

GAADT (Genetic Algorithm based on Abstracts Data Types): uma abordagem teórica e aplicativa

Manoel Agamemnon Lopes
Setor de Agroinformática
Centro de Ciências Agrárias
Universidade Federal de Alagoas
E-mail: mal@ceca.ufal.br

Roberta Vilhena Vieira Lopes
Grupo de Inteligência Artificial
Instituto de Computação
Universidade Federal de Alagoas
E-mail: rv21@hotmail.com

Resumo: Este artigo apresenta uma visão geral da Teoria dos Algoritmos Genéticos baseada em Tipos Abstrato de Dados. Aqui são levantadas algumas questões epistemológicas sobre o uso de uma lógica depreendida da Teoria da Origem das Espécies de Darwin (Darwin, 1959), (Gould, 1992) para justificar a mudança de nível de abstração como visão apropriada na busca de uma metodologia de resolução de certos tipos de problemas, que serão aqui chamados de Problemas Evolutivos (Vieira, 2003). Por este motivo, essa teoria aborda uma representação do indivíduo estratificada em três níveis epistemológicos de percepção: cromossomo, gene e base e da população como um conjunto de indivíduos construídos a partir desses insumos. Essa percepção é suficientemente rica ao ponto de permitir que a evolução seja vista como um movimento em certa direção determinada pelo ambiente. A lógica resultante dessa visão da evolução será utilizada aqui para resolver problemas.

Com essa teórica torna-se possível a caracterização de certas propriedades dos algoritmos genéticos; identificar os problemas que podem ser solucionados por estes e fazer comparações com outros modelos. A adoção de uma linguagem rigorosa para a descrição dos algoritmos genéticos e, em consequência, o estabelecimento de uma teoria formal para o tratamento dos mesmos, dará o cunho científico e um conjunto de critérios de aplicabilidade que, atualmente, esses poderosos mecanismos carecem. Por fim, é apresentada uma metodologia, bem como aplicação que é desenvolvida no decorrer do trabalho.

Palavras Chave: Algoritmo Genético, Processos Evolutivos, problemas Evolutivos, convergência.

Abstract: This article presents an overview of the Theory of Genetic Algorithms based on Abstract Data Types. Here are some epistemological questions raised about the use of a logic deduced from the Theory of the Origin of Species Darwin (Darwin, 1959), (Gould, 1992) to justify the change in level of abstraction as a vision in search of an appropriate methodology for solving certain types of problems that are here called Evolutionary problems (Vieira, 2003). For this reason, this theory deals with a representation of the individual epistemological stratified into three levels of perception: chromosome, gene and base and the population as a collection of individuals based on these inputs. This perception is sufficiently rich enough to allow evolution to be seen as a move in a certain direction determined by the environment. The logical result of this view of evolution is used here to solve problems.

With this theory it is possible to characterize certain properties of genetic algorithms, to identify problems that can be solved by them, and make comparisons with other models. The adoption of a rigorous language for the description of genetic algorithms and, consequently, the establishment of a formal theory for the treatment of the same, give the stamp of scientific and a set of applicability criteria that currently lack these powerful mechanisms. Finally, a method is presented, and the application that is developed during the work.

Keywords: Genetic Algorithm, Evolutionary Processes, Evolutionary problems, convergence.

1. Introdução

1.1 Considerações iniciais

A ideia de se utilizar a lógica evolutiva sugerida pela Teoria da Origem das Espécies de Darwin para resolver problemas complexos, além de epistemologicamente interessante, é muito promissora no sentido de se prestar para abordar uma grande quantidade de problemas que oferecem um elevado grau de dificuldades. O desenvolvimento de algoritmos genéticos baseados nesta ideia, tem se tornado, ao longo do tempo, um mecanismo computacional adequado ao tratamento de problemas dentro de ambientes complexos evolutivos, com destaque para os ambientes que oferecem uma grande biodiversidade. O nosso planeta é sem dúvida um grande fonte de problemas desta natureza: problemas climáticos e pluviométricos, problemas de controle de tráfego de transito, nas grandes cidades, problemas de aproximação de funções, problemas de distribuição de grandes empresas de abastecimento, problemas de monitoramento de pacientes em UTI's hospitalares, entre muitos outros.

Entre os problemas encontrados nas pesquisas sobre algoritmo genético predecessores ao GAADT, a aproximação de funções não-contínuas (De Jong, 1992), (Yuret, 1994) é certamente um dos mais importantes para a computabilidade. Dentro do entendimento da Tese de Turing-Church, as funções computáveis são necessariamente contínuas, entretanto a possibilidade de se aproximar funções não-contínuas por outro mecanismo de computação que não os estabelecidos pelo paradigma acima citado, abre uma nova perspectiva de uma computação fora deste cânone. Infelizmente esta questão não será aqui tratada, mas certamente numa outra oportunidade.

Diante de algumas questões importantes da construção dos algoritmos genéticos, tais como: "as limitações" do tamanho fixo dos cromossomos e do tamanho também fixo da população, bem como a artificialidade da operação de substituição para renovação das populações geradas, como em Holland e seus seguidores mais imediatos (Holland, 1975), (Goldberg, 1994), Michalewicz (1996), foi pensando em contornar estes problemas que se optou por desenvolver um novo modelo de algoritmo genético baseada em Tipos Abstratos de Dados (TAD). As questões que se pretende responder com o algoritmo genético baseado em tipos abstratos de dados são principalmente; a liberação desses algoritmos da restrição sobre a representação fixa do alfabeto binário e do tamanho fixo da população, bem como a redução das operações genéticas a somente duas: cruzamento e mutação.(Mustafa, 2011),(Thamilselvan, 2011)

A adoção da abordagem de TAD, para construção de algoritmos genéticos, como a feita em (Vieira, 2003), permitiu entre outras coisas, o desenvolvimento de uma sistemática de especificação do mesmo a partir do estudo das componentes básicas do problema para o qual, pretende-se que, ele seja solução. Isto representará um avanço na pragmática de uso dos algoritmos genéticos e na amplitude do espectro de aplicação dos mesmos. Uma questão decorrente da adoção de visão de TAD para o desenvolvimento de algoritmo genético baseado no processo de evolução das espécies é poder evitar que, num dado momento do processo, possa existir várias cópias de um mesmo cromossomo, e com isso permitir que entre os cromossomos resultantes de um cruzamento possa haver cromossomos menos adaptados do que os seus pais. Dentro da visão de que o algoritmo genético é solução de um problema, tal perspectiva é contraproducente com respeito à ideia de busca dos melhores resultados (cromossomos mais adaptados). As questões da representação binária do cromossomo apresentada em (Holland, 1975) parece levar críticas sobre a pré-definição da estrutura de dados a ser adotada, no caso vetor de tamanho fixo, que está no sentido inverso do precursor natural adotado pelos engenheiros da computação na busca de uma solução computacional para um problema, onde primeiro se estuda o problema para só depois definir-se a estrutura de dado utilizada para modelar os objetos do domínio do problema. Neste sentido a representação com TAD permite que um estudo em primeira mão do problema seja feito e em função deste e só depois, já num momento operacional, uma representação adequada seja adotada. Outro importante ganho com a visão acima referida é a portabilidade das soluções via algoritmo genético, uma vez que o mesmo "esquema de solução" pode resolver uma família de problemas, dependendo da natureza e do ambiente dos mesmos.

Este artigo apresenta o algoritmo genético baseado em tipos abstratos de dados como uma resposta para as questões citadas acima. Para isso foram levadas em conta a interpretação dos autores e a visão de um biólogo naturalista (Darwin, 1959), (Gould, 1992), (Dennett, 1995) bem como a de um geneticista (Futuyama, 1992), (Basile, 1975) sobre o processo de evolução das espécies. A escolha da linguagem da Teoria dos Conjuntos deve-se ao fato dela ser uma linguagem bastante difundida dentro da comunidade científica para a qual é dirigido o presente trabalho. A abordagem, para a representação do cromossomo, proposta é baseada em dois tipos abstratos: o gene e a base, os quais conferem ao cromossomo a flexibilidade de modelar diferentes estruturas de dados dependendo do modo como estes tipos forem especificados para o problema. Na metáfora adotada a base é o alfabeto, com o qual se escrevem as sílabas (genes) da palavra (cromossomo).

A representação estratificada do cromossomo também permitiu que se olhasse para dentro do cromossomo e possa se pregar sobre a relevância de um dado gene para o cromossomo. Percepção esta explorada pela operação de cruzamento apresentada, que trás como novidade uma definição dos pares de cromossomos habilitados a cruzarem, uma predicação dos genes que expressam o mesmo atributo e uma qualificação do gene que expressa um dado atributo com maior relevância para o problema.

O nascimento e a morte dos cromossomos obedecem ao critério de ponto de corte, que nada mais é do que uma medida baseada no valor da adaptação dos cromossomos que fazem parte da população considerada. O nascimento de um cromossomo ocorre sem nenhuma restrição, ou seja, todos os cromossomos gerados, por cruzamento ou por mutação, irão fazer parte da próxima população. Os cromossomos destinados a morrer são submetidos ao operador de mutação e registrados na população de cromossomos mortos. A aplicação da mutação sobre os cromossomos, destinados à morte, é uma forma de permitir que os genes relevantes para o problema destes cromossomos permaneçam na população através de seus descendentes. Enquanto o registro na população de indivíduos mortos é usado como uma memória biológica para evitar que cromossomos já avaliados e descartados possam reaparecer na população, acarretando uma diminuição da qualidade do resultado encontrado pelo algoritmo genético. Mas para viabilizar todas estas novidades se faz necessário que a representação da população seja uma estrutura com tamanho finito em cada momento, mas não limitada, nem fixa, e que o número de ocorrência de um cromossomo na população não afete o comportamento do algoritmo. No presente trabalho a estrutura para cromossomo, adotada, foi a de conjunto.

