

**UNIVERSIDADE DO VALE DO ITAJAÍ**  
**PRÓ-REITORIA DE PÓS-GRADUAÇÃO, PESQUISA, EXTENSÃO E CULTURA**  
**PROGRAMA DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA**

**DISTRIBUIÇÃO DE SERVIÇOS EM LABORATÓRIO DE  
METROLOGIA UTILIZANDO ALGORITMOS GENÉTICOS**

**Dissertação de Mestrado**

por

Jonny Ivon Beckert

Raimundo Celeste Ghizoni Teive, Dr.  
Orientador

São José (SC), fevereiro de 2009

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**UNIVERSIDADE DO VALE DO ITAJAÍ**  
**PRÓ-REITORIA DE PÓS-GRADUAÇÃO, PESQUISA, EXTENSÃO E CULTURA**  
**PROGRAMA DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA**

**DISTRIBUIÇÃO DE SERVIÇOS EM LABORATÓRIO DE  
METROLOGIA UTILIZANDO ALGORITMOS GENÉTICOS**

por

Jonny Ivon Beckert

Dissertação de mestrado apresentado como  
requisito parcial à obtenção do grau de Mestre  
em Computação Aplicada.

Orientador: Raimundo Celeste Ghizoni Teive, Dr.

São José (SC), fevereiro de 2009

## SUMÁRIO

LISTA DE FIGURAS .....	v
LISTA DE TABELAS .....	vi
LISTA DE ABREVIATURAS.....	viii
LISTA DE SÍMBOLOS.....	ix
RESUMO .....	x
ABSTRACT .....	xi
<b>1 INTRODUÇÃO .....</b>	<b>1</b>
<b>1.1 O PROBLEMA DE DISTRIBUIÇÃO DE SERVIÇOS METROLÓGICOS   APLICADOS A UM LABORATÓRIO PRESTADOR DE SERVIÇOS DE   METROLOGIA.....</b>	<b>1</b>
<b>1.2 OBJETIVOS .....</b>	<b>8</b>
<b>1.1.1 OBJETIVO GERAL.....</b>	<b>8</b>
<b>1.1.2 OBJETIVOS ESPECÍFICOS .....</b>	<b>8</b>
<b>1.3 MOTIVAÇÃO.....</b>	<b>9</b>
<b>1.4 JUSTIFICATIVA .....</b>	<b>9</b>
<b>1.5 ESTRUTURA DO TRABALHO.....</b>	<b>10</b>
<b>2 Fundamentação teórica.....</b>	<b>11</b>
<b>2.1 O PROBLEMA DE <i>JOB SCHEDULING</i>.....</b>	<b>11</b>
<b>2.1.1 JOB SHEDULING, MÉTODOS UTILIZADOS .....</b>	<b>11</b>
<b>2.1.2 JOB SHEDULING, UTILIZANDO ALGORITMO GENÉTICO.....</b>	<b>15</b>
<b>2.2 ALGORITMOS GENÉTICOS - CONCEITOS PRINCIPAIS .....</b>	<b>17</b>
<b>2.3 CONSIDERAÇÕES FINAIS .....</b>	<b>23</b>
<b>3 DESENVOLVIMENTO .....</b>	<b>25</b>
<b>3.1 METODOLOGIA E PLANO DO TRABALHO.....</b>	<b>25</b>
<b>3.2 MODELO DE ALGORITMO GENÉTICO.....</b>	<b>27</b>
<b>3.2.1 ESQUELETO DO PROGRAMA .....</b>	<b>27</b>
<b>3.2.2 DEFININDO A FUNÇÃO DE AVALIAÇÃO .....</b>	<b>31</b>
<b>3.2.3 DEFININDO OS PARÂMETROS DO ALGORITMO GENÉTICO.....</b>	<b>33</b>
<b>3.2.4 TRATANDO ITENS CLONES.....</b>	<b>35</b>
<b>3.2.5 INCORPORANDO DADOS REAIS .....</b>	<b>37</b>
<b>3.2.6 INCORPORANDO ALGORITMO DE APRENDIZAGEM .....</b>	<b>45</b>
<b>3.3 RESUMO E COMENTÁRIOS .....</b>	<b>48</b>
<b>3.4 SEQUÊNCIA DE AÇÕES DO ALGORITMO GENÉTICO .....</b>	<b>49</b>
<b>4 TESTES E RESULTADOS .....</b>	<b>52</b>
<b>4.1 METODOLOGIA USADA PARA OS TESTES .....</b>	<b>52</b>
<b>4.2 RESULTADOS .....</b>	<b>53</b>

<b>4.3</b>	<b>VALIDAÇÃO.....</b>	<b>63</b>
<b>4.3.1</b>	<b>COMPARATIVO .....</b>	<b>63</b>
<b>4.3.2</b>	<b>INTERPRETAÇÃO .....</b>	<b>64</b>
<b>5</b>	<b>CONSIDERAÇÕES FINAIS .....</b>	<b>66</b>
<b>5.1</b>	<b>EXPECTATIVAS .....</b>	<b>66</b>
<b>5.2</b>	<b>CONCLUSÃO.....</b>	<b>67</b>
<b>5.3</b>	<b>SUGESTÃO PARA TRABALHOS FUTUROS.....</b>	<b>68</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>70</b>
	<b>Apendice A.....</b>	<b>74</b>
	<b>Apendice B.....</b>	<b>77</b>
	<b>Apendice C.....</b>	<b>80</b>
	<b>Apêndice D.....</b>	<b>81</b>

## LISTA DE FIGURAS

Modelo conceitual simplificado da problemática de distribuição de serviços.....	5
Tipos de paquímetros.....	6
Imagem de alguns tipos serviços executados por recursos diferentes.....	6
Diagrama posicionamento dos Algoritmos Genéticos como Técnica de Busca.....	19
Algoritmo genético básico.....	20
Cruzamento uniforme.....	23
Cruzamento de um ponto.....	23
Lista de recursos para testes iniciais.....	28
Lista de itens recebidos para testes iniciais.....	28
Modelo de cromossomo adotado.....	29
Representação vazia do cromossomo.....	30
Representação cromossomo preenchido.....	30
Representação cromossomo preenchido com clones.....	30
Representação cromossomo preenchido com o item clone evidenciado.....	30
Tela de parâmetros do AG.....	34
Algoritmo genético melhorado.....	37
Recursos reais disponíveis e os tipos de serviço que executam.....	38
Itens recebidos e serviços que devem ser executados.....	39
Agendamento real realizado dos itens recebidos.....	40
Gráfico de distribuição dos itens por recurso.....	41
Gráfico de distribuição dos itens nos recursos.....	41
Configuração do AG (versão melhorada).....	42
Tela de execução e resultados do AG.....	43
Resultados interpretados da forma binária para legível humana.....	44
Gráfico da distribuição de toda a população do AG.....	45
Tela de execução do AG de aprendizagem.....	47
Sequência simplificada dos blocos de execução do algoritmo genético final.....	50
Interpretação dos objetos nos gráficos 30 a 34.....	59
Distribuição realizada pelo especialista.....	60
Distribuição do AG. Com agrupamento, sem clones.....	60
Distribuição do AG. Com agrupamento, com clones.....	61
Distribuição do AG. Sem agrupamento, sem clones.....	61
Distribuição do AG. Sem agrupamento, com clones.....	62
Gráficos de distribuição dos itens recebidos (sem AG e com AG).....	63
Gráficos de distribuição dos melhores resultados.....	64

## LISTA DE TABELAS

Tabela 1. Terminologia AG .....	20
Tabela 2. Principais tipos de representação do cromossomo .....	22
Tabela 3. Resultados dos testes aplicados a um AG simples .....	35
Tabela 4. Resultados dos testes aplicados a um AG melhorado .....	36
Tabela 5. Resultados dos testes usando agrupamento com clones .....	55
Tabela 6. Resultados dos testes usando agrupamento sem clones .....	56
Tabela 7. Resultados dos testes sem agrupamento com clones .....	57
Tabela 8. Resultados dos testes sem agrupamento sem clones .....	58

## **LISTA DAS EQUAÇÕES**

Equação 1. Função de avaliação do Algoritmo Genético .....	32
--	----



## LISTA DE ABREVIATURAS

AG	Algoritmo Genético
ERP	<i>Enterprise Resource Planning</i>
INMETRO	Instituto Nacional de Metrologia, Normalização e Qualidade Industrial
SAP	Sistema de Gestão Empresarial
UNIVALI	Universidade do Vale do Itajaí
R1 a R9	Numero de itens alocados pelo especialista para o recurso de mesmo número
G1 a G9	Numero de itens alocados pelo algoritmo genético para o recurso de mesmo número
TR	Quantidade de itens agendados pelo especialista no dia
TG	Quantidade de itens agendados pelo algoritmo genético no dia
TSR	Tipo serviço recurso
TSI	Tipo serviço item
FMR	Faixa de trabalho do recurso
FMI	Faixa de medição do item
PE	Prazo de entrega
TMR	Total minutos recurso
TAR	Total acumulado recurso (minutos)
NIR	Número de itens recebidos
NRD	Número de recursos disponíveis
NIC	Número de itens no cromossomo
NC	Número de itens clones
TEI	Tempo de execução do serviço no item
i,r	Índices referentes ao item e recurso no cromossomo

## LISTA DE SÍMBOLOS

®	Marca Registrada
$\delta$	delta
$\mu$	mi
$\lambda$	lambda
°	grau

## RESUMO

BECKERT, Jonny Ivon. DISTRIBUIÇÃO DE SERVIÇOS EM LABORATÓRIO DE METROLOGIA UTILIZANDO ALGORITMOS GENÉTICOS. São José, 2009. Projeto de Dissertação (Mestrado em Computação Aplicada)–Programa de Mestrado Acadêmico em Computação Aplicada, Universidade do Vale do Itajaí, São José, 2009.

O problema de alocação de serviços em um conjunto limitado de máquinas, por mais simples que possa parecer, possui detalhes e restrições, que em um problema real, podem tornar a tarefa extremamente complexa. Prioridade do serviço, sobrecarga no recurso que irá executar o serviço, matéria-prima disponível para execução das tarefas, técnico habilitado, histórico de paradas do recurso, tempo de execução do serviço, tempo ocioso e disponibilidade de pessoal especializado são alguns aspectos a serem considerados, quando se tem a necessidade de realizar a alocação de uma tarefa dentro de uma grade de serviços pré-definidos. Atualmente, existem no mercado alguns softwares de gerenciamento, utilizados principalmente pelo ramo industrial, que se propõem a minimizar os problemas associados à distribuição de serviços.

Geralmente os sistemas integrados de gestão, conhecidos como ERP (*Enterprise Resource Planning*), possuem um módulo responsável por esta tarefa. Muitas vezes, as rotinas que envolvem a tarefa de distribuição de serviço neste módulo não levam em conta todas as variáveis envolvidas, devido a grande dificuldade de inserir estas variáveis no algoritmo e encontrar uma solução de otimização da distribuição. Além disto, quando a capacidade da máquina é variável, a complexidade do problema aumenta, dificultando assim a utilização de metodologias tradicionais baseadas em grafos ou programação matemática. Técnicas da inteligência artificial, particularmente os algoritmos genéticos têm sido aplicados ao problema de alocação de serviços (*job scheduling*) com grande sucesso, conforme se pode observar na literatura.

O objetivo principal deste trabalho é o desenvolvimento de um sistema computacional, baseado em Algoritmo Genético, para resolver o problema da alocação ótima de serviços em laboratório de metrologia, considerando as restrições técnicas e temporais do problema.

A modelagem do problema de alocação otimizada de serviços um número limitado de recursos com Algoritmos Genéticos, permitiu a consideração de aspectos importantes da solução do problema real, tais como: tipo de serviço do recurso e do item, faixa de trabalho do recurso, faixa de medição do item, prazo de entrega, quantidade de itens e recursos disponíveis, tempo de execução do serviço e do item. Além disto, pode-se enfatizar como contribuições adicionais do algoritmo implementado como a questão do tratamento de itens clones e o desenvolvimento de um Algoritmo Genético de Aprendizagem, o qual visa possibilitar a definição automática dos parâmetros do Algoritmo Genético principal, envolvendo tipo de crossover, tipo de seleção, valores adequados para as taxas de mutação, elitismo e torneio. Resultados preliminares obtidos com o algoritmo desenvolvido, utilizando-se dados reais, demonstram que a abordagem proposta neste trabalho é uma solução promissora e aplicável na prática.

Palavras-chave: Algoritmo genético. Metrologia. Ordem de serviço..

## ABSTRACT

BECKERT, Jonny Ivon. *SERVICES ALLOCATION IN A METROLOGY LABORATORY USING GENETIC ALGORITHMS*. São José, 2009. Dissertation Project (Applied Computing Mastership)-Academic Mastership Program in Applied Computing Sciences, Universidad do Vale do Itajaí, São José, 2009.

*As simple as it might look, the problem of allocating a group of tasks to be executed with a limited number of available machines will always present a number of details and restrictions that in certain cases can result in a solution of great complexity. Some of the variables involved are: service priority, available raw materials, machinery or equipment capacities, qualified technical personnel existence, production historical records, service execution time, machinery idle times, additional services that could call for specialized personnel.*

*Currently, it's easy to find some software solutions which provide this type of logistics to the industrial sector. The main goal of these solutions is to minimize the problems created when dealing with services distribution.*

*Generally, integrated management systems, known as ERP (Enterprise Resource Planning) are provided with an entire module to deal with services distribution; unfortunately it's not uncommon for this module to discard some or many of the factors involved. The specificities of each different case and the complexity to introduce these variables into the algorithm flux have a crucial role to play when choosing the best solution; the best alternative will depend heavily on its flexibility and versatility.*

*When a machine has variable capacity, as it is in many cases, the problem's complexity greatly increases and the traditional solution techniques like mathematic programming fall short of the solution. More successful results have been observed when using artificial intelligence techniques, particularly using genetic algorithms to solve the problem of services allocation (job scheduling) as there are written reports about it.*

*This work introduces the modeling and solution for cases of services allocation using metrology laboratories and the use of a genetic algorithm tailored specifically to each case. It takes into account, for each case, all the variables and restrictions that the specialized personnel will face in a metrology laboratory.*

*The solution modeling to optimize the services allocation among a limited number of resources using Genetic Algorithms, allows a better understanding about important aspects in real case situations, e.g. service type performed by the resource, item characteristics, resources' working range, item's metrology range, delivery times attainable, items and available resources quantities, services execution times. Worthwhile to mention and additional benefits obtained using the implemented algorithms are a better understanding of the problem when dealing with cloned items and the development of an associated Learning Genetic Algorithm with which a parameter definition automation for the main Genetic Algorithm Implementation is intended. These parameters are: crossover type, selection type, adequate mutation rate values, elitism and tournament. Preliminary results obtained using the proposed algorithm are showing the viability of the solution and it's applicability in practical cases.*

*Key words: Genetic Algorithm, Metrology, Service Order.*

# 1 INTRODUÇÃO

## 1.1 O PROBLEMA DE DISTRIBUIÇÃO DE SERVIÇOS METROLÓGICOS APLICADOS A UM LABORATÓRIO PRESTADOR DE SERVIÇOS DE METROLOGIA

O problema de distribuição de serviços (itens) em um conjunto de recursos (máquinas), o qual na literatura técnica é usualmente tratado como *job scheduling*, é um problema geral de otimização de recursos que existe em várias áreas, como por exemplo o caso de alocação de tarefas para um conjunto de processadores em um problema de processamento paralelo.

No problema foco deste trabalho, tem-se um laboratório prestador de serviços de calibração que tem como finalidade realizar atividades de medição de peças, calibração e manutenção de instrumentos, podendo estas peças e/ou instrumentos serem dos mais diversos tipos e grandezas. Paquímetros, micrômetros, calibradores, termopares, manômetros, voltímetros, são alguns exemplos destes instrumentos.

Dependendo do serviço a ser realizado, os recursos utilizados e o controle das grandezas de influência podem ser diferentes para cada tipo de instrumento e de grandeza. Adicionalmente, dependendo da aplicação do instrumento no processo produtivo, os critérios para realização do serviço no laboratório de metrologia também podem variar. Por exemplo: Um paquímetro utilizado no recebimento de materiais para medir diâmetro dos parafusos recebidos, possui critérios de aceitação diferentes de um paquímetro usado na ferramentaria, onde as medições são realizadas em moldes e em outros ferramentais. Ou seja, os procedimentos de calibração podem variar de acordo com a aplicação do instrumento no processo produtivo.

Por isso é fundamental conhecer qual é a real necessidade do cliente. Inicialmente precisa ser definido o tipo de serviço a ser realizado: verificação, calibração, manutenção de um instrumento ou então a medição de uma peça, como por exemplo, o controle dimensional das características de uma carcaça de motor. Na seqüência, deve ser estabelecido o procedimento de medição ou calibração a ser adotado.

Além de procedimentos distintos, o cliente ainda pode optar por serviços realizados parcialmente. Ou seja, no caso do paquímetro, o cliente pode solicitar uma calibração completa, ou então somente um dos seus componentes (os bicos, por exemplo). Outra variável existente nos laboratórios de calibração é se o serviço a ser executado é acreditado ou apenas rastreado. Por serviço acreditado, entende-se que o laboratório pertence a uma rede nacional de calibração, e que o referido serviço consta em uma lista autorizada pelo organismo acreditador. No Brasil, este processo de acreditação é de responsabilidade do INMETRO (Instituto Nacional de Metrologia, Normalização e Qualidade Industrial).

De acordo com o procedimento selecionado, um mesmo instrumento pode requerer recursos distintos para a realização do serviço. Ainda para a alocação dos serviços, é necessário ter estabelecido o prazo de entrega e a prioridade para o serviço solicitado. Com isso, tem-se a maioria das variáveis de entrada referentes ao serviço a ser realizado. Também devem ser verificadas as variáveis locais. Conforme o tipo de serviço, tipo de item (peça ou instrumento), tipo de grandeza e o procedimento adotado para execução do serviço, deve ser verificado se o laboratório possui os recursos necessários para sua execução, tais como:

- se o técnico que fará o serviço está habilitado tecnicamente para executar o procedimento do serviço;
- se o recurso onde será executado o serviço comporta o tipo de item e sua grandeza;
- se existe o material de expediente para executar o serviço.

Quando o laboratório possui os recursos de acordo com as necessidades, então o serviço parte para a etapa do agendamento, caso contrário o item é enviado a um laboratório terceirizado. Na aplicação descrita, é caracterizado somente um serviço, porém pode ocorrer que um instrumento seja sujeito a mais serviços, tais como manutenção e calibração. Um instrumento recebido para manutenção pode requerer uma calibração posterior. Cada item tem prioridade e prazo de entrega diferente, pois dependem do tempo de execução da tarefa e da urgência do cliente em receber os resultados. Os serviços em recursos que requerem tempos diferentes por procedimento possuem limitação em sua carga horária.

Alguns tipos de peças e instrumentos, dependendo da tarefa a ser executada, exigem que eles sejam climatizados a uma temperatura específica antes da medição ou da calibração, respectivamente.

A parada do recurso também deve ser levada em consideração, incluindo esta possibilidade na distribuição das tarefas. Esta parada pode ser devido a um histórico já observado sobre o recurso ou de uma parada programada para manutenção preventiva.

Um recurso (no caso de um laboratório de metrologia) pode ser tanto um dispositivo, uma máquina complexa ou uma pessoa que executa determinada tarefa. Cada recurso tem suas especificações e, conseqüentemente, também limitações. Em se tratando de um padrão de referência, a limitação pode estar na sua faixa de medição, na sua resolução e na sua incerteza de medição. Quando o recurso a ser utilizado é um indivíduo (técnico metrologista), este recurso deverá possuir capacitação para sua execução, o que leva a mais uma variável complicadora, a capacitação do recurso.

Na Figura 1, tem-se um mapa conceitual resumido, descrevendo a seqüência de ações necessárias para a distribuição de serviços em um laboratório prestador de serviços metrológicos. Ao receber um determinado item no setor de recebimento, é necessário saber o que se deseja realizar de serviço neste item. Uma vez determinado o tipo de serviço (medição, manutenção ou calibração...) é necessário agendar este serviço solicitado. Se o item recebido for uma peça (bloco de motor, carcaça, etc...) o serviço é determinado como uma medição (não se calibra um bloco de motor, apenas faz-se medição de características solicitadas). Quando o item recebido for um instrumento de medição, o serviço pode ser um ajuste, manutenção (por exemplo, reparar o cursor de um paquímetro que está travando), verificação ou ainda uma calibração. Cada família de instrumentos possui tipos de itens diferentes o que faz com que exista um procedimento específico para executar um determinado serviço conforme o tipo de item recebido.

Pode-se observar o exemplo de acordo com a Figura 1. No setor de recebimento, num suposto laboratório é dado entrada em um item para um determinado tipo de serviço à ser agendado. Supondo que seja um paquímetro. Primeira análise a ser feita é que tipo de paquímetro foi recebido. Para facilitar, iremos adotar um paquímetro convencional 0 a 25 mm, totalmente mecânico, sem nenhum recurso digital. Caso o cliente tenha solicitado um serviço de calibração, o item segue agora para o agendamento, onde alguns detalhes irão decidir este agendamento. A faixa de medição do item (grandeza), resolução e tipo de instrumento serão detalhes que direcionarão qual recurso executará o serviço no item. Se for acreditado ou rastreado implicará na forma da execução do serviço, e portanto, poderá influenciar no recurso que receberá o item, bem como, se o elemento humano (técnico executante) está apto a executar o procedimento. Pode suceder que o

procedimento tenha de ser executado conforme especificação de uma norma deste cliente. Neste nosso exemplo, vamos supor que tudo segue a padronização do laboratório que recebeu o item. Parte-se para a execução do serviço, nesse momento entra outro detalhe que é a prioridade ou prazo de entrega. Se todos os recursos estiverem ocupados, impreterivelmente ou o item fica na fila de espera ou o serviço é terceirizado para garantir o atendimento no prazo a este cliente. Agendado, o item é climatizado, executa-se o serviço, emite-se o certificado e devolve-se o item ao cliente. Seqüencialmente fica fácil de perceber os caminhos que segue um item porém se multiplicarmos estes detalhes por um número maior de itens e tipos diferentes, com especificações diferentes, procedimentos complexos, habilitações mais diversas dos técnicos executantes, teremos uma visão da complexidade destes agendamentos.



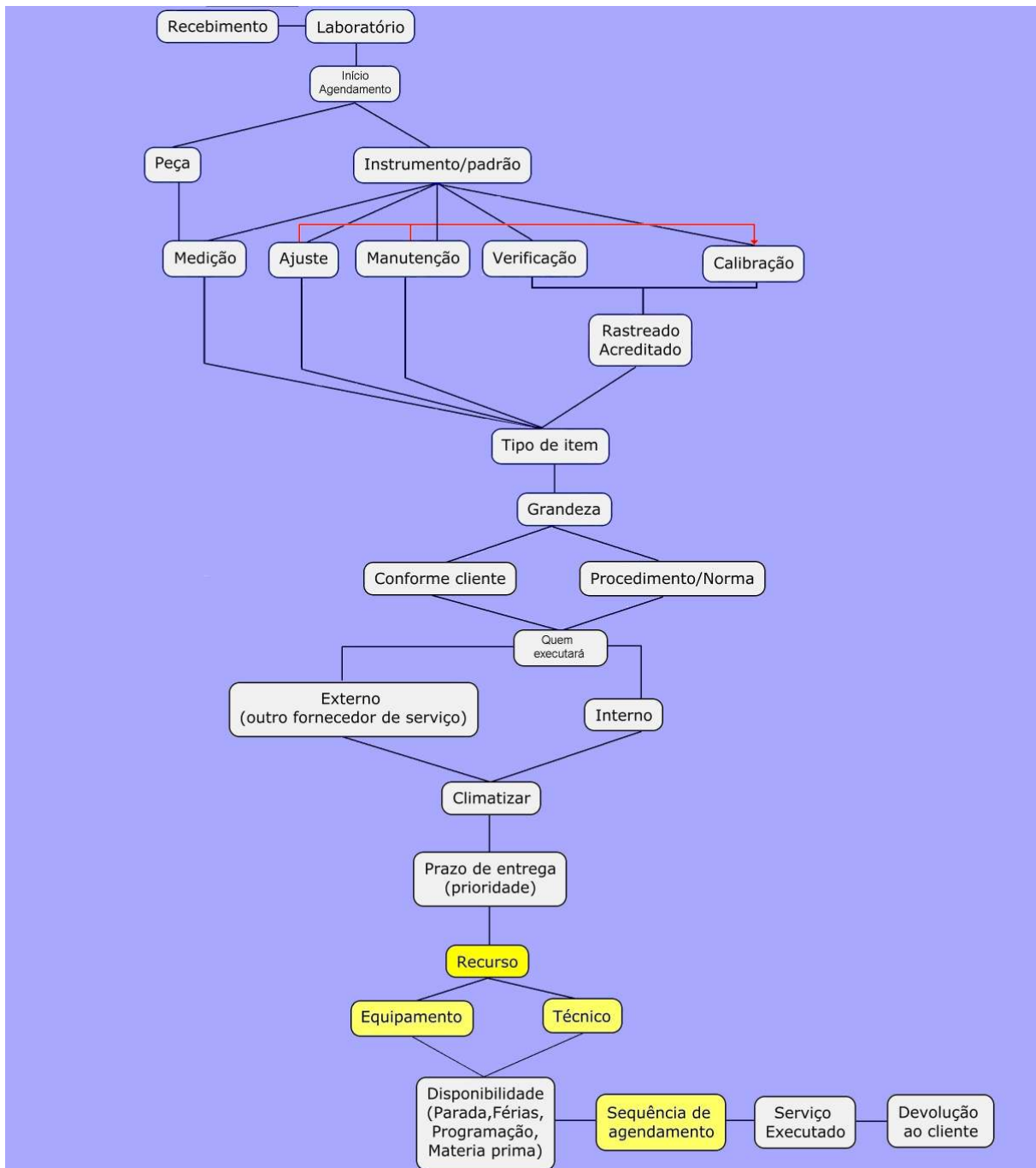


Figura 1. Modelo conceitual simplificado da problemática de distribuição de serviços

Para ilustrar melhor o problema em foco são disponibilizadas a seguir duas imagens, a Figura 2 apresenta alguns tipos de instrumentos de medição, porém todos da mesma família denominada paquímetro; observa-se assim que os itens podem ter detalhes totalmente diferentes,

mesmo pertencendo a uma mesma categoria de família e a Figura 3, apresenta alguns recursos diferentes executando serviços diferentes por técnicos especializados.

