

Universidade Federal de São João del-Rei
Algoritmo Bioinspirados
Implementação de Algoritmo Genético com Representação Real

Felipe Francisco Rios de Melo

1. INTRODUÇÃO

Um algoritmo genético (AG) é uma técnica de busca utilizada para achar soluções aproximadas em problemas de otimização e busca. (usam técnicas inspiradas pela biologia evolutiva como hereditariedade, mutação, seleção natural e recombinação)

Estes métodos estão sendo utilizados, cada vez mais, pela comunidade de inteligência artificial para obter modelos de inteligência computacional (Barreto 1997).

Por se tratar de uma técnica bastante difundida e geralmente oferecer bons resultados, os algoritmos genéticos são usados para solucionar diversos tipos de problemas, entre eles: Sistemas de classificação, escalonamento e grade horária.

2. IMPLEMENTAÇÃO

A seguir, será mostrado um resumo das decisões tomadas para a implementação do Algoritmo Genético:

2.1. Estrutura de dados

Foi implementado uma classe nomeada AG, que é instanciada uma única vez, contendo os atributos e métodos necessários para inicialização e manipulação do algoritmo.

- Atributos: tamanho da população, número de gerações, taxa de mutação, probabilidade de vitória na seleção por torneio, número de bits de cada indivíduo, o limite inferior (xmin) e superior (xmax);
- Métodos: gerar população, avaliar população, seleção por roleta, crossover utilizando o algoritmo blend- $\alpha\beta$ ou blend- α , mutação e etc.

Além da classe AG, existe também a classe Indivíduo, usada para alocar os parâmetros (x1 e x2) de cada indivíduo, bem como também, seu fitness na função objetivo.

2.2 População

A população foi gerada como uma matriz com $n \times m$, onde n é o tamanho da população e m o número de genes. O valor de cada elemento é obtido a partir do sorteio de um número real entre $xmin$ e max .

2.3. Indivíduos

Obtida a população inicial, para cada indivíduo da população, instanciamos um objeto da classe *Individuo*, e setamos o seu $x1$ e $x2$, que são números reais compreendidos no intervalo definido por $xmin$ e $xmax$.

2.4. Função Objetivo

Aplica-se a função objetivo, guardando o resultado no atributo *fitness* na instância da classe *Individuo*, referente ao indivíduo em questão.

2.5. Elitismo

Busca em todos os indivíduos, àquele que obteve a melhor avaliação (mais próximo de zero), e guarda-o para ser usado mais adiante.

2.6. Seleção

Nesta implementação, a seleção dos indivíduos para o cruzamento é feito pelo método da roleta. Neste método, cada indivíduo da população é representado na roleta proporcionalmente ao seu índice de aptidão. Assim, aos indivíduos com alta aptidão é dada uma porção maior da roleta, enquanto aos de aptidão mais baixa é dada uma porção relativamente menor da roleta. Finalmente, a roleta é girada um determinado número de vezes, dependendo do tamanho da população, e são escolhidos, como indivíduos que participarão da próxima geração, aqueles sorteados na roleta.

2.7. Crossover

Antes de executar o crossover, sorteia-se um número de 0 a 100, se este número for menor que a taxa de cruzamento, pré-estabelecido na instanciação da AG, prossegue-se com o cruzamento. Caso contrário os pais que iriam realizar o cruzamento são enviados para compor a nova população que está sendo gerada.

Os cruzamentos mais comuns para representações por números reais são os baseados em média ou os que envolvem o sorteio de valores em torno dos valores de cada características dos pais, como o BLX- α e o BLX- $\alpha\beta$.

α e β são parâmetros reais positivos, os valores sugeridos são: $\alpha = 0.75$ e $\beta = 0.25$.

A principal diferença entre os dois algoritmos é que no BLX- α , os valores a serem sorteados estão em torno dos valores dos dois pais indistintamente, enquanto no BLX- $\alpha\beta$ os valores estão mais próximos do pai com o melhor fitness, por isso os possíveis valores de α serão superiores aos valores de β .

