

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
MESTRADO EM INFORMÁTICA**

RÔMULO FERREIRA DOURO

**ALGORITMOS GENÉTICOS –
SEMINÁRIO APRESENTADO PARA A DISCIPLINA OTIMIZAÇÃO COMBINATÓRIA E
METAHEURÍSTICAS**

VITÓRIA
2010

RÔMULO FERREIRA DOURO

**ALGORITMOS GENÉTICOS –
SEMINÁRIO APRESENTADO PARA A DISCIPLINA OTIMIZAÇÃO COMBINATÓRIA E
METAHEURÍSTICAS**

Seminário apresentado à Universidade Federal do Espírito Santo como requisito de aprovação para a disciplina Otimização Combinatória e Metaheurísticas ministrada pela prof.^a Maria Cristina Rangel.

VITÓRIA
2010

LISTA DE FIGURAS

FIGURA 1 – Função de avaliação.....	10
FIGURA 2 – Roleta.....	11
FIGURA 3 – Exemplo de reprodução.....	12
FIGURA 4 – Exemplo de crossover para o operador CX.....	13
FIGURA 5 – Exemplo da operação de mutação.....	13
FIGURA 6 – Esquema de um AG básico.....	15
FIGURA 7 – Distribuição da população de um AG (Mestre X Escravos).....	18

LISTA DE TABELAS

TABELA 1 – Exemplo de representação.....10

TABELA 2 – Cálculo da roleta.....11

SUMÁRIO

LISTA DE FIGURAS.....	2
LISTA DE TABELAS	3
SUMÁRIO.....	4
1 INTRODUÇÃO.....	6
1.1 ABORDAGENS HEURÍSTICAS	6
1.2 ALGORITMOS EVOLUCIONÁRIOS.....	7
2 ALGORITMOS GENÉTICOS.....	8
2.1 CONCEITOS BÁSICOS	8
2.1.1 Representação.....	9
2.1.1 Geração da população inicial.....	11
2.1.3 Seleção.....	11
2.1.4 Reprodução.....	12
2.1.5 Mutação.....	13
2.1.6 Parâmetros genéticos.....	14
2.2 PROCESSO ALGORITMICO.....	14
2.2.1 Busca local.....	16
2.2.2 Algoritmos genéticos paralelos.....	16
3 EXEMPLOS DE APLICAÇÕES.....	19
3.1 ROBÓTICA DE COMBATE A ACIDENTES AMBIENTAIS.....	19
3.2 DOBRAMENTO DE PREOTEÍNAS.....	19

<u>3.3 CONFIGURAÇÃO TEMPORAL PARA MERCADO FINANCEIRO.....</u>	<u>19</u>
<u>3.4 JUST-IN-TIME SCHEDULING.....</u>	<u>20</u>
<u>3.5 SEQUENCIAMENTO COM PENALIDADES.....</u>	<u>20</u>
<u>4 REFERÊNCIAS.....</u>	<u>22</u>

1 INTRODUÇÃO

Algoritmo Genético (AG) é uma técnica computacional inspirada nos princípios evolutivos da teoria de Charles Darwin (GOLDBERG, 1989) que se vale em aplicar os preceitos evolucionários das espécies em sua visão microscópica (especificamente na área da genética).

Tal técnica se trata de uma heurística que constitui parte do chamado grupo dos Algoritmos Evolucionários (CORREIA, 2005).

1.1 ABORDAGENS HEURÍSTICAS

As abordagens heurísticas clássicas se valem do acaso para desvencilharem as nuances dos problemas que abordam. Contudo, atualmente têm sido aplicado com bastante afinco o foco na resolução de problemas através do uso de metaheurísticas, as quais são adaptadas a cada caso específico seguindo uma estratégia de resolução geral (GOLDBARG; LUNA, 2000).

A estratégia de uso de uma heurística está baseada no fato de se descobrir uma solução viável para um problema e, nesse contexto, Goldbarg e Luna (2000) trazem a seguinte definição:

Uma heurística é uma técnica que busca alcançar uma boa solução utilizando um esforço computacional considerado razoável, sendo capaz de garantir a viabilidade ou otimalidade da solução encontrada ou, ainda, em muitos casos, ambas, especialmente nas ocasiões em que essa busca partir de uma solução viável próxima ao ótimo.

Uma ideia de meta-heurística são as conceituadas como Algoritmos Evolucionários que advém de um ramo de pesquisa computacional chamado de Vida Artificial (CORREIA, 2005) os quais se baseiam em observar as semelhanças entre os problemas biológicos e modelos computacionais, buscando encontrar modelos de computação para problemas da biologia e, analogamente, inspirações biológicas para soluções computacionais.

1.2 ALGORITMOS EVOLUCIONÁRIOS

No contexto das aplicações fundamentadas em ideias relacionadas com situações da biologia encontram-se os Algoritmos Evolucionários (AE) que são motivados pelas ideias referentes à evolução natural (CORREIA, 2005).

