

PADRONIZAÇÃO DE POLÍTICAS DE SUBSTITUIÇÃO PARA A FERRAMENTA ToGAI

Dallessandro Soares Vianna¹

dalessandro@pq.cnpq.br

Fábio Duncan de Souza²

fduncan@iff.edu.br

Amaro José de Souza Neto²

amaro.neto@gmail.com

Rodrigo Alessandro Gonçalves Lobo²

rodrigoaglobo@gmail.com

Diego Rangel da Silva²

diegords@gmail.com

Marcilene de Fátima Dianin Vianna³

marcilenedianin@gmail.com

1. Universidade Federal Fluminense, Pólo Universitário Rio das Ostras – UFF/PURO
Rua Recife, s/n, Jardim Bela Vista, Rio das Ostras, RJ 28890-000.

2. Universidade Candido Mendes – UCAM-Campos
Rua Anita Pessanha, 100, Parque São Caetano, Campos dos Goytacazes, RJ 28040-320.

3. Universidade Estadual Norte Fluminense Darcy Ribeiro
Avenida Alberto Lamego, 2000, Parque Califórnia, Campos dos Goytacazes, RJ 28013-600.

RESUMO

Uma das metaheurísticas mais utilizadas para solucionar problemas de otimização combinatória são os algoritmos genéticos (AGs). A ferramenta ToGAI foi desenvolvida para facilitar a implementação de AGs, facilitando que neste sejam testados diversas políticas de seleção de indivíduos, evolução da população e critérios de parada com custo operacional praticamente nulo. O objetivo deste trabalho é incrementar a ferramenta ToGAI, permitindo que diferentes políticas de substituição de indivíduos possam ser também avaliadas com pouco esforço. Políticas de substituição de indivíduos representam estratégias de como atualizar a população após a geração de novos indivíduos pelo operador de cruzamento. Foram adicionadas cinco políticas de substituição de indivíduos a ferramenta. Esta nova versão da ToGAI foi avaliada com dois AGs disponíveis na literatura e que foram desenvolvidos com a ferramenta ToGAI. Os experimentos realizados mostraram que a variação da política de substituição de indivíduos pode melhorar sensivelmente os resultados finais do AG.

PALAVRAS CHAVES. Algoritmos Genéticos. Políticas de Substituição de indivíduos. ToGAI.

ABSTRACT

One of the most widely used metaheuristics for solving combinatorial optimization problems are genetic algorithms (GAs). ToGAI was developed to facilitate the implementation of GAs, facilitating that it be tested various policies of individuals selection, of population evolution and stop criteria with almost zero operating cost. The objective is to improve the ToGAI tool, allowing that different individuals replacement policies can also be evaluated with little effort. Individuals replacement policies representing strategies of how to update the population following the generation of new individuals by crossover operator. It was added five individuals replacement policies to ToGAI. This new ToGAI version was evaluated with two GAs found in literature and that were developed using ToGAI. Our experiments showed that the variation of individuals replacement policy can significantly improve the final results of the AG.

KEYWORDS. Genetic algorithms. Individuals replacement policies. ToGAI.

1. INTRODUÇÃO

Problemas de otimização combinatória aparecem freqüentemente em vários setores da economia. Grande parte dos problemas de otimização são intratáveis por natureza ou são grandes o suficiente para tornar inviável o uso de algoritmos exatos [VIANNA *et al.*, 1999]. Atualmente, a principal estratégia para solucionar este tipo de problema é fazer uso de metaheurísticas. Estas são heurísticas direcionadas à otimização global de um problema, pois geram procedimentos de busca em vizinhanças que objetivam evitar a parada prematura em ótimos locais e proporcionar soluções satisfatórias. Alguns exemplos de metaheurísticas são: Busca Tabu, *Simulated Annealing*, GRASP, Algoritmos Genéticos, dentre outras.

Pertencente a classe dos Algoritmos Evolucionários, os Algoritmos Genéticos (AGs) consistem na analogia entre otimização e os mecanismos da genética, combinando os conceitos de adaptação seletiva, troca de material genético e sobrevivência dos indivíduos mais capazes [GOLDBERG, 1989].