Na implementação da AG, é dado a opção de executar o algoritmo tanto pelo método BLX- α , quanto pelo método BLX- $\alpha\beta$. E, é gerado dois filhos por cruzamento.

Em seguida, adiciona-se os dois filhos à nova população, criando instâncias para cada um deles de classe *Individuo*. Se não houve cruzamento, como já dito acima, adiciona-se os pais à nova população;

2.8. Nova População

Este processo de seleção (seção 2.6) e crossovers (seção 2.7), é repetido até que a nova população tenha obtido o tamanho da população anterior.

Logo, a nova população está no tamanho fixado, porém, devido a decisão de implementação do Elitismo - o melhor indivíduo da geração, sobrevive para a geração seguinte - é necessário abrir um espaço na nova população para o melhor indivíduo. Então, optou-se por aleatoriamente, remover um indivíduo da nova população para inserir o indivíduo mais bem avaliado.

2.9. Mutação

Com a nova população em mãos, para cada indivíduo dela, e para cada gene do indivíduo, sorteamos um inteiro entre 0 e 100, se este inteiro for menor ou igual a taxa de mutação (estipulada na instanciação da classe AG), o gene do indivíduo sofrerá uma mutação.

A mutação provoca a substituição do valor do gene do indivíduo por um novo número real aleatório no intervalo $[xmin, xmax]$.

2.10. Gerações

O passo a passo da seção 2.4 até a 2.9 é repetido para todas as gerações. A quantidade de gerações é estipulado no momento da instanciação da classe AG.

Em cada geração, o melhor indivíduo é imprimido na tela, junto a sua avaliação na função objetivo.

3. ANÁLISE DOS RESULTADOS

Para analisar a eficiência do algoritmo genético, foi realizado um experimento fatorial completo variando taxas de mutação, os valores de tamanho de população, a taxa de cruzamento e número de gerações, nos seguintes valores:

Taxa de mutação	1%	5%	10%
Taxa de cruzamento	60%	80%	100%
Tamanho da população	25	50	100
Número de gerações	25	50	100

Para cada uma das 81 combinações possíveis dos parâmetros acima, foi realizada 20 execuções do algoritmo, para obter um resultado consistente. Resultando num total de 1620 execuções, estas foram efetuadas de maneira automatizada pelo arquivo python *analysis.py*.

Para os outros parâmetros, neste experimento, os seguintes valores foram fixados:

número de genes: 2; xmin: -2; xmax: 2; alpha: 0.75; beta: 0.25

Ps: a identificação da combinação de parâmetros referente aos dados mostrados nesta análise estarão no padrão (*taxa de mutação - taxa de cruzamento - tamanho da população - quantidade de gerações*). Por exemplo: 10-80-100-100.

A seguinte tabela, contém as combinações, junto à média do fitness de convergência das 20 execuções para cada combinação.

TABELA 1. combinações e suas respectivas médias de fitness do melhor indivíduo, ordenado pela combinação com melhor fitness médio.

Combinação	Fitness médio
5-100-100-100	4.440892098500626e-16
10-100-100-100	4.440892098500626e-16

1-80-100-100	1.865174681370263e-15
1-100-100-100	3.819167204710538e-15
10-100-50-100	6.282085962538985e-13
10-60-100-50	5.341541431391761e-10
10-100-100-50	8.411634944138769e-10
10-80-100-50	9.212402662939213e-09
5-100-100-50	4.737817524969046e-08
	.
	.
	.
1-60-25-50	0.026511480113778508
10-60-50-25	0.026737590583486525
10-60-25-25	0.030490920791975195
1-60-25-25	0.04094003674244755
5-60-25-25	0.04905479191869366
1-100-25-25	0.09466348965731823

