



Ministério da Educação – Brasil
Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM
Minas Gerais – Brasil
Revista Vozes dos Vales: Publicações Acadêmicas
Reg.: 120.2.095 – 2011 – UFVJM
ISSN: 2238-6424
QUALIS/CAPES – LATINDEX
Nº. 11 – Ano VI – 05/2017
<http://www.ufvjm.edu.br/vozes>

Algoritmos Genéticos aplicados à otimização de áreas

Prof^a. Dr^a. Lorena Sophia Campos de Oliveira
Doutora em Engenharia Elétrica - UFES – Brasil
Docente do Instituto Ciência, Engenharia e Tecnologia
Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM – Brasil
<http://lattes.cnpq.br/1419505294146027>
E-mail: lorena.oliveira@ufvjm.edu.br

Franciele Leal Farias
Discente em Engenharia de Produção da
Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM
<http://lattes.cnpq.br/2027068565616129>
E-mail: franleal09@hotmail.com

Jhieson Luiz Seles
Discente em Engenharia Civil da
Universidade Federal dos Vales do Jequitinhonha e Mucuri – UFVJM
<http://lattes.cnpq.br/6825084965355262>
E-mail: jhiesonx@hotmail.com

Resumo: A otimização de áreas é uma técnica cada vez mais procurada na sociedade atual, que sempre está em busca da melhor utilização de espaços. Almejando a simplificação desse processo, os Algoritmos Genéticos foram escolhidos para realizar a busca pela aplicação mais viável e que solucione este tipo de problema. A construção do algoritmo foi implementada no ambiente NetBeans e na linguagem Java. O algoritmo foi capaz de encontrar uma solução satisfatória, determinando a melhor localização possível para que a quantidade de circunferências desejadas abrangesse a maior parte da área determinada.

Palavras-chave: Algoritmos Genéticos, otimização de áreas, circunferências, Inteligência Artificial.

INTRODUÇÃO

Os Algoritmos Genéticos (AGs) são inspirados no processo de modificação natural das populações, segundo os princípios da seleção natural de Darwin (FILITTO, 2008).

Darwin propôs o conceito de seleção natural opondo-se à corrente de pensamento da época (o criacionismo) em favor da idéia de evolução das espécies, que foi descrita no seu livro de 1859, “A Origem das Espécies” (CORREA, 2002).

Já no final da década de 60, John Holland estudou formalmente a evolução das espécies e propôs um modelo heurístico computacional que, ao ser implementado, poderia oferecer boas soluções para problemas considerados insolúveis computacionalmente, até aquela época. O trabalho de Holland apresenta os Algoritmos Genéticos como uma metáfora para os processos evolutivos, estudando a adaptação e a evolução no mundo real, simulando-a por linhas de códigos. Porém, os AGs transcenderam o papel originalmente imaginado por Holland, transformando-se em uma ferramenta de uso altamente disseminado pelos cientistas da computação (LINDEN, 2012).

Os Algoritmos Genéticos são uma classe de procedimentos incluídos como uma das técnicas de Inteligência Artificial, que se destaca pelo desempenho adequado em otimização de processos (LINDEN, 2012).

No atual cenário de competitividade mundial, o tema otimização se encontra constantemente preso a procura por maneiras de potencializar processos nas mais diferentes áreas de conhecimento. Essa busca é caracterizada pelo anseio de encontrar maneiras de maximizar os ganhos tendo o mínimo de perdas possíveis. A partir dessa corrida estimulada pelo avanço científico atual, diversos estudiosos tem se dedicado a encontrar métodos computacionais que sejam genéricos e bastante abrangentes, possibilitando o seu uso em diversos problemas (MUNDIM, DELAVY, 2008).

Este trabalho tem por objetivo a criação de um algoritmo capaz de determinar a distribuição de várias circunferências em uma área retangular ou quadrangular, impedindo que as mesmas ultrapassem as delimitações da área. Porém quanto às posições das circunferências, estas poderão se sobrescrever ou não, dependendo

da necessidade do usuário, das informações que serão fornecidas e da opção escolhida antes do algoritmo iniciar a resolução do problema.

O algoritmo desenvolvido não possui um problema ou função específica, por isso é possível utilizá-lo para solucionar diversos problemas baseados na otimização de áreas. Como o espaço a ser preenchido é retangular ou quadrado, e são utilizadas circunferências para preenchê-lo, este algoritmo poderia ser aplicado à alocação de sistemas de irrigação ou a distribuição de roteadores, por exemplo.

Através deste trabalho será possível determinar a ocupação das áreas sem que o usuário possua conhecimento sobre Algoritmos Genéticos ou sobre a ferramenta utilizada. Será necessário, apenas, que ele possua as informações que devem ser preenchidas no início do algoritmo.

