

TRANSGENÉTICA COMPUTACIONAL: UMA APLICAÇÃO AO PROBLEMA QUADRÁTICO DE ALOCAÇÃO

Marco César Goldbarg *

Elizabeth Ferreira Gouvêa Goldbarg

Departamento de Informática e Matemática Aplicada

Universidade Federal do Rio Grande do Norte

Natal – RN

E-mails: goldb@ufrnet.br

bgouvea@ufrnet.br

** Corresponding author/autor para quem as correspondências devem ser encaminhadas*

Recebido em 08/2000, aceito em 10/2002 após 3 revisões

Resumo

O presente trabalho apresenta uma nova abordagem, denominada Transgenética Computacional. A metáfora baseia-se na utilização de informações meméticas e no emprego dos fluxos extra e intracelulares para planejar e executar manipulações genéticas no contexto dos algoritmos evolucionários. A pesquisa desenvolve duas linhas de algoritmos: a primeira utilizando-se de ambos os fluxos para informar o processo de busca evolucionária; a segunda utiliza-se exclusivamente da manipulação intracelular. São apresentados os agentes da Transgenética Computacional. São examinadas propriedades resultantes da interação cromossomo \times agente de manipulação que se mostram semelhantes às do processo imunológico natural. Ao final são relatados resultados computacionais para o Problema Quadrático de Alocação.

Palavras-chave: algoritmos transgenéticos, computação evolucionária, problema quadrático de alocação.

Abstract

This paper introduces the Computational Transgenetic approach. The metaphor is based on the use of memetic pieces of information and on the extra and intracellular flows to design and accomplish genetic manipulation in the chromosomes of a given population of an evolutionary algorithm. The research develops two algorithmic approaches. The first algorithmic approach uses both the intra and extra-cellular flows to guide an evolutionary search process. The second one uses, uniquely, intracellular manipulation. Computational Transgenetic agents are presented. Properties resulting from chromosome \times agent interactions, similar to the natural immunologic process, are examined. Finally, the paper reports the results of computational experiments of applying both techniques to the Quadratic Assignment Problem.

Keywords: transgenetic algorithms, evolutionary computation, quadratic assignment problem.

1. Introdução

A Computação Evolucionária clássica baseia-se fortemente na utilização do paradigma neodarwiniano da evolução gradual e estocástica. Um dos princípios dessa abordagem pode ser resumido pela frase: *A natureza não dá saltos*. Isso é, contudo, contestado por diversos autores (Marguli & Sagan, 1991; Bourguignon, 1989; Gould, 1993) que apresentam teorias complementares justificando outros mecanismos evolucionários. A Transgenética Computacional – TC – se baseia no emprego dessas abordagens mais recentes para construir seus algoritmos computacionais. A proposta amplia o escopo da computação evolucionária clássica pela inclusão de dois novos contextos: o epigenético e o intracelular, e pela utilização do paradigma procariótico de recombinação que é bastante propício à manipulação direta do código ADN e à consideração de informações ambientais – dentro do processo de evolução.

A abordagem proposta é aplicada a um clássico problema NP-árduo da Otimização Combinatória, o Problema Quadrático de Alocação. Dadas duas matrizes quadradas de coeficientes reais $M_F = [f_{ij}]$ e $M_D = [d_{ij}]$, $1 \leq i, j \leq n$, o Problema Quadrático de Alocação consiste em encontrar:

$$\min_{\rho \in S_n} \sum_{i,j} f_{\rho(i)\rho(j)} d_{i,j}$$

determinando a permutação correspondente ρ no conjunto S_n de todas as permutações de $\{1, 2, \dots, n\}$.

Uma das maiores aplicações do PQA é na teoria de localização, onde se deseja instalar facilidades em locais previamente determinados. Considerando-se $f_{\rho(i)\rho(j)}$ o fluxo de material que vai da facilidade $\rho(i)$ para a facilidade $\rho(j)$ e d_{ij} a distância da localidade i à localidade j , o objetivo é determinar o custo mínimo de atribuir, simultaneamente, as facilidades $\rho(i)$ e $\rho(j)$ a pares de locais i e j , respectivamente. Cada custo resulta da soma dos produtos $f_{\rho(i)\rho(j)} d_{ij}$, correspondente à alocação definida pela permutação ρ de n elementos. Çela (1998) apresenta um estudo detalhado desse problema.

Os fundamentos da abordagem proposta são apresentados na seção 2. A seção 3 apresenta a Transgenética Computacional. A seção 4 descreve as duas classes de algoritmos transgenéticos. A seção 5 apresenta uma aplicação da técnica ao Problema Quadrático de Alocação e reporta os resultados computacionais. Finalmente, na seção 6 são apresentadas as conclusões do artigo.

2. Fundamentos da Abordagem

A Transgenética Computacional, TC, é uma metaheurística que utiliza a infiltração planejada de informações como uma ferramenta do aperfeiçoamento das informações genéticas. Ela fundamenta sua ação em paradigmas do fluxo intracelular e através de agentes de informação que manipulam o contexto genético. As informações infiltradas através dos agentes são criadas e controladas com base no paradigma epigenético. As informações em formato não genético que são utilizadas para manipular o processo evolucionário são denominadas *memes*. Face à sua proposta de manipular e guiar o conteúdo genético, a TC utiliza-se de mecanismos mais simples de recombinação genética que imitam o paradigma procariótico de adaptação, isto é, a transferência de genes entre organismos procarióticos realizada através dos processos de transdução, transformação e conjugação (Shapiro, 1988). Os algoritmos da TC estão divididos

em duas classes principais. A primeira é a classe dos Algoritmos Transgenéticos Extra-Intracelulares, *ATEIs*, que baseiam o seu processo evolucionário nos paradigmas extracelular, intracelular e epigenético. A segunda classe é a dos Algoritmos Proto-Gene, *ProtoG*, que desenvolvem seu processo evolucionário baseando-se apenas nos paradigmas intracelular e epigenético. Como consequência disso, o algoritmo *ProtoG* não realiza a chamada reprodução sexual como base para o compartilhamento de informações entre seus indivíduos.

O fluxo extracelular caracteriza-se pela troca de informações no contexto extracelular, ou seja, entre células. Os Algoritmos Genéticos, apresentados por Holland (1975), são técnicas clássicas da Computação Evolucionária para resolver Problemas de Otimização Combinatória. De modo geral os Algoritmos Genéticos (*AGs*) utilizam os chamados mecanismos de seleção, cruzamento e mutação para operacionalizar um processo de busca estocástica em um espaço de soluções. Esses mecanismos procuram copiar o processo natural de seleção e reprodução dos cromossomos na formação de novos indivíduos, codificando as soluções de um problema em um formato “cromossômico” e submetendo-as a um processo pseudoevolutivo que unifica estratégias de seleção e reprodução em um só estágio. Nessa estratégia a informação genética dos pais é combinada e repassada aos filhos, que são constituídos como novos indivíduos. Tal via de replicação e alteração genética é conhecida como extracelular (Kargupta, 1998). O processo pseudoevolutivo assim modelado é simples e não implica em grandes obstáculos para a implementação computacional. Todavia, por suas simplificações, replica apenas uma pequena parte dos mecanismos que atuam na seleção das espécies (Lumsden & Wilson, 1981; Brooks & Wiley, 1986).

Por sua vez, o processo de transformação do ADN pode estar sujeito a pressões evolucionárias importantes que não se expressam pela via extracelular. Os seres procarióticos, por exemplo, utilizam uma forma não sexual para compartilhar informações genéticas que podem ser originadas de fissão binária, conjugação ou transdução (Campbell, 1996). A via imunológica e a transferência horizontal também representam outros caminhos para a alteração do ADN celular (Behe, 1996). As interações dessa via de transformação do ADN são ditas intracelulares.

As pressões adaptativas e evolucionárias, no entanto, não são exercidas somente através de contextos físicos. Os seres vivos estão sujeitos a pressões que emergem de interações mais sutis, relacionadas com os indivíduos da própria espécie, e esses, por sua vez, estão imersos em contextos ainda mais complexos – as sociedades – que possuem pressões evolutivas próprias. Alguns aspectos das pressões evolutivas que ocorrem fora do contexto genético são abordados em uma disciplina denominada *epigenética*. A cultura, como a biologia, exhibe padrões de evolução – exploração adaptativa e transformação de um espaço de informações através de variação, seleção e transmissão (Boyd & Richerson, 1985). No entanto a evolução cultural exhibe padrões e regras de desenvolvimento diferentes da biológica. Os memes são unidades de informação que podem difundir-se por uma cultura, como os genes disseminam-se por um reservatório gênico (Wright, 2000). O termo “meme” foi sugerido por Dawkins (1976). Na definição de Dawkins um meme pode ser entendido como uma idéia.

A ligação entre genes e memes ocorre uma vez que os genes prescrevem as regras epigenéticas, que são regularidades da percepção sensorial e do desenvolvimento mental que animam e canalizam a aquisição da cultura. A cultura ajuda a determinar quais dos genes prescritos sobrevivem e se multiplicam de uma geração à próxima. Novos genes bem-sucedidos alteram as regras epigenéticas da população. As regras epigenéticas alteradas mudam a direção e a eficácia dos canais de aquisição cultural realimentando o processo de co-evolução gene \times meme (Wilson, 1998).

A dificuldade de considerar a epigenética no contexto da Computação Evolucionária é discutida por Chattoe (1998). Na Computação Evolucionária os primeiros trabalhos dentro desse tema simplesmente associaram memes às informações obtidas fora do fluxo extracelular. Nos Algoritmos Meméticos, conforme definidos por Merz & Freisleben (1999) e Radcliffe & Surry (1994), o estágio epigenético é expresso, principalmente, através de um aperfeiçoamento dos cromossomos via procedimentos de busca local. Tais algoritmos também são denominados de Algoritmos Genéticos Híbridos (Houck *et al.*, 1996). O atual trabalho propõe que os memes sejam definidos em conformidade com Plotkin (1996) que considera memes como informações codificadas e transmitidas através de meios não genéticos. Dessa forma, define-se um meme computacional como uma informação que possa ser utilizada para rearranjar um cromossomo ou um bloco de genes. A memética na TC será utilizada no sentido de criar um processo contínuo de obtenção e infiltração de informações para o interior do contexto genético. Além disso, os memes poderão estar sujeitos a processos de competição e seleção pertinentes ao meio cultural.

