algoritmo, teríamos que conseguir encontrar a solução que seria qual máquina deveria fazer qual OT.

No início foi adicionada uma variável para definir que materiais poderiam ser tingidos em cada máquina. Foi utilizado como exemplo 3 máquinas (TM01, TM02 e TM03) e foram definidas as variáveis como sendo TM01_MATERIAL, TM02_MATERIAL e TM03_MATERIAL.

Tendo as variáveis dos materiais definidas, o próximo passo seria definir o domínio de cada variável. Para isso foi criado um único domínio com todos os materiais que possuem OTs pendentes na fábrica. Dessa forma, seriam analisadas todas as possibilidades de alocação de materiais em cada máquina.

O objetivo de criar o domínio baseado apenas nas OTs pendentes seria uma forma de otimização dos dados para que o processo de cálculo fosse mais rápido. Não existe a

necessidade de enviar materiais que não estão relacionados a nenhuma OT, pois estes jamais serão alocados na produção.

Após a definição das variáveis dos materiais, foi necessário criar as variáveis das cores e dos pesos de cada máquina. A definição foi exatamente a mesma que as variáveis dos materiais, onde para cada máquina foi atribuído uma variável de cor e outra para o peso, ficando então: TM01_COR; TM01_PESO; TM02_COR; TM02_PESO; TM03_COR; TM03_PESO.

O domínio das variáveis das cores foi definido como sendo todas as cores que tivessem OTs pendentes na empresa, e o domínio dos pesos como sendo todos os pesos das OTs pendentes na empresa, otimizando novamente o tempo de processamento ao restringir o tamanho do domínio.

O problema da criação das variáveis ainda não havia terminado, pois havia a necessidade de criar a última variável que iria conter o número da OT. Desta forma foi criada a variável OT, sendo uma para cada máquina: TM01_OT; TM02_OT; TM03_OT. O domínio das variáveis OT ficou sendo o número de todas as OTs pendentes.

Terminando a definição das variáveis e domínios, passou-se a definir as restrições. Ao definir as restrições para um CSP, deve-se indicar se aceita ou rejeita e são determinadas conforme as características de cada máquina, ou seja, se a máquina: somente aceita a cor branca; se rejeita apenas a cor preta; se rejeita o material poliéster; se aceita quantidades entre 100 e 200 kg. O exemplo mostrado no Quadro 2 permite demonstrar uma lista das restrições por máquina e a opção de restringir ou aceitar.

Máquina	Cor	Opção
TM01	BRANCA	ACEITA
TM01	PRETA	REJEITA
TM02	AZUL	ACEITA
TM02	PRETA	ACEITA
TM03	BRANCA	REJEITA
TM03	PRETA	REJEITA
TM03	AZUL	ACEITA

Quadro 2 - Restrição das cores por máquina

A restrição de materiais por máquina é demonstrada no Quadro 3 e pode ser observado

as opções de aceitar e rejeitar o material na máquina.

Máquina	Material	Opção
TM01	10601 – MEIA MALHA	ACEITA
TM01	40301 – COTTON	ACEITA
TM01	55419 – MEIA MALHA PV	REJEITA
TM02	55419 – MEIA MALHA PV	ACEITA
TM03	20001 – RIBANA 2X1	ACEITA
TM03	10601 – MEIA MALHA	ACEITA

Quadro 3 - Restrição dos materiais por máquina

As restrições de peso por máquina não precisa utilizar o status pois restringe utilizando os operadores maior/igual (\geq) e menor/igual (\leq), conforme demonstrado no Quadro 4.

Máquina	Capacidade Mínima	Capacidade Máxima
TM01	100	130
TM02	150	180
TM03	150	180

Quadro 4 - Restrição dos pesos por máquina

A restrição das OTs relaciona os números das Ordens de Tingimento com o seu material, cor e peso. Desta forma, quando o algoritmo atribuir um valor para cada variável e fizer os teste de restrição de material, cor e peso terá que verificar se estes três valores estão de acordo com uma OT da lista. Esta restrição não está relacionada com nenhuma máquina e também não existe a opção de rejeição, pois somente o nó do grafo que tiver as quatro variáveis (OT, MATERIAL, COR e PESO) correspondentes a um dos registros da lista de restrição de OT é que pode ser considerado um estado válido.

Como a mesma OT não pode ser alocada a outra máquina ao mesmo tempo, a última restrição desta modelagem faz esta verificação.

5 DESENVOLVIMENTO DO SOFTWARE

Neste capítulo é descrito o desenvolvimento do software, apresentando-se os requisitos do sistema, especificação e a sua implementação, demonstrando-se as técnicas, ferramentas utilizadas e a operacionalidade do software. Para finalizar são comentados os resultados e discussões.

5.1 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

O sistema de alocação deve possuir os seguintes Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF):

- a) o sistema deve permitir a importação das restrições das máquinas (RF);
- b) o sistema deve permitir a importação das restrições das malhas (RF);
- c) o sistema deve permitir a importação das restrições das cores (RF);
- d) o sistema deve permitir a importação das ordens de produção de tingimento (RF);
- e) o sistema deve disponibilizar uma interface para poder executar o algoritmo CSP (RF);
- f) o sistema deve disponibilizar uma interface baseada em caixas de texto, *frames* e botões para poder visualizar o resultado da alocação das máquinas (RF);
- g) o sistema deve ser implementado utilizando a linguagem Java (RNF).

A utilização da linguagem Java para implementação do sistema justifica-se pela necessidade de bibliotecas *open sources*, como a biblioteca CHOCO.

5.2 ESPECIFICAÇÃO

Para a especificação foi utilizada a ferramenta *Enterprise Architect* (EA) (ENTERPRISE ARCHITECT, 2009) que utiliza a linguagem visual *Unified Modeling Language* (UML) (BOOCH; RUMBAUGH, JACOBSON, 2002). Os diagramas de casos de uso, classes e seqüência são utilizados para dar-se o entendimento do funcionamento do

software.

5.2.1 Casos de uso

O diagrama de casos de uso é demonstrado na Figura 7, onde se pode observar a simplicidade do software, pois o processo principal concentra-se no caso de uso 03, que possui a processamento do CSP.

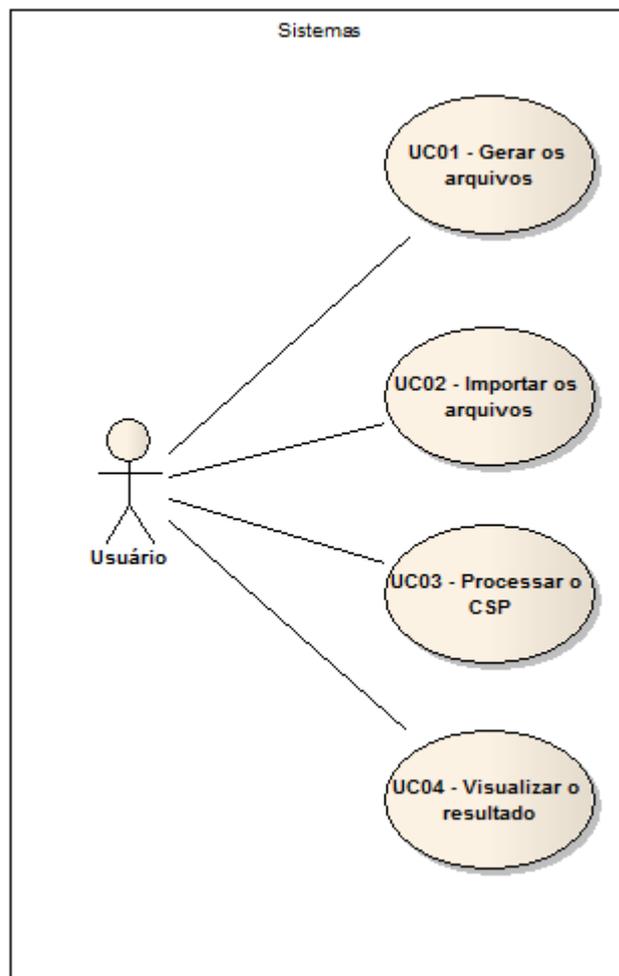


Figura 7 - Diagrama de casos de uso do software

5.2.2 Classes

As classes do sistema foram separadas em importação dos dados e processamento do algoritmo CSP. Na Figura 8 são demonstradas as classes utilizadas para importação dos dados

dos arquivos textos e na Figura 9 são demonstradas as classes utilizadas no processamento da alocação.