A proposta de uma teoria de processos evolutivos como está em (Vieira, 2003) tanto serve para demonstrar que o GAADT é convergente, bem como se constitui em um ambiente onde pode ser comparados: o GAADT com outros modelos de

algoritmo genético. Os fundamentos topológicos dessa teoria são: o conceito de população de estagnação, a ideia de monotonicidade e o teorema do ponto fixo para operadores evolutivos monotônicos. A única exigência para o uso dessa teoria, no caso do GAADT, é que o algoritmo manipule uma população de cromossomos, para os quais existe uma função de adaptação que reflita uma capacidade de adaptação “melhor ou igual” a capacidade da adaptação média.

2. Algoritmos Genéticos via Tipo de Dados Abstratos

2.1 - Aspectos iniciais

Neste trabalho, os indivíduos serão representados por seu material genético (cromossomos), o qual tem nas bases suas unidades elementares de formação. Por exemplo, no caso da espécie humana, o material genético é constituído pelos elementos das bases, sendo estas somente quatro: adenina (A), timina(T), guanina(G), citosina(C). Com as bases são formados os genes e com estes os cromossomos. Metaforicamente o material genético, no exemplo do Problema do Caixeiro Viajante, são os caminhos completos (cromossomos) constituídos pelos elementos da base, neste caso os nomes das cidades, os caminhos e as distâncias entre elas. A ideia do uso dos Tipos Abstratos de Dados – TAD, para a montagem desse arcabouço teórico que embasará a teórica aqui exposta, tem como principal objetivo o adiamento da implementação para uma etapa posterior quando do enfrentamento de um problema específico, isto é, nessa etapa inicial não haverá compromisso com as interpretações nem com a representação que vier a ser usada.

Def.2.1.1.(Base) - Uma base B para uma população P é o conjunto não vazio, de todos os materiais genéticos usados para a formação das características que compõem seus indivíduos (cromossomos).

Por exemplo, no Problema do Caixeiro Viajante – PCV, (Bhagel, 2011), dependendo da variação adotada, o conjunto $B = C \cup C_m \cup R_I$, onde C é o conjunto das cidades envolvidas, C_m é o conjunto dos caminhos que ligam duas cidades quaisquer do problema e R_I é o eventual conjunto de rótulos associados a cada caminho. Por exemplo: a distância entre as duas cidades extremas de cada caminho.

Os elementos de uma base se agrupam em cadeias para formar as características (genes) dos indivíduos. Dependendo dessas características é possível dizer, se um indivíduo é ou não do mesmo grupo taxonômico (espécie, família) de outro indivíduo. Isso decorre de que há uma “lei de formação” para indicar como as bases devem ser agrupadas para formação de uma característica. Tal “lei” deve também garantir que o cruzamento de dois indivíduos, por exemplo, que compartilhem de uma mesma característica resulte em um indivíduo com essa característica. Nesse trabalho essa “lei” será representada por um conjunto de *axiomas de formação de genes (AFG)*, o qual deverá ser definido para cada caso de acordo com a semântica atribuída ao gene. Por exemplo, se o gene para o PCV é uma cadeia formada pelos nomes das cidades de partida e chegada, significando a existência de um caminho entre essas cidades e um valor associado a esse caminho, um cromossomo é um caminho completo, isto é, um caminho que parte de uma cidade e retorna a ela mesma, passando por todas as demais. Assim, o conjunto *AFG* estabelecerá, entre outras condições, que o nome da cidade de partida deve ser diferente do nome da cidade de chegada, no caso de um gene (arco).

Def.2.1.2. (Gene) - Um gene $g = \langle b_1, b_2, \dots, b_n \rangle$ é uma cadeia (n-upla ordenada) formada pelos elementos b_i da base B e que satisfaz às regras contidas em AFG.

Por exemplo: No caso do PCV, o conjunto G de genes pode ser escrito como segue:

$$G = \{ g \mid \exists C_A, C_B \in N \square, \exists r_{AB} \in R_I \wedge g = \langle C_A, C_B, r_{AB} \rangle \}$$

Ainda no PCV para n cidades de N, um gene poderia ser um caminho encadeado entre várias cidades, representado da seguinte forma: $g_i = \langle C_i, C_{i+1}, r_i \rangle$, tais que, C_i é i-ésima cidade do PCV em pauta, C_{i+1} é (i+1)-ésima cidade e r_i é a distancia entre C_i, C_{i+1} . Ou $g_j = \langle b_j, b_{j+1} \rangle$, onde $b_j = \langle C_j, C_{j+1}, r_j \rangle$ e $b_{j+1} = \langle C_{j+1}, C_{j+2}, r_{j+1} \rangle$. Aqui deve ser observado que, enquanto g_i é um gene formado por um caminho entre duas cidades vizinhas, g_j pode ou não ser um gene formado por três cidades, mas isso deve estar descrito através dos axiomas. No caso, $AFG = \{ afg_1, afg_2, afg_3, \}$ como abaixo:

$$\begin{aligned} afg_1 &\equiv \forall g \in G (g = \langle C_A, C_B, r_{AB} \rangle \rightarrow C_A \neq C_B); \\ afg_2 &\equiv \forall g, g' \in G (g = \langle C_A, C_B, r_{AB} \rangle, g' = \langle C_A, C_B, r'_{AB} \rangle \wedge r_{AB} \neq r'_{AB} \rightarrow g \neq g'); \\ afg_3 &\equiv \forall g \in G, \exists C_x, C_y \in N, \exists r \in R_I (g = \langle C_x, C_y, r \rangle, \forall g' \in G ((g' = \langle C_x, C_y, r \rangle) \rightarrow g = g')). \end{aligned}$$

Os genes são agrupados em cadeias para formar os cromossomos c (indivíduos) da população. O conjunto de genes $\{g_1, g_2, \dots, g_n\}$ que compõe um dado cromossomo c serve para indicar as características do indivíduo representado por c , logo a ordem em que os genes encontram-se dispostos no cromossomo não altera o indivíduo identificado pelas características indicadas por desse conjunto de genes. Nesse trabalho, um cromossomo (indivíduo) será representado por um conjunto genes que obedece às regras contidas em AFC (*axiomas de formação de cromossomos*), o qual deverá ser definido para cada situação de acordo com a semântica adotada para o cromossomo. Por exemplo, se o cromossomo para o Problema do Caixeiro Viajante é uma cadeia formada pelos genes, ou seja, caminhos como definido anteriormente, então entre os axiomas de AFC poderiam figurar um para dizer que o nome da cidade de chegada do gene i deve ser igual ao nome da cidade de partida do gene $i+1$ e outro para dizer que o nome da cidade de chegada do último gene deve ser igual ao nome da cidade de partida do primeiro gene.

Def.2.1.3. (Cromossomo) - Um cromossomo $c = \{g_1, g_2, \dots, g_n\}$ de P é um conjunto de genes obedecendo às condições estabelecidas pelo AFC da população P .

Seja G o conjunto de todos os genes que são utilizados para a construção dos cromossomos c de P , o conjunto de todos os c 's obtidos de G e que satisfazem aos axiomas de **AFC** será denominado por [**AFC**].

Por exemplo: No caso do PCV, a solução aqui almejada é um caminho que passe por todas as k cidades consideradas no problema, uma única vez e que seja o menor possível. Assim, seja $C = \{c / \exists g_1, g_2, \dots, g_n \in G \wedge c = \{g_1, g_2, \dots, g_n\} \in \text{AFC}\}$, onde $\text{AFC} = \{afg_1, afg_2, afg_3, \dots\}$ como abaixo:

$$\begin{aligned} \text{Afc}_1 &\equiv \forall c = \{g_1, g_2, \dots, g_n\} \in [\text{AFC}] \wedge \forall i, j (1 \leq i, j \leq n-1) (g_i = \langle C_i, C_{i+1}, r_i \rangle \wedge g_j = \langle C_j, C_{j+1}, r_j \rangle) \rightarrow (j=i+1 \rightarrow C_{i+1} = C_j); \\ \text{Afc}_2 &\equiv \forall c = \{g_1, g_2, \dots, g_n\} \in [\text{AFC}] \wedge (g_1 = \langle C_1, C_2, r_1 \rangle \wedge g_n = \langle C_n, C_{n+1}, r_n \rangle) \rightarrow (C_{n+1} = C_1); \\ \text{Afc}_3 &\equiv \forall c = \{g_1, g_2, \dots, g_n\} \in [\text{AFC}] (n = k+1) \end{aligned}$$

2.2 - Operadores Genéticos

Os operadores genéticos são aplicados aos cromossomos da população P para, usando o material genético desses cromossomos, construir novos cromossomos que atendam melhor aos requisitos Rq de um ambiente A . Existem dois tipos de operadores genéticos, os de reprodução e os de mutação. Os operadores genéticos de reprodução caracterizam-se por combinar os genes de vários cromossomos (cromossomos-pai) para formar outros cromossomos (cromossomos-filho), enquanto que os operadores genéticos de mutação caracterizam-se por alterar os genes de um cromossomo para formar outros cromossomos (cromossomo-mutante) distintos do inicial. É claro que os genes de um cromossomo c , a serem alterados, não podem ser em número tal que altere as características do cromossomo de origem de modo que os cromossomos resultantes não devam se perder o vínculo evolutivo com a origem, ou seja, devem se manter ainda da mesma linhagem do da origem. Dependendo do problema e aconselhável que alguma restrição seja feita a quantos ou a quais genes de um cromossomo sujeito a mutação.

Os genes dos cromossomos-pai que farão parte dos cromossomos-filho são aqueles que melhor satisfizerem aos requisitos Rq de um ambiente A , denominados de genes-dominantes. Esses genes não apresentam o mesmo nível epistemológico dos genes dominantes propostos por Mendel em seu trabalho com ervilhas (Basile, 1975), já que existe uma grande diferença entre dizer que um gene g satisfaz melhor os requisitos Rq de um ambiente A do que um gene g' e dizer que o fator hereditário de um gene g é superior ao fator hereditário de um gene g' como em Mendel.