METODOLOGIA

A teoria da evolução, ou seleção natural, diz que na natureza todos os indivíduos dentro de uma mesma espécie competem entre si. Aqueles que não são bem sucedidos serão menos reproduzidos, e sua descendência reduzida. Dessa forma, a probabilidade de ter seus genes propagados nas futuras gerações é menor (LINDEN, 2012).

A combinação entre as características dos indivíduos que possuem maior probabilidade de sobrevivência podem produzir um novo indivíduo, com características que o torna mais adaptado ao meio em que vive. Desse modo, um indivíduo descendente é ligeiramente diferente de seus pais, podendo ser mais ou menos adaptado ao meio em que vive (LINDEN, 2012).

Para entender o funcionamento de um Algoritmo Genético, é necessário conhecer um pouco da biologia envolvida no processo. Anos depois das teorias apresentadas por Darwin, no início do século XX, Gregor Mendel compreendeu que o processo de transmissão de informações estava associado a uma unidade básica de informação denominada gene (LINDEN, 2012).

Basicamente, todo indivíduo é formado por uma ou mais células e dentro de cada uma delas o organismo possui uma cópia completa do conjunto de cromossomos que descrevem o organismo, sendo este denominado genoma. Os cromossomos vêm em pares, sendo o número de pares variável para cada espécie.

Um cromossomo consiste em genes, sendo que cada gene é uma região do DNA que tem uma posição específica no cromossomo e controla uma característica hereditária também específica (LINDEN, 2012).

Os AGs procuram resolver questões diárias, através de um processo iterativo, de busca da melhor solução para um problema. O processo trata da transformação de um conjunto com número fixo de estruturas (população de indivíduos) em um novo conjunto, utilizando as três principais operações genéticas: seleção, reprodução e mutação (FILITTO, 2008; RIBEIRO FILHO, 2000).

Estrutura de um AG

A Figura 1 apresenta um fluxograma que representa as etapas de um Algoritmo Genético. Primeiramente a população foi gerada utilizando a codificação binária. Em seguida ela foi inicializada de maneira randômica. O próximo passo foi a seleção dos indivíduos que participaram do cruzamento através do método do giro de roleta. Na sequência ocorreu o cruzamento e a mutação dos indivíduos selecionados, para tanto foi utilizado o cruzamento uniforme e a mutação por inversão de bits. No decorrer do artigo serão descritos cada um dos métodos utilizados.

A avaliação do critério de parada descrito na Figura 1 corresponde ao número de gerações indicadas para o algoritmo realizar todos os processos, ou seja, a quantidade de interações que o algoritmo deverá realizar.

Será descrito a seguir todo o processo que foi aplicado ao algoritmo criado para este trabalho.

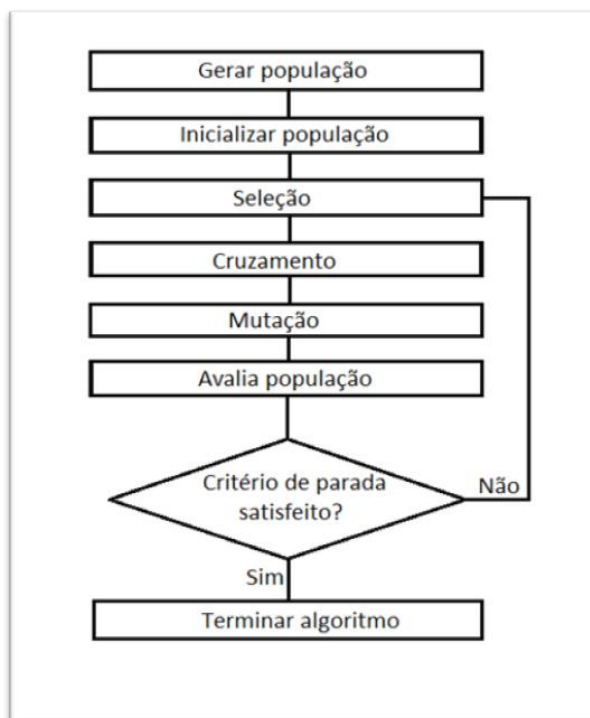


Figura 1: Representação de um Algoritmo Genético em forma de Fluxograma

Geração da população

Os principais componentes dos Algoritmos Genéticos são:

Indivíduos - Os indivíduos são os componentes principais do Algoritmo Genético, pois através de sua manipulação são codificadas as possíveis soluções do problema. Genoma e cromossoma são termos utilizados como sinônimos de indivíduos na área da Inteligência Artificial (IA) e sugere que eles são formados por um conjunto de genes (FILITTO, 2008; SOARES JÚNIOR, 2011).

Populações – As populações são constituídas por conjunto de indivíduos. A sua evolução só é possível pelas dinâmicas populacionais e pelos processos de manipulação (FILITTO, 2008; SOARES JÚNIOR, 2011).