A informação memética no contexto computacional pode se originar de várias fontes tais como resultados teóricos conhecidos para um problema ou uma instância, heurísticas, análise de cenário, etc, constituindo um banco de memes ou uma base de memes a ser utilizada pelo algoritmo. O processo de infiltração memética no contexto genético é operacionalizado na TC através de agentes computacionais análogos a agentes do fluxo intracelular. As técnicas de manipulação intencional se baseiam nas utilizadas pela Engenharia Genética. A Engenharia Genética, EG, segundo Dudley (1990), é uma parte da ciência que estuda os mecanismos intracelulares de interação genética, bem como seu impacto sobre os organismos. Para facilitar o processo de manipulação a EG tem empregado agentes já existentes na natureza que se especializaram no uso do fluxo intracelular. Tais agentes são chamados de vetores, dentre os quais dois dos mais utilizados são os vírus e as partículas genéticas móveis.

A construção de uma metáfora computacional que imite de perto o paradigma biológico é uma pretensão extremamente ambiciosa e, na atualidade, supostamente fora do estado da arte. Apesar disso a TC caminha no sentido da proposta de Moreno (1986) que afirma que a única via para descrever um sistema auto-reprodutor e auto-organizado é unir duas formas de modelagem: a físico-experimental, aqui representada pelas ferramentas, regras e agentes do fluxo intra e extracelular, e a lógico-formal, aqui representada pelo contexto epigenético. Aceitando a premissa de Davies (1998) sobre o fato dos seres vivos poderem ser considerados sistemas de processamento de informações, a proposta da TC aprofunda seu contato com a realidade do mundo natural, porque entende que uma abordagem que se proponha a imitar uma metáfora deve evitar principalmente ser simplista. A metáfora está apoiada no uso de ferramentas de manipulação de cromossomos de modo intencional e planejado e na expressão dos memes através dos agentes do fluxo intracelular.

3. Transgenética Computacional

A Transgenética Computacional é uma proposta multi-agente baseada em três idéias centrais:

- ✓ Infiltrar, de forma planejada, informações no processo evolucionário com o objetivo de informá-lo e guiá-lo.
- ✓ Planejar o processo de infiltração de informações e de construção de agentes através do uso do paradigma epigenético.
- ✓ Utilizar a recombinação procariótica como ferramenta complementar de preservação da informação genética.

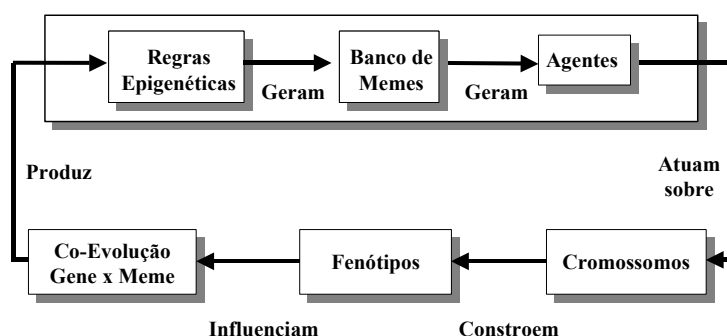


Figura 1 – Fluxograma do ciclo evolucionário da Transgenética Computacional

A metáfora está apoiada no uso de ferramentas de manipulação de cromossomos de modo intencional e planejado e na expressão dos memes através dos agentes do fluxo intracelular. A figura 1 apresenta o fluxograma do ciclo evolucionário da TC. O bloco destacado no quadro resume as inovações introduzidas pela abordagem transgenética.

As regras epigenéticas são utilizadas para gerar propostas de manipulação – os memes. O conjunto de memes forma um repositório, denominado banco de memes. Os memes, que a princípio são informações não genéticas, são codificados em agentes do fluxo intracelular que, inseridos no processo evolucionário, modificam os cromossomos alterando seus fenótipos. Os fenótipos modificados traduzem novas informações codificadas em meio genético. Tais informações podem gerar novos memes que podem conduzir a alterações nas regras epigenéticas, ou por elas serem selecionados e modificados. O quadro da co-evolução gene \times meme caracteriza que o processo de replicação e seleção da informação engloba uma interação entre dois formatos diferentes de informação: genes e memes. O conceito de co-evolução foi introduzido por Paul Ehrlich & Peter Raven (1964) e está fundamentado na ocorrência de adaptações recíprocas entre duas populações (Bowles & Gintis, 1998). A evolução da população, no caso dos algoritmos transgenéticos, pode resultar apenas do processo de manipulação realizado pelos agentes, ou estar agregada simultaneamente a um procedimento de reprodução sexual. Esse fato é determinante para a construção das duas classes básicas de algoritmos da TC: os ProtoG e os ATEIs.

Para que os memes sejam expressos no meio genético são utilizados agentes de informação, chamados de agentes transgenéticos. Esses agentes são utilizados para representar a informação memética através de um código que permita que uma proposta de manipulação possa ser convenientemente expressa no contexto genético. Além disso, os agentes transportam os memes e operacionalizam processos de competição entre memes, e entre os memes e as informações contidas nos cromossomos da população. Um agente transgenético, portanto é composto por um ou mais memes e um método de manipulação, conforme ilustrado na parte relativa ao contexto epigenético na figura 2. O repositório de memes, base para a orientação do processo de manipulação, pode ser constituído com informações obtidas de várias fontes, tanto endógenas como exógenas ao processo evolucionário. Um meme pode ser carregado em um agente transgenético tanto no formato de um fragmento de ADN – originando um bloco de montagem – ou como uma regra de manipulação genética. As regras denominadas do “tipo 1” na figura 2 são as responsáveis pela tradução do meme para uma representação compatível com a finalidade do agente transgenético e seu método de atuação, definido pelas regras do “tipo 2”.

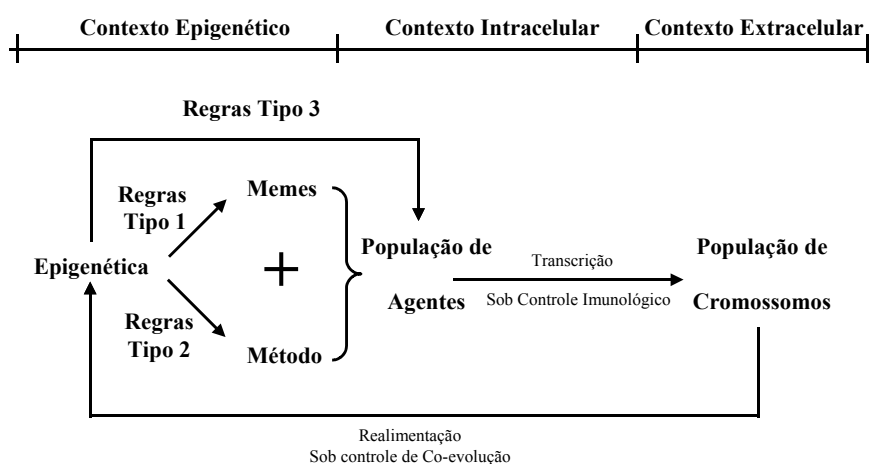


Figura 2 – Processo geral de construção de agentes transgenéticos

Um cromossomo da população também pode ser considerado como fonte de obtenção de memes, uma vez que ele próprio pode ser pensado como uma proposta completa de organização. Conseqüentemente, de uma solução podem se originar propostas parciais para o re-arranjo de blocos de genes. A população pode, portanto, contribuir para a alteração do banco de memes e para mudanças na formação dos agentes. A co-evolução gene \times meme, nesse caso, tem seus parâmetros de controle definidos pelas regras denominadas do “tipo 3”. Os cromossomos, por sua vez, podem estar ou não sujeitos a um processo de recombinação no contexto extracelular, dependendo do algoritmo transgenético adotado.

O contexto intracelular permite ainda que a TC possa incorporar o que aqui será referenciado como “ruído”. O “ruído” nos contextos intracelulares naturais provém de interferência ambiental aleatória. Através da utilização de agentes transgenéticos constituídos a partir de ruído, a abordagem pode se beneficiar do enriquecimento decorrente da incorporação de informações de diversificação.

Pode-se dizer que a TC é uma técnica de busca informada e co-evolucionária, onde genes e memes são atuadores simultâneos. Ela permite que diversas formas de informação sejam capazes de interagir em um processo de seleção guiado pela melhoria da adequação da população e pelo desempenho dos agentes.

Utilizando o paradigma de recombinação procariótica a TC pode dispensar, eventualmente, o efeito da diversificação genética introduzida pelos operadores de cruzamento, as regras de mutação, os procedimentos de migração, a elitização, etc.

3.1 Os memes na Transgenética Computacional

Os agentes transgenéticos (a serem definidos no item 3.2) podem transportar tanto ruído como memes originados de um processo de planejamento definido por regras epigenéticas. Tais regras podem traduzir diversas informações de um problema considerado ou de uma instância específica. Por exemplo, as regras epigenéticas podem expressar informações de um limite inferior, de resultados heurísticos, de análise de características da população, etc. Os memes podem ser constituídos como blocos de montagem ou como procedimentos de manipulação.

Por exemplo, um limite bem conhecido para o PQA é dado pela relaxação que o transforma em um Problema de Alocação Linear, PAL (Abreu, 1984). Em termos da Teoria dos Grafos, essa relaxação corresponde a transformar um problema de alocação de vértices, o PQA, em um problema de alocação de arestas, o PAL. Tal relaxação resulta na solução ótima de qualquer instância de ordem 3. Assim, dado um PQA de ordem n , pode-se construir um meme escolhendo-se três cidades e três facilidades aleatórias e resolvendo o problema de alocação linear associado. Como resultado, tem-se uma solução parcial de comprimento 3 que pode compor um bloco de montagem, isto é, um segmento de ADN. Pode-se também implementar um processo de competição de memes, resolvendo-se para um determinado terno de cidades, todos os PALs associados a quaisquer alocações de três facilidades e escolhendo-se como vencedor, o meme que produzir o menor custo de alocação.

