Ministério da Educação

Universidade Tecnológica Federal do Paraná

Pró-Reitoria de Pesquisa e Pós-Graduação

Relatório Final de Atividades

Algoritmos Genéticos Paralelos para a Inferência de Redes Gênicas

Leandro Takeshi Hattori
Voluntário
Tecnologia em Análise e Desenvolvimento de Sistemas
Data de ingresso no programa: 08/2012
Orientador: Prof. Dr. Fabrício Martins Lopes

Área do Conhecimento: 10300007 - Ciência da Computação

CAMPUS CORNÉLIO PROCÓPIO, 2013

LEANDRO TAKESHI HATTORI
FABRÍCIO MARTINS LOPES

Algoritmos Genéticos Paralelos para a Inferência de Redes Gênicas

Relatório Científico do Programa de Iniciação
Científica da Universidade Tecnológica
Federal do Paraná.

CAMPUS CORNÉLIO PROCÓPIO, 2013

Sumário

INTRODUÇÃO

Um organismo pode ser estudado com um aglomerado de reações bioquímicas. Uma complexa rede é formada pelo envio e recebimento de mensagens efetuadas por estas reações. Estas redes vêm sendo alvo de diversos estudos com o objetivo de entender os mecanismos de controle celular com base em entidades biológicas como, por exemplo, genes e RNA (*Ribonucleic Acid*). Entretanto, ainda existe muito a ser descoberto sobre os mecanismos de controle celular.

Um modo de entender os mecanismos de controle celular é observando os dados temporais dos níveis de expressão dos genes. Com a evolução de técnicas de extração de informação molecular se tornou possível analisar grandes quantidades de genes e seus níveis de expressões como, por exemplo, a técnica de RNA-Seq [58]. Um dos grandes desafios é conseguir recuperar uma GRN (*Gene Regulatory Network*) com base nos dados de expressão, devido uma grande quantidade de variáveis (genes) e poucos experimentos (amostras) produzidos. Então, métodos computacionais e estatísticos têm sido desenvolvidos buscando inferir GRNs com maior grau de similaridade possível a rede biológica.

A inferência de GRNs é fundamentada no dogma central da biologia molecular, o qual se baseia no estado funcional de um organismo a partir da expressão de seus genes [19]. Portanto, ao termos o conhecimento de uma GRN é possível analisar informações sobre seu funcionamento e comportamento celular. Como, por exemplo, o funcionamento de diversas vias regulatórias, ciclo celular, bem com o mapeamento de alterações provocadas por estímulos. Dado aos argumentos, tal problema é representado como um dos grandes desafios da bioinformática e alvo de pesquisas como o projeto DREAM (*Dialogue for Reverse Engineering Assessment and methods*)[54].

Com uma GRN definida é possível entender diversas características biológicas como suas interações moleculares e suas funções biológicas. Então, tais redes podem ser utilizadas para estudos de doenças e gerar prognóstico mais específicos, bem como medicamentos mais eficazes [11], e realizar estudos mais aprofundados sobre doenças como, por exemplo, o câncer.

Para a representação de uma GRN é possível utilizar genes binários e funções booleanas que definem a dinâmica de uma GRN por meio de um circuito lógico [34]. Ou seja, cada gene possui um conjunto de genes preditores, estas ligações formam uma rede ou também denominada na literatura de rede booleana BNs (*Boolean Networks*). Estes modelos são de simples representação, e apesar desta propriedade tem apresentado bons resultados. Em simulações de BNs como *Drosophila melanogaster* [51], ciclo celular da levedura [36], e entre outras pesquisas tem sido aplicados com êxito. Então, a partir destas BNs é possível gerar Redes de Genes Artificiais (AGN), aplicar e validar métodos de inferência de GRNs, dado que toda a estrutura da rede passa a ser conhecida.

Visto que as AGNs foram desenvolvidas e os dados dos perfis de expressão simulados podem ser gerados, é possível realizar o processo de inferência e validação das GRNs. E utilizar como base para a inferência de AGNs o método de reconhecimento de padrões como, por exemplo, o método seleção de características [31]. A seleção de característica é constituída de duas partes, um algoritmo de busca e a função critério. São exemplos de pesquisas sobre a inferência de GRNs utilizando seleção de característica [37, 28, 61, 20]. O algoritmo de busca e a função critério utilizada neste trabalho são respectivamente o algoritmo genético (AG) [30] com modelo de ilhas [47] e a função baseada na entropia de condicional média [40].

REVISÃO BIBLIOGRÁFICA

Entropia e Informação Mútua

A entropia da termodinâmica foi apresentada por Rudolf Clausius considerando apenas apresentações macroscópicas [12]. Em 1877, Ludiwing Boltzmann apresentou que a entropia de Clausius poderia ser representada em probabilidade ligada a configuração de sistema microscópico [8], tal entropia ficou conhecida como entropia de Boltzmann-Gibbs. A forma discreta da entropia de Boltzmann é apresentada a seguir [56]:

$$H_{BG}(X) = -K \sum_{i=1}^{W} p_i \log p_i, \tag{1}$$

sendo K a constante de Boltzmann e possuindo valor igual a 1 em áreas distinta da área da física [56], e as probabilidades de p_i são equivalentes as W configurações microscópicas possíveis, logo:

$$\sum_{i=1}^{W} p_i = 1. \tag{2}$$

A entropia, posteriormente, foi aplicada na Teoria da Informação pelo pesquisador Claude Shannon [52]. A entropia desenvolvida permite indicar a quantidade de informação contida em uma determinada fonte, bem como possibilita graduar a desordem de um conjunto de dados [7]. Dado uma variável aleatória X que pode assumir valores booleanos como, por exemplo, 0 e 1. A entropia de Shannon assim como a entropia de Boltzmann permite determinar em termos probabilísticos as possíveis ocorrências destas variáveis aleatórias ($P(x)$):

$$H(X) = -\sum_{x \in X} P(x) \log P(x), \tag{3}$$

tal que

$$\sum_{x \in X} P(x) = 1. \tag{4}$$

Ou seja, a entropia de Shannon apresenta como resultado a medida da incerteza dada uma determinada variável, então quanto maior o resultado da função, consequentemente maior será a incerteza de predizer tal variável. Fazendo uso de duas variáveis (X e Y) em conjunto, a entropia conjunta é definida:

$$H(X, Y) = -\sum_{x \in X, y \in Y} P(x, y) \log P(x, y), \tag{5}$$

no qual as varáveis aleatórias X e Y em conjunto é representado pela probabilidade de $P(x, y)$.

A entropia condicional é representada por $H(Y|x)$, tal entropia calcula a incerteza de uma variável aleatória Y dado o valor de uma instância da variável aleatória x conhecida. ou seja, quanto menor o resultado da entropia condicional maior serão as chances da variável Y predizer a variável x [35]. A entropia condicional é definida sequentemente:

$$H(Y|x) = -\sum_{y \in Y} P(y|x) \log P(y|x). \tag{6}$$

A entropia condicional média é definida pela média ponderadas das estropias condicionais de todas as instâncias $x \in X$ [32]. A entropia condicional média é definida como:

$$H(Y|X) = \sum_{x \in X} P(x) H(Y|x), \tag{7}$$

no qual $H(Y|x)$ representa a entropia condicional e $H(Y|X)$ representa um valor no qual quanto menor o valor, maior será a informação de Y pela observação de X.