Lopes (2006) mostra que está incluso o uso de modelos inspirados na natureza dentro da área computacional da Inteligência Artificial (IA) e chama a atenção para a Computação Evolucionária (CE), que se inspira no processo de evolução natural proposto por Darwin, citando alguns métodos estabelecidos com o uso dessa técnica:

- a) Algoritmos Genéticos (AG);
- b) Programação Genética (PG);
- c) Sistemas de Classificadores (estes derivados dos AG's);
- d) Estratégias Evolucionárias (EE);
- e) Programação Evolucionária (PE).

Segundo Correia (2005) os AE's são usados em problemas de otimização onde se tem um conjunto de elementos chamado de população a qual, é avaliada de acordo com a função objetivo estabelecida pelo problema, sendo os AE's organizados em passos chamados de gerações, em conformidade com o modelo de adaptação do processo de evolução. Os AE's se assemelham em sua sequência de trabalho onde inicialmente é gerada, de forma mais simples aleatoriamente, uma população e em seguida são aplicados alguns métodos, a saber:

- a) Seleção: avalia um grupo a fim de definir os melhores pares para promover o acasalamento;
- b) Reprodução: aplica recombinação nos elementos do grupo de acasalamento criando descendentes baseados no material desses elementos;
- c) Mutação: modifica o material da população alterando aleatoriamente um indivíduo;

- d) Substituição: troca a população anterior total ou parcialmente pela nova geração promovendo melhorias;
- e) Repete os passos até que se atinja o critério de parada.

Lopes (2006) frisa o princípio da seleção natural quando se aplicam métodos com o intuito de selecionar os melhores indivíduos dentro de uma população e também indica algumas vantagens para o uso dos AE's sendo algumas delas o fato de se ter uma implementação razoavelmente fácil e de apresentar características que os tornam modularizáveis e paralelizáveis.

2 ALGORITMOS GENÉTICOS

Os algoritmos genéticos (AG), concebidos inicialmente em 1960 por Holland (1975) citado por Cintra (2007), a fim de estudar os fenômenos naturais referentes à adaptação das espécies e à seleção natural os quais, de acordo com Lopes (2006), tratam-se de procedimentos interativos implicitamente paralelos e com capacidade para encontrar uma resposta a um problema dentro de um tempo condicionado como aceitável, mesmo não sendo essa resposta a solução ótima para o determinado problema. Para tratar os AG's geralmente são usados nomes provenientes da biologia, como cromossomos e genes.

2.1 CONCEITOS BÁSICOS

Como parte integrante do grupo dos AE's, os AG's possuem características específicas de representação do indivíduo (cromossomo) além das operações de seleção, reprodução e mutação bem definidas e, portanto, sendo a base de seu processo de evolução. Essas operações fazem parte do chamado Algoritmo Genético Canônico o qual seria a forma mais simples do AG (CORREIA, 2005).

Uma solução de determinado problema onde se aplica o uso de um AG é representada por um conjunto de genes, cuja união forma um cromossomo ou indivíduo. E o conjunto de indivíduos denota a chamada população. A qualidade de um indivíduo é denotada pela sua aptidão ou, também chamado, fitness (PESSINI, 2003).

Dando sequência, Lopes (2006) denota a respeito da função de fitness e a função objetivo, sendo a função de fitness, capaz de condicionar a qualidade da solução do problema ao avaliar um indivíduo, incorporada em si a função objetivo com devidos ajustes. No decorrer da decodificação podem ocorrer situações onde indivíduos presentes em um cromossomo extrapolem o valor limite da solução o que ocasiona um questionamento quanto ao que fazer com determinado cromossomo e três saídas são propostas:

- a) Desconsiderar a solução;
- b) Reparar o cromossomo originário;
- c) Considerar a solução aplicando um valor de redução ao seu fitness.

A terceira alternativa é mais adotada, pois, dentre todas, é a que permite um tratamento com menos perdas. Esses tratamentos são baseados em métodos de seleção dentre inúmeros propostos e todos com suas vantagens e desvantagens (LOPES, 2006).

2.1.1 Representação

A representação de um indivíduo em um AG é determinada pela necessidade de seu emprego em determinado problema, podendo ser feita através de uma string, conjunto de bits ou até mesmo uma árvore. A literatura nos mostra, em geral, sua representação como uma sequência binária (CORREIA, 2005). MICHALEWICZ (1996) também mostra tal representação em exemplos do uso de um AG.

Lopes (2006) traz uma boa conceituação a respeito dos AG's e nela encontram-se detalhes quanto a utilização e desafios enfrentados na implementação desse método. Um dos desafios diz respeito à codificação, sendo esta dividida, em teoria, codificação por números binários, reais e inteiros. A escolha da codificação a ser usada é de fundamental importância para o sucesso na execução do método.