O que se quer ressaltar é que o tipo de serviço implicará num procedimento que depende do tipo de item recebido. Um paquímetro universal terá o serviço executado conforme um procedimento para este item, e será diferente de um micrômetro, uma régua, um manômetro ou qualquer outro tipo de instrumento. Isso denota tempos variados de execução.



Figura 2. Tipos de paquímetros



Figura 3. Imagem de alguns tipos serviços executados por recursos diferentes

Fonte: Laboratório de metrologia da Sociedade Educacional de SC

Uma vez definido o procedimento a ser adotado, o próximo passo é determinar quem executará o serviço. Se o laboratório prestador de serviço possuir condições técnicas e os dispositivos apropriados e tempo disponível, obviamente executará o serviço, caso contrário será

necessário enviar o item recebido a um laboratório externo, aguardar sua execução e retorno. Partindo-se do pressuposto que o laboratório irá executar o serviço, é adicionada ao tempo de execução uma etapa denominada de climatização, onde o item terá de aguardar a estabilização da temperatura, para que esteja dentro de um valor padrão determinado no procedimento correspondente. Concluída esta etapa, e sendo ela interna, tem-se que determinar quem executará o serviço e qual dispositivo ou recurso será utilizado. Por exemplo, qualquer técnico habilitado e treinado poderá executar o serviço de calibração de um paquímetro ou micrômetro. No entanto, quando um dispositivo especial requer um recurso específico, tal como uma máquina de medição tridimensional, então um agendamento do serviço deve ser feito para este recurso.

Adicionalmente temos uma variável que é a mais preocupante em toda esta problemática: o tempo de serviço definido pelo prazo de entrega. Um item classificado como urgente deve ter prioridade maior que um item que esta apenas obedecendo a um cronograma de serviços previstos. Após a realização dos serviços propriamente ditos, resta finalmente embalar o item e devolver ao solicitante..

Existem atualmente no mercado alguns softwares de gerenciamento, utilizados principalmente pelo ramo industrial, que se propõem a minimizar estes problemas associados com a distribuição de serviços. Geralmente são programas de ERP (*Enterprise Resource Planning*), onde existe um módulo responsável por esta tarefa. As rotinas que envolvem a tarefa de distribuição de serviço nestes módulos muitas vezes não levam em conta todas as variáveis envolvidas, devido a grande dificuldade de inserir estas variáveis no algoritmo e encontrar uma solução de otimização da distribuição.

O uso de métodos exatos, baseados em programação matemática, para resolução deste problema de alocação ótima de serviços ou de tarefas para um determinado processo ou máquina, na maioria das aplicações é inviável devido a impossibilidade de se formular uma função objetivo analítica e/ou da existência de restrições subjetivas, não matemáticas. É um problema *NP-HARD* de difícil utilização de métodos exatos. NP-hard é uma classe de complexidade que reúne problemas que não têm uma fórmula geral de solução. Por exemplo, uma conta de multiplicar não é NP-hard, porque existe um procedimento, o algoritmo da multiplicação, que se for aplicado corretamente gera o produto de dois números, não importa que números sejam esses. Já para encontrar a saída de um labirinto não existe nenhum algoritmo onde, digamos, informando-se o comprimento médio dos

corredores e o número esquinas à direita, obtém-se uma rota de saída. O único jeito de resolver um problema NP-hard é testar todas as soluções plausíveis, até que uma delas funcione.

Nestes casos, poderia ser verificada a viabilidade de uma modelagem baseada em algoritmos genéticos (AG). Como razões da utilização do AG citam-se:

- Enquadra-se no perfil do problema;
- Ser razoavelmente eficiente para a totalidade de problemas;
- Ter boa performance na apresentação solução viável do problema;
- Possuir amplo espaço de pesquisa;
- Ter sido usado com sucesso na resolução de problemas similares;
- Fácil alteração das regras que definem a solução do problema;
- Não precisa de modelo analítico nem de cálculo de derivadas;
- Ser extremamente simples de ser implementado.

## **1.2 OBJETIVOS**

Tendo explanado sobre o problema que é a distribuição de itens em um laboratório de metrologia, quer-se agora listar os objetivos desta dissertação, partindo do geral para o específico.

### **1.1.1 OBJETIVO GERAL**

O objetivo principal deste trabalho é o desenvolvimento de um sistema computacional, baseado em Algoritmo Genético, para resolver o problema da alocação ótima de serviços em laboratório de metrologia, considerando as restrições técnicas e temporais do problema.

### **1.1.2 OBJETIVOS ESPECÍFICOS**

Propõem se ainda como os objetivos específicos:

- Realizar uma revisão bibliográfica descrevendo o estado da arte na solução do problema geral de alocação ótima de serviços em um conjunto limitado de recursos/máquinas, para se obter o estado de arte da problemática;
- Modelar o problema de alocação ótima de serviços metrológicos, considerando todas as restrições intrínsecas do problema, através da modelagem de AG em linguagem de alto nível;
- Validar o software desenvolvido com dados reais da área de metrologia;
- Comparar desempenho do sistema implementado com sistemas convencionais, baseados na experiência humana e/ou ERP.

### **1.3 MOTIVAÇÃO**

Com certeza o elemento motivador principal que levou a elaboração deste trabalho foi o conhecimento da existência da problemática do problema de distribuição de serviço em um laboratório de metrologia e a necessidade de se minimizar este problema.

“Atacar” este problema implica em profundo conhecimento de toda uma sequência de operações que se inicia no recebimento do item até a entrega do item ao solicitante do serviço. Devido a complexidade e a quantidade de variáveis envolvidas, a procura de um método de resolução que atenda facilidade de manutenção do código fonte e performance satisfatória dos resultados obtidos, torna interessante a implementação com uso de técnicas de inteligência artificial, especificamente o uso de algoritmos genéticos.

O conhecimento adquirido na área de inteligência artificial, especificamente o tema de algoritmos genéticos, permitiu a associação da possibilidade de ter sucesso na tentativa da resolução da problemática adotando este enfoque.

### **1.4 JUSTIFICATIVA**

A necessidade dos laboratórios de metrologia executar os serviços diversos e aplica-los aos itens recebidos e assim atenderem todos os seus clientes; encontra alguns obstáculos já mencionados durante a introdução deste trabalho. Resumindo, temos a distribuição correta dos itens

de maneira a atender os prazos de entrega como o maior problema encontrado, e assim a justificativa para o desenvolvimento de uma metodologia computacional que venha a resolver este problema.

## **1.5 ESTRUTURA DO TRABALHO**

A composição deste trabalho, obedece uma sequência cronológica de acordo como as dificuldades foram avançando. Inicialmente é dada uma introdução a problemática de distribuição dos serviços para um laboratório de metrologia. No capítulo dois definimos os objetivos e esta estrutura, bem como a fundamentação teórica sobre o assunto e um pequeno estudo da arte do que encontramos de metodologias e artigos que tratam sobre este tema. Também se inclui uma pequena explicação sobre que é algoritmos genéticos.

Inicia-se o desenvolvimento propriamente dito a partir do capítulo três, onde se apresenta a metodologia empregada, a definição do cromossomo, a função *fitness* e as primeiras versões implementadas na tentativa de resolver a problemática. No mesmo capítulo, tenta-se apresentar as dificuldades encontradas no que levaram a evolução da versão do algoritmo, tratamento com dados reais, aparecimento de itens clones e culminando com a incorporação de um algoritmo de aprendizagem.

No quarto capítulo, tem-se os testes efetuados, a apresentação dos resultados e suas interpretações. Faz-se também a validação destes resultados e o comparativo entre a versão implementada utilizando o AG e a versão manual utilizando um técnico especialista.

Encerra-se o trabalho, apresentando no capítulo cinco as conclusões e sugestões para trabalhos futuros. E por fim, a bibliografia acessada no decorrer de todo este processo.

## 2 FUNDAMENTAÇÃO TEÓRICA

Para que se pudesse realizar este trabalho, foi necessário realizar uma extensa fundamentação teórica. Esta fundamentação foi iniciada apresentando o termo mais comumente utilizado que englobasse o problema estudo deste trabalho. Após verificou-se que metodologias estavam sendo utilizadas como tentativas de solução. Aprofundou-se na metodologia de melhor retorno no que se refere a dificuldade de implementação aliada a resultados obtidos e finalmente uma pequena conclusão sobre este capítulo.

### 2.1 O PROBLEMA DE *JOB SCHEDULING*

O termo mais próximo que podemos utilizar como sinônimo do tema desta dissertação é *Job Scheduling*, que simplificando de maneira extrema teríamos “*job*” como sinônimo de serviços e “*scheduler*” como sinônimo de agendamento. Juntando as duas palavras teremos o agendamento de serviços ou tarefas que são executadas seguindo regras específicas. Exemplificando: Serviços ou tarefas para preparar uma salada como “pegar uma fruta”, “lavar a fruta”, “descascar a fruta”, “cortar a fruta” etc... teriam que obrigatoriamente seguir uma pré-determinada seqüência, afinal não se pode descascar uma banana (por exemplo), antes de pegá-la e lavá-la.

#### 2.1.1 *JOB SCHEDULING*, MÉTODOS UTILIZADOS

Os problemas de escalonamento ou alocação de tarefas (*job scheduling*), são aqueles que “envolvem alocação de recursos no tempo com a finalidade de executar uma série de tarefas” (BAKER, 1974). Esses problemas são de difícil programação e têm sido um grande desafio computacional, não existindo ainda um algoritmo eficiente que garanta a obtenção da solução. O problema de alocação de recursos, tema desta dissertação, é considerado como um problema de otimização NP (não polinomial) difícil, sendo um dos mais difíceis problemas de otimização combinatória. Diversas metodologias foram desenvolvidas como soluções para o problema, entre as quais podemos citar:

- Métodos exatos - estes algoritmos escolhem a solução ótima através de regras aplicadas sobre o conjunto que contém todas as soluções. Esses algoritmos têm seu tempo computacional variando polinomialmente com o tamanho do problema;

- Algoritmo de Johnson (JOHNSON, 1954) - Determina a distância entre todos os pares de vértices de um grafo. O algoritmo de Johnson é um algoritmo de aproximação probabilístico para o problema da satisfatibilidade máxima para resolver o problema em que dois produtos devem ser processados por duas máquinas (2x2). Mas mesmo problemas 3x3, nx2 com não mais que três operações por produto e nx3 com não mais que 2 operações por produto são NP (não polinomial) difícil;
- *Branch and Bound* - as técnicas de Branch and Bound usam uma estrutura dinâmica de árvore em que nós folha representam as várias soluções possíveis. A busca começa de um nó raiz que é expandido aumentando a amplitude da busca. O próximo nó a ser expandido depende da técnica usada, sendo que a solução obtida é necessariamente inteira;
- Métodos de aproximação - os métodos de aproximação tentam diminuir o esforço computacional exigido pelo problema de alocação de tarefas, para isso sacrificam a solução ótima. Estes métodos garantem uma solução muito próxima da ótima (subótima), e que por isso pode substituí-la;
- Regras de priorização (*priority dispatch rules*) - o método atribui às soluções uma prioridade. É um método fácil de implementar e que exige poucos recursos computacionais, sendo bastante popular (BAKER, 1974; FRENCH, 1982; MORTON e PENTICO, 1993). Em cada passo as seqüências capazes de serem expandidas são classificadas através de sua prioridade. A mais prioritária é expandida. Vários passos são necessários para que uma solução válida seja alcançada;
- Heurísticas baseadas em pontos críticos (*bottleneck based Heuristics*) - o algoritmo de *Shifting bottleneck* (adiamento de ponto crítico) foi proposto por Adams, Balas e Zawack ( ADAMS, J., BALAS, E., ZAWACK, D., 1988) . Este método resolve um subproblema com uma máquina, sendo resolvido para cada máquina ainda não seqüenciada e o resultado é usado para encontrar um ponto crítico na máquina de maior *makespan*. (Cada nó representa uma operação, isto é, o processamento da tarefa *i* na máquina *k*). Ao nó é associado um peso. O *makespan* corresponde ao comprimento do maior caminho do nó). Toda vez que uma nova máquina é seqüenciada, toda a seqüência anterior deve ser submetida novamente à otimização. Alguns trabalhos são encontrados com alguma modificação na estrutura do algoritmo original, como a de Applegate e Cook (1991), em que a última máquina só é escolhida após percorrer novamente as



demais. Também técnicas híbridas são encontradas usando os princípios do *bottleneck*, como proposta por Balas e Vazacopoulos (1998), que usa uma busca em profundidade local com profundidade controlada;

- *Constraint Satisfaction* - as técnicas de limitação da satisfação reduzem o tamanho do espaço de busca aplicando regras restritivas no momento em que variáveis são selecionadas para expansão. Os primeiros trabalhos baseados em sistemas de escalonamento foram propostos por Fukumori (1980). Aarts e Nuijten em 1995, usaram a técnica com algumas regras diferenciadas (NUIJTEN, W.P.M. and AARTS, E.H.L.(1995));
- Redes Neurais Artificiais - as redes neurais são organizadas em uma estrutura baseada na estrutura das células nervosas do cérebro. LUH, ZHAO e THAKUR (1999) desenvolveram aplicações nesta área, usando redes neurais;
- Buscas Locais e Meta-heurísticas - as buscas locais refinam de forma exata uma solução encontrada por uma heurística. A heurística explora o espaço de busca procurando regiões promissoras, a busca local explora essas regiões procurando pontos ótimos. As meta-heurísticas são técnicas que, quando aplicadas a métodos de busca local, permitem a superação do ótimo local com vistas à obtenção de soluções de qualidade superior;
- *Greedy Randomised Adaptive Search Procedure* (GRASP) - consiste em um método iterativo probabilístico, onde a cada iteração é obtida uma solução do problema em estudo. Cada iteração contém duas fases, uma construtiva, que determina a solução submetida à busca local, outra iterativa que tenta obter alguma melhoria na solução corrente. Resende (1997) apresentou uma aplicação para o GRASP em *job-shop*;
- *Iterative Improvement* - consiste em uma busca iterativa por melhores resultados em uma vizinhança contando com um erro. Neste caso a vizinhança é limitada aceitando apenas um melhor resultado (erro nulo). O algoritmo muda de solução apenas se alguma solução vizinha for maior ou igual a atual. Aarts (1994) aplicou este algoritmo. Um estudo comparativo foi elaborado por Dorn *et al.* (1996), onde analisaram várias técnicas e seus desempenhos;
- Otimização do passo (*large step optimisation*) - desenvolvido por Martin *et al.* (1992), é constituído por duas fases. Na primeira fase um passo largo é otimizado e na segunda o passo otimizado é mais estreito. Esta otimização é feita visando evitar soluções locais;

- *Simulated annealing* - o algoritmo procura iterativamente o próximo candidato a ponto de mínimo na vizinhança do candidato corrente, agindo de acordo com a diferença entre os valores da função-objetivo. Evitam mínimos locais empregando uma busca aleatória que, por vezes, aceita vizinhos que podem ter valores maiores para a função-objetivo. Ou seja, em algumas iterações, o algoritmo tende a maximizar a função-objetivo em vez de minimizá-la. Van Laarhoven *et al.* (1992) construíram um método genérico para o *simulated annealing* controlado pelo tamanho da vizinhança. Este método foi considerado robusto e de fácil implementação. Outra importante definição de vizinhança
- foi proposta por Matsuo *et al.* (1988). Kolonko (1998) indicou que esse algoritmo usado para *job-shop* não converge e que a vizinhança padrão não é simétrica;
- Busca Tabu - os estudos de Busca Tabu tiveram início na década de 80, com os trabalhos de Glover (1997) e Hansen (1986). A Busca Tabu consiste basicamente em uma técnica de melhoria de solução, que considera as estruturas que permitam explorar eficientemente o histórico de todo o processo de busca. Ela tem como característica básica o fato de ser um método genérico que pode usar um tipo de heurística a cada passo e tendo como objetivo, escapar das armadilhas dos ótimos locais ainda distantes de um ótimo global;
- Algoritmos genéticos - o artigo de Davis (1985) foi um dos primeiros que sugeriram a aplicação de algoritmos genéticos aos problemas de *scheduling*. No artigo, ele observa a utilidade de métodos probabilísticos de busca local em ambientes com grandes áreas de busca e sugere uma representação indireta, onde o AG opera sobre uma lista que tem de ser decodificada para formar a programação (*schedule*). Desde Davis (1985), numerosas aplicações foram propostas, dirigindo-se aos vários tipos de agendamento (*scheduling*) existentes e suas variações.

Os métodos apresentados possuem cada um, detalhes que o favorecem conforme situação específica no contexto em que são inseridos. O método aplicando algoritmos genéticos possui uma amplitude maior de aplicabilidade quanto à problemática abordada neste trabalho. Existem na bibliografia técnica, aplicações envolvendo o uso de algoritmos genéticos em diversas áreas do conhecimento, tais como: análise da concessão de crédito, alocação de tarefas em processadores em problemas de processamento paralelo, política e planejamento habitacional, planejamento florestal,

controle ótimo de sistemas de recalque, sistema de apoio ao planejamento de serviços hospitalares e planejamento de rotas aéreas.

Particularmente, com relação ao problema de resolução de programação em grande escala (*Scheduling*), existem várias abordagens utilizando-se AG para solução do problema de alocação ótima de serviços. Alguns destes trabalhos podem ser citados: Problema de Programação em grande escala (YIN *et al*, 2007), Sistema de Execução Industrial Distribuído Híbrido (LIAU *et al*, 2006), Resolução de Problemas de Trabalho em Tojas (XING *et al*, 2007), Distribuição na Tarefa de Distribuição de Características (TRALESKI, 2005) e Escalonador de Ordens de Produção (KOHLENER, 2002).

### 2.1.2 JOB SCHEDULING, UTILIZANDO ALGORITMO GENÉTICO

Atualmente, pode ser encontrado um variado material bibliográfico tratando sobre o assunto de *job scheduling*. Segue uma pequena lista, coletada com idade cronológica não superior a sete anos atrás, envolvendo os artigos mais relevantes para o trabalho em foco.

- “*A Two-Stage Genetic Algorithm for Large-Size Scheduling Problem*”, (YIN et AL, 2007)  
Este artigo propõe a resolução de problemas de programação em grandes escalas. Nele é proposto encontrar uma melhor solução para o problema de *scheduling* através de um algoritmo de duas etapas. Na primeira etapa, o algoritmo genético tenta achar o *fittest* dos controles de parâmetros, isto é, o tamanho da população, a probabilidade de *crossover* e a probabilidade de mutação para um determinado problema. Na segunda etapa, através de um algoritmo genético auxiliar, são aplicados estes parâmetros e encontrada uma solução otimizada. Os autores comentam que para grandes problemas, as duas fases do algoritmo genético podem adquirir ótima solução e de maneira eficaz;
- “*A New Approach for Planning and Scheduling Problems in Hybrid Distributed Manufacturing Execution System*”, (LIAU et al, 2006)  
Este artigo está relacionado ao planejamento e a programação industrial. Os autores do artigo propõe um algoritmo genético que decompõe o planejamento em larga escala em partes que possuam o mesmo nível de problema de agendamento e ao final do artigo o autor compila alguns dados da aplicação do método, na problemática de planejamento e aponta que reduziu em 20% os prazos de vencimento, 8,6% na redução do armazenamento do produto de fabricação, economia de 10% no ciclo industrial interno, e 25% de redução com problemas de qualidade;

- “*An Improved Adaptive Genetic Algorithm for Job-Shop Scheduling Problem*” (XING et al, 2007)

Este artigo aborda a tentativa de resolução da programação de *job-shop*. Neste artigo, o autor comenta que muitos algoritmos genéticos convencionais não podem resolver o problema de *job-shop* de maneira rápida e eficaz. Assim propõe resolver o problema ótimo local. O algoritmo proposto pelo autor herda as características excelentes de cromossomos pai e cria soluções melhores por operação periódica e não linear que ajustam o *crossover* e a probabilidade de mutação. Os resultados experimentais demonstram que o algoritmo não fornece uma solução ótima e é de rápida convergência. Para este papel, uma função sigmóide melhorada é escolhida e que ajusta a função de *crossover* e a probabilidade de mutação. O autor comenta que este tipo de pesquisa baseada em algoritmos genéticos não linear adaptáveis necessitam de muita pesquisa e que muitos aspectos precisam ser melhorados;

- “*An accurate parallel genetic algorithm to schedule tasks on a cluster*”, (Moore, 2004)

É apresentado um algoritmo genético paralelo preciso para programar tarefas em um agrupamento e em “*An accurate and efficient parallel genetic algorithm to schedule tasks on a cluster*”, (Moore, 2003) apresenta o problema de programar tarefas múltiplas em um agrupamento de computadores. Nestes dois artigos, o autor comenta algumas recentes inovações na estimação matemática de parâmetros de algoritmos genéticos paralelos aplicados ao problema NP-completo em um agrupamento de computadores conectado por um servidor compartilhado. As experiências revelam que o algoritmo de programação paralelo desenvolve horários muito precisos quando as diretrizes de parâmetro são usadas;

- “*A Hybrid Genetic Algorithm for Process Scheduling in Distributed Operating Systems Considering Load Balancing*” (Haghighat and Nikravan, 2005)

É proposto neste trabalho um algoritmo genético híbrido para programação de processo em sistemas operacionais distribuídos que consideram balanceamento de carga. Neste artigo, os autores combinam o poder de algoritmos genéticos e recozimento simulado para resolver problemas de balanceamento de carga eficaz. Eles avaliaram o desempenho e eficiência do algoritmo proposto comparando com outros algoritmos tradicionais e heurísticos. Segundo os autores o algoritmo proposto superou os algoritmos existentes;

- “*An Indirect Genetic Algorithm for a Nurse Scheduling Problem*”, (Aickelin et AL,2004)

Este trabalho trata sobre problema de programação dos serviços executados por enfermeiras em um hospital. Este artigo aponta que embora os algoritmos genéticos foram

prosperamente usados para problemas semelhantes no passado, eles sempre tiveram que superar as limitações do paradigma do conflito entre objetivos e constrangimentos. A aproximação levada aqui é usar uma codificação indireta baseada em permutações das enfermeiras, e um decodificador heurístico que constrói horários destas permutações. Os experimentos realizados com base em 52 semanas de dados, avaliam três decodificadores diferentes com níveis variados de inteligência, e quatro operadores de *crossover* famosos. O autor aponta que melhores resultados são obtidos apresentando um operador de *crossover* híbrido e fazendo uso de saltos simples para reduzir o tamanho do espaço de solução. Os resultados revelam que o algoritmo proposto pode achar soluções boas e é mais rápido e mais flexível que uma Busca Tabu;

- “*Building Better Nurse Scheduling Algorithms*”, (Aickelin and White, 2004)

Neste artigo é abordado o mesmo tema do artigo anterior, trata sobre programação dos horários das enfermeiras. O artigo em resumo, o autor tenta desenvolver um método estatístico que compare os resultados obtidos e assim construir algoritmos mais eficientes;

- “*A Genetic Algorithm with an incomplete representation for the Job Shop Scheduling Problems*”, (SONG et al, 2004)

Um Algoritmo Genético com uma representação incompleta para problemas de programação de “*Job-Shop*” é proposto neste trabalho. Propõe-se um algoritmo genético com uma representação incompleta (o número de genes é menor que o número de operações) e aplica isto ao problema em questão. No algoritmo apresentado, o mais importante é a parte do horário decodificado de um cromossomo e o resto do horário é completado por uma regra heurística simples.

## 2.2 ALGORITMOS GENÉTICOS - CONCEITOS PRINCIPAIS

A extensão dos termos e técnicas empregadas dentro de um algoritmo genético é muito ampla, de modo que estes conceitos principais serão apenas um breve resumo de alguns termos e procedimentos para que o leitor menos familiarizado possa acompanhar o prosseguimento do conteúdo deste trabalho.

A primeira tentativa de representação, por meio de um modelo matemático, da teoria de Darwin, surgiu com o livro *The Genetic Theory of Natural Selection*, escrito pelo biólogo evolucionista Fisher (FISCHER, R.A. (1930)) . LINDEN (2006) em seu livro, afirma que “*Uma das primeiras tentativas de se associar a evolução natural a problemas de otimização foi feita em*

1957, quando Box apresentou seu esquema de operações evolucionárias. Estas eram um método de perturbar de forma sistemática duas ou três variáveis de controle de uma instalação, de forma análoga ao que entendemos hoje como operadores de mutação e seleção (GOLDBERG,1990). Logo depois no começo da década de 1960, Bledsoe e Bremmerman começaram a trabalhar com genes , usando tanto a representação binária quanto a inteira e real, e desenvolvendo precursores dos operadores de recombinação (crossover)”.