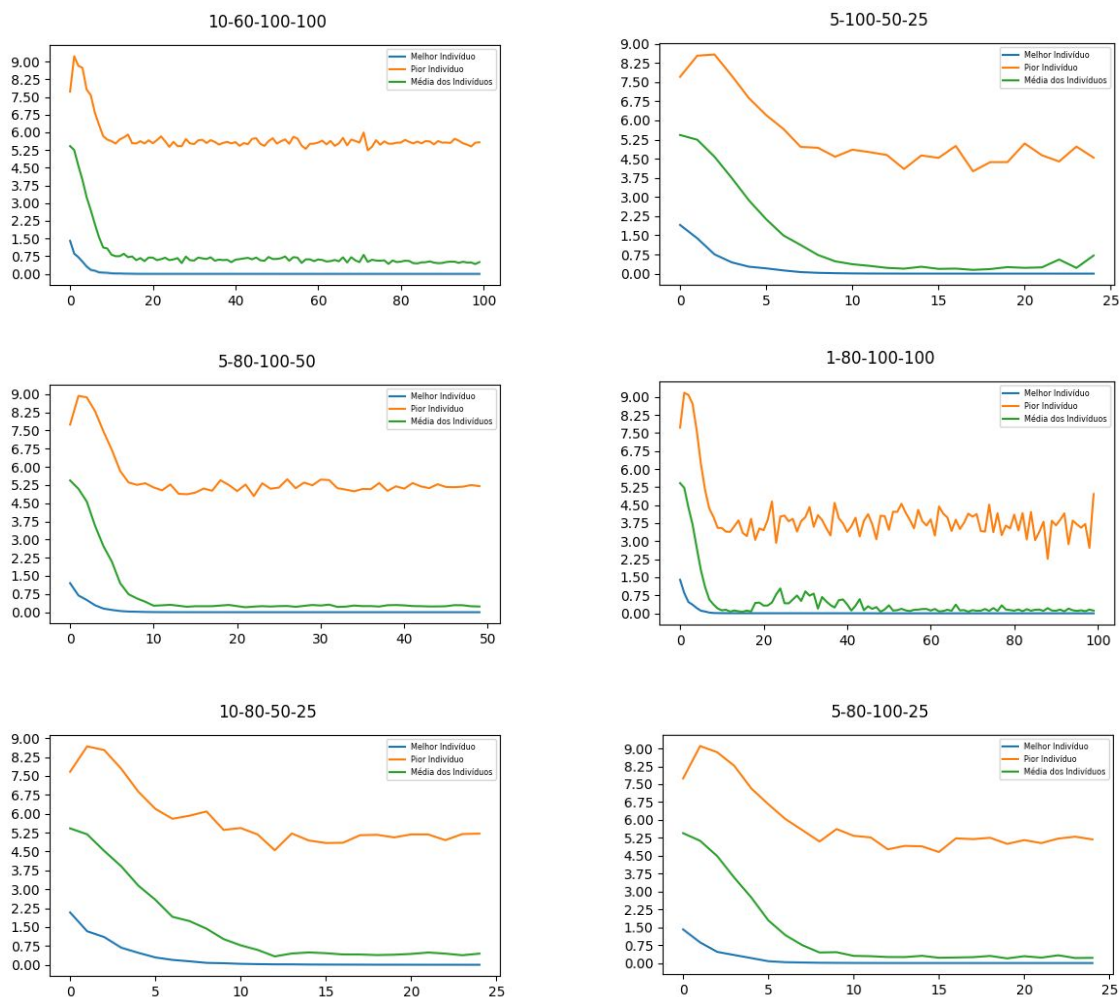
A notar pelas primeiras e últimas linhas da tabela, já podemos observar um certo padrão: as combinações com tamanho de população = 100, sobressai sobre as combinações com tamanho de população = 50, que por sua vez, sobressai sobre as combinações com tamanho de população = 25. O mesmo padrão se observa para o parâmetro de quantidade de gerações.