Aqui, essa ideia será vista do seguinte modo: um gene g satisfaz melhor um requisito rq do conjunto de requisitos Rq de um ambiente A do que o gene g' , com $g \square g'$, se o grau de adaptação do gene g for superior ao grau de adaptação do gene g' , e tanto o gene g quanto o gene g' satisfazendo os mesmos requisitos rq do ambiente A . Neste sentido, o grau de adaptação de um gene é dado por uma função *grau* e será considerado que não existem dois genes com o mesmo grau de adaptação satisfazendo ao conjunto de requisitos Rq de um ambiente A . Entretanto, será considerada a existência de um gene g_{\square} que satisfaz ao Rq de qualquer ambiente A e satisfaz também ao conjunto de axiomas de formação de genes **AFG** e ao conjunto de axiomas de formação de cromossomos **AFC**. Seu grau de adaptação é menor que o grau de qualquer outro gene de P . Tal gene, denotado por g_{\square} , será denominado de gene-inócuo.

Def.2.2.1. (Grau) - O grau de adaptação do gene é uma função grau do seguinte tipo:

$$\text{grau}: G \rightarrow K \text{ tal que,}$$

a cada gene $g \square G$ lhe é associado um único número de K (K é um corpo ordenado¹), chamado de *grau(g)*.

Por exemplo: Para o caso do PCV, em que são considerados: o conjunto N de n cidades e $K = \mathcal{R}$ - corpo dos números reais, $\square \square \square G = \{g = \langle c_i, c_j, r_k \rangle / \forall i (1 \leq i \leq n) \forall j (1 \leq j, k \leq n+1) \forall k (1 \leq k \leq m) c_i, c_j \in N \wedge r_k \in \mathcal{R}\}$, onde m é o número de todos os caminhos entre todos os pares de cidade do Problema em pauta. Assim, o grau poderia ser definido como sendo: $\text{grau}(g) = r_k$.

A ideia do *grau* é que ela reflita, segundo o observador e as condições do problema em pauta, uma estratificação comparada entre a adaptação dos genes de G . Um cromossomo é um conjunto de genes por definição, assim, poder-se-ia dizer que seu grau de adaptação é a soma dos graus de adaptação dos seus genes como segue:

Def.2.2.2.. (Adaptação) - O grau de adaptação de um cromossomo é uma função adapt do seguinte tipo:

$$\begin{aligned} \text{Adapt}: P \rightarrow K \\ \text{adapt}(c) = \sum_{g \in (c)} \text{grau}(g) \end{aligned}$$

Def.2.2.3. (Adaptação média) - A adaptação média de uma população, ou simplesmente adapt_m , é um função do seguinte tipo:

$$\text{Adapt}_m: \wp(P) \rightarrow K$$

¹ Um corpo ordenado é uma estrutura algébrica, com duas operações, sem divisores próprios de zero e munido de uma ordem. Ex: $\langle \mathcal{R}, \leq, +, \times, 0, 1 \rangle$, maiores detalhes ver em (Wint, 1974), (Roman, 1995)

$$adapt_m(P) = \frac{\sum_{c \in P} adapt(c)}{card(P)}$$

onde $card(X)$ é cardinalidade do conjunto X , que retorna o número de elementos de X .

Por exemplo: No caso do PCVB que tem sido considerado, o grau de adaptação do cromossomo e o grau de adaptação médio ficariam como segue: $adapt(c=\{g_1, g_2, \dots, g_n\}) = r_1 + r_2 + \dots + r_m$ e para qualquer $P \in \wp(P)$, a $adapt_m(P)$ é calculado como acima.

O operador genético de cruzamento, quando aplicado, pressupõe que uma escolha entre os genes dos cromossomos-pai foi feita levando-se em conta algum critério de escolha. Assim, torna-se necessário que antes da definição do operador de cruzamento deve ser definida uma função que realize tal escolha. O gene-dominante é identificado pela função *domi* que recebe um par de genes dos cromossomos-pai, e retorna o gene de 'melhor' grau de adaptação, se os genes fornecidos satisfazem um mesmo requisito rq do conjunto de requisito Rq de um ambiente A . Caso os genes fornecidos não satisfaçam um mesmo requisito rq do conjunto Rq de um ambiente A , então a função *domi* retornará g_{\square} , isto é, a função *domi* compara dois cromossomos absolutamente comparáveis e quantifica isso como abaixo:

Def.2.2.4. (Dominante) - O gene-dominante, ou simplesmente *domi*, é uma função do seguinte tipo:

$$\begin{aligned} & \text{Domi: } G \times G \rightarrow G \text{ tal que} \\ & \text{domi}(g, g') = \\ & \begin{array}{ll} g_{\lambda} & \text{se } \forall rq \in Rq \quad \neg(rq(g) \wedge rq(g')) \\ g & \text{se } \exists rq \in Rq \quad rq(g) \wedge rq(g') \wedge (grau(g) > grau(g')) \\ g' & \text{se } \exists rq \in Rq \quad rq(g) \wedge rq(g') \wedge (grau(g) < grau(g')) \end{array} \end{aligned}$$

onde $rq(g)$ significa que o gene g satisfaz a regra rq de Rq .

Por exemplo: ainda no caso do PCV, se for considerado uma rq como sendo um dos caminhos entre duas cidades C_A e C_B quaisquer de N , o gene dominante entre $g_i = \langle C_A, C_B, r_i \rangle \wedge g_j = \langle C_A, C_B, r_j \rangle$ que satisfazem rq é g_i se $r_i \leq r_j$ ou é g_{\square} . Aqui cabe uma observação sobre g_{\square} : A ideia do gene inócuo é da uma saída para a função *domi* quando dois genes, não comparáveis forem requeridos para a escolha do dominante entre eles;

Corolário 2.2.1 - $\square g, g'$ $domi(g, g') = domi(g', g)$

Prova: Imediata da definição. \square

A produção de novos cromossomos nos sistemas adaptativos complexos serve para direcionar a busca de cromossomos mais adaptados. No algoritmo genético esse papel é desempenhado pelos operadores genéticos de reprodução. O único operador genético de reprodução, que faz parte do processo de evolução dos cromossomos do ambiente de trabalho, para algoritmos genéticos, é o cruzamento. O operador genético de cruzamento gera cromossomos-filho a partir dos genes operados por uma função que proceda tal manejo genético. Metaforicamente essa função poderia ser chamada de fecundação. A função de fecundação, por sua vez, recebe o conjunto de genes de cada um dos cromossomos-pai, e retorna um conjunto de genes formado pelos genes dominantes dos cromossomos de origem.

Def. 2.2.5. (Fecundação) - A fecundação é uma função *fec* do seguinte tipo:

$$\begin{aligned} & \text{fec: } P \square P \square \square(G) \\ & \text{fec}(c=\{g_1, g_2, \dots, g_n\}, c'=\{g'_1, g'_2, \dots, g'_m\}) = \{h \mid m = \max\{n, m'\} \wedge \square 1 \leq j \leq m (h = \langle h_1, h_2, h_3, \dots, h_{m-1}, h_m \rangle} \\ & \wedge h_j = \text{domi}(g_j, g'_j)) \} \end{aligned}$$

Obs.: Se $m > n$, completa-se com g , as posições $n+1, n+2$, etc., até $n+k = m$ ou vice versa.

Corolário 2.2.2 - $\square c, c'$ $fec(c, c') = fec(c', c)$.

Prova: Óbvio. \square

A operação de cruzamento recebe dois cromossomos-pai e retorna uma população composta por todos os cromossomos formados a partir dos genes-dominante dos cromossomos fornecidos e que satisfaçam ao conjunto de axiomas de formação de cromossomos AFC. Para os objetivos deste artigo, o cruzamento é um operador genealógico que objetiva misturar o material genético de dois (ou mais) indivíduos diferentes, para a construção de um novo indivíduo diferente dos que lhe deram origem. No caso do exemplo que vem sendo considerado o cruzamento se faz entre dois indivíduos. Para isso as funções *domi* e *fec* foram preparadas com esse objetivo, eventualmente poderiam ser diferentes e envolver mais de duas entradas.

Def. 2.2.6. (Cruzamento) - O cruzamento é uma função *cruz* do seguinte tipo:

$$\begin{aligned} & \text{cruz: } P \square P \square \square(P) \\ & \text{cruz}(a=\{a_1, a_2, \dots, a_n\}, b=\{b_1, b_2, \dots, b_m\}) = \{c=\{c_1, c_2, \dots, c_k\} / \square i, j, k ((1 \leq i \leq n), (1 \leq j \leq m), (1 \leq k \leq k), k = \text{Max}\{n, m\} (c_i = \text{fec}(a_i, b_j))) \} \end{aligned}$$

Corolário 2.2.3 - $\square a, b (cruz(a, b) = cruz(b, a))$.

Prova: Decorre imediatamente dos Corolários 1e 2. \square

As alterações de cromossomos da população atual que não satisfazem aos requisitos **Rq** do ambiente **A**, servem como iniciação da exploração de pequenas variações desses cromossomos, que podem torná-los eventualmente aptos a satisfazerem **Rq**. No algoritmo genético esse papel é desempenhado pelos operadores genéticos de mutação. Os operadores genéticos de mutação, que serão aqui definidos, são inserção, supressão e troca. Os cromossomos-mutantes resultantes da ação de cada um desses operadores auxiliares, apresentarão parte dos genes contidos nos cromossomos que lhes deram origem. A operação de inserção **ins**, adiciona um novo gene ao cromossomo origem; a operação de supressão **del** remove um gene do cromossomo origem e a operação de troca **troc** remove um gene do cromossomo origem e lhe adiciona outro.