Algoritmo Genético

Os algoritmos genéticos (AG) são algoritmos heurísticos utilizados para resolver problemas de busca e otimização. Tal algoritmo possui o objetivo de resolver problemas de buscas não-triviais, no qual algoritmos convencionais não seriam capazes de resolver em tempo acessível. Este algoritmo está classificado dentro da classe de algoritmos evolutivos, assim como os algoritmos de Programação Evolutiva [24] e Estratégia Evolutiva [48]. O AG foi apresentado por John Holland [30] e desenvolvido pelo pesquisador Goldberg [25].

Os AGs são inspirados nos princípios da Teoria da Evolução de Charles Robert Darwin [16]. Aplicando os princípios de seleção e sobrevivência dos indivíduos mais adaptados ao meio e consequentemente, maior probabilidade deste indivíduo gerar mais descendentes, bem como o conceito de hereditariedade e mutação genética. Ou seja, realizar a busca do melhor resultado (otimização) selecionando os indivíduos (soluções candidatas), com base em seus graus de adaptabilidade (qualidade dos resultados), sendo que os descendentes herdarão as características de seus progenitores que foram selecionados, que serão passíveis a mutações genéticas (aplicação de métodos computacionais inspiradas em operadores naturais).

O AG possui a propriedade de ser amplamente adaptável a problemas, visto que apenas é necessária uma forma de representação do problema e uma função de avaliação dos possíveis resultados. Devido a esta propriedade os AGs foram aplicados em diversos problemas de busca e otimização com sucesso [9, 17, 44, 18, 4, 50, 41].

Outra propriedade que o AG possui é um alto grau de paralelização, devido aos dados serem altamente independestes. O argumento apresentado pode ser explicado dado ao fato dos indivíduos de uma população ser avaliados de forma independente. E da mesma forma que na natureza existem diversas subpopulações evoluindo concorrentemente, igualmente pode acontecer em um AG [42].

Os AGs realizam operações de seleção natural atuando sobre os componentes (indivíduos e população). O indivíduo é uma estrutura de dados que armazena a codificação de um possível resultado . A população é composta por um conjunto de indivíduos, na qual é subordinada a aplicação de operadores baseados na seleção natural.

Os operadores realizados por um AG são: a criação de uma população inicial e em seguida são aplicados um conjunto de operadores que se repetirão N vezes até atender a um determinado critério de parada. A cada iteração deste conjunto de operadores é realizada uma geração.

A cada geração são aplicados aos indivíduos da população os operadores do calculo da função *fitness*, seleção, *crossover* e mutação.

Componentes de um algoritmo genético

Indivíduo

A estrutura de um individuo é formada por um cromossomo e seu fitness. O cromossomo armazena a codificação do fenótipo em uma string de tamanho L, a qual é representada neste trabalho por valores binários variando os valores 0 ou 1. Um cromossomo C também pode gerar um conjunto de fenótipo $C = \{F_0, F_1, F_2, ..., F_n\}$, tal que F representa um fenótipo do conjunto. O tamanho do conjunto de fenótipos pode variar de acordo com o problema, sendo a quantidade utilizada neste trabalho é um conjunto de 5 fenótipos. Nos quais representam possíveis genes preditores para um gene alvo. A variável fitness representa a qualidade deste conjunto, tal valor é gerado após a avaliação do operador *Evaluation*. Um AG é composto por P indivíduos denominado população, a qual será sujeita a aplicações dos operadores genéticos. Os quais, são apresentados a seguir.

Operadores

População Inicial

A população inicial representa o primeiro operador do algoritmo genético sendo executada apenas uma vez. A população inicial tem como objetivo criar os primeiros indivíduos que irão compor a população. A criação dos indivíduos pode ocorrer de forma aleatória, ou se basear em algum método específico de inicialização dos indivíduos, dado o conhecimento a priori de bons cromossomos [46].

Função *Fitness*

A função *fitness* ou função objetivo é o operador responsável pela avaliação dos indivíduos da população (resposta para o problema) a cada geração executada. A avaliação consiste em expressar a qualidade de um problema em forma de uma função matemática [42], passando como parâmetro o fenótipo do indivíduo e obtendo com retorno da função a qualidade do indivíduo [46]. A função é aplicada a todos os indivíduos da população, para que o AG possa manipular estes dados em outros operadores.

Tal operador de avaliação do *fitness* do indivíduo é fundamental para o AG, visto que o *fitness* é utilizado como medida de adaptabilidade do indivíduo, ou seja, aumentando ou diminuindo suas chances de reprodução e sobrevivência [46].

Para que o operador funcione de forma eficiente a função *fitness* deve ser a mais representativa possível. Pois, a partir de uma melhor representação é possível ser mais expressivo nos operadores do AG e, por consequente, chegar a um resultado de forma mais rápida e precisa. Entretanto, para alguns tipos de problemas não é possível calcular com exatidão o grau de aptidão de um indivíduo, como problemas de predição de genes para inferência de

Seleção

O operador de seleção é responsável pela seletividade dos indivíduos da população para a reprodução [46], utilizando um determinado método probabilístico baseando nos *fitness* dos indivíduos da população. A seleção é executada imediatamente após o operador da função critério. Este operador possui diversos métodos que podem ser implementados [26]. Independentemente do método utilizado, os indivíduos com melhor *fitness* terão mais chances de serem selecionados. Entretanto, é importante observar que no processo de seleção não sejam contemplados apenas os melhores indivíduos, mas também indivíduos com *fitness* variados. A seleção apenas dos indivíduos melhores adaptados pode gerar uma convergência prematura dos resultados [55]. Alguns dos métodos existentes são resumidamente descritos a seguir [21].

O método *Stochastic Universal Sampling* é o método de roleta de forma mais elaborada. Este método garante que um individuo que possui *fitness* de melhor qualidade seja selecionado proporcionalmente.

No método *Tournament Selection* são selecionados N indivíduos aleatoriamente formando subgrupos de tamanho N. O indivíduo que possuir melhor *fitness* dentro do subconjunto será selecionado.

O *Truncation Selection* é o método de seleção mais simples, tem o objetivo selecionar os uma quantidade de melhores indivíduos da população e a partir deste subconjunto gerar os indivíduos da próxima população.

O método de *Rank Selection* também é baseado em uma roleta como no método de seleção *Roulette Wheell Selection* e *Stochastic Universal Sampling*. A construção da roleta é organizada por faixas uniformes e posicionamento ordenado. Ou seja, todos os indivíduos da população possuem o mesmo tamanho das faixas e sendo ordenados do melhor para o pior *fitness*.

