

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL

CAMPUS UNIVERSITÁRIO II – URUGUAIANA

FACI

**CIÊNCIA DA COMPUTAÇÃO – META-AG PARA DEFINIÇÃO DE HORÁRIOS
ACADÊMICOS**

ACADÊMICO: RAFAEL BLANCO LOPES

ORIENTADOR: RICARDO ANNES

TRABALHO DE CONCLUSÃO 2

Uruguaiana, julho 2005.

SUMÁRIO

1 INTRODUÇÃO.....	1
2 ALGORITMOS GENÉTICOS.....	3
2.1 CODIFICAÇÃO.....	6
2.2 CRUZAMENTO.....	7
2.3 MUTAÇÃO.....	7
2.4 UM MODELO PARA OTIMIZAR HORÁRIOS COM ALGORITMOS GENÉTICOS.....	8
3 META-ALGORITMOS GENÉTICOS.....	11
3.1 A DINÂMICA DOS PARÂMETROS.....	15
3.2 A DINÂMICA DO AG.....	16
3.3 IMPLEMENTAÇÃO.....	16
3.4 PARTES DO META ALGORITMO GENÉTICOS.....	17
3.5 O META-CROMOSSOMO.....	18
3.6 A META FUNÇÃO DE ADEQUAÇÃO.....	19
3.7 O PROJETO TURBO ALGORITMO GENÉTICO (TAG).....	20
4 META-AG PARA DEFINIÇÃO DE HORÁRIOS ACADÊMICOS.....	22
4.1 O FUNCIONAMENTO DO META-AG.....	22
4.2 PROBLEMA MOTIVADOR.....	23

4.3 O META AG IMPLEMENTADO.....	23
4.4 A META FUNÇÃO DE ADEQUAÇÃO.....	24
4.5 ESQUEMA DE CRUZAMENTOS.....	25
4.6 ESQUEMA DE MUTAÇÃO.....	25
4.7 CONFIGURAÇÃO E TESTES.....	26
4.8 CONFIGURAÇÃO DO AG.....	26
4.9 CONFIGURAÇÃO DO META-AG.....	26
4.10 CRITÉRIO DE COMPARAÇÃO.....	27
4.11 TESTES.....	27
4.12 RESULTADOS.....	28
5 CONCLUSÃO.....	30
6 TRABALHOS FUTUROS.....	31
7 APÊNDICE A : O META-AG PROPOSTO E SUAS TELAS.....	32
8 CONSIDERAÇÕES FINAIS.....	39
9 REFERÊNCIAS BIBLIOGRAFIA.....	40

LISTAS DE FIGURAS

1	Figura 1 – Melhores valores para De Jong.....	12
2	Figura 2 – Melhores valores para Grefenstette.....	12
3	Figura 3 - Melhores valores para os parâmetros segundo Schaffer.....	15
4	Figura 4 – Layout Meta-AG interagindo com AG.....	17
5	Figura 5 - “Lay-out” de um meta-cromossomo.....	18
6	Figura 6 - “Lay-out” de PC e PM.....	19
7-	Figura 7 - “Lay-out” do mc do Meta-AG para Definição de Horários Acadêmicos.....	24
8 -	Figura 8 – Gráfico do experimento com AG.....	27
9-	Figura 9 – Gráfico do experimento com Meta-AG.....	29
10-	Figura 10- Tela principal.....	33
11-	Figura 11 – Tela de opções do Meta-AG.....	33
12-	Figura 12 Tela de opções do AG.....	35
13-	Figura 13 – Tela de Execução.....	36
14-	Figura 14 – Tela de resultados.....	37

1- INTRODUÇÃO

Atualmente, os Algoritmos Genéticos (AG) vem sendo largamente utilizados para a solução dos mais diversos problemas de escolha. Nesse trabalho são descritas características gerais de um tipo específico de Algoritmo Genético chamado de Meta Algoritmo Genético (Meta-AG) , que dinamicamente altera os parâmetros do Algoritmo Genético (que serão citados nos próximos capítulos) para obter um rendimento melhor, conseqüentemente modificando o seu funcionamento.

No VIII SIMPÓSIO DE INFORMÁTICA E III MOSTRA DE SOFTWARE ACADÊMICO DA PUCRS – CÂMPUS URUGUAIANA, RS foi apresentado um projeto de pesquisa chamado UM MODELO PARA OTIMIZAR HORÁRIOS COM ALGORITMOS GENÉTICOS para a definição de grades de horário para a Faculdade de Informática (Ciência da Computação e Sistemas de Informação), baseado nesse Sistema desenvolvido implementamos um Sistema Meta-AG para esse AG em questão. Com o objetivo de fazer um estudo comparativo entre a resolução do problema dos horários utilizando um Meta-AG junto com um AG e utilizando apenas um AG, e tentar obter com esse META-AG uma convergência (horários sem colisão) ou bons horários (horários com poucas colisões) para o AG que modela o problema citado.

O fato que nos levou a escolher esse tema foi em especial o meu interesse e do orientador pelo assunto e a existência de um AG pronto para combinar com o Meta-AG.

O trabalho está organizado da seguinte forma. No capítulo 1 é feita uma introdução ao Meta-AG proposto. No capítulo 2 é descrita a teoria básica dos AGs (fundamental para o entendimento dos Meta-AGs sendo que Meta-AG é um tipo de AG), alguns aspectos que fazem parte dos AGs e que por consequência pertencem aos Meta-AGs tais como cromossomo, gene, mutação, cruzamento, seleção, gerações são abordados de maneira mais explicativa. No capítulo 3 serão abordados conceitos referentes a Meta-AGs tais como meta-cromossomo, função de adequação, codificação, histórico e pesquisas. No capítulo 4 é mostrada uma descrição detalhada do Meta-AG implementado, e os resultados obtidos do estudo comparativo realizado entre AG e Meta-AG. No capítulo 5 o autor apresenta as conclusões sobre o projeto. No capítulo 6 são propostos temas de interesse do autor para trabalhos futuros. No capítulo 7 são apresentadas as telas do sistema desenvolvido. No capítulo 8 o autor faz suas considerações finais. No capítulo 9 a bibliografia utilizada é apresentada.

2 ALGORITMOS GENÉTICOS

Os algoritmos genéticos são utilizados em grande parte para resolver problemas de busca e otimização, estando baseados na teoria darwiniana da evolução genética dos seres vivos.

Ao longo das gerações as diversas espécies de seres vivos evoluem de maneira a aumentarem a sua chance de sobrevivência a um determinado problema imposto pelo ambiente. Os AGs são capazes de criar soluções para problemas computacionais copiando a teoria da evolução das espécies que diz que apenas os mais fortes e mais aptos a sobreviver a uma dificuldade sobreviverão.

Na natureza os indivíduos que melhor se adaptam a um problema imposto tem mais chances de sobreviverem e conseqüentemente espalhar seus genes superiores para as próximas gerações, dessa forma as espécies evoluem, logrando características cada vez melhores adaptadas no ambiente em que vivem.

Os AGs usam uma analogia direta com o comportamento natural. Trabalham com um conjunto de indivíduos representando cada individuo uma solução para um problema dado.

Antes de cada geração é calculado para todos os indivíduos (na modelagem computacional chamado de cromossomo) o grau de aptidão daquela solução em relação a

um problema. Esse grau de aptidão determinará as chances do indivíduo reproduzir e conseqüentemente gerar descendentes (espalhar os seus genes para as próximas gerações), esses descendentes terão algumas características dos pais.

Quanto menor for o grau de aptidão de um indivíduo menor serão as chances dele ser selecionado para reprodução, portanto dificilmente perpetuará seus genes para as próximas gerações.