Na sequência, é necessário conhecer um pouco sobre o espaço de busca de soluções. Nele existem as soluções ótimas que são o conjunto de resultados que se pode obter para uma situação/problema. As soluções ótimas são divididas em:

Ótimo global – É o melhor resultado que pode ser obtido em um determinado problema podendo, muitas vezes, não ser alcançado ao fim da execução do algoritmo.

Ótimos locais – São os melhores resultados encontrados ao término do algoritmo, sendo que esses se aproximam do ótimo global, porém não chegam a se igualar a ele.

Métodos utilizados na geração da população

Os métodos utilizados neste algoritmo são:

Codificação Binária - Esse tipo de codificação foi a primeira a ser utilizada por pesquisadores de AGs. Por esse motivo e por sua simplicidade de uso, ela é a codificação mais comum no uso de Algoritmos Genéticos. A Figura 2 apresenta dois exemplos de indivíduos com codificação binária (OBITKO, SLAVÍK, 1999).

Cromossomo A	101100101100101011100101
Cromossomo B	111111100000110000011111

Figura 2: Codificação Binária

A Codificação Binária permite diversos possíveis cromossomos, mas para alguns problemas, essa codificação não é natural e as vezes precisa receber modificações antes dos cruzamentos e/ou mutações (OBITKO, SLAVÍK, 1999).

Inicialização randômica uniforme - nesse tipo de inicialização, cada gene do indivíduo receberá como valor um elemento do conjunto de alelos, sorteado de forma aleatoriamente uniforme (CORREA, 2002).

A função de inicialização é basicamente a síntese de uma população inicial, que será submetida à sequência de processos descritos em um Algoritmo Genético. Esta função recebe funções aleatórias que garantem indivíduos escolhidos de maneira que todo o espaço de pesquisa seja abrangido. A avaliação possui grande utilidade para selecionar a população que será utilizada, quando os dados são obtidos através do processo de mineração de dados (FILITTO, 2008; SOARES JÚNIOR, 2011).

Seleção por giro de roleta - Este método de seleção é feito através do cálculo do somatório de adequação da população (total), sorteando-se um valor i que pertença ao intervalo em questão. O indivíduo que corresponda à faixa do somatório onde i se encontra é selecionado (CORREA, 2002).

Em uma roleta são colocados todos os cromossomos da população. Cada seção dessa roleta é equivalente ao valor de aptidão do cromossomo ali contido. Assim, quanto maior for esse valor, maior será a fração da roleta correspondente a ele, e por consequência, maior será a chance daquele indivíduo ser selecionado. A Figura 3 é um gráfico que representa a roleta e mostra como seriam distribuídas as frações para cada indivíduo (OBITKO, SLAVÍK, 1999).

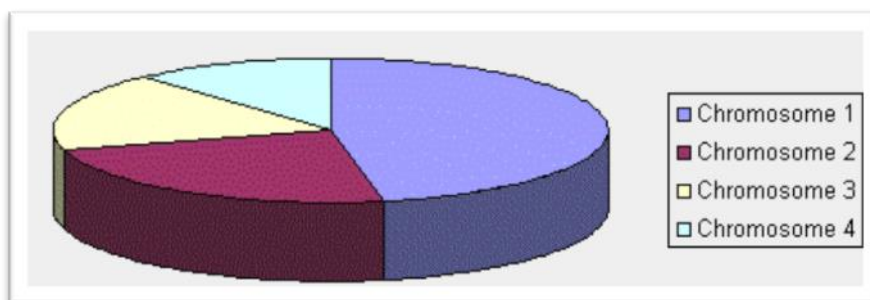


Figura 3: Gráfico de adequação (OBITKO, 1998)

Um algoritmo para descrever esse método pode ser descrito da seguinte forma:

[Soma] Calcule a soma dos valores de adequação de todos os cromossomos da população - soma S .

[Seleção] Gere um número aleatório no intervalo $(0, S)$ - r .

[Repetição] Percorra toda a população e some a adequação de 0 - soma S . Quando a soma s for maior que r , pare e retorne o cromossomo atual.

A função da seleção é escolher os elementos que possuem maior chance de solucionar o problema abordado. Os elementos que melhor se adaptarem ao ambiente terão maior probabilidade de participar do processo de reprodução (denominado *crossing over* ou *crossover*). A seleção acontece através de funções matemáticas que são determinadas pelo tipo de AG e pelo problema analisado. O operador de seleção é um componente fundamental do algoritmo e é classificado pela sua intensidade de seleção (FILITTO, 2008; SOARES JÚNIOR, 2011).