Embora os métodos apresentados a pouco tenham sua importância, neste trabalho foi adotado o método *Roulette Wheel Selection*, o qual é apresentado a seguir com informações mais detalhadas que os demais. Neste método os indivíduos estão contidos em uma faixa da roleta, sendo o tamanho da faixa relacionada proporcionalmente ao seu *fitness*. Essa seleção proporcional torna possível tanto a escolha de indivíduos melhores adaptados quanto daqueles não tão bem adaptados. Para cada indivíduo da população é calculada a seguinte equação [27]:

$$ps = \frac{f(x_i)}{\sum_{j=1}^{n} if(x_j)}, \tag{8}$$

na qual $f(x_i)$ representa o valor do *fitness* do indivíduo x_i que está sendo avaliado e n_i representa o valor total da população. Ou seja, indivíduos com *fitness* melhores terão uma porção maior da roleta, e consequentemente, maior probabilidade de serem selecionados, como na Figura 1 em que o indivíduo 1 terá maior probabilidade de ser selecionado, devido seu *fitness* ser de maior qualidade que dos demais.

Após a construção da roleta é gerado um número aleatório, então é verificado em qual faixa da roleta em que este número se encontra e a partir da verificação é selecionado o indivíduo pertencente a tal faixa da roleta. Na Figura 1, apesar do indivíduo 2 possuir um faixa menor que outros indivíduos o mesmo foi selecionado. Ou seja, este modelo possibilita a seleção dos indivíduos com diversas qualidade de *fitness*. Este método requer um nível de processamento elevado, em razão da montagem da roleta ser realizada diversas vezes. Devido, a necessidade de percorrer todos os indivíduos da população para realizar os cálculos da proporção de cada individuo da roleta.

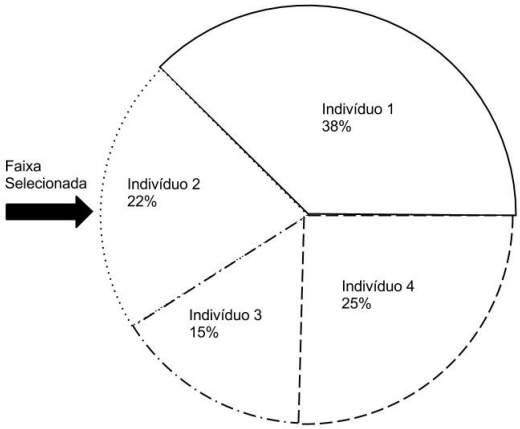

Figura 1: Modelo de seleção por roleta

Crossover

Depois de selecionado os indivíduos da população em casais (pares de indivíduos selecionados, denominados pais), tais pares irão passar pelo operador de *crossover* e gerar dois novos indivíduos, denominados filhos. O termo *crossover* é uma analogia utilizada para representar o cruzamento do cromossomo entre dois indivíduos, que representam biologicamente a troca de material genético [46]. Existem diversos métodos de *crossover*, nos quais a diferença basicamente está na variação da quantidade e nos locais dos pontos de cruzamento [22].

Existem diversas técnicas desenvolvidas para a realização do *crossover* como o *crossover* de um ponto, representado pelo exemplo 'a' na Figura 2. Primeiramente, tal técnica inicia pareando os cromossomos selecionados, então é escolhido uma posição que representará o ponto de quebra do cromossomo, as duas partes serão trocadas [29].

Outra técnica é o *crossover* de N-pontos apresentada na Figura 2 pelo exemplo 'b'. Tal técnica seleciona duas ou mais posições do cromossomo, gerando a quebra do cromossomo nestas posições. Então, os pedaços de cromossomos gerados serão trocados, ao final do processo dois novos cromossomos serão formados [22].

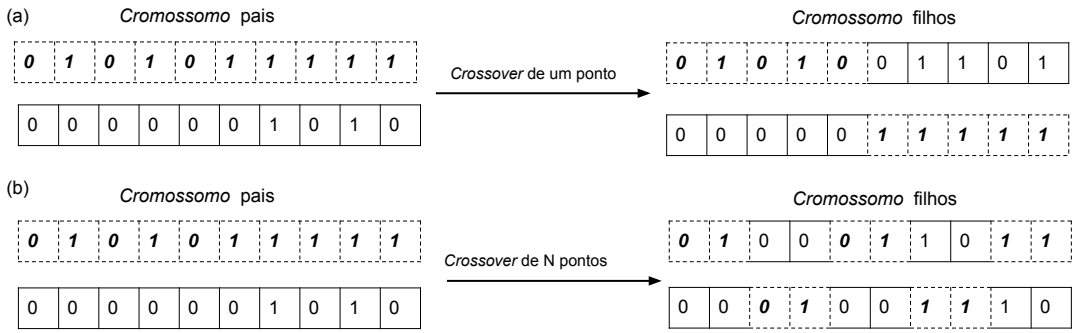

Figura 2: Modelo de crossover apresentado em (a) é o modelo de 1-ponto e o (b) é o modelo de N-pontos

Mutação

O operador de mutação é aplicado no cromossomo dos indivíduos filhos, tal operador muta uma determinada posição do cromossomo dada certa probabilidade [21]. Na Figura 3 para cada posição do cromossomo filho, é gerado um número aleatório que determinará se a posição do cromossomo vai ser mutado ou não, dado uma probabilidade definida. A Figura 3 apresenta o processo de mutação do cromossomo dada um cromossomo binário, mutando a primeira posição para seu valor inverso, de zero para um. O operador agrega uma boa característica ao processo do AG, a qual impede que uma população converja para um mínimo local [49].

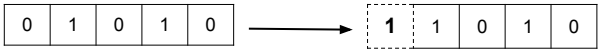

Figura 3: Mutação de um cromossomo binário com uma representação binária

Critério de Parada

O critério de parada é definido por meio de um operador condicional que pode representar o término das iterações da execução do AG, ou a execução de uma próxima geração, dado a uma determinada condição. O critério de parada também pode ser definido como o controlador do processo iterativo de um AG [46]. Como, por exemplo, caso encontre o melhor *fitness* caso encontre uma resposta com *fitness* aceitável ou executar G gerações pré estabelecidas, mesmo que tal busca não apresente uma resposta aceitável [29].

Elitismo

O operador de elitismo é muito utilizado nas implementações do AG [29]. Este operador permite que uma determinada quantidade de indivíduos de uma geração possa ser alocada na próxima geração [43]. Esta técnica é útil, devido ao conjunto dos melhores individuos *fitness* continuarem na próxima geração, logo a qualidade do *fitness* não irá decair. Entretanto, um cuidado é necessário em grandes quantidades de indivíduos elite, em razão da minimização da variedade genética da população, aumentando as chances da convergência prematura.

Modelo de Ilhas

Um dos problemas desafiadores que vem surgindo nas pesquisas atuais é o aumento expressivo da dimensionalidade e da complexidade dos problemas [1, 15, 10]. Algoritmos como os AGs necessitam de alto processamento computacional. Então, estratégias de paralelizarão foram propostas para aumentar o desempenho destes algoritmos de buscas e otimização.

O modelo de ilhas (MI) é um modelo de evolução de multipopulações (ilhas) que ocorre simultaneamente, a fim de melhorar o desempenho dos algoritmos evolutivos. A teoria base para a inspirarão do MI é pertencente a Teoria da Evolução do Equilíbrio Pontuado [13]. As primeiras pesquisas sobre os MIs foram desenvolvidas por Jettey e colaboradores aplicadas para os AGs [47]. Entretanto, os MIs também podem ser aplicados a outros algoritmos evolutivos como a evolução diferencial (ED) [27]. Assim como o AG, o MI é bastante flexível e consegue bordar diversos tipos de problemas [2, 6, 3, 57].