Isso ocorre, devido ao fato de quanto menor o número de gerações, menor será a quantidade de iteração para otimizar a solução. Enquanto que, quanto menor o tamanho da população, mais devagar será a curva de conversão da função objetivo, devido a pouca quantidade de indivíduos disponíveis para cruzarem, mutarem e etc.

A tabela acima, considera apenas a média do melhor indivíduo para cada combinação. Em alguns casos, nos algoritmos genéticos, pode acontecer de apenas alguns poucos indivíduos convergirem, devido a técnicas como o elitismo. Isto é, buscamos o melhor indivíduo de de uma determinada geração, e o mantemos para a próxima geração, porém, o restante da população, não consegue acompanhar a curva de evolução do melhor indivíduo.

Para analisarmos a evolução da população em geral ao passar das gerações, foi calculado a média do fitness dos indivíduos em cada geração.

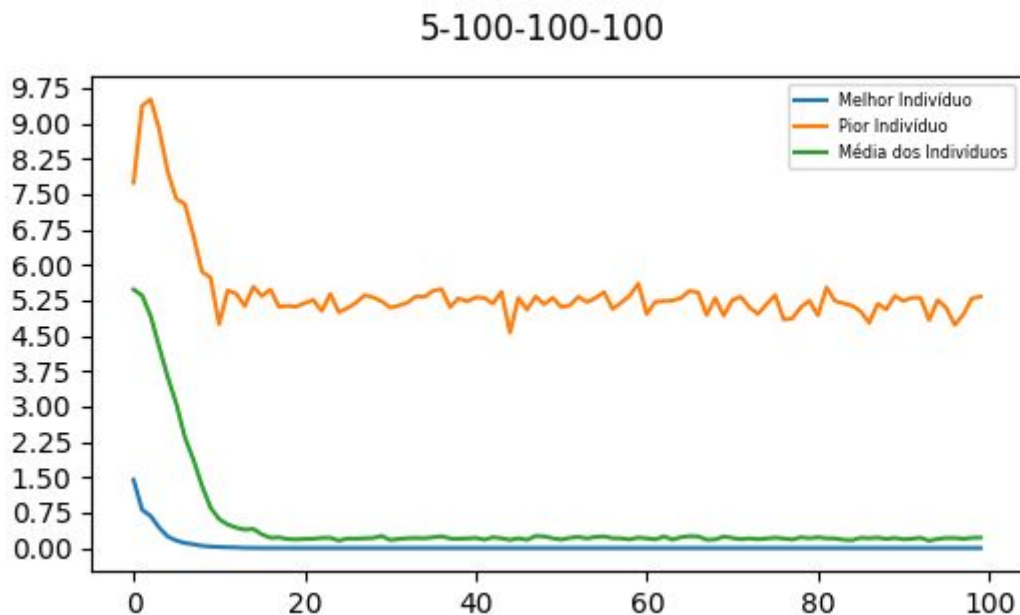
Abaixo, temos alguns gráficos que mostram a evolução dos indivíduos ao passar das gerações:



É possível perceber com esses 6 gráficos mostrados, e com todos os outros 75 restantes, que a população em geral foi muito bem na execução, pois a média dos indivíduos conseguiu acompanhar a evolução do melhor indivíduo.

Por meio da observação dos dados da tabela 1, e dos gráficos gerados, o conjunto de parâmetros com taxa de mutação = 5, taxa de cruzamento = 100, tamanho da população = 100 e quantidade de gerações = 100, obteve um excelente resultado, como pode ser visto a seguir:

GRÁFICO 1. Média das execuções da AG com o conjunto de parâmetros com taxa de mutação = 5, taxa de cruzamento = 100, tamanho da população = 100 e quantidade de gerações = 100.