Def. 2.2.7. (Inserção) - A inserção é uma função **ins** do seguinte tipo:

Ins: P x G → P tal que

$$\begin{aligned} Ins(c, g) &= c \cup \{g\} && \text{se } g \notin c \\ Ins(c, g) &= c && \text{senão} \end{aligned}$$

Deve ser observado que **ins** obedece as seguintes propriedades:

1. $\square c \in P, ins(c, g_\lambda) = c$;
2. $\square c \in P, \square g \in G (g \notin c \wedge g \neq g_\lambda \rightarrow (ins(c, g) \neq c \wedge ins(c, g) \in AFC))$;

Prova: Decorre imediatamente das definições correlatas. \square

Def. 2.2.8. (Supressão) - A supressão é uma função **del** do seguinte tipo:

del: P x G x P tal que

$$\begin{aligned} del(c, g) &= c - \{g\} && \text{se } g \in c \\ del(c, g) &= c && \text{senão} \end{aligned}$$

Certamente **del** satisfaz às seguintes propriedades:

1. $\square c \in P, del(c, g_\lambda) = c$
2. $\square c \in P, \square g \in G (g \notin c \wedge g \neq g_\lambda \rightarrow del(c, g) \neq c \wedge del(c, g) \in AFC)$;

Prova: Decorre imediatamente das definições correlatas. \square

Def. 2.2.9. (Troca) - A troca é uma função **troc** do seguinte tipo:

$$\begin{aligned} troc: P \times G \times G &\rightarrow P \text{ tal que} \\ troc(c, g, g') &= ((c - \{g\}) \cup \{g'\}) && \text{se } g \in c \text{ ou} \\ troc(c, g, g') &= c && \text{senão} \end{aligned}$$

Para a **troc**, as seguintes propriedades são satisfeitas:

1. $\square c \in P, \square g \in G (g \in c \rightarrow troc(c, g, g_\lambda) \in c) \rightarrow troc(c, g, g_\lambda) \in AFC$;
2. $\square c \in P, \square g \in G (g \in c \rightarrow troc(c, g_\lambda, g) \in c) \rightarrow troc(c, g_\lambda, g) \in AFC$;
3. $\square c \in P, \square g_i, g_j \in G (g_i \in c \wedge g_j \notin c) \rightarrow troc(c, g_i, g_j) \in c \rightarrow troc(c, g_i, g_j) \in AFC$;

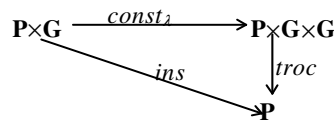
Prova: Decorre imediatamente das definições correlatas. \square

Proposição 2.2.1 - As funções de inserção **ins**, supressão **del** podem ser entendidas como casos particulares da função troca da seguinte forma:

$$ins(c, g) = troc(c, g_\square, g) \quad \text{e} \quad del(c, g) = troc(c, g, g_\square)$$

Prova: Obvia, imediata das definições 2.10; 2.11; 2.12.

O diagrama abaixo ilustra a proposição de que $ins(c, g) = troc(c, g_\square, g)$.



Para a função **del** o diagrama é análogo. \square

Esta proposição demonstra que a ação da mutação, por inserção ou por supressão, pode ser vista como casos particulares da mutação através da operação de troca. Desta forma pode-se definir a mutação como sendo o seguinte predicado:

Def. 2.2.10 (Mutação) - A mutação é um predicado **mut** $\subseteq P$ e tal que

$$mut(c) = \{c' / \square g, g' \in G \ c' = troc(c, g, g') \wedge c' \in AFC\}$$

Deve ser observado aqui, que a mutação representa um operador genético capaz de gerar indivíduos novos em A, enquanto o cruzamento não gera tal inovação; gera apenas os indivíduos que podem ser a partir do material genético existente até aquele momento evolutivo. No caso do PCV, uma motivação para a ideia de um operador de mutação pode ser encontrada nos seguintes fatos situacionais:

- a) A construção de uma ponte pode dar ensejo a um novo caminho entre duas cidade;
- b) A obstrução ou a destruição de uma ponte pode suprimir estrada;
- c) A duplicação de uma estrada pode eventualmente mudar o rótulo correspondente àquele gene;
- d) A eliminação ou a inclusão de uma cidade num certo roteiro de distribuição pode provocar alguma alterações no ambiente A do Problema, tanto na questão conjunto N, como no conjunto Rq.

2.3 - Evolução

Um algoritmo genético opera sobre populações de indivíduos que evoluem de acordo com as características de um ambiente A e fornece como saída, indivíduos de P ou de uma extensão P' de P inserida no mesmo ambiente A. Um ambiente A é uma 6-upla da seguinte forma $\langle P, \wp(P), Tx(P), \Sigma, Rq, AFG, AFC, P_o \rangle$, onde P é uma população com no mínimo um indivíduo, $\wp(P)$ é o conjunto potência de P, Tx é a classificação taxonômica (espécie e família) dos indivíduos da população P, Σ é um conjunto de operadores genealógicos que atuam sobre a população P (por exemplo: cruzamento, troca), Rq é um conjunto dos requisitos (características expressas através de fórmulas numa linguagem de primeira ordem) do ambiente A que influenciam a genealogia da população P, AFG é um conjunto de axiomas de formação dos genes dos indivíduos da população P, AFC é um conjunto de axiomas de formação dos cromossomos da população P e P_o é uma subpopulação de P, distinguida em $\wp(P)$, chamada de população inicial.

O processo de evolução darwinista, segundo o qual todas as espécies desenvolveram-se a partir de outras espécies, pela transmissão hereditária de pequenas variações, em sucessivas gerações, resultando na sobrevivência das espécies com melhor capacidade de adaptação ao ambiente (Darwin, 1959), é induzido pelas alterações ambientais produzidas pela natureza. Esse papel, desempenhado pela natureza, na visão da evolução darwinista, aqui será representado pelo algoritmo genético, que é quem submete os indivíduos de uma população às ações dos requisitos Rq do ambiente A, resultando assim, na geração de novos indivíduos a partir daqueles.

No caso do PCV, as interpretações para a 6-upla $A = \langle P, \wp(P), Tx(P), \Sigma, Rq, AFG, AFC, P_o \rangle$ poderia ser a seguinte:

P – o conjunto de todos os caminhos completos, com as respectivas distâncias, que ligam cada cidade a si mesma, passando por todas as demais, num determinado problema;

$\wp(P)$ – o conjunto de todos os subconjuntos de caminhos completos que pode ser formados com os subconjuntos de P;

Tx(P) – o conjunto de todos os caminhos completos sem pontes, unido ao conjunto dos caminhos asfaltados, etc.

Σ - conjunto de operadores genéticos, no caso do PCV aqui considerado, tem-se: $\square\square\square$ grau, adapt, adapt_m, ins, Del, troca, cruz, mut $\square\square$ de acordo com as definições acima;

Rq = {restrições do tipo: *tem ou não estrada parcial ou totalmente asfaltada; percurso de baixo consumo de combustível; obstrução parcial da estrada no trecho tal, etc., etc.*}

AFG = {afg₁, afg₂, afg₃ da definição 2.1.2}

AFC = {afc₁, afc₂, afc₃ da definição 2.1.3}

P_o = {(c₁, c₂, r₁) (c₂, c₃, r₂) (c₃, c₄, r₃), ..., (c_n, c₁, r_n)}, onde 1 ≤ i ≤ n = card(N) e N_i – conjunto de cidades e índice das cidades.

Observação: Se eventualmente duas cidades C_x, C_y não tem um caminho entre elas, o rótulo r_{xy} = 0, ou seja, o gene (C_x, C_y, 0) = g_λ. Desse modo, o P_o acima exibido pode ser entendido como um caminho natural a ser distinguido.

2.4 - O GAADT (Genetic Algorithm based on Abstracts Data Types)

Um algoritmo genético é um processo que recebe uma população P e depois de submetê-la a uma simulação de um processo evolutivo devolve uma população P'. Os indivíduos da população P' são: indivíduos da população P que satisfazem os requisitos Rq do ambiente A com grau de adaptação satisfatório, juntos com os novos indivíduos, se for o caso, resultantes da ação genealógica das operações de cruzamento e mutação, sobre os indivíduos da população P e que satisfazem os requisitos Rq do ambiente A e com grau de adaptação também satisfatórios. Diz-se então que a população P' evoluiu da população P através do algoritmo. Os indivíduos resultantes da ação dos operadores genéticos e que não satisfazem aos requisitos Rq do ambiente A ou não tem grau de adaptação satisfatório não participarão da construção da população P', podendo ser assim entendidos como fazendo parte da população dos indivíduos “mortos”, que não figurarão entre os elementos de P'. Os indivíduos dessa população não mais farão parte das populações seguintes trabalhadas pelo algoritmo genético. Não obstante, tais indivíduos serão ainda representados, com o objetivo manter seu registro para evitar que eles apareçam novamente na população P ao longo do processo. Essa restrição atende ao entendimento do processo de evolução darwinista, que não contempla a possibilidade de uma espécie extinta, voltar a aparecer num outro momento futuro.

Antes de ser apresentada uma definição para o modelo de algoritmo genético GAADT, deve-se observar a necessidade de se estabelecer um critério de seleção sobre a população P que oriente o corte de indivíduos que não devem figurar numa população seguinte a ela. Por exemplo, o grau de adaptação média de P. Na definição do GAADT, esse ponto de

corde será representado por um valor $t > 0$ denominado de t-estagnação e que serve para estabelecer o quanto a adaptação média esta estagnada, isto é, não se altera significativamente em função das novas interações. Isso será refetido na definição abaixo:

Def.2.3.1 (Algoritmo Genético) - Seja A um ambiente. O algoritmo Genético GAADT e um operador definido como abaixo:

GAADT : $A \rightarrow A$ tal que

$$GAADT(P_n) = P_{n+1} \equiv P_{otm} \text{ se } P_{otm} = \{c/ c \in P_{n+1} \wedge (|adapt_m(P_{n+1}) - adapt_m(P_n)|) \leq t\} \neq \emptyset \quad (1)$$

$$GAADT(P_n) = P_{n+1} \quad \text{senão} \quad (2)$$

Onde $P_{n+1} = cruz(a,b) \cup (mut(c) \cup Rq(P_n))$, P_0 é a população inicial considerada, $t > 0$ é um valor imposto pela natureza do problema em pauta.