Sendo assim, a cada geração se origina uma nova geração de descendentes da anterior, com características mais adaptadas aos problemas propostos. A cada nova geração as características da população de indivíduos esta mais adaptada a um problema, favorecendo assim exploração das áreas mais promissoras nas sucessivas gerações.

O poder dos AGs vem do fato de se tratar de uma técnica genérica que pode ser utilizada para uma grande variedade de problemas computacionais tais como de busca e de otimização. Geralmente o ótimo global (solução para o problema, cromossomo sem nenhuma colisão do problema) ou uma boa solução (poucas colisões do problema a ser solucionado) são encontrados em um tempo razoável, apesar de algumas vezes essa solução não ser encontrada, os AGs são uma ótima forma de modelar problemas a serem solucionados.

O ponto forte dos AGs é sua característica de utilização como um método genérico de solução de problemas, mas mesmo assim em existindo uma técnica específica para solucionar um problema a tendência é que o AG seja superado.

Basicamente os AGs tratam problemas de otimização como um processo iterativo de busca da melhor solução dentro do espaço de possíveis respostas para o problema. Inicia com um conjunto aleatório de soluções iniciais, que constituem a população inicial. Combinando os melhores representantes dessa população, obtém uma nova, que passa a substituir a anterior, caracterizando-se como a próxima geração. O mecanismo é repetido. A cada nova iteração a população é refinada gerando novas e

melhores soluções para o problema em questão, podendo culminar com a sua convergência.

O AG, é mostrado a seguir

Gerar população inicial;

Calcular a função de avaliação de cada indivíduo

Enquanto não (terminou) faça

Início (produz nova geração)

Para (tamanho da população)/2 faça

Início (Ciclo reprodutivo)

Selecionar dois indivíduos da geração anterior para cruzamento

(probabilidade de seleção proporcional à função de avaliação do indivíduo).

Cruzar com certa probabilidade obtendo os descendentes;

Mutação nos descendentes com certa probabilidade;

Calcular a função de avaliação dos descendentes que sofreram mutação;

Incluir os descendentes que sofreram mutação na nova geração;

Fim-para

Se (população convergiu) então

Terminou:=verdade;

Fim-enquanto;

Fim-Algoritmo

Alem da necessidade da codificação desse algoritmo em uma linguagem de programação é necessária a definição de uma função de ajuste ou adequação, que a cada geração associará um valor a cada solução, esse valor representa o grau de adequação de uma solução.

Os pais deverão ser selecionados dependendo da sua adequação, após serem cruzados gerarão novos indivíduos que formarão a nova população da próxima geração. Esse processo é repetido sucessivamente até que se chegue na **conversão** (uma solução se colisões) ou não do algoritmo, poderia ser definido também um limite para a conversão como por exemplo 100 gerações passando disso o programa não deve mais executar.

2.1 CODIFICAÇÃO

Os indivíduos da população são codificados através de um conjunto de genes, e esses genes agrupados dão origem a um individuo (daqui por diante referenciado como cromossomo).

Na maioria das implementações e bibliografia sobre AGs os indivíduos está representado de forma binária com seus genes assumindo valores que podem variar entre 0 e 1, embora isso não seja necessariamente obrigatório, podendo haver outros tipos de codificação, como no caso do AG para otimizar horários que é um dos temas de pesquisa desse trabalho.

O grau de adaptação de um cromossomo é dado pela avaliação de seu genótipo (o conjunto de genes que constituem um cromossomo), a avaliação é feita por uma função de adequação que é construída levando-se em conta um problema a ser revolvido pelo AG. Essa avaliação do genótipo atribui a cada cromossomo uma probabilidade que representa o seu grau de adaptabilidade..

Os cromossomos selecionados para a reprodução são os que melhor se adaptam ao problema mediante função de adaptação. Após serem selecionados o código genético dos pais é combinado para dar origem a um novo cromossomo que fará parte da nova geração.

2.2 CRUZAMENTO

Para os cruzamentos entre cromossomos selecionados, partes destes são igualmente seccionadas e os genes dessas secções são combinados para formar um novo cromossomo.

O cruzamento geralmente não se aplica a todos os indivíduos selecionados. Sendo somente escolhidos aqueles que tem um alto grau de adaptação de 50% a 99%, assim para os que não são selecionados a descendência é feita apenas duplicando os pais.

2.3 MUTAÇÃO

O operador de mutação atinge todos os indivíduos da população, a cada geração após o cruzamento todos os indivíduos tem uma pequena probabilidade de terem os genes dos seus cromossomos alterados (gene 0 passa a ser 1 e vice versa).

A princípio o operador de cruzamento pode parecer mais importante que o operador de mutação pois é mais evasivo, e permite um rápido e eficiente método de promover a criação de novos cromossomos a partir da mistura de genes. Mas o operador de Mutação também tem um papel importante, pois garante que todos os pontos do espaço de busca sejam examinados e é de fundamental importância para assegurar a convergência, principalmente com a utilização de Meta-AGs.

É importante um conceito de convergência a essa altura do estudo, Para De Jong (1975), se o AG foi corretamente implementado, a população evoluirá ao longo das gerações sucessivas de forma que a adaptação média estendida a todos os indivíduos da

população, assim como a adaptação do melhor indivíduo será incrementada de tal forma a convergir para o ótimo global.

A convergência para De Jong (1975) só ocorrerá com a uniformidade dos genes nos cromossomos, ou seja a grande maioria desses cromossomos deve compartilhar de pelo menos 95% do valor para um mesmo gene.

2.4 UM MODELO PARA OTIMIZAR HORÁRIOS COM ALGORITMOS GENÉTICOS

Foi proposto e implementado pelo professor Ricardo Annes e pelo aluno Madison Delano Tobal da Paz, um AG para solucionar o Problema da geração de grades de horários para os cursos de Ciência da computação e Sistemas de informação da FACI.

Uma grade de horários com suas disciplinas bem distribuídas nos diversos níveis que compõem um curso é fundamental para uma boa alocação de recursos humanos e materiais. A cada novo período letivo (ano, semestre) é necessário organizar e distribuir os horários de aulas de professores e alunos. Devido ao número elevado de restrições e disciplinas, a distribuição é considerada complexa.

A geração de grades de horários leva em consideração vários fatores tais como, as disciplinas que compõem cada nível de cada curso, a carga horária semanal de cada disciplina, o professor que ministra a disciplina (ou professores), a disponibilidade de cada professor e suas preferências. São restrições do problema a colisão de horários (professor com mais de uma disciplina no mesmo horário), a indisponibilidade de professores e a distância mínima entre aulas da mesma disciplina. Em certos casos também é necessário considerar algum grupo de alunos que tem uma ou mais disciplinas pendentes.

No modelo proposto em cada cromossomo (indivíduo da população), é codificada a distribuição das disciplinas dos níveis dos cursos ao longo da semana, nos diversos períodos de aula de cada dia. Também é codificado um campo para armazenar o

seu grau de ajuste (fitnes). O cálculo da função objetivo é feito somando as colisões de cada cromossomo de acordo com as restrições programadas (quanto menor o numero de colisões melhor o cromossomo estará adaptado ao problema). As colisões são indisponibilidade dos professores e a distância mínima entre aulas da mesma disciplina.

Um valor de adaptação zero (sem colisões) é uma solução ótima. A população inicial é calculada levando-se em consideração uma distribuição coerente, mesmo que haja conflitos entre os níveis. O operador de cruzamento foi definido de forma que os novos indivíduos da população são criados mesclando os cromossomos 1 e 2 do pai com os 3 e 4 da mãe, para formar uma nova geração.

O operador de mutação seleciona e troca dois períodos de um dos cinco níveis, a escolha do nível é feita de forma aleatória. Para selecionar a nova geração é aplicado o operador de elitismo (escolha dos melhores cromossomos de acordo com a função de adequação), para preservar os melhores indivíduos para a próxima geração.