Em meados da década de 60, John Holland dedicou-se ao estudo de processos naturais adaptáveis, propondo assim os AG's. Ele desenvolveu os AG's em conjunto com seus alunos e colegas da Universidade de Michigan nos anos 60 e 70, tendo como objetivos:

- estudar formalmente o fenômeno da adaptação como ocorre na natureza e
- desenvolver modelos em que os mecanismos da adaptação natural pudessem ser importados para os sistemas computacionais.

Como resultado do seu trabalho, em 1975, Holland editou *Adaptation in Natural and Artificial Systems* e em 1989, Goldberg edita *Genetic Algorithms in Search, Optimization and Machine Learning* , sendo atualmente considerados os livros mais importantes sobre AG's.

Os algoritmos genéticos são classificados como técnicas de busca. Na Figura 4, é apresentado um diagrama que posiciona os algoritmos evolucionários como técnica de busca. Eles são técnicas aleatórias-guiadas que, assim como as técnicas de resfriamento simulado (*simulated annealing*), têm componentes aleatórios, mas dependem do estado corrente para determinar seu próximo estado, aplicando critérios probabilísticos na definição dos estados futuros. Isto é, a informação conhecida direciona a busca, o que diferencia os algoritmos evolucionários de métodos puramente aleatórios como as “*random walks*”.

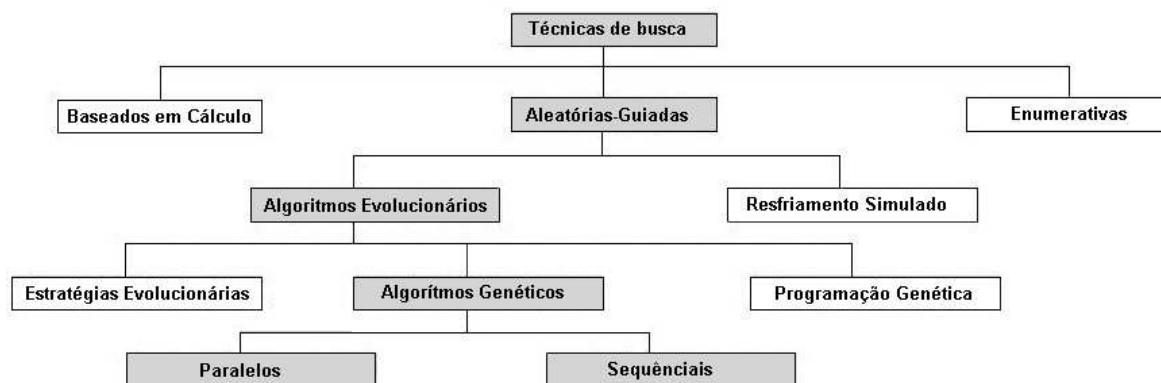


Figura 4. Diagrama posicionamento dos Algoritmos Genéticos como Técnica de Busca  
 Fonte: LINDEN (2006)

Conforme apresentado na Figura 4, os AGs são técnicas de otimização que utilizam princípios de busca aleatória, isto é, possuem componentes aleatórios mas usam as informações do estado corrente para guiar a busca ou definição das populações futuras, as quais são potenciais soluções do problema. AGs estão enquadrados como algoritmos evolucionários, podendo ter uma arquitetura sequencial (segue modelo básico, sem implementações de técnicas de paralelização) ou paralela. A implementação de técnicas de paralelização tem sido usadas como alternativas para o aumento de desempenho de algoritmos genéticos, sobretudo pela ineficiência atingida na execução dessas aplicações em computadores mais simples. Além da restrição em relação ao poder computacional encontrado em um processador comercial. A possibilidade de utilizar implementações diferentes nos procedimentos internos do AG, em relação a versão sequencial, é uma vantagem bastante interessante e atrativa (PETTEY et al. 1987), (FILHO et al. 1993), (ADAMIDIS 1994), (PAZ-CANTU 1997). Não se pretende entrar no mérito da arquitetura paralela ou quando utilizar a computação paralela, mas cabe enfatizar que para este tipo de tecnologia os AG são excelentes candidatos para paralelização visto que já existem alguns modelos que poderiam facilmente aproveitar-se desta tecnologia, como *Panmitic* (cada AG sendo executado num processador distinto), *Island* e *Finely Grained*.

Os algoritmos evolucionários usam modelos computacionais dos processos de evolução como uma ferramenta para resolução de problemas e conceitos de simulação de evolução das espécies por seleção, mutação e reprodução. Mantém uma população de estruturas denominadas indivíduos ou cromossomos onde cada indivíduo recebe uma avaliação que é a quantificação da sua qualidade como solução para o problema. Os operadores genéticos são aplicados para simular a

sobrevivência do mais apto. É importante observar que operadores genéticos consistem em aproximações computacionais de fenômenos da natureza como reprodução, mutação genética ou qualquer outra forma que a imaginação do programador puder reproduzir. Na Figura 5, tem-se o código básico de um algoritmo genético.

```

T:= 0
Inicializa_população P(0)
Enquanto não terminar faça
    Avalie_população P(T)
    P' := Selecciona_pais P(t)
    P' := Recombinação_e_mutação P'
    Avalie_população P'
    P(T+1) = Selecione_sobreviventes P(T), P'
    T := T+1
Fim enquanto

```

Figura 5. Algoritmo genético básico

Devido a inspiração na genética, na evolução das espécies e na biologia, muitos termos usados nestas áreas são utilizados também nos AGs. (Tabela 1).

Tabela 1. Terminologia AG

<b>Linguagem natural</b>	<b>AG</b>
Cromossomo	indivíduo, cromossomo, string, árvore
Gen	característica
Alelo	valor
Lócus	posição
Genótipo	estrutura
Fenótipo	conjunto de parâmetros

Fonte: LINDEN (2006).

O uso de AGs vem ao encontro na resolução de alguns problemas, como quando existem domínios onde o conhecimento específico não pode ser embutido ou calculado computacionalmente. Poder manter uma população de soluções que são avaliadas simultaneamente. Ser global, pois não usam apenas informações locais, conseqüentemente não ficam presos a máximos locais. A busca no AG, não é totalmente aleatória, pois leva em conta a situação da população corrente para determinar o próximo estado de busca. Também não é afetada por descontinuidade na função ou nas derivadas, pois não as usa. Isto é muito útil onde não temos como



calcular a derivada. O AG é capaz de lidar com funções discretas e contínuas (reais, discretas, booleanas ou não numéricas (categóricas)).

A representação das possíveis soluções para o espaço de busca de um problema define a estrutura do cromossomo a ser manipulado pelo algoritmo. A representação do cromossomo depende do tipo de problema e do que, essencialmente, se deseja manipular geneticamente. Os principais tipos de representação são relacionados a seguir, através de exemplos:

- Codificação Binária: é a mais comum devido a sua simplicidade. Cada cromossomo é uma string de bits – 0 ou 1 (Crom: A = 1 0 1 1 0 0 1 0 1 1, Crom: B = 1 1 1 1 1 1 0 0 0 0). Como exemplo, tem-se o problema da mochila, onde cada bit diz se um elemento está ou não na mochila;
- Codificação por permutação: mais usado em problemas de ordenação. Cada cromossomo é uma string de números que representa uma posição numa seqüência (Crom A: 1 5 3 2 6 4 7 9 8, Crom B: 8 5 6 7 2 3 1 4 9). Como exemplo, tem-se o problema do caixeiro viajante, onde os cromossomos descrevem a ordem em que o caixeiro irá visitar as cidades;
- Codificação por valor: usado em problemas onde valores mais complicados são necessários. Cada cromossomo é uma seqüência de valores (Crom A: 1.2324 5.3243 0.4556 2.3293 2.4545, Crom B: ABDJEIFJDHDIERJFDLDFLFEGT . Crom C: (back), (back), (right), (forward), (left)). Como exemplo de uso numa estrutura, encontrar pesos para uma rede neural, onde os valores reais num cromossomo representam pesos em uma rede neural.

Na tabela 2, tem-se outras formas de representações com exemplos de aplicação.

Tabela 2. Principais tipos de representação do cromossomo

<b>Representação</b>	<b>Problemas</b>
Binária	Numéricos, Inteiros
Números Reais	Numéricos
Permutação de símbolos	Baseados em ordem
Símbolos repetidos	Grupamento

Fonte: Pacheco (1999)

A avaliação é o elo entre o AG e o mundo externo. A avaliação é feita através de uma função que melhor representa o problema e tem por objetivo fornecer uma medida de aptidão de cada indivíduo na população corrente, que deve dirigir o processo de busca. A função de avaliação é para um AG o que o meio ambiente é para seres humanos. As funções de avaliação são específicas de cada problema.

A seleção dos pais (cromossomos) deve simular o mecanismo de seleção natural que atua sobre as espécies biológicas, onde os pais mais aptos geram mais filhos e onde os pais menos aptos também podem gerar descendentes. Consequentemente privilegia-se os indivíduos com função de avaliação alta sem desprezar os com função de avaliação extremamente baixa. Se apenas os melhores indivíduos reproduzirem, a população tenderá a ser composta de indivíduos cada vez mais semelhantes e sem diversidade, ocorrendo a convergência genética, sendo este processo chamado de elitismo. Existem várias técnicas de seleção, conforme apresentadas no Anexo A, o qual foi criado apenas com o intuito de apresentar a existência de diversos métodos de seleção e um resumo básico de como é implementado.

Ao cruzarem-se dois indivíduos, os progenitores escolhidos na população através de uma operação de seleção geram um novo indivíduo descendente. Isto é feito a partir de material genético dos cromossomos dos progenitores que uma vez combinado produz o cromossomo de um novo indivíduo ou descendente. A este processo chamamos de *crossover*. Seguem dois exemplos de tipos de cruzamento: Cruzamento uniforme (Figura 6) e Cruzamento de um ponto (Figura 7). No cruzamento uniforme, para cada gene é sorteado um número 0 ou 1. Se for 1, o filho 1 recebe o gene corrente do Pai 1 e o filho 2 recebe o gene corrente do Pai 2. Se for 0, o filho 1 recebe o gene corrente do Pai 2 e o filho 2 recebe o gene corrente do Pai 1.

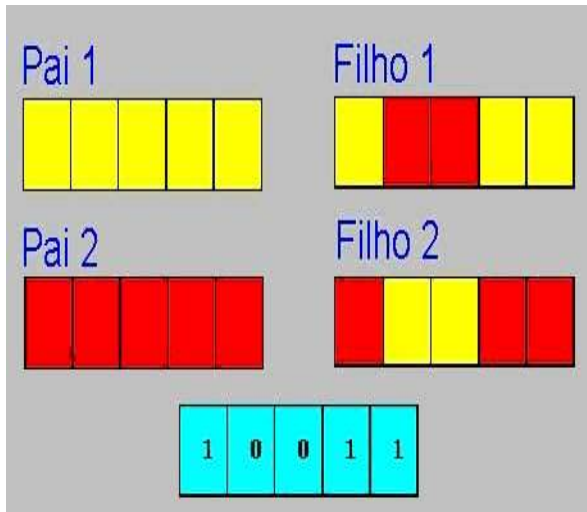


Figura 6. Cruzamento uniforme

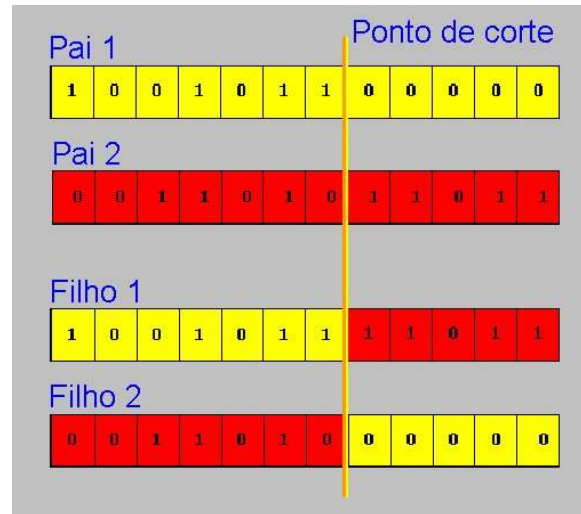


Figura 7. Cruzamento de um ponto

Depois de compostos os filhos, pode-se aplicar a mutação. Atua sobre o gene, alterando seu valor aleatoriamente. A idéia intuitiva por trás do operador de mutação é criar uma variabilidade extra na população, mas sem destruir o progresso já obtido com a busca. Uma taxa baixa diminui a inclusão de indivíduos novos na população e provoca o encontro de resultados baseados em máximos locais, restringindo o espaço de busca. Uma taxa muito alta transforma a busca do algoritmo genético em busca essencialmente aleatória.

### 2.3 CONSIDERAÇÕES FINAIS

Como conclusão deste estudo bibliográfico, em função das várias aplicações de sucesso propostas na literatura para o problema do *Job scheduling*, nas suas mais variadas formas, é que os algoritmos genéticos são uma ferramenta relevante para efetuar buscas em espaços praticamente infinitos, além de serem razoavelmente eficientes para a totalidade dos problemas existentes. Cabe salientar que alguma ferramenta especificamente projetado para o problema tratado, poderia ser mais eficiente, contudo, ocorrendo perturbações no problema original trariam quase nenhum prejuízo aos AGs, mas possivelmente, a inutilidade de outros métodos ou ferramenta. Resumindo, o AG é uma técnica extremamente eficiente com boas soluções para o problema, são extremamente simples de se implementar e de se modificar. O custo de pessoal para implementação é mínimo.

A pequena lista de casos apresentada, por si só encoraja a utilização do AG para resolução do problema tema desta dissertação. Obviamente que cada um destes artigos mencionados, possui detalhes diferentes entre si e é justamente esta diferença que torna encorajador o uso do AG, bastaria apenas modelar o problema e inseri-lo dentro do algoritmo genético, respeitando o modelo encontrado.

Em todos os artigos pesquisados, os autores deram ênfase ao fato de que definir o modelo do cromossomo usado e a função de avaliação devem merecer especial atenção para o sucesso do algoritmo ser capaz de resolver o problema em questão.

### 3 DESENVOLVIMENTO

Durante o desenvolvimento deste trabalho fez-se necessário, o levantamento de uma quantidade considerável de informações relacionadas a algoritmos genéticos e *scheduling*. Estas informações, trabalhadas serviram de base para a construção de uma aplicação específica, com o objetivo de aumentar o desempenho quanto a tarefa de alocação dos serviços de metrologia aos recursos disponíveis. Esta aplicação foi se delineando no decorrer que se foi assimilando os detalhes mais técnicos envolvendo a técnica de algoritmos genéticos (mutação, *crossover*, formas de representação, técnicas de seleção). No decorrer do projeto ocorreram novas implementações e melhorias no algoritmo genético adotado, tornando-o mais eficiente e capacitado para um aumento ou redução na demanda de serviços e ou de recursos disponíveis.

Na execução das versões do sistema, foi utilizado um computador Pentium 4, 3 GHz, utilizando Windows XP *Professional* com 2 Gb de memória e um notebook 2.66 GHz com 1Gb de memória, para instalação do compilador onde o sistema foi implementado. Como compilador foi utilizado o Delphi 5.0 da Borland ® versão *professional*. Os dados para testes foram retirados do histórico do banco de dados de um laboratório prestador de serviços de metrologia da cidade de Joinville, SC. O banco de dados utilizado foi editado para não comprometer informações pertinentes a empresas e laboratórios.

#### 3.1 METODOLOGIA E PLANO DO TRABALHO

A programação do sistema segue os princípios gerais dos métodos baseados em objetos. O paradigma de objetos é o mais recomendado para orientar a modelagem quando se visa a reutilização de software, dado que os objetos ocultam a implementação dos atributos e das funções que oferecem ao ambiente externo. Encapsulados no componente, estes atributos e funções podem ser reutilizados por diversos outros sistemas, exatamente como propõe a reutilização de produtos de software. Os produtos de software também podem ser entendidos como objetos que ocultam a implementação de atributos e funções.

Desta forma, a implementação pode utilizar o mesmo paradigma utilizado na análise, simplificando a construção do sistema (CONSTANTINE, 1995). Os métodos orientados por objetos permitem uma clara separação entre as fases de análise e implementação, permitindo que se trate do problema, independentemente de como será construído o sistema. (DEJESUS, 1995). O modelo é

formado por um conjunto de diagramas e documentos associados às fases da sua evolução. Parte-se de uma visão geral, que é decomposta em diagramas de classes, diagramas de estado e diagramas de eventos. Procura-se aumentar o nível de detalhes do modelo até o ponto em se pode iniciar a implementação, traduzindo as classe do modelo em produtos de software. A notação apresentada pelo método OMT (*Object Modeling Technique*), desenvolvido por Rumbaugh *et al.*(1993) é utilizada visto que ela oferece a flexibilidade para modelar sistemas de um modo geral, e há disponibilidade de uma ferramenta CASE para dar apoio à construção dos diagramas (MICROGOLD, 1996).

O plano de trabalho foi dividido nas seguintes fases:

- Leituras e levantamento do estado de arte. Refinar estado de arte, especificamente para o caso de algoritmos genéticos e *scheduling*;
- Levantamento das variáveis reais que fazem parte na equação para resolução do problema de distribuição de serviços junto a alguns principais laboratórios prestadores de serviços metrológicos;
- Confronto das variáveis levantadas com as metodologias existentes no que se refere a algoritmo genético, para poder definir que algoritmo melhor se enquadra para estas variáveis e adaptá-lo para que possa ser implementado;
- Verificação da necessidade de redefinir a inclusão de recursos (mais disponibilidade dedicação, participação e envolvimento de laboratórios que possam servir como fonte de informações e testes) com base a situação cronológica do projeto ;
- Levantamento das informações e definição da base de dados. Nesta fase, definir todos as variáveis que irão compor o espaço do problema, mapear estas variáveis e documentá-las;
- Modelagem do algoritmo genético propriamente dito, com destaque para a definição do cromossoma e função fitness para o problema de job scheduling em um laboratório de metrologia
- Implementação do algoritmo na prática e sua interface (em Object Pascal - Delphi);

- Calibração dos parâmetros do AG (taxa de mutação, ponto de crossover, seleção dos pais, entre outras variáveis, todas dependendo da forma como a técnica foi implementada) através de testes com dados reais;
- Realização de testes, verificando a performance do sistema construído, seus gargalos, ajustes de eficiência, correção de erros.

## **3.2 MODELO DE ALGORITMO GENÉTICO**

No decorrer do desenvolvimento desta dissertação, diversas versões do modelo foram implementadas. A medida que uma determinada versão foi se tornando estável, partiu-se para inserir mais detalhes, restrições e melhorias em sua performance. Segue um resumo histórico destas versões.

### **3.2.1 ESQUELETO DO PROGRAMA**

Inicialmente desenvolveu-se um algoritmo genético básico para resolver o problema de distribuição de itens em alguns recursos disponíveis, não levando em consideração grande parte das restrições pertinentes ao problema. Este algoritmo serviu de base para posteriormente criar um modelo melhorado, projetado para promover um aumento no desempenho do algoritmo e na quantidade de soluções satisfatórias para a problemática. Como o objetivo deste trabalho é a otimização deste algoritmo básico, procurou-se construí-lo de tal maneira que permitisse futuros incrementos em seu modelo. Como dados para alimentar esta versão inicial, criou-se duas listas fixas, sem especial atenção as diversas características destes dados (restrições).

Na Figura 9 e 10 é apresentada a lista dos itens e recursos que fizeram parte do estudo inicial. Percebe-se que os itens possuem tipos de serviços diferentes e os recursos que executam estes serviços por sua vez podem realizar somente determinados tipos de serviços. Estes itens e recursos serviram para simular o problema em foco.

As restrições que constituem a parte central do problema são tratadas no momento de pontuar o cromossomo.

Recurso	Minutos	Horas	Serviço	Disponível
Rec 1	480	8	Calibração	S
Rec 2	1440	24	Calibração	S
Rec 3	480	8	Manutenção	S
Rec 4	720	12	Manutenção	S
Rec 5	1440	24	Calibração	S
Rec 6	480	8	Calibração	S
Rec 7	480	8	Calibração	S
Rec 8	480	8	Manutenção	N
Rec 9	720	12	Calibração	N
Rec 10	480	8	Calibração	S

Figura 9. Lista de recursos para testes iniciais

Item recebido	Tipo serviço	Tempo (min)	Preparo (min)	Considerar
Item 1	Calibração	45	0	S
Item 2	Calibração	35	0	S
Item 3	Calibração	35	0	S
Item 4	Manutenção	60	0	S
Item 5	Manutenção	45	0	S
Item 6	Calibração	135	10	S
Item 7	Calibração	45	0	S
Item 8	Manutenção	35	0	S
Item 9	Calibração	35	0	S
Item 10	Calibração	25	0	S
Item 11	Manutenção	120	0	S
Item 12	Calibração	300	25	S
Item 13	Calibração	450	15	S
Item 14	Calibração	125	0	S
Item 15	Calibração	450	15	S
Item 16	Calibração	25	0	S
Item 17	Calibração	25	0	S
Item 18	Calibração	60	0	S
Item 19	Calibração	45	0	S
Item 20	Calibração	120	10	S

Figura 10. Lista de itens recebidos para testes iniciais

O modelo de cromossomo adotado é o de uma string de “0” e “1” de tamanho igual a quantidade de recursos envolvidos vezes a quantidade de itens envolvidos. A presença de “1” em uma determinada posição da string implica que aquele referido item estará alocado para aquele



referido recurso. O modelo pode ser melhor compreendido através da Figura 11. Nesta figura, exemplo, representa-se um cromossomo projetado para atender até 5 itens e 3 recursos. O cromossomo é dividido em 3 partes, cada uma destas partes representa um recurso. Para cada uma destas partes, pode-se alocar 5 itens, cada um representado por um caracter. Tem-se assim um cromossomo com 15 caracteres que representam assim estes 3 recursos e 5 itens. Caso tenha-se 20 itens para sofrer serviços de metrologia e 10 recursos disponíveis, o modelo de cromossomo adotado neste caso é de 20x10 alelos; ou seja 200 caracteres;

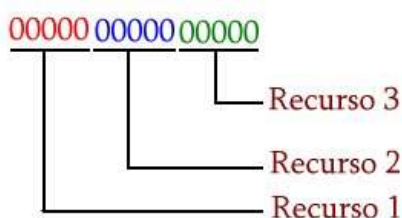


Figura 11. Modelo de cromossomo adotado

A utilização desta forma do cromossomo, permite que se possa tratar de maneira simples o cromossomo propriamente dito, aplicar mutação, *crossover* e identificar a situação do alelo dentro do cromossomo facilmente.

Para uma melhor compreensão do cromossomo (mais detalhes podem ser observados na seção “tratando itens clones”), será apresentada mais um exemplo a título de ilustração de como o cromossomo aqui foi modelado. Será usado o modelo de cromossomo ilustrado da Figura 11. Supondo ter-se a disposição 3 recursos vazios, ou seja nada fora alocado a estes recursos e fora recebido 5 itens para serem distribuídos nestes 3 recursos, teremos isso ilustrado na figura seguinte a Figura 12, onde os itens a serem distribuídos são representados pelos retângulos de cor vinho, azul, verde, azul e laranja. Os retângulos brancos são os espaços vazios que podem ser preenchidos com estes itens. Na Figura 13, tem-se alocados 5 itens nestes recursos. Dois no primeiro, dois no segundo e um no terceiro recurso.

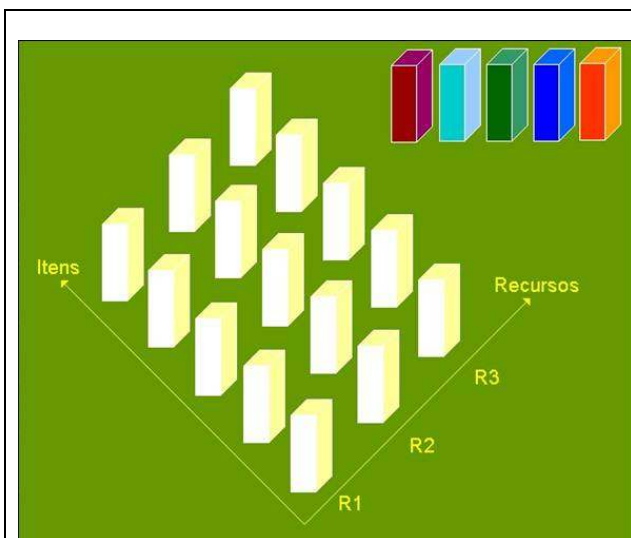


Figura 12. Representação vazia do cromossomo

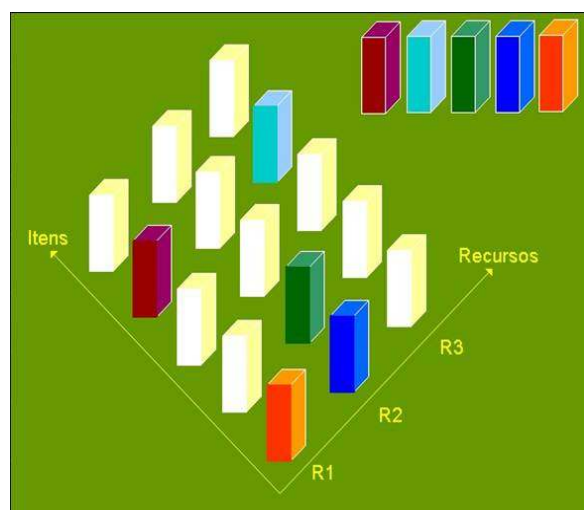


Figura 13. Representação cromossomo preenchido

O cromossomo representado na Figura 12 seria “00000 00000 00000” enquanto que o mesmo cromossomo na Figura 13 seria “10010 11000 00010”. Aumentando a complexidade do exemplo vamos supor que tenhamos o seguinte caso de distribuição neste mesmo exemplo, conforme a Figura 14.