Um importante passo da seleção é a função de avaliação que consiste em um elo entre o Algoritmo Genético e o meio ao qual ele está inserido. A avaliação é feita através de uma função inserida no AG que traduz o problema analisado, e retorna o nível de aptidão de cada indivíduo da população. Essa parte do algoritmo,

denominada avaliação, também é conhecida como função objetivo. Um exemplo de função de avaliação é a função matemática $f(x) = x^2 + 15$, que quando aplicada a população determinará qual é o mais apto (FILITTO, 2008; SOARES JÚNIOR, 2011).

O Cruzamento e a Mutação são dois operadores básicos de um Algoritmo Genético. O desempenho de um AG vai depender muito deles, e a implementação dos mesmos depende da codificação e do problema (OBITKO, SLAVÍK, 1999).

O operador de cruzamento tem a função de combinar partes dos cromossomos dos pais, para criar o cromossomo dos filhos. Uma das funções do cruzamento é o crossover, onde uma determinada parte de dois indivíduos são trocados gerando um descendente diferente dos originais (FILITTO, 2008; SOARES JÚNIOR, 2011).

A probabilidade (taxa) de Cruzamento indica a probabilidade de atuação do operador de cruzamento, com variação de 0 a 100%. Quanto maior esta probabilidade, maior será a inserção de novos indivíduos na população. Isto pode trazer um resultado indesejado, pois como boa parte da população estará sendo substituída, pode ocorrer a perda de indivíduos com alta aptidão. Por outro lado, com um valor baixo o processamento do algoritmo pode tornar-se demasiadamente lento (LIMA, 2006). No geral esta taxa deve ser alta, estando entre 80%-95%. No entanto, para alguns problemas específicos, resultados mostraram que uma taxa de aproximadamente 60% é melhor (OBITKO, SLAVÍK, 1999).

O operador de mutação é responsável pela introdução de pequenas mudanças que podem, ou não, ocorrer nos cromossomos dos descendentes (FILITTO, 2008; SOARES JÚNIOR, 2011).

A Probabilidade (taxa) de Mutação indica a probabilidade de atuação do operador de mutação, com variação de 0 a 100%. Uma pequena probabilidade de mutação possibilita que se chegue a qualquer ponto do espaço de busca, enquanto que uma alta probabilidade de mutação pode tornar a busca essencialmente aleatória (LIMA, 2006). Diferentemente da taxa de cruzamento, a taxa de mutação deve ser muito baixa. Melhores taxas aparentam estar entre 0.5%-1% (OBITKO, SLAVÍK, 1999).

Cruzamento Uniforme - os bits são copiados aleatoriamente do primeiro ou segundo pai (OBITKO, SLAVÍK, 1999).

A Figura 4 apresenta o cruzamento propriamente dito, com indivíduos com codificação binária.

$$11001011 + 11011101 = 11011111$$

Figura 3: Crossover Binário Uniforme

Mutação: Inversão de Bit - determinados bits são invertidos. (OBITKO, SLAVÍK, 1999)

Na Figura 5 a mutação é realizada em um indivíduo binário e a mudança acontece no segundo Bit.

$$11001001 \Rightarrow 10001001$$

Figura 5: Inversão de Bit Binário

Especificações do Algoritmo

Adaptação dinâmica - No algoritmo foi utilizado o método da adaptação dinâmica, onde é quantificado o valor de avaliação de cada indivíduo e o valor médio do grupo para verificação da homogeneidade da população. Esse método possui grande importância no algoritmo, pois ele é capaz de determinar se a população utilizada possui, ou não, uma boa diversificação. Através desse processo de determinação será definida a ocorrência do crossover ou da mutação. A análise é feita a partir da seguinte equação:

$$f_{cp} = 1 - \frac{f_{med}}{f_{max}} \quad (1)$$

Nesta equação, f_{med} é o valor médio das avaliações, f_{max} é o valor da melhor avaliação e f_{cp} é o fator de convergência do algoritmo. Se $f_{cp} \leq 0.01$ significa a homogeneidade da população, assim é invertida a probabilidade de uso dos operadores genéticos deixando o operador de mutação com maior chance de ser escolhido. Se $f_{cp} > 0.01$ significa que população está diversificada e a probabilidade de uso do operador de crossover permanece maior.

Quando o valor de f_{cp} é menor ou igual a 0.01, a população possui indivíduos muito semelhantes, por isso é importante elevar a probabilidade de uso do operador de mutação para que possa ocorrer a diversificação da população, pois não seria interessante realizar o cruzamento entre indivíduos extremamente semelhantes.

Elitismo - No decorrer do processo de gerações, o indivíduo perfeito pode ser encontrado e quando o algoritmo continua o seu desenvolvimento, ele pode acabar sendo perdido no cruzamento ou na mutação. Para evitar sua perda, foi inserido o elitismo. Este método é responsável por armazenar o indivíduo perfeito quando ele é encontrado e impede que ele seja modificado.