O algoritmo do **Modelo Proposto** é composto dos seguintes passos:

1. A INICIALIZAÇÃO – criação da população inicial; as disciplinas com horário são distribuídas aleatoriamente em cada nível e a seguir as demais disciplinas são incluídas para completar o horário;
2. AVALIAÇÃO – cada indivíduo é avaliado e são calculadas quantas colisões ocorrem;
3. NOVA POPULAÇÃO – Uma nova população é criada com os seguintes passos.
4. SELEÇÃO – os pais selecionados aleatoriamente são cruzados
5. CRUZAMENTO – cada filho é composto de parte de cromossomos do “pai” e da “mãe”.

6. MUTAÇÃO – os cromossomos tem dois períodos trocados.
7. AVALIAÇÃO – todos os indivíduos da população tem o seu genótipo avaliado e o número de colisões é determinado para cada cromossomo.
8. SUBSTITUIÇÃO – dentre a nova população e a anterior utilizando a escolha elitista são escolhidos os melhores indivíduos.
9. TESTE – se o critério de fim é alcançado, terminar. Caso contrário voltar para o passo 3. Os critérios de parada são o número de gerações ou a obtenção de soluções sem colisões.

3 META-ALGORITMOS GENÉTICOS

Um dos problemas que devem ser enfrentados, para quem pretende usar um AG, reside na escolha dos parâmetros. Todos os AGs usam, pelo menos, três parâmetros numéricos (alguns podem usar outros, como tipo de combinação), esse parâmetros são Probabilidade de mutação, Recombinação e Tamanho da população.

Esses parâmetros em um AG, são estáticos não se alteram durante todo o processamento, sendo assim fica evidenciado uma grande variação no desempenho do AG em função dos parâmetros escolhidos.

Para resolver esse problema foi desenvolvido, um tipo especial de AG o Meta AG, que pode alterar dinamicamente os parâmetros de um AG, para obter uma melhora no seu aproveitamento.

Talvez o maior problema enfrentado na implementação de AGs possa ser, a falta de literatura sobre a escolha de seus parâmetros e usualmente são utilizados parâmetros escolhidos empiricamente (obtidos através da experiência do pesquisador).

Alguns pesquisadores destacam-se no estudo de parâmetros para o AG entre os quais pode-se destacar: **De Jong**, Grefenstette, Goldberg e Schaffer.

O pioneiro De Jong, elaborou cinco funções de teste, foi dele também a criação de duas medidas de desempenho: O desempenho **online** que é definido como a média de

adequação de todos os cromossomos da população durante o processamento e o desempenho **offline**, que é a medida apenas do melhor cromossomo da população para o cômputo do desempenho.

De Jong também usou comparativamente a recombinação multiponto com a recombinação simples e concluiu que não houve diferenças significativas no desempenho.

Após seus experimentos De Jong apontou os seguintes parâmetros como os ideais na implementação de AG.

Valores de De Jong

Tamanho da população	Prob. de recombinação	Prob de Mutação
50-100	60%	0,001

Figura 1 – Melhores valores para De Jong

Grefenstette em 1986, provavelmente foi o primeiro pesquisador a tratar a escolha dos parâmetros do AG como um problema (um meta-problema) e usou um segundo AG (Meta-AG) para dinamicamente encontrar os parâmetros que obtenham o melhor desempenho na execução de um AG, após suas pesquisas Grefenstette chegou aos seguintes resultados para o desempenho online.

Valores de Grefenstette

Tamanho da população	Prob de recombinação	Mutação
30	0,95%	0,01

Figura 2 – Melhores valores para Grefenstette

Os valores encontrados por Grefenstette são diferentes dos encontrados por De Jong.

Goldberg após suas investigações teóricas, fornece uma fórmula para o tamanho da população em função do tamanho (**length**) do cromossomo:

$$\text{Pop} = 1,65 * 2^{0,21 * \text{length}}$$

Aplicando-se essa fórmula obtêm-se populações de 7, 30, 130 indivíduos para cromossomos de comprimento (length) 10, 20, 30, 40 bits, respectivamente.

Pela falta de consistência entre estes resultados empíricos, Schaffer realizou um extenso trabalho para quantificar e identificar a influência destes parâmetros de controle. Em seu trabalho além das cinco funções formuladas por De Jong, outras cinco foram criadas por Schaffer num total de dez problemas. Os experimentos foram feitos com diferentes sementes para cada combinação dos valores de parâmetros do AG, que seguem:

- 1- Tamanho da população com seis valores: (10 – 20 – 30 – 50 – 100 – 200);
- 2- Probabilidade de recombinação com 10 valores (0,05 até 0,95);
- 3- Probabilidade de mutação com 7 valores (0,001 – 0,002 – 0,005 – 0,01 – 0,02 – 0,05 – 0,10);
- 4- Recombinação simples de um ou dois pontos.

Foram feitas 8400 pesquisas com AG ($10 * 6 * 10 * 7 * 2$) onde cada pesquisa abrangeu 10.000 avaliações com as estatísticas gravadas em intervalos de 1000 avaliações.

Essas pesquisas consumiram cerca de 1,5 anos de CPU (maquinas Sun3 e Vax) e os dados obtidos ocuparam cerca de 85 megabytes de espaço em disco.

Os resultados, baseados no desempenho online foram os seguintes.

1- Existe uma forte interação entre o tamanho (pop), taxa de recombinação (pc) e taxa de mutação (pm).

2- Surpreendentemente, a interação da função com os três parâmetros acima foram estatisticamente rejeitados, indicando que o desempenho não depende do tipo de função que está sendo otimizada e sim dos parâmetros com que o AG trabalha.

3- Recombinação de 2 pontos tem um desempenho melhor do que a de 1 ponto apenas para populações entre 50 e 100 cromossomos, sem diferenças significativas para outros tamanhos de populações;

4- Populações de tamanho pequeno (~ 10) são muito sensíveis à taxa de mutação e menos à taxa de recombinação. Um bom desempenho pode ser conseguido com $pm=0,02$ e $pc=0,85$;

5- Com o aumento da população, a sensibilidade à variação da taxa de mutação cai, e um bom desempenho pode ser conseguido com: pm entre 0,002 e 0,005, para $pop=50$;

6- Altas taxas de recombinação melhoram o desempenho com pequenas populações, prejudicando-o com populações maiores.

7- O efeito da seleção e mutação, sem recombinação é mais forte do que se supunha, formando um poderoso algoritmo de busca (Naive Evolution).

Pode-se resumir os resultados do experimento de Schaffer na tabela:

Valores de Schaffer

Tamanho da população	Prob de recombinação	Prob de mutação
20-30	0,75-0,95	0,005-0,01

Figura 3 - Melhores valores para os parâmetros segundo Schaffer

3.1 A DINÂMICA DOS PARÂMETROS

Mesmo com essa exaustiva quantidade de experimentos testados por Shaffer, os parâmetros encontrados como ótimos, estão dentro de uma faixa de valores relativamente grande e, dependendo dos valores escolhidos, podem levar a significativas diferenças no desempenho.

Uma coisa muito importante que esses estudos não levaram em consideração é a dinâmica interna do processo. Pode se dizer que: usar uma mesma taxa de probabilidade para todo o processo de busca em um AG é o mesmo que usar uma mesma marcha de carro, durante todo o trajeto de uma viagem.

A dificuldade em se achar a taxa adequada (p_c , p_m) durante o processamento do AG, pode ter sido o motivo destes estudos terem se centrado em uma taxa fixa.

