

# INTERFACE COMPUTACIONAL PARA APLICAÇÕES DE ALGORITMO GENÉTICO

Máx Ricardo Pantoja da Trindade<sup>1</sup>  
Rejane Barros Araújo<sup>2</sup>

*RESUMO - O objetivo deste trabalho é apresentar um ambiente computacional, com interfaces amigáveis para ensino de Algoritmos Genéticos (AG's), de forma a facilitar ao usuário um melhor entendimento da aplicação do AG. Como o AG é um método utilizado basicamente para resolver problemas em pesquisa numérica, otimização de funções e aprendizagem de máquina, optou-se por escolher quatro aplicações diferentes usando o algoritmo genético: O Problema do Caixeiro Viajante, O Dilema do Prisioneiro, Otimização de Funções e a Sintonização de um PID (Controlador Proporcional Integral Derivativo).*

## INTRODUÇÃO

Este artigo trata de *Algoritmos Genéticos*, que são definidos como uma família de modelos computacionais inspirados na teoria da evolução, que incorporam uma solução potencial para um problema específico numa estrutura semelhante a de um cromossomo e aplicam operadores de seleção e de "cross-over" a essas estruturas de forma a preservar informações críticas relativas à solução do problema (Barreto, 1999; Dias, 1998). Normalmente os AG's são vistos como otimizadores de funções, embora a quantidade de problemas para os quais os AG's se aplicam seja bastante abrangente. Uma das vantagens dos algoritmos genéticos é a simplificação que eles permitem na formulação e solução de problemas de otimização.

Em todas as quatro aplicações de AG, descritas a seguir, a estratégia adotada é a elitista (Sprinner, 1992 & Dias, 1998), na qual o indivíduo mais adaptado ao meio é sempre mantido na próxima geração. Uma outra estratégia adotada, e que se mostrou bastante eficaz no aumento da velocidade de convergência do AG, foi a aplicação de uma nova mutação com uma alta probabilidade (30-50%). Esta mutação adicional somenSalesman Problem – TSP) consiste em visitar  $n$  cidades. O caixeiro sai de uma determinada cidade (por exemplo, cidade 1), vai visitan-

do cada uma das  $n-1$  cidades restantes, sem passar novamente pela cidade já visitada, e retorna para a cidade 1. Dessa maneira, a trajetória descrita pelo caixeiro forma um percurso fechado. O objetivo deste problema clássico de otimização é determinar a seqüência de cidades a serem visitadas, de forma que a distância total percorrido pelo caixeiro seja a menor possível (Sprineer, 1992).

No Problema do Dilema do Prisioneiro existem dois indivíduos (Alice e Bob) que são mantidos presos por terem praticado um crime em conjunto.

Os dois prisioneiros são mantidos em selas separadas, sem nenhuma possibilidade de comunicação entre ambos. A polícia oferece para Alice o seguinte trato: se ela confessar o crime e concordar em testemunhar contra Bob, ela ganha liberdade condicional, enquanto que Bob passa 5 anos na cadeia;

Entretanto, se ao mesmo tempo Bob confessar o crime e concordar em testemunhar contra Alice, esta última será desacreditada, e cada um deles irá cumprir 4 anos de pena por admitirem a culpa.

Alice é informada de que um trato nas mesmas condições está sendo oferecido a Bob. Ambos sabem que se permanecerem em silêncio e não se acusarem mutuamente, poderão ser con-

<sup>1</sup> Professor Adjunto I da UNAMA – Mestre em Engenharia Elétrica (Automação) pela UFPA

<sup>2</sup> Engenheira Elétrica – Mestra em Engenharia Elétrica pela UFPA

denados somente a uma pena mais branda de 2 anos na cadeia, cada um.

Portanto existem duas escolhas: Cooperar (C) mantendo silêncio, ou trair (D) (to defect) acusando-o.

Os resultados possíveis para uma única partida são descritos na matriz de recompensa ilustrada na Tab. 1.