Para exemplificar a concretização de um meme como um procedimento de manipulação, considere ainda a regra epigenética “relaxar o PQA como PAL”. Nesse caso, dado um certo cromossomo da população, escolhe-se ao acaso três cidades, verifica-se a alocação de facilidades para as cidades escolhidas em sua configuração e resolve-se o PAL correspondente.

3.2 Agentes da Transgenética Computacional

Um agente transgenético pode ser composto por um ou mais memes e deverá possuir um método de operação. No presente trabalho são definidos dois agentes: Vírus e Partícula Genética Móvel.

O agente Vírus foi primeiramente definido no trabalho de Goldberg *et al.* (1999) como um agente capaz de alterar diretamente a configuração cromossômica de um indivíduo e, conseqüentemente, sua adequação mantendo a integridade da informação transcrita ao longo de um certo número de iterações do algoritmo. Um Vírus, na metáfora transgenética, é uma cadeia de informação (I) associada a um método de Contaminação (Φ), e um método de Descontaminação (∇), formando uma tripla $\zeta = (I, \Phi, \nabla)$. O processo que vai desde a avaliação da resistência do indivíduo até a invasão e transcrição no cromossomo hospedeiro é denominado contaminação. O processo que controla a extinção dos efeitos da transcrição é denominado descontaminação.

Considere P uma população de N indivíduos, $P = \{S_1, S_2, \dots, S_N\}$, onde cada indivíduo S_i , $i=1, \dots, N$, é uma cadeia ou um conjunto de valores inteiros de comprimento n representando uma solução do problema e f a função adequação que retorna um número real positivo na avaliação de cada indivíduo, onde:

$$f : S_i \rightarrow \mathbb{R}^+, i = 1, \dots, N$$

A Contaminação é constituída basicamente de três etapas: ataque, transcrição e duração.

- **Ataque (A)**

Trata-se de uma etapa de avaliação que visa decidir se o cromossomo alvo é sensível ou não a um determinado tipo de manipulação. Essa fase pode ser definida através da função A , $A: S_i \rightarrow \{\text{falso}, \text{verdadeiro}\}$, $i = 1, \dots, N$, que determina a susceptibilidade do cromossomo ao ataque. Essa susceptibilidade depende de características inerentes ao Vírus e ao próprio indivíduo. Se o resultado da futura manipulação atender a critérios pré-estabelecidos pelo

projetista do algoritmo, então $A(S_j) = \text{“verdadeiro”}$, caso contrário o valor de $A(S_j)$ dependerá de um esquema imunológico de resistência ao ataque. Um exemplo de critério determinante da susceptibilidade à manipulação é a melhoria da adequação do cromossomo.

Na natureza o processo de contaminação por um microorganismo é um evento probabilístico, uma vez que os seres vivos possuem eficientes sistemas de defesa. Esse fato é considerado no modelo de contaminação artificial. No experimento computacional do atual trabalho foi utilizada a fórmula (01)

$$P = e^{\beta \frac{Val - Val_Cont}{|Média_Adequação - Val|}} \quad (01)$$

para determinar a aceitação das transcrições que redundem em diminuição da adequação do cromossomo. As variáveis Val e Val_Cont correspondem à adequação do indivíduo antes e depois da contaminação respectivamente, $Média_Adequação$ é o valor médio das adequações dos indivíduos da população e β é um parâmetro de ajuste que visa tornar o esquema mais ou menos permeável à aceitação de transcrições que pioram a adequação, especialmente importante nos estágios iniciais da evolução da população. O valor de P foi comparado com um valor sorteado ($Vsor$) em uma distribuição uniforme no intervalo $[0,1]$. Se $P > Vsor$, então $A(S_j) = \text{“verdadeiro”}$, em caso contrário $A(S_j) = \text{“falso”}$. No caso do denominador ser menor que o numerador então $P = 0$. A fórmula (01) foi inspirada no trabalho de Moritz (1990) que descreve a difusão de memes em uma população de hospedeiros.

• Transcrição (Γ)

Caso o cromossomo não apresente a necessária resistência ao ataque viral, isto é $A(S_j) = \text{“verdadeiro”}$, então a informação genética transportada pelo agente será transferida para o cromossomo de acordo com um certo *Operador de Transcrição* (Γ_i) pertencente ao conjunto Γ de todos os s possíveis operadores de transcrição do *Vírus*. O resultado da transcrição é a formação de um novo cromossomo S_j' , que incorpora a informação transportada pelo *Vírus*. Os indivíduos da população que sofrem alterações genéticas decorrentes da ação de um agente do fluxo intracelular são denominados manipulados. A manipulação decorrente da ação de um *Vírus* é uma transcrição onde: $\Gamma = (\Gamma_1, \Gamma_2, \dots, \Gamma_s)$, $\Gamma_i : S_j \rightarrow S_j' \quad j = 1, \dots, N$

• Duração (Ψ)

Ψ é uma função que determina a permanência da transcrição tanto nas futuras gerações, quanto no cromossomo contaminado. Denominando por Ω um coeficiente que pondera a estratégia de equilíbrio intensificação \times diversificação, e por $\Delta f(S_j)$ a variação da função adequação caso a contaminação ocorra, então:

$$\Psi: \Omega[\Delta f(S_j)] \rightarrow Z^+$$

O sucesso da contaminação viral produz, via de regra, uma alteração na adequação do indivíduo. Além disso, pode produzir restrições nas regras reprodutivas no caso do algoritmo contemplar um processo evolucionário que utilize o fluxo extracelular. Para um cromossomo manipulado S_j' a cadeia transcrita é considerada imutável durante todo o período de validade da contaminação. Em consequência os operadores de cruzamento, mutação e demais regras que por ventura sejam utilizadas na constituição evolucionária de blocos de montagem não poderão alterar a cadeia transcrita do *Vírus*, chamada de cadeia transgenética. A cadeia

transgenética é um bloco de montagem geneticamente dominante. No caso do cromossomo manipulado ser submetido a reprodução sexual (*ATEIs*) pode ocorrer conflito entre as cadeias transgenéticas dos pais na geração de um filho. Nessa hipótese se faz necessário um critério para o desempate da dominância. No caso das implementações do presente trabalho a dominância transgenética foi sempre definida em favor da transcrição que acarretasse a melhor adequação para o cromossomo.

A descontaminação faz cessar os efeitos da manipulação viral. Decorrido o prazo especificado pela função Ψ o cromossomo é liberado de suas restrições, voltando a apresentar um comportamento algorítmico irrestrito. Essa liberação, dependendo do tipo de agente de manipulação utilizado, poderá exigir uma operação específica. O operador responsável por colocar o indivíduo livre das restrições impostas por um agente de manipulação é denominado Operador de Alta (Π) onde:

$$\Pi_i : S'_j \rightarrow S_j \quad j = 1, \dots, N, \quad \Pi = (\Pi_1, \Pi_2, \dots, \Pi_r)$$

Para exemplificar o conceito de Vírus será descrito um dos tipos existentes chamado de vírus esquema, utilizado nos algoritmos do presente trabalho.

O *Vírus Esquema* (V^e) transporta uma cadeia de informações no formato de bloco de montagem, ou seja uma solução parcial. No caso do PQA, o vírus esquema carrega uma cadeia do tipo pares de alocação cidade \times facilidade. Considere a instância Nug5 (Nugent *et al.*, 1968) representada pelas matrizes mostradas na figura 3.

Fluxo				
0	5	2	4	1
5	0	3	0	2
2	3	0	0	0
4	0	0	0	5
1	2	0	5	0

Distância				
0	1	1	2	3
1	0	2	1	2
1	2	0	1	2
2	1	1	0	1
3	2	2	1	0

Figura 3 – Matrizes da instância Nug5

Uma solução para uma instância do PQA de ordem n pode ser representada como uma permutação de n na qual o elemento j na i -ésima posição da permutação representa a facilidade que foi atribuída à localidade i . Por exemplo, para a instância Nug5 que é de ordem 5, a solução representada pela permutação (3 2 4 1 5) é aquela na qual as facilidades 3, 2, 4, 1 e 5 foram atribuídas às localidades 1, 2, 3, 4 e 5, respectivamente. Considere o cromossomo $S_1 = (1 \ 3 \ 4 \ 5 \ 2)$ com adequação igual a 89 dada pelo valor da função objetivo e um vírus esquema v_1^e carregado com um meme de comprimento 2 representando os pares (1,4) e (4,4) de alocação cidade \times facilidade. A figura 4 mostra a transcrição de v_1^e em S_1 . Essa transcrição implica fixar os pares representados na cadeia do vírus na configuração do cromossomo. O método de operação do vírus estabelece a forma pela qual o cromossomo será reorganizado de modo a comportar a sua cadeia. No caso exemplificado na figura 4 o operador de transcrição executa operações de 2-troca para inserir a cadeia requerida. A transcrição do vírus implica em uma nova configuração para o cromossomo, $S_1' = (3 \ 1 \ 5 \ 4 \ 2)$ com adequação igual a 70. Como a transcrição do vírus melhora a adequação do cromossomo, então $A(S_1) = \text{“verdadeiro”}$ e S_1 é transformado em S_1' .

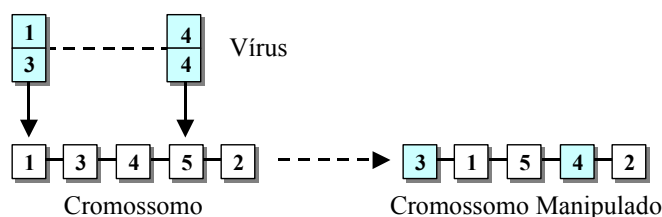


Figura 4 – Uma forma de manipulação de um vírus esquema

Devido à variedade de tipos e representações possíveis para a informação no formato não genético, é indispensável que existam tipos específicos de vírus que possam incorporar os diferentes métodos necessários à tradução dessa informação em formato genético. Outros exemplos de vírus podem ser encontrados nos trabalhos de Goldberg & Gouvêa (2000, 2001).