No processo acima, o ramo (1) é obviamente uma condição de saída e de parada do algoritmo e o ramo (2) é o viés recursivo do processo.

Considerando o exemplo que vem sendo construído desde o inicio deste trabalho, a instanciação do GAADT para o PCV aqui imaginado é a seguinte: Da definição 2.3.1 somente P_{otm} é que precisa ser esclarecido. Com efeito considere-se a seguinte definição:

$$P_{otm} = \{C_i = (c_i, c_{i+1}, r_i) \mid (c_{i+1}, c_{i+2}, r_{i+1}) \mid (c_{i+2}, c_{i+3}, r_{i+2}), \dots, (c_{i+n}, c_{i+n}, r_{i+n}) / C_i \in P_{n+1} \mid \forall k (1 \leq k \leq n) (\sum_{i=1}^n adapt(c_i) \leq \sum_{i=K}^n adapt(c_K)) \}$$

3. Convergência de processos evolutivos

3.1 Considerações iniciais

O estudo da convergência de algoritmo genético oferece algumas dificuldades técnicas relevantes. Primeiro porque tal estudo não pode ser feito com base nas teorias usuais de convergência, segundo porque as condições topológicas são muito especiais. O fato do ambiente A ser um ambiente evolutivo significa que, com exceção de P_0 (População Inicial) cada população P_i possui indivíduos novos e perde alguns com respeito à população anterior. Isso acarreta que a taxonomia populacional pode ser vista como um encaixamento parcial de uma população P_i na população P_{i+1} , ou seja, P_i estaria contido parcialmente em P_{i+1} , representado por $P_i \sqsubset P_{i+1}$. Ver figura Fig.3.1 abaixo.

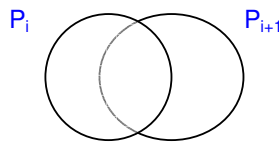


Figura 3.1: Encaixe das populações

Assim, a ideia de convergência deveria contemplar a seguinte situação: sendo P_{sol} uma subpopulação de P_i , para algum i , formada por indivíduos que atendam a uma condição de adaptação $adapt$ desejada, ter-se-ia que para um $i \leq n_0$, com n_0 suficientemente grande, a população P_i , agora, deveria ser quase P_{sol} , no sentido de que os indivíduos que diferenciariam P_i de P_{sol} teriam de certo modo uma presença desprezível. Desse modo, poderia se dizer que a população P_i convergiria para P_{sol} . (Vieira&Lopes, 1999a), (Vieira&Lopes, 2000). A idéia de convergência introduzida acima permite a observação dos seguintes casos: Primeiro, a população desejada P_{sol} estaria contida em P_0 – assim a seqüência P_0, P_1, \dots, P_n , teria uma convergência por redução (diminuição) da população P_0 até um P_n quase P_{sol} ; Segundo, P_{sol} estaria contida parcialmente em P_0 - desse modo, a seqüência P_0, P_1, \dots, P_n evoluiria (crescendo) para um P_n que contivesse P_{sol} e depois seria reduzida como no primeiro caso. Finalmente, no terceiro caso, em que a interseção entre P_0 e P_n seria vazia, então a seqüência P_0, P_1, \dots, P_n , evoluiria até produzir um P_n que contivesse P_{sol} e depois se comportaria como no primeiro caso. Esses encaixes parciais serão aqui chamados de encaixes taxonômicos.

Na Figura 3.1, pode-se observar que os encaixes taxonômicos das populações se dão parcialmente, embora tendam a um encaixe “quase completo” no limite deste processo. Ver Fig. 3.2 abaixo.

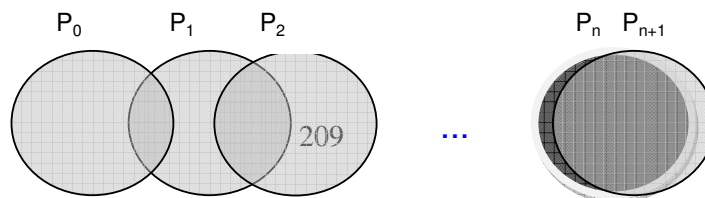


Figura 3.2: Encaixe “quase completo” das populações

Este comportamento se deve à constante alteração causada pelo surgimento (nascimento) de novos cromossomos gerados pela ação dos operadores genéticos de cruzamento e mutação, ou ao contrário, pela supressão (morte) dos cromossomos considerados inadaptados ao ambiente durante o processo de evolução da população. Logo, necessita-se de um novo esboço teórico para mostrar a convergência dentro de ambientes evolutivos. É com este intuito que será desenvolvida a seguinte teoria.

3.2 – Elementos da Teoria dos Processos Evolutivos

Dado um ambiente evolutivo $A = \langle P, \wp(P), AFG, AFC, Rq, \Sigma, P_0 \rangle$, onde P , $\wp(P)$ representam, respectivamente, a População P do ambiente e o conjunto de todos os subconjuntos de P , AFG - conjuntos de regras de formação dos genes, AFC - conjuntos de regras de formação dos cromossomos, Rq – conjuntos de regras do meio ambiente que atuam sobre os indivíduos de P , Σ - conjunto das operações genéticas que atuam sobre P e $\wp(P)$, e finalmente P_0 representa um subconjunto distinguido (inicial) de P , a partir do qual é iniciado o processo de evolução a ser observado; e dado também duas populações $P_1, P_2 \in \wp(P)$, devemos tentar destacar o fato dessas (sub) populações terem o mesmo valor de adaptação média. Isso porque do ponto de vista da utilização da Metáfora da Teoria das Origens das Espécies de Darwin, (Darwin, 1959), para resolver problemas, as populações com essa propriedade têm a mesma relevância já que a adaptação média é o ponto de referência natural para a seleção dos “melhores” candidatos segundo o perfil dos resultados desejados. Essa pretensão está refletida na definição abaixo.

Def. 3.2.1.(Equivalência Evolutiva) – A equivalência evolutiva é uma relação \equiv_E , do seguinte tipo: $\equiv_E: \wp(P) \times \wp(P)$, tal que $P_1 \equiv_E P_2$, se e somente se, $adapt_m(P_1) = adapt_m(P_2)$.

Com o mesmo intuito, com respeito ao uso da adaptação média como critério de adaptação, se uma subpopulação P_1 tem sua adaptação média inferior à de P_2 , podemos depreender que a subpopulação P_2 possui provavelmente indivíduos (resultados) mais próximos dos desejados que a população P_1 . Isso independentemente de P_1, P_2 possuírem ou não indivíduos em comum, ou seja, do ponto de vista da solução do problema, só a adaptação interessa como critério de seleção de resultados. Desse modo para se estabelecer uma hierarquia de qualidade dos resultados obtidos devemos imaginar a definição que segue:

Def. 3.2.2.(Ordem Evolutiva) – A ordem evolutiva é uma relação \subseteq_E do seguinte tipo:

$\subseteq_E: \wp(P) \times \wp(P)$, tal que $P_1 \subseteq_E P_2$, se e somente se, $adapt_m(P_1) \leq adapt_m(P_2)$

Deve-se observar que “ \subseteq_E ” é a relação de ordem sobre o corpo K , referido na definição do grau de adaptação de um gene (Vieira, 2003)), condição esta que será herdada pela relação de equivalência evolutiva, como será mostrado abaixo.

Proposição 3.2.1 – $S(P) = \langle \wp(P), \subseteq_E \rangle$ é um sistema ordenado, isto é, a relação \subseteq_E é uma relação de ordem parcial sobre $\wp(P)$.

Prova: Com efeito, são verificadas as seguintes propriedades:

- i. $\forall P_1 \in \wp(P) (P_1 \subseteq_E P_1)$ - A prova é direto da definição. \square
- ii. $\forall P_1, P_2 \in \wp(P) (P_1 \subseteq_E P_2 \wedge P_2 \subseteq_E P_1 \rightarrow P_1 \equiv_E P_2)$. Com efeito, se $P_1 \subseteq_E P_2$, então pela definição de ordem evolutiva tem-se que $adapt_m(P_1) \leq adapt_m(P_2)$. Por outro lado, de forma similar tem-se que $adapt_m(P_2) \leq adapt_m(P_1)$. Como “ \leq ” é uma relação de ordem sobre o corpo K , então se obtém que $adapt_m(P_1) = adapt_m(P_2)$. Pela definição de equivalência evolutiva, temos que $P_1 \equiv_E P_2$. \square
- iii. $\forall P_1, P_2, P_3 \in \wp(P) P_1 \subseteq_E P_2 \wedge P_2 \subseteq_E P_3 \rightarrow P_1 \subseteq_E P_3$ – É Imediata da definição de ordem evolutiva e das hipóteses que $adapt_m(P_1) \leq adapt_m(P_2) \leq adapt_m(P_3)$. E assim decorre imediatamente a transitividade, isto é, $P_1 \subseteq_E P_3$. \square

Um algoritmo genético assim como qualquer outro algoritmo da computação evolutiva atua sobre populações de cromossomos que vão se entrelaçando através das relações de parentesco, as quais são geradas por procedimentos computacionais que simulam convenientemente a Teoria das Origens das Espécies de Darwin, (Vieira&Iopes, 2000), (Goldberg, 1997), (Holland, 1975). A conveniência antes referida é no sentido de que os algoritmos genéticos têm como objetivo a solução de problema, (Lopes, 1981) e não necessariamente reproduzir a evolução das espécies como na natureza, com em (Dannet, 1995). A ideia é que os algoritmos genéticos façam parte do que será chamado, de agora em diante, de família dos processos evolutivos.