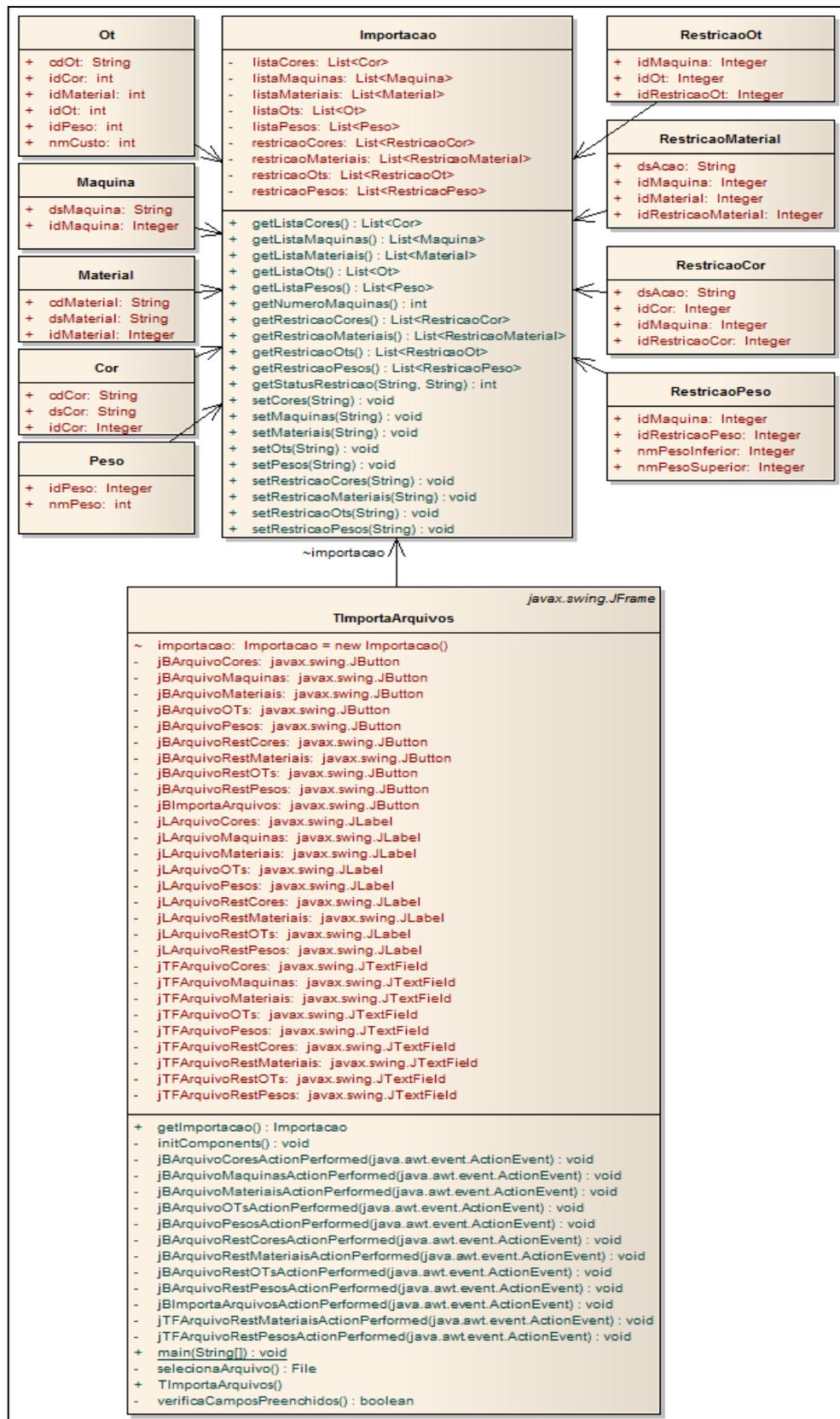


Figura 8 - Diagrama das classes de importação dos dados

No Quadro 5 são demonstradas as responsabilidades de cada classe da Figura 8.

Classe	Responsabilidade
RestricaoMaterial	Utilizada na importação das restrições das máquinas referente aos materiais que podem ser processados ou não. O campo STATUS indica se (A)ceita ou (R)ejeita o material na máquina.
RestricaoCor	Utilizada na importação das restrições das máquinas referente às cores que podem ser processadas ou não. O campo STATUS indica se (A)ceita ou (R)ejeita a cor na máquina.
RestricaoPeso	Utilizada na importação das restrições das máquinas referente aos pesos que podem ser processados ou não. Os pesos são determinados em faixas por máquina.
RestricaoOt	Utilizada na importação das OTs, onde é feito o relacionamento entre a OT, material, cor e peso.
Importacao	Utilizada para armazenar as listas dos materiais, cores, pesos, OTs e as suas restrições.
TImportaArquivos	Utilizada para selecionar os arquivos que serão importados.

Quadro 5 - Responsabilidades das classes de importação

Os dados são disponibilizados pelo sistema de gerenciamento de OTs através de arquivos textos e as classes de importação são responsáveis pela leitura dos dados e disponibilizam na classe `Importacao` os métodos para retornar as listas. Depois de carregados os dados, é necessário executar o algoritmo que está na classe `Tinturaria`. Esta classe é chamada pela classe `TProcessaAlocacao`, onde é possível verificar o resultado da alocação por máquina.



Figura 9 - Classes para processamento da alocação

5.2.3 Variáveis e domínios do CSP

Antes da execução do algoritmo, é necessário atribuir os valores aos domínios que estão associados às variáveis. Na modelagem do sistema foi definido que cada variável deveria estar relacionado a sua máquina, conforme exemplo abaixo utilizando a máquina TM01:

- variável `TM01_COR` terá no domínio todas as cores de todas as OTs, conforme

Quadro 6;

- b) variável TM01_MATERIAL terá no domínio todos os materiais de todas as OTs, conforme Quadro 7;
- c) variável TM01_PESO terá no domínio todos os pesos de todas as OTs, conforme Quadro 8;
- d) variável TM01_OT terá no domínio todos os números de todas as OTs, conforme Quadro 9.

IdCor	Cor	Descrição
1	001	BRANCO
2	422	AZUL
3	539	PRETO
4	643	VERMELHO

Quadro 6 – Exemplo de valores do domínio das cores

IdMaterial	Material	Descrição
1	50501	Meia Malha Penteadada
2	50502	Meia Malha OpenEnd
3	90401	Viscoflex
4	10506	Plush
5	30405	Ribana
6	30506	Ribana Penteadada

Quadro 7 - Exemplo de valores do domínio dos materiais

IdPeso	Peso
1	90
2	120
3	420
4	270
5	430
6	250

Quadro 8 – Exemplo de valores do domínio de pesos

OT	Material	Cor	Peso
100102	50501	Branca	90
101345	50502	Azul	120
101347	90401	Preta	420
120748	10506	Vermelha	270
120945	30405	Branca	430
130784	30506	Azul	250

Quadro 9 – Exemplo de valores do domínio das OTs

5.2.4 Restrições

Antes do início da execução do algoritmo, é necessário definir as restrições que cada variável deverá ter. Estas restrições são importadas através das classes `RestricaoMaterial`, `RestricaoCor`, `RestricaoPeso` e `RestricaoOt` e armazenadas na classe `Importacao` para posteriormente ser utilizada na classe `Tinturaria`.

Alguns exemplos de restrições de materiais são demonstrados no Quadro 10, onde é indicado se a restrição é de aceitação ou rejeição. Quando existem rejeições com status de aceitação, é feito um conjunto utilizando a clausula OU e quando forem rejeições com status de rejeição, é feito um conjunto utilizando a clausula E, ficando da seguinte forma: materiais aceitos na máquina TM01 = 50501 OU 90401; materiais com rejeição na máquina TM01 = 50502 E 30506.

Máquina	Material	Status
TM01	50501	Aceita
TM01	90401	Aceita
TM01	50502	Rejeita
TM01	30506	Rejeita
TM02	30405	Aceita
TM02	50501	Rejeita

Quadro 10 - Exemplo de restrições de materiais nas máquinas

Nas restrições de cores demonstradas no Quadro 11, é possível observar os mesmos

critérios de E e OU utilizados nas restrições de materiais, onde são criadas as listas das cores aceitas e rejeitadas separadamente e ao criar as restrições, são utilizados os critérios E para rejeição e OU para aceitação.

Máquina	Cor	Status
TM01	Branca	Aceita
TM01	Rosa	Aceita
TM01	Preta	Rejeita
TM01	Azul	Rejeita
TM02	Branca	Rejeita
TM02	Preta	Aceita
TM02	Azul	Aceita
TM02	Rosa	Rejeita

Quadro 11 - Exemplo de restrições de cores nas máquinas

Nas restrições de peso, muda-se um pouco pois não existe o status de rejeição, somente de aceitação e os valores são determinados em faixas de valores por máquina, conforme demonstrados no Quadro 12.