A mudança dessas taxas (p_c , p_m) influenciam o comportamento do AG, pode-se definir a parametrização das duas principais forças que dirigem o comportamento do AG a Diversidade Populacional e a Pressão Seletiva.

Quanto maior a Diversidade Populacional, mais o AG espalha seus indivíduos pelo espaço de busca procurando locais mais promissores, estes mesmos fatores diminuem os efeitos da pressão seletiva.

Quanto maior a pressão seletiva mais rápido o AG convergirá, portanto fatores que controlam a Pressão Seletiva (função de adequação, fator de escalamento e método de seleção), se contrapõem a Diversidade Populacional.

O ajuste dinâmico, em tempo de execução, destas forças, através da escolha dos métodos e parâmetros que as controlam, pode ser uma maneira de otimizar a velocidade do AG para um ponto de ótimo aceitável.

3.2 A DINÂMICA DO META-AG

Inicialmente, a população é iniciada aleatoriamente para evitar que sejam inicializados super-indivíduos, que por apresentarem algumas características se adaptam melhor ao problema, que pode resultar em mais filhos para esse super-indivíduo, levando a população a se aglutinar em torno deles.

Isto resultaria em uma convergência prematura, que resultaria em uma perda de diversidade populacional. Em geral, essa convergência rápida não é desejada, porque o espaço de busca ainda não foi totalmente vasculhado, alguns pontos promissores com alta adequabilidade, poderiam existir. Então no início do processo é recomendável que valores para parâmetros que mantenham a diversidade populacional, sejam mantidos tais como, uma função de escalonamento que limite o número de filhos por indivíduo, e aumento da probabilidade de mutação.

Uma vez considerado que o espaço de busca esteja bem explorado os parâmetros que mantinham a diversidade populacional são mudados, para acelerar a convergência é utilizada uma função de escalonamento que permita que super-indivíduos gerem mais de um filho e a probabilidade de mutação pode ser diminuída.

3.3 IMPLEMENTAÇÃO

Embora a dinâmica do Meta-AG seja conhecida e bem estudada, não se pode dizer o mesmo do momento em que os parâmetros devem ser trocados. Além disso os

parâmetros que delinarem o Meta-AG sofrerão a influência dos resultados que eles mesmo provocaram a fim de se reajustarem conforme a necessidade.

3.4 PARTES DO META ALGORITMO GENÉTICOS

O primeiro que sugeriu a utilização de um Meta-AG foi Weinberg em 1970.

Embora Weinberg tenha, teoricamente, definido como um AG (Meta-AG) poderia dinamicamente controlar os parâmetros de um AG, ele nunca chegou a implementar (posteriormente, Grefenstette em 1986 implementou-o).

O esquema de um Meta-AG pode ser entendido mais facilmente tendo por base a figura abaixo.

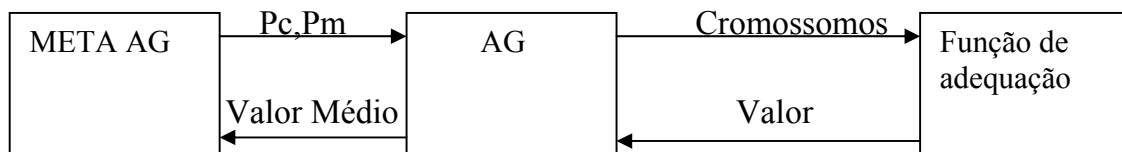


Figura 4 – Layout Meta-AG interagindo com AG

O primeiro procedimento envia (geralmente a cada 1 ou 2 gerações) para o AG a probabilidade de reprodução (P_c) e a probabilidade de mutação (P_m), para que o AG possa construir a próxima geração. Após esse processamento do AG, o AG envia uma mensagem para o META AG com o valor médio de adequação de sua população as probabilidades enviadas pelo META-AG. Assim o META-AG poderá avaliar o rendimento das probabilidades que ele enviou ao AG.

O segundo procedimento, chamado de AG, que resolve o problema modelado com os parâmetros enviados pelo Meta-AG.

O terceiro procedimento é chamado Função de adequação (FA). A cada geração a FA recebe do AG os indivíduos da população para calcular a adequação de cada indivíduo ao problema dado, e retorna ao AG o valor da adequação de cada indivíduo.

Esse valor de adequação será utilizado para calcular a probabilidade do indivíduo reproduzir.

Deve-se ressaltar que o Meta-AG, em si, é um AG e, portanto, também deverá ser parametrizado por, ao menos, uma tripla (pc,pm,pop).

Existe uma contradição o Meta-AG é um AG que controla os parâmetros de outro AG, sendo o próprio Meta-AG por ser um AG parametrizado por um conjunto de parâmetros.

Os parâmetros enviados pelo Meta-AG ao AG devem ser alterados dinamicamente (a cada geração podem ser mudados os parâmetros), para aumentar o desempenho do AG.

3.5 O META-CROMOSSOMO

O Meta-Cromossomo de um AG deve codificar os principais parâmetros enviados a um AG: pc, pm, tipo de cruzamento.



Figura 5 - “Lay-out” de um meta-cromossomo

Outros parâmetros que devem ser passados ao AG tais como (tipo de combinação) devem ser adequadamente codificados no Meta-Cromossomo.

- Representação Interna

Um meta-cromossomo é normalmente representado em binário, como no exemplo abaixo.

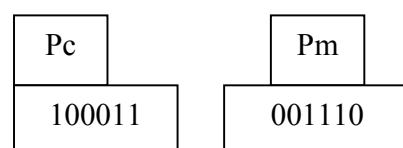


Figura 6 - “Lay-out” de PC e PM

Por se tratar de probabilidades, deve-se tomar cada gene, de cada campo, como sendo um dígito binário de potência negativa de 2. No exemplo:

$$pc=0*2^{-1}+1*2^{-2}+0*2^{-3}.....=0,34$$

$$pm= 1*2^{-1}+1*2^{-2}+0*2^{-3}....=0,76$$

A precisão de valores para pc e pm dependerá do tamanho do cromossomo. Com uma precisão de 4 casas decimais para pc e 4 casas decimais para pm, por exemplo, em um cromossomo codificado em binário, acarretaria um tamanho de cromossomo de 28 genes (que poderiam ser 28 bits ou 28 bytes ou 28 inteiros, conforme a implementação da representação dos genes no cromossomo)

Com 4 casas decimais, o menor valor (positivo) que se pode escrever é 0,0001 e, em binário, são necessários 14 bits para se ter esta precisão.

3.6 A META FUNÇÃO DE ADEQUAÇÃO

A Meta Função de Adequação fará a avaliação do aproveitamento do Meta-cromossomo, utilizado pelo AG para calcular a sua adequação. Isto é, deve-se avaliar a influência de pc e pm atuando no AG, que trata diretamente com o problema do usuário via função de adequação.

Uma das maneiras de se fazer isso, seria computando-se o incremento do valor médio da população do AG-trad.

O algoritmo abaixo foi proposto e utilizado neste trabalho:

- 1- Transferem-se os valores pc e pm dos meta-cromossomos como sendo as novas probabilidades de recombinação e mutação respectivamente, do AG;

- 2- Executa-se N (em geral 1 ou 2) gerações.
- 3- Calcula-se o número médio de colisões ($\text{numero_total}/N$)
- 4- Associa-se o número médio de colisões ao meta-cromossomo que interagiu com o AG.

Esta meta-função de adequação foi projetada de maneira a privilegiar os meta-cromossomos que produzam diminuição no número médio de colisões de horários.

3.7 O PROJETO TURBO ALGORITMO GENÉTICO (TAG)

Um exemplo prático da utilização de um Meta-AG para controlar os parâmetros de um AG dinamicamente, é o projeto TAG. Este projeto foi desenvolvido para ser apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Engenharia, pelo aluno João Carlos Holland de Barcellos.