Em suma, como exemplo, um indivíduo pode ser representado através de um binário na forma mostrada na Tabela 1.

TABELA 1 – Exemplo de representação
 Fonte: CORREIA, 2005 (Com adaptação)

i	binário	x	f(x)
1	000100	4	240
2	110111	55	495
3	011001	25	975
4	101111	47	799

No exemplo dado na Tabela 1, temos a representação binária dos números que compõem a função a qual se objetiva determinar o máximo, $f(x)=1024-(x-32)^2$ conforme a Figura 1:

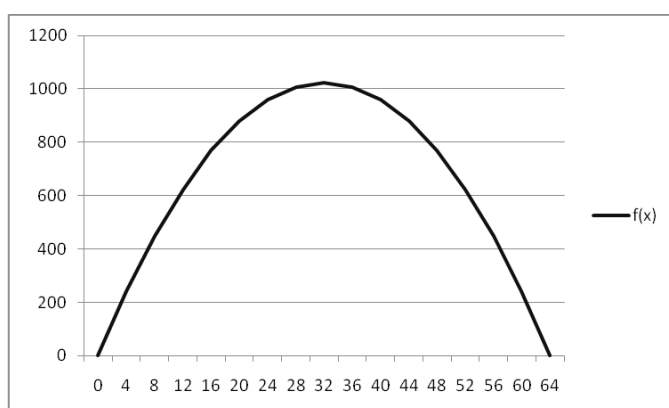


FIGURA 1 – Função de avaliação
 Fonte: CORREIA, 2005 (Com adaptação)

Não obstante, a representação de um indivíduo deve ser dada através de conceitos que infiram sobre o problema a ser tratado. Para o Problema do Caixeiro Viajante, por exemplo, devemos ter uma representação do tipo sequencia de inteiros que representem o caminho a ser seguido pelo caixeiro (MICHALEWICZ, 1996), assim:

- $s = \{3, 5, 2, 1, 6, 4\}$

Essa seria uma solução para um problema abrangendo 6 cidades e seu valor seria a soma das distâncias entre as cidades adjacentes:

- $d(3,5) + d(5,2) + d(2,1) + d(1,6) + d(6,4) + d(4,3)$

O fitness, ou função de avaliação, de um indivíduo é geralmente dado pela própria função objetivo salvo em casos em que haja a necessidade de se averiguar outros quesitos para o cálculo da aptidão (GOLDBERG, 1989).

2.1.1 Geração da população inicial

Para a geração da população inicial a literatura nos traz, como mais simples, um modelo randômico fundamentado na aleatoriedade dos genes dispostos no indivíduo. Como exemplo pode-se observar a própria Tabela 1 onde teremos uma população de 4 indivíduos onde seus valores escalares (valores de x) são {4, 55, 25, 47} sendo uma produção aleatória de valores da representação binária. Atenta-se para, em geral, evitar-se a representação repetida de um mesmo valor para combater a ambiguidade de soluções (PESSINI, 2003).

2.1.3 Seleção

A seleção é geralmente determinada pelo método da roleta. Tal método é utilizado calculando o seu índice de aptidão conforme o exemplo seguinte na Tabela 2. Quanto maior a aptidão de um indivíduo mais espaço este terá na roleta e quanto menos aptidão menor será o espaço na roleta conforme mostrado no exemplo da Figura 2.

TABELA 2 – Cálculo da roleta

Fonte: CORREIA, 2005 (Com adaptação)

i	binário	x	f(x)	aptidão
1	000100	4	240	0,10
2	110111	55	495	0,20
3	011001	25	975	0,39
4	101111	47	799	0,32
Soma f(x) =			2509	

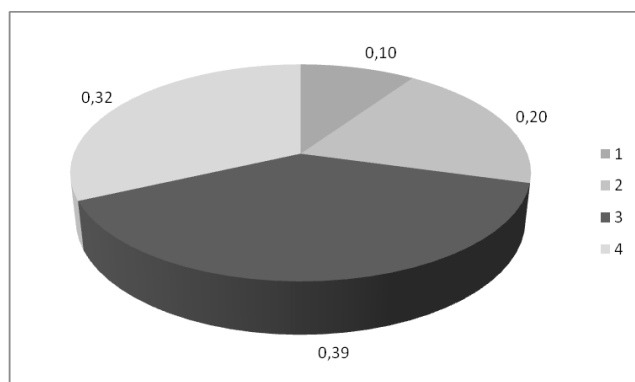


FIGURA 2 – Roleta

Fonte: CORREIA, 2005 (Com adaptação)

Ao executar a roleta é escolhido aleatoriamente um indivíduo de acordo com a probabilidade dada pela aptidão do mesmo. Com isso pode-se escolher indivíduos mais bem adaptados sem deixar de lado os menos aptos primando pela diversidade dos mesmos.