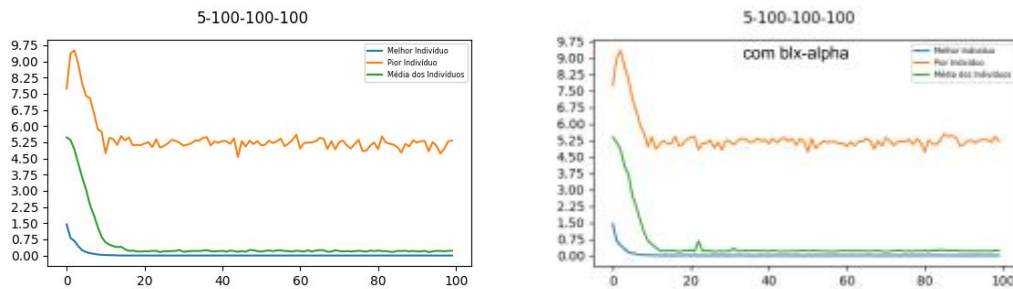
Pela tabela 1, vimos que esta configuração obteve a melhor média de melhores indivíduos. Agora, através deste gráfico, é possível notar que a média dos indivíduos chegou muito próximo do melhor fitness alcançado na execução. Confirmando, desta forma, a qualidade deste conjunto de parâmetros para otimizar a função objetivo dada.

Outra informação que podemos tirar da tabela 1 e dos gráficos gerados é que os melhores conjuntos de valores para os parâmetros, possui 100 gerações, e uma população de 100 indivíduos. Devido a estes altos valores, estas configurações, têm um custo computacional maior. Porém este custo é compensado pela média do fitness do melhor indivíduo, que possui uma minimização da função objetivo da ordem de grandeza de 10^{-16} , enquanto que o melhor conjunto de solução que não contém uma população de 100 indivíduos e 100 gerações conseguiu uma minimização da ordem de grandeza de 10^{-10} .

Trocando o algoritmo de cruzamento do BLX- $\alpha\beta$ para o BLX- α , e realizando o experimento fatorial completo, também obtemos uma predominância das combinações com 100 gerações e 100 indivíduos.

Abaixo, podemos ver observar a similaridade da configuração 5-100-100-100 no algoritmo BLX- $\alpha\beta$ e no algoritmo blx- α .

GRÁFICO 2. Configuração 5-100-100-100 com o algoritmo BLX- $\alpha\beta$ e BLX- α



Comparando a implementação do algoritmo genético com representação real em relação ao algoritmo genético com representação binária, constatou-se que a implementação com representação real obteve melhores resultados. Os melhores resultados flutuaram na ordem de grandeza de 10^{-16} , enquanto na implementação binária o máximo que se conseguiu minimizar foi 0.1799478151580023.

Alguns fatores contribuem para esta melhora na otimização pela implementação real. Como por exemplo:

A escolha do algoritmo de roleta para a seleção, que ao contrário do algoritmo usado na outra implementação, o torneio, os indivíduos possuem uma chance de ser escolhido proporcional ao seu fitness.

A técnica de mutação usada, que substitui totalmente o valor de um gene por outro valor aleatório, enquanto que na implementação anterior, a mutação acontecia em apenas um único bit de um binário de 6 bits que representa o gene. A substituição total do valor de um gene, e não só a troca de um bit, pode causar uma perturbação maior no conteúdo genético do indivíduo, fazendo com que ele consiga sair dos ótimos locais com mais facilidade.

4. CONCLUSÃO

Com este exercício de implementação foi possível colocar em prática a codificação de um algoritmo genético com representação real, e entender um pouco mais sobre o seu funcionamento.

Quanto às análises dos resultados, apesar de superficiais, foi capaz de mostrar a importância de variar os parâmetros para encontrar uma configuração que resulte em uma aceitável otimização da função objetivo.

5. REFERÊNCIAS BIBLIOGRÁFICAS

O que são algoritmos genéticos? - StackOverflow - Disponível em:
<<https://pt.stackoverflow.com/questions/185996/o-que-s%C3%A3o-algoritmos-gen%C3%A9ticos>>