Ferramentas utilizadas

NetBeans - A ferramenta utilizada para implementação do Algoritmo Genético proposto é o *NetBeans*, que é um ambiente de desenvolvimento integrado (IDE) para desenvolvimento de softwares em diversas linguagens. Ele oferece aos usuários as ferramentas necessárias para criar aplicativos profissionais de *desktop*, empresariais, *Web* e multiplataformas. O *NetBeans* fornece uma base sólida para a criação de projetos e módulos, possuindo um grande conjunto de bibliotecas, módulos e uma documentação vasta bem organizada (NETBEANS, s.d.).

O *NetBeans* ainda possui um framework baseado em Java para implementação de Algoritmos Genéticos, o JGAP (*Java Genetic Algorithms Package*). Essa ferramenta proporciona mecanismos genéticos básicos de fácil utilização para solucionar problemas aplicando os princípios evolutivos (MEFFERT, ROTSTAN, 2012).

GeoGebra - O GeoGebra também é uma ferramenta utilizada neste trabalho. Ele se trata de uma multiplataforma de matemática dinâmica que integra geometria, álgebra, gráficos, estatística, cálculo e planilha eletrônica em um único ambiente. É completamente interativo e possui uma interface gráfica simples e funcional (GEOGEBRA, s.d.). O GeoGebra foi utilizado nesta pesquisa para representar graficamente as posições determinadas pelo algoritmo, facilitando a visualização da ocupação da área desejada.

RESULTADOS E DISCUSSÕES

Primeiros Testes

Após o algoritmo ser desenvolvido foram realizados alguns testes para demonstrar o seu funcionamento e seu desempenho satisfatório.

Os quatro primeiros testes foram realizados utilizando valores diferentes, apenas para os parâmetros precisão e a possibilidade das circunferências se sobreporem. Os testes podem ser visualizados na Tabela 1.

Para a melhor compreensão do problema é necessário explicar que o valor da precisão descrito na tabela corresponde à distância entre os pontos da área que serão analisados. Por exemplo, nesse primeiro teste a área utilizada é de 10x10 m² e a precisão é de 1 metro, então o algoritmo dividirá a área em 100 pontos separados de 1 em 1 metro. Através dessa separação será possível visualizar quais desses pontos foram cobertos pelas circunferências e assim analisar se a área especificada está sendo ou não coberta adequadamente.

Variáveis	1º teste	2º teste	3º teste	4º teste
Área(m ²)	(10x10)	(10x10)	(10x10)	(10x10)
Nº circunferências	5	5	5	5
Raio(m)	2	2	2	2
Taxa de mutação	5%	5%	5%	5%
Probabilidade de uso do operador de mutação*	15%	15%	15%	15%
Probabilidade de uso do operador de crossover*	85%	85%	85%	85%
Tamanho da população	50	50	50	50
Número de gerações	250	250	250	250
Precisão(m)	1	0,5	0,5	1
Circunferências sobreescritas	Não	Não	Sim	Sim
Avaliação do indivíduo	65**	245	245	65**

Tabela 1: Resultados dos quatro primeiros testes

Nas duas variáveis que se encontram identificadas com (*) os seus valores não são fixos devido à utilização da adaptação dinâmica da população.

Os dois valores identificados com (**) mostram que um mesmo valor de avaliação foi encontrado para o melhor indivíduo do Teste 1 (Figura 6) e do Teste 4 (Figura 9), sendo estes realizados com o mesmo valor de precisão mas com a possibilidade das circunferências se sobreporem diferentes. No primeiro caso, o indivíduo realmente representa a solução ótima para o problema, como pode ser observada na Figura 6. O mesmo não ocorre no Teste 4, onde a solução encontrada não corresponde a solução ótima, como pode ser observado na Figura 9. Isso ocorre principalmente pelo fato de que o algoritmo percorreu os pontos com uma precisão pequena, dessa forma os pontos que eram cobertos mais de uma vez foram pulados no momento da verificação. Outro fator que teve influência para que o resultado fosse o satisfatório no primeiro caso foi o fato de que as circunferências não poderiam ser sobrescritas, assim uma avaliação complementar foi realizada eliminando os falsos ótimos encontrados.

A Figura 7 mostra o resultado do Teste 2 e a Figura 8 mostra o resultado do Teste 3. Ambos os testes tiveram resultados satisfatórios.

