

ALGORITMO GENÉTICO: REVISÃO HISTÓRICA E EXEMPLIFICAÇÃO

Ana Paula Grosko¹, José Robeto Gorski², João da Silva Dias³

^{1,2}Alunos do Programa de Pós-graduação em Tecnologia em Saúde (PPGTS)

Pontifícia Universidade Católica do Paraná – PUCPR

³Coordenador e Professor do Programa de Pós-graduação em Tecnologia em Saúde (PPGTS)

Pontifícia Universidade Católica do Paraná - PUCPR

Resumo - Este artigo tem como objetivo descrever de forma clara um levantamento teórico referente ao algoritmo genético (AG), bem como um exemplo que será a minimização de uma equação matemática.

Palavras-chave: Algoritmo Genético; Computação Evolutiva.

Abstract - This article has as objective to describe of clear form a referring theoretical survey to the genetic algorithm (GAC), as well as an example that will be the minimização of a mathematical equation.

Key-words: Genetic algorithm; Evolutiva computation.

Introdução

Em meados da década de 70, John H. Holland propõem a técnica de algoritmo genético (AG), inspirada nas teorias darwinianas. Deste estudo surgiu a publicação “*Adaptation in Natural and Artificial Systems*” [5], [18].

O AG é considerado um método robusto, utilizado basicamente para resolver problemas em pesquisas numéricas, otimização de funções e aprendizagem de máquina, dentre outras áreas [6].

A aplicação do AG é destacada em sistemas classificadores de dados que ocorrem geralmente na ordenação destes dados para um propósito, por exemplo, para uma simples recuperação ou para efetuar uma análise de dados [9].

Whitley [20] cita que o AG é frequentemente descrito como um método de busca global, não utilizando gradiente de informação e podendo ser combinado com outros métodos para refinamento de buscas quando há aproximação de um máximo ou mínimo local.

Atualmente o AG é muito empregado na resolução de problemas de bioinformática, como exemplo, a descoberta da estrutura de RNA [7], para a predição de “*non-coding*” RNA (ncRNA) [17], para a construção de mapas de DNA [19] e para a descoberta de regiões regulatórias [1], [8].

Computação evolutiva

A computação evolucionária se inspira na teoria evolutiva para o desenvolvimento de métodos computacionais. Na computação

evolucionária o algoritmo mais conhecido é o algoritmo genético (AG) [2], [3].

O algoritmo genético é uma subdivisão do algoritmo evolucionário, onde também se encontra a programação evolucionária (PE) e a estratégia evolucionária (EE). Todos partilham de uma base conceitual comum, que consiste na simulação da evolução de estruturas individuais, via processo de seleção e os operadores de busca, referidos como operadores genéticos (OG), tais como mutação e “*crossover*” (cruzamento ou recombinação). Todo o processo depende do grau de adaptação, ou seja, do “*fitness*” (aptidão), do indivíduo frente ao ambiente. A seleção, inspirada na seleção natural das espécies, preconiza que os indivíduos mais aptos ou com melhor grau de adaptação ao meio terão maiores chances de repassar o seu material genético para as próximas gerações. Assim, quanto maior a aptidão do indivíduo, maiores são as chances do material genético deste estar presente na próxima geração.

Algoritmo genético e sua inspiração biológica

A primeira teoria sobre evolução das espécies foi proposta em 1809, pelo naturalista francês Jean Baptiste Pierre Antoine de Monet, conhecido como Lamarck. Para Lamarck as características que um animal adquire durante sua vida podem ser transmitidas hereditariamente, este estudo ficou conhecido pela ciência como a “lei do uso e desuso” [4].

Charles Darwin vem debater a teoria de Lamarck, de forma agressiva tentando de forma científica explicar como as espécies evoluem. A

seleção natural é um processo de evolução, geralmente aceito pela comunidade científica como a melhor explicação para a adaptação. O meio ambiente seleciona os seres mais aptos, em geral, só estes conseguem reproduzir-se e os menos adaptados são eliminados ou pelo menos reduzidos em um primeiro momento a uma minoria. Assim, só as diferenças que facilitam a sobrevivência são transmitidas à geração seguinte [16].