Os AGs são iniciados com um conjunto de soluções chamado população. Soluções que formam uma população são utilizadas para, através de cruzamentos, formarem uma nova população. Isto é motivado pela esperança de que a nova população seja melhor do que a primeira. A seleção de indivíduos da população para formar novas gerações é feita de acordo com uma certa adequação. Normalmente quanto melhores forem, mais chances de reprodução estes indivíduos terão. Este procedimento é repetido até atingir um critério de parada.

Uma característica existente nos algoritmos genéticos é o número de variações possíveis que podem ser implementadas de acordo com o problema para o qual se deseja encontrar uma solução [GOLDBERG, 1989]. Porém, uma vez implementado o código de um algoritmo genético, a reutilização deste código para um problema distinto pode ser pequena.

Em [VIANNA & DUNCAN, 2006] é proposta a ferramenta ToGAI (*Tool for Genetic Algorithm Implementation*) que tem por objetivo facilitar e otimizar o processo de desenvolvimento de AGs através da reutilização de código.

Para a ferramenta ToGAI foi definido um modelo de implementação que combina criações e mudanças automáticas de código com programação e personalização do código pelo usuário. Utilizando deste modelo, os usuários que possuem problemas computacionalmente difíceis a serem solucionados, terão como fatores motivadores a facilidade e a agilidade no desenvolvimento de soluções baseadas em algoritmos genéticos. A ferramenta ToGAI permite ao usuário fazer combinações de características e parâmetros na sua aplicação genética, muitas vezes eliminando a necessidade de nova codificação, possibilitando assim que testes de desempenho sejam facilmente realizados com o objetivo de encontrar as melhores políticas para o problema. A ferramenta disponibiliza, na versão atual, nove políticas de seleção, cinco critérios de parada e três políticas de evolução da população que, se combinadas, podem gerar mais de cem versões diferentes de AGs. Estas combinações não necessitam de nova implementação por parte do usuário, facilitando assim, a melhor escolha de configuração para o problema abordado.

O objetivo deste trabalho é incorporar à ferramenta ToGAI a padronização da etapa de Substituição de Indivíduos (*Replacement Phase*). Nesta etapa deve-se decidir se os indivíduos gerados pelo processo de cruzamento serão descartados ou inseridos na população. Neste último caso, é necessário saber que indivíduos da população serão eliminados para que os novos possam assumir os seus lugares. Existem diferentes maneiras de realizar esta escolha, ou seja, diferentes estratégias de Substituição de Indivíduos antigos por novos. Neste trabalho foram implementadas cinco estratégias de Substituição de Indivíduos, todas respeitando o modelo de implementação da ferramenta ToGAI [VIANNA & DUNCAN, 2005].

O restante deste trabalho está organizado da seguinte maneira: na Seção 2 será

apresentado o modelo de implementação que serve como base para a ferramenta ToGAI. Esta ferramenta, bem como as políticas já inseridas nesta, será descrita na Seção 3. Na Seção 4 serão detalhadas as políticas de Substituição de Indivíduos implementadas. Por fim, são apresentadas as conclusões e as referências bibliográficas.

2. MODELO PARA DESENVOLVIMENTO DE ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos, na sua formulação original, possuem o alfabeto binário como forma de representar uma solução em um espaço de busca [GOLDBERG, 1989]. Contudo, tanto o método de representação quanto o alfabeto genético podem variar de problema para problema. Radcliffe (1992) afirma que boas propriedades dos AGs não são bem aproveitadas se a técnica binária de representação for utilizada para determinados problemas. Isto motiva a implementação de AGs utilizando representação não binária para resolver de forma “mais natural” problemas específicos.

Devido às características acima citadas, é importante deixar a definição da estrutura do cromossomo para o usuário. Assim, os algoritmos desenvolvidos não terão sua eficiência prejudicada por alguma restrição do modelo. A Figura 1 apresenta, codificada em C ANSI, a estrutura de um indivíduo da população dentro do modelo de implementação da ferramenta ToGAI. A estrutura *tpChromossome* define a estrutura interna do cromossomo. O corpo desta estrutura é de responsabilidade do usuário, que deve estruturá-lo da maneira que achar mais conveniente para resolução do problema em questão. A estrutura *tpIndivid* é definida pelo modelo. Esta possui um dado do tipo *tpChromossome*, representando o cromossomo que define o indivíduo, e um dado que representa o valor de avaliação deste indivíduo. Este dado é do tipo *tpObj* que é, por padrão, associado ao tipo *float*, mas pode ser modificado pelo usuário.