Máquina	Peso Mínimo	Peso Máximo
TM01	100	120
TM02	150	180
TM03	150	180

Quadro 12- Exemplo de restrições de pesos nas máquinas

5.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas para implementar o sistema. Inicialmente é descrita a biblioteca CHOCO (CHOCO, 2009), que foi utilizada na implementação através da ferramenta NetBeans (NETBEANS, 2009) e em seguida é detalhada a implementação do sistema.

5.3.1 CHOCO

O CHOCO é uma biblioteca desenvolvida em JAVA para solução de CSPs que possui algoritmos de *backtracking* para aumentar o desempenho da busca em profundidade. Permite que o desenvolvedor faça a modelagem do problema e não se preocupe como o ele é resolvido.

Esta biblioteca possui alguns objetos básicos para a solução do CSP:

- a) Problema (*problem*) é o elemento central do CHOCO. As variáveis e restrições são associadas a ele;
- b) Variáveis (*variables*) são os “desconhecidos” do problema. Através da atribuição dos valores do domínio (*domain*) que é definido por um conjunto de valores ou simplesmente através de um valor mínimo e máximo. O domínio é criado no mesmo momento que a variável;
- c) Restrições (*constraints*) definem a relação que deve ser satisfeita entre o valor e a variável.

No desenvolvimento do programa, inicialmente deve ser declarado o problema e em seguida definidas as variáveis com os seus domínios. Os domínios inteiros podem ser definidos informando o seu valor final ou uma faixa de valores.

Após estes passos, devem-se declarar as restrições e adicioná-la ao problema. As restrições podem ser simples (igual, não igual, maior, menor, maior igual, entre outros) ou compostas (se $\text{variável}_x + \text{variável}_y = 7$ então $\text{variável}_x = 1$).

O problema das n -rainhas pode ser facilmente resolvido através da classe descrita no Quadro 13.

```

package chocosamples;
import choco.*;
public class Queen {
    public static void main(String[] args) {
        System.out.println(new Queen().demo());
    }
    public String demo() {
        StringBuffer ret = new StringBuffer();
        int n = 6;
        // Cria o problema
        Problem pb = new Problem();
        // Cria as variáveis
        choco.integer.IntDomainVar[] queens = new
choco.integer.IntDomainVar[n];
        for (int i = 0; i < n; i++) {
            queens[i] = pb.makeEnumIntVar("Q" + i, 1, n);
        }
        // Cria as restrições (constraints)
        for (int i = 0; i < n; i++) {
            for (int j = i + 1; j < n; j++) {
                int k = j - i;
                pb.post(pb.neq(queens[i], queens[j]));
                pb.post(pb.neq(queens[i], pb.plus(queens[j], k)));
                pb.post(pb.neq(queens[i], pb.minus(queens[j], k)));
            }
        }
        // Cria o objeto que irá resolver e define o solver como true,
        // para retornar todas as soluções
        Solver s = pb.getSolver();
        pb.solve(true);
        // Resultado
        ret.append("O problema das n-rainhas pergunta como colocar n
rainhas num tabuleiro de xadrez " +
                "com dimensões n x n da forma que nenhuma delas possa
atacar a outra com apenas um movimento.\n");
        ret.append("Variável n = " + n + "\n\n");
        String line = "+";
        for (int i = 0; i < n; i++) line += "----+";
        line += "\n";
        for (int sol = 0; sol < s.getSearchSolver().solutions.size();
sol++) {
            Solution solution = (Solution)
s.getSearchSolver().solutions.get(sol);
            ret.append(line);
            for (int i = 0; i < n; i++) {
                ret.append("|");
                for (int j = 0; j < n; j++) {
                    ret.append((solution.getValue(i) == j + 1) ? " * |"
: "  |");
                }
                ret.append("\n" + line);
            }
            ret.append("\n\n\n");
        }
        return ret.toString();
    }
}

```

Quadro 13 - Exemplo das n -rainhas utilizando a biblioteca CHOCO

5.3.2 Operacionalidade da implementação

Basicamente o sistema consiste em importar os arquivos textos contendo as informações das OTs, processar o algoritmo CSP e exibir o resultado da alocação das máquinas. Para que isso seja possível, o sistema da tinturaria deve exportar os dados em formato texto, separados em: OTs; materiais; cores; pesos; restrições dos materiais por máquina; restrições das cores por máquina; restrições dos pesos por máquina; restrição das OTs nulas por máquina.

5.3.2.1 *Layout* dos arquivos textos

O arquivo de materiais deve conter apenas os materiais que são utilizados pelas OTs, não sendo necessário informar todos os materiais do cadastro do sistema de tinturaria. Isso já é uma forma de restringir a quantidade de registros que o algoritmo irá testar para chegar numa solução. O Quadro 14 mostra o *layout* do arquivo texto.

Campo	Posição Inicial	Posição Final	Observação
idMaterial	1	4	Código do material que será utilizado para referenciar com a OT
cdMaterial	5	19	Código do material no sistema da tinturaria
dsMaterial	20	69	Descrição do material no sistema da tinturaria

Quadro 14 - *Layout* do arquivo texto dos materiais

O arquivo de cores deve conter apenas as cores que são utilizadas pelas OTs, não sendo necessário informar todas as cores do cadastro do sistema da tinturaria. O Quadro 15 mostra o *layout* do arquivo texto.

Campo	Posição Inicial	Posição Final	Observação
idCor	1	4	Código da cor que será utilizada para referenciar com a OT
cdCor	5	10	Código da cor no sistema da tinturaria
dsCor	11	60	Descrição da cor no sistema da tinturaria

Quadro 15 – *Layout* do arquivo texto das cores

O arquivo de pesos deve conter apenas os pesos que são utilizados pelas OTs. O Quadro 16 mostra o *layout* do arquivo texto.

Campo	Posição Inicial	Posição Final	Observação
idPeso	1	4	Código do peso que será utilizado para referenciar com a OT
nmPeso	5	9	Peso em quilos da OT

Quadro 16 - *Layout* do arquivo texto dos pesos

O arquivo das OTs deve conter todas as que estiverem pendentes, informando o código do material, da cor e do peso para relacionar com as suas respectivas listas. A utilização dessas listas contendo campos ID para fazer o relacionamento deve-se ao fato que a biblioteca CHOCO utiliza variáveis do tipo inteiro para fazer a execução do algoritmo. Pode-se observar que no Quadro 17 existem os campos para relacionar com as listas de material, cor e peso e também existe um campo de custo, onde é informado um número para indicar que a cor é mais escura que a outra. No momento da definição da seqüência do tingimento, este campo irá servir como critério.

Campo	Posição Inicial	Posição Final	Observação
idOT	1	4	Código da OT
cdOT	5	10	Número da OT no sistema da tinturaria
idCor	11	14	Código da cor para relacionar com a lista das cores
idMaterial	15	18	Código do material para relacionar com a lista dos materiais
idPeso	19	22	Código do peso para relacionar com a lista dos pesos
nmCusto	23	26	Custo do tingimento da cor

Quadro 17 - *Layout* do arquivo texto das OTs

No arquivo de restrição dos materiais por máquina, são relacionadas ao lado de cada restrição se a máquina de tingimento aceita (ação = A) ou se rejeita (ação = R) o material. O Quadro 18 demonstra o *layout*.

Campo	Posição Inicial	Posição Final	Observação
idRestricaoMaterial	1	4	Código da restrição do material
idMaquina	5	8	Código da maquina que irá receber a restrição
idMaterial	9	12	Código do material que será aceito/rejeitado
dsAcao	13	13	Ação que será tomada pelo algoritmo. (A)ceita ou (R)ejeita o material na máquina

Quadro 18 - *Layout* do arquivo texto das restrições dos materiais por máquina

No arquivo de restrições das cores por máquina, são relacionadas ao lado de cada restrição se a máquina de tingimento aceita (ação = A) ou se rejeita (ação = R) a cor. O Quadro 19 demonstra o respectivo *layout*.

Campo	Posição Inicial	Posição Final	Observação
idRestricaoCor	1	4	Código da restrição da cor
idMaquina	5	8	Código da maquina que irá receber a restrição
idMaterial	9	12	Código da cor que será aceita/rejeitada
dsAcao	13	13	Ação que será tomada pelo algoritmo. (A)ceita ou (R)ejeita a cor na máquina

Quadro 19 - *Layout* do arquivo texto das restrições das cores por máquina

No arquivo de restrições dos pesos por máquina, são relacionados os valores máximos e mínimos que cada máquina pode comportar, não precisando desta forma do campo de ação (dsacao) para saber se aceita ou rejeita. O Quadro 20 demonstra o *layout*.