Tabela 1: Matriz de recompensa

		Jogador B	
		Cooperar (C)	Trair (D)
Jogador A	Cooperar (C)	(3,3)	(0,5)
	Trair (D)	(5,0)	(1,1)

O dilema consiste no fato de que se ambos os jogadores traírem, cada um recebe uma recompensa menor de que aquela que ganharia caso os dois cooperassem (Fogel, 1995).

Na Otimização de Função, implementou-se um AG para encontrar os pontos de máximos ou de mínimos de uma função de duas variáveis.

Para o caso da sintonização do PID, O AG é utilizado para encontrar os valores ótimos de  $K_p$  (Ganho Proporcional),  $K_i$  (Ganho Integral) e  $K_d$  (Ganho Derivativo), para uma planta qualquer de no máximo 3ª ordem, que minimize o sinal de erro entre a saída da planta e o sinal de referência (Rensburg, 1998), como mostra a Fig. 1.

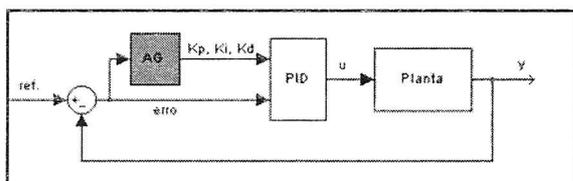


Figura 1: Sistema em Malha Fechada com uma Planta, Controlador PID e AG

Nesta implementação, utilizou-se os modelos discretos (método de Tustin) da planta e do controlador.

## VISÃO GERAL DO AMBIENTE

Os Ambientes Computacionais para Ensino de Engenharia vem ganhando espaço de forma progressiva e significativa na área educacional e até mesmo nas industriais, para treinamento de pessoal. Este recurso é capaz de substituir vários aparelhos eletrônicos de propósito específico e aparelhos de medições, por computadores de propósito geral munidos de *hardware e software* específicos para o tratamento dos dados. Há vários aspectos vantajosos no emprego destes ambientes, como: flexibilidade, versatilidade e principalmente boa relação entre custo e benefício. Hoje existe uma forte demanda para desenvolvimento de ambientes computacionais de ensino de engenharia de propósito geral ou específico, que venham atender à realidade das instituições de ensino de engenharia do país.

O ambiente foi criado levando-se em consideração aspectos como: modularidade, simplicidade de uso e facilidade de manutenção.

A modularidade permite que futuras extensões, manutenções e atualizações possam ser realizadas de forma rápida e segura. A *interface* simples e intuitiva ao usuário deve propiciar o mínimo de tempo possível para adaptação ao ambiente, evitando-se assim possíveis desmotivações por parte do mesmo.

O ambiente foi desenvolvido no Matlab & Guide (Mathworks, 1997), pelo mesmo apresentar várias características desejadas, como: fácil manipulação pelos usuários (por ser bastante difundido no meio acadêmico), ter funções definidas (pois facilita a implementação do AG e diminui o esforço computacional) e facilidade de desenvolver interfaces amigáveis para o usuário.

## FUNCIONALIDADE DO AMBIENTE

Neste tópico são apresentadas de forma sistêmica as interfaces criadas. A Fig. 2 apresenta a tela principal do ambiente, onde pode-se observar os botões que dão acesso as quatro aplicações atuais.