<pre>/* Estrutura do cromossomo */ typedef struct chromossome { /* Deverá ser incluída aqui a estrutura de dados do cromossomo*/ } tpChromossome;</pre>	<pre>/* Estrutura dos indivíduos: formada pelos cromossomos e pelas suas funções objetivo. */ typedef struct individual { tpChromossome chrom; tpObj obj; } tpIndivid;</pre>
---	--

Figura 1 – Definição, na linguagem C ANSI, das estruturas que formam um indivíduo.

Fonte: Vianna & Duncan (2006).

Algumas etapas de um AG são dependentes da estrutura cromossômica (*tpChromossome*), ou seja, precisam conhecer a estrutura interna do cromossomo para realizar uma consulta e/ou uma modificação. São elas a geração da população inicial e os operadores genéticos de cruzamento (*crossover*) e de mutação. Outras etapas de um AG são independentes, tais como: seleção, evolução da população e critérios de parada.

A ferramenta ToGAI, que baseia-se neste modelo, tem como objetivo diminuir o trabalho do usuário ao desenvolver um algoritmo genético para um determinado problema. Além disso, o modelo torna a implementação do AG organizada (modulada) o suficiente para evitar o “retrabalho” com a modificação de certos métodos. Por exemplo, uma vez implementado um AG, é possível testar diferentes políticas de seleção de indivíduos neste AG sem a necessidade de novas codificações. Isso é possível, pois os métodos de seleção de

indivíduos, de evolução da população e os critérios de parada são dependentes apenas da estrutura *tpIndivid*, sem a necessidade de acessar a estrutura interna do cromossomo (*tpChromosome*).

2.1. Etapas Dependentes da Estrutura Cromossômica

Como mencionado anteriormente, a geração da população inicial, os operadores genéticos de cruzamento e de mutação fazem uso da estrutura cromossômica nas suas implementações. Neste modelo, o usuário é responsável pela codificação destas etapas, porém este deve respeitar os protótipos de função previamente definidos. Como exemplo, o cabeçalho para criação da população inicial está definido a seguir: **void initialPopulation (int iSizePop, tpPopulation pop)**, onde *iSizePop* é um valor numérico que representa o tamanho da população a ser construída e *pop* é a população a ser definida (o tipo *tpPopulation* representa um vetor de elementos do tipo *tpIndivid*).

2.2. Etapas Independentes da Estrutura Cromossômica

Diferentes políticas de seleção, de evolução da população e de critério de parada estão disponíveis no modelo, ficando o usuário responsável apenas pela escolha de qual utilizar.

2.2.1. Seleção

Foram definidas no modelo diversas políticas para que o usuário possa escolher a que se adapta melhor ao problema a ser resolvido. Para isso, todas as políticas obedecem ao cabeçalho definido pelo modelo para os métodos de seleção. Este cabeçalho está definido a seguir: **void Selection (int iSizePop, tpPopulation pop, int iTotSelect, int* pSelected)**, onde *iSizePop* é um valor numérico que representa o tamanho da população *pop*, *iTotSelect* representa o número de indivíduos de *pop* que devem ser selecionados, cujos índices devem ser armazenados no vetor *pSelected*.

As políticas de seleção, disponíveis atualmente, para serem utilizadas no modelo são: Roleta [GOLDBERG, 1989], Torneio [BLICKLE & THIELE, 1995], Torneio Proporcional, Seleção Aleatória [GOLDBERG, 1989], Seleção Determinística, Seleção Estocástica por Resto Sem Reposição, Seleção Estocástica por Resto Com Reposição [GOLDBERG, 1989], Seleção Estocástica Sem Reposição [DE JONG, 1975] e Seleção por Ranking [BLICKLE & THIELE, 1995]. Este conjunto de políticas poderá ser visto com mais detalhes em [VIANNA & SOUZA, 2005].