A possibilidade de evolução paralela entre as populações se deve a propriedade dos algoritmos evolutivos serem fracamente acoplados, possibilitando sua decomposição. Neste modelo as multipopulações evoluem independentemente e paralelamente, sendo que em determinadas gerações, as populações realizam trocas de indivíduos [13]. Ou seja, a evolução é independente das multipopulações e gera uma competição para encontrar o melhor indivíduo, entretanto existe uma cooperação entre as multipopulações, que ocorre por meio do processo de migração [27].

Redes Complexas

As redes complexas são uma extensão da teoria dos grafos proposto por Leonard Euler [14]. O primeiro modelo de redes complexas desenvolvidos foi proposto por Paul Erdös e Alfréd Rényi (ER) em 1959 [23]. Posteriormente outros modelos foram desenvolvidos com o objetivo de representar sistemas reais como, por exemplo, o modelo mundo pequeno mundo-pequeno ou *small-word* (WS) [59] e livre de escala ou *scale-free* (BA) [5].

Estas redes têm o objetivo de simular sistemas reais como, por exemplo, representar um sistema biológico [33]. Cada modelo de rede complexa apresenta distintas topologias e propriedades bem definidas, neste contexto as GRNs podem ser bem representadas.

Uma rede complexa é representada por um grafo que possui V_m vértices ligados por A_n arestas. Para gerar uma rede complexa dois parâmetros são definido o número de vértices (genes) e o grau médio $< k >$ de arestas.

A topologia ER desenvolvida pelos pesquisadores Erdös e Rényi [23] é formada por ligações entre vértices de forma aleatória e com distribuição uniforme. Esta topologia tenta não realizar a ligação entre mesmos vértices e não gerar muitas conexões em um único vértice.

A topologia desenvolvida pelos pesquisadores Watts e Strogatz (WS) [59] não é realizada totalmente aleatória. Esta topologia é baseada ao fenômeno mundo pequeno [45], o qual é baseado no conceito de que a média das distâncias de uma pessoa a qualquer outra é aproximadamente 6. Neste contexto, os vértices da topologia WS são ligados aos vértices mais próximos.

O modelo desenvolvido pelos pesquisadores Barabási e Albert (BA) [5] é formada por muitos vértices pouco conectados a outros poucos vértices muito conectados. A formação da topologia dividida em duas etapas: o crescimento e a preferência linear do crescimento.

MATERIAIS E MÉTODOS

Framework Watchmaker

O *Watchmaker* é um *framework* que permite a abstração do desenvolvimento do algoritmo genético e do modelo de ilhas sendo implementada na linguagem Java [21]. Também é apresentado diversos outros recursos como métodos de seleção como *Roulette Wheel Selection* , *Tournament Selection* e entre outros. Outros recursos disponíveis são os operadores de elilistimo, critério de parada (ver) e facilidade para inserir a função *fitness* do problema abordado. Documentação e *feedback* de qualidade são outros recursos importantes, facilitando o processo de aprendizagem e aplicação do *framework*.

Configuração dos Operadores Genéticos

- **Indivíduos**: para a representação do cromossomo foi adotado valores discretos e gerados 5 fenótipos de cada cromossomo, Nos quais representam os possíveis genes preditores.
- **População**: o tamanho da população para o AG é de 250 indivíduos e para cada ilha do MI é definida uma quantidade de 50 indivíduos por ilha, e ilhas de tamanho 2, 3 e 5.
- **População Inicial**: o método aleatório foi adotado para gerar os indivíduos da população inicial.
- **Função *Fitness***: a função utilizada para os testes do AG e MI é baseada na entropia condicional média, a qual foi desenvolvida por [38].
- **Seleção**: para realizar a seleção dos indivíduos foi utilizado o método *Roulette Wheel Selection*.
  - **Elitismo**: o método de elitismo também foi aplicado ao processo evolutivo da população. Sendo adotada uma taxa de 2% dos melhores indivíduos da população.
- ***Crossover***: o método de *crossover* aplicado é o *crossover* de um ponto.
- **Mutação**: o operador de mutação é definido como mutado de acordo com um alfabeto binário (0 e 1) com probabilidade de 5% de cada gene do cromossomo ser mutado.
- **Critério de Parada**: o critério de parada utilizada neste trabalho é baseado na quantidade de gerações executadas. A quantidade para o AG e MI foi definida 200 e 70 gerações para a busca dos preditores para cada gene.
- **Modelo de ilhas**: a abordagem de topologia utilizada neste trabalho é a topologia de migração em anel. Sendo definida que a taxa de migração de indivíduos de uma população para outra será de 2% a cada 20 gerações. E utilizado o algoritmo evolutivo nas ilhas o próprio AG.

RESULTADOS E DISCUSSÕES

O hardware utilizado para executar os experimentos possui um processador Intel(R) Core(TM) i7-3820 CPU @ 3.60GHz, memória cache 1024 KB, memória principal com 8 Gb e sistema operacional Linux Kubuntu.

Algoritmo de busca: AG e MI

Nesta seção são apresentados os resultados obtidos a partir da aplicação do algoritmo genético e do algoritmo genético com modelo de ilhas, tendo o objetivo inferir redes de regulação gênica. Para a inferência de redes foram utilizados as AGNs e a função critério descritos em [39].

Para todos os experimentos foram utilizados redes contendo 100 genes, o tamanho do sinal, ou instâncias de tempo observado, igual a 100. O grau médio das ligações entre os genes $< k >$ igual a 2. As topologias aplicadas foram Barabási-Albert (BA), Erdös-Rényi (ER), Watts-Strogatz (WS). No modelo de ilhas, a quantidade de ilhas adotadas foram 2, 3 e 5 ilhas, para cada experimento de inferência de GRNs foram geradas diferentes AGNs.

Para medir o grau de similaridade entre as redes inferidas pelos modelos e pelos AGNs, foi adotado o PPV e para obter o desempenho dos algoritmos foram coletados os tempos de execução de todos os experimentos. Para o cálculo do valor do PPV e do tempo foram executados 10 simulações.

O resultado do primeiro experimento executado para comparar as duas estratégias de busca: AG e AG com MI é apresentada pela Figura 4. Tal Figura apresenta os resultados da variação da quantidade de ilhas e a execução do AG, pela média do PPV de todos os experimentos, considerando as variações das topologias BA, ER e WS. É possível observar que utilizando o algoritmo com MI com 2 e 3 ilhas, a média melhores respostas comparado ao AG. Entretanto, ao executar o MI com 5 ilhas, a médoa do PPV da rede inferida mostra menor valor.

As Figuras 5, 6 e 7 apresentam os respectivos resultados das médias do PPV das topologias BA, ER e WS, pelas variações das quantidades de ilhas. A topologia BA apresentou os melhores resultados utilizando as estratégias de AG e MI, quando comparado com as demais topologias. Entretanto, um comportamento decrescente de forma aproximadamente linear do PPV pode ser observado, quando são adicionados mais ilhas no MI. Na topologia ER o modelo de ilhas apresentou melhor PPV em comparação ao AG, podendo ser observado uma acentuada melhora do PVV no MI utilizando 3 ilhas. Os resultados da topologia WS tiveram os menores valores de PPV utilizando AG e MI, também é observado que o MI com 5 ilhas obteve uma ligeira melhoria do PPV.

