

# O Algoritmo Evolutivo como método projetual

The Evolutionary Algorithm as design method

## Jarryer Andrade De Martino

Universidade Estadual de Campinas, Brasil  
jarryer\_martino@hotmail.com

## Gabriela Celani

Universidade Estadual de Campinas, Brasil  
celani@fec.unicamp.br

### ABSTRACT

The evolutionary algorithm contributes significantly to the search for solutions to difficult problems of understanding and problems that have more than one solution, and all of which are satisfactory. This method is used by some architects as an optimization technique for design process; furthermore it is a possibility to obtain solutions which may not have been imagined by the designer. The aim of this article is to introduce the main concepts of evolutionary algorithm and to explain its structure. The article ends with example of the application of evolutionary algorithms in solving design problems.

**KEYWORDS:** algoritmo evolutivo; método de projeto; diversidade.

### Introdução

O objetivo deste artigo é apresentar a utilização dos algoritmos evolutivos como um possível método de busca de soluções no processo de projeto. Para isso, é de grande importância a conceituação e a definição de seu vocabulário, fundamentado na área biológica, e o esclarecimento da estrutura que compõe os algoritmos evolutivos. Este repertório é fundamental para a compreensão dos motivos que conduziram à utilização destes métodos em propostas de metodologia projetual embasadas na teoria dos problemas perniciosos (*Wicked Problems*). O artigo é finalizado com exemplos da aplicação destes algoritmos evolutivos na solução de problemas de layout e na otimização de formas e reforços de estruturas.

### Os Algoritmos Evolutivos (AE)

Os algoritmos evolutivos correspondem a um conjunto de métodos de busca, otimização, aprendizagem e modelagem de dados inspirados na evolução biológica (BENTLEY, 1999). São algoritmos inspirados na teoria evolucionista de Darwin apresentando como principais características a reprodução com herança genética, a

variação aleatória em uma população de indivíduos e a aplicação de seleção natural para gerar as próximas gerações. Esses métodos foram desenvolvidos por cientistas da computação nas décadas de 1950 e 1960, originando a área de pesquisa denominada de Computação Evolutiva. Os principais métodos que integram os algoritmos evolutivos são: Algoritmo Genético (AG), Estratégia Evolutiva (EE) e Programação Evolutiva (PE). O AG foi criado por John Holland na década de 1960 e foi formalizado em 1975 através da publicação do livro *Adaptation in Natural and Artificial Systems*. O objetivo inicial de Holland foi desenvolver mecanismos de adaptação natural que pudessem ser implementados no computador para estudar o fenômeno de adaptação que ocorre na natureza (MITCHEL, 1999). O método de EE (criado na década de 1960 e complementado em 1973 por Rechenberg e Schwefel) e de PE (formalizado por Fogel, Owens e Walsh em 1966) foi desenvolvido com o objetivo de solucionar problemas específicos. Esses algoritmos possuem uma estrutura básica constituída por populações de indivíduos (representando as possíveis soluções de um problema), operadores de diversidade (responsáveis pela diversificação dos indivíduos) e funções de aptidão dos indivíduos (que verificam o grau de satisfação das

soluções obtidas). O que diferencia um método do outro são as representações das soluções (cadeias binárias, vetores de números ou árvores), os operadores de diversidade utilizados em cada processo, os operadores de seleção e a maneira como são geradas as populações.

As representações das soluções estão relacionadas com a escolha de um sistema para a codificação do problema, como exemplo temos a codificação do ser humano em um DNA (Fig. 1), que apresenta moléculas com as instruções genéticas coordenando o desenvolvimento e funcionamento dos seres vivos. A escolha de uma codificação não adequada pode acarretar em resultados insatisfatórios, pois ao serem aplicados os operadores de diversidade (recombinação e mutação) serão gerados resultados infactíveis. A codificação binária é a mais utilizada no AG devido à simplicidade e por ter sido proposta inicialmente por John Holland. Essa codificação baseia-se na representação cromossomial composta por genes, sendo que cada gene corresponde a menor unidade de informação genética definindo um atributo ou característica. O conjunto de genes em sequência configura um cromossomo ou indivíduo (denominação mais comum no AE). A estrutura composta pelos genes define um genótipo, e o resultado da interação desse conteúdo genético com o ambiente manifestando-se no comportamento, fisiologia e morfologia do indivíduo corresponde ao fenótipo.