2.2.2. Evolução da População

Nesta etapa será definida a porcentagem de indivíduos substituídos em cada iteração após o método de cruzamento [GOLDBERG, 1989]. Foram definidas no modelo algumas políticas para que o usuário possa escolher a que se adapta melhor ao problema a ser resolvido. Para isso, as políticas devem obedecer ao cabeçalho definido pelo modelo para os métodos de evolução da população. Este cabeçalho está definido a seguir: **void PopulationEvolution (int iSizePop, tpPopulation pop, tpIndivid *bestIndivid, int iTotSelect,**

int *iTotChildren*), onde *iSizePop* é um valor numérico que representa o tamanho da população *pop*, *bestIndivid* é a estrutura que armazena a melhor solução até o momento, *iTotSelect* representa o número de indivíduos de *pop* que devem ser selecionados para o cruzamento e *iTotChildren* o número de indivíduos gerados por este cruzamento.

As políticas de evolução, disponíveis atualmente, para serem utilizadas no modelo são: SGA [GOLDBERG, 1989], SSGA [SYSWERDA, 1989] e Parâmetro GAP [CARBONO *et al.*, 2005]. Este conjunto de políticas poderá ser visto com mais detalhes em [VIANNA & SOUZA, 2005].

2.2.3. Critério de Parada

Na etapa de avaliação do critério de parada é verificado se o AG deve continuar ou não evoluindo sua população. Foram definidas no modelo algumas políticas para que o usuário possa escolher a que se adapta melhor ao problema a ser resolvido. Para isso, todas as políticas devem obedecer ao cabeçalho definido pelo modelo para os critérios de parada. Este cabeçalho está definido a seguir: **void** *StopCriterion* (**int** *iSizePop*, *tpPopulation pop*, *tpIndivid *bestIndivid*, **int** *iTotSelect*, **int** *iTotChildren*), onde *iSizePop* é um valor numérico que representa o tamanho da população *pop*, *bestIndivid* é a estrutura que armazena a melhor solução até o momento, *iTotSelect* representa o número de indivíduos de *pop* que devem ser selecionados para o cruzamento e *iTotChildren* o número de indivíduos gerados.

Os critérios de parada, disponíveis atualmente, para serem utilizadas no modelo são: Número de Gerações [GOLDBERG, 1989], Melhora da Solução em um Determinado Número de Gerações [SOARES, 1997], Convergência da População, Tempo de Computação [CHOU *et al.*, 1999] e Valor Alvo [RIBEIRO & VIANNA, 2003]. Este conjunto de critérios poderá ser visto com mais detalhes em [VIANNA & SOUZA, 2005].

3. FERRAMENTA ToGAI

Utilizando dos conceitos e do modelo descrito na Seção 2, a ferramenta ToGAI (*Tool for Genetic Algorithms Implementation*) – desenvolvida em JAVA, porém gerando código na linguagem C ANSI – tem como função gerar automaticamente parte do código fonte para a implementação de Algoritmos Genéticos. A parte do código dependente da estrutura cromossômica terá suas declarações, cabeçalhos e protótipos de função gerados automaticamente pela ferramenta, porém o conteúdo das funções e definições de algumas estruturas deverão ser implementadas pelo usuário. No entanto, a parte do código independente da estrutura cromossômica é automaticamente gerada pela ferramenta, que fornece ainda, variações de políticas para serem usadas nas diferentes etapas do algoritmo.

A ferramenta é capaz de gerar uma nova implementação de algoritmo genético, ou pode modificar um algoritmo genético (criado pela ferramenta), alterando as etapas independentes da estrutura cromossômica.

4. ESTRATÉGIAS DE SUBSTITUIÇÃO DE INDIVÍDUOS

Durante a operação de cruzamento são selecionados indivíduos (pais) que são combinados gerando novos indivíduos (filhos). Após esta etapa é necessário escolher dentre todos os indivíduos (pais e filhos) que participaram do processo de cruzamento, quais serão

mantidos na população e quais serão descartados. Para ajudar nesta tomada de decisão, existem diferentes estratégias, denominadas Estratégia de Substituição de Indivíduos, que utilizam critérios variados para decidir quais os indivíduos mais aptos e que podem contribuir melhor na evolução da população.

No decorrer desta seção serão descritas as diferentes políticas de substituição implementadas.