Kubota *et al.* (1996) e Fukuda *et al.* (1999) apresentam um algoritmo denominado VEGA – Virus Evolutionary Genetic Algorithm onde usam também o conceito de Vírus. Os autores utilizam a mesma terminologia, contudo não é possível traçar um paralelo entre esse trabalho e a proposta da TC. VEGA é um algoritmo que trabalha com duas populações – uma de vírus e outra de hospedeiros. Basicamente os vírus são gerados a partir de subcadeias dos cromossomos e possuem dois operadores: de transcriptase reversa e transdução. A transcrição realizada pelo denominado agente “vírus” pode ser desfeita imediatamente após a sua realização pelo processo de cruzamento dos cromossomos. O trabalho dos autores citados representa uma pequena modificação da proposta dos Algoritmos Genéticos, uma vez que o trabalho dos vírus é fundamentalmente realizar operações parciais de cruzamento, ou seja, são agentes que operacionalizam uma forma especial de operador para as cadeias da população. O trabalho de Fukuda não desenvolve o conceito de “resistência” imunológica como presentemente apresentado, nem elabora o conceito de infiltração de informação pela via intracelular. Além disso, a TC é uma técnica de manipulação planejada que não depende da existência de um algoritmo genético de substrato, uma vez que pode desenvolver seu processo evolucionário exclusivamente com base no paradigma de recombinação procariótica.

Outro agente utilizado neste trabalho é a Partícula Genética Móvel. As partículas genéticas móveis são tipos especiais de vírus esquema. Na natureza, elementos genéticos móveis são fragmentos de ADN que podem ser fagocitados pelas células, acabando por se integrar ao ADN do núcleo. No contexto computacional define-se uma PGM como um agente constituído por uma cadeia de informação (I) e um método de contaminação (C), formando uma dupla $PGM = (I, C)$. A PGM não possui método de descontaminação uma vez que sua cadeia, quando transcrita, não é tornada inviolável, isto é, não fixa a informação transcrita como geneticamente dominante. A regra de resistência imunológica ao ataque da PGM é a da estrita melhoria da adequação. Os operadores de transcrição são semelhantes aos operadores dos Vírus. Portanto, uma PGM pode ser considerada como um caso especial de vírus esquema com tempo de vida igual a zero e critério guloso de resistência imunológica. As PGM podem representar um vetor extremamente eficiente e simples para o transporte de informações. Por não obrigarem que a inviolabilidade de uma certa transcrição seja mantida, permitem ao processo de busca uma maior liberdade de diversificação. Esses agentes constituem a base do paradigma de manipulação proposto pelo algoritmo Proto-Gene.

A idéia da imunologia na TC está representada no fato da população de cromossomos ser capaz de resistir à manipulação dos agentes em função do desenvolvimento do processo

evolucionário e da proposta de manipulação do agente. O efeito imunológico é uma ferramenta para operacionalizar a co-evolução agentes × população de cromossomos. Na medida em que as iterações do processo evolucionário se seguem, a população vai se tornando resistente às cepas de um crescente número de agentes transgenéticos. As informações do banco são atualizadas por informações que vão sendo geradas na população de cromossomos durante o processo evolucionário, criando novas e mais eficientes cepas de agentes que mesclam informações exógenas e endógenas. Quando a população se torna extremamente resistente a todas as manipulações propostas pelos agentes, o algoritmo converge por resistência imunológica da população.

4. Algoritmos Transgenéticos

O presente trabalho aborda duas possibilidades de compor o processo de manipulação transgenética. A primeira utiliza as três vias de evolução consideradas pela TC. Tais algoritmos são denominados Algoritmos Transgenéticos Extra-Intracelulares (ATEIs). Uma segunda classe utiliza-se somente da evolução memética e da evolução por recombinação procariótica. Essa classe é denominada Proto-Gene (ProtoG). Nela os cromossomos atendem ao paradigma de recombinação procariótica sendo exclusivamente expostos a ataques de agentes do fluxo intracelular. Os ProtoG, portanto, propõem que a manipulação via agentes seja o único mecanismo de compartilhamento e difusão de informações no processo evolucionário.

Os Algoritmos Transgenéticos permitem que vários níveis de competição sejam viabilizados no contexto do processo evolucionário. Conforme descrito no item 3.1, memes podem competir para formar agentes. Também, diversos agentes podem competir para transcrever sua informação em um cromossomo. Uma competição no contexto dos agentes é um processo de concorrência que julga o efeito de várias propostas de manipulações pretendidas. Os critérios de julgamento podem incluir a melhoria na adequação dos cromossomos ou outros tais como uma maior alteração da configuração cromossômica. Finalmente, um agente compete com a informação já incorporada nos cromossomos. A fórmula (01) representa um modelo para o julgamento dessa competição. Agentes vencedores nessa etapa conseguem a transcrição.

4.1 Algoritmos Transgenéticos Extra e Intracelular (ATEIs)

De uma forma geral pode-se descrever os ATEIs como nos quadros I e II, sendo eles de três tipos básicos: ATEI, ATEI-I e ATEI-IT. As setas marcam os passos que correspondem aos de um algoritmo genético clássico. Não há necessidade, porém, da inclusão da mutação, uma vez que os agentes transgenéticos, se requerido, podem ser projetados especificamente para promover a diversificação. O quadro I descreve um ATEI que não comporta evolução imunológica, ou seja, o banco de memes desse algoritmo não é atualizado com informações geradas na população. Cabe ressaltar que as informações geradas no processo evolucionário não são apenas as geradas nas novas configurações de cromossomos que surjam na população. Tais informações podem, por exemplo, advir do controle de diversidade na população. No caso específico do PQA, pode-se detectar que um certo par cidade × facilidade não está representado ou possui pouca representação na população. O quadro II descreve um ATEI que possui um critério para atualização do seu banco de memes durante o processo evolucionário. O ATEI-IT é basicamente igual ao ATEI-I, com a diferença que existe competição entre os agentes para o ataque a um cromossomo (conforme é mostrado no item 5.3). Para tornar simples a descrição do algoritmo, o critério adotado é o surgimento de uma solução melhor que todas as existentes até uma determinada iteração do algoritmo.

Quadro I – Algoritmo Transgenético Extra-Intracelular

<i>Algoritmo Transgenético Extra-Intracelular</i>
<p>Início Carregar um Banco de Memes Desenhar um conjunto de agentes de manipulação com base no Banco de Memes Gerar uma população inicial e avaliá-la Repetir Carregar um subconjunto de agentes de manipulação → Selecionar um conjunto de pais na população Manipular os cromossomos pais sensíveis → Cruzar os pais Dar Alta aos indivíduos liberados → Avaliar a Adequação da população Até que o Critério_de_Parada seja atendido Fim</p>

Quadro II – AT Extra e Intracelular com Evolução Imunológica

<i>Algoritmo Transgenético Extra e Intracelular com Evolução Imunológica</i>
<p>Início Carregar um Banco de Memes Desenhar um conjunto de agentes de manipulação com base no Banco de Memes Repetir Gerar uma população inicial e avaliá-la Repetir Carregar um subconjunto de agentes de manipulação Manipular os indivíduos sensíveis da população de acordo com regras transgenéticas → Cruzar os pais Dar Alta aos indivíduos liberados → Avaliar a Adequação da população → Selecionar um conjunto de pais na população Até que Critério_de_Parada_Evolucionário seja atendido Se Sol_Corrente \leq Melhor_Solução então Início Melhor_Solução \leftarrow Sol_Corrente Incluir a Melhor_Solução no Banco de Memes Redesenhar o conjunto de agentes de manipulação com base no Banco de Memes Fim Até que o Critério_de_Parada_Imunológico seja atendido Fim</p>

4.2 Algoritmos Proto-Gene (ProtoG)

Uma metáfora adequada à descrição do ProtoG é a da “sopa primordial” ou o caldo de aminoácidos do oceano primitivo. Nesse algoritmo os cromossomos são expostos a um ataque direto dos agentes do fluxo intracelular que devem competir entre si e com as defesas dos cromossomos para concretizarem a transcrição de seus códigos. Nesse cenário

evolucionário existe um grande espaço para construção de um processo de competição entre agentes do fluxo intracelular e aberto à inclusão do ruído.

Os Algoritmos ProtoG são compostos de duas fases. A primeira inclui a obtenção de memes e a construção dos agentes. Na segunda fase, os agentes carregados com os memes vencedores do processo de competição na primeira fase, atuam sobre os cromossomos da população tentando introduzir suas informações. Também na segunda fase ocorre o ataque dos agentes carregados com ruído. A cada iteração, um ou mais agentes percorrem a população manipulando os cromossomos sensíveis. No caso do surgimento de uma informação de boa qualidade, o algoritmo poderá realimentar o banco de memes com informações derivadas desse código genético. A Figura 5 mostra um quadro geral da arquitetura dos algoritmos ProtoG, resumindo as duas fases e sua interação.

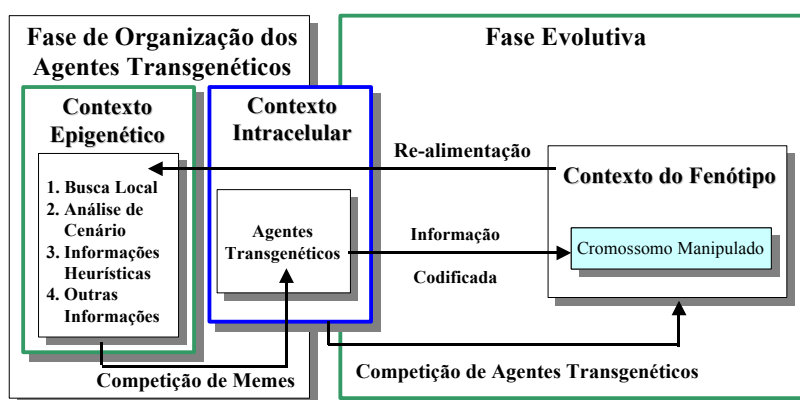
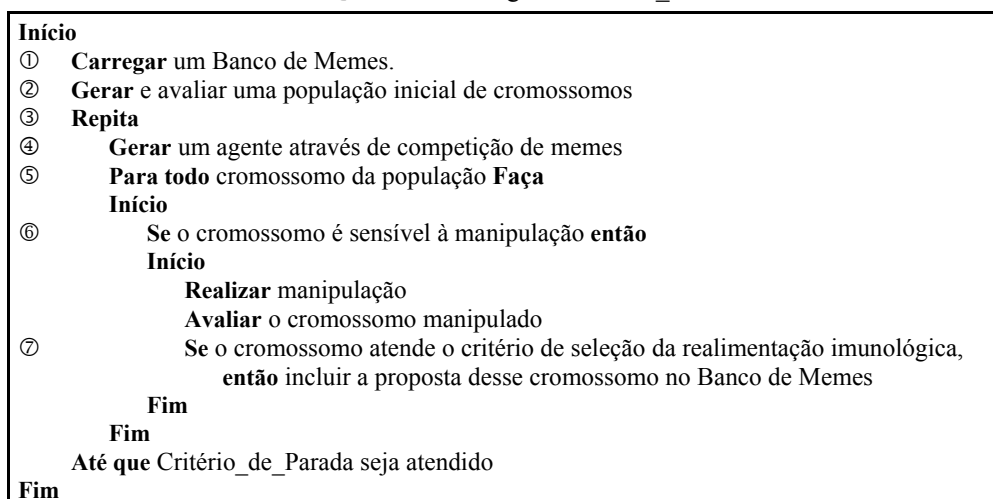


Figura 5 – Arquitetura dos Algoritmos Proto-Gene

Os algoritmos ProtoG podem ser descritos, de uma forma geral, como apresentados no quadro III.