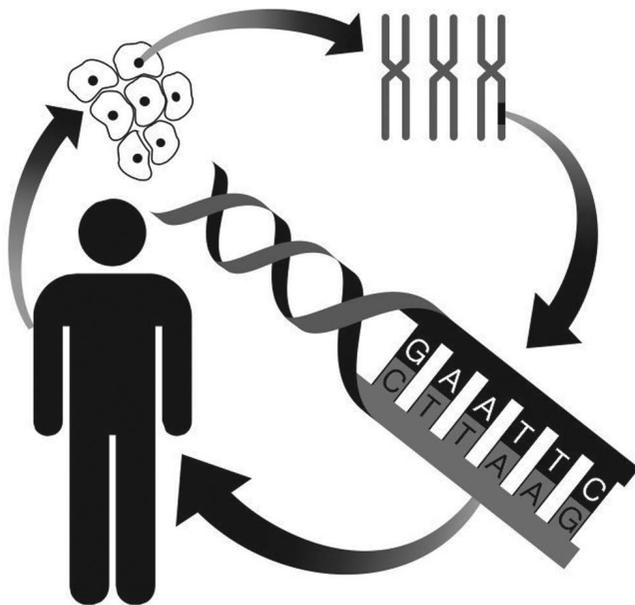


Fig. 1. O ser humano como exemplo do resultado interpretativo de um código, o DNA. Fonte: elaboração própria.

Os operadores de diversidade são fundamentais e podem ocorrer através da recombinação e a mutação durante o cruzamento dos “cromossomos pais”. A recombinação é a troca de partes entre dois ou mais “cromossomos pais” e a mutação é uma alteração aleatória aplicada em um deles. Esse processo dará origem aos “cromossomos

filhos” que serão avaliados conforme a sua qualidade através de uma função de avaliação (*fitness function*) que contem todas as restrições necessárias do problema a ser resolvido, verificando o atendimento às necessidades do problema proposto e aplicando penalidades aos que desrespeitarem as restrições (LINDEN, 2008). A partir dessa avaliação é possível atribuir um valor de *fitness* para cada indivíduo que possibilitará aos operadores de seleção identificar, classificar e selecionar os melhores indivíduos para compor a nova população. Esses operadores de seleção envolvem algoritmos que utilizam técnicas determinísticas ou probabilísticas como os da roleta, do torneio, do ranking, o bi classista ou o elitista. Cada ciclo desse representado na Fig. 2 corresponde a uma iteração que será executada repetidas vezes até a obtenção de indivíduos com características desejadas ou por um determinado período de tempo.

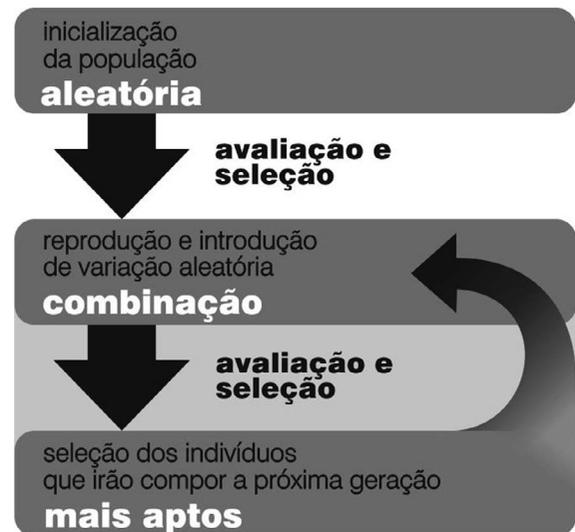


Fig. 2. Estrutura típica de um algoritmo evolutivo considerando uma iteração. Fonte: elaboração própria.

A utilização desses algoritmos ocorre em situações onde a busca por soluções encontra-se em um espaço de soluções muito grande. Alguns exemplos dessas situações são os problemas de difícil entendimento e os problemas que possuem mais de uma solução, dentre as quais não existe uma melhor que as demais. É importante destacar que o AE busca a melhor solução de uma maneira aleatória-guiada (LINDEN, 2008), diferentemente do método de busca gradiente (*hill climbing*), que buscará encontrar o máximo de uma função a partir da sua derivada correndo o risco de ficar preso em um máximo local. O algoritmo evolutivo consegue explorar diferentes regiões e pontos do espaço de soluções devido aos operadores de diversidade e seleção, o que contribui para encontrar o máximo global, por isso é denominada de uma técnica de otimização global. Outra questão a ser considerada no AE é a possibilidade de obter uma geração pior que as anteriores, pois a combinação de boas soluções não necessariamente resultará em uma população

melhor, por isso a necessidade de uma boa definição dos operadores de seleção, que irão contribuir para contornar essa situação através dos valores de *fitness*. O processo do AE em geral apresenta a tendência de gerar boas soluções, e a utilização desse método visa justamente encontrar as melhores diante da diversidade gerada durante as iterações.