Campo	Posição Inicial	Posição Final	Observação
idRestricaoPeso	1	4	Código da restrição do peso
idMaquina	5	8	Código da maquina que irá receber a restrição
nmPesoInferior	9	12	Peso mínimo da OT para que a máquina possa ser operada
nmPesoSuperior	13	16	Peso máximo da OT para que a máquina possa ser operada

Quadro 20 - *Layout* do arquivo das restrições do peso por máquina

No arquivo das restrições nulas por máquina, são relacionadas às OTs que indicam a possibilidade de não haver OT válida para ser seqüenciada na máquina.

Campo	Posição Inicial	Posição Final	Observação
idRestricaoNula	1	4	Código da restrição do OT nula
idMaquina	5	8	Código da maquina que irá receber a restrição
idOT	9	12	Código da OT nula, que está na lista das OTs

Quadro 21 - *Layout* do arquivo das restrições de OTs nulas

5.3.2.2 Importação dos arquivos textos

A importação de todos os arquivos textos necessários para o processamento do algoritmo são executadas numa única tela, demonstrada na Figura 10. Esta tela, ao carregar os arquivos textos, disponibiliza para a classe `Tinturaria` os arquivos em forma de lista, ou seja, uma lista para cada arquivo importado.

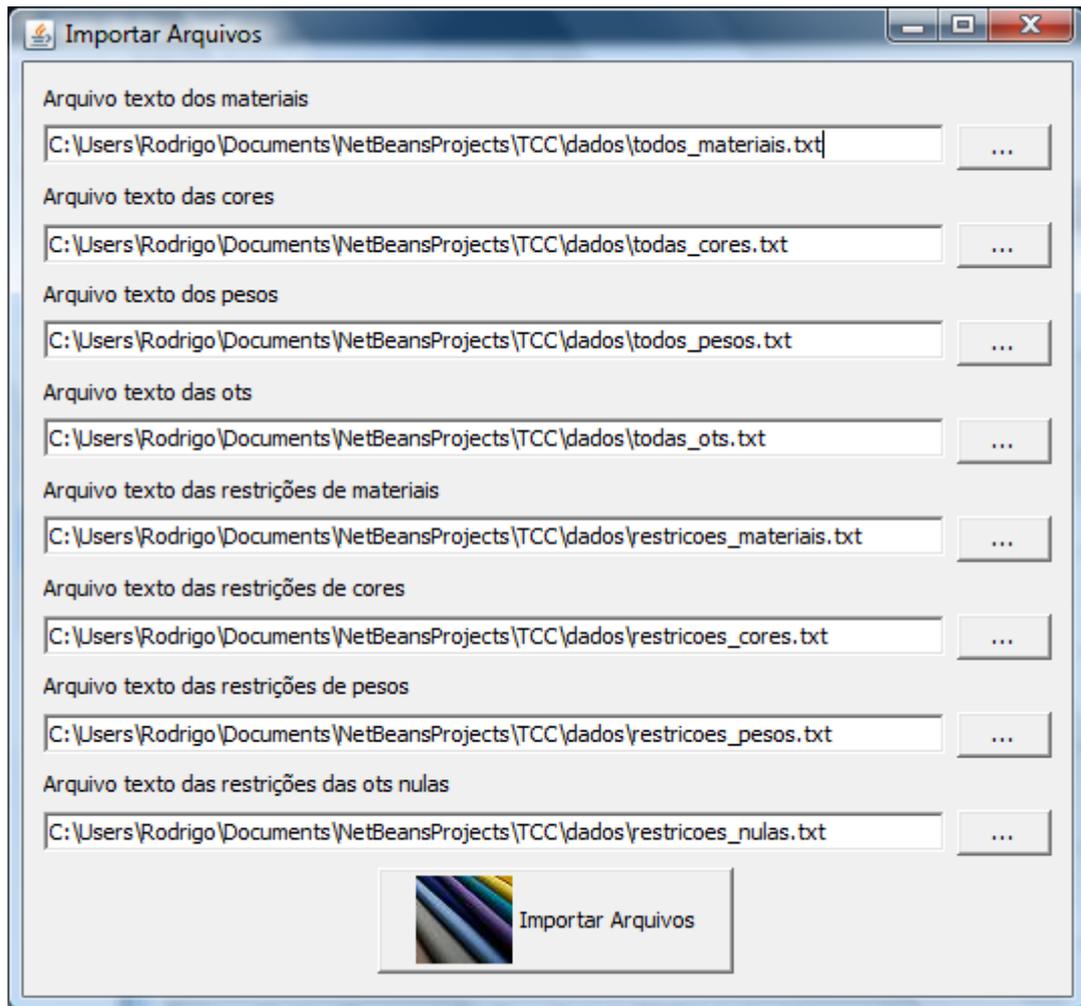


Figura 10- Tela de importação dos arquivos

5.3.2.3 Exemplo de arquivos textos

Para demonstrar o uso do sistema, foram gerados os arquivos contendo informações de algumas OTs para serem alocadas em 3 máquinas, conforme relacionado no Quadro 22.

```

todas_maquinas.txt
=====
1   TM01
2   TM02
3   TM03

todas_ots.txt
=====
1   0117151   1   1   1
2   0117171   1   2   2
3   0117182   2   2   3
4   0117193   1   1   4
5   0117204   1   1   5
6   9999015   3   3   9999
7   9999025   3   3   9999
8   9999035   3   3   9999

todos_materiais.txt
=====
1   04.071.0041   FIO 167/48 100% POLIESTER
2   04.071.0042   FIO DE ALGODAO
3   04.071.0043   FIO DE ALGODAO
4   04.071.0044   FIO DE ALGODAO
5   NULO          MAQUINA SEM TRABALHO

todas_cores.txt
=====
1   000   BRANCO
2   001   PRETA
3   NULA  NULA

todos_pesos.txt
=====
1   1
2   2
3   3

restricoes_cores.txt
=====
2   2   1   A
3   2   3   A

restricoes_nulas.txt
=====
1   1   7
2   1   8
4   2   6
5   2   8
7   3   6
8   3   7

restricoes_pesos.txt
=====
1   1   1   1000
2   2   1   1000
3   3   1   1000

```

Quadro 22 - Exemplo de arquivos textos

Na tela inicial do sistema, deve-se executar a opção do menu Operações → Processa Alocação que irá abrir a tela da Figura 11. Nesta tela deve-se pressionar o botão para “Importar Arquivos”, conforme Figura 10 para proceder a importação dos arquivos textos gerados pelo sistema da tinturaria. Alguns arquivos não são obrigatórios, como os arquivos de restrição de materiais, cores e pesos. Os demais são obrigatórios pois referem-se ao arquivo principal, que é o arquivo das OTs.