A seleção natural depende muito das condições ambientais, podendo selecionar características de um determinado organismo ajudando na reprodução e sobrevivência deste, os organismos que não possuem tal característica podem vir a morrer antes que se reproduzam ou serem menos prolíficos que os organismos que apresentam a característica. À medida que as condições ambientais não variem essas características continuam sendo adaptativas tornando-se comum na população. Certas características são preservadas devido à vantagem seletiva que conferem, permitindo que o indivíduo reproduza-se, conseqüentemente, deixe mais descendentes, através de diversas interações, os organismos podem vir a desenvolver características adaptativas muito complexas [10], [16].

A teoria da seleção natural não é aplicada somente a organismos biológicos, podendo ser aplicada a qualquer organismo que se reproduz de modo a envolver tanto a hereditariedade como a variação [15]. Assim, pode-se ocorrer seleção natural no reino não biológico.

Alguns pesquisadores buscaram na natureza a inspiração necessária para pesquisar e desenvolver novas técnicas de busca de soluções para determinados problemas. Na natureza o processo de seleção natural demonstra que seres mais preparados (aptos) compete com os recursos naturais impostos, tendo assim uma maior probabilidade de sobreviver, conseqüentemente, disseminar o seu código genético [18]. Com o passar das gerações, através de sucessivos cruzamentos e mutações que ocorrem com as espécies, estes tendem a estar cada vez mais adaptados ao meio ambiente em que vivem.

O AG trabalha com uma população no qual cada indivíduo é um candidato a solução do problema. A função de otimização representa o ambiente no qual a população inicial encontra-se. Emprega-se no AG a mesma terminologia e os mesmos princípios da teoria evolutiva e da genética [5].

O AG relacionado com a seleção natural pode ser expressado como [4].

- 1 – SE há organismo que se reproduzem;
- 2 – SE os descendentes herdam as características de seus genitores;
- 3 – SE há variação nas características;
- 4 – SE o ambiente não suporta todos os indivíduos de uma população em crescimento;
- 5 – ENTÃO os indivíduos que apresentarem menor adaptação (determinadas pelo ambiente) morrerão;
- 6 – ENTÃO os indivíduos que apresentarem maior grau de adaptação (determinadas pelo ambiente) prosperarão.

Como resultado desse processo tem-se a evolução das espécies.

Componentes do algoritmo genético

Geralmente existem apenas dois componentes principais utilizados no AG, que dependem do problema a ser resolvido: a representação do problema e a função de adaptação [20]. Os outros componentes que completam este processo são: população, seleção, cruzamento e a mutação [11].

Fundamental para estrutura do AG é o mecanismo que será utilizado para a representação de uma problematização, dependendo essencialmente de sua natureza para ser resolvido. Esta representação pode usar números binários (0 e 1), números inteiros ou reais [15], [18].

Geralmente, cada possível solução (indivíduo ou cromossomo), possui um tamanho fixo. Por exemplo, se for usado a representação binária (0 e 1, representada por K) e o indivíduo ter tamanho (T) igual a 6, todos os indivíduos terão o mesmo tamanho. Por exemplo, 011101 e 111010, conclui-se que a quantidade de indivíduos possíveis para esta representação será 2^6 , ou generalizando K^T , onde K pode variar de acordo com o alfabeto utilizado. O importante desta representação é que cada indivíduo representa um ponto no espaço das possíveis soluções para o problema [13].

A função de adaptação é a função que deve ser otimizada, ela possui o mecanismo de evolução para cada indivíduo [18]. Também conhecida como função objetivo. Ela representa o meio e permitirá avaliar cada indivíduo da população, gerando uma pontuação de acordo com o desempenho deste. O desempenho em última análise representará a chance deste indivíduo participar do processo reprodutivo nas próximas gerações. A avaliação é independente, mas a qualidade do seu grau de adaptação ao ambiente vai depender dos demais indivíduos da população [20].