De maneira geral, as variações dos resultados se mostraram diferentes para cada topologia abordada. Não foi possível identificar os elementos que causaram tais variações. Neste sentido, é necessário a execução de um número maior de experimentos, a fim de identificar as potenciais causas das variações que foram detectadas neste trabalho. Inclusive mais parâmetros podem ser considerados como a variação do número de ligações, tamanho do sinal e quantidade de genes.

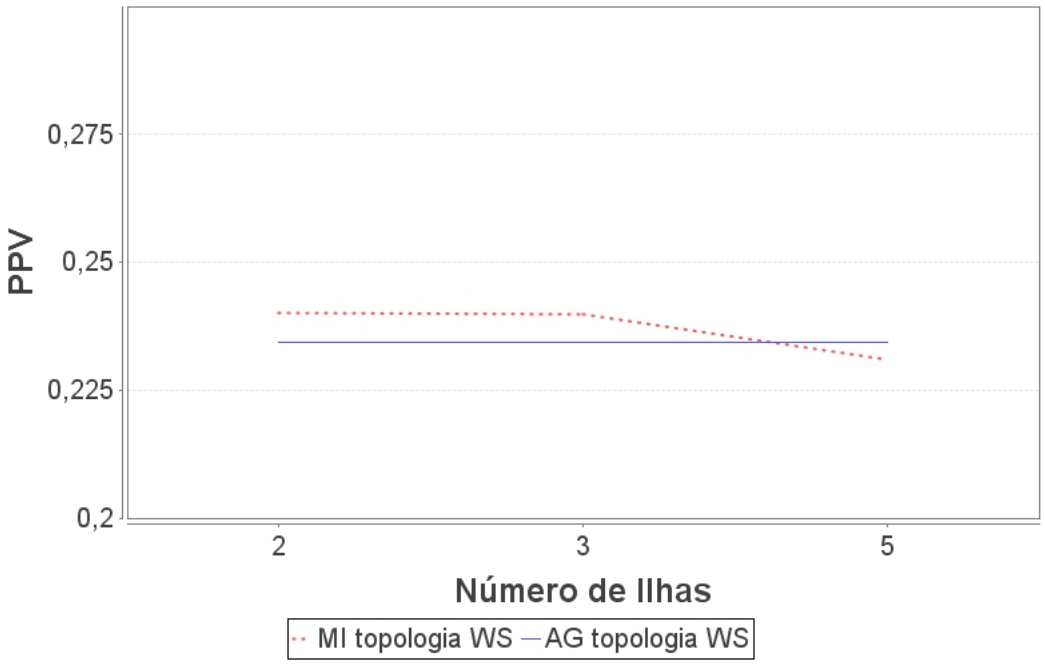

Figura 4: Medida de PPV obtida pela inferência de redes utilizando a estratégias de AG e AG com MI, aplicando 2,3 e 5 ilhas para o MI. Os valores da média do PPV obtido representa a execução de 10 experimentos.