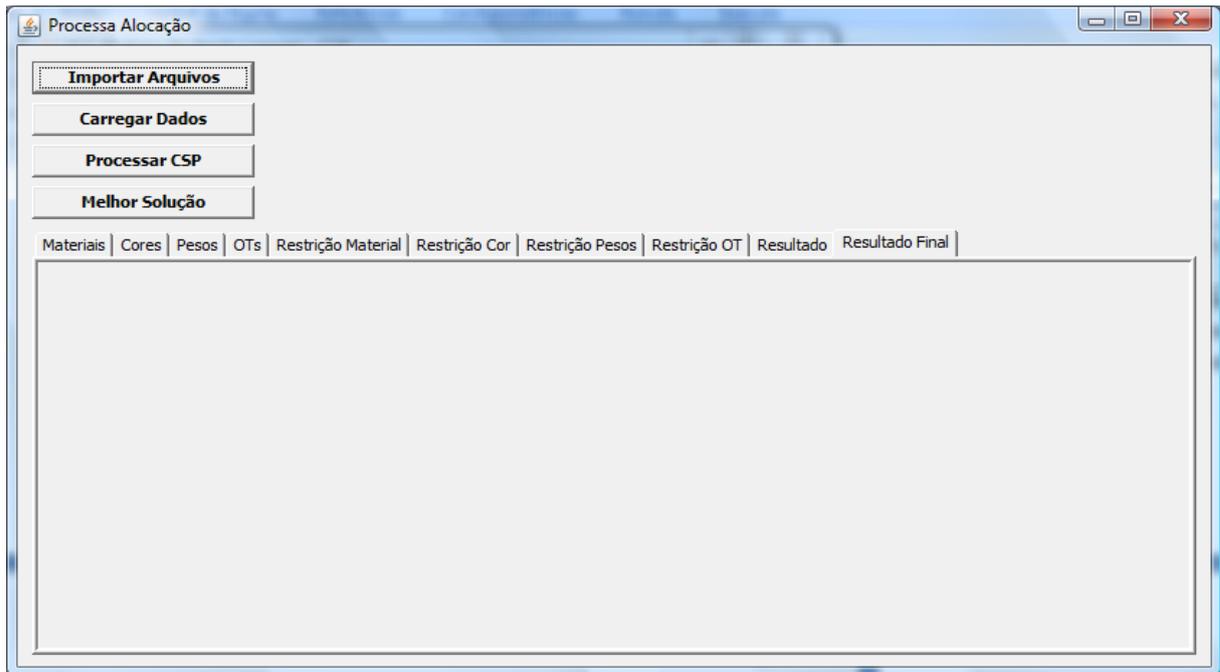


Figura 11- Tela de processamento da alocação

Após o processo de importação dos arquivos, é necessário carregar as listas e as tabelas para visualizar o conteúdo dos arquivos. Na Figura 12 é demonstrada a lista das OTs que foram carregadas e observa-se que existem registros com custo 9999. Durante o desenvolvimento do sistema, observou-se que um dos requisitos para que a solução fosse encontrada seria que todas as variáveis do problema estivessem preenchidas, ou seja, mesmo que não houvesse uma solução de alocação das OTs para todas as máquinas, as que o sistema não conseguisse alocar deveriam ser preenchidas com uma OT “nula”. Para cada máquina, foi criada uma OT com o custo 9999 e ao final da lista de materiais, cores e pesos também foram criados registros para indicar o registro ‘nulo’, conforme exemplo no Quadro 22.

Tempo de processamento: 14 segundos

id	OT	idMaterial	idCor	idPeso	nmCusto
1	011715	1	1	1	1
2	011717	1	1	2	2
3	011718	2	2	2	3
4	011719	3	1	1	4
5	011720	4	1	1	5
6	999901	5	3	3	9999
7	999902	5	3	3	9999
8	999903	5	3	3	9999

Figura 12 - Lista das OTs que foram importadas do arquivo texto

5.3.2.4 Processar o CSP

Ao pressionar o botão para processar o CSP, o sistema irá passar o objeto `Importacao` contendo as listas para o objeto `Tinturaria`, que irá processar o algoritmo CSP através da biblioteca CHOCO. No Quadro 23 é criado o problema e definido um array para cada variável, pois são criadas as mesmas variáveis para todas as máquinas. Após a criação das variáveis são atribuídos os seus domínios, que são iguais para todas as máquinas. Observa-se que o segundo e o terceiro parâmetro no momento da criação do domínio são os intervalos de valores que serão utilizados na criação da árvore de soluções do problema. Por este motivo os arquivos textos possuem um número sequencial no primeiro campo de cada lista. No exemplo do arquivo de materiais, a variável `MATERIAL` inicia no número 1 e vai até o número 5.

```

// Cria o problema a ser resolvido
Problem pb = new Problem();

// Cria as variáveis
IntDomainVar OT[] = new IntDomainVar[maquinas+1];
IntDomainVar COR[] = new IntDomainVar[maquinas+1];
IntDomainVar MATERIAL[] = new IntDomainVar[maquinas+1];
IntDomainVar PESO[] = new IntDomainVar[maquinas+1];

// Cria os domínios das variáveis
for (int i = 1; i <= maquinas; i++) {
    // Altera para String para poder adicionar o zero a esquerda
    ordem = ""+i;
    if (ordem.length() == 1) ordem = "0"+ordem;
    OT[i] = pb.makeEnumIntVar("TM"+ordem+"_OT", 1,
        imp.getListaOts().size());
    COR[i] = pb.makeEnumIntVar("TM"+ordem+"_COR", 1,
        imp.getListaCores().size());
    MATERIAL[i] = pb.makeEnumIntVar("TM"+ordem+"_MATERIAL", 1,
        imp.getListaMateriais().size());
    PESO[i] = pb.makeEnumIntVar("TM"+ordem+"_PESO", 1,
        imp.getListaPesos().size());
}

```

Quadro 23 - Criação do problema e das variáveis utilizando a biblioteca CHOCO

Para criar as restrições de cada variável, foram analisadas as listas de restrições e em alguns casos, as particularidades das restrições obrigaram uma mudança na lógica. No Quadro 22, ao se analisar o arquivo texto das restrições de cores, observa-se que o campo *Status* indica se a cor é aceita (A) ou rejeitada (R) na máquina. Se na máquina 1 a cor branca for aceita e a preta também, foi criada uma restrição utilizando o operador lógico OU, indicando que na máquina 1 a cor é igual a branca OU igual a preta. Para o caso de indicar que a cor branca é rejeitada e a cor preta também, foi utilizado o operador lógico E, ficando então a regra indicando que na máquina 1 a cor não pode ser igual a branca E preta.

No Quadro 24 pode-se observar que no início são criadas as listas para armazenar as restrições do tipo OU e E. Durante o processamento da lista das restrições das cores, é observado o campo *status*, para verificar se aceita ou rejeita. Se o *status* indicar que aceita a cor, é adicionado na lista OU, caso contrário na lista E. Ao final da verificação de todas as restrições das cores, as listas que estiverem preenchidas são enviadas para o problema, utilizando o método *post*. Apenas o arquivo de restrições de cores e materiais que possuem a lista separadas por OU e E. No caso das restrições de peso, é utilizado apenas o operador lógico E, pois cada máquina tem o peso máximo e mínimo definido apenas uma única vez.

```

if (imp.getRestricaoCores().size() > 0) {
    // Array para conter as restrições e depois
    // montar um OR/AND entre as restrições da lista
    Constraint[] arrayRestCoresOR =
        new Constraint[imp.getStatusRestricao("A", "cores")];
    Constraint [] arrayRestCoresAND =
        new Constraint[imp.getStatusRestricao("R", "cores")];
    nContadorOR = 0;
    nContado = 0;
    for (int i = 0; i < imp.getRestricaoCores().size(); i++) {
        String dsAcao = imp.getRestricaoCores().get(i).dsAcao;
        Integer idMaquina = imp.getRestricaoCores().get(i).idMaquina;
        Integer idCor = imp.getRestricaoCores().get(i).idCor;
        // Caso a ação for Aceita
        if (dsAcao.equalsIgnoreCase("A")) {
            arrayRestCoresOR[nContadorOR++] =
                pb.eq(COR[idMaquina], idCor);
        // Caso a ação for Rejeita
        } else {
            arrayRestCoresAND[nContadorAND++] =
                pb.neq(COR[idMaquina], idCor);
        }
    }
    // Cria a lista de ORs e ANDs das restrições das cores
    if (arrayRestCoresOR.length > 0)
        pb.post(pb.or(arrayRestCoresOR));
    if (arrayRestCoresAND.length > 0)
        pb.post(pb.and(arrayRestCoresAND));
}

```

Quadro 24 - Criação da restrição para as cores

Como o algoritmo testa todos os valores que estão no domínio de cada variável, existem momentos que no preenchimento de todas as variáveis, será testado um número de OT com um material que não está vinculado a ele. Neste caso, a restrição torna-se mais complexa que as restrições anteriores e é necessário utilizar a expressão IFTHEN. No Quadro 25, observa-se na expressão que o teste indica que, ao preencher a variável OT, a biblioteca CHOCO deverá testar a COR, MATERIAL e PESO para ver se realmente o resultado é da OT.

```

if (imp.getListaOts().size() > 0) {
    for (int i = 0; i < imp.getListaOts().size(); i++) {
        for (int j = 1; j <= maquinas; j++) {
            pb.post(pb.ifThen(
                pb.eq(OT[j], imp.getListaOts().get(i).idOt),
                pb.and(
                    pb.eq(COR[j], imp.getListaOts().get(i).idCor),
                    pb.eq(MATERIAL[j], imp.getListaOts().get(i).idMaterial),
                    pb.eq(PESO[j], imp.getListaOts().get(i).idPeso)));
            }
        }
    }
}

```

Quadro 25 - Restrições das OTs

Para evitar que o algoritmo encontrasse uma solução que pudesse ter a mesma OT alocada em mais de uma máquina, foi criada uma restrição que indicasse que para cada máquina não pode ter uma OT equivalente em outra máquina, conforme demonstrado no

Quadro 26.

```

for (int i = 1; i <= maquinas; i++) {
    for (int j = 1; j <= maquinas; j++) {
        if (i != j) pb.post(pb.neq(OT[i], OT[j]));
    }
}

```

Quadro 26 - Restrição das OTs que não podem estar em mais de uma máquina

O próximo passo após as restrições terem sido importadas é a execução do algoritmo de CSP pela biblioteca CHOCO, que é feito pelo método `getSolver`. Para indicar se deseja retornar a primeira ou todas as soluções, é utilizado o método `solve`. No desenvolvimento do sistema foi utilizada a opção como `true`, pois existia a necessidade de retornar todas as soluções possíveis para posteriormente analisar qual seria a melhor, através do custo da OT.