Esse TC mostra um estudo comparativo feito entre os Meta-AGs (implementados) com seus tipos de Meta-Cromossomos e AGs, para a solução do problema do caixeiro viajante e da obtenção das raízes.

Os Meta-AGs implementados no projeto TAG, iniciam sua meta-população (em torno de 10 meta-cromossomos) com valores aleatórios para pc e pm) com a seguinte restrição: os valores iniciais de pc são sempre superiores a 0,5 e os valores para pm são sempre inferiores a 0,5. Após montada esta população inicial, tais restrições deixam de vigorar e, pc e pm podem assumir quaisquer valores entre 0 e 1. O AG, que está acoplado ao Meta-AG, também é inicializado normalmente e sua população avaliada.

O meta-cromossomo pode codificar além das probabilidades de combinação e de mutação o tipo de seleção e tipo de recombinação.

A Seleção pode ser Proporcional (a probabilidade do cromossomo ser escolhido na reprodução é proporcional ao valor da função de adequação) ou Seleção por Ranking

(a probabilidade do cromossomo ser escolhido varia linearmente com o grau de ordem (Rank) da classificação de sua função de adequação).

A combinação pode ser Multiponto (a quantidade de pontos, bem como sua localização no cromossomo, são escolhidos aleatoriamente entre zero e o número de bits do cromossomo) ou simples (de um único ponto).

O Critério de comparação utilizado foi o número de gerações necessárias para resolver o problema.

Os resultados obtidos mostraram que o Meta-AG mostrou-se em problemas combinatórios levemente superior quando este último possuía um bom ajuste de seus parâmetros, e muito superior ao AG, quando este não estava em seu ideal paramétrico.

Contudo o Meta-AG mostrou-se inferior ao AG em problemas combinatórios. As razões talvez sejam uma maior sensibilidade destes problemas às variações nas taxas de mutação e recombinação em relação aos problemas não combinatórios

Os testes realizados mostraram que, apesar do projeto arrojado, o desempenho do TAG deixou a desejar (provavelmente devido ao caráter altamente disruptivo do operador de recombinação multiponto). Contudo, uma variação sua, o Meta AG-Rank, um Meta-AG com o critério de seleção funcionando adaptativamente (entre Ranking e Proporcional) e método de recombinação simples, não adaptativo, se mostrou, no caso não combinatório, sempre superior aos demais testados e com desempenho relativamente equivalente, no caso combinatório.

4 META-AG PARA DEFINIÇÃO DE HORÁRIOS ACADÊMICOS

A teoria a respeito de Meta-AGs é muito escassa (o que se torna um obstáculo a ser superado), nos poucos trabalhos existentes os parâmetros aconselháveis como ideais estão numa faixa de valores relativamente grande, dependendo do valor escolhido, podem levar a uma significativa diferença no desempenho.

4.1 O FUNCIONAMENTO DO META-AG

O Meta-AG, sendo ele próprio um AG, tem seu funcionamento regido por parâmetros como, população de meta-cromossomos (mcs), probabilidades de mutação (pm) e probabilidade de cruzamento (pc). O AG que executa a partir de parâmetros (pc e pm, codificados nos mcs), também é inicializado e tem a sua população inicial avaliada.

Após esta fase inicial, o Meta-AG envia o primeiro mc para o AG que executa com os parâmetros enviados um número de gerações pré determinado pelo usuário, após a execução é calculado para o mc a sua adequação. O processo é repetido sucessivamente.

É importante lembrar que cada mc trabalha com a população do AG deixada pelo mc anterior, isso significa que a população do AG não é reinicializada para cada mc que seja testado, diminuindo consideravelmente um possível “over-head”.

4.2 PROBLEMA MOTIVADOR

A definição de uma grade de horários sempre foi um problema considerável em que a utilização de técnicas heurística (baseadas na experiência) era uma das poucas soluções, tais técnicas falhavam no que deveria ser a sua principal qualidade (facilitar a resolução do problema), geralmente um profissional (na maioria das vezes um professor) precisava manualmente definir os horários o que levava várias horas e era certeza de dor de cabeça, várias restrições deveriam ser levadas em conta tais como, folga de professores, distância mínima entre as aulas da mesma disciplina (preferencialmente, é indesejado que uma disciplina de quatro créditos seja ministrada em um mesmo dia), tempo mínimo de descanso para o professor (professores que dão aula nos últimos períodos da noite não podem dar aula nos primeiros períodos do próximo dia), é para facilitar esse processo de definição de horários que os AGs são um bom método. A partir de um conjunto de grades de horários codificadas (chamadas de cromossomos), a cada geração são calculadas as colisões existentes em toda a população de cromossomos, esse processo é repetido sucessivamente procurando achar horários ótimos (sem colisões) ou bons (poucas colisões).

4.3 O META AG IMPLEMENTADO

O Projeto foi batizado de META ALGORITMO PARA DEFINIÇÃO DE HORÁRIOS ACADÊMICOS, o objetivo desse projeto é a elaboração de um Meta-AG para enviar dinamicamente a cada 1 ou 2 gerações parâmetros (pc,pm), ao AG dos horários da informática . A otimização dos parâmetros ocorre dinamicamente ao longo das gerações, através da função de adequação é visado avaliar (no caso dessa implementação o numero médio de colisões da população do AG), todos os mcs que parametrizam o AG. Após terem sido avaliados os mcs (codificação de pc, pm) o melhor deles será escolhido para fazer parte da nova geração do Meta-AG (mpop) e o pior será excluído, os demais mcs serão obtidos através de cruzamento binário. Com isso espera-se otimizar os parâmetros para o problema apresentado (AG) ao longo das gerações.

A próxima etapa a ser realizada é a mutação dos mcs, geralmente uma pequena probabilidade dos genes de um mc mudar o seu valor, nessa implementação os genes dos mcs são codificados em binário utilizando vetores, ou seja as probabilidades a serem enviadas ao AG podem assumir valores que variam de 0% a 99%. Com a codificação binária um mc que represente 55% para probabilidade de cruzamento e 2% para probabilidade de mutação será codificado da seguinte maneira.

Representação Interna dos mcs

Probabilidade de Combinação	Probabilidade de Mutação	Adq
0110111	0000010	10

Figura 7 - “Lay-out” do mc do Meta-AG para Definição de Horários Acadêmicos

Por se tratar de probabilidades, deve-se tomar cada gene, de cada campo, como sendo um dígito binário de potência de 2.

$$P_c = 1x2^0 + 1x2^1 + 1x2^2 + 1x2^3 \dots$$

$$P_m = 1x2^0 + 1x2^1 + 1x2^2 + 1x2^3 \dots$$

Após a mutação os mcs estão prontos para enviarem parâmetros ao AG.

4.4 A META FUNÇÃO DE ADEQUAÇÃO

A função de adequação de um Meta-AG deve avaliar ao máximo a influência dos parâmetros p_c e p_m codificados nos mcs, privilegiando os melhores mcs que terão mais chances de estarem presentes nas próximas meta-populações (mpop).

Nesta implementação foi levado em consideração o número médio de colisões dos cromossomos do AG (todos os horários do AG são avaliados após isso é dividido o

número de colisões total pelo numero de horários **desempenho online de De Jong**, esse valor é associado ao mc que acabou de interagir com o AG.

4.5 ESQUEMA DE CRUZAMENTOS

Os mcs que remanescerem ao processo de seleção do melhor mc e exclusão do pior mc, serão alvo de cruzamentos binários para definição dos outros mcs que farão parte da próxima mpop.