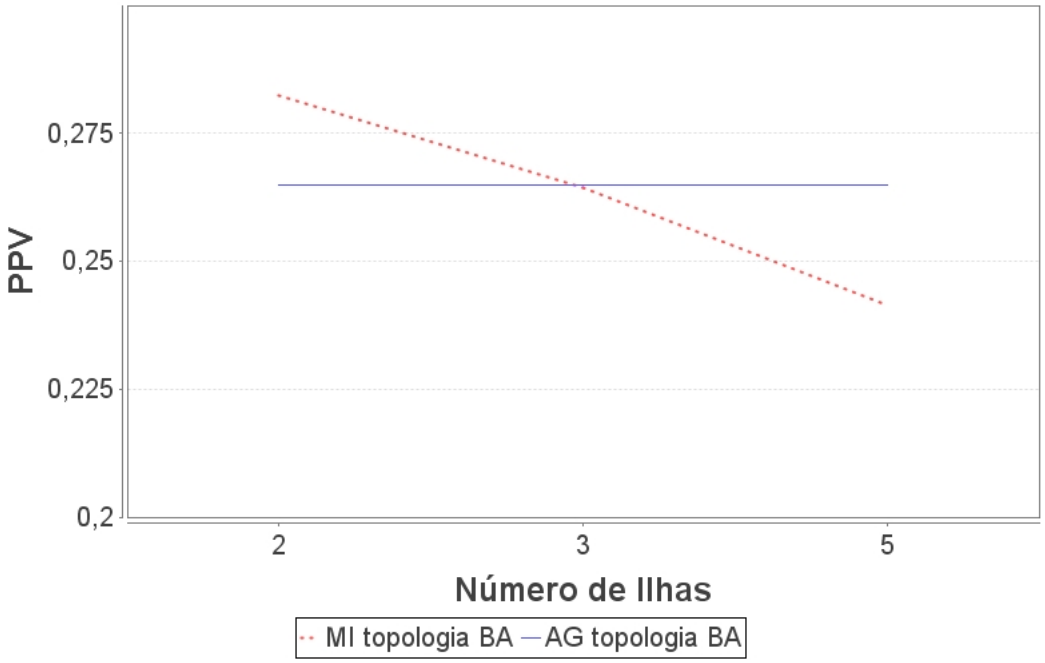

Figura 5: Média do PPV das redes inferidas pela topoloiga BA

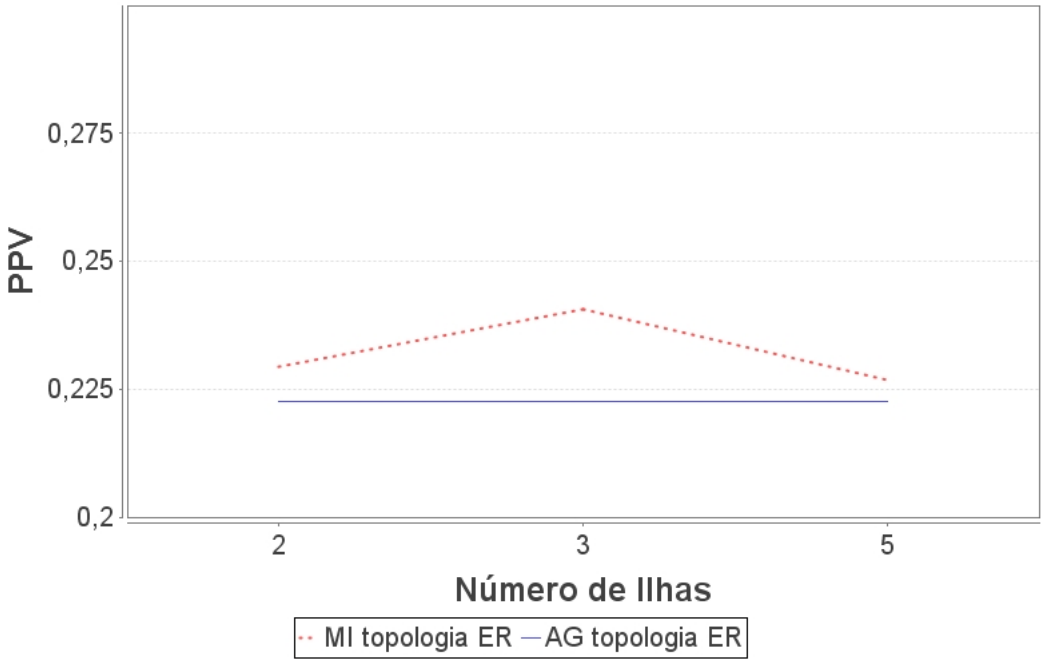

Figura 6: Média do PPV das redes inferidas pela topoloiga ER

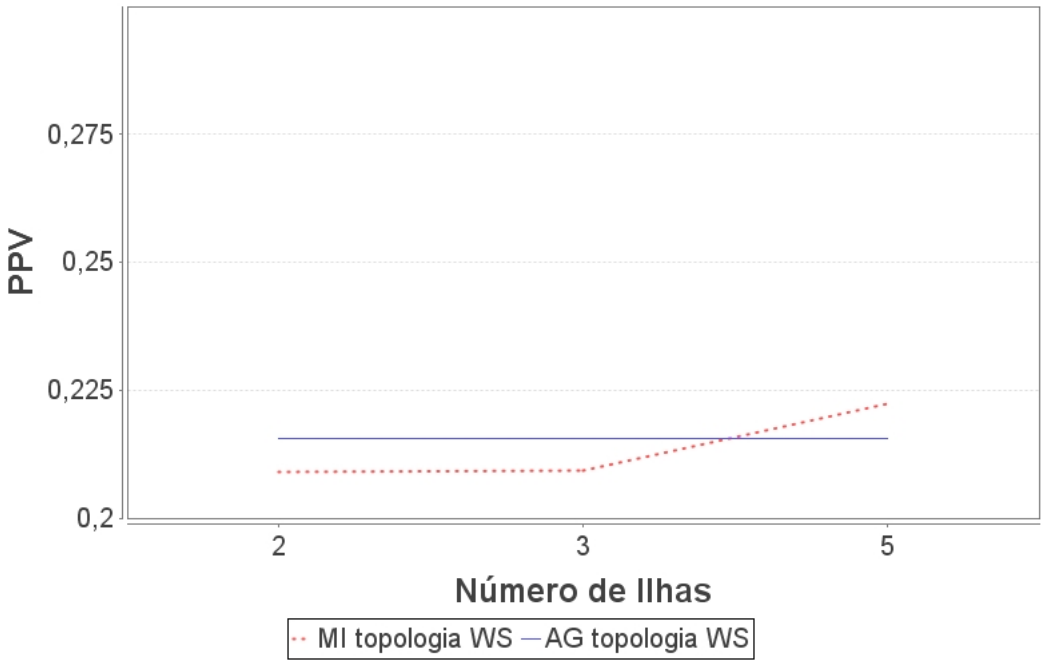

Figura 7: Média do PPV das redes inferidas pela topoloiga WS

No segundo experimento realizado, o objetivo foi avaliar o tempo computacional entre as estratégias de busca. Apresentando a mesma estratégia utilizada para o PPV, a Figura 8 apresenta o gráfico do tempo dado a variação das ilhas pelo AG com MI e o AG. É possível observar que todas as variações de ilhas (2, 3 e 5) do MI apresentaram tempo de execução superior ao AG. Este comportamento pode ser explicado pelo processo de sincronização do MI. Ou seja, para a execução do operador de migração, todas as ilhas devem estar sincronizadas e após este processo os indivíduos serão migrados. Este processo gera um *overhead* em razão do tempo de espera da sincronização, o que é agravado pela quantidade de genes a serem inferidos, consequentemente, esta estratégia consome muito tempo, comprometendo o desempenho do algoritmo.

Nas Figuras 9, 10 e 11 são apresentados as médias dos tempos computacionais gerados pela inferência das topologias BA, WS e ER respectivamente. É possível observar o grau semelhança entre os tempos computacionais gerados nas topologia pelos dois métodos de busca (AG e MI), o que sugere que a variação de topologia não é tão relevante no desempenho computacional para a inferência das redes .

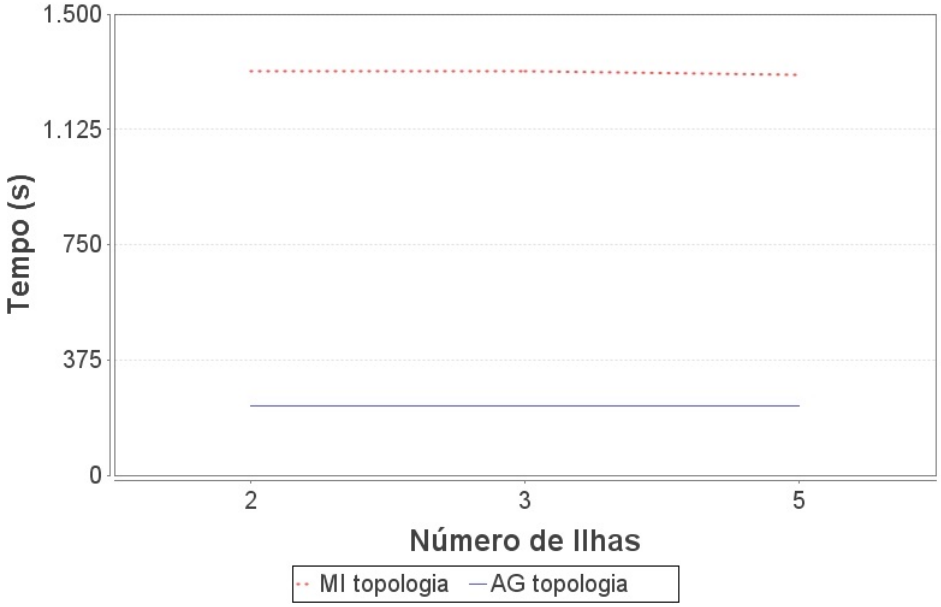

Figura 8: Média do tempo de todas as topologias dos algoritmo genético e do modelo de ilhas

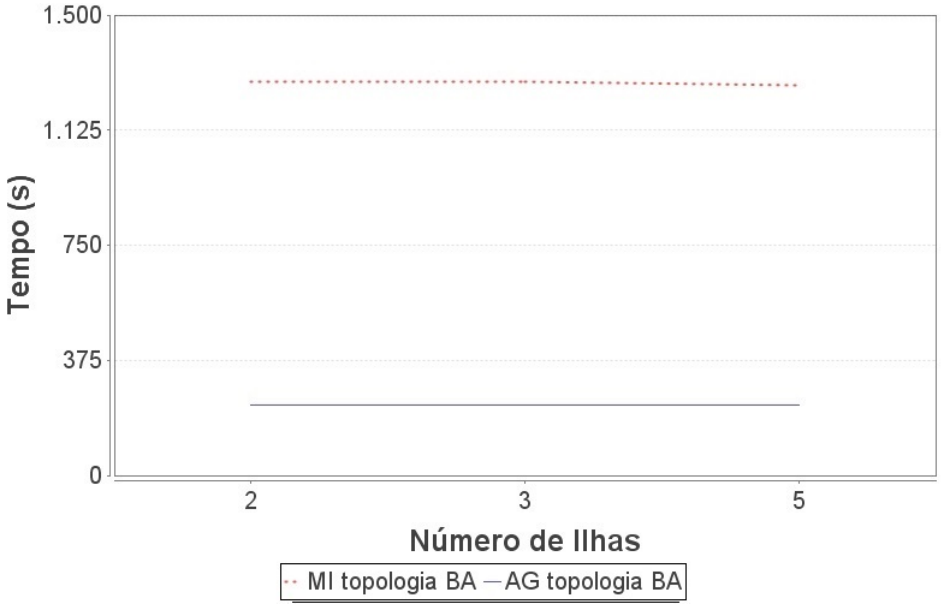

Figura 9: Média do tempo computacional gerado pela inferência da topologia BA pelos métodos de algoritmo genético e do modelo de ilhas