2.1.4 Reprodução

A reprodução, também chamada de crossover, é caracterizada pela troca de informações genéticas entre um indivíduo e outro selecionados a partir de sua aptidão. Como exemplo de ato reprodutório será mostrado a seguir na Figura 3.

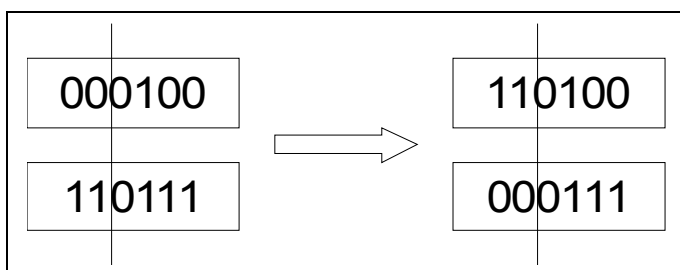


FIGURA 3 – Exemplo de reprodução

A operação de reprodução gera um ou dois filhos com base no material genético dos indivíduos pais. Através da troca de material após a definição de pontos de corte entre os genes adjacentes dos pais (PESSINI, 2003).

Em seu trabalho Silva e Oliveira (2006) fazem um estudo a respeito de vários operadores de reprodução aplicados ao Problema do Caixeiro Viajante no qual se destaca o operador HX o qual se vale de informações heurísticas para proceder com a reprodução. Tal operador procede da seguinte forma:

- a) Uma cidade inicial é escolhida aleatoriamente em um dos pais
- b) São comparados os arcos em ambos os pais deixando a cidade escolhida e selecionando o menor dos arcos
- c) Se esse menor arco formar um ciclo então deve-se escolher um outro arco aleatório
- d) Os passos 'b' e 'c' são repetidos até que se sejam inseridas todas as cidades no novo filho produzido

Silva e Oliveira (2006) destacam que tal operador consome muitos recursos de máquina pois gera um sobrecarregamento no processo, por outro lado, o operador CX é o que consome menor tempo de processamento. Esse operador funciona trabalhando sobre um subconjunto de cidades que ocupa um mesmo subconjunto de posições em ambos os pais. Esse subconjunto é transcrito do primeiro pai (p1) para o filho (f1) e o restante das posições é preenchido com as cidades constantes no segundo pai (p2). A figura 4 mostra um exemplo de como se procede esse operador.

$$p_1 = (3, 2, 1, 4, 5, 6)$$

$$p_2 = (6, 3, 4, 1, 5, 2)$$

$$f_1 = (3, 2, 4, 1, 5, 6)$$

FIGURA 4 – Exemplo de crossover para o operador CX
Fonte: SILVA e OLIVEIRA, 2006

2.1.5 Mutação

Conforme Pessini (2003) a operação de mutação consiste na alteração de um ou mais genes visando assim a geração de material genético diversificado e, por consequência, a obtenção de novos indivíduos modificados a partir de um previamente escolhido como base.

A Figura 5 dá exemplo de como pode-se proceder com a operação de mutação. Nela temos um exemplo onde um gene é escolhido e seu valor alterado.

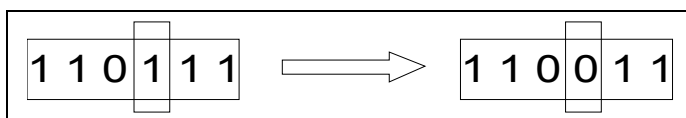


FIGURA 5 – Exemplo da operação de mutação

De acordo com Miichalewicz (1996) outros operadores de mutação podem ser utilizados principalmente se aplicarmos ao PCV, por exemplo. Em seu trabalho Silva e Oliveira (2006) citam o operador de mutação por inversão onde uma dada sequência de um indivíduo é selecionada aleatoriamente e as posições relativas entre as cidades desta sequência é invertida.

2.1.6 Parâmetros genéticos

De acordo com Pessini (2003) alguns parâmetros devem ser analisados buscando inferir sobre como os mesmos podem influenciar no comportamento do AG. Dentre os parâmetros destacam-se os seguintes:

- Tamanho da população: com um tamanho pequeno temos uma maior possibilidade de cairmos em armadilhas como máximos locais. Com o aumento do tamanho abrangemos um espaço de soluções maior e por consequência mais possibilidades de adquirir um ótimo, não obstante, com o aumento do tamanho da população enfrentamos o problema de tempo de execução e recursos computacionais;
- Taxa de cruzamento: dependendo do valor dessa taxa a população pode convergir mais lentamente (valores baixos) ou poderá ocorrer mesmo a perda de material genético de boa qualidade (valores altos) dada a alta probabilidade de troca de material para a geração de novas estruturas;
- Taxa de mutação: é um parâmetro importante dado que serve para prevenir a fuga de um espaço de busca limitado evitando a estagnação na permanência de um mínimo local contudo, com uma taxa muito elevada, a busca se torna essencialmente aleatória;
- Taxa de substituição: é basicamente a quantidade de indivíduos que deve ser substituída na população atual para que seja formada a nova população a ser utilizada no passo seguinte do processo evolutivo (próxima geração).