## Os problemas de difícil solução e os AG

Os processos projetuais até a década de 1960 não deixavam evidentes os procedimentos adotados na solução dos problemas, ocorrendo uma busca heurística por alternativas. Desta forma, via-se a necessidade de uma sistematização que permitisse compreender melhor os processos mentais adotados pelos arquitetos e a possibilidade de fazer análise em cada fase do processo de projeto, o que possibilitaria exercer um maior controle sobre todo o processo projetual e identificar as possíveis incongruências (JONES, 1992).

Essa questão metodológica torna-se evidente quando Jones (1992) compara os projetistas à caixa preta e à caixa de vidro. No primeiro caso considera o processo como um sistema fechado e complexo, apresentando uma estrutura interna desconhecida e não analisada, sendo avaliados apenas os estímulos de entrada e a resposta de saída. No segundo caso, a caixa de vidro, o pensamento é externalizado e inteiramente explicado. Os objetivos, as variáveis e os critérios são fixados com antecedência, e a avaliação segue uma estrutura lógica, permitindo justificar todas as tomadas de decisões (JONES, 1992, p.50). No mesmo período de 1960, Horst Rittel formulou a abordagem dos problemas perniciosos (*wicked problems*). São problemas que não possuem uma formulação definida, apresentam soluções melhores e piores, e permitem sempre mais do que uma possibilidade de explicação. Isso fez com que ele propusesse uma abordagem composta por duas fases, a definição do problema e a solução do problema (BUCHANAN, 1992). A primeira é uma sequência analítica em que o projetista determina todos os possíveis elementos do problema e especifica o que é requerido para uma solução projetual ideal. E a segunda fase é uma sequência sintética em que os diferentes requerimentos para a solução do problema são combinados e balanceados entre si, criando estratégias para a obtenção da solução ideal. Nessa linha de raciocínio é fundamental referenciar Mitchell (1975), que também discutiu o projeto de arquitetura como um processo de solução de problemas. A partir de um enfoque na própria definição do problema, a solução deste será um comportamento intencional orientado por metas, ou seja, um sistema operativo.

Assim, ao adotar um método projetual orientado por metas, ou seja, um sistema operativo, este processo assume uma estruturação algorítmica possível de integrá-lo com os métodos de algoritmos evolutivos e

que poderiam ser implementados computacionalmente. Diante desse cenário na década de 1960, os arquitetos encontraram uma possibilidade de aplicação dessas teorias, vindas do campo da inteligência artificial, para resolver problemas de difícil solução em arquitetura. O desenvolvimento computacional e a implementação daqueles métodos evolutivos potencializaram a utilização do computador no processo projetual, contribuindo com a execução dos algoritmos e tomadas de decisões durante o processo, fazendo dessa ferramenta não apenas uma extensão da mente humana, mas um parceiro no processo de projeto com aptidões específicas (TERZIDIS, 2006). Os arquitetos e os projetistas de interiores visualizaram a possibilidade de utilização desses novos métodos na área de organização de layouts de edifícios comerciais, universidades, hospitais e lojas de departamento. Além da utilização na solução de problemas de diferentes escalas, desde a distribuição de atividades na cidade até a alocação de equipamentos, pessoas ou grupos em um simples pavimento de um edifício.

Assim, Liggett (2000) apresenta três principais caminhos para as técnicas de solução de problemas desse tipo. O primeiro envolve a otimização de um único critério de função, partindo de uma solução inicial que sofrerá constantemente alterações até chegar a uma organização ideal. O segundo trabalha com a teoria dos grafos, focando a relação de adjacência e proximidades das atividades em um modelo topológico. O terceiro preocupa-se em achar a satisfação para um conjunto diversificado de restrições ou relações, definindo um modelo topológico e geométrico da solução. É importante salientar que a utilização dessas técnicas objetivavam a contribuição na fase de concepção projetual, momento em que as diferentes possibilidades surgem para a solução de um problema e que através da utilização dos algoritmos evolutivos era possível identificar as que melhor atendiam às necessidades. Além da possibilidade de gerar uma maior diversidade talvez não imaginada pelo projetista, excluindo a possibilidade de uma solução tendenciosa de um problema. Assim, a busca pelas soluções ideais não seguem às cegas, uma vez que essas técnicas são capazes de gerar, comparar e avaliar as diversas possibilidades.