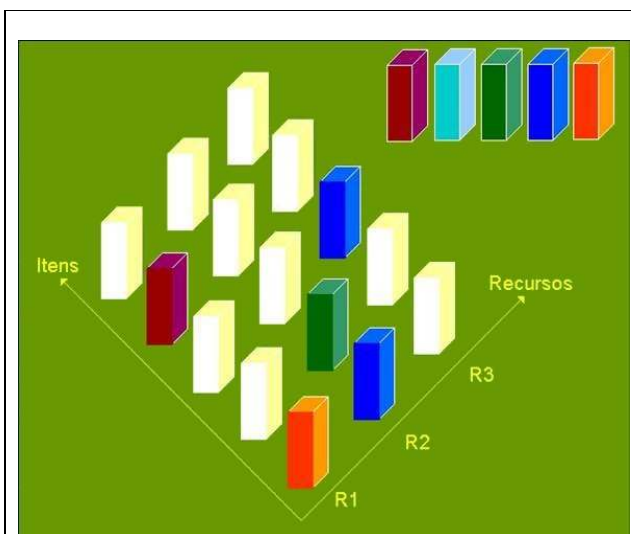


Figura 14. Representação cromossomo preenchido com clones

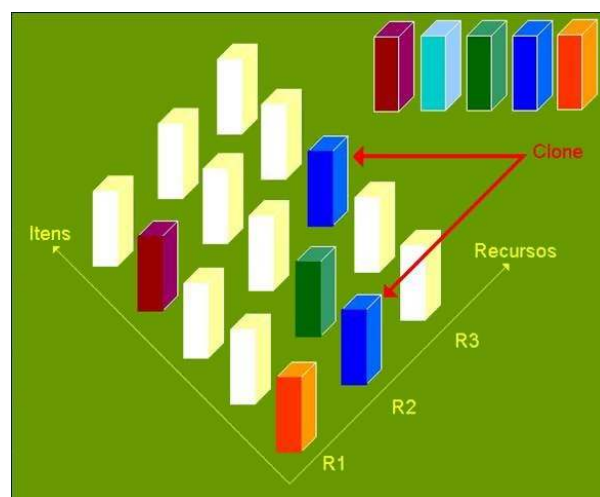


Figura 15. Representação cromossomo preenchido com o item clone evidenciado

Tem-se nesta figura a representação do cromossomo como sendo “10010 11000 00100” perfeito, temos os 5 itens distribuídos nos três recursos. Porém ao analisar verifica-se que o quarto

item (azul) esta repetido no recurso 2 e no recurso 3. Esta repetição apontada na Figura 15 é um erro que deve ser corrigido, afinal um item não pode ser executado em dois recursos diferentes. Denomina-se esta repetição de um mesmo item como sendo um item clone.

### **3.2.2 DEFININDO A FUNÇÃO DE AVALIAÇÃO**

O êxito ou bom desempenho de qualquer algoritmo genético é a função de avaliação. Ela que irá informar se o indivíduo tem ou não condições de representar uma solução e quanto esta próximo do esperado se encontra esta solução.

Nossa função de avaliação é composta de duas partes. A primeira irá avaliar as restrições envolvidas e a segunda caso tenha sido aprovado irá pontuar o indivíduo. Caso o indivíduo não se qualifique nas restrições, sua pontuação será zero. Espera-se assim eliminar os candidatos que não possam oferecer margem de aceitação para a resolução do problema em questão. Não são aplicados métodos de penalizações e sim verificações das restrições que atendam à solução.

Como principais restrições analisadas temos:

- A capacidade volume trabalho recurso. O recurso não pode receber mais itens para execução do que a sua capacidade de trabalho.
- Compatibilidade tipo de item X recurso. Se o recurso é específico para um determinado tipo de item então não pode receber um item diferente para qual foi projetado.
- Compatibilidade tipo de serviço X tipo de serviço que o recurso executa. Se um determinado recurso executa apenas calibração (por exemplo) não pode receber um item que tenha como serviço manutenção.
- Compatibilidade faixa de trabalho do recurso X faixa de medição do item. Se um determinado recurso foi projetado para fornecer leituras de 0 a 100 mm (exemplo), não podemos alocar um item que tenha uma faixa de trabalho de 200 a 300 mm.
- Quantidade de itens a serem agendados. Não é uma restrição, porém irá ajudar a pontuar o cromossomo. Quanto maior o numero de itens agendados presentes no cromossomo, maior será sua pontuação.

- Prazo de entrega do item. Os itens devem ser agendados, obedecendo o prazo de entrega previsto. Se um item de prazo de entrega posterior, for agendado antes de um item mais urgente, estaremos penalizando este último e favorecendo um item não urgente.

Assim nossa função de avaliação exibida na Equação 1 retorna um valor que representará o quão satisfatório o cromossomo é como candidato a solução do problema.

Equação 1. Função de avaliação do Algoritmo Genético

$$OK = ((NIC \leq NIR) \text{ e } (TSR_r = TSI_i) \text{ e } (FMR_r \geq FMI_i) \text{ e } (TMR_r \leq (TAR_r + TEI_i)))$$

se OK então

$$f = \% \text{ NIC}$$

senão

$$f = 0;$$

(Variáveis)

- ▶ TSR – tipo serviço recurso
- ▶ TSI – tipo serviço item
- ▶ FMR – faixa de trabalho do recurso
- ▶ FMI – faixa de medição do item
- ▶ PE – prazo de entrega
- ▶ TMR – total minutos recurso
- ▶ TAR – total acumulado recurso (minutos)
- ▶ NIR – número de itens recebidos
- ▶ NRD – número de recursos disponíveis
- ▶ NIC – número de itens no cromossomo
- ▶ NC – número de itens clones
- ▶ TEI – tempo de execução do serviço no item
- ▶ OK – se restrições de acordo
- ▶  $i,r$  – índices referentes ao item e recurso, no do cromossomo

A medida que o problema principal torna-se mais complexo, podemos de maneira fácil alterar a função de avaliação para contemplar a nova exigência deste problema. Vê-se aqui, a parte mais interessante do algoritmo genético que é a sua manutenção.

### 3.2.3 DEFININDO OS PARÂMETROS DO ALGORITMO GENÉTICO

A definição dos parâmetros deve ser realizada com atenção. Incorporou-se três métodos de *crossover* (*Crossover* de dois pontos, de três pontos e uniforme), por serem os mais simples de implementação. Para a seleção dos pais, foi utilizada a técnica do Torneio, ficando outras técnicas para estudo posterior. Na Figura 16, são apresentadas diferentes possibilidades de ajuste dos parâmetros do algoritmo genético, constantes no *front end* da versão inicial desenvolvida. Embora a tela apresentada contenha outros valores, as adotadas neste trabalho foram:

- Técnica de troca de população: troca de toda a população;
- Tamanho da população : 100;
- Número de gerações : 2000;
- Crossover: crossover de dois pontos;
- Tipo de seleção: torneio , taxa de 25 %;
- Tipo de avaliação: normalizado (adotado função específica ) ;
- Função de avaliação: percentual de distribuição sem clones no cromossomo, isto é, maior número de itens distribuídos nos recursos disponibilizados.

O tamanho da população foi empírica. A escolha do tamanho foi definida, levando unicamente em conta reduzir o tempo de processamento e manter-se dentro da capacidade de memória do computador utilizado para executar o algoritmo.

O número de gerações levou em conta explicitamente como critério, manter-se dentro um tempo de processamento aceitável..

Cabe uma pequena observação no que se refere a taxa de elitismo e taxa de torneio. Supondo que tenha se definido uma população de 100 indivíduos e variando estas taxas tem-se o relacionamento entre elas de acordo com o exemplo seguinte: supondo 0% de taxa de elitismo e 25% de torneio. Significa que 25% dos 100 indivíduos da população participarão do torneio. Já com uma taxa de elitismo de 40%, tem-se 60% da população sobrando. A taxa de torneio é aplicada então a este 60%, pois 40 % dos indivíduos já estão separados para a próxima geração. Desta

maneira simplificada, a taxa de elitismo separa os melhores pontuados para permanecerem na próxima geração e os demais parâmetros são aplicados nos itens restantes.

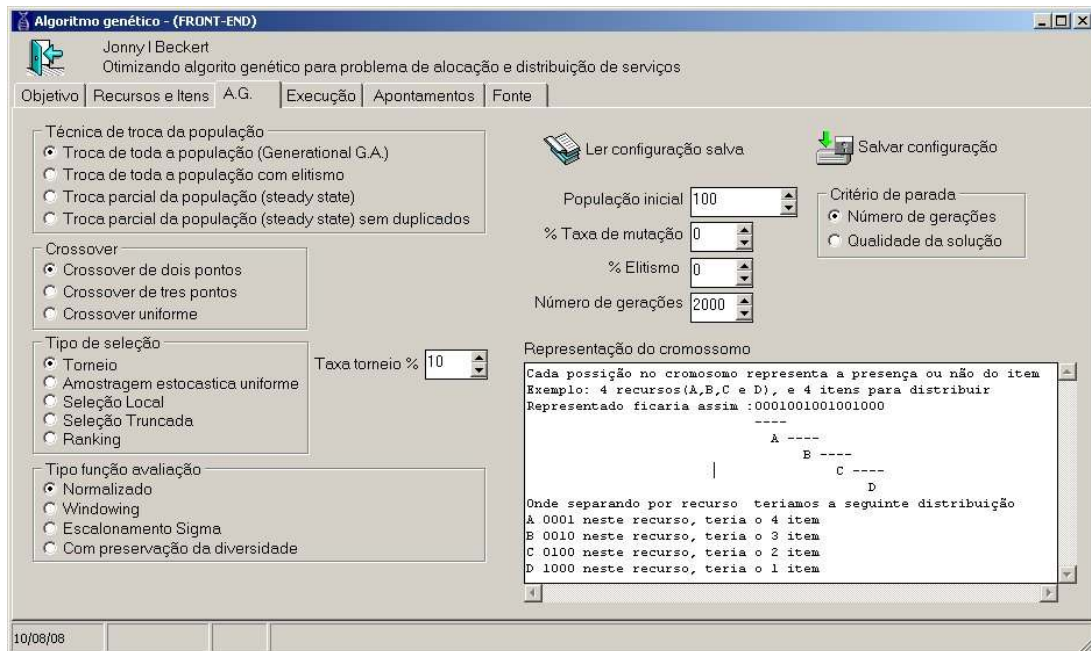


Figura 16. Tela de parâmetros do AG.

Nesta versão inicial do algoritmo genético, não foi implementada nenhuma melhoria no código do algoritmo, sendo o mesmo em sua forma simplificada seguindo praticamente o básico do algoritmo genético canônico.

Foi realizada uma série de execuções utilizando dois tipos de testes:

- O primeiro, sem considerar as restrições, ou seja, sem levar em conta se o item alocado a uma determinada máquina pode ou não ser executado. Neste tipo de teste apenas foi considerado que os itens devem ser distribuídos.
- No segundo tipo de teste levaram-se em consideração as restrições, ou seja, o item poderia ser somente alocado a um recurso se este recurso permitisse executar o serviço.

Desta maneira pode-se avaliar o impacto que o processamento das restrições impunham na performance do programa.

Tabela 3. Resultados dos testes aplicados a um AG simples.

Tipo de teste	Média de soluções viáveis encontradas	Média do tempo execução
Sem uso de restrições	6 soluções	59 segundos
Com uso de restrições	8 soluções	1h e 5 minutos

Como era esperado, o desempenho do segundo tipo de teste foi bem inferior ao primeiro tipo pois tem-se de alocar tempo de processamento para executar as restrições solicitadas. Na tabela 3, são registradas as médias de resultados obtidos com os dois tipos de testes, aplicados a um algoritmo genético simples, sem adotar métodos de otimização no código do algoritmo implementado.

### 3.2.4 TRATANDO ITENS CLONES

A função de avaliação do algoritmo genético representa o percentual de distribuição sem clones no cromossomo. Exemplificando, tendo três recursos disponíveis e dez itens para serem distribuídos nestes três recursos, o melhor *fitness* é aquele que conseguiu distribuir os dez itens nestes recursos. Porém existe um detalhe. Não pode haver clones, ou seja, neste hipotético exemplo o cromossomo seria uma string de 3 x 10 caracteres, sendo que a cada dez caracteres representaríamos um recurso. Se nenhum item for distribuído teríamos uma string com 30 “zero”. Se todos os 10 itens forem alocados, teríamos nesta string 20 “zeros” e dez “uns”. Considerando a seguinte string “0101000100 0010100010 1000011001”, são dez itens distribuídos em três recursos sem colisão. Já a string “0101000100 0010100010 1000101001” também possui dez itens, mas o quinto item esta repetido no segundo e no terceiro recurso, ocorrendo uma colisão ou clonagem.

Ficou identificado que seria necessário inserir uma técnica de otimização no algoritmo. Embora a técnica descrita neste artigo seja de certa maneira intuitiva, fez-se uma pesquisa para verificar consenso de mesmas formas de pensar de outros autores envolvidos com problemática semelhante.

De acordo com os ensaios realizados no algoritmo genético simples, não é possível adotar um modelo onde se permita somente ao algoritmo genético encontrar a distribuição dos itens para os recursos. É necessário aperfeiçoar o processo para reduzir o tempo dedicado. Como as restrições principais referem-se a um item que esteja sujeito a um determinado tipo de serviço que somente pode ser executado em um determinado recurso, então é necessário eliminar esta restrição ou

separá-la do gargalo do algoritmo. E a maneira mais simples é não misturar itens e recursos diferentes. Assim, resolveu-se agrupar os itens com o mesmo tipo de serviço e agrupar os recursos que permitem este mesmo tipo de serviço. Com isso, foi eliminada a restrição, pois ela já está sendo tratada separadamente. Basicamente, amplia-se assim o espaço de busca quando retira-se as restrições da função objetivo; diferente do algoritmo anterior onde as restrições estão centradas no momento da pontuação do cromossomo, as restrições de maior impacto foram separadas para fora do laço central. Desta forma, o algoritmo genético melhorado consiste em um laço onde antes de montar o cromossomo, são ordenados e agrupados os itens e recursos. Cada solução encontrada é armazenada e ao final do algoritmo juntam-se as soluções, atendendo assim a 100% dos casos e com a vantagem de se ter a performance melhorada.

Na Tabela 4 são apresentados os resultados de uma nova série de testes realizados utilizando este algoritmo melhorado e no qual foram consideradas as idéias obtidas dos autores anteriormente referenciados, adaptando ao modelo inicial.

Tabela 4. Resultados dos testes aplicados a um AG melhorado

Tipo de teste	Média de soluções encontradas	Média tempo execução
Sem uso de elitismo	36 soluções	43 segundos
Com uso de elitismo	62 soluções	37 segundos

Na Figura 17 é apresentado o código simplificado do algoritmo genético melhorado. Basicamente foram inseridos procedimentos de agrupamento associados a algumas regras básicas, como por exemplo:

- Agrupar todos os recursos por tipo de serviço;
- Agrupar os itens recebidos por tipo de serviço;
- Colocar os recursos com mais carga disponível em primeiro;
- Deixar os recursos com histórico de paradas por último;
- Nos itens a serem agendados, ordená-los por tempo de climatização ficando assim por último os de maior tempo (enquanto um item com menor tempo de climatização é executado, o de maior tempo aguarda a estabilização da temperatura).



O uso de paralelismo (vários A.G sendo executados simultaneamente para cada agrupamento) seria interessante, porém não foi implementado.

```
T0 := 0
Ordena recursos
Ordena itens
Repita
  Agrupa recurso
  Agrupa itens
  T1:= 0
  Inicializa_população P(0)
  Enquanto não terminar grupo e atingir a meta faça
    Avalie_população P(T1)
    P' := Selecciona_pais P(T1)
    P' := Recombinação_e_mutação P'
    Avalie_população P'
    P(T1+1) = Selecciona_sobreviventes P(T1), P'
    T1 := T1+1
  Fim enquanto não terminar grupo
T0 := T0 + 1
Até todos os agrupamentos terem sido concluídos
Mostra os resultados de cada agrupamento
```

Figura 17. Algoritmo genético melhorado

As rotinas de “Agrupa recursos” e “Agrupa itens” devem conter o máximo de regras para remover as restrições de dentro da função de avaliação na função “Avalie população”.

### 3.2.5 INCORPORANDO DADOS REAIS

Tendo agora um modelo testado, criou-se um novo sistema, mas que passasse agora a utilizar dados reais de itens recebidos e recursos disponíveis. Com dados reais e uma quantidade maior de dados alimentando o algoritmo, teria-se uma idéia real de aplicabilidade ou não do modelo criado.

Na Figura 18 observa-se a tela na qual, além de selecionarmos a base de dados de entrada, visualiza-se os recursos existentes e a que tipos de itens estes recursos prestam-se para trabalho, bem como as faixas de trabalho destes recursos.

The screenshot shows a software window titled "Distribuição de serviços utilizando A.G." with a menu bar (Objetivo, Base de dados reais, Configuração do AG, Execução, Gráficos, Interpretação, Aprendizagem) and a toolbar (Contactar, Desconectar). A text field contains "C:\DissertacaoJonny". Below are two tables: "Recursos cadastrados" and "Serviços que o recurso corrente executa".

Código	Recurso a ser agendado	Capacidade	Tempo médio
1	Máquina de medição longitudinal	450	30
2	Não geométricos	240	30
3	Instrumentação	450	15
4	Máquina de medição por coordenadas	450	60
5	Microscópio de medição	480	20
6	Máquina de medição de rugosidade e perfil	480	50
7	Comparador eletromec. de blocos-padrão	480	
8	Laser interferométrico		240
9	Calibrações em Campo		

Serviço	Família	Tipo de item na família	Faixa inicial	Faixa final	Tempo médio
Calibração	Calibrador anel cilíndrico de referência	Todos	3,5	220	20
Calibração	Calibrador anel roscado cilíndrico	Todos	3	90	30
Calibração	Calibrador anel roscado cônico	Todos	3	90	40
Calibração	Calibrador tampão cilíndrico de referência	Todos	0	200	15
Calibração	Calibrador tampão roscado cilíndrico	Todos	0	200	25
Calibração	Calibrador tampão roscado cônico	Todos	0	200	30
Calibração	Escalas	Lupa graduada	0	20	30
Calibração	Gabarito	Lâmina de folga			60
Calibração	Haste-padrão	Todos	0	575	20
Calibração	Instrumento de comprimento	Ogiva	0	100	30
Calibração	Micrômetro	Micrômetro Interno	25	300	20
Calibração	Padrão comprimento (exceto anel, bloco e haste)	Todos	0	575	15
Calibração	Relógio	Todos	10	50	30

Figura 18. Recursos reais disponíveis e os tipos de serviço que executam

Na Figura 19 são apresentados os itens recebidos. Percebe-se a quantidade enorme de itens diferentes que são recebidos diariamente para a execução de serviços diversos. Aproveitou-se que seria necessário validar o sistema desenvolvido, para incorporar esta validação no próprio *front-end* construído. Dividiu-se então esta versão em duas partes, a primeira que servisse para apresentar e quantificar um cenário real de distribuição realizada. A segunda parte iria pegar estes mesmos dados e aplicar o AG, e apresentaria as soluções encontradas. O confronto entre o real e ao apresentado iria fazer a validação do sistema.

Distribuição de serviços utilizando A.G.

Objetivo Base de dados reais Configuração do AG Execução Gráficos Interpretação Aprendizagem

As tabelas devem estar nesta pasta, caso não estejam, especifique o local  
 C:\Dissertacao\Jonny

Recursos cadastrados Itens recebidos Agendamentos

Mostrar somente os itens não agendados Mostrar todos

Prazo entrega	Qtidade	Item recebido	Serviço	NrOS	Seq.	Srv.Cred.	Laboratorio	Certificado	Agendado	Aprovado
20/07/05	1	Calibrador anel liso cilíndrico	Calibração	10350	3	Sim	0	Sim	Sim	Sim
20/07/05	1	Calibrador anel liso cilíndrico	Calibração	10350	4	Sim	0	Sim	Não	Sim
20/07/05	1	Tambor micrométrico (fuso)	Calibração	10350	5	Sim	0	Sim	Sim	Sim
20/07/05	1	Calibrador anel liso cilíndrico	Calibração	10350	6	Sim	0	Sim	Sim	Sim
20/07/05	1	Calibrador anel liso cilíndrico	Calibração	10350	7	Sim	0	Sim	Sim	Sim
20/07/05	1	Calibrador anel liso cilíndrico	Calibração	10350	8	Sim	0	Sim	Sim	Sim
20/07/05	1	Tambor micrométrico (fuso)	Calibração	10350	9	Sim	0	Sim	Sim	Sim
20/07/05	1	Calibrador anel liso cilíndrico	Calibração	10350	10	Sim	0	Sim	Não	Sim
20/07/05	1	Calibrador anel liso cilíndrico	Calibração	10350	11	Sim	0	Sim	Não	Sim
20/07/05	1	Calibrador anel liso cilíndrico	Calibração	10350	12	Sim	0	Sim	Não	Sim
20/07/05	1	Micrômetro de profundidade	Calibração	10350	13	Sim	0	Sim	Não	Sim
20/07/05	1	Relógio comparador centesimal	Calibração	10350	14	Sim	0	Sim	Não	Sim
20/07/05	1	Tambor micrométrico (fuso)	Calibração	10350	15	Sim	0	Sim	Não	Sim
20/07/05	1	Comparador para diâmetros internos	Calibração	10350	16	Sim	0	Sim	Não	Sim
20/07/05	1	Relógio comparador centesimal	Calibração	10350	17	Sim	0	Sim	Não	Sim
20/07/05	1	Comparador para diâmetros internos	Calibração	10350	18	Sim	0	Sim	Não	Sim
20/07/05	1	Relógio comparador centesimal	Calibração	10350	19	Sim	0	Sim	Não	Sim
20/07/05	1	Micrômetro de profundidade	Calibração	10350	20	Não	0	Sim	Não	Sim
20/07/05	1	Micrômetro de profundidade	Calibração	10350	21	Não	0	Sim	Não	Sim
20/07/05	1	Comparador para diâmetros internos	Calibração	10350	22	Sim	0	Sim	Não	Sim
20/07/05	1	Relógio comparador centesimal	Calibração	10350	23	Sim	0	Sim	Não	Sim
20/07/05	1	Comparador para diâmetros internos	Calibração	10350	24	Sim	0	Sim	Não	Sim
20/07/05	1	Relógio comparador centesimal	Calibração	10350	25	Sim	0	Sim	Não	Sim

26/10/08

Figura 19. Itens recebidos e serviços que devem ser executados

Nas Figuras 20, 21 e 22 temos respectivamente o agendamento real realizado no período indicado, o gráfico de distribuição dos itens para um único recurso e o gráfico de distribuição de todos os itens em todos os recursos. É visível que os recursos são mal aproveitados e a distribuição dos itens não esta otimizada.

Distribuição de serviços utilizando A.G.