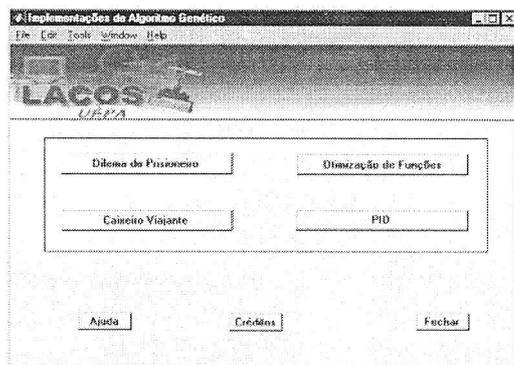


Figura 2: Tela Principal

### Dilema do Prisioneiro

Na interface do Dilema do Prisioneiro o usuário pode escolher dois tipos de seleção para implementar o AG: roleta simples ou roleta ponderada. O botão **O Que é o Dilema do Prisioneiro?** dá acesso a uma pequena descrição do problema para que o usuário tenha conhecimento do mesmo. O usuário poderá escolher os valores de probabilidade de mutação ( $P_m$ ), probabilidade de cruzamento ( $P_c$ ), tamanho da população e número de gerações. Depois da sintonização de todos os parâmetros, existe um botão para iniciar o AG e um outro botão para visualizar os gráficos da média e do melhor indivíduo por geração, como pode ser visto na Fig. 3.

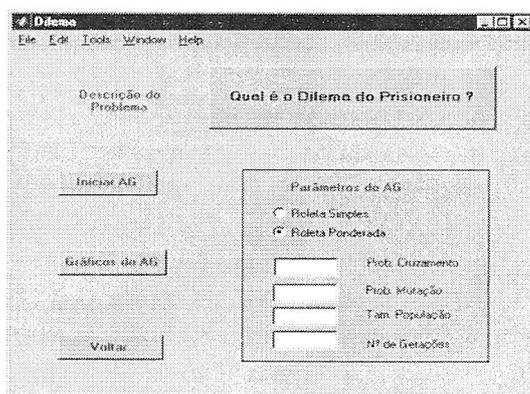


Figura 3: Interface do Dilema do Prisioneiro

Com esta implementação, o usuário tem a oportunidade de analisar o problema do dilema com mais de um tipo de seleção e valores diferentes de  $P_m$ ,  $P_c$ , tamanho da população e número de gerações, podendo analisar a influência dos mesmos no comportamento do AG.

### Caixeiro Viajante

Na interface do Caixeiro Viajante, o usuário pode escolher os parâmetros do AG, ou seja,  $P_m$ ,  $P_c$ , tamanho da população e número de gerações, assim como o número de cidades a serem visitadas. O usuário poderá também escolher o tipo de seleção (roleta simples ou ponderada), cruzamento (OX ou PMX) e mutação (inversa ou translocação) a ser utilizado (Sprineer, 1992). Após o usuário clicar no botão **Mapa**, aparecerá na tela em cinza o mapa do Estado do Pará, com as cidades dispostas aleatoriamente. A partir desse ponto, pode-se iniciar o AG, clicando-se no botão **Iniciar**. O melhor caminho é mostrado no mapa a cada 10 gerações. A adaptação média da população e do melhor indivíduo podem ser vistas após a finalização do AG, através do botão **Gráficos** da Fig. 4.

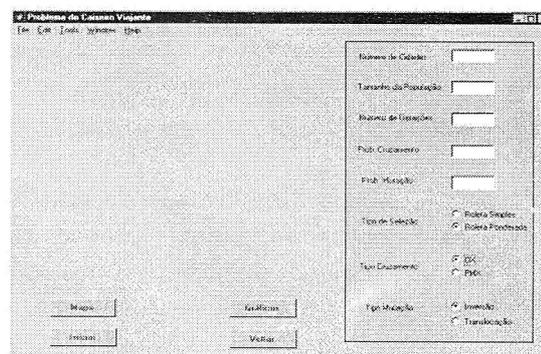


Figura 4: Interface do Caixeiro Viajante

### Otimização de Função

Na interface da Otimização de Função, o usuário poderá encontrar os pontos de máximos ou mínimos de qualquer função de duas variáveis. Nesta interface, o usuário precisará digitar a fun-

ção desejada e escolher se quer maximizar ou minimizar. Além disso, deve-se definir os pontos de máximos ( $X_{1max}$  e  $X_{2max}$ ) e de mínimos ( $X_{1min}$  e  $X_{2min}$ ), que são os intervalos de busca da solução.