Meta-cromossomo1: 1,0,~~0~~,1,1,0,1
 Meta-cromossomo2: 0,1,~~0~~,0,0,1,0
 Filho1: 1,0,0,0,0,1,0
 Filho2: 0,1,0,1,1,0,1

O ponto de corte agiu no exemplo acima na posição 2, é importante explicar que o ponto de corte varia aleatoriamente a cada novo cruzamento da posição 2 a 6.

4.6 ESQUEMA DE MUTAÇÃO

Após o cruzamento a última etapa para a geração da nova mpop são as mutações. Todos os genes binários dos mcs tem a probabilidade de terem os seus valores alterados, essa probabilidade é definida pelo usuário, geralmente é escolhido um pequeno valor, (em torno de 1% e 5%) porque uma mutação muito alta pode destruir bons mcs já encontrados. A mutação é mostrada conforme layout abaixo. Após a mutação a nova mpop é gerada e os mcs estão prontos para enviarem parâmetros ao AG.

Meta-cromossomo1: 0,1,1,0,0,1.
 Meta-cromossomo1 mutado: 0,1,1,0,0,0.

Como observado o gene 7 sofreu mutação e seu valor foi alterado de 1 para 0.

4.7 CONFIGURAÇÃO E TESTES

Para realizar um estudo comparativo entre Meta-AG e AG, foram feitos vários experimentos, no qual o modelo proposto e publicado na III mostra de software Acadêmico da PUCRS- Campus Uruguaiana-Rs foi utilizado.

Após alguns testes preliminares durante a construção do modelo foi percebido que o número de gerações muito alto não influenciava no desempenho do Meta-AG e nem do AG, os mesmos tendem a estagnar após um número pequeno de gerações parando de gerar novas soluções. Boas soluções ficam em torno de 100 gerações, feito os testes preliminares foram definidos os parâmetros para o estudo comparativo.

4.8 CONFIGURAÇÃO DO AG

Os testes foram realizados com uma população de 300000 cromossomos para o AG, os parâmetros do AG foram fixados em 1% para pm e 65% para pc. Foram escolhidos esses valores para pm e pc por serem, segundo a bibliografia, no assunto, os melhores.

4.9 CONFIGURAÇÃO DO META-AG

Nos testes preliminares não ficaram registradas relações entre o tamanho da meta-população e o desempenho do Meta-AG, portanto no estudo comparativo a população de mcs ficou fixada em 10 mcs. No Meta-AG a codificação de pc e pm iniciais dos mcs não ultrapassam 2% para pm e não são menores que 90% para pc, após inicializada a primeira mpop tais restrições deixam de vigorar e os valores podem variar entre 1% e 99% nas demais mpops.

Altas taxas de cruzamento na primeira geração do Meta-AG são preferíveis para que o material genético dos cromossomos seja bem espalhado, busca-se também obter um bom ajuste inicial (baixo numero de colisões) devido a estagnação do modelo em

soluções, após um pequeno número de gerações (busca-se a convergência rápida antes da estagnação).

Uma característica introduzida é a de Loop-Aval, ela determina quantas gerações cada mc irá interagir com o AG. Hipoteticamente quanto maior o Loop-Aval mais adaptados estarão os meta-cromossomos ao problema. Após isso pc e pm do próximo mc é enviado perfazendo o processo.

O Meta-AG como um AG também tem seus mcs vasculhados pela mutação, foi escolhido 1% para mutação do Meta-AG. O tamanho da população do AG e o número de gerações são respectivamente 300000 e 100.

4.10 CRITÉRIO DE COMPARAÇÃO

O critério de comparação utilizado foi o horário com menos colisões em grupos de 10 gerações (desempenho online).

Poder-se ia também adotar o tempo de processamento gasto para resolver o problema. Existe uma diferença: o tempo de processamento leva em conta o “Overhead” do processamento interno do algoritmo (sua eficácia) enquanto que o número de gerações preocupa-se mais com a eficiência do mesmo.

A diferença é sutil: Se o problema a ser otimizado for complexo e, como o algoritmo interno dos AG é relativamente pouco consumidor de CPU, o consumo de processamento ficará quase que inteiramente dedicado na avaliação da função objetivo. Isto é, usando o número de gerações e não tempo de CPU dá-se ênfase à aplicações de AG voltadas para a otimização de problemas complexos pois, procedendo desta maneira, exclui-se o tempo gasto em seu processamento interno.

4.11 TESTES

O modelo proposto foi testado nos cursos do Departamento de Informática da FACI do Campus Uruguaiana da PUCRS.

O Meta-AG foi implementado na linguagem pascal, no ambiente de programação Delphi, executado em processador Intel Pentium 4 2000Ghz e 256Mb de RAM.

4.12 RESULTADOS

Experimento 1- AG

Parâmetros: $P_c=65\%$, $P_m=1\%$, População 300000, Gerações 100

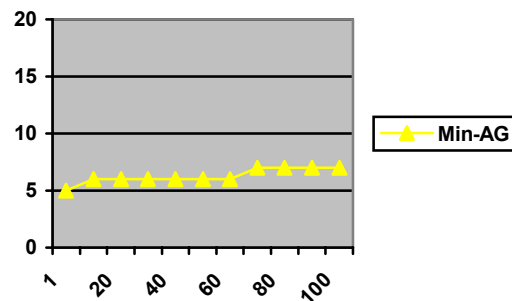


Figura 8 – Gráfico do experimento com AG

Como pode ser observado no gráfico do experimento acima, o melhor ajuste foi do primeiro cromossomo com 5 colisões, após isso entre a 10ª geração e a 60ª houve uma estagnação com 6 colisões para o melhor resultado, entre a 70ª e a 100ª geração o AG foi ainda pior não conseguindo gerar um horário com menos de 7 colisões.

Experimento 2- Meta-AG

Parâmetros: Meta-população 10, $P_c\text{-meta} < 2\%$, Loop-Aval 2, demais parâmetros iguais aos do AG.

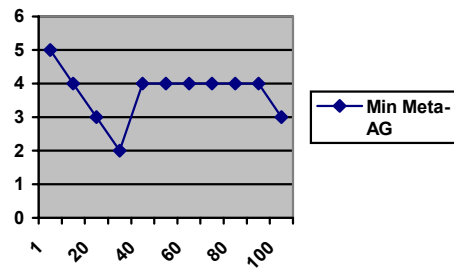


Figura 9 – Gráfico do experimento com Meta-AG

Como pode ser observado, após uma rápida queda no número mínimo de colisões do experimento acima, o melhor ajuste foi obtido entre a 30^a e a 40^a geração com 2 colisões, após isso entre a 40^a geração e a 90^a houve uma estagnação com 4 colisões para o melhor resultado. No último grupo de gerações observado houve uma melhora em apenas uma colisão.

5 CONCLUSÃO

Meta-AGs fornecem uma poderosa ferramenta para controle dos parâmetros de um AG, os resultados foram satisfatórios, apesar de não ter se obtido a convergência, foram encontrados horários bons com poucas colisões 4,3,2. O horário com duas colisões muito provavelmente possa ser convertido em um horário sem colisões apenas permutando as disciplinas colidentes. Os bons horários obtidos podem ser em poucos minutos acertados por um professor utilizando técnicas heurísticas.

6 TRABALHOS FUTUROS

A elaboração de outras funções de adequação na qual os mcs seriam avaliados pelo menor número mínimo de colisões durante o processamento do mc (desempenho offline) e a elaboração de técnicas heurísticas para resolução de horários bons que não atingiram a convergência.

7 APÊNDICE A : O META-AG PROPOSTO E SUAS TELAS

O programa META-AG PARA DEFINIÇÃO DE HORÁRIOS ACADÊMICOS, tem como objetivo a realização de um estudo comparativo entre a solução com Meta-AG e AG, foi desenvolvido no ambiente Delphi, roda em plataforma Windows.