2.2 PROCESSO ALGORITMICO

A estrutura de um algoritmo genético, em sua forma básica (Figura 2), segue os preceitos obedecidos por operadores genéticos, considerados os mais importantes, que fazem as funções de reprodução ou crossover, mutação e inversão para que, no final das operações genéticas, se obtenha uma nova geração de indivíduos (LOPES, 2006).

Pode-se observar a questão do critério de parada (Figura 6) que pode se valer da situação de convergência a qual Dejong (1975) citado por Lopes (2006) define da seguinte forma: quando 95% da população possuir o mesmo gene então ocorre a convergência deste gene e a partir do momento em que todos os genes de uma população convergiram, diz-se que esta população convergiu. Não obstante o critério de parada é, em geral, definido observando nuances do problema a ser tratado podendo ser empregado como um número máximo de gerações no processo evolutivo da população (RIBEIRO; SOUZA; De SOUZA, 2009).

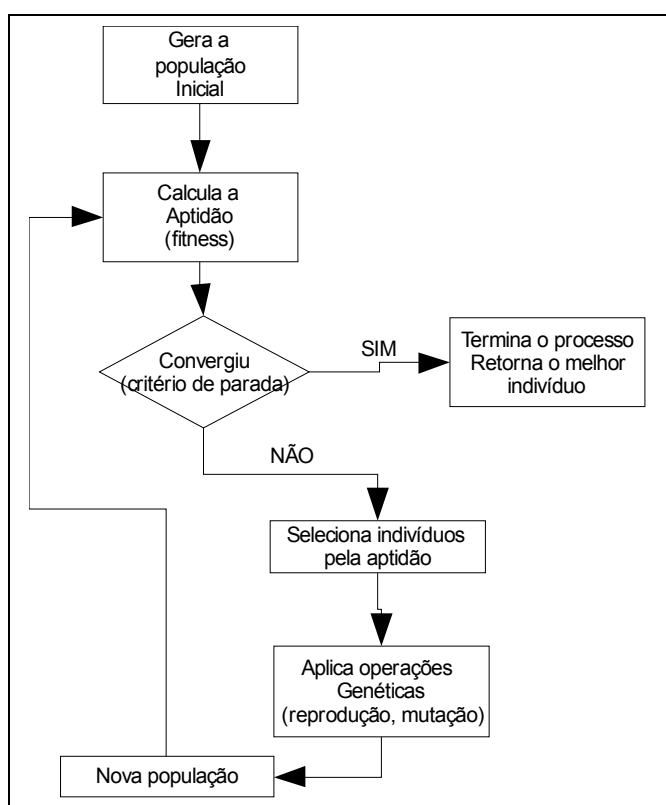


FIGURA 6 – Esquema de um AG básico
Fonte: CINTRA, 2007 (Com adaptação)

O fator convergência pode gerar dificuldades quanto à busca da solução de acordo com a diversidade genética da população. Com o passar do tempo indivíduos de maior fitness se sobressaem e geram mais descendentes que os outros, o que diminui a diversidade genética em favor de um determinado grupo. Isso se dá, em geral, pela pressão seletiva, a qual denota que seres de menor fitness desaparecerão com o passar do tempo e, a partir disso, o domínio dos indivíduos de maior aptidão dominarão a população fazendo com que o método convirja para, provavelmente, um máximo local (LOPES, 2006).

2.2.1 Busca local

Como alternativa de saída para ótimos locais, podem ser adicionadas técnicas de busca local a heurísticas como meio de aprimoramento das soluções, hibridizando o algoritmo genético através do uso conjunto como da heurística 2-opt. A técnica 2-opt pode ser adaptada e aplicada como um fator de melhoramento das soluções obtidas sendo executada a cada nova geração e aplicada a todos os indivíduos da população. A técnica 2-opt é descrita na literatura como uma estratégia de melhoria que parte de um ciclo hamiltoniano (GOLDBARG; LUNA, 2000). Tal técnica é evidente de uso em problemas como o Caixeiro Viajante, contudo, é passível de adaptação para outros problemas.

A heurística 2-opt é um procedimento derivado do conjunto de heurísticas k-opt ou de k-substituições, onde o trabalho executado é tratado com base em dois arcos do ciclo estabelecido. Rodrigues (2000) demonstra o movimento 2-opt, baseado no Problema do Caixeiro Viajante, onde uma nova solução é gerada através da remoção de dois arcos e então os caminhos gerados são reconectados invertendo-se a ordem anterior. Esse procedimento de remoção de arcos e reconfiguração do caminho é repetido até que uma solução melhor seja encontrada ou que se esgotem as possíveis combinações de arcos.