Objetivo Base de dados reais | Configuração do AG | Execução | Gráficos | Interpretação | Aprendizagem

Contatar Desconectar As tabelas devem estar nesta pasta, caso não estejam, especifique o local  
C:\DissertacaoJonny

Recursos cadastrados | Itens recebidos | Agendamentos

Apresentar somente serviços agendados para o período de: 01/05/07 à 01/06/07

Agendamentos realizados | Gráfico de distribuição do recurso corrente | Gráfico de distribuição de todos os recursos

Máquina de medição longitudinal

Recurso	Agendado para	Nº ordem	Executado em	Tecnico	Início	Fim	Situação	Item	Previsão	Real
1	02/05/07	10	03/05/07	2619-8	22:21:54	22:22:03	Concluído	Calibrador tampão roscado cilíndrico (PNP)	20	
1	02/05/07	20	03/05/07	2619-8	22:21:55	22:22:02	Concluído	Calibrador tampão roscado cilíndrico (PNP)	20	
1	02/05/07	30	03/05/07	2619-8	22:21:56	22:22:02	Concluído	Calibrador tampão roscado cilíndrico (PNP)	20	
1	02/05/07	40	03/05/07	2619-8	22:21:56	22:22:01	Concluído	Calibrador tampão roscado cilíndrico (PNP)	20	
1	02/05/07	50	03/05/07	2619-8	22:21:57	22:22:01	Concluído	Calibrador tampão roscado cilíndrico (PNP)	20	
1	02/05/07	60	03/05/07	2619-8	22:21:57	22:22:00	Concluído	Calibrador tampão roscado cilíndrico (PNP)	20	
1	02/05/07	70	03/05/07	2619-8	22:21:58	22:22:00	Concluído	Calibrador tampão roscado cilíndrico (PNP)	20	
1	02/05/07	80	27/04/07	2970-5	10:03:19	10:03:21	Concluído	Calibrador tampão liso cilíndrico (PNP)	20	
1	02/05/07	90	27/04/07	2970-5	10:03:24	10:03:26	Concluído	Calibrador tampão liso cilíndrico (PNP)	15	
1	02/05/07	100	27/04/07	2970-5	10:03:27	10:03:28	Concluído	Calibrador tampão liso cilíndrico (PNP)	15	
1	02/05/07	110	27/04/07	2970-5	10:03:29	10:03:30	Concluído	Calibrador tampão liso cilíndrico (PNP)	15	
1	02/05/07	120	27/04/07	2970-5	10:03:31	10:03:32	Concluído	Calibrador tampão liso cilíndrico (PNP)	15	
1	02/05/07	130	27/04/07	2970-5	10:03:33	10:03:35	Concluído	Calibrador tampão liso cilíndrico (PNP)	15	
1	02/05/07	140	27/04/07	2970-5	10:03:37	10:03:38	Concluído	Calibrador tampão liso cilíndrico (PNP)	15	
1	02/05/07	150	03/05/07	2619-8	22:22:19	22:22:21	Concluído	Calibrador anel roscado cilíndrico	15	
1	02/05/07	160	03/05/07	2619-8	22:22:19	22:22:21	Concluído	Calibrador anel roscado cilíndrico	20	
1	02/05/07	170	07/05/07	2619-8	13:12:02	13:12:03	Concluído	Jogo de arames para medição de roscas	40	
1	02/05/07	180	27/04/07	2970-5	13:04:18	13:04:20	Concluído	Calibrador tampão liso cilíndrico (PNP)	15	
1	02/05/07	190	27/04/07	2970-5	13:04:25	13:04:26	Concluído	Calibrador tampão liso cilíndrico (PNP)	15	
1	02/05/07	200	27/04/07	2970-5	13:04:29	13:04:30	Concluído	Calibrador anel liso cilíndrico	15	

26/10/08

Figura 20. Agendamento real realizado dos itens recebidos

Na Figura 20, temos a tela mostrando um determinado recurso, no caso “Máquina de medição longitudinal”, e os itens alocados a este recurso. Na Figura 21 o gráfico de distribuição para este mesmo recurso, dos itens alocados ao longo de um determinado período. Observa-se, que o especialista técnico ao escolher um determinado recurso, comete o erro de alocar o máximo de itens a este recurso, não balanceando entre outros recursos disponíveis que venham a executar o mesmo serviço. Outro erro ocorre, quando o especialista recebe um item e escolhe um recurso para que execute o serviço deste item; não observando se dentro de todos os itens a serem alocados, exista outro item mais urgente que este. Quando isto acontece, ou o especialista mantém o agendamento já feito ou é obrigado a refazer todo o agendamento dos itens envolvidos. Não é necessário ressaltar que retrabalho é sinônimo de perda financeira para a empresa e/ ou laboratório.



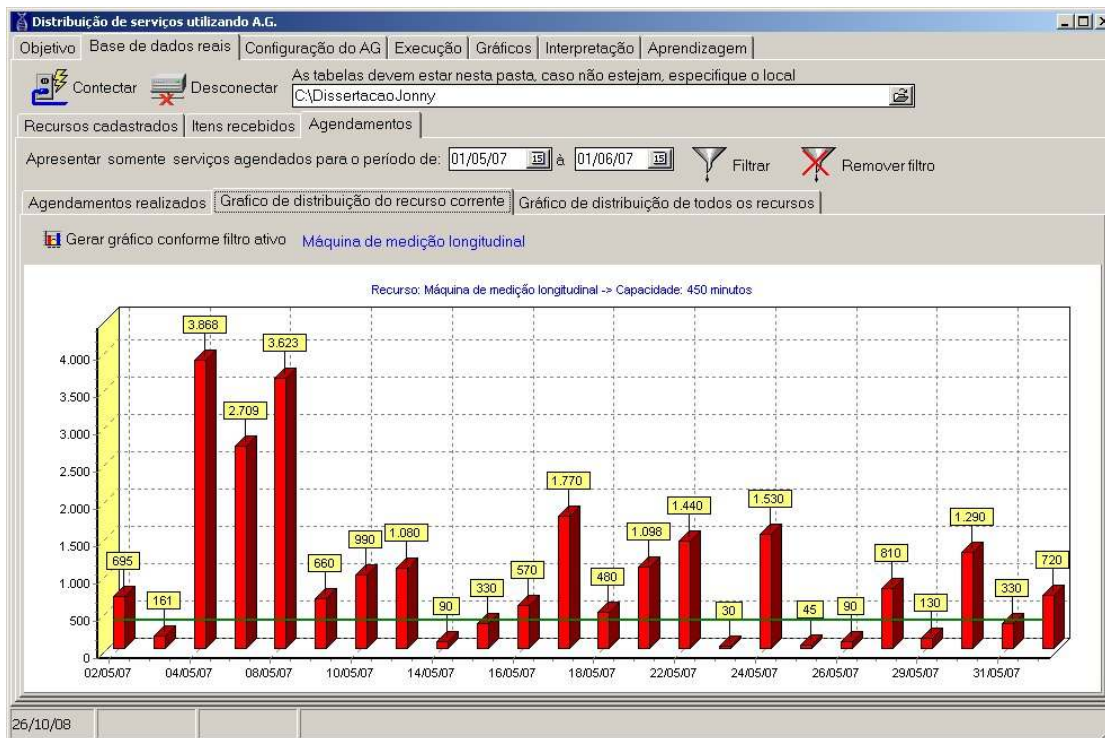


Figura 21. Gráfico de distribuição dos itens por recurso

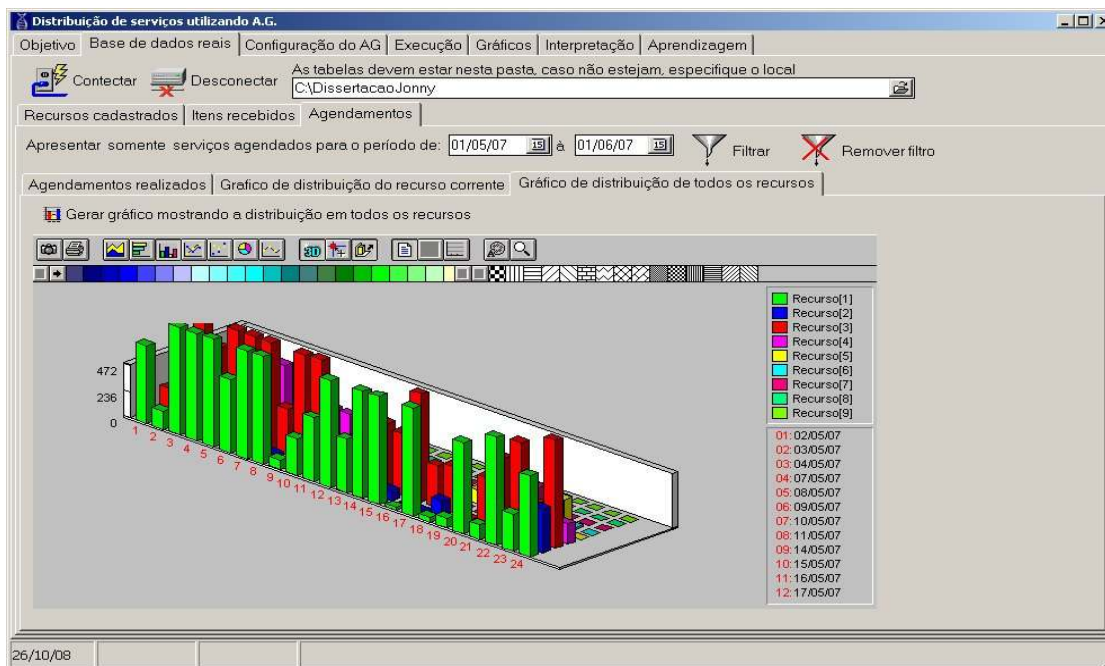


Figura 22. Gráfico de distribuição dos itens nos recursos

A tela onde são definidos os parâmetros do algoritmo genético pode ser visto na Figura 23 . A inclusão de opções como não permitir que as soluções apresentadas possuam ou não itens clones (mesmo item agendado para mais de um recurso) são vistas nesta seção do programa.

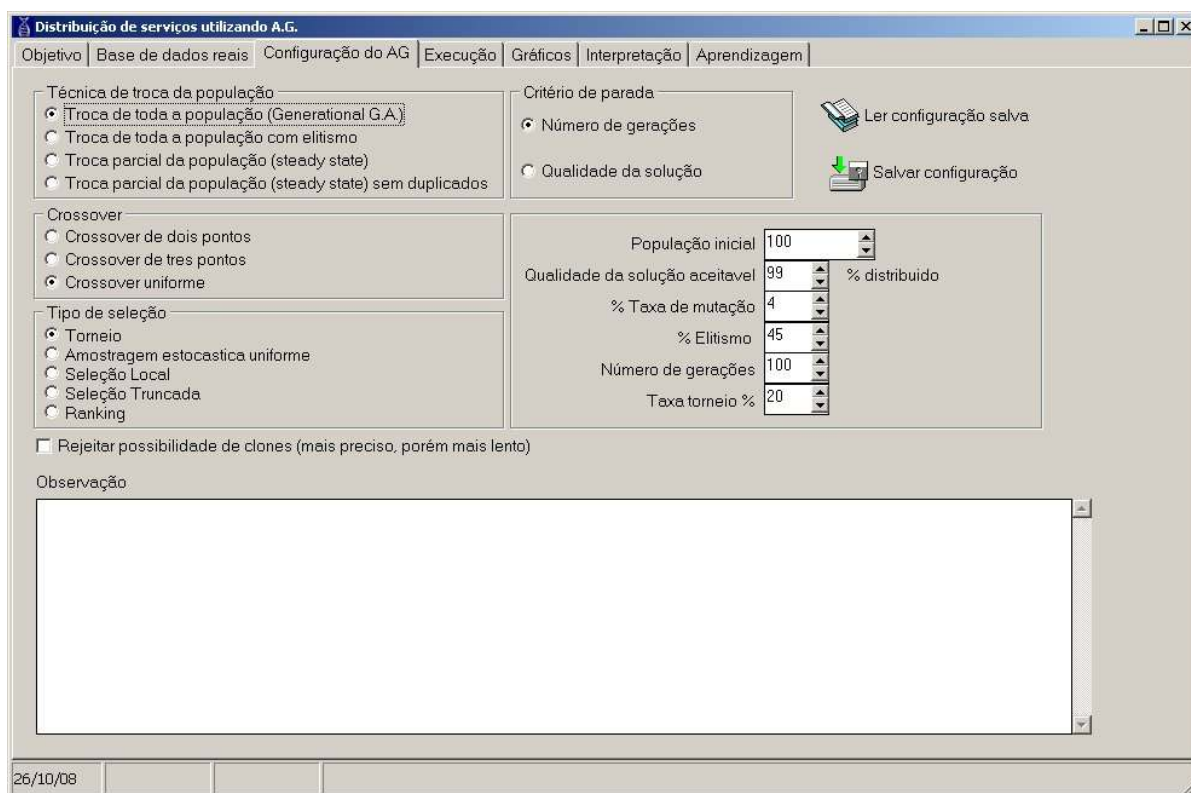


Figura 23. Configuração do AG (versão melhorada)

A tela de execução do AG responsável por distribuir os itens recebidos, vista da Figura 24, apresenta algumas opções que a título de demonstração, permaneceram no programa, como por exemplo, a opção de realizar ou não a otimização por agrupamento, a opção de interpretar os resultados da forma binária para inteligível ao usuário.

A apresentação dos resultados é organizada de tal maneira, que temos inicialmente os parâmetros que irão compor a execução e todos os resultados (cada indivíduo da população) em ordem crescente de *fitness* (do pior para o melhor). O conteúdo completo do resultado gerado pode ser observado no Apêndice A

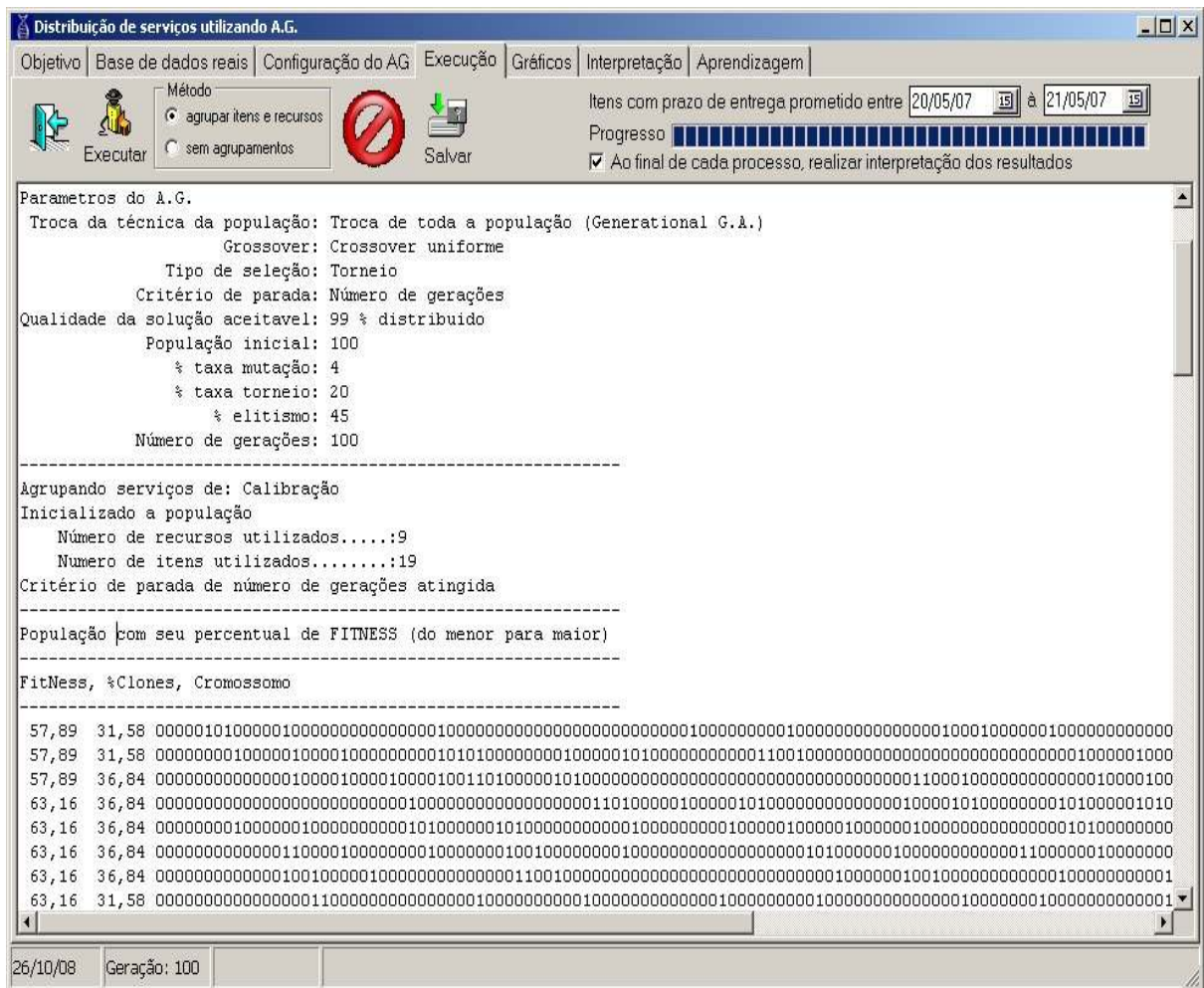


Figura 24. Tela de execução e resultados do AG

A interpretação do resultado é vista na Figura 25. Cada solução (no formato binário) encontrada é armazenada em um vetor, de modo que o usuário possa visualizar a interpretação de todas estas soluções. Para isto basta apenas selecionar a solução e clicar em um ícone ao lado para que esta solução seja interpretada e exibida ao usuário.

Nesta figura é exibida a presença de itens clones e são destacados em vermelho para que o usuário perceba de maneira bem evidenciada, quais são estes itens. Caso tenha-se selecionado para que o programa trate itens clones, então nenhum item clone é encontrado em alguma solução. O programa trata os itens clones da seguinte maneira: Na função de pontuação da população são verificadas diversas restrições e penaliza o indivíduo conforme estas restrições. Um dos índices é o

número de itens clones no cromossomo analisado. Se a opção de não permitir itens clones não estiver assinalada, então o algoritmo simplesmente penaliza e continua. Já se a opção estiver marcada, então o alelo do cromossomo é “mutado” ou seja, passa para o estado de não presente “0”, e o cromossomo é enviado de volta a população. No Apêndice B, temos os resultados obtidos quando selecionamos a opção de não permitir itens clones. No Apêndice C, temos a interpretação completa da melhor solução das soluções apresentadas do apêndice A.

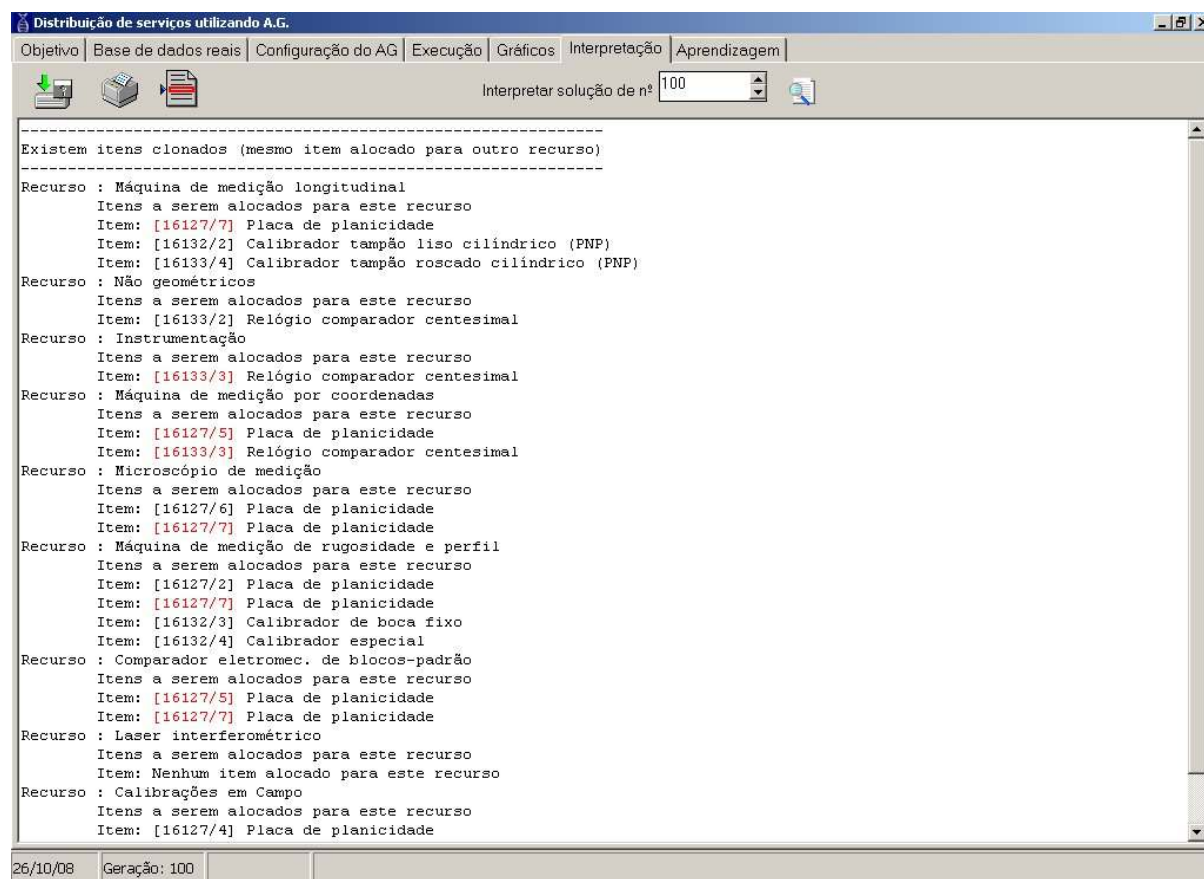


Figura 25. Resultados interpretados da forma binária para legível humana

Para facilitar ao usuário qual solução seria mais atraente ser interpretada, foi disponibilizado um gráfico que mostra a distribuição de toda a população do AG executado. Este gráfico pode ser visto na Figura 26.

Deve-se observar que neste gráfico não se consegue ver toda a população em um lance somente. É necessário o uso da barra de rolagem na parte inferior do gráfico.



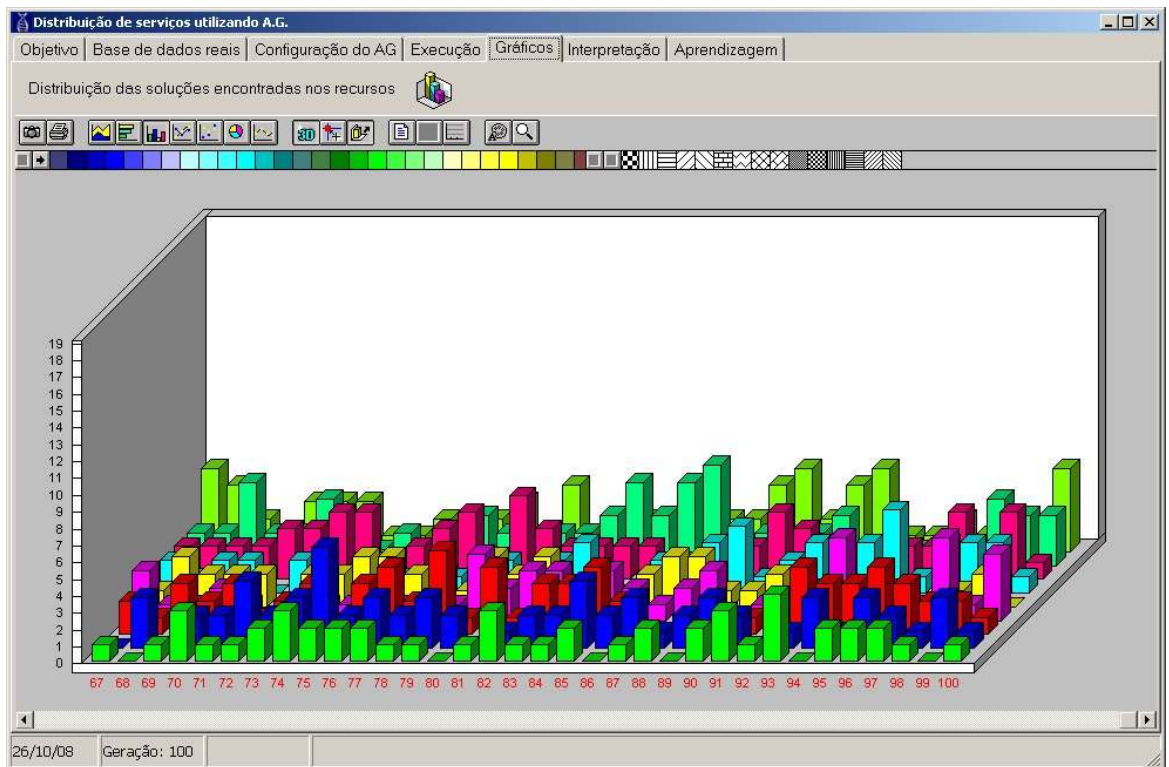


Figura 26. Gráfico da distribuição de toda a população do AG

Para facilitar ao leitor deste trabalho, foi incluído no Apêndice D, todas as figuras referente ao gráfico de distribuição de toda a população referente ao apêndice A. Desta maneira o leitor pode conferir todas as saídas geradas pelo programa implementado.

### 3.2.6 INCORPORANDO ALGORITMO DE APRENDIZAGEM

Concluído os testes, realizado o confronto do AG com a mesma solução pelo método convencional, surgiu uma dúvida que culminou com a idéia da criação de uma nova versão, mais aperfeiçoada e tecnicamente mais completa. “Qual a melhor combinação de parâmetros no AG?”, esta pergunta realmente faz sentido, já que a quantidade de combinações e seus valores tornam quase infinitas as possibilidades. Usar 0,4 ou 12% de mutação? Usar Elitismo ou não, se sim qual percentual?, Qual a taxa de torneio?, que tipo de *crossover* usar?

Sentiu-se a necessidade de que o AG aprendesse os melhores parâmetros. Construiu-se então um novo AG responsável unicamente para promover a melhor combinação destes parâmetros. Passamos a ter um algoritmo genético dentro de outro algoritmo genético. O AG de aprendizagem

iria para cada combinação encontrada, executar uma chamada para o AG encarregado de realizar a distribuição dos itens nos recursos disponíveis. Este AG de aprendizagem armazenaria a melhor combinação encontrada.

O critério de parada do AG de aprendizagem é o numero de interações realizadas, porém dentro de cada AG permanece o critério deste definido conforme especificado nos parâmetros do AG ou seja ou numero de interações ou qualidade da solução atingida.

Obviamente a performance do sistema ficaria inviável se realizássemos sempre a execução do AG de aprendizagem. Então se estabeleceu que a aprendizagem fosse uma opção do sistema para em intervalos de tempos realizar a “calibração” dos parâmetros. Assim, armazenaria se os melhores parâmetros para tempos de muito serviço (muitos itens recebidos) e para poucos serviços ou qualquer outra situação que surgisse.