Foram feitos vários procedimentos, quando obtidos horários bons (menos de 5 colisões) ou ótimos os resultados são gravados em arquivos de texto para posterior verificação, também podem ser carregados parâmetros para a execução através de arquivos, com o objetivo de que o usuário não se prenda ao computador inserindo parâmetros para uma nova interação do sistema.

1-Esta é a primeira tela do programa uma opção deve ser escolhida

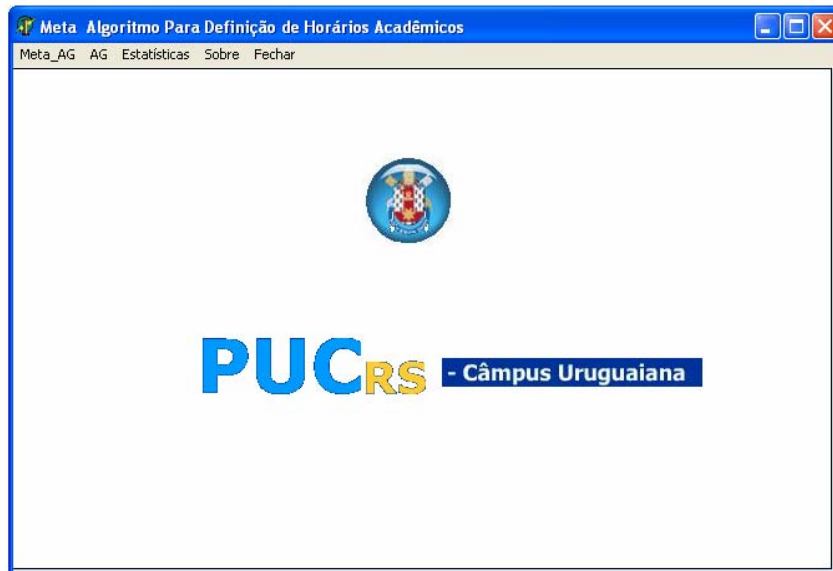


Figura 10 – Tela principal

2-Escolhendo Meta_AG essa tela é apresentada para definição das opções de execução

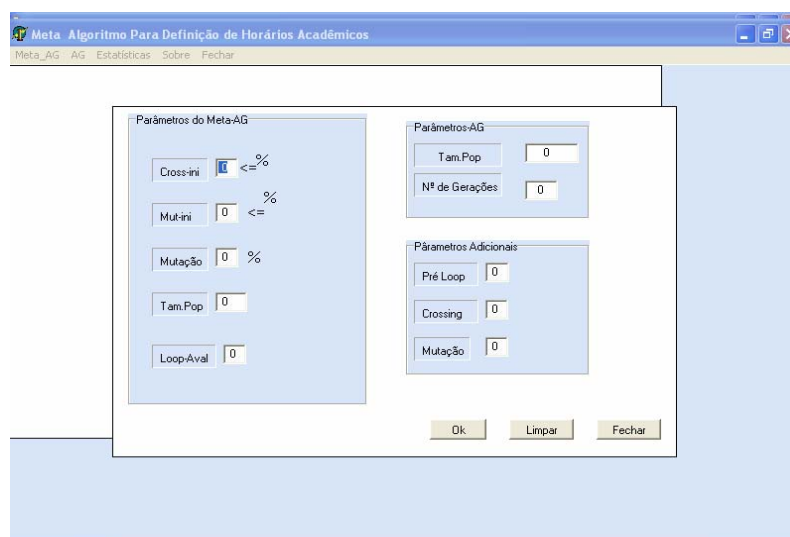


Figura 11 – Tela de opções do Meta-AG

Botão ou Campo	Descrição
Cross- $ini \geq$	Inicializa (pc) dos mcs da primeira mpop
Mut- $ini \leq$	Inicializa (pm) dos mcs da primeira mpop
Mutação	(pm) do Meta-AG
Meta-Tam-Pop	Tamanho da população de mcs
Loop-Aval	Nº de gerações de iteração com o AG de cada mc
Tam-Pop	Tamanho da população do AG
Nº Gerações	Nº de gerações do experimento
Pré-loop	Nº de gerações antes de ativar o Meta-AG
Pré-Crossing	Pc do pré loop
Pré-Mutação	Pm do pré loop
Ok	Aceita parâmetros
Limpar	Limpa todos as caixas de texto
Fechar	Fecha a tela de opções do Meta-AG

3-Escolhendo AG essa tela é apresentada para definição das opções de execução

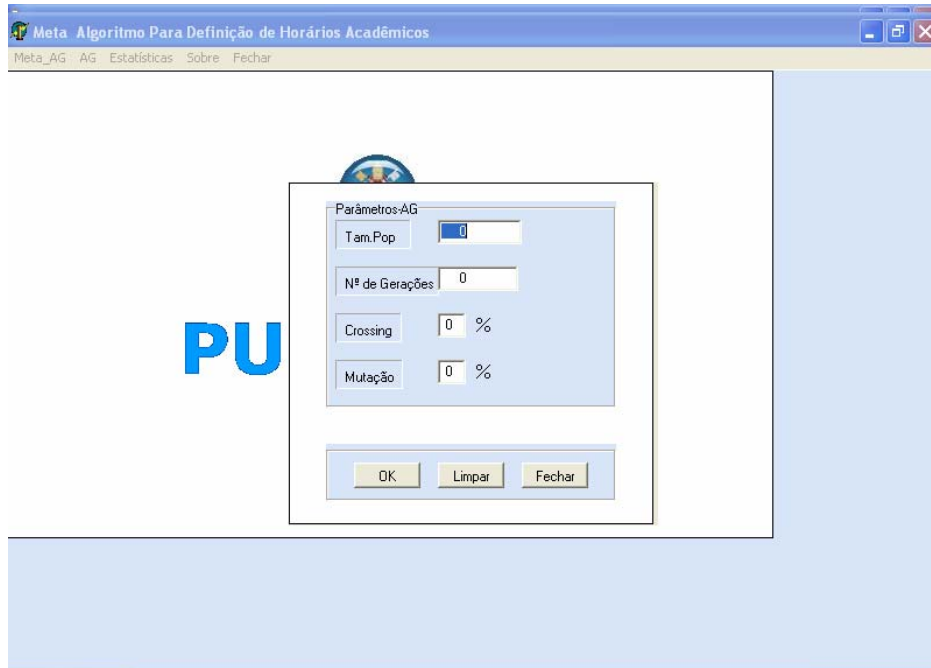


Figura 12 – Tela de opções do AG

Botão ou Campo	Descrição
Tam-Pop	Tamanho da população do AG
Nº de Gerações	Nº de gerações do experimento
Crossing	Pc dos cromossomos
Mutaç�o	Pm dos cromossomos
OK	Aceita par�metros
Limpar	Limpa todos as caixas de texto
Fechar	Fecha a tela de opç�es do