Antes de iniciar o programa de Otimização de Função, o usuário deve escolher os parâmetros do AG, como  $P_m$ ,  $P_c$ , tamanho da população e número de gerações. O tipo de seleção também é uma outra flexibilidade da interface, ou seja, o usuário pode fazer a análise do AG tanto com seleção simples, como seleção ponderada. Depois de todos os parâmetros serem estabelecidos, o usuário deve clicar no botão **Iniciar AG** da Fig. 5, e aparecerá na tela branca um gráfico 3D da superfície da função sob estudo e seu ponto de mínimo ou de máximo obtido pelo GA, a cada 10 gerações. O fitness médio e do melhor indivíduo também podem ser vistos, após o usuário clicar no botão **Gráficos do AG**.

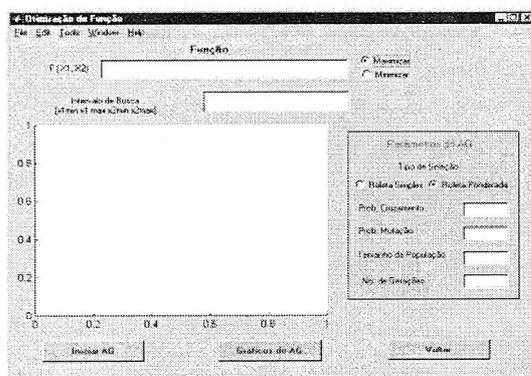


Figura 5: Interface da Otimização de Função Sintonização do PID

Na interface da Sintonização de um PID (Controlador Proporcional Integral Derivativo), o usuário poderá obter os melhores valores de  $K_p$  (Ganho Proporcional),  $K_i$  (Ganho Integral) e  $K_d$  (Ganho Derivativo), usando o AG para qualquer planta de, no máximo 3ª ordem, bastando para isso escolher os parâmetros do AG, ou seja,  $P_m$ ,  $P_c$ , tamanho da população e número de gerações. O usuário deve escolher também a planta a ser controlada, tendo que digitar o numerador, o denominador e o período de amostragem. Deve-se ter a preocupação de escolher adequadamente

os limites de  $K_p$ ,  $K_i$  e  $K_d$  (Datta, 2000). Após o usuário clicar no botão **Iniciar AG**, aparecerá na tela branca, a cada dez gerações, o sinal da saída da planta controlada pelo PID, cujos parâmetros são obtidos pelo AG. Para visualizar o fitness médio da população e o do melhor indivíduo, deve-se clicar no botão **Gráficos do AG** da Fig. 6.

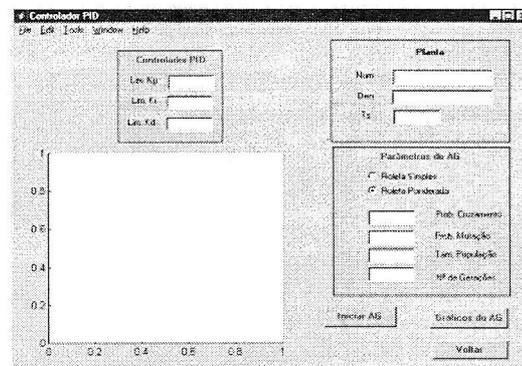


Figura 6: Interface da Sintonização do PID

## SIMULAÇÕES

Foram realizadas algumas simulações utilizando os recursos das interfaces, para se mostrar a funcionalidade de cada uma delas no estudo dos Algoritmos Genéticos. Em seguida, serão mostrados os resultados de uma simulação de cada aplicação.

### Dilema do Prisioneiro

Para o Dilema do Prisioneiro escolheu-se os seguintes parâmetros do AG:  $P_m = 0,01$ ;  $P_c = 0,85$ ; tamanho da população de 20 indivíduos, 400 gerações e seleção do tipo roleta simples. Com esses parâmetros, obteve-se os gráficos do fitness médio da população e *fitness* do melhor indivíduo, como pode ser visto na Fig. 7.

Fazendo-se uma análise do gráfico observa-se que a partir da geração 50 o AG converge, mantendo os indivíduos melhores adaptados nas próximas gerações. Os resultados são bem satisfatórios, pois o AG consegue convergir para um valor ótimo, em poucas gerações.