Def. 3.2.3.(Processo Evolutivo) – Um processo evolutivo é uma função PE do seguinte tipo:

$$\text{PE: } \wp(\mathbf{P}) \rightarrow \wp(\mathbf{P}), \text{ tal que } \forall P_1 \in \wp(\mathbf{P}) (P_1 \subseteq_E \text{PE}(P_1)).$$

Deve ser observado que, o que caracteriza um processo evolutivo é fundamentalmente a estratégia de melhoramento da adaptação média dos indivíduos. Isso, dentro da concepção de que os indivíduos são metaforicamente candidatas a resultados de problemas e os suficientemente adaptados, são os resultados desejados. O que se pretende como qualidade do processo evolutivo PE é que exista uma população $P' \in \wp(\text{População})$, denominada de *população de estagnação* (Ponto Fixo), tal que a variação da adaptação média da população P' , obtida em decorrência de aplicações sucessivas do processo PE a partir da população inicial P_0 é tão pequena, que o valor da adaptação média desta população praticamente não se altera a partir de determinadas interações. Isso é compatível com a ideia intuída no começo desta seção.

Def. 3.2.4.(População Estagnada) – Uma população $P' \in \wp(\mathbf{P})$ é dita estagnada (ponto fixo) com respeito a um processo evolutivo PE, se e somente se,

$$\forall P' \in \wp(\mathbf{P}) (P' \subseteq_E \text{PE}(P') \rightarrow \text{adapt}_m(P') = \text{adapt}_m(\text{PE}(P'))).$$

Proposição 3.2.2 – Se P' é uma população estagnada e PE é um processo evolutivo sobre $\wp(\mathbf{P})$, então $\text{PE}(P') \equiv_E P'$.

Prova: Imediata das Def. 3.2.1, 3.2.3 e 2.4 □.

Esta proposição expressa que a população estagnada é invariante com respeito à aplicação do processo evolutivo, isto é, nos termos da Proposição 3.2.2, P' é dito um ponto de estagnação (fixo) do processo evolutivo PE. Essa invariância significa que os indivíduos da população estagnada alcançaram a melhor adaptação, do ponto de vista de se ter o algoritmo genético como solução de seu problema associado e dessa forma, os indivíduos mais adaptados são os candidatos aos resultados que se deseja.

Def. 3.2.5.(Monotonicidade) – Um processo evolutivo é dito monotônico, se e somente se,

$$\forall P_1, P_2 \in \wp(\text{População}) (P_1 \subseteq_E P_2 \rightarrow \text{PE}(P_1) \subseteq_E \text{PE}(P_2))$$

Def. 3.2.6.(Relações Evolutivas) – As relações interseção evolutiva (\cap_E), união evolutiva (\cup_E) e diferença evolutiva ($-_E$) sobre $\wp(\mathbf{P})$, são definidas do seguinte modo:

- i. $P_1 \cap_E P_2 \equiv \{x / x \in P_1 \cap P_2 \wedge \text{adapt}(x) \leq \min[\text{adapt}_m(P_1), \text{adapt}_m(P_2)]\}$
- ii. $P_1 \cup_E P_2 \equiv \{x / x \in P_1 \cup P_2 \wedge \text{adapt}(x) \geq \max[\text{adapt}_m(P_1), \text{adapt}_m(P_2)]\}$
- iii. $P_1 -_E P_2 \equiv P_1 - \{x / x \in P_2 \wedge \text{adapt}(x) \leq \text{adapt}_m(P_1)\}$

Proposição 3.2.2. As quais relações da Def. 3.2.6 obedecem às seguintes propriedades:

Propriedade 3.2.1 - $P_1 \cap_E P_2 \subseteq_E P_1$; Prova: $\text{adapt}_m(P_1 \cap_E P_2) \leq \text{adapt}_m(P_1)$

Propriedade 3.2.2 - $P_1 \cap_E P_2 \subseteq_E P_2$; Prova: $\text{adapt}_m(P_1 \cap_E P_2) \leq \text{adapt}_m(P_2)$

Propriedade 3.2.3 - $P_1 \subseteq_E P_1 \cup_E P_2$; Prova: $\text{adapt}_m(P_1) \leq \text{adapt}_m(P_1 \cup_E P_2)$

Propriedade 3.2.4 - $P_1 \subseteq_E P_1 \cup_E P_2$; Prova: $\text{adapt}_m(P_1) \leq \text{adapt}_m(P_1 \cup_E P_2)$

Propriedade 3.2.5 - $P_1 \subseteq_E P_1 -_E P_2$; Prova: $\text{adapt}_m(P_1) \leq \text{adapt}_m(P_1 -_E P_2)$.

Lema 3.2.1 – Se $P_1 \subseteq_E P_2 \rightarrow P_1 \cap_E P_2 \equiv_E P_1$

Prova: Deve-se observar que, se $P_1 \subseteq_E P_2$, então $\min[\text{adapt}_m(P_1), \text{adapt}_m(P_2)] = \text{adapt}_m(P_1)$, logo $P_1 \subseteq_E P_1 \cap_E P_2$. Pelas propriedades (i) e (ii) da relação de ordem tem-se finalmente $P_1 \cap_E P_2 \equiv_E P_1$ □

Lema 3.2.2 – Se $P_1 \subseteq_E P_2 \rightarrow P_1 \cup_E P_2 \equiv_E P_2$

Prova: Dual da prova do Lema 1

Proposição 3.2.3 – Um processo evolutivo PE é dito “monotônico”, se e somente se, $\forall P_1, P_2 \in \wp(\text{População})$ $\text{PE}(P_1 \cap_E P_2) \subseteq_E \text{PE}(P_1) \cap_E \text{PE}(P_2)$

Prova: Considere que $\forall P_1, P_2 \in \wp(\mathbf{P})$, $P_1 \cap_E P_2 \subseteq_E P_1$ e $P_1 \cap_E P_2 \subseteq_E P_2$, pelas propriedades i e ii da Def.2.6. Como por hipótese, PE é um processo evolutivo monotônico, então $\text{PE}(P_1 \cap_E P_2) \subseteq_E \text{PE}(P_1)$ e $\text{PE}(P_1 \cap_E P_2) \subseteq_E \text{PE}(P_2)$. Assim, tem-se que $\text{PE}(P_1 \cap_E P_2) \subseteq_E \text{PE}(P_1) \cap_E \text{PE}(P_2)$. Inversamente: vamos então considerar como hipótese que $\forall P_1, P_2 \in \wp(\mathbf{P})$, $\text{PE}(P_1 \cap_E P_2) \subseteq_E \text{PE}(P_1) \cap_E \text{PE}(P_2)$. Assim, tem-se pela propriedade ii da Def.2.2 que $(P_1 \cap_E P_2) \subseteq_E \text{PE}(P_1)$. Como $P_1 \subseteq_E P_2$ tem-se que $\text{PE}(P_1 \cap_E P_2) \subseteq_E \text{PE}(P_1) \cap_E \text{PE}(P_2) \subseteq_E \text{PE}(P_2)$, pelo Prop. 2.1, temos que $P_1 \cap_E P_2 \equiv_E P_1$ e em decorrência $\text{PE}(P_1 \cap_E P_2) \equiv_E \text{PE}(P_1) \subseteq_E \text{PE}(P_2)$ □

Proposição 3.2.4 – Se PE é um processo evolutivo, então $\forall j \in \aleph$, vale que

$$\text{PE} \left(\bigcap_{i \in \aleph} P_i \right) \subseteq \left(\bigcap_{i \in \aleph} \text{PE} (P_i) \right)$$

Prova: Por indução finita □

Teorema 3.2.1.(Teorema do Ponto Fixo) – Se um processo PE é evolutivo e monotônico, então ele evolui para uma população de estagnação, isto é, todo processo evolutivo monotônico tem ponto fixo.

Prova: Seja $S = \bigcap_{j \in \mathbb{N}} \left\{ P_j / PE(P_j) \subseteq_E P_j \right\}$. Como PE é evolutivo, tem-se que, $\forall X \in \wp(P) X \subseteq_E PE(X)$, em particular se

$X=S$, então $S \subseteq_E PE(S)$. Por outro lado, tem-se que pelo fato de PE ser monotônico então $PE \left(\bigcap_{i \in \mathbb{N}} P_i \right) \subseteq \left(\bigcap_{i \in \mathbb{N}} PE(P_i) \right)$.

Assim a propriedade acima também é satisfeita para S, logo $PE(S) \subseteq_E S$ e deste modo $PE(S) \equiv_E S \square$

Com o teorema acima, será possível predicar sobre a convergência de um processo evolutivo. Esta qualidade serve para garantir que a evolução de uma população conduzida por um processo evolutivo monotônico, pára quando uma população estagnada é atingida, isto é, num problema, interpretados os elementos desse problema segundo a Metáfora de Darwin, a população estagnada é constituída dos resultados finais do problema, fazendo assim com que a condição de solução do problema desempenhada pelo algoritmo genético, seja realmente exercida.

3.3 – Convergência do GAADT

Nesta seção será feito o estudo da convergência do algoritmo genético GAADT como está definido em Def. 2.4.1 (Vieira 2003). Basicamente o que se pretende mostrar aqui é que o GAADT é um processo evolutivo monotônico, isto é, possui uma população de estagnação.

Lema 3.3.1 – O GAADT é um processo evolutivo.