2.2.2 Algoritmos genéticos paralelos

Para tratar de problemas encontrados no mundo real, tal qual os problemas de otimização combinatória inseridos nos contextos dos trabalhos aqui citados, se faz necessária a utilização de um conjunto populacional de número bastante elevado com o intuito de explorar da melhor forma o espaço de soluções. É nesse ponto, quando se tem um grande espaço de busca, que os AG's se sobressaem em relação a outros métodos, pois se beneficia ao tratar de uma população mais elevada. Não obstante, ao se trabalhar com um espaço de busca de tamanho elevado, a forma clássica do AG se torna onerosa demais uma vez que se faz necessária a presença de uma população de número elevado e, em conjunto, também devido à complexidade computacional dos cálculos das funções de aptidão

aplicadas aos indivíduos de tal população. Fatores esses que elevam, ao final, o tempo necessário para realizar o processamento. Daí a iniciativa de se usar AG's paralelos, aplicando a filosofia de "dividir para conquistar" (LOPES, 2006).

Campos, Yoshizaki e Belfiore (2006) descrevem de forma sucinta a utilização de algoritmos genéticos aplicados em conjunto com programação paralela na resolução do problema abordado pelos mesmos e frisam a importância que deve ter a escolha da estrutura que irá codificar a solução (DNA) de um indivíduo da população além de mencionarem a geração aleatória de indivíduos para a população inicial como uma estratégia ruim mas com a possibilidade de exploração de um espaço mais amplo sendo que outra estratégia seria a utilização de uma heurística a fim de obter tal população inicial porém, essa abordagem poderia levar mais rapidamente a uma solução baseada em um ótimo local.

Lopes (2006) traz uma conceituação a respeito dos tipos de implementações em paralelo de algoritmos genéticos e foca seu trabalho principalmente no chamado Algoritmo Genético Insular (AGI) o qual se baseia em um conjunto de populações capazes de evoluir de forma independente sendo essa aplicação de possível implementação em uma rede de computadores onde cada máquina receberia uma população e trabalharia sobre ela havendo a possibilidade de se trabalhar, também, com políticas migratórias entre as populações formando uma topologia de ligação entre as populações como um grafo dirigido onde os vértices representam as populações e as arestas os seus fluxos de migração.

Segundo Maresky (1994) citado por Lopes (2006) o uso da capacidade migratória pode ocasionar problemas quanto ao fato de indivíduos com maior fitness sendo incluídos em uma população a qual poderá se tornar "conquistada" por este indivíduo fazendo com que a mesma se atenha a se fixar num ótimo local e da mesma forma existe também a possibilidade de um imigrante com baixa taxa de evolução terminar por ser descartado na nova população fazendo com que seu material genético seja perdido. No entanto, segundo Tanese (1989) citado por Lopes (2006) os AGI's têm demonstrado relevante valor quando comparados às implementações seqüenciais dos AG's o que vêm justificando a prática de sua implementação.

Em sua implementação paralela do Algoritmo Genético, Campos, Yoshizaki e Belfiore (2006) utilizam o método de ilhas, permitindo ao algoritmo explorar um número mais elevado de indivíduos observando que são implementadas rotinas para controle de migrações e a utilização de um nó mestre, chamado nó 0 (zero), o qual compara as soluções fornecidas pelas demais instancias do algoritmo e retorna a melhor solução obtida.

Da forma semelhante à descrita acima Pessini (2003) utiliza um modelo de algoritmo genético paralelo – Mestre X Escravo – que possui um nó, chamado de supervisor ou mestre, que é responsável por criar as populações e também tem o papel de intermediar as relações de migração entre as populações. Neste trabalho também é tratada a política de migração de indivíduos mantendo os indivíduos que irão migrar numa região abstrata chamada de “espaço” (Figura 7).

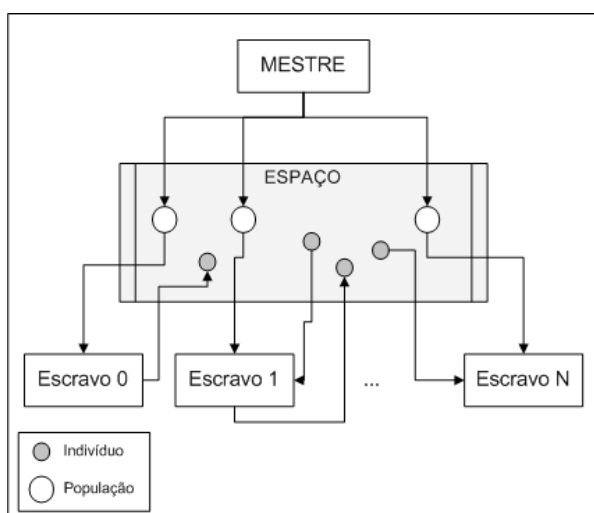


FIGURA 7 – Distribuição da população de um AG (Mestre X Escravos)
Fonte: PESSINI, 2003 (Com adaptação)

Lopes (2006) ainda traz os conceitos relacionados a outros tipos de implementações paralelas de algoritmos genéticos que são:

- a) População Global com Paralelismo (PGP): onde um processador contém a população e os demais processadores, chamados escravos, são incumbidos de procederem a avaliação de cada indivíduo. Esta implementação é usada em casos onde o cálculo de viabilidade de um indivíduo é bastante custoso.
- b) Algoritmo Genético Celular (AGC): é uma variação do AGI citado anteriormente, sendo implementada em computadores paralelos sendo que

para cada processador estará fixada a tarefa de um indivíduo e suas interações além do que, tais interações só são realizadas entre processadores vizinhos.