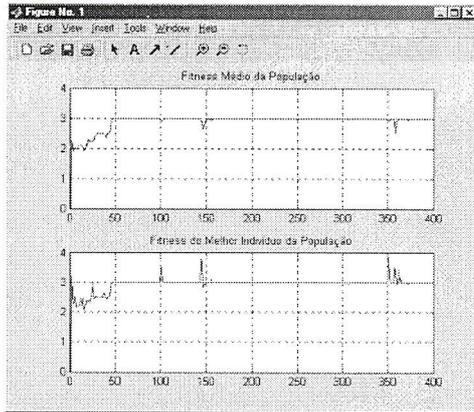


Figura 7: Gráfico da Média e do Melhor Indivíduo por Geração para o Dilema do Prisioneiro

### Caixeiro Viajante

No caso do Caixeiro Viajante, deseja-se determinar o menor caminho percorrido pelo caixeiro para visitar 15 cidades, dispostas aleatoriamente no mapa do Estado do Pará. Os pontos em azul indicam as localizações das cidades e as linhas em amarelo definem o menor caminho percorrido pelo caixeiro, após execução do AG.

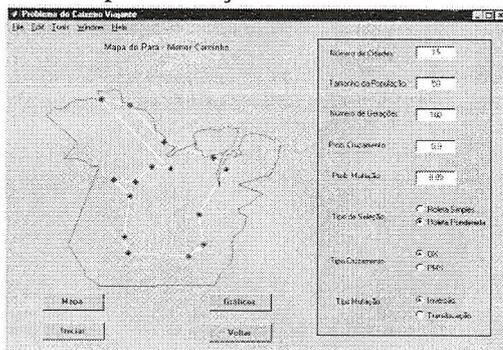


Figura 8: Interface do Caixeiro Viajante com os Parâmetros Escolhidos e o Menor Caminho obtido pelo AG

Os parâmetros do AG, o tipo de seleção, cruzamento e de mutação podem ser vistos na Fig. 8. Os resultados do *fitness* médio da população e do melhor indivíduo são mostrados na Fig. 9. Nela, observa-se que o menor caminho, ou seja, o maior valor de *fitness* do melhor indivíduo é obtido na geração 37, isto mostra que para os parâmetros utilizados, o AG pode ser considerado eficiente.

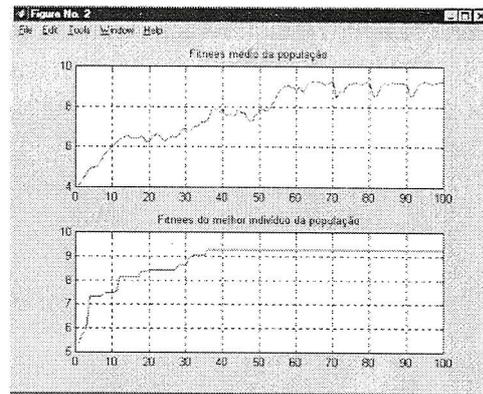


Figura 9: Gráfico da Média e do Melhor Indivíduo a cada Geração para o Caixeiro Viajante

Através de várias simulações, o usuário poderá verificar que os tipos de cruzamento e de mutação não exercem influências significativas na velocidade de convergência do AG. Por outro lado, a utilização da estratégia elitista neste problema é de extrema importância, pois garante que o melhor indivíduo (menor caminho) não seja perdido durante os processos de seleção, cruzamento e mutação. Já o tipo de seleção tem uma grande influência na velocidade de convergência do AG.

### Otimização de Função

Para a Otimização de Função, digitou-se uma função de duas variáveis e escolheu-se a opção maximizar, definindo-se também o intervalo de busca da solução, como mostrado na figura 10.

A Fig. 10 mostra os parâmetros necessários para executar o AG. Depois de iniciar o AG, observa-se um gráfico em 3D, onde o ponto em vermelho é o valor obtido pelo AG e o ponto em azul é o valor máximo da função.