4.1. Substituição dos Piores Pais

Esta estratégia tem como principal objetivo substituir os piores pais pelos filhos gerados após processo de cruzamento de indivíduos [RIBEIRO & VIANNA, 2003; WHITLEY, 1989], priorizando assim a inserção de descendentes na população. Com relação a quais e quantos filhos serão inseridos na população, há certas particularidades:

- quando o número de filhos gerados for menor que o número de pais, os piores pais serão substituídos por todos os filhos;
- quando o número de filhos gerados for igual ao número de pais, será substituído diretamente, todos os pais por todos os filhos; e
- quando o número de filhos gerados for maior que o número de pais, todos os pais serão substituídos pelos melhores filhos.

Na Figura 3 é apresentado o pseudo-código deste procedimento.

Procedimento SUBSTITUI (*pop*, *numPai*, *numFilho*)

Entrada

pop – população corrente;
numPai – número de pais selecionados;
numFilho - número de filhos gerados;

1 - **Início**

2 - **Se** o *numFilho* < *numPai* **então**

3 - Substitua, em *pop*, os *numFilho* piores pais pelos filhos;

4 - **Senão**

5 - **Se** o *numFilho* = *numPai* **então**

6 - Substitua, em *pop*, todos os pais pelos filhos;

7 - **Senão** /**numFilho* > *numPais* */

8 - Substitua, em *pop*, os pais pelos *numPais* melhores filhos;

9 - **Fim-se**

10 - **Fim-se**

11 - **Fim-** SUBSTITUI.

Figura 3. Pseudo-código da estratégia de substituição “substituição dos piores pais”.

4.2. Recombinação Elitista

Esta estratégia tem como principal objetivo manter sempre os melhores pais e os melhores filhos gerados após processo de cruzamento de indivíduos [VIANNA *et al.*, 1999; WAKUNDA & ZELL, 2000], priorizando a permanência dos indivíduos melhores (elitismo). Com relação às quais e quantos filhos serão inseridos na população, há certas particularidades:

- quando o número de filhos for maior ou igual o número de pais, então o número de melhores pais que permanecerão, será a metade do número de pais selecionados. O restante será completado pelos melhores filhos; e
- quando o número de filhos for menor que o número de pais, então o número de melhores filhos que serão inseridos, será a metade do número de filhos gerados. O restante será preenchido pelos melhores pais.

Na Figura 4 é apresentado o pseudo-código deste procedimento.

Procedimento SUBSTITUI (*pop*, *numPai*, *numFilho*)

Entrada

pop – população corrente;
numPai – número de pais selecionados;
numFilho - número de filhos gerados;

1- **Início**

2 - **Se** *numFilho* \geq *numPai* **então**

3 - *numMelhPai* = metade de *numPai*;

4 - *numMelhFilho* = *numPai* – *numMelhPai*;

5 - **Senão** /* *numFilho* < *numPai* */

6 - *numMelhFilho* = metade de *numFilho*;

7 - *numMelhPai* = *numPai* – *numMelhFilho*;

8 - **Fim-Se**

9 - Marcar os *numMelhPai* melhores pais;

10 - Substitua, em *pop*, os pais não marcados como melhores pais pelos
 numMelhFilho melhores filhos;

11 - **Fim-** SUBSTITUI.

Figura 4. Pseudo-código da estratégia de substituição “recombinação elitista”.

4.3. Ordem de Chegada – FIFO (*first in first out*)

Esta estratégia de substituição tem como principal objetivo substituir os indivíduos mais velhos pelos filhos gerados após o processo de cruzamento de indivíduos [GHOSH *et al.*, 1998; VAVAK & FOGARTY, 1996]. Ela propõe a inserção direta de todos os filhos na população, substituindo os indivíduos com idade mais avançada pelos filhos, não importando o número de pais selecionados.

É importante destacar que para que este algoritmo seja utilizado na ferramenta ToGAI, foi necessário alterar a estrutura do indivíduo, adicionado a esta um atributo para o controle da idade do indivíduo.

Na Figura 5 é apresentado o pseudo-código deste procedimento.

Procedimento SUBSTITUI (*pop*, *numPai*, *numFilho*)**Entrada**

pop – população corrente;
numPai – número de pais selecionados;
numFilho - número de filhos gerados;

1 - **Início**

2 - Substitua, em *pop*, os *numFilho* indivíduos mais velhos pelos filhos;

3 - **Fim-** SUBSTITUI.