No Quadro 27, demonstra-se a execução do algoritmo para resolver o problema e passagem do parâmetro para o `solver` indicando para retornar todas as soluções e é guardado o horário inicial e final do processo para calcular o tempo de processamento do algoritmo. Após o processamento são armazenadas na `listaSolucoes` todas as soluções encontradas pelo algoritmo, para posteriormente analisar através do custo de cada solução qual será a seqüência com o menor custo.

```

//Inicializa o solver, quer irá resolver o problema
solver = pb.getSolver();
// true retorna todas as soluções. false retorna apenas a primeira
pb.solve(true);
// Guarda todas as soluções das máquinas
Solucao sol;
// Guarda o tempo de processamento
tempoProcessamento =
    "Tempo de processamento: "+ ((System.currentTimeMillis() - inicio)/1000);
// Cria uma lista das soluções para poder manusear posteriormente
for (int i = 0; i < solver.getSearchSolver().solutions.size(); i++) {
    solution = (Solution) solver.getSearchSolver().solutions.get(i);
    // Guarda a solução de cada máquina
    SolucaoMaquina solMaq;
    // Guarda as soluções das máquinas
    sol = new Solucao();
    for (int j = 1; j <= maquinas; j++) {
        solMaq = new SolucaoMaquina();
        solMaq.setIdMaquina(j);
        solMaq.setIdOt(solution.getValue(pb.getIntVarIndex(OT[j])));
        solMaq.setIdMaterial(
            solution.getValue(pb.getIntVarIndex(MATERIAL[j])));
        solMaq.setIdCor(solution.getValue(pb.getIntVarIndex(COR[j])));
        solMaq.setIdPeso(
            solution.getValue(pb.getIntVarIndex(PESO[j])));
        solMaq.setNmCusto(getCustoOT(solMaq.getIdOt()));
        sol.addSolucaoMaquina(solMaq);
    }
    // Adiciona a solução na lista
    this.listaSolucoes.add(sol);
}

```

Quadro 27 - Execução do algoritmo CSP e retorno da solução

5.3.2.5 Encontrar a melhor solução

Terminando a execução do algoritmo e tendo a lista das soluções encontradas, o próximo passo é a execução do método `melhorSolucao` que cria uma nova lista a partir da lista de soluções encontradas. Nesta nova lista, são analisadas todas as soluções, e colocadas numa ordem baseada no custo da solução. Se numa determinada lista de soluções, a primeira solução tem a máquina 1 com a OT com custo 2 e na máquina 2 a OT com custo 5, o custo total desta solução será 7. Se na lista de soluções não tiver outra solução com custo menor, esta será considerada a solução com o menor custo. Em seguida é verificado todas as soluções da lista que possuem as OTs que estavam na melhor solução e são removidas. Com a nova lista, volta-se a verificar a melhor solução até o momento que todas as OTs estejam alocadas.

O método `melhorSolucao` é demonstrado no

```
// Retorna a melhor solução das máquinas,
// baseado no custo de cada OT
private void melhorSolucao () {
    // Guarda uma cópia da lista de soluções
    // original para ir eliminando
    // a cada solução considerada melhor
    for (int i = 0; i < this.listaSolucao.size(); i++) {
        this.listaSolucaoTemporaria.add(this.listaSolucao.get(i));
    }
    // Cria uma nova lista com as melhores soluções
    while (this.listaSolucaoTemporaria.size() > 0) {
        melhorSolucao =
            tint.getSolucaoValida(this.listaSolucaoTemporaria);
        if (melhorSolucao != null) {
            listaSolucaoFinal.add(melhorSolucao);
            this.listaSolucaoTemporaria =
                criaNovaListaTemporaria(melhorSolucao);
        }
    }
    // Atualiza a tabela do resultado final
    atualizaResultadoFinal();
}
```

Quadro 28 - Cálculo da melhor solução

5.3.2.6 Resultado da execução do algoritmo CSP na base de dados

Após o processamento do algoritmo e a busca pelo melhor custo, pode ser visualizado a alocação das OTs nas máquinas conforme a Figura 13. No exemplo com 5 OTs e 3 máquinas, não havia OTs suficientes para completar as máquinas e por este motivo, a máquina 1 recebeu a OT com o número 6, indicando que não tem uma segunda OT para atingir. Nas outras máquinas, observa-se que cada uma tem duas OTs para executar. A

máquina 2 terá as OTs 2 e 5 enquanto a máquina 3 terá as OTs 1 e 4. O tempo total de processamento deste exemplo ficou em aproximadamente 14 segundos.

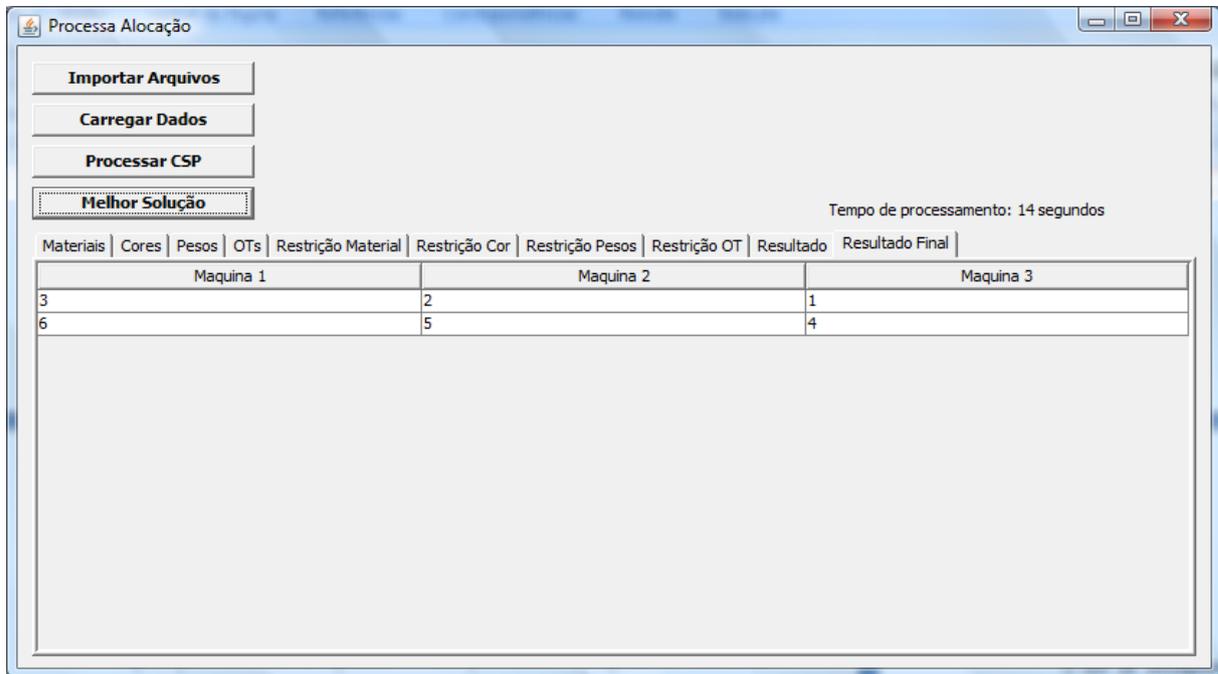


Figura 13 - Resultado do algoritmo e a melhor solução encontrada

No anexo B é demonstrado a tela do sistema da tinturaria, que lista as OTs por máquina e a sua seqüência.

5.4 RESULTADOS E DISCUSSÃO

Este trabalho partiu da dificuldade existente na área de tinturaria em analisar todas as restrições existentes no processo têxtil e na possibilidade de empregar as técnicas de CSP para facilitar a alocação das máquinas.

Durante o desenvolvimento do trabalho, foram encontradas várias bibliotecas para o processamento de algoritmos de CSP. Devido à grande quantidade de métodos disponíveis e por ser *OpenSource*, foi escolhida a biblioteca CHOCO. Inicialmente foram feitos testes com esta biblioteca e foi encontrada uma limitação no número de soluções retornadas, que estava limitada dentro dos fontes da biblioteca em apenas cinco soluções. Após a análise da documentação e alteração no parâmetro, a biblioteca foi novamente compilada e o número de soluções foi alterado para 2000.