Na Fig. 11, observar-se o *fitness* médio e *fitness* do melhor indivíduo da população por geração. Neste último, verifica-se que o ponto máximo da função é obtido na geração 18. A partir da geração 35, observa-se através do gráfico, *fitness* médio, que há uma grande variação na curva da adaptação média da população. Isto ocorre devido a estratégia adotada nas implementações do AG,

na qual há um grande aumento na probabilidade de mutação dos indivíduos da população, quando o valor do fitness do melhor indivíduo é mantido por algumas gerações.

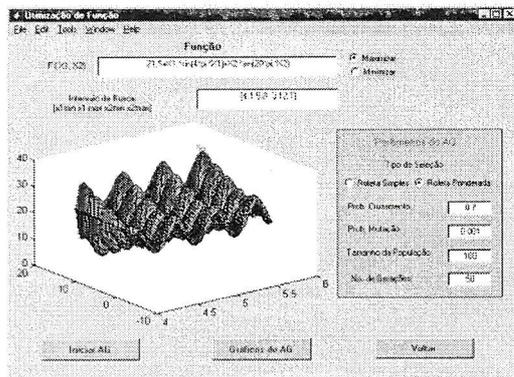


Figura 10: Interface da Otimização da Função com os Parâmetros Necessários.

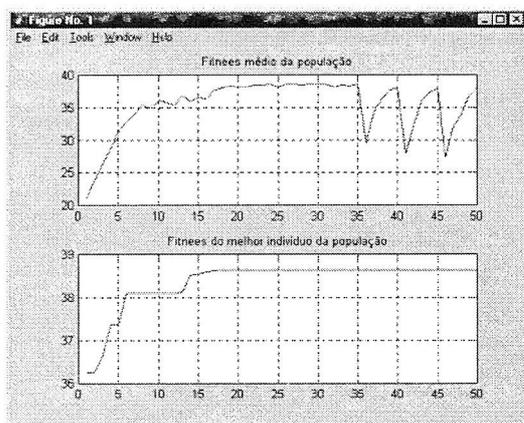


Figura 11: Gráfico da Adaptação Média e do Melhor indivíduo por Geração da Otimização de Função

#### Sintonização do PID

Na sintonização do PID, utilizou-se uma planta de 2ª ordem com pólos complexos e conjugados em  $-0,6 \pm j1,9079$  e sem zeros, cuja resposta ao degrau é mostrado na Fig. 12. Na implementação do Algoritmo Genético para esta aplicação, a planta e o controlador PID são discretizados com período de amostragem de 0.05 segundos.

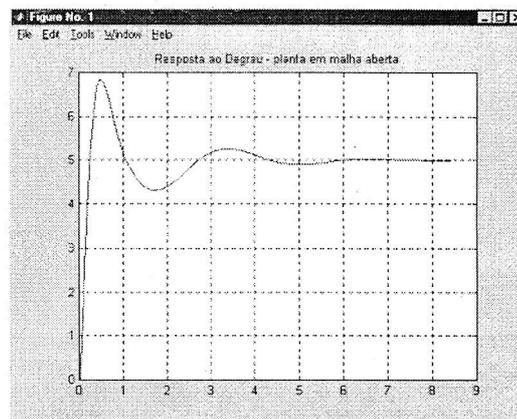


Figura 12: Gráfico do Sinal de Saída da Planta em Malha Aberta

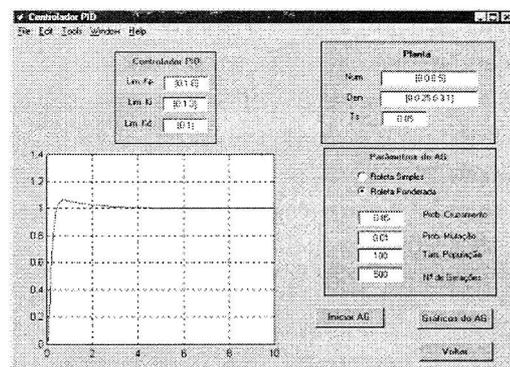


Figura 13: Interface da Sintonização do PID com o Gráfico de Saída da Planta com o Controlador Sintonizado.