Prova: Seja $P_n \in \wp(\text{População})$ logo, pela definição do GAADT tem-se que:

Caso 1 – Se $P_n = P_{\text{otm}}$ então a conclusão é imediata; $P_n \equiv_E \text{GAADT}(P_n)$. Logo, processo evolutivo \square

Caso 2a – Se $P_n \neq P_{\text{otm}}$ e $P_{\text{otm}} \sqsubseteq_E P_n$ então $\text{GAADT}(P_n) = P_{\text{otm}}$ e tem-se pela definição de P_{otm} , que $\text{adapt}_m(P_{\text{otm}}) \geq t$ enquanto em P_n há pelo menos um elemento c cuja $\text{adapt}(c) < t$ Assim $\text{adapt}_m(P_n) \leq \text{adapt}_m(P_{\text{otm}})$ e desse modo $P_n \sqsubseteq_E \text{GAADT}(P_n)$. Logo processo evolutivo \square

Caso 2b – Se $P_n \neq P_{\text{otm}}$, e $P_{\text{otm}} \cap_E P_n \neq \square \square$ então $\text{GAADT}(P_n) = P_{\text{otm}}$. Assim cai-se no caso 2a e desse modo GAADT é um processo evolutivo \square

Caso 2c – Se $P_n \neq P_{\text{otm}}$, e $P_{\text{otm}} \cap_E P_n = \emptyset$ então $\text{GAADT}(P_n) = P_{n+1}$, onde $P_{n+1} = \text{cruz}(a,b) \cup (\text{mut}(c) \cup \text{Rq}(P_n))$ e $\text{Rq}(P_n) \neq \emptyset$. Verificar se $\text{adapt}_m(P_n) \leq \text{adapt}_m(P_{n+1})$. Com efeito, deve-se considerar a adaptação média de P_{n+1} , ou seja, $\text{adapt}_m(P_{n+1}) = \text{adapt}_m(\text{cruz}(a,b) \cup (\text{mut}(c) \cup \text{Rq}(P_n)))$. É fácil verificar pelas definições que: $\text{adapt}_m(\{a,b\}) \leq \text{adapt}_m(\text{cruz}(a,b))$, $\text{adapt}(c) \leq \text{adapt}_m(\text{mut}(c))$, bem como $\text{adapt}_m(P_n) \leq \text{adapt}_m(\text{Rq}(P_n))$. Desse modo, temos que $\text{adapt}_m(P_n) \leq \text{adapt}_m(P_{n+1})$ e em consequência $P_n \subseteq_E \text{GAADT}(P_n)$. Logo GAADT é um processo evolutivo \square

Lema 3.3.2 – O GAADT é um processo evolutivo monotônico.

Prova: Pelo Lema 3.3.1 o GAADT é um processo evolutivo e desse modo resta mostrar que o GAADT é monotônico. Com efeito tem-se $\text{GAADT}(P_1 \cap_E P_2) \subseteq_E \text{GAADT}(P_1 \cap_E P_2)$ pela condição i da prova da Proposição 3.2.3. Como por hipótese $P_1 \subseteq_E P_2$, então $P_1 \cap_E P_2 \equiv_E P_1$ e assim teremos $\text{GAADT}(P_1) \subseteq_E \text{GAADT}(P_1 \cap_E P_2)$. Por outro lado $\text{GAADT}(P_1 \cap_E P_2) \subseteq_E \text{GAADT}(P_2)$ porque pela propriedade 3.2.2 temos $\text{adapt}_m(P_1 \cap_E P_2) \leq \text{adapt}_m(P_2)$. Assim $\text{GAADT}(P_1) \subseteq_E \text{GAADT}(P_2)$, ou seja $\forall P_1, P_2 \in \wp(P) (P_1 \subseteq_E P_2)$ então $\text{GAADT}(P_1) \subseteq_E \text{GAADT}(P_2)$ e portanto GAADT é monotônico. \square

Teorema 3.3.3 – O algoritmo genético GAADT possui ponto de estagnação, ou seja, é um processo evolutivo convergente.

Prova: Lema 3.3.1 associado ao lema 3.3.2. \square

Deve ser observado que a convergência do GAADT, estabelecida pelo teorema acima, não está subordinada a qualquer exigência especial. A sua convergência lhe é intrínseca, fazendo parte da sua definição. O significado desse teorema é de grande importância para o estudo dos algoritmos genético e em particular para o estudo do GAADT, como está proposto em (Vieira&Lopes, 1999), (Vieira&Lopes, 2000) e (Vieira, 2003).

Teorema 3.3.4 – Seja o GAADT munido de uma função de adaptação “melhor que adaptação média, isto é, $\forall P \in \wp(P) \text{adapt}_m(P) \leq \text{adapt}(P)$ ” então, o GAADT é convergente.

Prova: Decorre imediato da aplicação dos: lema 3.3.1, Lema 3.3.2, Teorema 3.3.3,

Duas observações relevantes se fazem necessárias a partir dos Lemas 3.3.1., 3.3.2 e do Teorema 3.3.4.:

A primeira: Em decorrência desses resultados o GAADT é um algoritmo convergente, isto é, possui uma população de estagnação na qual podem ser escolhidos, a partir de certo n_0 de interações, qualquer das populações resultantes imagens do GAADT;

A segunda: não foram requeridas quaisquer exigências sobre as populações iniciais ou

4. Metodologia do GAADT

4.1. – Considerações gerais

A ideia de problema, como está na TGP - Teoria Geral de Problemas (Lopes, 1981), estabelece que um problema é uma estrutura do seguinte tipo: $\wp = \langle D, R, d, r, A \rangle$, onde D e R são conjuntos não vazios, d é um predicado sobre D, r é um predicado sobre o produto cartesiano de D por R e A é um predicado de segunda ordem sobre o conjunto das funções de D em R. O significado dessas componentes são os seguintes: D é o domínio dos dados do problema, R o domínio dos resultados, d é o subconjunto de D interessante para o problema P, r é a relação intencional entre os dados e os correspondentes resultados desejados, isto é, o problema propriamente dito e A é a condição de admissibilidade sobre os mecanismos possíveis de mapeamento de dados em resultados, ou seja, as condições de admissibilidade de solucionáticas. Desse modo, uma solução para \wp é uma função *sol*: $D \rightarrow R$, que satisfaz as seguintes condições: i) $\forall f \in d (f, \text{sol}(f)) \in r$; ii) *sol* de D em R satisfaz a propriedade A, isto é, *sol* resolve \wp e denota-se por $\text{sol} \Vdash \wp$.

Na estrutura de \wp podem ser observadas duas naturezas distintas de componentes: a estrutural, formada pelos conjuntos D e R e a intencional, formada pelas componentes predicativas d, r e A. O significado dessa distinção de natureza das componentes é de muita importância para o estudo da TGP (Lopes, 1981) no que diz respeito à solubilidade de problemas e seus aspectos metodológicos e taxonômicos. Neste sentido, para uma caracterização de uma família de problemas deve-se primeiramente estabelecer a especificidades das componentes estruturais e intencionais. Por exemplo, a família dos problemas solúveis por algoritmos genéticos, entre outras características, deve ter as seguintes características necessárias:

- a) *Os conjuntos de dados e resultados devem ser da mesma natureza e, por conseguinte sua representação em \wp é a mesma, isto é, $D=R$. Isso se deve primordialmente ao fato de que esses problemas, cujas solucionáticas requeridas são algoritmos genéticos, tratam sobre populações e os resultados também são populações;*
- b) *Para que D possa ser metaforicamente visto como uma população é necessário que seus elementos sejam particionáveis em partes não necessariamente da mesma natureza dos elementos de D. Ou seja, os indivíduos de uma população, quando denotados metaforicamente por cromossomos, devem permitir, quando for o caso, uma representação em partes elementares (genes ou bases, por exemplo);*
- c) *Deve existir uma relação "θ" sobre D tal que o sistema $\langle D, \theta \rangle$ seja dirigido, isto significa que $\forall [\forall a, b \in D (a \theta b \vee b \theta a) \rightarrow \exists c \in D (a \theta c \wedge b \theta c)]$. Esta condição abre a possibilidade de que no conjunto D possam ser exibidos indivíduos que antes não o eram, ou seja, o conjunto D cresça ao longo do processo no sentido de adquirir (exibir) novos indivíduos;*

O GAADT que foi acima descrito se presta para ser aplicado a diferentes problemas. A qualidade dos resultados encontrados depende da especificação dos tipos abstratos manipulado por este algoritmo, da adequação dos operadores genéticos e das restrições impostas pelo ambiente. Portanto, o GAADT, quando instanciado para um problema particular, deve preservar todas as suas propriedades fundamentais. Em (Vieira&Lopes, 2000), (Vieira, 2003) é apresentada uma metodologia (heurística), baseada em 7 passos, para modelar os requisitos do problema \wp a ser resolvido pelo GAADT, como abaixo:

- 1 *Examinar o problema, com o objetivo de determinar qual a natureza da solução requerida no sentido de orientar a escolha metafórica dos tipos básicos: base, gene e cromossomo;*
- 2 *Identificar a representação do gene inócuo g_λ e a população inicial P_0 ;*
- 3 *Definir as funções grau de adaptação do gene grau e peso do gene no cromossomo Θ , e a relação atributo-Relevante;*
- 4 *Definir os elementos de R_q (F, M e r) usados pela função cruz de acordo com o problema;*
- 5 *Definir os elementos de R_q (pcorte, t e k) utilizados pela função GAADT;*
- 6 *Construir o algoritmo;*
- 7 *Verificar os eventuais ajustes relativos à verificação dos tipos manipulados pelas funções, relações e operações manipuladas pela função GAADT, e a modelagem do problema através da definição dos componentes do ambiente.*

Em (Vieira, 2003) alguns modelos de algoritmos genéticos foram sedimentados sobre os seguintes princípios:

- para uma dada classe de problemas existe uma função de codificação que mapeia os possíveis resultados do problema na representação adotada para o cromossomo;
- existe uma função de adaptação que pode ser aplicada em cada cromossomo, a qual informa o quão adaptado ao ambiente este cromossomo está;
- os cromossomos mais adaptados geram descendentes, os quais tendem a herdar as características adaptadas do cromossomo ancestral;
- os cromossomos menos adaptados tendem a desaparecer.