	Meta-AG
--	---------

4-Tela que mostra o resultado da execução do programa

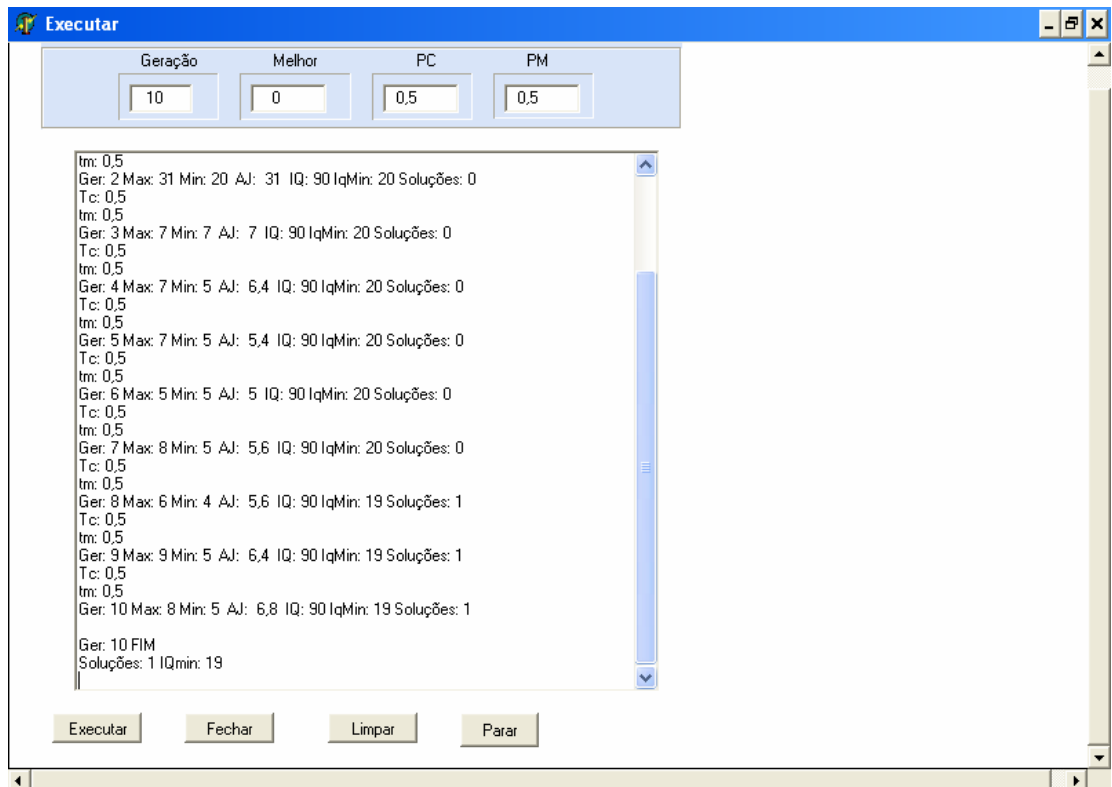


Figura 13 – Tela de Execução

Botão ou Campo	Descrição
Executar	Começa a execução do Experimento
Fechar	Fecha tela de Execução
Limpar	Limpa campos de texto e richedit
Parar	Para a execução do Experimento

Geração	Mostra a geração corrente
Melhor	Mostra melhor ajuste encontrado
Pc	Probabilidade de cruzamento usada
Pm	Probabilidade de mutação usada

5-Tela de Estatísticas: Mostra estatísticas dos Experimentos

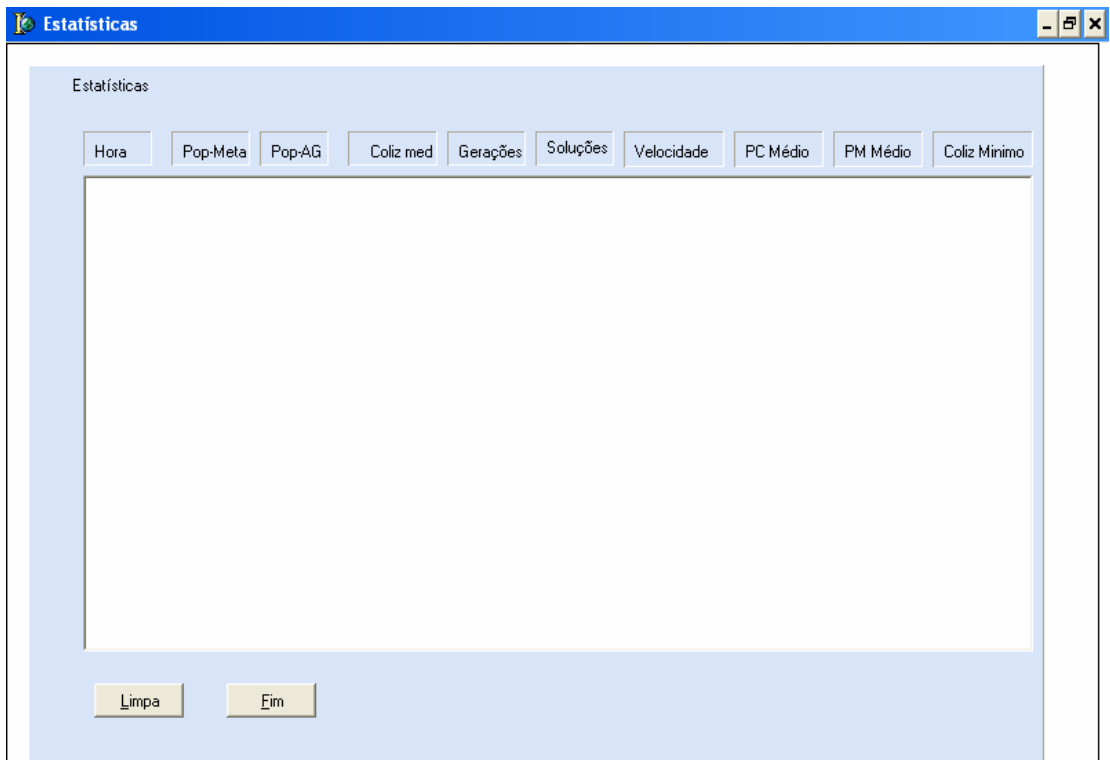


Figura 14 – Tela de resultados

Botões	Descrição
Limpa	Limpa o richtext
Fim	Fecha tela de Estatísticas

8. CONSIDERAÇÕES FINAIS

Neste trabalhos foram mostrados os resultados e características obtidos da implementação de um Meta-AG como proposto na cadeira de TC1.

No capítulo 2 foi apresentado uma revisão sobre os AGs bem como os parâmetros envolvidos em sua modelagem.

No capítulo 3 foram abordados conceitos referentes a Meta-AGs tais como o seu funcionamento, partes que o compõem, motivação para a pesquisa, também foi explicado rapidamente o projeto TAG por ser o trabalho no qual nos inspiramos para o desenvolvimento do Meta-AG em questão.

No capítulo 4 foi feita uma explanação sobre o Meta-AG desenvolvido mostrando suas características, e os resultados do estudo comparativo que foi o objetivo deste trabalho.

No capítulo 5 foi feita a conclusão do autor sobre os resultados obtidos nos testes.

No capítulo 6 foram citados trabalhos futuros a serem realizados referentes a ferramenta desenvolvida.

No capítulo 7 foram mostradas as telas gráficas do projeto bem como suas funcionalidades.

No capítulo 9 foram transcritas as referências bibliográficas utilizada no tc2.

9- REFERÊNCIAS BIBLIOGRAFICAS

Rocha Fernandes, Anita Maria da. Inteligência Artificial: 1^o, Florianópolis SC: Visual Books, 2003, 115.

Holland de Barcellos, João Carlos. Algoritmos Genéticos Adaptativos um Estudo Comparativo, 2000.
Tese (Mestrado em Engenharia) – Escola Politécnica de São Paulo

Annes, Ricardo; Delano Tobal da Paz, Madison. Simpósio de Informática e Mostra de Software Acadêmico da PUCRS – Câmpus Uruguaiana – RS, 8, 2003, Uruguaiana, MODELO PARA OTIMIZAR HORÁRIOS COM ALGORIMOS GENÉTICOS, Uruguaiana: Hífen, 2003, 19p.

ENGO, Frank. COMO PROGRAMAR EM Delphi 3. 1^a. São Paulo: Makron Books, 1997. 388p.

RINALDI, Robert. TURBO PASCAL. 10^a. São Paulo: Érica Ltda, 1998. 525.