Figura 5. Pseudo-código da estratégia de substituição “ordem de chegada”.

4.4. Ordem de Chegada Com Torneio (Conservação)

Esta estratégia combina a estratégia de substituição Ordem de Chegada (Subseção 4.3) com o operador de seleção Torneio [VAVAK, 1997]. Nela, para cada filho a ser inserido na população, são selecionados o indivíduo mais velho da população e outro indivíduo qualquer escolhido aleatoriamente. Então, é realizado um torneio entre os dois indivíduos selecionados e o vencedor do torneio é escolhido para permanecer na população. O outro indivíduo é substituído pelo filho. Deste modo o elitismo é implicitamente assegurado.

Na Figura 6 é apresentado o pseudo-código deste procedimento.

Procedimento SUBSTITUI (*pop*, *numPai*, *numFilho*)**Entrada**

pop – população corrente;
numPai – número de pais selecionados;
numFilho - número de filhos gerados;

1 - **Início**

2 - Para cada filho gerado

3 - *i1* = indivíduo mais velho da população

4 - *i2* = indivíduo escolhido aleatoriamente

5 - Entre *i1* e *i2*, selecione o pior indivíduo

7 - Substitua em *pop*, o pior indivíduo pelo filho corrente.

6 - **Fim-para**7 - **Fim-** SUBSTITUI.

Figura 6. Pseudo-código da estratégia de substituição “ordem de chegada com torneio”.

4.5. Substituição dos Piores Indivíduos

Esta estratégia é usada como padrão para resolver diversos problemas de Algoritmos Genéticos [VIANNA & SOUZA, 2006]. Ela apenas substitui os piores indivíduos da população pelos novos indivíduos (filhos) que são gerados. A sua diferença para a política Substituição dos Piores Pais (Subseção 4.1) é que o seu espaço de busca por piores indivíduos é toda população e não somente entre os indivíduos pais selecionados para cruzamento.

Na Figura 7 é apresentado o pseudo-código deste procedimento.

Procedimento SUBSTITUI (*pop*, *numPai*, *numFilho*)**Entrada**

pop – população corrente;
numPai – número de pais selecionados;
numFilho - número de filhos gerados;

1 - **Início**

2 - Substitua em *pop*, os *numFilho* indivíduos piores pelos filhos;

3 - **Fim**- SUBSTITUI.

Figura 7. Pseudo-código da estratégia de substituição “ordem de chegada com torneio”.

5. EXPERIMENTOS COMPUTACIONAIS

No intuito de avaliar a nova versão da ferramenta ToGAI, após a inclusão das políticas de substituição de indivíduos, dois AGs elaborados pela ferramenta e já publicados foram reconduzidos a ferramenta ToGAI para uma avaliação sobre a política de substituição de indivíduos que usavam, uma vez que agora a ferramenta permite que novas políticas de substituição sejam testadas com custo operacional praticamente nulo.

Os dois AGs avaliados foram:

- O algoritmo **AG+PR+LS** que é um AG híbrido desenvolvido para o problema da filogenia [RIBEIRO & VIANNA, 2003; RIBEIRO & VIANNA, 2009]. Este AG utiliza um operador de cruzamento baseado na técnica de reconexão por caminhos e, esporadicamente, uma busca local é aplicada aos indivíduos da população. Este algoritmo foi dentre seis AGs implementados para o problema em questão o que apresentou os melhores resultados. Ele foi publicado no International Journal in Operational Research [RIBEIRO & VIANNA, 2009].
- O algoritmo **AG1** que é um AG desenvolvido para o problema de roteamento de veículos com coleta e entrega simultâneas e janela de tempo [LOPES & VIANNA, 2009]. Este AG também utiliza reconexão por caminhos como operador de cruzamento. Dentre três heurísticas (dois AGs e um GRASP) desenvolvidas para o problema em questão, este AG foi o que apresentou os melhores resultados. Ele foi publicado no XVI Simpósio de Engenharia de Produção [LOPES & VIANNA, 2009].

Ambos os AGs acima descritos utilizam, em sua versão original, como política de substituição de indivíduos a política “Substituição dos Piores Pais” descrita na Subseção 4.1.