A população é um conjunto de indivíduos candidatos à solução do problema. A população é uma matriz com duas dimensões, podendo ser representado como $N_p \times N_T$, onde N_p é a quantidade de indivíduos de uma determinada população e N_T é o tamanho de cada indivíduo [11].

O tamanho da população irá depender do problema. Assim quanto maior a população, maior a chance de encontrar a solução para o problema. Porém, maior será o tempo de processamento, ou seja, a escolha do tamanho de uma população irá depender de alguma heurística utilizada pelo usuário e de sua experiência [5]. Caso o tamanho da população aumente muito, o processo estará se aproximando de uma busca exaustiva. Portanto, deixa ser interessante como método alternativo de busca de soluções.

Na natureza o processo de seleção garante que os indivíduos mais adaptados tenham maiores chances de sobreviver. O conceito utilizado no AG é o mesmo [18].

A seleção é aplicada na população corrente para criar uma população intermediária que irá passar pelos processos de cruzamento e mutação (que serão explicados mais adiante) para a geração da próxima população [20].

Existem várias técnicas de seleção, a utilizada no algoritmo clássico, sugerido por Holland (1975) é conhecida como *roulette wheel* [15]. Nesta técnica é atribuída a cada indivíduo uma probabilidade de passar para a próxima geração. Essa probabilidade é proporcional a sua adaptação ao ambiente, em relação à somatória da adaptação de todos os indivíduos, sendo maior a probabilidade de dos indivíduos mais adaptados serem sorteados [21].

Durante o processo de seleção pode-se perder um indivíduo com um alto grau de adaptação, para evitar este problema usa-se um conceito conhecido como elitismo, onde o melhor indivíduo é mantido na próxima população. Desta forma, garante-se que o seu material genético fará parte do processo na próxima geração [21].

Existem outras técnicas para o processo de seleção, que podem ser estudadas em [6,11,13,18,20 e 21].

Segundo Srinivas & Patnaik [18], após a seleção vem o cruzamento, sendo esta operação essencial para o AG, onde indivíduos pré-selecionados formam pares aleatórios para o processo de cruzamento.

O cruzamento consiste na manipulação do material genético existente na população e permitirá a criação de um ou mais indivíduos (filhos) dos indivíduos selecionados (pais) pela seleção. No final deste processo a população

geralmente permanece do mesmo tamanho da população anterior [11].

Como esclarece Dias & Barreto [5] e Zuben [21], existem várias formas de cruzamento, dentre elas destacam-se: cruzamento uniforme, de 1-partição, de 2-partições e cruzamento com (n)-partições. Cada uma delas apresenta características que podem ou não ser importante, tudo dependerá do problema (ambiente) e dos parâmetros utilizados para o AG.

A mutação permite que indivíduos da nova geração sofram pequenas alterações, permitindo assim uma possibilidade de busca maior no espaço do problema. A mutação possui o papel de repor ou acrescentar um material genético inexistente na população atual, seja por ter sido perdido ou por nunca ter existido em populações anteriores. O processo inicia-se com a escolha de um ponto aleatório de um indivíduo, depois é aplicado uma taxa de probabilidade de troca deste ponto (*bit*) por um outro *bit* [13].

Whitley [20], explica que a taxa de mutação geralmente é pequena, seguindo a inspiração biológica esse valor é da ordem de 1%. Contudo, na prática pode-se trabalhar com taxas bem maiores, dependendo do objetivo, topologia da superfície de erro e até mesmo se será ou não empregado o elitismo no processo de busca de uma solução. Para Srinivas & Patnaik [18], a mutação é um operador secundário que pode restaurar material genético perdido ou inexistente em gerações anteriores.