Um dos exemplos de aplicação do algoritmo genético é a utilização para a organização espacial de uma planta residencial apresentado no artigo de Üngör *et al* (2011) e intitulado *A Mass Customization Oriented Housing Design Model*. O objetivo foi propor um software que pudesse envolver o usuário no processo de projeto utilizando o método de AG para a obtenção de resultados. O usuário inicia a interface definindo os cômodos e as metragens quadradas que irão compor a unidade de habitação. A partir dessas informações o software utilizará o algoritmo evolutivo para definir as posições dos cômodos e adequá-los formalmente à melhor

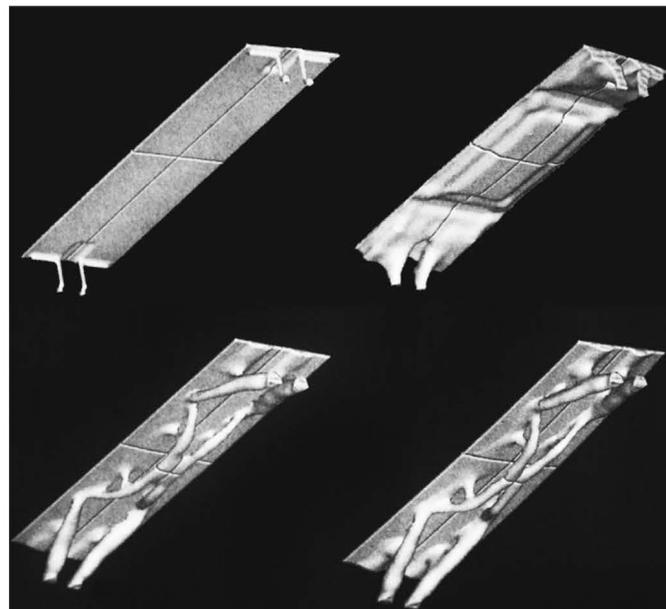
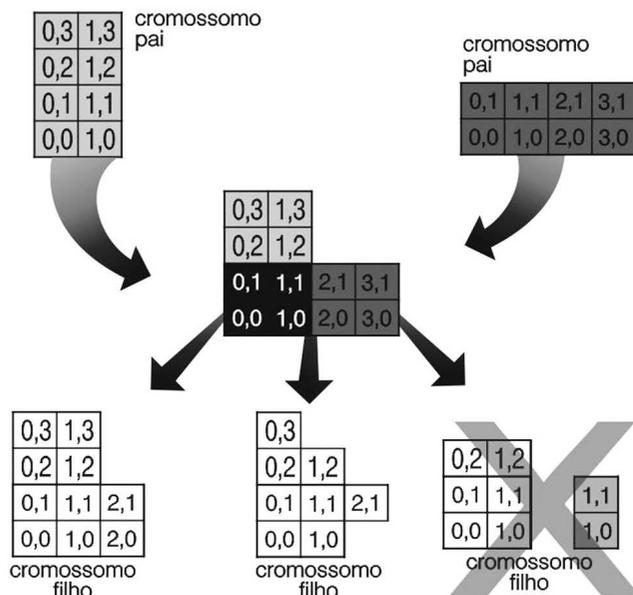


Fig. 3. Esquerda: exemplo de cruzamento. Fonte: Üngör, 2011. Direita: Processo de OEE. Fonte: Meredith, 2008.

metragem quadrada solicitada. Os resultados são salvos em um banco de dados e as melhores soluções obtidas apresentadas visualmente (conforme a função objetivo - *fitness function*) para que o usuário possa escolher. A estrutura desse algoritmo está embasada em uma malha constituída por vários quadrados unitários. Cada quadrado unitário corresponde a uma posição no espaço definindo um gene, de maneira que o conjunto deles representará a área de um cômodo, como pode ser visto à esquerda na Fig. 3.