Quadro III – Algoritmo Proto_G



Para esclarecer a execução dos passos descritos no quadro III será apresentado um exemplo numérico considerando uma aplicação ao Problema do Caixeiro Viajante, PCV, sobre o grafo $G=(N,A)$ caracterizado pela matriz de distância mostrada na tabela 1.

Tabela 1 – Matriz de Distância do PCV

Nó \ Nó	1	2	3	4	5	6	7	8	9
1		8	12	15	17	9	5	7	6
2			6	8	15	16	10	12	18
3				13	6	7	9	3	19
4					6	8	10	4	12
5						4	12	20	22
6							12	10	9
7								3	4
8									12

Supondo que algumas das regras epigenéticas do contexto epigenético sejam:

1. Incluir no banco de memes estruturas de conexão minimal – Árvores Geradoras Mínimas.
2. Incluir no banco de memes as estruturas genéticas dos melhores indivíduos da população.
3. Selecionar as estruturas que vão fornecer as cadeias que serão transportadas pelos agentes transgenéticos através de critérios convenientes de competição.
4. Selecionar, através de critérios convenientes de competição, as cadeias que serão escolhidas como memes de manipulação.
5. Utilizar cadeias de comprimento C , $C \leq 2$, para formar os memes dos agentes transgenéticos.
6. Utilizar a Partícula Genética Móvel como agente de manipulação.

Os passos do quadro III podem ser assim calculados:

① Carregar um Banco de Memes.

Conforme as regras anteriores, o banco de memes poderá conter blocos de uma Árvore Geradora Mínima (AGM) e de um ou mais dos melhores indivíduos da população. Neste exemplo o passo 1 poderia representar a colocação dos códigos abaixo transcritos e representados por seqüências de arestas, em que, para o primeiro passo do algoritmo, o melhor indivíduo é obtido por uma busca na população de cromossomos. Denominado por Estrutura_Global_1 o melhor indivíduo da população tem-se, por exemplo:

Estrutura_Global_1: (4-5)(5-6)(6-8)(8-2)(2-1)(1-3)(3-7)(7-9)(9-4). A adequação é dada pelo somatório dos pesos das arestas do ciclo que é igual a 77.

Estrutura_Global_2: (2-3)(3-8)(8-4)(8-7)(7-1)(7-9)(7-5)(5-6) obtida de uma AGM do grafo que representa a instância estudada. O valor dessa árvore é 35.

② Gerar e avaliar uma população inicial de cromossomos.

Supondo que a população inicial seja gerada aleatoriamente, pode-se considerar que os seguintes cromossomos, além do melhor indivíduo anteriormente listado em Estrutura_Global_1 fazem parte da população:

Cromossomo_1: (1-2)(2-3)(3-4)(4-5)(5-6)(6-7)(7-8)(8-9)(9-1) com adequação igual a 78.

Cromossomo_2: (3-9)(9-4)(4-5)(5-8)(8-1)(1-6)(6-2)(2-7)(7-3) com adequação igual a 108.

④ Gerar um agente através de competição de memes do banco.

O primeiro nível de competição do algoritmo é caracterizado pela seleção das estruturas que serão utilizadas como doadoras de memes. Supondo que a estrutura Hamiltoniana de melhor adequação e a AGM do banco com menor grau fossem selecionadas, as cadeias mostradas na tabela 2 poderiam ser obtidas para compor a proposta de manipulação da partícula genética móvel. Cada cadeia possui comprimento 2 e seu valor é calculado pelo somatório dos pesos das arestas que a compõem.

Tabela 2 – Cadeias que formam memes para o PCV

Proposta_Global_1		Proposta_Global_2	
Meme	Valor	Meme	Valor
(4-5)(5-6)	10	(3-8)(8-4)	7
(6-8)(8-2)	22	(1-7)(7-9)	9
(2-1)(1-3)	10	(9-7)(7-5)	10
(3-7)(7-9)	13	(3-8)(8-7)	6
(9-4)(4-5)	18	(1-7)(7-5)	11

Supondo agora que o critério de seleção das subcadeias fosse o menor custo, nesse caso o meme vencedor seria a cadeia (3-8)(8-7), com valor 6. Pode-se observar que a cadeia selecionada coincide, por motivos fortuitos, com as duas menores arestas do grafo.

⑥ Se o cromossomo é sensível à manipulação então

A verificação da sensibilidade dos cromossomos à manipulação da PGM é realizada a partir do cálculo do benefício causado pela possível transcrição, conforme prevê o método de atuação das PGM. Considerando-se ainda que a PGM incorpora um procedimento simples de transcrição, com base na permutação da cadeia do cromossomo – válido para o caso do grafo completo – então, são possíveis as seguintes transcrições do meme vencedor sobre os dois cromossomos da população selecionados para exemplificar o procedimento.

da implementação desse algoritmo foi testar o impacto do planejamento transgenético na aceleração de um algoritmo genético tradicional, bem como seu poder de guiar a busca quando são disponíveis memes de alta qualidade. A população inicial do algoritmo foi sempre gerada aleatoriamente. O operador de cruzamento empregado foi o Insertpath (Ahuja *et al.*, 1997). Todos os cromossomos são considerados equiprováveis para escolha reprodutiva. O ataque viral se dá por escolha aleatória dos vírus e, imediatamente após a escolha dos dois cromossomos que vão gerar a prole. Os memes que constituíram as cadeias dos vírus foram escolhidos aleatoriamente a partir de trechos do cromossomo da melhor solução conhecida para a instância. Foram utilizados vírus de tamanho constante e proporcional ao tamanho da instância. O número de vírus foi determinado de forma proporcional ao tamanho da instância. As instâncias foram solucionadas tanto pelo AG de substrato do ATEI como pelo ATEI em teste. A tabela 3 detalha os dados de cada instância para os testes dos algoritmos descritos nos itens 5.1 a 5.3.

Tabela 3 – Dados dos Experimentos da tabela 4

Tamanho da Instância	Tamanho da População	<i>ATEI, ATEI-I, ATEI-IT</i>	
		<i>Quantidade de Agentes</i>	<i>Tamanho dos Agentes</i>
20-25-26	1000	20	3
30-32-36	1000	20	4
60-64	1000	60	8
100	1000	100	15

A tabela 4 apresenta o resultado obtido pela implementação do algoritmo ATEI em comparação com os dados fornecidos pelo algoritmo genético de substrato. A coluna *Instância* mostra o nome da instância e a coluna *Melhor Solução* mostra a melhor solução encontrada na literatura. A coluna *AG de Substrato* está dividida em três partes: *Solução* que mostra a melhor solução obtida pelo algoritmo de substrato, *Iterações* e *Tempo* que mostram, respectivamente, o número de iterações e o tempo de processamento do algoritmo genético de substrato. As soluções encontradas pelo ATEI foram, em todos os casos, de valor igual à melhor solução da literatura, sendo que no caso de HAD20 e ESC32E, as soluções encontradas são diferentes das apresentadas no QAPLIB (2000), contudo de mesmo valor. Portanto, as colunas relativas ao ATEI mostram apenas o número de iterações e o tempo de processamento para a obtenção da melhor solução conforme a literatura. O ATEI utilizou apenas o vírus esquema como agente de manipulação. Os resultados computacionais demonstram o efeito favorável da infiltração das informações. O AG de substrato obtém soluções que ficam, em média 19% afastadas da melhor solução apresentada no QAPLIB. Além disso o ATEI encontra a melhor solução da literatura com uma economia média de tempo em relação ao AG de substrato de aproximadamente 63%.

Tabela 4 – Resultados dos testes do ATEI

Instância	Melhor Solução	AG de Substrato			ATEI	
		Solução	Iterações	Tempo	Iterações	Tempo
HAD20	6922	6954	12253	10	725	2
NUG20	2570	2722	5667	4	855	1
ROU20	725522	777202	3444	3	715	1
SCR20	110030	135468	3232	2	348	1
TAI20a	703482	754346	5460	4	692	1
CHR20a	2192	3130	10027	8	512	1
CHR22a	6156	7814	3778	4	977	3
CHR25a	3796	6910	6141	10	548	2
BUR26a	5426670	5511015	4220	7	660	3
ESC32a	130	244	2113	7	1775	14
STE36c	8239110	9622982	18611	84	1455	16
TAI60a	7208572	7759584	20250	268	788	40
TAI60b	608215054	747287035	117595	404	994	49
SKO64	48498	53864	8662	241	1428	83
SKO100a	152002	163182	42684	3164	1077	224
WIL100	273038	291748	3395	348	1207	242

5.2 ATEI Imunológico (ATEI-I)

O ATEI-I foi constituído com base no ATEI do experimento do item 5.1, conforme o algoritmo apresentado no quadro II. O presente algoritmo visa testar a possibilidade de guiar o processo evolucionário a partir da resistência imunológica de uma certa população. Ao fim de cada ciclo, ou seja, quando o algoritmo transgenético converge pela resistência imunológica da população, o banco de memes pode ser realimentado caso tenha emergido alguma solução de qualidade. O banco de memes inicial foi carregado com a melhor solução obtida pelo algoritmo genético de substrato. As tabelas 5, 6, 7 e 8 resumem os resultados deste teste computacional.