Percebe-se em problemas práticos que o AG pode tender a gerar indivíduos muito parecidos, perdendo diversidade em sua população. E esta população não necessariamente esta direcionada para o ponto ótimo da resolução do problema proposto. Para minimizar a perda de diversidade, pode-se usar uma taxa de mutação em níveis mais altos que sugeridos na literatura.

Teorema fundamental do algoritmo genético

Os conceitos anteriormente explicados são a base para a equação matemática que explica o funcionamento do AG.

Também conhecida como Teoria do esquema, foi proposta por Holland em 1975 [20].

Um esquema pode ser definido como sendo um subconjunto de um indivíduo com certas posições similares [12]. Um exemplo pode ser visto na tabela 1:

Tabela 1: Exemplo de um indivíduo e 3 possíveis esquemas.

Indivíduo	11011
-----------	-------

Esquema (E1)	1***1
Esquema (E2)	11**1
Esquema (E3)	**01*

Os esquemas tem duas propriedades que os quantificam: A ordem do esquema ($O(E)$) e o comprimento do esquema ($\delta(E)$).

A ordem de um esquema E , $O(E)$, representa o número de 0's e 1's fixos no esquema, por exemplo no esquema 11**1, a ordem $O(E) = 3$.

O comprimento do esquema E , $\delta(E)$, representa a distância entre a primeira e a última posição de interesse no esquema. Por exemplo o esquema 1***1 tem comprimento $\delta(E) = 5 - 1 = 4$, pois a última posição é 5 e a primeira é 1 [5].

Srinivas & Patnaik (1994), representam o teorema fundamental do AG como segue abaixo:

$$N(h, t+1) \geq N(h, t) \frac{f(h, t)}{f'(t)} \left[1 - pc \frac{\delta(h)}{l-1} - pm o(h) \right]$$

Onde:

$f(h, t)$: É o valor médio da adaptação do esquema h na geração t ;

$f'(t)$: É o valor médio da adaptação da população na geração t ;

pc : Probabilidade de cruzamento;

pm : Probabilidade de mutação;

$\delta(h)$: Comprimento definido de um esquema;

$o(h)$: Ordem de um esquema h ;

$N(h, t)$: Número esperado de instâncias de um esquema h na geração t ;

l : É o número de *bits* em uma *string*.

Funcionamento do algoritmo genético

Este tópico foi descrito com base nas seguintes referências da literatura, [5], [11], [15], [18], [20], [21].

O algoritmo genético básico envolve seis passos: tendo início na geração da população; avaliação da população; teste de convergência ou critério de término para a otimização; seleção e aplicação dos operadores do AG; e criação de uma nova geração. Podemos visualizar o funcionamento do AG em etapas que definem: a função de adaptação; as variáveis e parâmetros do AG; com isso gera-se a população inicial que avaliará cada indivíduo. O critério de término caracteriza-se: a seleção de indivíduos; o processo de cruzamento; o processo de mutação e encerra-se com a avaliação de cada indivíduo, munidos dos valores para impressão.

A primeira etapa é a definição de qual será a função para representar o problema, esta é a "chave" para um resultado satisfatório do AG.

Após a definição da função pode-se acrescentar a parametrização do sistema, ou seja, é neste momento que as variáveis são inicializadas, alguns exemplos de variáveis que devem ser iniciadas são: tamanho da população; quantidade de geração; taxa de cruzamento; taxa de mutação; tamanho do indivíduo entre outros.

O terceiro passo é a geração da população de forma aleatória com a quantidade de indivíduos e seu tamanho determinado em parâmetro, com distribuição uniforme pelo espaço de busca. A quinta etapa é analisar o critério da parada do sistema, que pode ser pelo número de gerações estabelecidas, pelo tempo de execução ou algum outro indicador.