Os parâmetros do AG implementado para sintonizar o Controlador PID e os limites de  $K_p$ ,  $K_i$  e  $K_d$  estão mostrados na Fig. 13, onde se tem o gráfico de saída da planta com o controlador PID, no qual se observa uma redução consideravelmente no número de oscilações e na amplitude do sobressinal do sistema. Observa-se também que o sistema não tem erro em regime permanente, o que era de se esperar por causa da parte integral do controlador. Para esta situação, o AG obteve os seguintes valores dos parâmetros do Controlador PID:  $K_p=3.56$ ,  $K_i=2.45$  e  $K_d=0.62$ .

Na Fig. 14 têm-se o *fitness* médio e o *fitness* do melhor indivíduo por geração. Nela pode-se

observar que na geração 50 o AG consegue convergir, adaptando sempre o melhor indivíduo da geração anterior à próxima (elitista), melhorando assim o desempenho do controlador.

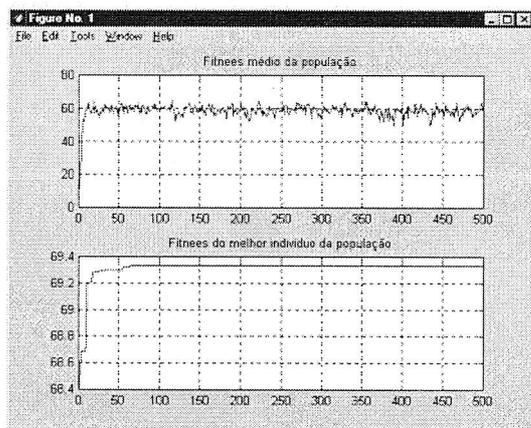


Figura 14: Gráfico da Média e do Melhor Indivíduo por Geração

## CONCLUSÕES

As interfaces criadas para a simulação de várias técnicas de implementação de AG foram bastante satisfatórias, conseguindo-se fazer adaptações e colher resultados bastante interessantes, além de se ter conseguido criar interfaces simples e interativas com os usuários.

Observou-se também que as estratégias utilizadas no Algoritmo Genético mostram-se bastante eficazes na solução de problemas de otimização de funções, entre outros. Entretanto, o tempo de processamento gasto para realizar todas as ope-

rações, tais como seleção, cruzamento e mutação, é relativamente alto e aumenta consideravelmente quando o número de indivíduos da população cresce. Portanto, a utilização do AG's na forma híbrida, como por exemplo: métodos matemáticos de otimização-AG, RNA-AG, Lógica Fuzzy-AG, podem ser uma ótima opção, para superar esta dificuldade.

## REFERÊNCIAS BIBLIOGRÁFICAS

BARRETO, J. M. (1999). **Inteligência Artificial no Limiar do Século XXI Abordagem Híbrida**. Florianópolis.

DATTA, A.; HO, M. & BHATTACHARYA, S. P. (2000). **Structure and Synthesis of PID Controllers**.

DIAS, J. S. & BARRETO, J. M. (1998). **Algoritmo Genético: Inspiração Biológica na Solução de Problemas - Uma Introdução**. Pesquisa Naval Nº 11.

FOGEL, D. B. (1995). **Evolutionary Computation Toward a New Philosophy of Machine Intelligence**. IEEE Press.

MATHWORKS. (1997). **Building GUIs with MATLAB**, ver 5.0.

RENSBURG, P. J. V., SHAW, I. S. & WHY J. D. V. (1998). **Adaptive PID – Control using a Genetic Algorithm**. IEEE Conference on Knowledge-Based Intelligent Electronic System, Australia, 133-138.

SPRINEER, VERLAG. (1992). **Genetic Algorithms + Data Structures Evolutionary Programs**. Berlim.