O ATEI-I foi aplicado às instâncias Esc128, Kra30a e Nug25. A tabela 5 mostra o desempenho do ATEI-I na solução da instância Esc128. A solução inicial foi obtida pelo algoritmo genético de substrato com o valor 152, encontrando-se afastada em 138% da melhor solução da literatura. Cada linha da tabela representa uma iteração imunológica. A melhor solução da iteração realimenta o banco de memes e orienta a construção dos vírus esquema na iteração seguinte. Foram experimentados, através de vários esquemas transgenéticos, dois comprimentos para o vírus, respectivamente 25 e 15 posições, permitindo-se que o parâmetro β da fórmula (01) pudesse assumir os valores 0,5, 1 e 2. Os esquemas foram rodados simultaneamente, sendo que a tabela 5 registra os valores obtidos para o melhor cromossomo dentro de cada iteração, bem como os parâmetros que levaram a esse resultado. Foram consideradas, em cada iteração, todas as combinações dos parâmetros anteriormente descritos. Observa-se que uma solução cujo valor é igual ao melhor relatado na literatura é encontrada na iteração 14. Ao carregar o banco de memes com essa solução não foi mais possível produzir uma solução de valor inferior a 64. A solução convergiu para esse último valor para todas as combinações dos parâmetros. Na tentativa de explorar mais profundamente a região dessa melhor configuração, criou-se um esquema especial apresentado na linha 15

que redundou em nova solução de valor igual a 64. Noticia-se que a utilização do valor $\beta = 2$ produz uma taxa de aceitação de transcrições que pioram a adequação de, aproximadamente, 25% do total de manipulações. Valores menores de β aumentam essa faixa chegando a taxas de aceitação de até 30% do total de manipulações para β igual a 0,5. Ressalte-se que o fator que produz a diferença de desempenho entre as iterações é apenas a qualidade da solução que alimenta o banco de memes. A cada iteração uma nova população é gerada aleatoriamente, sendo que a emergência de um cromossomo campeão ocorre porque uma cepa mais poderosa de vírus é utilizada no ataque transgenético.

Tabela 5 – Resultados do ATEI-I aplicado à instância Esc128

Iteração	Valor da Solução	Vírus		Número de Cruzamentos	β
		Tamanho	Quantidade		
01	152	---	---		---
02	136	25	250	3076	1
03	120	25	250	3511	1
04	112	25	250	5113	1
05	106	25	250	3754	1
06	96	15	200	3940	1
07	90	25	250	3929	2
08	84	25	250	2376	1
09	82	25	250	3428	0,5
10	80	25	250	2206	2
11	78	25	250	3294	2
12	74	25	250	3781	1
13	66	25	250	2510	0,5
14	64	25	250	3306	1
15	64	35	250	1636	3

A tabela 6 mostra o resultado do teste na instância Kra30a. Nesse caso a solução inicial obtida pelo algoritmo genético de substrato tem valor 100590 e encontra-se afastada em 13,15% da melhor solução da literatura. Observe-se que, nesse experimento, foram testados esquemas de parâmetros diferentes do utilizado no caso do Esc128. Como no caso anterior, cada iteração testou o desempenho das combinações de tamanhos e de valores de β relacionados na sua linha. Os melhores resultados qualitativos estão lançados na coluna *Valor da Solução*. A coluna *Número de Cruzamentos* da tabela 6 acumula todos os cruzamentos realizados ao longo das rodadas necessárias a testar os conjuntos de configurações, dando uma idéia do esforço computacional necessário.

Tabela 6 – Resultados do ATEI-I aplicado à instância Kra30a

Iteração	Valor da Solução	Vírus		Número de Cruzamentos	β
		Tamanho	Quantidade		
01	100590	---	---	10201	---
02	93400	2-3-5	100	45734	1; 2; 4
03	93400	2-3-5	100	15706	1; 0,5; 2
04	88900	2-3-5	100	58693	0,2; 2,5; 7

A tabela 7 resume o experimento para a instância Nug25. A solução inicial foi obtida pelo algoritmo genético de substrato e tem valor 4060, encontrando-se afastada em 8,44% do ótimo. Neste teste foi possível examinar um número maior de combinações de parâmetros em virtude do tamanho da instância. Observe-se que o algoritmo não alcança o melhor valor da literatura quando ocorre a parada pelo critério da resistência imunológica.

A tabela 8 mostra um novo teste de evolução imunológica na mesma instância Nug25, utilizando-se os mesmos parâmetros no ATEI-I, contudo carregando-se o banco de memes com uma solução inicial de melhor qualidade (com valor 3864) afastada 3,21% do ótimo. Nesse caso a solução inicial de alimentação do banco de memes foi obtida por escolha aleatória na população final do ATEI do experimento descrito no item 5.1.

Tabela 7 – Resultados da primeira aplicação do ATEI-I à instância Nug25

Iteração	Valor da Solução	Vírus		Número de Cruzamentos	β
		Tamanho	Quantidade		
01	4060	---	---	6553	---
02	3870	2-3-4-5	100	41764	0,2; 0,5; 1,0 2,5; 4,0; 5,0
03	3754	2-3-4-5	100	69500	0,2; 0,5; 1,0 2,5; 4,0; 5,0

Tabela 8 – Resultados da segunda aplicação do ATEI-I à instância Nug25

Iteração	Valor da Solução	Vírus		Número de Cruzamentos	β
		Tamanho	Quantidade		
01	3864	---	---	2101	---
02	3744	2-3-4-5	100	36850	0,2; 0,5; 1,0 2,5; 4,0; 5,0

5.3 ATEI Imunológico em Time (ATEI-IT)

O ATEI-IT foi constituído basicamente como o ATEI-I do experimento descrito no item 5.2, adicionando-se, na etapa intracelular, um novo processo de competição entre os agentes. Neste experimento o banco de memes é obtido a partir de um certo número Q das melhores soluções existentes na população de algoritmo ATEI-I do experimento 5.2, conforme especificado na coluna *Soluções e Afastamento % do ótimo* da tabela 9.

Os memes competem para formar agentes da seguinte forma: escolhe-se um tamanho p aleatório entre 7 e 15% da ordem da instância para o meme. Sorteiam-se p localidades para compor os memes em competição. Cada solução que está no time possui uma alocação facilidade \times localidade própria para as p localidades. Assim, cada solução do banco vai fornecer a sua alocação como a proposta para os p pares localidade \times facilidade, formando-se um número de memes em competição igual ao número Q de soluções do time. Foi utilizado como critério de seleção para a escolha do meme vencedor o custo da alocação composta em cada meme. O meme de menor custo é escolhido para formar o vírus. Por esse processo são formados r vírus. Dentre os r vírus formados pelas regras anteriores um percentual deles é sorteado, com equi-probabilidade, para atacar um cromossomo. O que obtiver o melhor valor

de adequação com sua transcrição é o vencedor. Se a transcrição do vencedor piora a adequação do cromossomo ele é submetido ao critério de resistência imunológica definido pela fórmula (01) para confirmar ou não sua manipulação.

A tabela 9 mostra a evolução imunológica de um ATEI-IT aplicado à solução da instância Nug25. No caso do ATEI-IT testado o banco de memes é constituído por um time de soluções composto por 5 soluções. As soluções foram obtidas por escolha aleatória na população final do ATEI-I do experimento 5.2. A segunda coluna mostra o valor da adequação das soluções utilizadas para compor o time de soluções.

Tabela 9 – Resultados do ATEI-IT aplicado à instância Nug25

Iteração	Soluções		Vírus			Número de Cruzamentos	β
	Soluções e Afastamento % do ótimo	Solução do ATEI-IT	Tamanho	Número de Vírus no Time	Quantidade de vírus		
01	3750 – 0,16% 3754 – 0,27% 3792 – 1,28% 3898 – 4,11% 3786 – 1,12%	3744	3	20	200	2607	0,2

5.4 Proto-Gene (ProtoG)

Para formar o banco de memes utilizou-se um algoritmo multistart que em sua fase construtiva gera soluções aleatórias e em sua fase de busca local utiliza uma vizinhança de 2-troca. Neste experimento os agentes empregados foram as Partículas Genéticas Móveis. O tamanho das PGMs foi escolhido de modo aleatório e com probabilidade uniforme entre 7 e 15% do tamanho do cromossomo. Os agentes são constituídos de forma análoga ao ATEI-IT apresentado no item 5.3. A população é gerada aleatoriamente. O método de ataque determina que cada agente transgenético gerado na primeira fase procure manipular todos os indivíduos da população. A regra de parada adotada foi de 1500 iterações sem melhoria de qualquer indivíduo da população ou no máximo 100 000 iterações totais. O tamanho do time de soluções que constitui o banco de memes do ProtoG é igual 100 e a população é constituída por 2000 cromossomos. Foi observado um comportamento bastante homogêneo para o algoritmo quando fixados os parâmetros do esquema transgenético. Assim optou-se, face ao grande número de instâncias e esquemas a examinar, apresentar o resultado de uma única rodada do algoritmo, mesmo que alguns dos trabalhos de comparação tenham apresentado o melhor resultado obtido em um grande número de rodadas.

As tabelas 10 e 11 resumem os resultados obtidos na implementação do algoritmo ProtoG em um conjunto de 70 instâncias do QAPLIB. As instâncias selecionadas para os testes computacionais têm ordem entre 20 e 100. Os algoritmos cujos resultados são apresentados foram desenvolvidos segundo diferentes abordagens metaheurísticas e reconhecidamente reportam bons resultados para o conjunto de instâncias que relatam.

A coluna *Melhor Solução* mostra os melhores resultados registrados no QAPLIB. As demais colunas apresentam os resultados em termos de afastamento percentual em relação ao valor apresentado na primeira coluna. A coluna *Grasp* foi obtida do trabalho de Resende *et al.* (1996) e de Maniezzo & Colomi (1999). Essa coluna apresenta o melhor resultado reportado. A coluna *Multistart* foi obtida através da implementação efetuada pelos autores para a composição do banco de memes. O valor constante da coluna é o melhor resultado dentre 100 rodadas do

algoritmo Multistart. A coluna *Reinf. Learning* mostra os resultados relatados nos trabalhos de Miagkikh & Punch III (1999a, 1999b). A coluna *Ant System* foi obtida do trabalho de Maniezzo & Colomi (1999). A coluna *GDH* (Genetic-Descent Hybrid Method) foi obtida em Taillard & Gambardella (1997) e representa um valor médio de afastamento da melhor solução.