Os resultados obtidos com o desenvolvimento deste trabalho se mostraram satisfatórios, visto que os objetivos foram alcançados. A idéia inicial de desenvolver um sistema que analisasse as regras de tinturaria foi alcançada e a facilidade na inclusão de novas restrições tornou o sistema de fácil configuração.

Quanto ao tempo de processamento, o que se pode observar foi o aumento do número de soluções a partir do momento que o número de máquinas foi aumentado. O tempo de processamento conseqüentemente aumentou, mas numa proporção maior que se inicialmente esperado, conforme o Quadro 29.

Máquinas	Ots	Soluções	Tempo aproximado em segundos
3	5	115	14
4	5	428	996

Quadro 29 - Tempo de resposta para encontrar todas as soluções

Quanto aos trabalhos correlatos, este se difere por tratar da área de tinturaria dentro de uma empresa têxtil. Quanto às semelhanças, podemos encontrar no trabalho de Kammer (KAMMER, 2006) o uso da biblioteca CHOCO.

Na Rovitex, atualmente a tinturaria conta com 34 máquinas de tingimento, aproximadamente 55 toneladas de malhas já alocadas nas máquinas que equivalem a 260 OTs em 63 cores diferentes. Sua capacidade atual permite o tingimento de 16 toneladas de malha tingida por dia, que suprem a sua necessidade interna para confecção.

Para poder modelar um problema real da tinturaria da Rovitex, deve-se considerar que na mesma OT podem existir mais de um material com fibras diferentes e que a capacidade das máquinas é variável pelo material que está sendo tingido na máquina, ou seja, se na máquina TM02 for tingir poliéster, a capacidade varia entre 135 e 157 quilos e no caso de tingir algodão, a capacidade varia entre 125 e 147 quilos.

6 CONCLUSÕES

Este trabalho propôs o uso de CSP para solução de problemas de escalonamento de tarefas em tinturaria. O uso da Inteligência Artificial (IA) em sistemas de alocação de recursos ainda é pouco utilizado e por este motivo tornou-se o grande motivador para o início deste projeto.

A técnica de CSP pode ser empregada em várias áreas de uma empresa têxtil, onde a variedade e a quantidade de restrições nos mostram um bom universo para ser explorado, desde a área produtiva até área administrativa.

O objetivo principal deste trabalho que era a alocação das OTs nas máquinas foi atingido e o uso da biblioteca CHOCO foi de extrema importância para o resultado alcançado. A facilidade na implementação com o uso desta biblioteca compensou a dificuldade inicial na definição da modelagem do problema. Inicialmente os processos de tingimento foram modelados como um CSP, as restrições levaram em consideração cores, peso de malha e tipo de malha. A programação de malhas é definida externamente pelo sistema ERP da empresa, é carregado através de arquivos texto. O sistema desenvolvido então resolve o CSP utilizando a biblioteca CHOCO e exibe textualmente o resultado de seqüenciamento de ordens de tingimento.

A principal vantagem encontrada neste trabalho foi o uso de inteligência artificial para resolver problemas comuns numa empresa têxtil.

Quanto à limitação encontrada na conclusão deste trabalho foi a utilização de um algoritmo simples para encontrara a “melhor” solução, o que poderá ser resolvido através da aplicação de algoritmos de otimização, chamados *Constraint Optimization Problem* (COP).

6.1 EXTENSÕES

Como sugestão para trabalhos futuros que abordem este tema, seria a utilização de CSP com otimização, chamado de *Constraint Optimization Problem* (COP) para otimização dos resultados da alocação, visando a melhoria do aproveitamento do tempo das máquinas e a melhora na velocidade de retorno das soluções encontradas pelo algoritmo.

A interface para visualização da alocação dos gráficos deveria mostrar as OTs em

seqüência de alocação, mostrando as cores, números de OT e pesos, conforme o anexo B. Isso iria facilitar a leitura da alocação e identificar possíveis problemas na ordem das cores.

Outra sugestão seria a aplicação da alocação com CSP em tempo real, prevendo assim as quebras de máquinas ou manutenções preventivas. Desta forma, poderiam ser realocadas automaticamente as OTs para outras máquinas.

REFERÊNCIAS BIBLIOGRÁFICAS

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. Tradução Fábio Freitas. Rio de Janeiro: Campos, 2002.

BRAILSFORD, Sally C.; POTTS, Chris N.; SMITH, Barbara M. Constraint satisfaction problems: algorithms and applications. **European Journal of Operational Research**, Southampton, v. 119, n. 3, p. 557-581, 16 dez. 1999.

BRUNS, Mauricio. **Aplicação da técnica de satisfação de restrições distribuídas no sincronismo de semáforos de uma malha viária**. 2005. 57 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

CHOCO. França, 2009. Disponível em: <<http://www.emn.fr/x-info/choco-solver/doku.php>> . Acesso em: 3 nov. 2009.

ENTERPRISE ARCHITECT. Austrália, 2009. Disponível em: <<http://www.sparxsystems.com.au/about.html>>. Acesso em: 2 nov. 2009.

KAMMER, Rodrigo. **Sistema de informação para logística de distribuição de mercadorias utilizando a técnica CSP**. 2006. 45 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

MARTINS, Petrônio G.; LAUGENI, Fernando P. **Administração da produção**. 2. ed. São Paulo: Saraiva, 2005. 562 p.

MONKS, Joseph G. **Administração da produção**. Tradução Lauro Santos Blandy. São Paulo: McGraw-Hill, 1987. 502 p. Tradução de Theory and problems of operations management.

NETBEANS. República Checa, 2009. Disponível em: <<http://www.netbeans.org/>>. Acesso em: 3 nov. 2009.

PIRES, Alexsandro S. **Implementação de um algoritmo heurístico para problemas de restrições**. 2006. 59 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

RUSSEL, Stuart J.; NORVIG, Peter. **Inteligência artificial**. Rio de Janeiro: Elsevier, 2004. 1021 p. Tradução da segunda edição.

SANTOS, Fernando dos. **Implementação distribuída do algoritmo ADOPT**. 2005. 115 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

TSANG, Edward. **Foundations of constraint satisfaction**. Colchester: Academic Press, 1993. 421 p.

ANEXO A – Ordem de Tingimento

Data	30/10/2009	ROVITEX IND. E COM. DE MALHAS LTDA.	Página	1
Hora	09:14:13	Ordem de Tinturaria	RTI36137	
OTN	452423	Grupo 452359/ 22	Usr. Impressão	ADMIN
Relação	TING FIO/FIXO	Gestão	17313/ 17186	Cadastro 21/10/2009
Receita	601611	GELO-CV	Tempo Tingimento	05:45:00
Cor	611	GELO	Máquina	TM17
Finalidade	PROGRAMAÇÃO SEMANA 48			
Tipo de Malha	06	VISCOSE/ELASTANO CONVECCIONAL	Entrega	06/11/2009
Mat. Origem	50599	Descrição	VISCOFLEX (32/28)	
Cor	025	MALHA CRUA		
Material	50503	Descrição	Gramatura	Qtde Bruta
		M/M VISCOSE FLEX ROTATIVA (505)- 1.80	215	198.00Kg
		ESTAMPADO 20985	LISTRAS 198 FUNDO 611	Qtde Liq 174.23Kg
				Largura 1.8000
Composição:		VISCOSE	96 %	
		ELASTANO	4 %	
TEAR	AREA DE PREPARACAO	DATA	TITULO	LOTE
FORNECEDOR				
Roteiro				
Material	50503	M/M VISCOSE FLEX ROTATIVA (505)- 1.80		
452	PROGRAMADO			
001	PESAR MALHA CRUA			
002	PREPARAR MALHA			
539	HIDRO-RELAXADEIRA			
005	TINGIR	Al. 09:03 Al. 09:00 Al. 08:59 Al. 08:58		
006	ABRIR			
498	RAMAR 1			
101	ESTAMPA ROTATIVA			
499	RAMAR			
600	LIBERACAO TINTURARIA			
004	RELAXADEIRA			
Artigo	Cor	Estampa	OTN	Peso
				Material
				Coleção
				Agrupador Cores
				Agrupador
		Peso Total	0,00	
		Produto		Cor