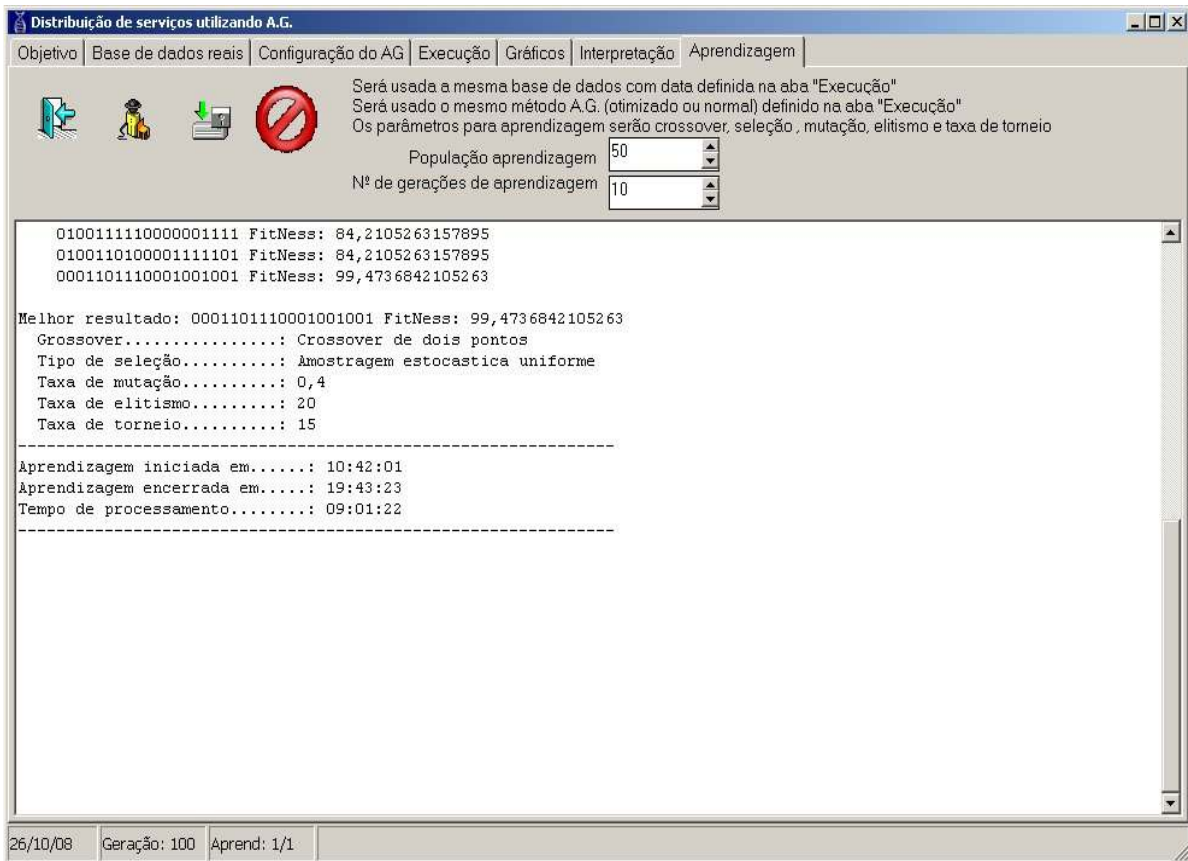


Figura 27. Tela de execução do AG de aprendizagem

Na Figura 27 é apresentada a tela com a execução do AG de aprendizagem. Os parâmetros avaliados são *crossover*, tipo de seleção, % mutação, % elitismo, % torneio.

Inicialmente na construção do AG de aprendizagem, implementou-se utilizando algumas restrições nos valores máximos que os parâmetros avaliados poderiam receber.

Estas restrições são as seguintes:

- Taxa de mutação não poderá ser maior que 63%
- Taxa de torneio não poderá ser superior a 63%
- Taxa de elitismo não poderá ser superior a 63%

Alguns destes valores foram imputados através de leituras encontradas em artigos e livros sobre algoritmos genéticos.

Outra razão de impor estes valores máximos é o cromossomo utilizado (mais adiante é descrito o cromossomo de aprendizagem), 6 (seis) caracteres permitem variar de “000000” até “111111” ou seja de 0 a 63 em decimal. O uso de 7 (sete) caracteres passaria do valor 100 em decimal o que acarretaria em implementar um algoritmo especial para limitar em 100, aumentando assim o tempo de processamento, sem mencionar que um cromossomo muito grande iria implicar num tempo de processamento também proporcional.

No capítulo final é recomendado que se efetue a inclusão de mais parâmetros e eliminar a restrição da taxa máxima de 63%.

A função de avaliação do AG de aprendizagem pega o melhor *fitness* do AG primário.

O cromossomo do AG de aprendizagem é binário e é composto por uma string de 22 caracteres (somente “0” ou “1”) dividida em 5 grupos.

- Crossover - 2 caracteres
- Seleção - 2 caracteres
- Mutação - 6 caracteres
- Elitismo - 6 caracteres
- Torneio - 6 caracteres

Ao executar o AG de aprendizagem algumas observações devem ser levadas em consideração. Não executar um número de gerações muito elevado, pois irá demorar muito. Se o AG primário tiver 1000 gerações e o de aprendizagem 1000, significa que serão executadas 1000.000 gerações no AG primário.

### **3.3 RESUMO E COMENTÁRIOS**

Apresentou-se um breve resumo do desenvolvimento do algoritmo genético que trate o problema da distribuição dos itens em um laboratório de metrologia; desde o esqueleto inicial até o algoritmo genético auxiliar de aprendizagem. Como resultado, temos assim um aplicativo final que abrange as seguintes possibilidades e procedimentos.

- Uma área de entrada dos itens reais e os recursos disponíveis para serem utilizados no processo de distribuição destes itens recebidos. Esta área de entrada mantém todo um histórico de recebimentos e distribuições realizadas.

- Visualizar graficamente a situação dos recursos quanto ao volume de serviços já alocados, tanto individualmente (recurso a recurso) ou todos os recursos simultaneamente.
- Filtros para visualizar apenas datas (de agendamento) desejadas pelo usuário.
- Uma área de configuração dos parâmetros do AG, onde pode-se salvar a configuração para facilitar o usuário em seu dia a dia. O aplicativo ao iniciar, automaticamente lê a última configuração salva.
- Uma área de execução do AG, onde defini-se o período desejado e de que forma o AG deve se comportar quanto a preparar os itens para serem inseridos dentro da população e seu sequenciamento (com ou sem agrupamento, com ou sem itens clones) .
- Uma área onde as soluções apresentadas possam ser convertidas do formato binário para um formato legível e facilmente interpretado pelo usuário.
  - Permitir interagir com o usuário de modo que possa analisar todas as soluções apresentadas, inclusive as reprovadas pelo AG.
  - Permitir a visualização gráfica de cada resultado da população final.
- Ter uma área que pode ser acionada conforme interesse do usuário, onde o próprio aplicativo sugira os parâmetros do AG que retornem uma maior quantidade e qualidade nas soluções a serem apresentadas (executa-se um AG auxiliar que irá aleatoriamente variar alguns parâmetros do AG principal e executá-lo, armazenando as melhores configurações e sugerindo ao final a melhor qualificada).

### **3.4 SEQUÊNCIA DE AÇÕES DO ALGORITMO GENÉTICO**

Quanto ao algoritmo genético principal, vamos chamar assim, ao ser executado, procederá a sequência de ações da seguinte maneira conforme apresentado na Figura 28. Esta sequência, esta obviamente de forma simples e reduzida. Para cada item da sequência existem funções auxiliares, de conversão, ordenamento, classificação, detalhes na função de avaliação e acesso a dados (valor do alelo no cromossomo).

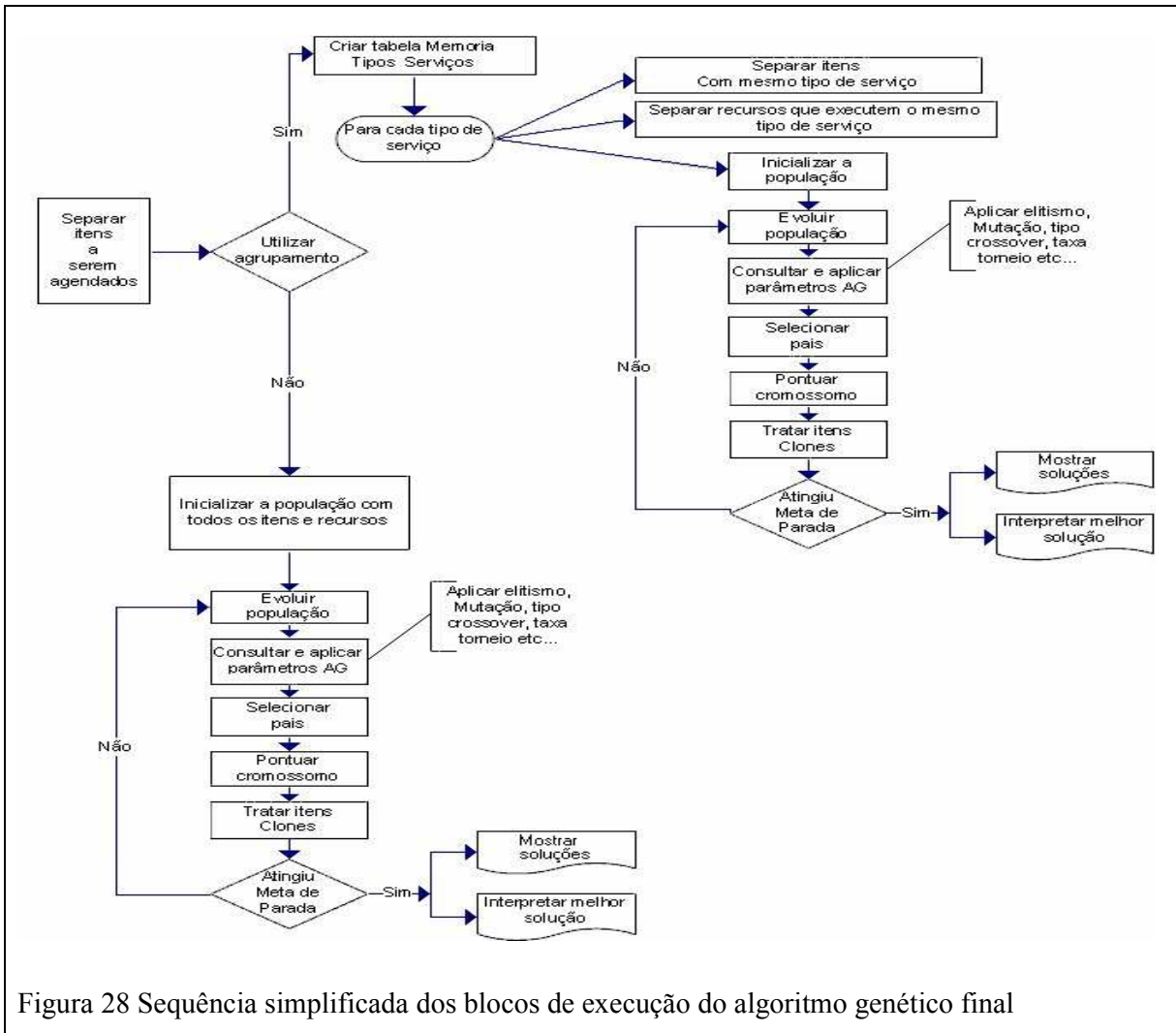


Figura 28 Sequência simplificada dos blocos de execução do algoritmo genético final

O bloco principal, executado na aba do aplicativo onde se dá o *start* do programa, inicialmente separa todos os itens a serem agendados conforme o período informado pelo usuário.

Deste ponto em diante todos estes itens, recursos e tabelas auxiliares são inseridos em tabelas na memória para agilizar acessos.

Na sequência, o programa verifica se o usuário quer otimizar, realizando agrupamento ou não.

Se sim, é posicionado um cursor na tabela de serviços e para cada tipo de serviço, e para cada registro agrupa os itens e recursos com o mesmo tipo de serviço e executa o AG para este grupo.

Se não, então simplesmente todos os itens marcados para serem agendados e recursos disponíveis estarão misturados e executa-se um único AG com todos estes itens e recursos.

Na mesma Figura 28, temos repetido o bloco do AG propriamente dito. É proposital apenas para facilitar a explicação do sequenciamento dos blocos.

As evoluções realizadas, partindo-se desde o esqueleto do programa até a implementação do algoritmo auxiliar de aprendizagem, apresentam a conclusão de que o potencial do algoritmo em implementação ainda não foi atingido e pode ser substancialmente melhorado, incorporando mais técnicas de AG dentro da estrutura do programa.

## 4 TESTES E RESULTADOS

Durante o decorrer da dissertação, inúmeros testes foram realizados. Os primeiros, envolvendo apenas dados fictícios formaram o esqueleto do programa em desenvolvimento. A medida que o programa foi evoluindo em suas versões foram inseridos dados reais, com uso de um volume também real de informações.

Faz-se agora necessário analisar os resultados encontrados durante a fase de testes com intuito de avaliar o procedimento implementado e a modelagem proposta.

### 4.1 METODOLOGIA USADA PARA OS TESTES

Construíram-se algumas tabelas contendo resultados das execuções de cada teste realizado, utilizando sempre uma mesma configuração para AG. Não foi preocupação trabalharmos na configuração do AG, uma vez que implementou-se uma rotina específica para obter a melhor configuração (AG de aprendizagem). O que se fez, foi racionalmente, pelos testes já realizados e apresentados nas versões anteriores, analisar o modelo em si e a performance. Testes mais apurados, envolvendo variação nos parâmetros do AG devem ser estimulados, o que provavelmente representariam apenas pequenos ajustes de performance.

Configuração do AG usada nos testes:

- Troca de toda a população com elitismo (20%);
- *Crossover* de 2 pontos;
- Tipo seleção dos pais, por torneio (20%);
- Critério de parada: número de gerações ;
- Número de gerações com técnica de agrupamento: 1000;
- Número de gerações sem técnica de agrupamento 100;
- População inicial 100 indivíduos;
- Taxa de mutação 2%.

Os dados (itens e recursos) foram os itens recebidos e com prazo de entrega para período de 01/05/2007 a 31/05/2007 perfazendo assim diversas situações diferentes com quantidades de itens



diferentes em cada situação para serem distribuídos entre os nove recursos disponíveis. Optou-se por este período porque teríamos como posteriormente confrontar as soluções propostas pelo AG com os dados de como haviam estes mesmos itens sido distribuídos.

Uma observação importante deve ser colocada, o número de gerações utilizando a técnica de agrupamento de itens e recursos, é de 1000 gerações enquanto que sem o uso desta técnica, passou-se para 100 gerações. A razão desta alteração é devido ao tempo de demora para levantamento de todas as informações para o período de pesquisa.

Com estes dados e parâmetros realizamos os seguintes testes:

- Uso de agrupamento com clones
- Uso de agrupamento sem clones
- Sem agrupamento com clones
- Sem agrupamento sem clones

## 4.2 RESULTADOS

Os resultados obtidos são apresentados em uma série de tabelas, separadas por tipo de teste. Lembrando que estes resultados, como mencionado, estão especificamente amarrados com a configuração dos parâmetros do AG especificada anteriormente.

Legenda usada nas tabelas de resultados dos testes

- Dia = Dia no período definido para testes
- R1 a R9 = N° de itens alocados pelo especialista para o recurso de mesmo número
- G1 a G9 = N° de itens alocados pelo AG para o recurso de mesmo número
- Soluções = N° de soluções satisfatórias encontradas pelo AG ( *Fitness* > 80% )
- Tempo = Tempo que o AG levou para encontrar as soluções
- % Clones = Média do percentual de clones existentes nas soluções
- TR = Quantidade de itens agendados pelo especialista no dia
- TG = Quantidade de itens agendados pelo AG no dia

A coluna “Soluções”, existente nas tabelas de resultados dos testes realizados no período definido, significa a quantidade de soluções que o AG encontrou para agendar os itens nos recursos

disponibilizados para aquela data que tenham atingido um percentual superior a 80% . A quantidade de itens distribuídos na melhor destas solução é encontrada na coluna “TG” , enquanto que na coluna “TR” temos a quantidade de itens distribuídos pelo especialista para a mesma data

Tabela 5. Resultados dos testes usando agrupamento com clones

Dia	R1	G1	R2	G2	R3	G3	R4	G4	R5	G5	R6	G6	R7	G7	R8	G8	R9	G9	TR	TG	%Clones	Soluções	Tempo (seg.)
1																						Feriado	
2	15	6	0	6	32	7	2	5	0	8	0	7	0	8	0	1	0	4	49	52	18,87	1	29
3	43	6	0	7	28	6	4	7	0	6	0	9	0	6	0	8	0	13	75	68	18,57	1	31
4	31	0	2	0	6	2	2	0	4	1	0	2	1	2	0	0	0	1	46	8	13,50	16	6
5																						Sábado	
6																						Domingo	
7	21	13	0	9	26	7	6	14	0	8	0	12	0	11	0	13	0	10	53	97	21,50	0	23
8	63	8	0	7	48	6	2	7	5	7	0	11	0	5	0	7	0	4	118	62	20,49	0	71
9	71	15	11	19	65	20	2	9	1	15	0	17	0	12	0	17	0	14	150	138	21,65	0	47
10	39	4	0	0	38	3	1	4	6	4	0	2	1	2	0	6	0	3	85	28	18,75	1	31
11	75	5	9	1	43	1	12	2	0	4	3	4	0	4	0	2	0	2	142	25	17,42	0	76
12																						Sábado	
13																						Domingo	
14	9	8	0	7	29	7	4	7	2	10	0	10	0	2	0	5	0	10	44	66	16,13	5	18
15	17	0	0	1	17	3	0	1	2	5	2	5	0	1	0	2	0	1	38	19	14,81	3	13
16	16	7	1	6	36	1	0	4	0	10	0	5	0	9	0	9	0	6	53	57	12,28	2	25
17	4	14	3	13	31	7	2	12	0	11	1	9	2	10	0	11	0	14	43	101	16,13	9	17
18	28	10	1	10	33	10	15	10	1	16	1	7	3	7	0	11	0	18	82	99	20,90	0	43
19																						Sábado	
20																						Domingo	
21	10	3	0	3	9	3	3	2	0	2	0	3	0	2	0	1	0	3	22	22	13,79	3	13
22	40	5	0	2	40	1	3	4	0	2	0	2	0	2	0	8	0	4	83	30	17,24	1	33
23	58	2	14	0	46	0	0	3	0	1	0	1	0	1	0	2	0	3	118	13	22,55	0	58
24	67	0	11	0	77	0	1	0	1	0	0	0	0	0	0	0	0	0	157	0	18,81	0	57
25	1	6	1	6	8	7	0	5	0	8	0	7	0	8	0	1	0	4	10	52	16,67	4	12
26																						Sábado	
27																						Domingo	
28	19	7	0	6	10	10	2	6	8	8	1	6	1	2	0	4	0	4	41	53	16,67	5	12
29	2	13	0	9	22	21	0	14	0	12	1	15	0	12	0	14	0	10	25	120	16,48	2	14
30	19	13	5	9	12	7	5	14	1	8	0	12	0	11	0	13	0	10	42	97	17,31	3	20
31	0	8	0	7	27	6	1	7	0	7	0	11	0	5	0	7	0	4	28	62	14,29	4	8

Tabela 6. Resultados dos testes usando agrupamento sem clones

Dia	R1	G1	R2	G2	R3	G3	R4	G4	R5	G5	R6	G6	R7	G7	R8	G8	R9	G9	TR	TG	%Clones	Soluções	Tempo (seg.)
1																						Feriado	
2	15	4	0	7	32	6	2	7	0	6	0	9	0	13	0	0	0	0	49	52	0	26	48
3	43	9	0	10	28	14	4	6	0	16	0	5	0	9	0	0	0	0	75	69	0	20	66
4	31	0	2	5	6	0	2	1	4	2	0	2	1	0	0	0	0	0	46	10	0	38	5
5																						Sábado	
6																						Domingo	
7	21	8	0	5	26	8	6	6	0	5	0	13	0	7	0	0	0	0	53	52	0	20	50
8	63	23	0	15	48	13	2	9	5	15	0	18	0	28	0	0	0	0	118	121	0	19	112
9	71	14	11	13	65	14	2	15	1	15	0	11	0	14	0	0	0	0	150	96	0	20	93
10	39	7	0	11	38	10	1	13	6	6	0	4	1	12	0	0	0	0	85	63	0	16	57
11	75	22	9	12	43	20	12	24	0	22	3	18	0	20	0	0	0	0	142	138	0	20	128
12																						Sábado	
13																						Domingo	
14	9	5	0	5	29	2	4	5	2	8	0	3	0	2	0	0	0	0	44	30	0	70	26
15	17	2	0	3	17	6	0	3	2	5	2	5	0	2	0	0	0	0	38	26	0	20	20
16	16	5	1	6	36	4	0	4	0	5	0	8	0	6	0	0	0	0	53	38	0	21	36
17	4	2	3	5	31	5	2	3	0	8	1	1	2	6	0	0	0	0	43	30	0	20	28
18	28	10	1	7	33	8	15	9	1	8	1	10	3	14	0	0	0	0	82	66	0	19	32
19																						Sábado	
20																						Domingo	
21	10	2	0	4	9	2	3	4	0	2	0	1	0	3	0	0	0	0	22	18	0	22	8
22	40	8	0	8	40	8	3	10	0	8	0	10	0	5	0	0	0	0	83	57	0	97	54
23	58	8	14	21	46	13	0	10	0	21	0	15	0	13	0	0	0	0	118	101	0	89	99
24	67	15	11	12	77	18	1	10	1	14	0	16	0	15	0	0	0	0	157	100	0	100	96
25	1	5	1	3	8	1	0	5	0	1	0	5	0	3	0	0	0	0	10	23	0	100	13
26																						Sábado	
27																						Domingo	
28	19	1	0	3	10	2	2	3	8	4	1	3	1	1	0	0	0	0	41	17	0	38	18
29	2	6	0	5	22	1	0	0	0	2	1	4	0	4	0	0	0	0	25	22	0	23	18
30	19	6	5	7	12	5	5	5	1	3	0	0	0	4	0	0	0	0	42	30	0	20	28
31	0	2	0	3	27	2	1	1	0	1	0	1	0	3	0	0	0	0	28	13	0	20	10

Tabela 7. Resultados dos testes sem agrupamento com clones

Dia	R1	G1	R2	G2	R3	G3	R4	G4	R5	G5	R6	G6	R7	G7	R8	G8	R9	G9	TR	TG	%Clones	Soluções	Tempo (min.)
1																						Feriado	
2	15	3	0	4	32	6	2	2	0	10	0	6	0	7	0	7	0	6	49	51	16,98	2	11:45
3	43	8	0	12	28	7	4	5	0	11	0	6	0	6	0	5	0	12	75	72	19,18	0	17:58
4	31	2	2	2	6	0	2	1	4	0	0	1	1	2	0	1	0	1	46	10	15,91	12	01:51
5																						Sábado	
6																						Domingo	
7	21	9	0	7	26	7	6	3	0	5	0	5	0	3	0	8	0	6	53	53	18,52	0	11:57
8	63	14	0	18	48	15	2	15	5	8	0	14	0	14	0	16	0	7	118	121	19,35	0	38:17
9	71	13	11	11	65	13	2	12	1	8	0	11	0	11	0	10	0	6	150	95	19,59	1	26:37
10	39	6	0	7	38	7	1	3	6	8	0	5	1	10	0	6	0	11	85	63	20,31	1	14:58
11	75	16	9	13	43	18	12	14	0	19	3	11	0	14	0	19	0	14	142	138	22,30	0	45:26
12																						Sábado	
13																						Domingo	
14	9	3	0	3	29	4	4	3	2	1	0	4	0	5	0	4	0	4	44	31	18,06	5	06:00
15	17	2	0	1	17	4	0	3	2	2	2	3	0	4	0	3	0	3	38	25	15,43	6	05:05
16	16	5	1	4	36	5	0	0	0	4	0	8	0	6	0	3	0	4	53	39	20,51	3	07:55
17	4	2	3	3	31	5	2	0	0	1	1	3	2	3	0	9	0	4	43	30	18,75	0	06:16
18	28	6	1	8	33	7	15	10	1	8	1	7	3	6	0	7	0	9	82	68	18,57	0	16:54
19																						Sábado	
20																						Domingo	
21	10	0	0	3	9	4	3	1	0	1	0	2	0	3	0	1	0	3	22	18	15,79	3	03:25
22	40	7	0	3	40	9	3	3	0	5	0	11	0	8	0	6	0	5	83	57	18,97	3	13:08
23	58	14	14	9	46	15	0	14	0	9	0	8	0	11	0	8	0	13	118	101	20,59	1	28:45
24	67	10	11	12	77	17	1	17	1	12	0	7	0	5	0	14	0	7	157	101	18,81	2	28:16
25	1	1	1	5	8	4	0	2	0	6	0	0	0	1	0	0	0	4	10	23	16,67	3	04:28
26																						Sábado	
27																						Domingo	
28	19	4	0	1	10	4	2	4	8	1	1	2	1	1	0	0	0	0	41	17	11,11	3	03:14
29	2	4	0	2	22	4	0	2	0	1	1	3	0	3	0	1	0	1	25	21	14,78	5	04:13
30	19	2	5	3	12	4	5	3	1	5	0	3	0	3	0	4	0	4	42	31	17,05	7	06:00
31	0	1	0	2	27	1	1	0	0	3	0	0	0	2	0	1	0	3	28	13	11,90	6	02:25

Tabela 8. Resultados dos testes sem agrupamento sem clones

Dia	R1	G1	R2	G2	R3	G3	R4	G4	R5	G5	R6	G6	R7	G7	R8	G8	R9	G9	TR	TG	%Clones	Soluções	Tempo (min.)
1																				0		Feriado	
2	15	9	0	9	32	9	2	7	0	5	0	5	0	8	0	0	0	0	49	52	0	13	11:45
3	43	11	0	6	28	13	4	13	0	11	0	3	0	15	0	0	0	0	75	72	0	0	18:02
4	31	2	2	2	6	1	2	3	4	0	0	0	1	2	0	0	0	0	46	10	0	26	01:52
5																				0		Sábado	
6																				0		Domingo	
7	21	15	0	14	26	15	6	18	0	12	0	14	0	8	0	0	0	0	53	96	0	13	12:06
8	63	10	0	8	48	11	2	12	5	9	0	9	0	4	0	0	0	0	118	63	0	1	38:20
9	71	22	11	21	65	13	2	17	1	27	0	21	0	17	0	0	0	0	150	138	0	6	26:52
10	39	5	0	3	38	3	1	5	6	4	0	7	1	3	0	0	0	0	85	30	0	10	15:07
11	75	3	9	1	43	5	12	3	0	7	3	3	0	4	0	0	0	0	142	26	0	2	45:32
12																				0		Sábado	
13																				0		Domingo	
14	9	8	0	10	29	11	4	8	2	11	0	9	0	12	0	0	0	0	44	69	0	29	06:06
15	17	3	0	1	17	2	0	3	2	1	2	6	0	2	0	0	0	0	38	18	0	22	05:10
16	16	10	1	9	36	12	0	5	0	5	0	9	0	7	0	0	0	0	53	57	0	18	08:02
17	4	9	3	18	31	15	2	24	0	12	1	12	2	11	0	0	0	0	43	101	0	0	06:18
18	28	14	1	15	33	14	15	17	1	15	1	17	3	8	0	0	0	0	82	100	0	0	16:57
19																				0		Sábado	
20																				0		Domingo	
21	10	4	0	4	9	2	3	2	0	4	0	1	0	5	0	0	0	0	22	22	0	22	03:24
22	40	5	0	2	40	7	3	5	0	5	0	3	0	3	0	0	0	0	83	30	0	10	13:09
23	58	2	14	4	46	2	0	2	0	1	0	2	0	0	0	0	0	0	118	13	0	6	28:41
24	67	0	11	0	77	0	1	0	1	0	0	0	0	0	0	0	0	0	157	0	0	6	28:23
25	1	9	1	9	8	9	0	7	0	5	0	5	0	8	0	0	0	0	10	52	0	24	04:28
26																				0		Sábado	
27																				0		Domingo	
28	19	6	0	9	10	6	2	8	8	8	1	7	1	9	0	0	0	0	41	53	0	28	03:12
29	2	11	0	22	22	15	0	19	0	12	1	12	0	18	0	0	0	0	25	109	0	23	04:14
30	19	15	5	14	12	15	5	18	1	12	0	14	0	8	0	0	0	0	42	96	0	18	06:01
31	0	10	0	8	27	11	1	12	0	9	0	9	0	4	0	0	0	0	28	63	0	64	02:26

Nas Figuras 30 a 34, temos uma série de gráficos representando os testes realizados. Para melhor interpretar o gráfico, a Figura 29 apresenta a explicação (legenda) destes gráficos.