Nos testes realizados com a nova versão da ferramenta ToGAI, foi realizado o mesmo conjunto de experimentos realizados com as versões originais. No entanto, para cada um dos AGs (**AG+PR+LS** e **AG1**) foram desenvolvidas cinco versões, uma com cada política de substituição de indivíduos descrita na Seção 4.

Quando o algoritmo **AG+PR+LS** foi analisado, a política “Substituição dos Piores Pais”, originalmente utilizada, foi uma das duas políticas que se adaptaram melhor ao algoritmo. No entanto, a política “Substituição dos Piores Indivíduos”, descrita na Subseção 4.5, foi um sutilmente superior.

Quando o algoritmo **AG1** foi avaliado, foi verificado que a política “Substituição dos

Piores Pais”, originalmente utilizada, não está entre as políticas que se adaptaram melhor ao algoritmo. As políticas “Substituição dos Piores Indivíduos” e “Ordem de Chegada Com Torneio”, descritas respectivamente nas Subseções 4.5 e 4.4, se mostraram melhor adaptadas ao problema, melhorando sensivelmente os resultados finais do algoritmo. A política “Ordem de Chegada Com Torneio” foi a que se mostrou melhor adaptada ao problema nos experimentos.

5. CONCLUSÃO

A ferramenta ToGAI (*Tool for Genetic Algorithm mplementation*) tem por objetivo facilitar e otimizar o processo de desenvolvimento de Algoritmos Genéticos (AGs) através da reutilização de código. A versão atual da ferramenta já tem padronizadas – isto é, implementadas e disponíveis para o usuário – as políticas de seleção, as estratégias de evolução da população e critérios de parada.

O presente trabalho padroniza a operação de Substituição de Indivíduos, respeitando o modelo de implementação proposto pela ferramenta ToGAI [VIANNA & SOUZA, 2005]. Foram implementadas 5 entre as 6 políticas de substituição encontradas na literatura. A política “Aproximação Evolutiva” [DE JONG, 1975] não foi implementada pois, para isto, era necessário conhecer a estrutura interna do cromossomo para que uma estimativa de aproximação fosse calculada, fato que vai contra os conceitos fundamentais da ferramenta ToGAI.

Foram realizados experimentos com a nova versão da ferramenta ToGAI (após a inserção das políticas de substituição de indivíduos) com dois AGs disponíveis na literatura científica e que foram desenvolvidos, em sua versão original, com a ferramenta ToGAI. Estes experimentos mostraram que a mudança da política de substituição de indivíduos pode melhorar sensivelmente os resultados, o que demonstra a utilidade da nova versão da ferramenta ToGAI.

Incorporando as estratégias de substituição, aqui implementadas, à ferramenta ToGAI, esta se tornou ainda mais versátil e teve um reuso de código ainda maior, o que facilita ainda mais o desenvolvimento de AGs a partir desta ferramenta.

AGRADECIMENTOS

Este trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), pelo Parque de Alta Tecnologia do Norte Fluminense (TECNORTE) e pela Fundação Estadual do Norte Fluminense (FENORTE).

REFERÊNCIAS

- BLICKLE, T.; THIELE, L. (1995): “A COMPARISION OF SELECTION SCHEMES USED IN GENETIC ALGORITHMS”. TIK-Report nr 11 version 2. Swiss Federal Institute of Technology (ETH), Zurich, December.
- CARBONO, A. J. J.; MENEZES, I. F. M.; MARTHA, L. F. (2005): “MOORING PATTERN OPTIMIZATION USING GENETIC ALGORITHMS”. Em: Proc. of the WCSMO6, Rio de Janeiro, 182.1-182.9.