A seleção dos indivíduos da população é um passo importante, ou seja, os indivíduos selecionados têm a maior probabilidade de participarem do processo de escolha para uma nova geração. O melhor material genético tem maior chance de ser selecionado. Após a escolha, são aplicados os operadores de cruzamento e/ou mutação. Primeiramente, o cruzamento, dividindo a população em pares de cromossomos da população atual e a cada novo indivíduo aplica-se o processo de mutação, gerando desta forma uma nova população em substituição a anterior. Os indivíduos da nova geração são novamente avaliados pela função de adaptação, começando assim um novo ciclo a partir da quinta etapa, até a condição de término ser atendida.

Finalmente, quando a condição de término é atendida o indivíduo mais apto é considerado como um forte candidato para ser a solução que estava sendo procurada. Em última análise, a impressão do resultado final demonstra as informações decorrente a necessidade que o usuário tem, podendo, por exemplo, mostrar o melhor indivíduo da última geração, ou mostrar a última população inteira, ou qualquer indivíduo que se possa extrair as informações.

Aplicação prática

A aplicação prática deste trabalho visa minimizar uma equação simples, onde poderá ser analisada a relação entre os principais operadores do AG (Seleção, Cruzamento e Mutação).

O programa foi desenvolvido na linguagem MATLAB, versão 7.0 (R14), usando como ferramenta o *ToolBox* específico para o AG.

O código fonte do arquivo (.m) (nome do arquivo *my_fun.m*) é como segue abaixo:

function $z = my_fun(x)$
 $z = x(1)^2 - 2*x(1)*x(2) + 6*x(1) + x(2)^2 - 6*x(2);$

Matematicamente esta função é igual a: $x_1^2 - 2x_1x_2 + 6x_1 + x_2^2 - 6x_2$.

Na tabela 2 apresenta-se algumas parametrizações do AG com variações em seus operadores, para algumas execuções (Sim.1, Sim. 4 e Sim. 7) foram geradas figuras com a evolução das gerações e os melhores indivíduos entre todas as gerações.

	Sim. 1	Sim. 2	Sim.3	Sim.4	Sim.5	Sim.6	Sim.7
População	30	30	30	30	30	30	30
Elitismo	2	2	0	0	2	2	2
Taxa Cruzamento	80%	80%	80%	80%	80%	0	80%
Taxa Mutação	1%	2%	1%	2%	0	2%	20%
Gerações	100	100	100	100	100	100	100
Valor x1	0,0300	0,0084	0,0821	0,0681	0,1130	0,022	0,0094

Tabela 2: Sete tipos diferentes de parametrização para o AG, bem como seus resultados.

O tipo de seleção utilizada foi *roulette wheel*, a distribuição da população foi uniforme e o cruzamento foi do tipo 1-partição.

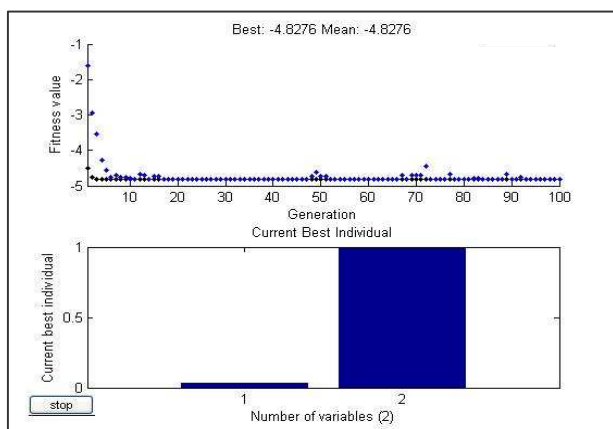


Figura1: Referente aos parâmetros da Tabela 2, coluna Sim.1.

Observou-se nas execuções do algoritmo que o operador da mutação é fundamental para uma melhor convergência para o mínimo global (que é a melhor solução possível, diferente do mínimo local que é uma solução boa somente para um determinado ponto do problema). O elitismo contribuiu para acelerar o processo, pois este sempre preserva o melhor indivíduo da geração atual para a próxima geração, sendo substituído quando um indivíduo de melhor escore surge.