Tabela 10 – Resultados da implementação ProtoG – 1ª parte

Instância	Valor Solução	GRASP	Multistart	Reinf. Learning	Ant System	GDH	ProtoG
BUR26a	5426670	0	0,09	0	0	0,03	0
BUR26b	3817852	0	0,01	0	0	0,04	0
BUR26c	5426795	0	0,003	0	0	0	0
BUR26d	3821225	0	0,004	0	0	0,0004	0
BUR26e	5386879	0	0,008	0	0	0,0012	0
BUR26f	3782044	0	0	0	0	0	0
BUR26g	10117172	0	0	0	0	0,001	0
BUR26h	7098658	0	0,008	–	0	0,001	0
CHR20a	2192	1,5	9,6	0	0	–	0
CHR20b	2298	5,92	18,1	0	2,79	–	0
CHR20c	14142	0	18,6	0	0	–	0
CHR22a	6156	2,31	5,9	0	0	–	0
CHR22b	6194	1,7	5,4	0	0,97	–	0
CHR25a	3796	2,32	18,2	–	0	–	0
ESC32a	130	1,54	7,69	0	0	–	0
ESC32b	168	0	14,28	0	0	–	0
ESC32c	642	0	0	0	0	–	0
ESC32d	200	0	0	0	0	–	0
ESC32e	2	0	0	0	0	–	0
ESC32f	2	0	0	0	0	–	0
ESC32g	6	0	0	0	0	–	0
ESC32h	438	0	0,45	–	0	–	0
ESC64a	116	0	0	0	–	–	0
ESC128	64	0	0	–	–	–	0
HAD20	6922	0	0	–	0	–	0
KRA30a	88900	0	3,4	0	0	1,08	0
KRA30b	91420	0,2	0,9	0	0	0,17	0,2
LIPA20a	3683	0	1,3	0	0	–	0
LIPA20b	27076	0	0	–	0	–	0
LIPA30a	13178	0	1,6	0	0	–	0
LIPA30b	151426	0	0	–	0	–	0
LIPA40a	31538	1,13	1,3	0	1,02	–	1,0
LIPA40b	476581	0	16,9	–	0	–	0
LIPA50a	62093	0,98	1,08	–	–	–	0,9
LIPA50b	1210244	0	0	–	–	–	0
LIPA60a	107218	0,8	0,93	–	–	–	0,93
LIPA60b	2520135	0	19,1	–	–	–	0
NUG20	2570	0	0,6	0	0	0,02	0
NUG21	2438	–	0,7	–	–	–	0
NUG22	3596	–	0	–	–	–	0

Tabela 11 – Resultados da implementação ProtoG – 2ª parte

Instância	Valor Sol	GRASP	Multistart	Reinf. Learning	Ant System	GDH	ProtoG
NUG24	3488	–	0,9	–	–	–	0
NUG25	3744	–	0,5	–	–	–	0
NUG30	6124	0,2	1,0	0	0	0,16	0,06
ROU20	725522	0	1,7	–	0	–	0
SCR20	110030	0	0,03	0	0	–	0
STE36a	9526	1,81	4,2	0	0,76	–	0
STE36b	15852	0,92	5,3	0	0,25	–	0
STE36c	8239110	0,64	1,44	0	0,33	–	0,16
SKO42	15812	0,48	0,9	0	–	0,09	0,22
SKO49	23386	0,3	1,1	0	–	0,19	0,11
SKO56	34458	0,51	0,9	0	–	0,15	0,23
TAI20a	703482	0	2,9	–	0	–	0
TAI20b	122455319	–	0,6	–	–	0,09	0
TAI25a	1167256	1,34	2,2	–	0,55	–	1,4
TAI25b	344355646	–	0,3	–	–	0,14	0
TAI30a	1818146	1,58	3,0	–	1,46	–	0,6
TAI30b	637117113	–	0,5	–	–	0,03	0
TAI35a	2422002	1,9	2,97	–	1,64	–	1,4
TAI35b	283315445	–	0,56	–	–	0,019	0
TAI40a	3139370	2,2	3,1	–	2,05	1,307	1,26
TAI40b	637250948	–	0,24	–	–	0,20	0
TAI50a	4941410	–	3,43	–	–	–	1,9
TAI50b	458821517	–	0,4	–	–	0,009	0,018
TAI60a	7208572	–	3,4	1,26	–	–	3,4
TAI60b	608215054	–	0,7	0	–	0,009	0,048
TAI80b	818415043	–	3,1	–	–	0,22	0,8
THO30	149936	0	1,3	0	0	–	0
THO40	240516	0,62	1,4	0,66	0,66	–	0,66
WIL50	48816	–	0,4	–	–	–	0,1
WIL100	273038	0,33	0,83	–	–	–	0,68

Para resumir os resultados das tabelas 10 e 11 são apresentados os gráficos das figuras 7 e 8. O eixo vertical da figura 7(a) apresenta o percentual de soluções, obtidas pelos algoritmos, que tiveram afastamento 0 da melhor solução apresentada no QAPLIB. Observa-se que o ProtoG apresenta o terceiro melhor resultado posicionando-se atrás do Reinforcement Learning e praticamente empatado com o Ant System. O eixo vertical da figura 7(b) corresponde ao afastamento médio das soluções obtidas pelos algoritmos em relação à melhor solução apresentada no QAPLIB. Observa-se que os algoritmos Grasp e Ant System apresentam os maiores afastamentos médios. O Reinforcement Learning apresenta o menor afastamento, seguido pelo GDH e pelo ProtoG.

Cada algoritmo apresenta um número diferente de testes, sendo: 70 para o ProtoG e o Multistart; 55 para o GRASP; 45 para o Ant-System; 38 para o Reinforcement Learning; e 24 para o GDH. O trabalho referente ao algoritmo de Reinforcement Learning relata dentre as instâncias do experimento computacional, apenas uma instância da classe *Tainna*, onde *nn* representa a ordem da instância, uma classe onde a maioria dos algoritmos encontra dificuldade de solução.

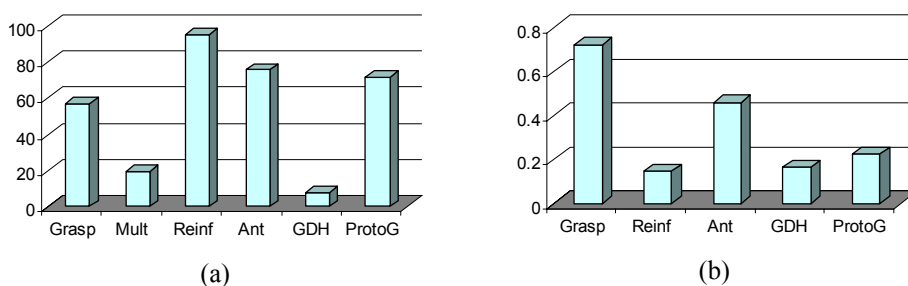


Figura 7 – (a) Afastamento 0. (b) Afastamento médio.

A figura 8 apresenta uma comparação de desempenho qualitativo absoluto entre o algoritmo ProtoG e cada um dos demais algoritmos. São consideradas em cada quadro apenas as instâncias reportadas em comum nos respectivos experimentos computacionais. O eixo vertical dos gráficos apresenta o número de instâncias para as quais algum dos algoritmos apresentou melhores resultados. A barra da esquerda de cada gráfico representa o número de instâncias em que o algoritmo comparado com o ProtoG obteve uma solução de melhor qualidade. A barra central representa o número de instâncias nas quais o ProtoG obteve uma solução de melhor qualidade. A barra da direita representa o número de casos nos quais as soluções são iguais para ambos.

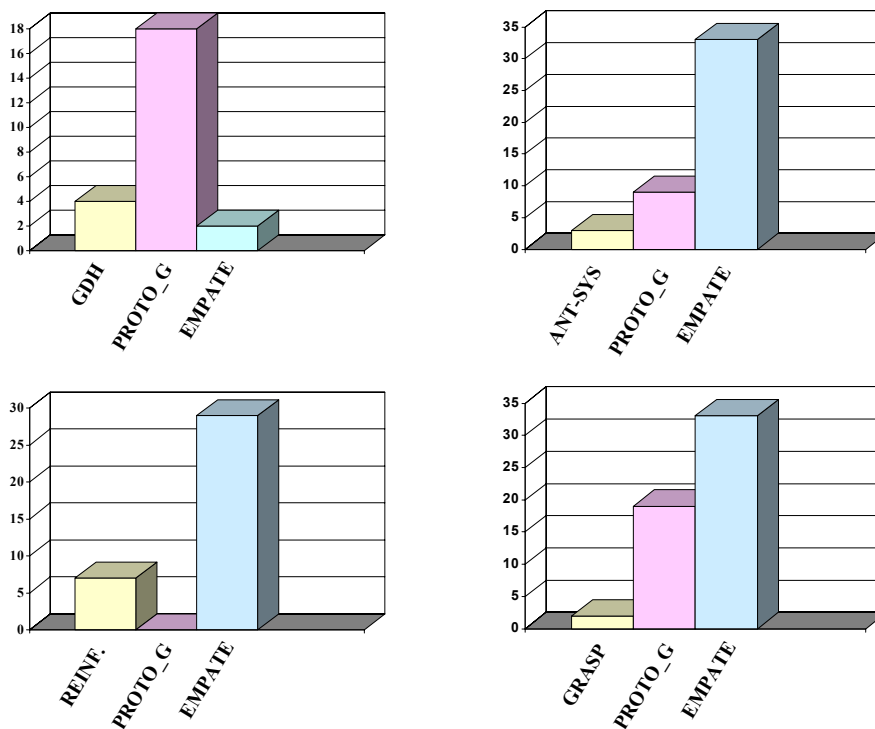


Figura 8 – Comparação de desempenho ProtoG × Algoritmos da literatura

6. Conclusões

A implementação do ATEI mostrou que a técnica descrita no presente trabalho como de “transcrição viral”, é realmente capaz de direcionar a busca evolucionária no sentido da informação embutida na cadeia do vírus que a transporta. As manipulações virais não promovem qualquer perda para a capacidade de convergência do método evolucionário, pelo contrário, podem inclusive acelerar esse processo. A qualidade das informações é um fator fundamental para promover a eficácia do método, conforme é verificado na tabela 4, até mesmo quando apenas partes da configuração ótima, no máximo 15%, são utilizadas na formação das cadeias virais, o algoritmo converge em 100% dos casos para a solução ótima.