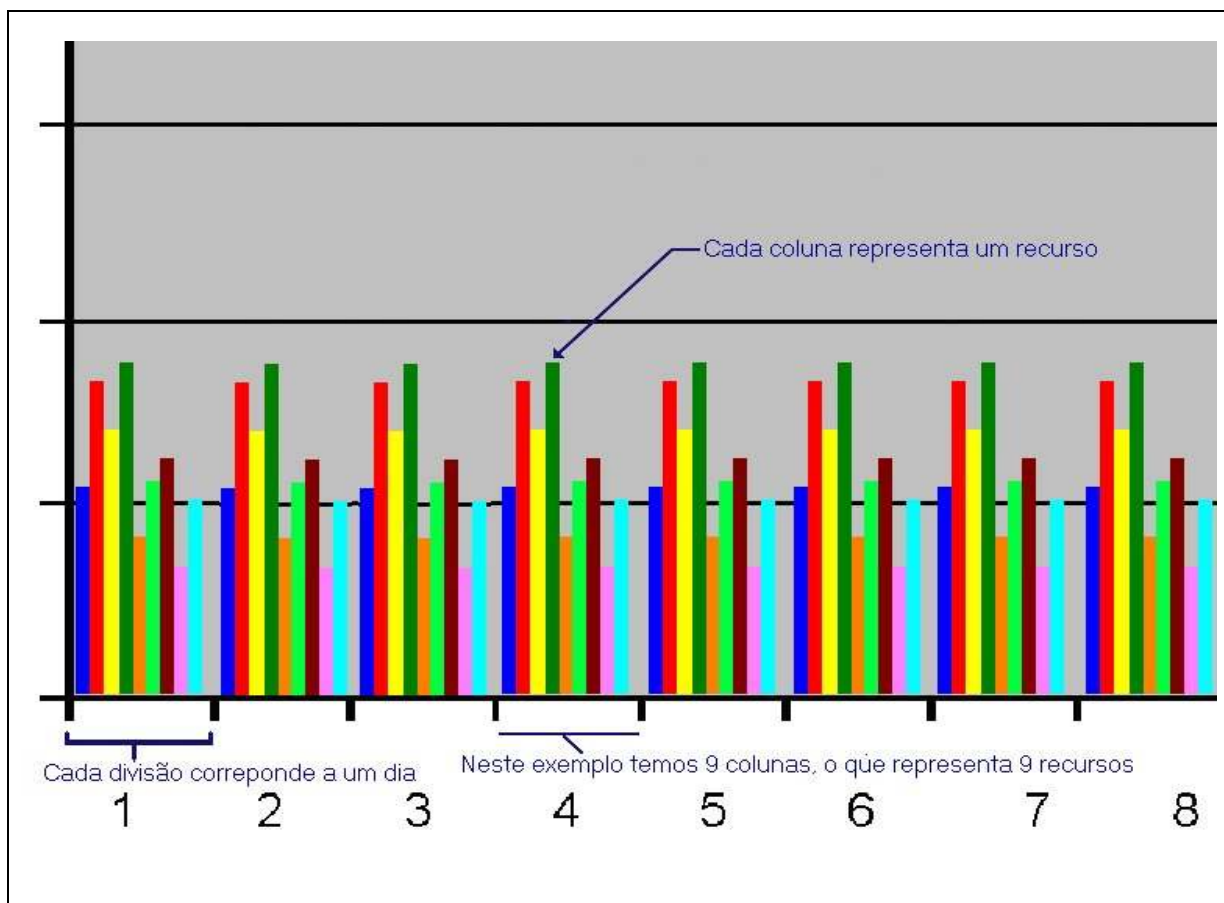


Figura 29. Interpretação dos objetos nos gráficos 30 a 34

No eixo X, temos os testes realizados. Cada identificação no eixo X corresponde ao dia no mês do teste. No eixo Y temos a quantidade de itens alocados a um recurso. Cada fração do eixo X, que representa um dia, possui 9 colunas, onde cada coluna representa um recurso (recurso 1 ao recurso 9) que nas tabelas anteriores, são representadas pelas colunas R1 a R9 ou G1 a G9. No eixo X, as datas 1, 5 e 6, 12 e 13, 19 e 20, 26 e 27 não possuem valores por ser um sábado e domingo, dias em que o laboratório em questão não exercia atividades. A idéia destes gráficos, é mostrar apenas a distribuição e quantidades máximas. A interpretação dos valores de maneira mais acurada deverá ser feita nas tabelas de onde estes gráficos foram obtidos (Tabela 5 a Tabela 8).

Cabe aqui lembrar que o especialista em questão muitas vezes é um funcionário de carreira que por experiência adquirida passa a realizar a tarefa de agendamento dos itens recebidos. A

contratação de um especialista com capacitação técnica é muito onerosa e muitas vezes opta-se por empregá-lo em tarefas mais nobres como a realização do serviço propriamente dita.

A representação gráfica da distribuição para o período de testes feito pelo especialista pode ser vista na Figura 30. Esta distribuição será confrontada com os gráficos de distribuição dos outros testes realizados.

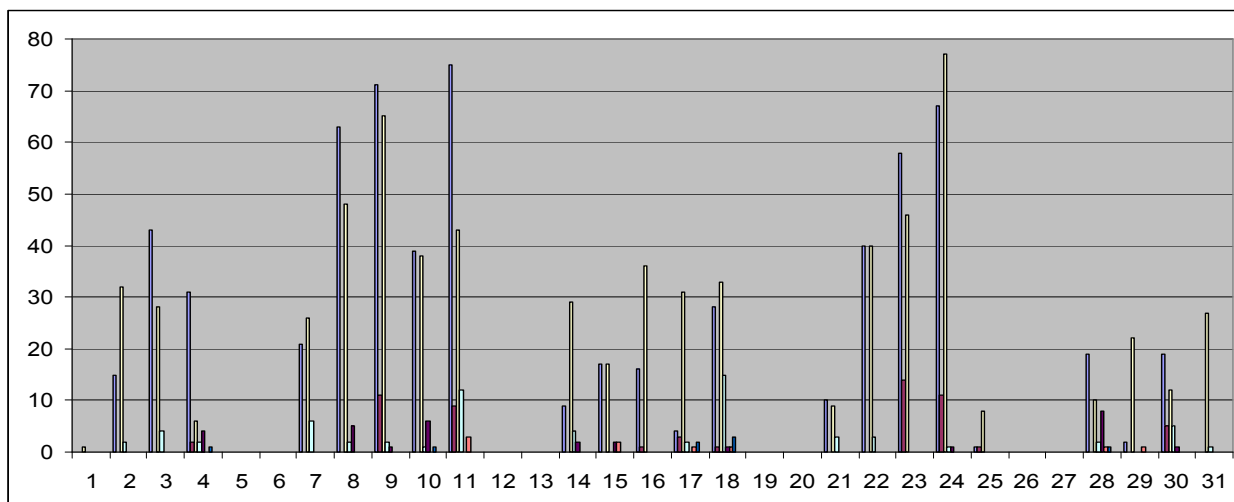


Figura 30. Distribuição realizada pelo especialista

A Figura 31 representa a distribuição realizada pelo AG no teste com uso de agrupamento, sem o tratamento de clones.

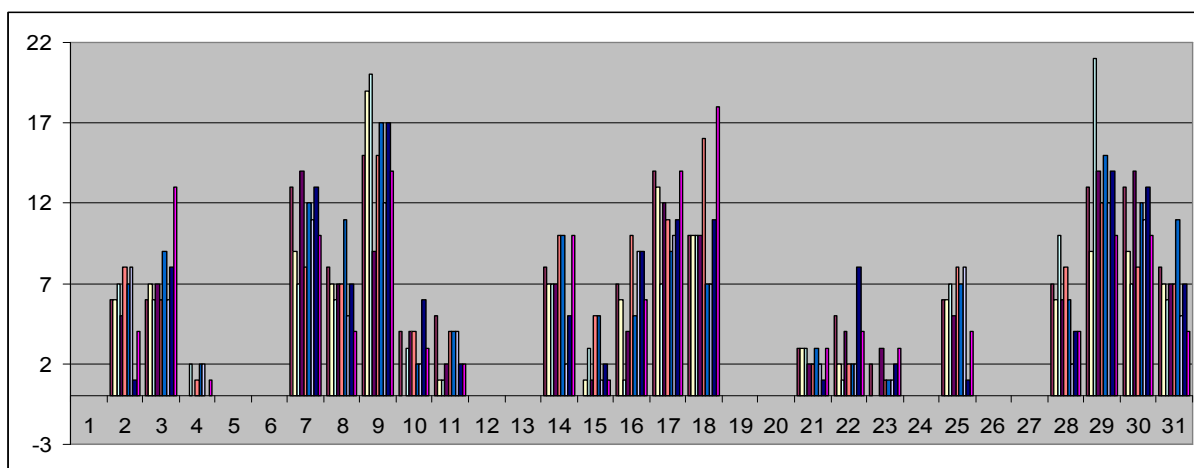


Figura 31. Distribuição do AG. Com agrupamento, sem clones



A Figura 32 representa a distribuição realizada pelo AG no teste com uso de agrupamento, com o tratamento de clones.

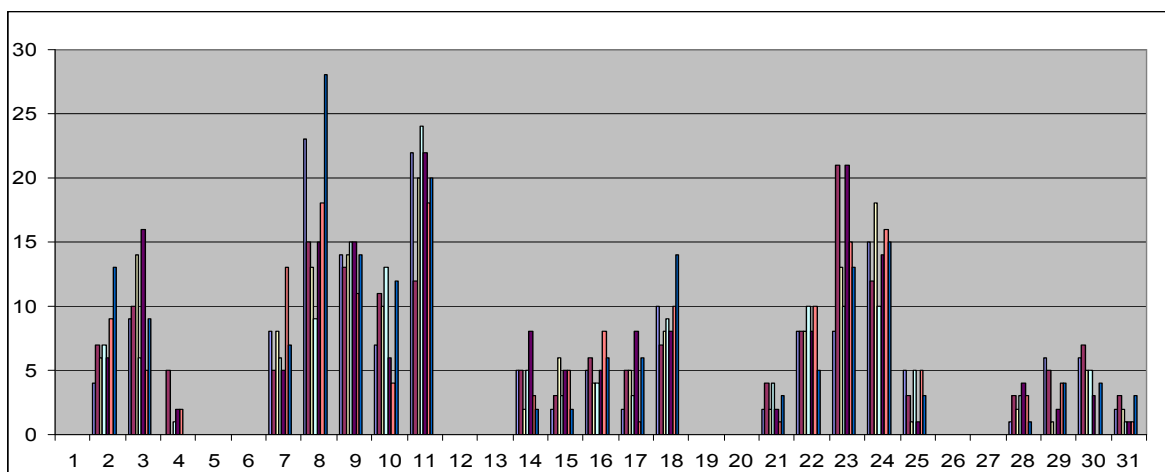


Figura 32. Distribuição do AG. Com agrupamento, com clones

A Figura 33 representa a distribuição realizada pelo AG no teste sem o uso de agrupamento, e sem o tratamento de clones.

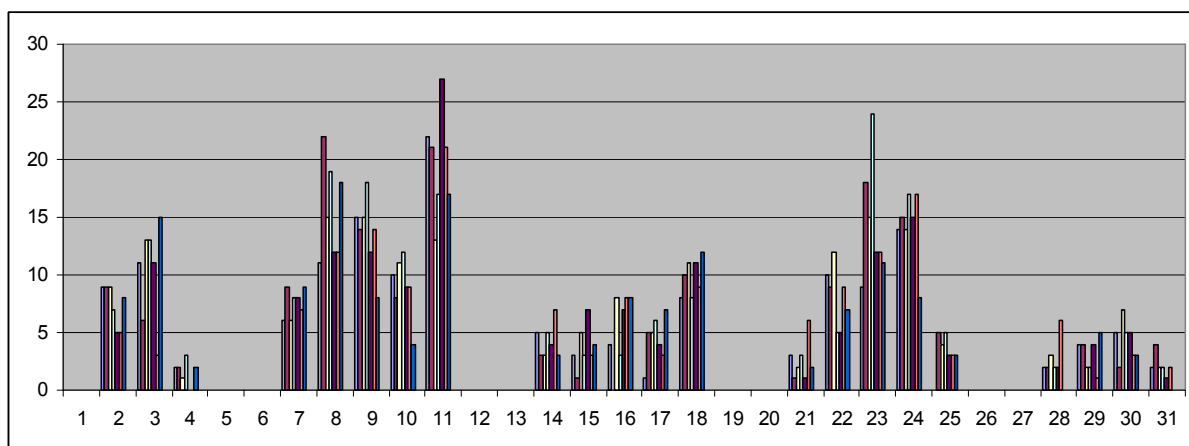


Figura 33. Distribuição do AG. Sem agrupamento, sem clones

A Figura 34 representa a distribuição realizada pelo AG no teste sem o uso de agrupamento, e com o tratamento de clones.

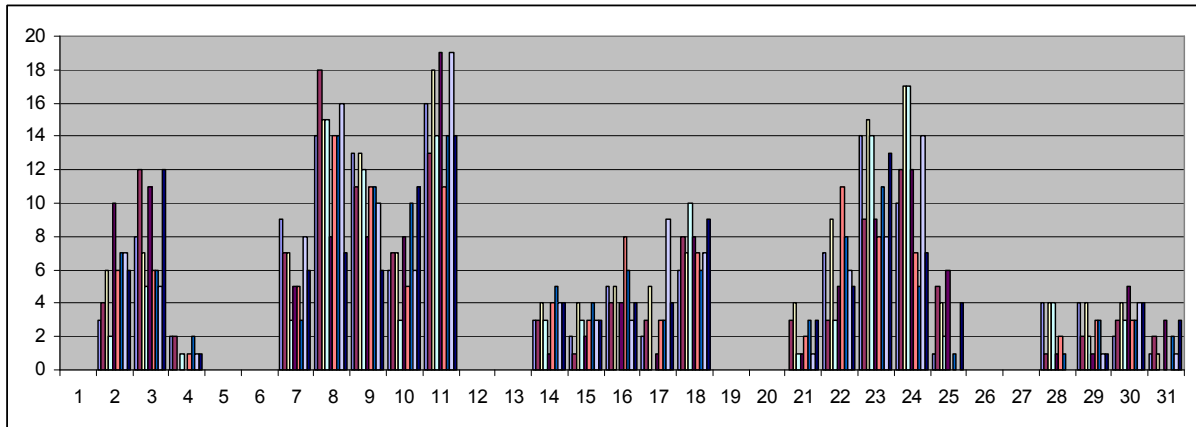


Figura 34. Distribuição do AG. Sem agrupamento, com clones

Interpretando os resultados, presume-se como que a opção de agrupamento dos itens e recursos e a técnica de tratar o surgimento de clones, é a melhor opção do AG implementado, devendo provavelmente ser a metodologia a ser adotada dentro de um aplicativo comercial, devido a maior quantidade de número de soluções e tempo de execução do AG. Embora todos os indivíduos dentro do espaço de busca sejam soluções, define-se aqui que solução é ter atingido um percentual de distribuição aceitável, preferencialmente 100%, e que quando não atingido este percentual, permita ao especialista o distribuição manual dos itens faltantes.

A opção sem agrupamento é inviável devido a baixa performance em termos de tempo de execução.

Pode-se verificar que a distribuição do agendamento realizado pelo AG está mais homogêneo e também foi respeita a capacidade de trabalho do recurso.

Convém lembrar que nos gráficos e testes onde não se faz uso do tratamento de clones, os dados alocados pelo AG nos recursos não podem ser considerados sem a interpretação acompanhada de um especialista.

## 4.3 VALIDAÇÃO

A validação do programa, basicamente consiste em comparar os dados obtidos pelo AG com os dados dos itens agendados realmente para o mesmo período. Em suma pegou-se o melhor resultado de cada dia agendado e estes dados foram colocados lado a lado com os mesmos itens que haviam sido agendados por um processo convencional, não informatizado automaticamente.

### 4.3.1 COMPARATIVO

O comparativo entre o método sem AG e o com AG pode ser visto através da Figura 28, onde são apresentados dois gráficos, respectivamente gráfico da distribuição sem o uso de algoritmo genético e gráfico de distribuição da melhor solução encontrada através do uso de algoritmo genético. A Figura 35 é apenas uma representação de um dos testes de comparação, apenas com o intuito de visualizar a comparação realizada em cada um destes testes.

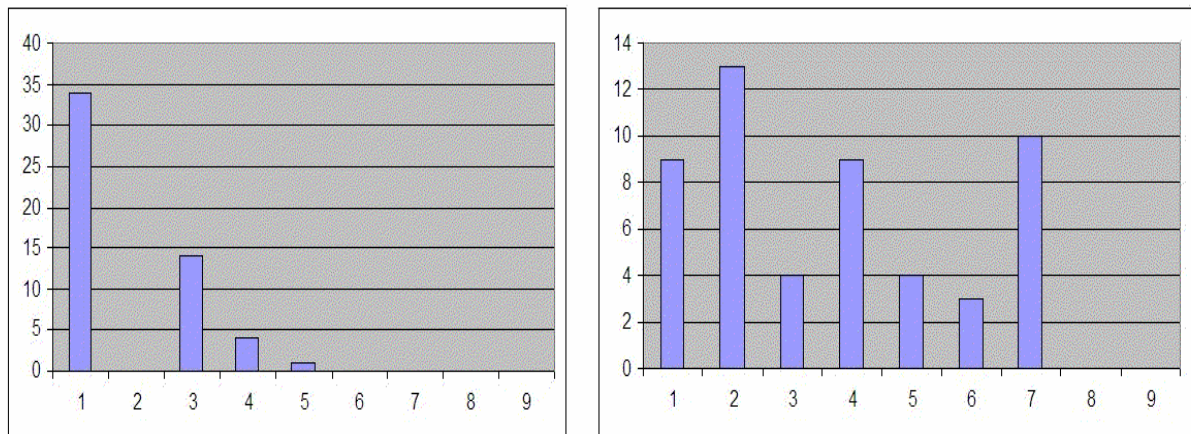


Figura 35. Gráficos de distribuição dos itens recebidos (sem AG e com AG)

Analisando o primeiro gráfico (esquerda), tem-se os 53 itens recebidos alocados em 4 recursos dos 9 disponibilizados, também percebe-se que no recurso 1, houve estouro na capacidade do recurso (mais itens alocados que a capacidade do recurso). No segundo gráfico (direita), utilizando AG temos os mesmos 53 itens, porém melhor distribuídos do que em relação a distribuição sem uso de AG. Evidentemente houve recursos disponibilizados mas que não poderiam executar o serviço em questão, razão porque existem recursos sem nada agendado.

### 4.3.2 INTERPRETAÇÃO

Pelos dados analisados, conclui-se que pelo método tradicional, anterior ao AG, os itens estavam sendo mal distribuídos, presume-se que em virtude de estarem com sobrecarga já anteriormente provocada pelas más distribuições já feitas. As distribuições realizadas pelo AG, obedeceram as regras definidas como restrições no agendamento (um item não pode ser alocado para um recurso que não tenha competência para sua execução como faixa de trabalho, por exemplo). O agendamento realizado pelo AG seguiu as restrições impostas, o que pelo método tradicional muitas vezes foi ignorado.

Obviamente que alguns detalhes não puderam ser incorporados ao AG, o que em parte justificou que no método tradicional houvesse um acúmulo de itens alocados superior à capacidade do recurso, que é a experiência e capacidade do operador técnico. Neste sentido recomenda-se que o AG seja agente que pré-informe o agendamento, ou seja apresente o agendamento e o usuário responsável para definir o agendamento, dê a posição final. Desta maneira a simbiose AG e especialista tornam o agendamento perfeito.

Em praticamente todos os testes comparativos, houve estouro na capacidade do recurso pelo método tradicional sem o uso de AG. Define-se aqui estouro como sendo que a somatória do tempo de execução dos itens agendados ao recurso é maior que a carga de trabalho possível do recurso. Este estouro pode ser visualizado nas tabelas de resultados, comparando para a mesma data no recurso as colunas “R” e “G”, bem como os totais “TR” e “TG”. Bem visível as valores das colunas “R” maior que as “G”, evidenciando excesso de serviço para aquele recurso. Os agendamentos realizados utilizando AG foram melhor distribuídos e também em 100% dos casos foi respeitada a capacidade máxima de trabalho do recurso.

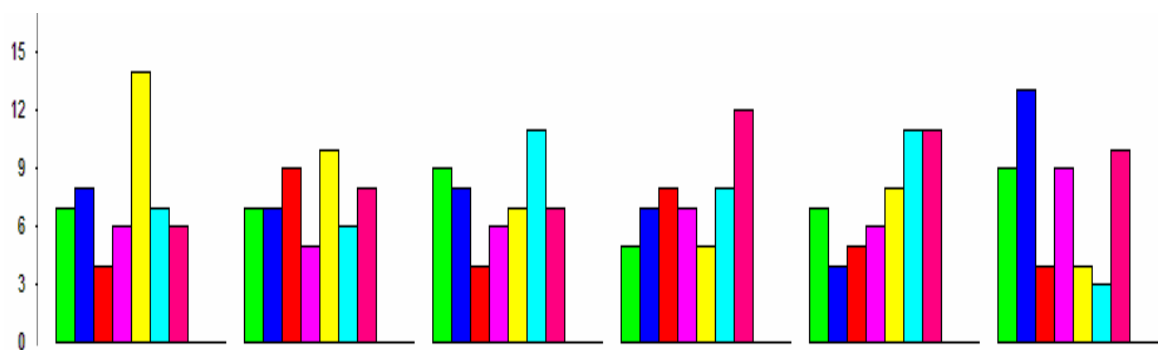


Figura 36. Gráficos de distribuição dos melhores resultados

Apenas como título de confirmação no que tange a respeitar a capacidade dos recursos e a uma melhor distribuição dos itens, a Figura 36 apresenta o gráfico de distribuição das 6 melhores soluções apresentadas pelo AG para um mesmo caso de validação. A figura não possui legendas, é apresentada apenas para mostrar a homogeneidade das distribuições. As soluções apresentadas foram distribuídas ao longo dos recursos (colunas) com mais homogeneidade, não tendo algum recurso específico com excesso de serviço alocado.