3 EXEMPLOS DE APLICAÇÕES

Como exemplos de aplicações serão mostrados estudos apresentados em alguns trabalhos do VII Encontro Nacional de Inteligência Artificial (ENIA 2009).

3.1 ROBÓTICA DE COMBATE A ACIDENTES AMBIENTAIS

No primeiro trabalho, Pessin e Osório (2009) utilizam a biblioteca GALib¹ para tratar o problema de atuação de um grupo robótico no apoio a acidentes ambientais como um incêndio florestal. Naquele trabalho foram estudados vários métodos da biblioteca GALib através de simulações computacionais.

3.2 DOBRAMENTO DE PROTEÍNAS

O segundo trabalho, elaborado por Benítez e Lopes (2009), aborda um tema da área de bioinformática onde um Algoritmo Genético Paralelo é desenvolvido para ser aplicado ao Problema de Dobramento de Proteínas. Tal problema se vale do fato de que as proteínas, em sua função biológica, dependem da sua estrutura tridimensional a qual é obtida pelo dobramento das mesmas sobre as próprias na sua formação no ribossomo.

3.3 CONFIGURAÇÃO TEMPORAL PARA MERCADO FINANCEIRO

Marques e Gomes (2009) usam o algoritmo genético a fim de encontrar a mais lucrativa configuração de janelas de tempo a serem empregadas na construção do indicador de análise técnica do mercado financeiro chamado de *MACD*.

¹ <http://lancet.mit.edu/ga/>

3.4 JUST-IN-TIME SCHEDULING

No trabalho de Araujo, Santos e Arroyo (2009) utilizam um algoritmo genético combinado a busca local para tratar o problema de just-in-time scheduling, o qual indica que a execução de uma determinada tarefa em um ambiente deve terminar o mais próximo possível da sua data de entrega objetivando a não permanência do bem em estoque, o que geraria custo adicional. A definição do problema naquele trabalho denominado de just-in-time job shop scheduling (JITJSSP) é tal que existem n tarefas, $J=\{J_1, J_2, \dots, J_n\}$ e um grupo de m máquinas $M=\{M_1, M_2, \dots, M_m\}$. Cada tarefa J_i possui uma sequência de operações $O_i = \{O^1_i, O^2_i, \dots, O^m_i\}$, onde O^k_i é a k -ésima operação da tarefa J_i e a função objetivo possui dois coeficientes α e β , para adiantamento e atraso, respectivamente, que são aplicados para denotarem as penalidades inerentes ao não cumprimento da janela de tempo especificada.

3.5 SEQUENCIAMENTO COM PENALIDADES

Ribeiro, Souza e Souza (2009) tratam o problema que é semelhante ao mostrado na seção 3.4 aqui chamado de PSUMAA (Problema de Sequenciamento em Uma Máquina com penalidades por Antecipação e Atraso). Este trabalho será apresentado de maneira mais ampla por se tratar de um artigo com maiores detalhes do emprego do algoritmo genético.

Neste trabalho o indivíduo é representado através de um vetor v com n tarefas cuja posição no vetor indica a ordem de produção, tomando como exemplo o vetor $v = \{5, 4, 1, 3, 2\}$ temos que a tarefa 5 é a primeira a ser processada e a 2 é a última. A avaliação do indivíduo é dada pela própria função objetivo denotada naquele trabalho. A população inicial é gerada usando a técnica GRASP e a população é formada com 200 indivíduos sendo os mesmos ordenados pela sua aptidão e os 100 piores são eliminados contando ao final com apenas os 100 melhores indivíduos inicialmente gerados.

Aqui também é usado um método de busca local chamado de Busca Randômica onde são selecionadas duas tarefas, aleatoriamente, e são trocadas de posição. Se essa troca gerar uma melhora na função de avaliação então se mantém a troca e

essa passa a ser a solução corrente, caso contrário, a troca é desfeita e procura-se outra tarefa a ser trocada aleatoriamente. Se durante esse procedimento nenhuma troca resultar em alguma melhora então são usados movimentos de realocação. Esse procedimento é executado com um número máximo pré-definido de iterações.