- CHOU, H.; PREMKUMAR, G.; CHU, C. H. (1999): "GENETIC ALGORITHMS AND NETWORK DESIGN: AN ANALYSIS OF FACTORS INFLUENCING GA'S PERFORMANCE". From the e-Business Research Center Working Paper, 1-38.
- DE JONG, K. A. (1975): "ANALYSIS OF THE BEHAVIOR OF A CLASS OF GENETIC ADAPTIVE SYSTEMS". PhD. thesis, Dep.Computer and Communication Sciences, Univ. Michigan, Ann Arbor.
- GHOSH, A.; TSUTSUI, S.; TANAKA, H. (1998): "FUNCTION OPTIMIZATION IN NONSTATIONARY ENVIRONMENT USING STEADY STATE GENETIC ALGORITHMS WITH AGING OF INDIVIDUALS". Proc. of the IEEE International Conference on Evolutionary Computation, (IEEE Press, New York) 666-671.
- GOLDBERG, D. E. (1989): "GENETIC ALGORITHMS IN SEARCH, OPTIMIZATION AND MACHINE LEARNING". Reading MA: Addison Wesley.
- LOPES, L. P. V.; VIANNA, D. S. (2009): UMA HEURÍSTICA GRASP+VND E UM ALGORITMO GENÉTICO COM RECONEXÃO POR CAMINHOS PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA E ENTREGA SIMULTÂNEA E JANELA DE TEMPO. Anais do XVI Simpósio de Engenharia de Produção, 25-36.
- RADCLIFFE, N. J. (1992): "NONLINEAR GENETIC REPRESENTATIONS". Parallel Problem Solving from Nature 2, R.M'anner and B. Manderick (Ed.), 259–268.
- RIBEIRO, C. C.; VIANNA, D. S. (2003): "A GENETIC ALGORITHM FOR THE PHYLOGENY PROBLEM USING AN OPTIMIZED CROSSOVER STRATEGY BASED ON PATH-RELINKING", Revista Tecnologia da Informação - RTInfo, v.3, 67-70.
- RIBEIRO, C. C.; VIANNA, D. S. (2009): A HYBRID GENETIC ALGORITHM FOR THE PHYLOGENY PROBLEM USING PATH-RELINKING AS A PROGRESSIVE CROSSOVER STRATEGY. International Transactions in Operational Research, v.16, 641-657.
- SOARES, G. L. (1997): "ALGORITMO GENÉTICO: ESTUDO, NOVAS TÉCNICAS E APLICAÇÕES". Dissertação de Mestrado, UFMG, Belo Horizonte.
- SYSWERDA, G. (1989): "UNIFORM CROSSOVER IN GENETIC ALGORITHMS". In J. D. Schaffer (Ed.), Proceedings of the Third International Conference on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, 2-9.
- VAVAK, F.; FOGARTY, T. C. (1996): "COMPARISON OF STEADY STATE AND GENERATIONAL GENETIC ALGORITHMS FOR USE IN NONSTATIONARY ENVIRONMENTS". Proc. Of the 1996 IEEE International Conference on Evolutionary Computation, (IEEE Press, New York) 192- 195.
- VAVAK, F. (1997): "GENETIC ALGORITHM BASED SELF-ADAPTIVE TECHNIQUES FOR DIRECT LOAD BALANCING IN NONSTATIONARY ENVIRONMENTS". PhD thesis, University of the West of England.
- VIANNA, D. S.; OCHI, L. S.; DRUMMOND, L. M. A. (1999): "A PARALLEL HYBRID EVOLUTIONARY METAHEURISTIC FOR THE PERIOD VEHICLE ROUTING PROBLEM WITH HETEROGENEOUS FLEET". Lecture Notes in Computer Science, 1388:216-225.
- VIANNA, D. S.; SOUZA, F. D. (2006): "ToGAI: TOOL FOR GENETIC ALGORITHM IMPLEMENTATION". Congreso Latino-Iberoamericano de Investigación Operativa. 153-158.
- VIANNA, D. S.; SOUZA, F. D. (2006): "UM MODELO DE IMPLEMENTAÇÃO PARA O DESENVOLVIMENTO DE ALGORITMOS GENÉTICOS". Relatório Técnico.

Universidade Candido Mendes – Campos. Submetido à Revista Produção On Line.

WAKUNDA, J.; ZELL, A. (2000): “MEDIAN-SELECTION FOR PARALLEL STEADY-STATE EVOLUTION STRATEGIES”. Proc. Parallel Problem Solving from Nature — PPSN VI, Lecture Notes in Computer Science 1917, Springer, 405-414.

WHITLEY, D. (1989): “THE GENITOR ALGORITHM AND SELECTION PRESSURE: WHY RANK-BASED ALLOCATION OF REPRODUCTIVE TRIALS IS BEST”. Proc. Of the Third Int. Conf. on Genetic Algorithms, J. David Schaffer (Ed.), (Morgan Kaufmann Publishers, San Mateo) 116-121.