## 5 CONSIDERAÇÕES FINAIS

Desde o início do processo da dissertação, até a apresentação deste, os riscos ou áreas de risco especificadas na fase do projeto de dissertação, tinham sido limitados apenas a acessos a informações e dados de laboratórios prestadores de serviços de calibração, tempo de desenvolvimento e dedicação a pesquisa, implementação e testes. Não se havia visualizado nenhum outro item que pudesse oferecer implicações no atraso ou a não execução do projeto..

Porém encontramos algumas dificuldades que não haviam sido previstas. Estas dificuldades acabaram enriquecendo o projeto. Os principais entraves foram os seguintes:

- Surgimento de novos fatores que não haviam sido previstos (itens clones)
- Velocidade de acesso a base de dados. A quantidade de acessos as informações no banco de dados ( cadastros, entrada de itens, características dos itens, etc...) tornava a leitura destes registros no banco de dados muito lenta, o que comprometia o AG.
- Definição dos parâmetros. O que resultou na implementação de um AG de aprendizagem.
- Validação do algoritmo. Fora sugerido a validação através do MATLAB, porém o domínio era restrito e a função de avaliação era por demais complexa ante a quantidade de restrições existentes. Depois de muita pesquisa, optou-se por realizar a validação através do comparativo entre real e solução apresentada.

### 5.1 EXPECTATIVAS

Esta dissertação tem um cunho experimental e prático. No que se refere ao aspecto experimental, tem-se o aprendizado da metodologia e tecnologia a ser adotada. No que se refere a prático, tem-se a implementação desta metodologia e teoria assimilada durante o processo do trabalho. Além de apresentar todas as etapas que ocorreram no desenvolvimento deste projeto também são relatadas as situações que influenciaram significativamente na alteração do cronograma pré-estabelecido no início do projeto.

Outra expectativa é que concretizando as melhorias esperadas neste projeto, possa-se aplicar a metodologia em outros módulos do software de gerenciamento dos meios de medição (LightLab), especificamente nos módulos de apoio a *chão de fábrica*.

Dar uma contribuição à solução de um problema complexo, de cunho prático e que tem uma relevância na área de metrologia, em especial para o pessoal técnico que trabalha com a programação e agendamento de serviços metrológicos. Implicitamente espera-se uma melhoria no atendimento a empresas (que terceirizam este serviço) que necessitam ter no mínimo seus itens entregues no prazo combinado com o prestador de serviço e ou com prazo menor que o acordado.

Também espera-se poder ajudar pessoas que trabalham na área de informática a conhecerem melhor o tema algoritmos genéticos e sua aplicação em *scheduling*. Assim, podem tomar decisões sobre o uso ou não da técnica em seus projetos.

Quanto a validação dos resultados, espera-se que estes possam fornecer informações que colaborem para que o que o método que se propõe na dissertação, possa trazer melhores benefícios e segurança que o método utilizado manualmente pelo especialista.

## 5.2 CONCLUSÃO

Os testes iniciais, utilizando o algoritmo genético canônico, embora apresentasse soluções, ficou a desejar quanto ao seu desempenho. Implementando-se idéias dos artigos de HUSSAIN e JOSH (1998), ZHIMING e CHUNWEI (1997) , MOHAMMED E SANJAY (1998), conseguiu-se melhorar significativamente tanto o desempenho como a quantidade de soluções candidatas.

Muitas vezes, ao se construir um algoritmo são desconsiderados detalhes óbvios que permitiriam ganhos expressivos de desempenho e de qualidade por isso a idéia de similarmente como uma população agrupa indivíduos com características semelhantes, a fim de evoluir e manter a espécie, também é possível utilizar esta mesma constatação no algoritmo proposto, onde desenvolveu-se a idéia de agrupamento.

A implementação de idéias muitas vezes esbarra na própria problemática. A quantidade de variáveis existentes pode tornar inviável uma técnica implementada com êxito para o mesmo tipo de problema, mas com uma quantidade de variáveis menor. Isso se verificou verdade ao passar-se da etapa de dados fictícios para dados reais.

O problema de distribuição de serviços em um conjunto limitado de máquinas em um laboratório de metrologia, considerando um problema real, conforme abordado neste trabalho, apresenta restrições de difícil modelagem analítica, sinalizando para uma abordagem heurística como os AG. Prioridade do serviço, sobrecarga no recurso que irá executar o serviço, histórico de

paradas do recurso, tempo de execução do serviço e tempo ocioso são exemplos de aspectos que devem ser considerados para que se possa obter uma solução otimizada e viável na prática.

De forma geral, os resultados mostraram que o uso do algoritmo genético retorna soluções candidatas de boa qualidade e em número maior, o que permite alternativas no agendamento dos itens, sendo que o desempenho apresentado também é motivador.

Devido a grande variedade de combinações possíveis nos parâmetros do algoritmo genético, partiu-se para as combinações mais racionais e posteriormente substituídas por um algoritmo genético de aprendizagem. Desta implementação observa-se ser de enorme contribuição, porém de uso menos intenso, uma vez que a quantidade de interações que o AG de aprendizagem provoca, torna muito lento para uso direto.

Sugere-se o uso do algoritmo genético, para resolver o problema de distribuição de serviços metrológicos, devido ao ótimo retorno no que se refere a produtividade e escolhas quanto as soluções apresentadas e por ter resolvido o problema existente.

### **5.3 SUGESTÃO PARA TRABALHOS FUTUROS**

Como sugestão para futuros trabalhos cita-se a implementação de outras técnicas de *crossover* e a melhora da implementação do AG de aprendizagem, incluindo mais parâmetros e a alteração do AG primário para um paralelo, tratando os vários agrupamentos simultaneamente.

Especificamente no que se refere ao AG de aprendizagem, realizar testes sem a restrição das taxas limitadas a 63% com o objetivo de verificar se taxas com valores maior que o limitado venham a apresentar uma melhor combinação de parâmetros que permitam melhor performance e quantidade de soluções satisfatórias.

Outro item interessante, é a alteração do procedimento de inicialização da população. O algoritmo do procedimento de inicialização no AG implementado, consiste em gerar uma população onde cada cromossomo criado é inicialmente composto de “0” (zeros) e enxertados aleatoriamente tantos “1” (huns) quantos forem a quantidade de itens a serem distribuídos. Sugere-se que os “1” inseridos já sejam previamente colocados nos recursos compatíveis com o tipo de serviço. Teoricamente estaremos criando já uma população inicial mais próxima da solução desejada, o que também teoricamente induziria a um maior número de soluções satisfatórias e um aumento considerável na performance do AG.



A utilização de esquemas (*SCHEMAS*) não foi abordada, porém acredito que o acréscimo na opção de *crossover*, utilizando esquemas possa também apresentar resultados interessantes para serem analisados.

Por último, mas não menos importante, realizar os mesmos testes em mais grupos de parâmetros diferentes e tentar mostrar graficamente a influência dos parâmetros na performance e quantidade de soluções apresentadas. Estes testes, embora o algoritmo de aprendizagem sugira a melhor combinação, são importantes para levantar esta correlação de parâmetros *versus* resultados.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ADAMIDIS, P. (1994). *Review of parallel genetic algorithms bibliography*. Aristotle University
- ADAMS, J., BALAS, E., ZAWACK, D., (1988) *The Shifting Bottleneck Procedure for Job shop Scheduling*, Management Science, v. 34, 1988, p. 391-401.
- AICKELIN, U. and WHITE, P.; (2004) *Building Better Nurse Scheduling Algorithms*. Annals of Operations Research, 128, pp 159-177.
- AICKELIN, U. , DOWSLAND, K.A. , (2004), *An Indirect Genetic Algorithm for a Nurse Scheduling Problem*; Computers & Operations Research, 31(5), pp 761-778, 2004
- APPLEGATE, D; COOK, W (1991), “*A computational study of the job-shop scheduling instance*”, ORSA Journal on Computing 3, 149-156.
- NUIJTEN, W.P.M. and AARTS, E.H.L.(1995) *A Computational Study of Constraint Satisfaction for Multiple Capacitated Job Shop Scheduling*, European Journal of Operational Research, Volume 90, Issue 2, 19 April 1996, Pages 269-284.
- ARROYO, J.C.E. (1998), **Um algoritmo genético não convencional para o problema de Steiner em grafos direcionados** - Tese de Mestrado - Programa de Pós-Graduação em Ciência da Computação
- BAKER, K. R. (1974), *Introduction to Sequencing and Scheduling*, John Wiley, New York.
- BALAS, E.; VAZACOPOULOS, A. (1998) *Guided local search with shifting bottleneck for job shop scheduling*. *Management Science*, v. 44, n. 2, p. 262-275.
- CONSTANTINE, L. (1995) *Software Objectives*. *Software Development*. v.3, n.6, p 87-88.
- DAVIS, L.;(1985); *Job shop scheduling with genetic algorithms*. In: *Proceedings of the Int. Conf. on Genetic Algorithms and their Applications*, pp 136–40, Pittsburgh, PA.
- DEJESUS, E. X. (1995) **Programação sem sustos**. Byte.
- DORN, J.; GIRSCH, M.; SKELE, G. and SLANY, W; (1996); *Comparison of iterative improvement techniques for schedule optimization*. European Journal of Operational Research.
- Edition.
- FILHO, J. L., ALIPPI, C., and TRELEAVEN, P. (1993). *Genetic Algorithm Programming Environments*, chapter Parallel Genetic Algorithms: Theory and Applications. IOS Press, j. stender
- FRENCH, S. (1982), *Sequencing and Scheduling An Introduction to the Mathematics of the Job-Shop*, John Wiley, New York.
- FISCHER, R.A. (1930). *The Genetical Theory of Natural Selection*, p. 1-21. Oxford University Press, Oxford.

- FUKUMORI, K; (1980); *Fundamental scheme for train scheduling*. MIT AI Memo 596, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, September
- GLOVER, F. e M. LAGUNA, M., (1997); *Tabu Search*. Kluwer, Boston.
- GOLDBERG, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*.
- HAGHIGHAT, A.T. and NIKRAVAN, M.. (2005) *A Hybrid Genetic Algorithm for Process Scheduling in Distributed Operating Systems Considering Load Balancing*. Parallel and Distributed Computing and Networks, 2005. Austria.
- HAGHIGHAT, A.T.; NIKRAVAN, M. *A Hybrid Genetic Algorithm for Process Scheduling in Distributed Operating Systems Considering Load Balancing*.  
www.actapress.com/PaperInfo.aspx?PaperID=19252
- HANSEN, P. (1986); *The steepest ascent mildest descent heuristic for combinatorial programming*, Congress on Numerical Methods in Combinatorial Optimization, Capri, .
- HOLLAND, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, University of Michigan Press (2nd ed.: MIT Press ,1992).
- HUSSAIN, M.F.; JOSHI, S.B. (1998) *A genetic algorithm for job shop scheduling problems with alternaterouting*. Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on, Outubro 1998, San Diego, CA, USA.
- JOHNSON, S.M., (1954). *Optimal Two and Three-Stage Production Schedules with Set-Up Times Included*, Naval Research Logistics Quarterly, Vol. 1, pp. 61-68.
- JONG, K. (1975) *An Analysis of the Behavior of a class of Genetic Adaptive System*. Tese de Doutorado, University of Michigan.
- KOHLER, W. (2002) *Escalonador de Ordens de Produção Utilizando Algoritmos Genéticos*, FURB/ Metalúrgica Siemens Ltda.
- LIAU, X.; BO, H.; MA, Y.; MENG, Q. (2006) *A New Approach for Planning and Scheduling Problems in Hybrid Distributed Manufacturing Execution System*. Proceedings of the 6th World Congress on Intelligent Control and Automation, Junho, 2006, Dalian, China.
- LINDEN, R.;(2006); **Algoritmos genéticos, uma importante ferramenta da inteligência computacional**. Editora Brasport.
- LUH, P.B., ZHAO, X, THAKUR, L.S., CHEN, K.H. , CHIUEH, T.D., CHANG, S.C.; (1999); *Architectural Design of Neural Network Hardware for Job Shop Scheduling*, CIRP Annals, Vol. 48, No. 1 , pp. 373-376
- MARTIN, O., OTTO, S.W., FELTEN, E.W. ;(1992); *Large-step Markov Chains for TSP Incorporations Local Search Heuristics*, Operations Research Letters, v. 11, p. 219-224.

- MATSUO, H., Suh, C.J., SULLIVAN, R.S. ;(1988); *A controlled search simulated annealing method for the general jobshop scheduling problem*, Working paper , Department of Management, The University of Texas at Austin,
- MICROGOLD ed. (1996) *with class User's Guide*. Manual de programa.
- MITCHELL, M. (1996) *An Introduction to Genetic Algorithms*. MITPress.
- MOHAMMED F. H.; SANJAY B. J. (1998) **A Genetic Algorithm for Job Shop Scheduling Problems with Alternate Routing**. Pennsylvania State University, University Park.
- MOORE, M. (2003); *An accurate and efficient parallel genetic algorithm to schedule tasks on a cluster*. International Parallel and Distributed Processing Symposium (IPDPS'03), 2003.
- MOORE, M. (2004); *An accurate parallel genetic algorithm to schedule tasks on a cluster*. IEEE - Parallel and Distributed Processing Symposium Publication Date:Maio.
- MORTON, T.E. ; PENTICO, D; (1993), *Heuristic Scheduling Systems - With Applications to Project Scheduling*, John Wiley, New York.
- NUIJTEN, W.P.M. ; AARTS, E.H.L.(1995); *A Computational Study of Constraint Satisfaction for Multiple Capacitated Job Shop Scheduling*  
of Thessaloniki
- ONO,I.; YAMAMURA,M.; KOBAYASHI,S.(1996) **Genetic algorithm for job shop scheduling problems Using job-based order Crossover**. Japan: Department of Intelligence Science,Graduate School of Interdisciplinary,Science and Engineering,Tokyo Institute of Technology.
- PACHECO, M.A.C. (1999); **Algoritmos genéticos: Princípios e aplicações**. Departamento de Engenharia Elétrica, Pontificia Universidade Católica do Rio de Janeiro.
- PARK, L.J. and PARK, C.H.(1995) **Genetic algorithm for job shop scheduling problems based on two representational schemes**. Vol. 31 N. 23, ELECTRONICS LETTERS.
- PAZ-CANTU, E. A. (1997). *A Summary of Research on Parallel Genetic Algorithms*. In GAL R PETTEY, C., LEUZE, M., and GREFENSTETTE, J. (1987). *Parallel genetic algorithm*. In Grefenstette,J., editor, Proceedings of the 2nd ICGA, pages 155–161. Lawrence Erlbaum Associates.
- RESENDE, M.G.C., FEO, T.A. and SMITH., S.H., (1998), *Fortran subroutines for approximate solution of maximum independent set problems using GRASP*. ACM Trans. Math. Software, 24:386-394.
- RSBT - **Revista da Sociedade Brasileira de Telecomunicações** (2003) Volume 18, Número 2.
- RUAN, D. (1997) *Intelligent Hybrid Systems – Fuzzy Logic, Neural Networks and Genetic Algorithms*, cap.1 e 11, Kluwer Academic Publishers.
- RUMBAUGH, et al. (1994) *Modelagem e Projetos Baseados em Objetos*. Rio de Janeiro, Campus.

SONG, Y., HUGHES, J.G., AZARMI, N., VOUDOURIS, V. ; (2004); *A Genetic Algorithm with an incomplete representation for the Job Shop Scheduling Problems*; CITESEER, citeseer.ist.psu.edu/530256.html

TRALESKI, R. (2005) *Otimização de um algoritmo de estimativa de distribuição na tarefa de seleção das características*; Dissertação de mestrado, Universidade Federal do Paraná.

ULDER, N.L.J., PESCH, E., VAN LAARHOVEN, P.J.M., BANDELT, H., AARTS, E.H.L.; (1994); *Genetic local search algorithm for the traveling salesman problem*. In 1st PPSN, pages 109-116.

VOGET, S.; KOLONKO, M.; (1998); *Multidimensional Optimization with a Fuzzy Genetic Algorithm*; Journal of Heuristics; citeseer.ist.psu.edu/758696.html;

XING, Y.; CHEN, Z.; SUN, J.; HU, L. (2007) *An Improved Adaptive Genetic Algorithm for Job-Shop Scheduling Problem*. Third International Conference on Natural Computation (ICNC 2007).

YIN, H.; HU, E.; WANG, Y.; XIAO, N. (2007) *A Two-Stage Genetic Algorithm for Large-Size Scheduling*. Proceedings of the IEEE International Conference on Automation and Logistics Agosto, 2007, Jinan, China.

YIN, H.; WANG, Y.; XIAO, N.; HU, E.; JIANG, Y. (2007) *Three-dimensional Encoding Genetic Algorithm for Job Shop Scheduling* International Conference on Computational Intelligence and Security Workshops.

ZHIMING, W, and CHUNWEI, Z. (1997) *A Genetic Algorithm for Job Shop Scheduling in Real Time*. Proceedings of the American Control Conference Albuquerque. New Mexico.















## APENDICE C

Interpretação do melhor resultado do A.G. executado  
Serviços de: Calibração

-----  
Existem itens clonados (mesmo item alocado para outro recurso)  
-----

Recurso : Máquina de medição longitudinal  
Itens a serem alocados para este recurso  
Item: [16132/3] Calibrador de boca fixo

Recurso : Não geométricos  
Itens a serem alocados para este recurso  
Item: [16127/3] Placa de planicidade  
Item: [16132/5] Gabarito para controle  
Item: [16133/2] Relógio comparador centesimal

Recurso : Instrumentação  
Itens a serem alocados para este recurso  
Item: [16127/1] Placa de planicidade

Recurso : Máquina de medição por coordenadas  
Itens a serem alocados para este recurso  
Item: [16127/5] Placa de planicidade  
Item: [16133/1] Relógio comparador centesimal  
Item: [16133/6] Calibrador tampão roscado cilíndrico (PNP)

Recurso : Microscópio de medição  
Itens a serem alocados para este recurso  
Item: [16127/6] Placa de planicidade

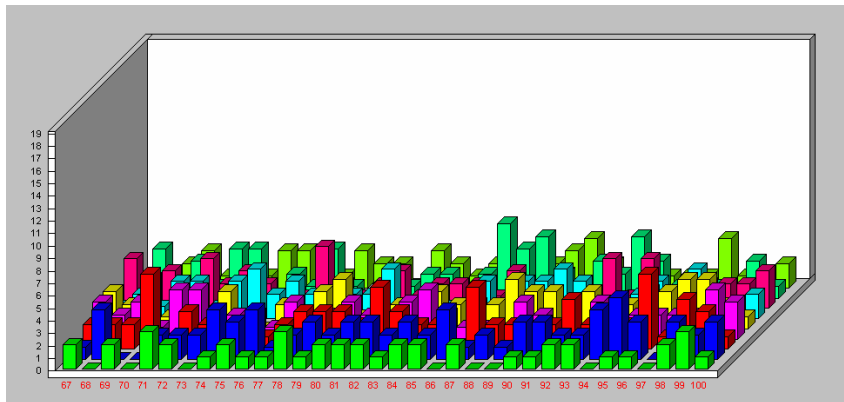
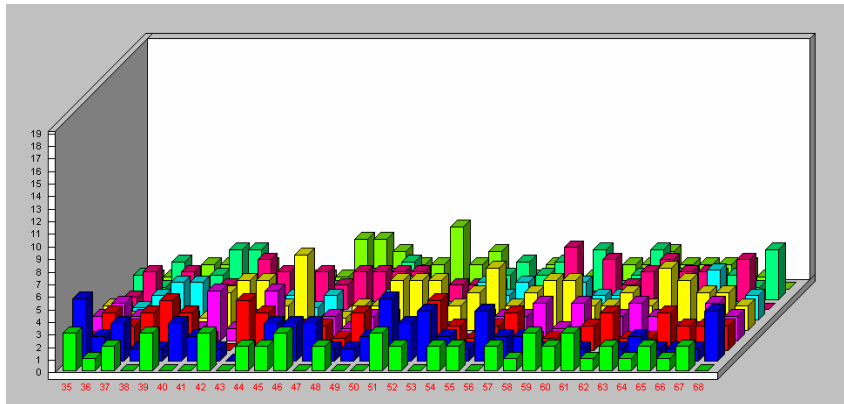
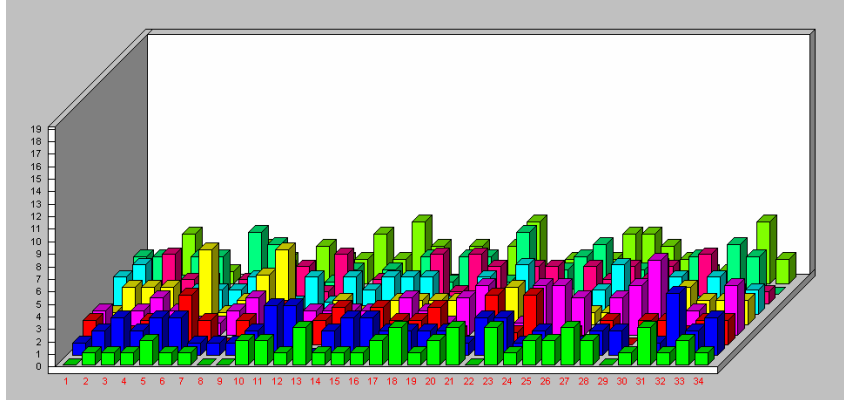
Recurso : Máquina de medição de rugosidade e perfil  
Itens a serem alocados para este recurso  
Item: [16132/2] Calibrador tampão liso cilíndrico (PNP)  
Item: [16133/3] Relógio comparador centesimal

Recurso : Comparador eletromec. de blocos-padrão  
Itens a serem alocados para este recurso  
Item: [16127/1] Placa de planicidade  
Item: [16132/1] Calibrador tampão liso cilíndrico (PNP)  
Item: [16132/3] Calibrador de boca fixo

Recurso : Laser interferométrico  
Itens a serem alocados para este recurso  
Item: [16133/4] Calibrador tampão roscado cilíndrico (PNP)

Recurso : Calibrações em Campo  
Itens a serem alocados para este recurso  
Item: [16132/3] Calibrador de boca fixo  
Item: [16132/4] Calibrador especial

# APÊNDICE D



## ANEXO A

### PRINCIPAIS MÉTODOS DE SELEÇÃO DE PAIS NO AGs.

- Método da roleta :
  - Cria-se uma roleta virtual;
  - Cada cromossomo recebe um pedaço proporcional a sua avaliação;
  - A soma dos pedaços não pode ultrapassar 100;
  - “Roda-se” a roleta e o selecionado é o indivíduo onde ela parar.
- Método do torneio:
  - Utiliza sucessivas disputas para realizar a seleção;
  - Para selecionar k indivíduos, realiza k disputas, cada disputa envolvendo n indivíduos escolhidos ao acaso;
  - O indivíduo de maior aptidão na disputa é selecionado;
  - É muito comum utilizar  $n = 3$ .
- Método da amostragem estocástica uniforme:
  - Semelhante à Roleta, mas para selecionar k indivíduos utiliza k agulhas igualmente espaçadas, girando-as em conjunto uma só vez;
  - Apresenta resultados menos variantes que a Roleta.
- Método Local:
  - Começa de uma solução inicial  $\delta$ ;
  - Repetidamente troca  $\delta$  pela melhor solução de sua vizinhança  $N(\delta)$ ;
  - Até que nenhuma melhor solução seja encontrada em  $N(\delta)$ ;
  - A solução resultante  $\delta$  é localmente ótima no sentido de que não existe melhor solução em sua vizinhança
- Seleção por ranking:
  - Existem vários tipos de modelos por ranking, o mais simples adota o seguinte critério:
  - Ordenar os cromossomos de acordo com seus valores de fitness, do maior para o menor;
  - Um valor arbitrário de fitness mínimo é escolhido como ponto de corte, onde qualquer cromossomo abaixo desse limiar é removido da população;
  - Há várias técnicas para completar a população que foi ‘dizimada’, seja através de reprodução dos cromossomos restantes, ou inserindo cromossomos aleatórios.
- Seleção truncada:
  - Apenas x % da população poderão ser escolhidos como pais da próxima geração;
  - Valores mais usuais de x 10 a 50%;
  - Os indivíduos são ordenados de acordo com a sua avaliação;
  - Somente os de posição 1 até a de corte, participarão da seleção;
    - Convergência genética veloz
    - Perda da diversidade;
    - Blickle (1997) demonstra que este método é o que causa a maior perda de diversidade entre todos os métodos.
- Seleção Elitista
  - são selecionados os N melhores indivíduos da população intermediária
  - Seleção Aleatória
  - são selecionados aleatoriamente N indivíduos da população intermediária
- Salvacionista
  - seleciona-se o melhor indivíduo e os outros aleatoriamente

- Não-salvacionista
  - seleciona aleatoriamente todos os indivíduos
    - Distribuição uniforme – todos têm a mesma chance
    - Distribuição proporcional
    - roulette wheel por fitness
    - roulette wheel por fitness com sigma scaling
    - seleção de Boltzmann
    - seleção por rank
- Seleção por diversidade
  - são selecionados os indivíduos mais diversos na população intermediária, a partir do melhor indivíduo
- Seleção bi-classista
  - são selecionados o P% melhores indivíduos e os (100-P)% piores indivíduos.
- Seleção Steady-State
  - a população original é mantida, com a exceção de alguns poucos indivíduos menos adaptados.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)



[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)