Neste trabalho também é empregado a técnica de elitismo para avaliar a sobrevivência dos indivíduos com o passar das gerações. Aqui sobrevivem 95% dos indivíduos mais aptos e 5% são submetidos a mutações.

O critério de parada para o algoritmo é o número máximo de gerações.

De acordo com os autores, ao avaliarem a técnica implementada, foram efetuados testes e, em comparação com os resultados dados anteriores àquele trabalho, o algoritmo implementado demonstrou melhor qualidade das soluções.

4 REFERÊNCIAS

- ARAÚJO, R. P.; SANTOS, A. G. dos; ARROYO, J. E. C.: **Algoritmo Genético e Busca Local para o problema Just-in-Time Job-Shop Scheduling**. 2009. In ENIA. 7. 2009, **Anais...**, Bento Gonçalves: SBC, 2009. Disponível em <<http://www.sbc.org.br/bibliotecadigital/download.php?paper=1416>>. Acesso em 14 set. 2010.
- BENÍTEZ, C. M. V.; LOPES, H. S.: **Algoritmo genético aplicado à predição da estrutura de proteínas utilizando o modelo 3D-HP Side Chain**. 2009. In ENIA. 7. 2009, **Anais...**, Bento Gonçalves: SBC, 2009. Disponível em <<http://www.sbc.org.br/bibliotecadigital/download.php?paper=1405>>. Acesso em 14 set. 2010.
- CINTRA, M. E. **Geração genética de regras fuzzy com pré-seleção de regras candidatas**. 2007. 101 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de São Carlos, São Carlos, 2007. On-line. Disponível em: <http://www.btdt.ufscar.br/tde_arquivos/3/TDE-2007-10-29T08:17:17Z-1540/Publico/DissMEC.pdf>. Acesso em: 03 jun 2008.
- CORREIA, L. **Vida artificial**. 2005. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO. 15., 2005, **Anais...**, São Leopoldo: SBC, 2005, On-line. Disponível em: <<http://www.sbc.org.br/bibliotecadigital/download.php?paper=414>>. Acesso em: 09 maio 2008.
- GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. Rio de Janeiro: Campus, 2000, 649 p.
- GOLDBERG, D.: **Genetic Algorithms in Search, Optimization and Machine Learning**, Addison-Wesley, 1989.
- GOLDBERG, D. et al.: **Critical deme size for serial and parallel genetic algorithms**, Technical Report IlliGAL 95002, Universidade de Illinois, 1995.
- MARQUES, F. C. R.; GOMES, R. M.: **Análise de séries temporais aplicadas ao mercado financeiro com o uso de Algoritmos Genéticos e Lógica Nebulosa**. 2009. In ENIA. 7. 2009, **Anais...**, Bento Gonçalves: SBC, 2009. Disponível em <<http://www.sbc.org.br/bibliotecadigital/download.php?paper=1408>>. Acesso em 14 set. 2010.
- LOPES, H. S. **Fundamentos de computação evolucionária e aplicações**. 2006. In: ESCOLA REGIONAL DE INFORMÁTICA DA SBC – PARANÁ. 8., 2006, **Anais...**, Bandeirantes. SBC, 2006.
- MICHALEWICZ, Z.: **Genetic Algorithms + Data Structures = Evolution Programs**, Springer-Verlag, 1996.
- PESSIN, G.; OSÓRIO, F.: **Algoritmos genéticos aplicados à formação e atuação de grupos robóticos**. 2009. In ENIA. 7. 2009, **Anais...**, Bento Gonçalves: SBC, 2009. Disponível em <<http://www.sbc.org.br/bibliotecadigital/download.php?paper=1436>>. Acesso em 14 set. 2010.
- RIBEIRO, F. F.; SOUZA, M. J. F.; SOUZA, S. R. de: **Um Algoritmo Genético Adaptativo para o problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção**. 2009. In ENIA. 7. 2009,

Anais..., Bento Gonçalves: SBC, 2009. Disponível em <<http://www.sbc.org.br/bibliotecadigital/download.php?paper=1417>>. Acesso em 14 set. 2010.

RODRIGUES, M. A. P. **Problema do caixeiro viajante – um algoritmo para resolução de problemas de grande porte baseado em busca local dirigida**. 2000. 104 f. Tese (Mestrado em Engenharia) – Universidade Federal de Santa Catarina, Florianópolis - SC, 2000. On-line. Disponível em: <<http://teses.eps.ufsc.br/defesa/pdf/4420.pdf>>. Acesso em: 29 set 2008

SILVA, A. F.; OLIVEIRA, A. C. de: **Algoritmos Genéticos: alguns experimentos com os operadores de cruzamento (“Crossover”) para o problema do caixeiro viajante assimétrico**. 2006. In ENEGEP. 26., 2006, **Anais...**, Fortaleza: ABEPRO, 2006. Disponível em <http://www.abepro.org.br/biblioteca/ENEGEP2006_TR460314_7093.pdf>. Acesso em 14 set. 2010.