A possibilidade prática da promoção da co-evolução gene \times meme foi testada no ATEI-I e no ATEI-IT. Pode-se observar que, mesmo quando o banco de memes possui apenas soluções de baixa qualidade, isto é, soluções que possuem um grande afastamento em relação aos melhores valores relatados na literatura, o algoritmo é capaz de convergir para soluções de boa qualidade. O efeito do algoritmo genético de substrato no sucesso desses algoritmos não é significativo, uma vez que por si só alcança apenas resultados de baixa qualidade, ficando afastado em média 19% da melhor solução da literatura, nos testes realizados no experimento do item 5.1.

O algoritmo ProtoG baseia-se apenas no paradigma de recombinação procariótica, não utilizando operações de cruzamento entre os indivíduos de uma população e mutação. O ProtoG é um algoritmo simples para implementação e requer poucos ajustes de parâmetros. A comparação dos resultados obtidos por esse algoritmo com o Ant-Sys e o Reinforcement Learning mostra a predominância dos empates em relação à qualidade das soluções obtidas. No caso da comparação com o GRASP e GDH, o ProtoG apresenta um desempenho superior devido ao maior número de melhores resultados encontrados. Note-se que esse tipo de algoritmo não só inova no sentido da infiltração da informação, como permite que um processo evolucionário possa ser desenvolvido sem os clássicos operadores e estratégias da Computação Evolucionária. O desempenho qualitativo desse algoritmo equivale aos melhores resultados reportados no estado da arte.

Trabalhos futuros poderão reunir as estratégias apresentadas em outras composições algorítmicas, de modo a tirar maior proveito possível do poder de guiar a busca apresentado pela Transgenética Computacional. Um exemplo dessas alternativas diz respeito à incorporação nos ATEI-IT de etapas de aperfeiçoamento via recombinação procariótica.

Referências Bibliográficas

- (1) Abreu, N.M.M. (1984). Um Estudo Combinatório do Problema Quadrático de Alocação segundo Koopmans e Beckmann. Tese de Doutorado, COPPE, UFRJ.
- (2) Abreu, N.M.M.; Boaventura Netto, P.O.; Querido, T.M. & Gouvêa, E.F. (2002). Classes of Quadratic Assignment Problem Instances: Isomorphism and Difficulty Measure using a Statistical Approach. *Discrete Mathematics*, **124**(1-3), 103-116.
- (3) Ahuja, R.K.; Orlin, J.B. & Tivari, A. (2000). A Greedy Genetic Algorithm for the Quadratic Assignment Problem. *Computers and Operations Research*, **27**(10), 917-934.
- (4) Behe, M. (1996). *Darwin's Black Box – The Biochemical Challenge to Evolution*. The Free Press, New York.

- (5) Bourguignon, A. (1989). *L'Homme Imprévu*. Presses Universitaires de France.
- (6) Bowles, S. & Gintis, H. (1998). The Evolution of Strong Reciprocity. Technical Report, University of Massachusetts, Amherst.
- (7) Boyd, R. & Richerson, P. (1985). *Culture and Theory of Social Evolution*. Sage Publications, London.
- (8) Brooks, D.R. & Wiley, E.O. (1986). *Evolution as Entropy*. University of Chicago Press.
- (9) Campbell, N.A. (1996). *Biology*. The Benjamin / Cumming Publishing Company, NY.
- (10) Çela, E. (1998). *The Quadratic Assignment Problem*. Kluwer Academic Publishers.
- (11) Chattoe, E. (1998). Just How (Un)realistic are Evolutionary Algorithms as Representations of Social Processes? *Journal of Artificial Societies and Social Simulation*, **1**(3), disponível em: <http://www.soc.surrey.ac.uk/JASSS/1/3/2.html>.
- (12) Davies, P. (1998). *The Fifth Miracle*. Orion Productions.
- (13) Dawkins, R. (1976). *The Selfish Gene*. Oxford University Press.
- (14) Dudley, W. (1990). *Genetic Engineering*. Greenhave Press, Inc., San Diego.
- (15) Ehrlich, P.R. & Raven, P.H. (1964). Butterflies and Plants: A Study in Coevolution. *Evolution*, **18**, 586-608.
- (16) Fukuda, T.; Kubota, N. & Shimojima, K. (1999). Virus – Evolutionary Genetic Algorithm and Its Application to Travelling Salesman Problem. **In: Evolutionary Computation, Theory and Applications** [edited by Xin Yao], World Scientific, 235-255.
- (17) Gambardela, L.M.; Taillard, E. & Dorigo, M. (1999). Ant Colonies for the Quadratic Assignment Problems. *Journal of the Operational Research Society*, **50**, 167-176.
- (18) Goldberg, M.C.; Ferreira, R.P.M. & Luna, H.P.L. (1999). Virus Infections for Acceleration of Genetic Algorithms: An Introductory Study. *III Metaheuristics International Congress*, Angra dos Reis, Brasil.
- (19) Goldberg, M.C. & Gouvêa, E.F. (2000). Metáfora Transgenética: Uma Nova Abordagem para a Computação Evolucionária. *Anais do XXXII Simpósio Brasileiro de Pesquisa Operacional*, (Outubro) Viçosa, MG, Brasil.
- (20) Goldberg, M.C. & Gouvêa, E.F. (2001). Transgenética Computacional. Relatório Técnico, Programa de Engenharia de Produção da COPPE/UFRJ.
- (21) Gould, S.J. (1993). Punctuated Equilibrium Comes of Age. *Nature*, **18**, 223-227.
- (22) Gouvêa, E.F. (2001). Transgenética Computacional: Um Estudo Algorítmico. Tese de Doutorado, Programa de Engenharia da Produção, COPPE-UFRJ, Rio de Janeiro.
- (23) Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.
- (24) Houck, C.R.; Joines, J.A.; Kay, M.G. & Wilson, J.R. (1997). Empirical Investigation of Benefits of Partial Lamarckianism. Technical Report V-10, Department of Industrial Engineering, North Carolina State University.

- (25) Kargupta, H. (1998). *Relation Learning In Gene Expression: Intron, Variable Length Representation, And All That*. Computational Science Methods Group, Los Alamos National Laboratory.
- (26) Kubota, N.; Shimojima, K. & Fukuda, T. (1996). The Role of Virus Infection in a Virus-Evolutionary Genetic Algorithm. *Journal of Applied Mathematics and Computer Science*, **6**(3), 415-429.
- (27) Lumsden, C.J. & Wilson, E.O. (1981). *Genes, Mind and Culture*. Harvard University Press.
- (28) Marguli, L. & Sagan, D. (1991). *The Origins of Sex: Three Billions Years of Genetic Recombinations*. New Haven, Yale University Press.
- (29) Maniezzo, V. & Colomi, A. (1999). The Ant System Applied to the Quadratic Assignment Problem. *IEEE Transactions on Knowledge and Data Engineering*, **11**(5), 769-778.
- (30) Merz P. & Freisleben, B. (1999). A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem. Technical Report TR-98-01, Department of Electrical Engineering and Computer Science (FB 12), University of Siegen.
- (31) Miagkikh, V.V. & Punch III, W.F. (1999a). An Approach to Solving Combinatorial Optimization Problems Using a Population of Reinforcement Learning Agents. Technical Report, Genetic Algorithms Research and Application Group (GARAGE), Michigan State University.
- (32) Miagkikh, V.V. & Punch III, W.F. (1999b). Global Search in Combinatorial Optimization Using Reinforcement Learning Algorithms. **In: Proceedings of the 1999 Congress on Evolutionary Computation (CEC99)**, 1, IEEE, 6-9 July, 189-196.
- (33) Miagkikh, V.V. & Punch III, W.F. (1999b). Global Search in Combinatorial Optimization Using Reinforcement Learning Algorithms. **In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99)**, Morgan Kaufmann.
- (34) Moreno, A. (1986). Epistémologie des modes de représentations des systèmes vivants. *Cahiers du CREA*, **9**, 191-221.
- (35) Moritz, A. (1990). *Memetic Science: I – General Introduction*. The Institute for Memetic Research, Panama City, Florida.
- (36) Nugent, C.E.; Vollman, T.E. & Ruml, E.J. (1968). An Experimental Comparison of Techniques for the Assignment of Techniques to Locations. *Operations Research*, **16**, 150-173.
- (37) Plotkin, H.C. (1996). Non-genetic transmission of information: candidate cognitive processes and the evolution of culture. *Behavioural Processes*, **35**, 207-213.
- (38) QAPLIB (2000). <http://www.opt.math.tu-graz.ac.at/qaplib/>
- (39) Radcliffe, N.J. & Surry, P.D. (1994). Formal Memetic Algorithms. *Lecture Notes in Computer Science*, **865**, Evolutionary Computing, Springer Verlag, Berlin, 1-16.
- (40) Resende, M.G.C.; Pardalos, P.M. & Li, Y. (1996). Algorithm 754: Fortran Subroutines for Approximate Solution of Dense Quadratic Assignment Problems Using GRASP. *ACM Transactions on Mathematical Software*, **22**(1), 104 -118.

- (41) Shapiro, J.A. (1988). Bacteria as Multicellular Organisms. *Scientific American*, **258**(6), 82-89.
- (42) Taillard, E.D. & Gambardella, L.M. (1997). Adaptive Memories for the QAP. Technical Report IDSIA-87-97, Lugano, Switzerland.
- (43) Vollmann, T.E. & Buffa, E.S. (1966). The Facilities Layout Problem in Perspective, *Management Science*, **12**.
- (44) Wilson, E.O. (1998). *Consilience* [edited by A.A. Knopfs], New York, USA.
- (45) Wright, R. (2000). *Nonzero*. Pantheon Books, New York.