Taxas de mutação mais altas ajudam a uma distribuição maior pelo espaço de busca, mas

deixam o AG com aspecto de busca aleatória/exaustiva, como pode ser visto na figura 3.

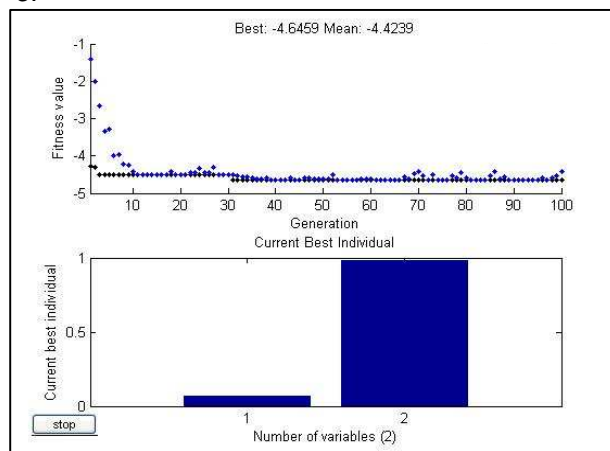


Figura2: Referente aos parâmetros da Tabela 2, coluna Sim.4.

O cruzamento sem taxa de mutação, tende a convergir rapidamente para um ponto e nele se manter constante com o passar das gerações.

Porém, este ponto não necessariamente será um bom resultado para o problema.

Sem cruzamento e mesmo com uma taxa de mutação relativamente baixa, obteve-se melhor resultado devido à cobertura maior pelo espaço de busca.

Para a execução deste exemplo foi usado um Pentium IV, 2,8 MHZ, com 768MB de RAM, o tempo de execução foi relativamente baixo para

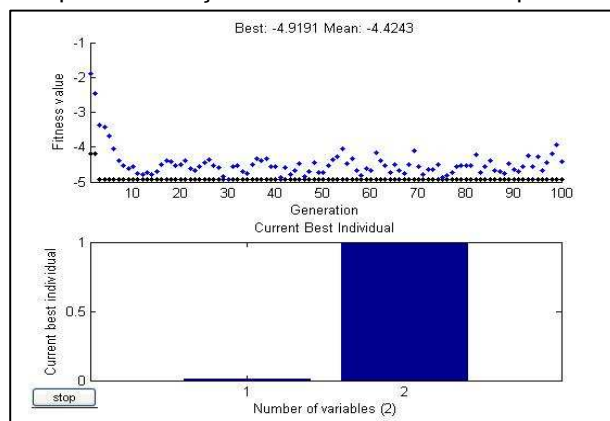


Figura3: Referente aos parâmetros da Tabela 2, coluna Sim.7.

todas as execuções, não ultrapassando 30 segundos para cada execução.

Discussão e Conclusão

Dentre os principais fatores que tem feito o AG uma técnica bem sucedida, pode-se destacar a simplicidade das operações e a facilidade de

implementação, eficácia na busca da região onde provavelmente encontra-se o máximo ou mínimo global e um bom desempenho em sua execução.