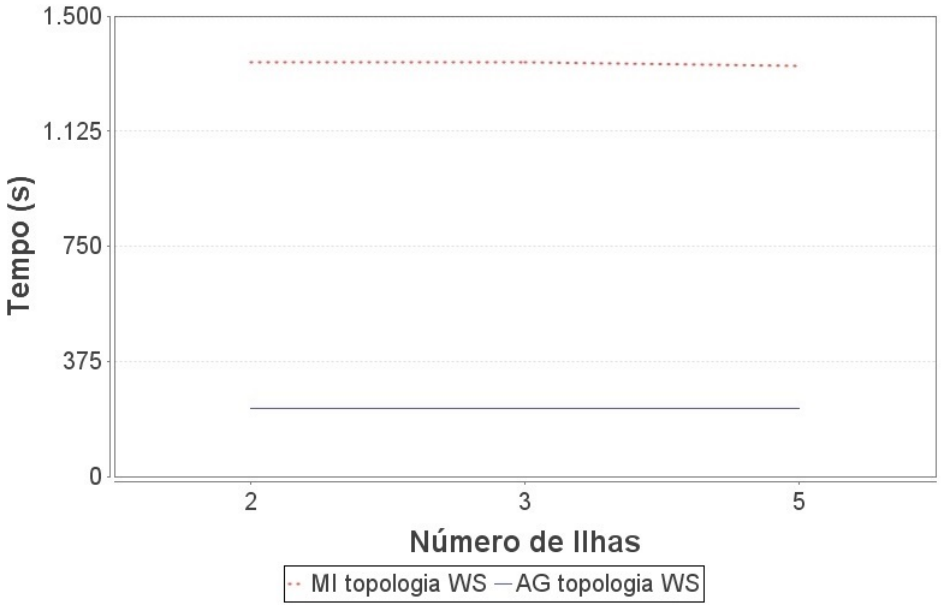

Figura 10: Média do tempo computacional gerado pela inferência da topologia WS pelos métodos de algoritmo genético e do modelo de ilhas

Conclusões

A partir de determinadas mudanças do ambiente, um organismo biológico pode responder de uma determinada forma, ajustando a expressão de seus genes. A conexão entre os genes formam uma grande rede complexa, na qual um gene pode regular muitos outros genes. Muitos estudos vêm sendo aplicados a tal problema, com o objetivo de compreender muitos problemas de áreas da biologia, física, psicologia, e pesquisas para desenvolvimento de remédios. Entretanto, ainda existe muito a ser descoberto sobre este processo biológico. Este trabalho aborda a inferência de GRNs a partir de um método de seleção de características, o qual é constituído por duas partes, uma função critério e um algoritmo de busca. A função critério abordada neste trabalho foi desenvolvida por [38], no qual é baseada na Entropia de Condicional Média [40]. O algoritmo de

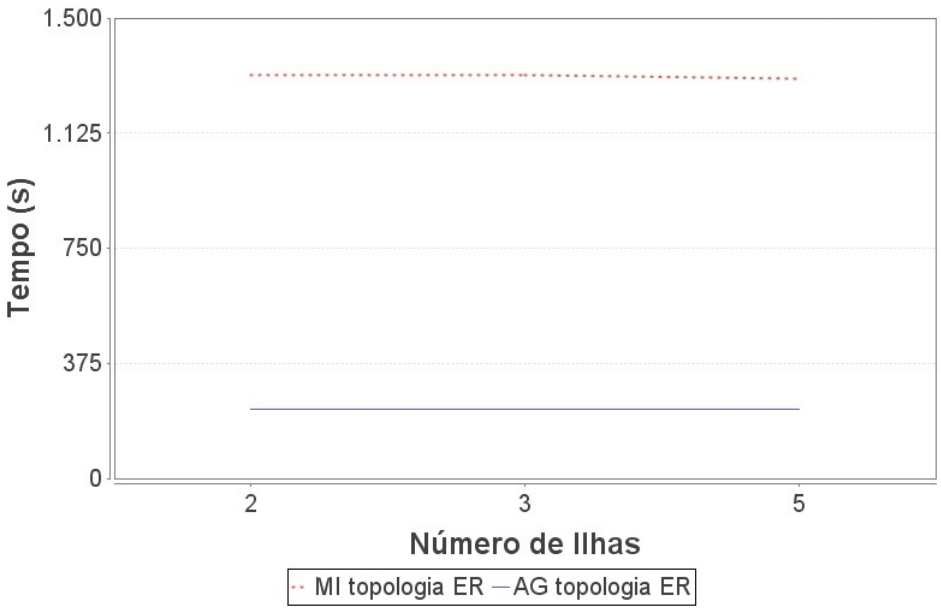

Figura 11: Média do tempo computacional gerado pela inferência da topologia ER pelos métodos de algoritmo genético e do modelo de ilhas

busca abordado neste trabalho é baseado no princípio da teoria evolucionista de Charles Robert Darwin [16], denominado algoritmo genético, e um algoritmo paralelo denominado modelo de ilhas, baseado na Teoria do Equilíbrio Pontuado [13].

Os resultados mostraram que o MI apresentam melhores valores médios do PPV com 2 e 3 ilhas e menor valor médio de PPV para 5 ilhas. O AG apresentou um melhor desempenho computacional comparado ao MI, uma possível resposta é a sincronização necessária no MI para realizar o operador de migração, gerando um *overhead* prejudicial para seu desempenho. Não foi possível identificar os elementos que causaram as variações do resultado do PPV nas diferentes topologias adotadas neste trabalho.

Referências Bibliográficas

[1] E. Alba. A Survey of Parallel Distributed Genetic Algorithms. *Complexity*, 4:31—-52, 1999.

[2] T. Bäck. *Evolutionary algorithms in theory and practice*. T. Bäck, 1994.

[3] T. Back, D. B. Fogel, and Z. Michalewicz. *Handbook of evolutionary computation*. IOP Publishing Ltd., 1997.

[4] T. Bäck and H.-P. Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, 1(1):1–23, 1993.

[5] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[6] A. D. Bethke. Comparison of genetic algorithms and gradient-based optimizers on parallel processors: Efficiency of use of processing capacity. 1976.

[7] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.