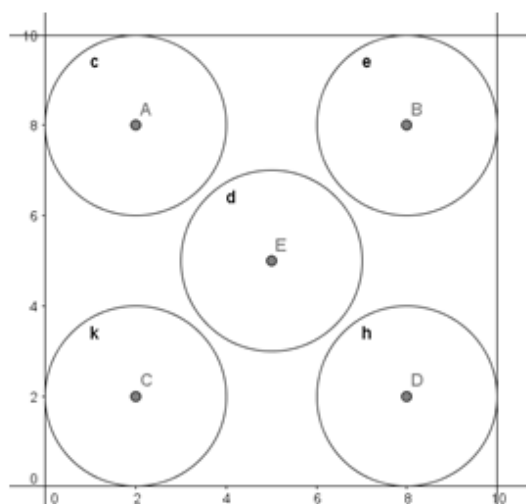


Figura 6: Resultado do Teste 1

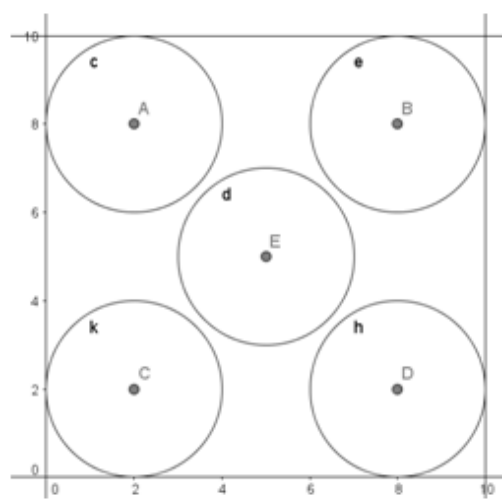


Figura 7: Resultado do Teste 2

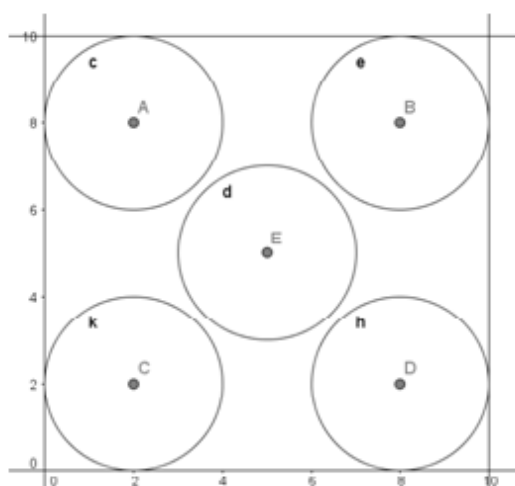


Figura 8: Resultado do Teste 3

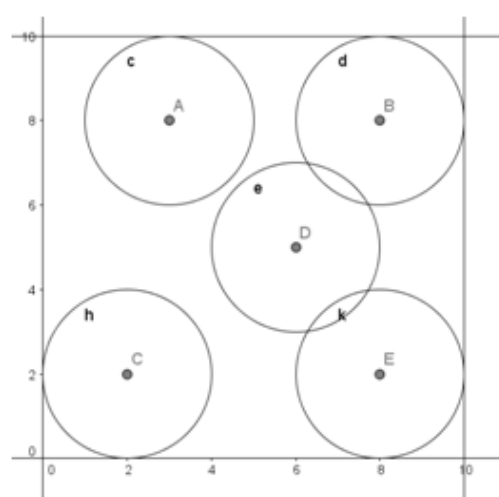


Figura 9: Resultado do Teste 4

Através desses primeiros testes, foi possível observar que quanto menor for o intervalo de um ponto a outro (a precisão analisada) melhor será a verificação da área abrangida pelas circunferências. Mas, em contrapartida, o tempo de execução do algoritmo aumenta drasticamente, pois o número de verificações se torna cada vez maior.

Testes posteriores

Os outros quatro testes realizados foram utilizados com a variação dos seguintes parâmetros: área, número de circunferências, tamanho da população, número de gerações, alteração da possibilidade das circunferências se sobreporem e da precisão. Os valores utilizados podem ser observados na Tabela 2.

Variáveis	1º teste	2º teste	3º teste	4º teste
Área(m ²)	(12x12)	(12x12)	(9x8)	(9x8)
Nº circunferências	8	8	5	5
Raio(m)	2	2	2	2
Taxa de mutação	5%	5%	5%	5%
Probabilidade de uso do operador de mutação*	15%	15%	15%	15%
Probabilidade de uso do operador de crossover*	85%	85%	85%	85%
Tamanho da população	50	30	30	30
Número de gerações	250	120	120	120
Precisão(m)	1	0,1	1	0,1
Circunferências sobrescritas	Não	Sim	Não	Sim
Avaliação do indivíduo	50	9997	26	5158

Tabela 2: Resultados dos quatro testes posteriores

Nas duas variáveis que se encontram identificadas com (*) os seus valores não são fixos devido à utilização da adaptação dinâmica da população.