Algumas condições foram definidas no algoritmo para evitar a criação de cromossomos com geometrias inválidas, como a sobreposição ou formação de espaços vazios fragmentando a área dos cômodos. A partir dessas condições inicia-se o processo adicionando de forma aleatória ao quadrado unitário inicial os quadrados unitários vizinhos. Esse procedimento ocorre até que se obtenha a área informada na interface do software. Esses cromossomos são cruzados e avaliados conforme a *fitness function*, que neste caso é uma função matemática que estabelece uma razão entre os números de lados do polígono formado e o tamanho da área. Quanto menor o valor da taxa, significa que suas formas são mais lineares do que as dos outros, assumindo assim uma posição mais elevada no ranking no processo de seleção. Este exemplo possibilita entender de maneira simples e direta como o AG pode ser implementado, definindo o processo de codificação (quadrado unitário como um gene e o cromossomo como o cômodo), o cruzamento e a seleção através da *fitness function*. Deve-se considerar que essa proposta tem como objetivo o de inserir o usuário não especializado na fase inicial de desenvolvimento projetual, servindo como ferramenta no processo de discussão entre o arquiteto e o usuário. Portanto, o resultado obtido não

corresponde ao projeto finalizado, mas sim, a definição das áreas, ou seja, uma setorização que possibilitará dar sequencia nas próximas etapas de desenvolvimento projetual.

Outro exemplo é a otimização estrutural evolucionária (OEE) que corresponde a um dos métodos de otimização estrutural topológica muito utilizada em vários campos da engenharia. Esse método se baseia na eliminação lenta e sucessiva de elementos ineficientes em uma estrutura, assim como a adição de material para o reforço em áreas onde ocorre a maior exigência estrutural. Isso irá proporcionar estruturas com máxima performance e com o mínimo de peso e com máxima resistência mecânica ou rigidez (SILVA, 2001). Exemplos de aplicação desse método é o projeto de uma nova estação para Florença proposto pelo arquiteto Arata Isozaki em 2003. A proposta conta com uma estrutura que possui o máximo de eficiência e o mínimo uso de material. O processo ocorreu de forma interativa contando com um trabalho conjunto entre arquitetos e engenheiros auxiliados pelo computador e pelo o método de otimização estrutural evolucionária. Através do AE foi possível realizar os cálculos estruturais que possibilitaram a geração gradativa de gráficos tridimensionais em diferentes estágios, como pode ser visto a direita na Fig.3, o que permitiu as análises formais e estruturais por aqueles profissionais até que se chegasse a um consenso estético e funcional para o

### Considerações finais

Encontrar as soluções que atendam satisfatoriamente e simultaneamente a todas as variáveis que norteiam um projeto é um problema de difícil solução. Por isso a adoção de métodos que utilizam o AE contribui consideravelmente para a busca dos melhores

resultados através da exploração de um amplo espaço de soluções. Deve-se considerar que a utilização desses métodos contribui para a obtenção de soluções não imaginadas pelo projetista e possibilitam a otimização dos resultados enriquecendo o processo projetual. Atualmente, diante do desenvolvimento das ferramentas digitais, torna-se cada vez mais viável e simples a implementação computacional dos AE justificando a retomada dessa técnica que já era explorada na década de 1960 pelos arquitetos e cientistas da computação. O próximo desafio desta pesquisa consistirá em desenvolver métodos de emprego destas técnicas que sejam relativamente fáceis de utilizar e realmente eficazes nas etapas iniciais do projeto, mesmo por estudantes de arquitetura sem qualquer experiência em programação, por meio do uso de implementações recentes, como por exemplo, a ferramenta Galápagos no plug-in Grasshopper, para Rhinoceros.

## Referências

BENTLEY, P. J. 1999. *Evolutionary Design by Computers*. San Francisco: Morgan Kaufmann Publishers Inc.

BUCHANAN, R. 1992. *Wicked problems in design thinking*. Acesso em março de 2012, de <http://www.jstor.org/stable/1511637>.

JONES, J. C. 1992. *Design methods*. New York: Wiley.

LIGGETT, R. 2000. *Automated facilities layout: past, present and future*. Acesso em abril de 2011 de [http://dx.doi.org/10.1016/S0926-5805\(99\)00005-9](http://dx.doi.org/10.1016/S0926-5805(99)00005-9).

LINDEN, R. 2008. *Algoritmos genéticos*. Rio de Janeiro: Brasport,.

MEREDITH, M. 2008. *From control to design*. Barcelona: Actar.

MITCHELL, M. 1996. *An Introduction to Genetic Algorithms*. Cambridge: A Bradford Book The MIT Press.

MITCHELL, W. J. 1975. *The theoretical foundation of computer-aided architectural design*. Londres: Pion Ltd.

SILVA, J. A. B. 2001. *Investigação de um método evolucionário de otimização estrutural*. Campinas: [s.n.].

TERZIDIS, K. 2006. *Algorithmic architecture*. Amsterdam: Elsevier/Architectural.

ÜNGÖR, Ö. et al. 2011. *A Mass Customization Oriented Housing Design model Based on Genetic Algorithm*. Ljubljana: eCAADe.