[8] L. Boltzmann, B. McGuinness, and P. Foulkes. *Theoretical physics and philosophical problems: Selected writings*, volume 5. Reidel Publishing Company, 1974.

[9] C. G. Braga. O uso de Algoritmos Genéticos para aplicação de Otimização de Sistemas Mecânicos, 1998.

[10] E. Cantú-Paz. A survey of parallel genetic algorithms. *Calculateurs paralleles, reseaux et systems repartis*, 10(2):141–171, 1998.

[11] N. Chandra and J. Padiadpu. Network approaches to drug discovery. *Expert opinion on drug discovery*, 8(1):7–20, Jan. 2013.

[12] R. Clausius. *The mechanical theory of heat*. Macmillan, 1879.

[13] J. P. Cohoon, S. U. Hegde, W. N. Martin, and D. Richards. Punctuated equilibria: a parallel genetic algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 148–154. L. Erlbaum Associates Inc., 1987.

[14] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, 2007.

[15] P. A. da Costa Filho and R. J. Poppi. Algoritmo genético em química. 22:405, 1999.

[16] C. Darwin and W. F. Bynum. *The origin of species by means of natural selection: or, the preservation of favored races in the struggle for life*. AL Burt, 2009.

[17] L. Davis. Handbook of genetic algorithms. 1991.

[18] K. A. De Jong. Analysis of the behavior of a class of genetic adaptive systems. 1975.

[19] P. D'haeseleer, S. Liang, and R. Somogyi. Gene expression data analysis and modeling. In *Pacific symposium on biocomputing*, volume 99, 1999.

[20] J. Dougherty, I. Tabus, and J. Astola. Inference of gene regulatory networks based on a universal minimum description length. *EURASIP Journal on Bioinformatics and Systems Biology*, 2008:5, 2008.

[21] D. W. Dyer. Watchmaker framework, 2010.

[22] J. Eiben, A.E. and Smith. *Introduction to Evolutionary Computing*. Springer Berlin Heidelberg, 2003.

[23] P. Erdös and R. Alfréd. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[24] D. Fogel. *Artificial intelligence through simulated evolution*. Wiley-IEEE Press, 2009.

[25] D. E. Goldberg. Genetic algorithms in search, optimization, and machine learning. 1989.

[26] D. E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. *Urbana*, 51:61801–62996, 1991.

[27] F. G. Guimar and F. Campelo. *Topologias Dinâmicas para Modelo em Ilhas usando Evolução Diferencial*. PhD thesis, Universidade Federal de Minas Gerais, 2011.

[28] R. F. Hashimoto, S. Kim, I. Shmulevich, W. Zhang, M. L. Bittner, and E. R. Dougherty. Growing genetic regulatory networks from seed genes. *Bioinformatics*, 20(8):1241–1247, 2004.

[29] C. Holland. Algoritmos Genéticos Adaptativos : Um estudo comparativo. 2000.

[30] J. H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.

[31] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):4–37, 2000.

[32] D. C. M. Junior. Seleçao de caracterısticas e prediçao intrinsecamente multivariada em identificaçao de redes de regulaçao gênica, 2008.

[33] S. Kauffman. Gene regulation networks: A theory for their global structure and behaviors. *Current topics in developmental biology*, 6:145–182, 1971.

[34] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

[35] A. Kelemen, A. Abraham, and Y. Chen. *Computational intelligence in bioinformatics*, volume 94. Springer, 2008.

[36] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang. The yeast cell-cycle network is robustly designed. *Proceedings of the National Academy of Sciences of the United States of America*, 101(14):4781–4786, 2004.

[37] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. In *Pacific symposium on biocomputing, architectures.pdf:pdf*, volume 3, page 2, 1998.

[38] F. M. Lopes. *Redes complexas de expressão gênica: síntese, identificação, análise e aplicações*. PhD thesis, Univesidade de São Paulo, 2011.

[39] F. M. Lopes, R. M. Cesar Jr, and L. D. F. Costa. Gene Expression Complex Networks: Synthesis, Identification, and Analysis. *Journal of Computational Biology*, 18(10):1353–1367, 2011.

[40] F. M. Lopes, D. C. Martins, and R. M. Cesar. Feature selection environment for genomic applications. *BMC bioinformatics*, 9(1):451, 2008.

[41] H. S. Lopes. Algoritmos genéticos em projetos de engenharia: aplicações e perspectivas futuras. *Anais do IV Simpósio Brasileiro de Automaçao Inteligente*, pages 64–74, 1999.

[42] D. C. Lucas. Algoritmos Genéticos: uma Introduçao1. 2002.

[43] J. Majumdar and A. K. Bhunia. Elitist genetic algorithm for assignment problem with imprecise goal. 177:684–692, 2007.

[44] Z. Michalewicz. *Genetic algorithms+ data structures= evolution programs*. springer, 1996.

[45] S. Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[46] M. A. C. Pacheco. Algoritmos genéticos: princípios e aplicações. *ICA: Laboratório de Inteligência Computacional Aplicada. Departamento de Engenharia Elétrica. Pontifícia Universidade Católica do Rio de Janeiro. Fonte desconhecida*, 1999.

[47] C. B. Pettey, M. R. Leuze, and J. J. Grefenstette. A parallel genetic algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 155–161. L. Erlbaum Associates Inc., 1987.

[48] I. Rechenberg. *Evolutionsstrategien*. Springer, 1978.

[49] G. Rudolph. Convergence analysis of canonical genetic algorithms. *Neural Networks, IEEE Transactions on*, 5(1):96–101, 1994.

[50] D. S. Sanches, O. Morandin, B. D. Muniz, and E. R. R. Kato. Modeling Strategy by Adaptive Genetic Algorithm for Production Reactive Scheduling with Simultaneous Use of Machines and AGVs. In *Computational Intelligence for Modelling Control & Automation, 2008 International Conference on*, pages 249–254. IEEE, 2008.

[51] Sanchez and D. Thieffry. A Logical Analysis of the< i> Drosophila</i> Gap-gene System. *Journal of theoretical Biology*, 211(2):115–141, 2001.

[52] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[53] M. Srinivas and L. M. Patnaik. Adaptive probabilities of crossover and mutation in genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(4):656–667, 1994.

[54] G. Stolovitzky, D. O. N. Monroe, and A. Califano. Dialogue on Reverse Engineering Assessment and Methods. *Annals of the New York Academy of Sciences*, 1115(1):1–22, 2007.

[55] D. Thierens and D. Goldberg. Convergence models of genetic algorithm selection schemes. In *Parallel problem solving from nature—PPSN III*, pages 119–129. Springer, 1994.

[56] C. Tsallis. Nonadditive entropy: the concept and its use. *The European Physical Journal A*, 40(3):257–266, 2009.

[57] R. K. Ursem. Diversity-guided evolutionary algorithms. pages 462–471, 2002.

[58] Z. Wang, M. Gerstein, and M. Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

[59] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442, 1998.

[60] D. WhitlePacheco1999y. A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85, 1994.

[61] X. Zhou, X. Wang, R. Pal, I. Ivanov, M. Bittner, and E. R. Dougherty. A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks. *Bioinformatics*, 20(17):2918–2927, 2004.