A Figura 10 representa o resultado do Teste 1, onde são alocadas oito circunferências que não podem se sobreescrever. A Figura 11 mostra o resultado do Teste 2, onde também foram alocadas oito circunferências, porém, estas podiam se sobreescrever e apresentavam uma precisão maior do que a do Teste 1. Analisando e comparando as duas figuras é possível perceber que quando foi utilizada uma precisão pequena no Teste 1, os pontos que se sobreescreviam não foram analisados e o algoritmo interpretou que não haviam circunferências se sobreescrevendo. Já no Teste 2, como a precisão era maior, o algoritmo conseguiu separar as circunferências de maneira que nenhuma se sobreescrevesse.

Através destes testes, pode-se perceber que quando a precisão utilizada é pequena, o algoritmo atingirá resultados mais satisfatórios se o usuário escolher a opção de que as circunferências não podem se sobreescrever.

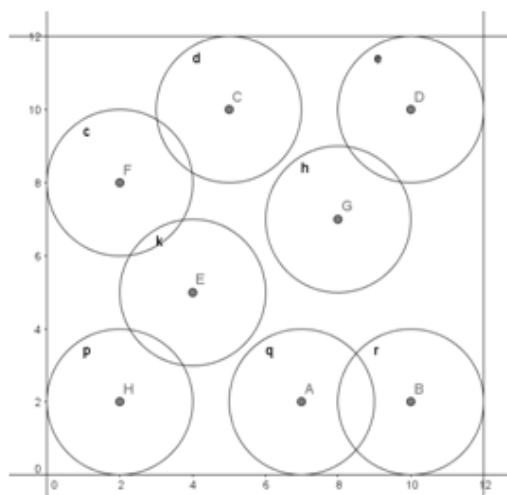


Figura 10: Resultado do Teste 1

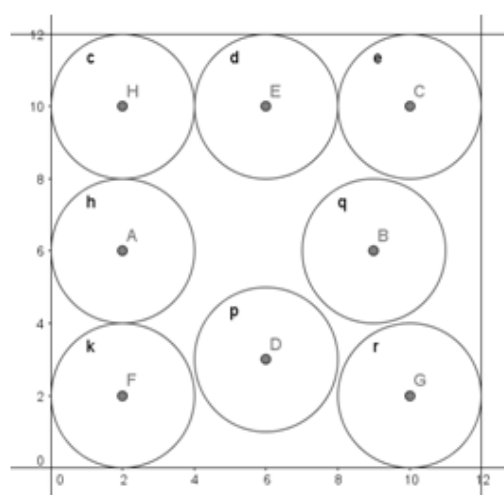


Figura 11: Resultado do Teste 2

A Figura 12 apresenta o resultado do Teste 3 que realiza a alocação de cinco circunferências, com precisão de um metro e que não permite a possibilidade de se sobreporem. A Figura 13 apresenta o resultado do Teste 4, onde as mesmas cinco circunferências são alocadas, porém, estas possuem uma precisão maior e podem ser sobreescritas. Analisando as duas figuras pode-se perceber que são extremamente semelhantes e que o algoritmo alcançou a mesma solução. Porém, observando a avaliação de ambos os testes na Tabela 2, é possível verificar que a avaliação do indivíduo do Teste 4 é muito superior ao do indivíduo do Teste 3, pois o último teste aceitava a possibilidade das circunferências se sobreporem.

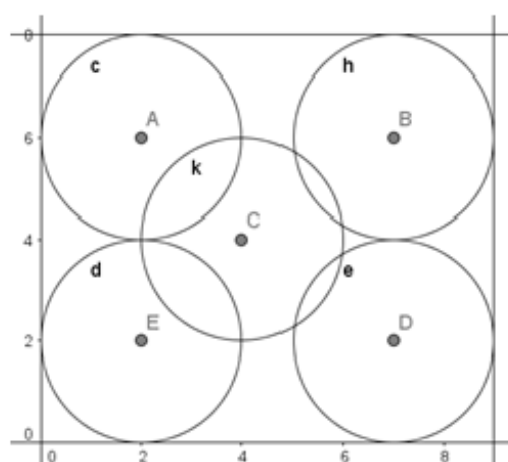
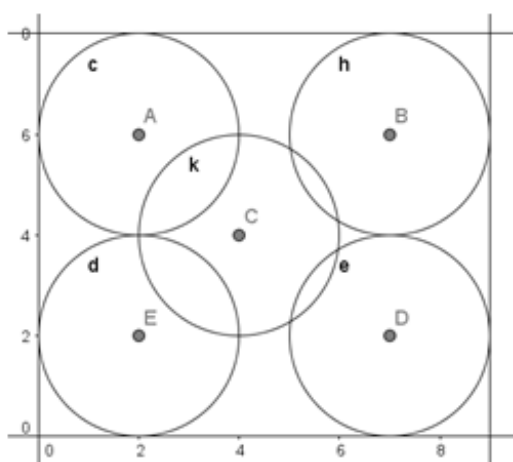


Figura 12: Resultado do Teste 3

Figura 13: Resultado do Teste 4

CONCLUSÃO

A otimização de processos se tornou uma questão bastante atual e estão sendo desenvolvidas novas técnicas que ajudem a solucionar este problema. Na busca por uma solução satisfatória, os Algoritmos Genéticos se destacaram por ser um método computacional genérico que possibilita uma implementação relativamente simples. O algoritmo desenvolvido no trabalho em questão possui a funcionalidade de se adaptar a diversos problemas e ajudar a solucioná-los sem que seja necessário o conhecimento sobre AGs.