Referências Bibliográficas

- [1] Aerts, S.; Loo, P.V.; Moreau, Y.; Moor, D.B. (2004), "A genetic algorithm for the detection of new cis-regulatory modules in sets of coregulated genes", *Bioinformatics*, Oxford University Press.
- [2] Azevedo, F.M. (1999), "Algoritmos genéticos em redes neurais artificiais", V Escola de Redes Neurais, promoção: Conselho Nacional de Redes Neurais – ITA, São José dos Campos, SP.
- [3] Barreto, J.M. (1996), "Conexionismo e a Resolução de Problemas", *Titular Professor Contest Dissertation*, Departamento de Informática e Estatística, UFSC, Florianópolis, SC.
- [4] Darwin, C. (2004), "A origem das espécies". Rio de Janeiro: Ediouro.
- [5] Dias, J.S.; Barreto, J.M. (1998), "Algoritmo genético: inspiração biológica na solução de problemas - uma introdução", *Revista Marítima Brasileira - Suplemento Especial*, Pesquisa Naval, nº 11, p. 105-128.
- [6] Fogel, D.B. (1995), "Evolutionary Computation", IEEE Press: Piscataway, NJ.
- [7] Fogel, G.B.; Porto, V.W.; Weekes, D.G.; Griffey, R.H.; Mcneil, J.A.; Lesnik, E.; Ecker, D.J.; Sampath, R. (2002), "Discovery of RNA structural elements using evolutionary computation", *Nucleic Acids Research*, v.30, nº 23, p. 5310-5317.
- [8] Fogel, G.B.; Weekes, D.G.; Varga, G.; Dow, E.R.; Harlow, H.B.; Onyia, J.E.; Su, C. (2004) "Discovery of sequence motifs related to co-expression of genes using evolutionary computation", *Nucleic Acids Research*, v. 32, nº 13, p. 3826-3835.
- [9] Furtado, J.C.; Lorena, L.A.N. (1998), "Algoritmo genético construtivo na otimização de problemas combinatórios de agrupamentos", III Oficina de cortes e empacotamento, lac.inpe.br.
- [10] Futuyama, D.J. (2003), "Biologia evolutiva", Ribeirão Preto: FUNPEC-RP.
- [11] Haupty, R.L.; Haupty, S.E. (2004), "Practical genetic algorithm", 2º ed. A John Wiley & Sons, Inc., Publications.
- [12] Holland, J.H. (1975), "Adaptation in natural and artificial systems", University of Michigan Press.
- [13] Koza, J.R. (1995), "Survey of genetic algorithms and genetic programming", *Proc. of Wescon 95*, IEEE Press, p. 589-594.
- [14] Matlab (2005), "Genetic algorithm and direct search toolbox user's guide", © COPYRIGHT 2004–2005 by The MathWorks, Inc.
- [15] Michalewicz, Z. (1996), "Genetic algorithm + data structures = evolution programs", 3º ed. Springer-Verlag.
- [16] Stearns, S.C. (2003), "Evolução: uma introdução", Atheneu: São Paulo.
- [17] Satrom, P.; Sneve, R.; Kristiansen, K.I.; Snove, O.; Grünfeld, T.; Rognes, T.; Seeberg, E. (2005), "Predicting non-coding RNA genes in *Escherichia Coli* with boosted genetic programming", *Nucleic Acids Research*, v.33, nº 10, p. 3263-3270.
- [18] Srinivas, M.; Patnaik, L.M. (1994), "Genetic algorithms: A survey", IEEE.
- [19] Walker J.D.; File, P.E.; Miller, C.J.; Samson, W.B. (1994), "Building DNA maps, a genetic algorithm based approach", *Advances in molecular bioinformatics*. S. Schulze-Kremer, IOS Press. p. 179-199.
- [20] Whitley, D. (1994), "A genetic algorithm tutorial", *Springer Science + Business Media B.V.*, Formerly Kluwer Academic. p. 65-85.
- [21] Zuben, F.J.V. (2000), "Computação evolutiva: uma abordagem pragmática", Tutorial: Notas de aula da disciplina IA707, DCA/FEEC/Unicamp.

Contato

- 1- Ana Paula Grosko, bióloga. E-mail: anap.g@terra.com.br. Aluna de Mestrado da Pontifícia Universidade Católica – PUCPR.
- 2- José Roberto Gorski – informata. E-mail: jrgorski@terra.com.br. Aluno de Mestrado da Pontifícia Universidade Católica – PUCPR.
- 3- João da Silva Dias, engenheiro. E-mail: jdias@ppgia.pucpr.br. Coordenador do curso de Mestrado em Tecnologia em Saúde da Pontifícia Universidade Católica – PUCPR.