Quadro 30 - Ordem de tingimento

ANEXO B – Alocação das máquinas manualmente

Listar todas as Máquinas

Desalocar OTN
430729

Cor Seleccionada

OTN	Material	Tp.M	Qtde

Agrupar

Total

Máximo

Diferença

Máquina

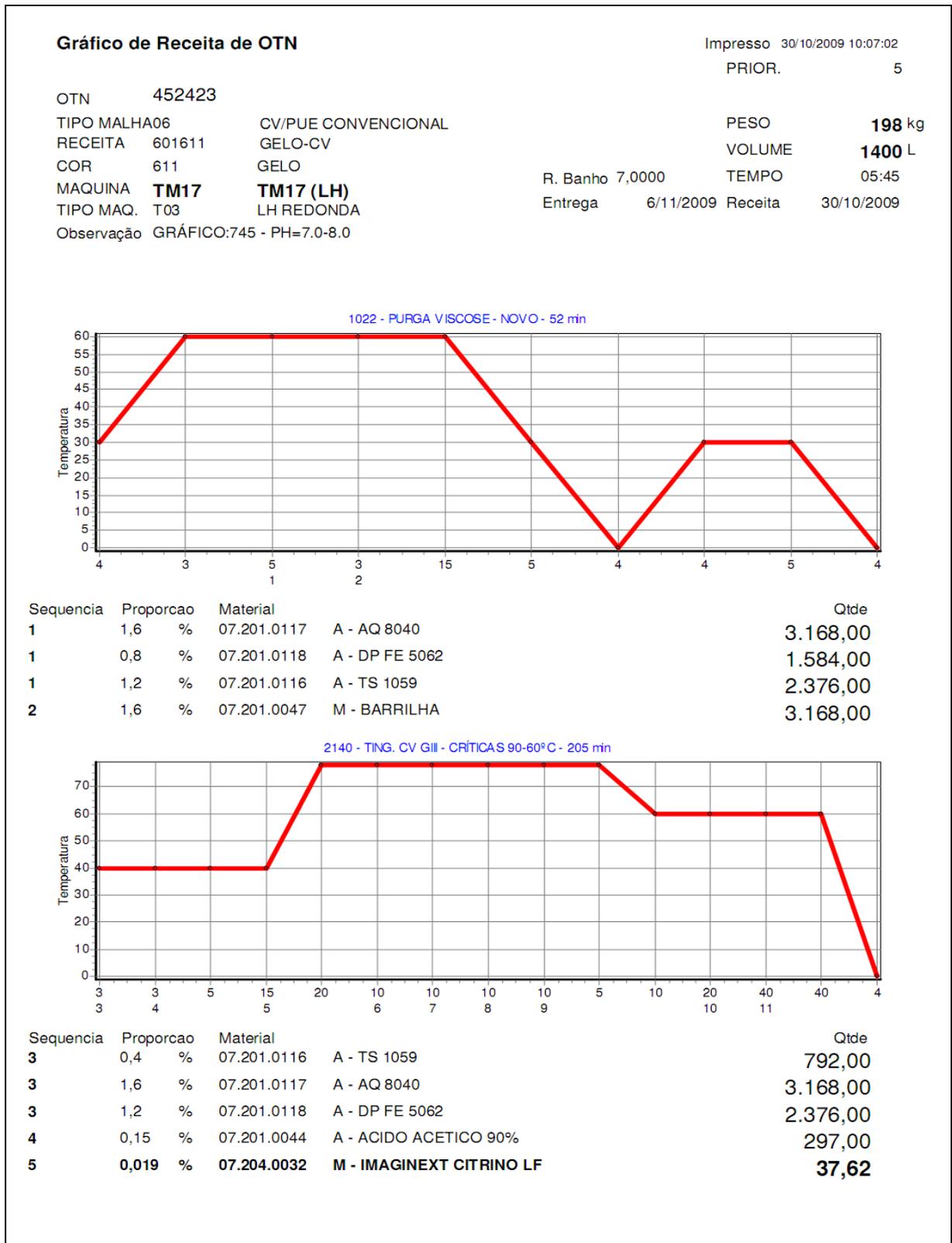
Material	Descrição	Qtde	Rolo	Propor

Seq	Agr	PCP	OTN	Pri	Cor	Peso	Período	Dt Entrega
1	33300	430729	5	663	90.00	43	2009	23/10/09
2	37172	421761	5	628	12.00	42	2009	20/10/09
3	37171	421788	5	628	80.00	42	2009	30/10/09
4	36833	441669	5	000	87.00	44	2009	30/10/09
5	37136	452578	5	052	90.00	45	2009	04/11/09
6	37137	452579	5	088	90.00	45	2009	04/11/09
7	36835	452432	5	068	90.00	45	2009	06/11/09
8	37241	452433	5	207	90.00	45	2009	06/11/09
9	37354	471564	5	600	90.00	47	2009	20/11/09
10	37351	471563	5	558	90.00	47	2009	20/11/09
11	37291	471570	5	168	90.00	47	2009	20/11/09
12	37322	471538	5	515	80.00	47	2009	20/11/09
13	37293	471571	5	181	94.00	47	2009	20/11/09
14	37356	471593	5	611	90.00	47	2009	20/11/09
15	37520	481421	5	011	80.00	48	2009	27/11/09
16	37551	481493	5	099	90.00	48	2009	27/11/09
17	37549	481487	5	098	90.00	48	2009	27/11/09
18	37589	481490	5	558	90.00	48	2009	27/11/09
19	37602	481466	5	628	90.00	48	2009	27/11/09
20	37605	481482	5	628	90.00	48	2009	27/11/09
21	37607	481491	5	628	90.00	48	2009	27/11/09
22	37543	481486	5	049	90.00	48	2009	27/11/09

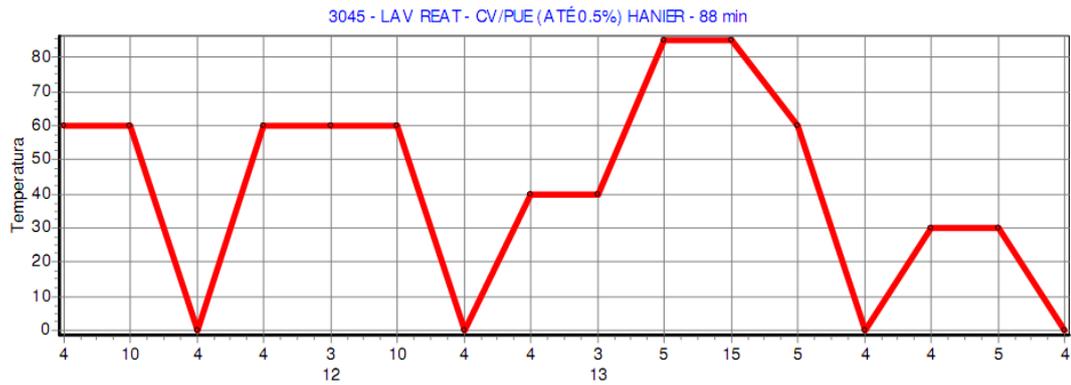
Máquina	Part.	Tempo	Mínimo	Máximo	Bocas
TF01 (FIO)	1	3:41			4
TF02 (FIO)	1	2:57			1
TL01 (LAB)	6	45:57			1
TM01 (TRD)	22	145:34			1
TM02 (TRD)	17	111:45			1
TM03 (TRD)	15	116:14			1
TM04 (TRD)	18	114:30			2
TM05 (TRD)	20	132:40			4
TM06 (MS)	15	93:36			1
TM07 (MS)	12	90:50			1
TM08 (MS)	14	97:16			1
TM09 (HT)	12	60:40			1
TM10 (HT)	13	95:42			1
TM11 (LH)	14	99:49			2
TM12 (LH)	19	98:41			2
TM13 (LH)	14	98:53			2
TM14 (EM)	17	108:6			2
TM15 (EM)	30	99:31			2
TM16 (EM)	36	100:51			1
TM17 (LH)	17	108:33			1
TM18 (LH)	17	115:28			1
TM19 (LH)	18	104:3			1
TM20 (LH)	16	111:4			1

Quadro 31 - Alocação de máquinas manualmente

ANEXO C – Gráfico de tingimento



5	0,0046	%	07.204.0033	M - IMAGINEXT RUBI	9,10
5	0,009	%	07.204.0034	M - IMAGINEXT TURMALINA	17,82
6	2	gl	07.201.0042	M - SAL REFINADO -600 PPM CACO3	2.772,00
7	4	gl	07.201.0042	M - SAL REFINADO -600 PPM CACO3	5.544,00
8	6	gl	07.201.0042	M - SAL REFINADO -600 PPM CACO3	8.316,00
9	8	gl	07.201.0042	M - SAL REFINADO -600 PPM CACO3	11.088,00
10	0,5	gl	07.201.0047	M - BARRILHA	693,00
11	4,5	gl	07.201.0047	M - BARRILHA	6.237,00



Sequencia	Proporcao		Material		Qtde
12	1	%	07.201.0044	A - ACIDO ACETICO 90%	1.980,00
13	1,6	%	07.201.0117	A - AQ 8040	3.168,00
13	0,5	%	07.201.0119	A - LR 3063	990,00

Figura 14- Gráfico de tingimento