Neste trabalho, é possível observar que alguns parâmetros devem ser levados em consideração na execução do algoritmo para que o resultado satisfatório seja alcançado. Como a alteração da taxa de mutação e de cruzamento realizada pela adaptação dinâmica. Bem como a importância da precisão em determinados problemas, já que o seu uso pode acarretar um tempo de execução mais longo, porém cabe ao usuário determinar qual a forma mais viável para solucionar o seu problema.

A possibilidade de circunferências se sobresscreverem pode causar problemas na execução do algoritmo, levando o mesmo a não alcançar uma solução satisfatória para o usuário.

O trabalho alcançou resultados satisfatórios. Porém, pretende-se modificar o método como este algoritmo funciona, alterando a forma de codificação dos indivíduos. Com a sequência deste trabalho espera-se encontrar novos resultados que ajudem a aumentar a sua eficiência e a sua área de atuação, podendo ser utilizado em várias áreas de pesquisa como em instalações hidráulicas voltadas a irrigação, otimização de processos ligados a engenharia de produção, entre outros.

Referências

CORREA, Diogo Lucas. **Algoritmos Genéticos: uma Introdução**. Apostila elaborada sob a orientação de Luís Otávio Álvares, para a disciplina de Ferramentas de Inteligência Artificial – Universidade Federal do Rio Grande do Sul. Porto Alegre. Março, 2002.

FILITTO, Danilo. **ALGORITMOS GENÉTICOS: UMA VISÃO EXPLANATÓRIA**. Revista multidisciplinar da UNIESP, Saber Acadêmico - nº 06 - Dezembro 2008/ISSN 1980-5950. Págs. 136-137.

GEOGEBRA [Internet]. **Visão geral sobre o Geogebra**. [s.d.] Disponível em: < <https://www.geogebra.org/> > . Acesso em 07 de julho de 2015.

LIMA, Mark Renato Campos. **Algoritmos genéticos na formação de grupos para aprendizagem cooperativa apoiada por computador**. Dissertação (Mestrado) – Universidade Federal do Maranhão. São Luiz. Maranhão, 2006.

LINDEN, Ricardo. **Algoritmos Genéticos**. 3.ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2012. 475p.

MEFFERT, Klaus; ROTSTAN, Neil. 2002-2012. Disponível em: <http://jgap.sourceforge.net>. Acesso em: 07 de jul. 2014.

MUNDIM, Kleber; DELAVY, Vilma. **Otimização Global de Processos Usando o Método Generalized Simulated Annealing**. Revista Processos Químicos – Pag 09-23. Julho – Dezembro, 2008.

NETBEANS [Internet]. Visão geral do NetBeans IDE. [S.d.] Disponível em: <https://netbeans.org/features/index_pt_BR.html> . Acesso em: 07 de jul. 2015.

OBITKO, Marek. **Introdução aos Algoritmos Genéticos**. Tradução por Hermelindo Pinheiro Manoel. Universidade de Ciências Aplicadas de Dresden, 1998. Disponível

em: <<http://www.obitko.com/tutorials/genetic-algorithms/portuguese/index.php>>
Acesso em: 20 de ago. 2014.

OBITKO, Marek; SLAVÍK, Pavel. **Visualization of Genetic Algorithms in a Learning Environment**. Spring Conference on Computer Graphics, SCCG'99. Bratislava: Comenius University, 1999, p. 101-106. ISBN 80-223-1357-2.

SOARES JÚNIOR, Amílcar. **X-GAT: Uma ferramenta baseada em xml para otimização com algoritmos genéticos**. Dissertação (Mestrado) – Universidade Federal da Paraíba. João Pessoa. Paraíba, 2011.

Processo de Avaliação por Pares: (*Blind Review* - Análise do Texto Anônimo)

Publicado na Revista Vozes dos Vales - www.ufvjm.edu.br/vozes em: 05/2017

Revista Científica Vozes dos Vales - UFVJM - Minas Gerais - Brasil

www.ufvjm.edu.br/vozes

www.facebook.com/revistavozesdosvales

UFVJM: 120.2.095-2011 - QUALIS/CAPES - LATINDEX: 22524 - ISSN: 2238-6424

Periódico Científico Eletrônico divulgado nos programas brasileiros *Stricto Sensu*

(Mestrados e Doutorados) e em universidades de 38 países,

em diversas áreas do conhecimento.