Debaixo da reflexão sobre algoritmos evolucionários, a respeito do primeiro princípio (Vieira, 2003), o GAADT foi proposto seguindo a ideia de que para cada classe de problema, dependendo da solução adotada, uma dada representação parece ser mais adequada do que a outra. Assim preferiu-se adotar o conceito de tipo abstrato de dados e uma representação do cromossomo estratificada em três níveis de percepção (cromossomo, gene e base) para modelar uma representação do

cromossomo independente da solução adotada e do problema a ser tratado. Dependendo da interpretação dada aos tipos (cromossomo, gene e base) é possível reproduzir qualquer uma das representações de cromossomos existentes.

O segundo princípio estabelece que a função de adaptação de um cromossomo deve ser compatível com o ambiente no qual este está inserido, sugerindo que, se o ambiente mudar, tal função deveria em princípio ser capaz de redirecionar a sua busca à procura do cromossomo mais adaptado para a nova configuração do ambiente. O GAADT trabalha sobre um ambiente definido como uma estrutura na qual um dos componentes é a população. Segundo este modelo, as mudanças ambientais são vistas como o marco do início de um novo período de evolução durante o qual os cromossomos da população atual, no processo de funcionamento do GAADT, irão sofrer a ação dos operadores genéticos com o intuito de construir uma nova população formada somente por cromossomos que satisfaçam aos requisitos do ambiente em questão. Após o período de evolução, se o ambiente permanecer estável, vem o período de estagnação, durante o qual a população não evolui. O período de estagnação é finalizado quando uma nova alteração ambiental, ocorre, dando início a um novo ciclo de um período de evolução seguido por um período de estagnação (Vieira, 2003). O resultado do problema para o ambiente atual, dentro do processo, é o cromossomo mais adaptado na população de estagnação atingida, se esta for compatível com uma condição de parada, a qual deverá ser definida durante o período de concepção do GAADT. A cada período de evolução do algoritmo existe uma população de cromossomos mortos associada, os quais são provenientes de uma das populações já trabalhadas pelo algoritmo genético e que já foram avaliadas e descartadas devido ao seu baixo grau de adaptação em comparação às configurações do ambiente mais evoluído.

O terceiro princípio fala sobre a hereditariedade das características adaptadas ao ambiente dos cromossomos-pais que devem ser passadas para os cromossomos-filhos. Neste ponto, temos uma diferença entre o GAADT e os demais algoritmos encontrados na literatura, onde no GAADT o operador de cruzamento constrói novos cromossomos somente com as características responsáveis pela adaptação dos cromossomos-pais ao ambiente, as quais serão denominadas de genes dominantes. Já nos demais algoritmos os operadores de cruzamento só se preocupam com o fato de que as características apresentadas nos cromossomos-filhos estejam presentes nos cromossomos pais. O último princípio descarta a possibilidade dos cromossomos não adaptados ao ambiente evoluírem e até conduzirem à geração de cromossomos mais adaptados do que os cromossomos mais adaptados da população atual. No GAADT os cromossomos não adaptados, antes de desaparecerem, serão todos submetidos à ação do operador genético de mutação, como uma forma de garantir a presença das características adaptadas ao ambiente destes organismos nas próximas gerações. No GAADT há uma história genealógica (táxon) dos cromossomos da população atual, para que uma explicação sobre o resultado encontrado para um dado problema possa ser gerada sempre que necessário.

5. Conclusão

Neste trabalho, foi apresentado um esboço de uma introdução a uma teoria para os algoritmos genéticos independente das interpretações dos seus tipos. Isso permitiu que através de representações adequadas dos seus tipos, se possa reproduzir alguns dos algoritmos genéticos conhecidos ou seus equivalentes. Desse modo será possível estabelecer comparações entre os vários algoritmos independentemente de testes e resultados efêmeros e particulares.

Deixando para depois, uma avaliação mais profunda do significado do algoritmo GAADT, comparativamente aos demais algoritmos genéticos, é possível observar, desde já, que o GAADT responde de imediato às indagações quanto ao tamanho fixo ou variável da população e a questão da representação. No GAADT, o tamanho da população é variável e desta forma, apresenta a vantagem de não estar submetido à tão importante restrição, como é o caso de outros algoritmos inspirados pelo algoritmo de Holland (Holland, 1975). Deve ser observado também, que a adoção de TAD para o desenvolvimento do modelo GAADT o torna independente da implementação que posteriormente será adotada.

A adoção dessa teoria, como ambiente teórico para a descrição de algoritmos genéticos, ainda requer resultados mais relevantes sobre a sua abrangência, o que será feito em trabalhos futuros. O modelo GAADT tem sido aplicado em vários campos de conhecimento a exemplo de (Ferreira, 2011) e (De Melo, 2011)

Neste trabalho ficou mostrado de forma sucinta que a ideia de tratar algoritmos genéticos através de um ambiente formal apropriado para a sua concepção, independentemente de representação, deixando esta para o momento de uma aplicação específica simplifica, em muito, a abordagem do problema e torna mais sistemática a sua especificação. Outra vantagem é que o algoritmo genético específico resultante dessa forma de abordagem é correto por construção.

Agradecimentos.

Os autores gostariam de agradecer a CAPES que através do Programa Nacional de Cooperação Acadêmica - PROCAD 1493/2007 - IC/UFAL-CIn/UFPE ensejou este trabalho.

Referências

Basile R., e Magalhães L. E.,(1975) *Citologia e Genética*, Ed. CULTRIX, São Paulo.

Bhagel, A. S. & Rastogi, R., (2011) *Effective Approaches for solving large travelling salesman Problems with small populations*.

International journal of Advances in Engineering <http://ijaer.com/index.html> (IJAER) , VOL No I, Issue No 1, january

- Darwin, C. (1959) *A Origem das Espécies*, Trad do Original: On the Origin of Species, Londres 1859, Hemus Editora LTDA, 5ª Edição, São Paulo.
- De Jong K. A.,(1992) *Genetic Algorithms Are Not Function Optimizers*. Foundations of Genetic Algorithms, Mosgan Kaufemann.
- De Melo, B R. P. (2011) *Um Sistema Adaptativo Para Detecção de Ondas de Eletrocardiografia*, Dissertação de Mestrado, Programa de Pós-Graduação em Modelagem Computacional de Conhecimento, Instituto de Computação Universidade Federal de Alagoas, Maceió;
- Denett, D. C., (1995) *A Perigosa Ideia de Darwin: a evolução e os significados da vida*, Editora Rocco LTDA - Ciência Atual, Rio de Janeiro.
- Ferreira, A. M. V. (2011) *Uma Especificação do GAADT para Diagnóstico de Arritmia Ventricular a ser Aplicada no Eletrocardiograma*, Dissertação de Mestrado, Programa de Pós-Graduação em Modelagem Computacional de Conhecimento, instituto de Computação da Universidade Federal de Alagoas, Maceió
- Futuyma D. J., (1992) *Biologia Evolutiva*, 2ª ed., Ribeirão Preto, Sociedade Brasileira de Genética/CNPq.
- Goldberg, D. E. (1992)
- Goldberg, D. E. (1994) *An analysis of boltzmann tournament selection: Part ii: An experimental analysis of Boltzmann tournament selection*, Illinois Genetic Algorithms Laboratory Report no 94007, University of Illinois.
- Goldberg, D. E. (1997) *Simple Genetic Algorithms and the minimal, deceptive problem, genetic algorithms and Simulated annealing*, Pitman.
- Gould S. Jay,(1992) *Darwin e os Grandes Enigmas da Vida*, 2ª ed., São Paulo: Martins Fontes, 1992.
- Holland, J. H. (1975) *Adaptation in Natural Artificial Systems*, University of Michigan, Michigan USA
- Lopes, M. A., (1981) “*Introdução a uma Teoria Geral de Problemas*”, Tese de Doutorado, Departamento de Informática da PUC/RJ, Rio de Janeiro.
- Michalewicz Z.,(1996) *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag, Berlim.
- Mustafa, K. (2011), *The effects of two new operators on genetic performance*. Journal Applied soft computing, volume 11issue1,pages 881-890.
- Roman, S., (1995) *Fiel Theory*, Springer-Verlag, New York.
- Roman, S., (1995), “Fiel Theory”, Springer-Verlag, New York.
- Thamilselvan, R., (2011) *Analysis of various alternate crossover strategies for genetic algorithm to solve job shop scheduling problems*. European Journal of Scientific Research, ISSN 1450-216X, vol 64, No 4, pp538-554.
- Vieira, R. V. & Lopes, M. A., (1999) *Uma Abordagem Teórica Inicial para os Algoritmos Genéticos através de Tipos Abstratos de Dados*, Relatório Técnico do Departamento de Informática da Universidade Federal de Pernambuco, RT-DI/UFPE Nº 002/99, Recife.
- Vieira, R. V. & Lopes, M. A., (1999a). *An Application of Abstract Data Types in Genetic Algorithms Construction: The Example of the Travelling Salesman Problem*, Anais da XIX International Conference of the Chilean Computer, IEEE, Recife/Pe
- Vieira, R. V. & Lopes, M. A., (2000) *A Construction of Phylogenetic Trees by Genetic Algorithms*, The 2000 International Conference on Mathematics and Engineering Techniques in Medicine and Biology Science (METMBS’2000), Monte Carlos, Las Vegas, Nevada, USA.
- Vieira, R. V. & Lopes, M. A., (2001) *Alinhamento de DNA’s via Algoritmo Genético*, Encontro Nacional de Inteligência Artificial - ENIA2001, Fortaleza CE.
- Vieira, R. V., (2003) *Algoritmo Genético baseado em Tipos Abstratos de Dados e sua Especificação em Z*, Tese de Doutorado em Ciência da Computação, Centro de Informática da Universidade Federal de Pernambuco, Recife PE.
- Winter, D. J., (1974) *The structure of Field*, Springer, New York.
- Yuret, D., (1994) *From Genetic Algorithms to Efficient opt5mization*, M. Sc. These, University of Massachusetts