INSTITUTO FEDERAL
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
Fluminense

MINISTÉRIO DA
EDUCAÇÃO

BRASIL
GOVERNO FEDERAL

ALGORITMO GENÉTICO (GA)

Aplicado ao problema da mochila binária

Iury Alcantara

Marcus Mariano

Mateus Terra Tavares Ramos

Vitor Baza Montoto

Campos dos Goytacazes

Agosto de 2024

1.0 A Metaheurística

1.1 Introdução

Inspirados pelos princípios da evolução natural propostos por Charles Darwin, os algoritmos genéticos surgiram na década de 1960. O conceito foi desenvolvido por John Holland, um dos pioneiros da área, para examinar como a seleção natural e a genética podem ser usadas para resolver problemas complicados. Os algoritmos genéticos foram inicialmente usados em pesquisas acadêmicas para modelar processos biológicos e em experimentos de otimização. Isso demonstrou sua capacidade de encontrar soluções quase ideais em espaços de busca grandes e complexos.

Os algoritmos genéticos são muito utilizados em problemas de otimização combinatória porque são excelentes em explorar espaços de solução grandes. Os algoritmos genéticos diferem de outros algoritmos, como os baseados em gradientes ou programação dinâmica, porque usam mutação, cruzamento e seleção para evoluir uma população de soluções por várias gerações. Eles podem evitar os locais mínimos e encontrar soluções que atendam a todo o mundo com essa abordagem. Isso é especialmente útil em problemas onde o espaço de busca é irregular ou mal compreendido.

1.2 Características

Algoritmos genéticos, baseados na seleção natural, são métodos de otimização que usam buscas estruturadas e paralelas para encontrar soluções ideais. Apesar da presença de elementos aleatórios, eles orientam a pesquisa usando dados acumulados ao longo das gerações, aplicando princípios de seleção e reprodução a uma população de candidatos. Uma população mais adaptada é formada por indivíduos mais aptos a se reproduzir.

Uma representação adequada dos problemas é necessária para o uso de algoritmos genéticos; isso geralmente é feito usando vetores binários para determinar a presença ou ausência de características. O método da Roleta é um exemplo de como as pessoas são selecionadas para a próxima geração com base em sua aptidão relativa. Este método permite uma evolução contínua, o que promove a diversidade e o desenvolvimento de soluções eficientes.

1.3 Componentes

Os algoritmos genéticos funcionam por meio de componentes básicos, como codificar as soluções, criar uma população inicial, usar operadores genéticos como mutação, seleção e cruzamento, e avaliar as soluções com base em uma função de aptidão. Essas partes funcionam juntas para simular o processo de evolução, o que permite que o algoritmo busque soluções melhores ao longo de várias gerações.

1.3.1 Cromossomo

O Algoritmo Genético trabalha com conjuntos de soluções do determinado problema que está tentando resolver. Esses conjuntos de soluções são chamados de cromossomos (ou indivíduos), sendo descritos como sequências de símbolos, chamados genes. Em uma analogia genética, os cromossomos são cadeias de genes que representam todos os traços característicos de um indivíduos e que são partilhados para os seres “filhos”, ou seja, aqueles gerados a partir de um casal de dois indivíduos.

Característica	A	B	C	D	E	F
Cromossomo 1 →	0	1	0	1	0	1
Cromossomo 2 →	1	0	0	1	0	1
Cromossomo 3 →	1	1	1	0	1	0
Cromossomo 4 →	1	0	0	0	0	0
Cromossomo 5 →	0	0	1	0	1	0

Figura 1 — *Visão dos cromossomos com base nas características do problema de otimização combinatória utilizado.* **Autoria própria**

1.3.2 Genes

Como os genes biológicos, os genes são as unidades de informação fundamentais que formam um indivíduo nos algoritmos genéticos. Cada gene representa uma característica ou parâmetro da solução específica que o algoritmo está tentando otimizar. As representações podem ser binárias, inteiras ou reais. Um cromossomo, que é uma solução completa para o problema, é formado por um conjunto de genes.

A adaptação e variabilidade de uma população dependem de seus genes. Os genes são combinados e alterados durante processos como cruzamento e mutação. Isso cria novas soluções e permite que o algoritmo explore diferentes partes do espaço de soluções. A evolução do algoritmo ao longo das gerações depende dessa variabilidade genética.

Característica		A	B	C	D	E	F
		0	1	0	0	1	0
		Gene 1	Gene 1	Gene 1	Gene 1	Gene 1	Gene 1

Figura 2 — *Visão dos genes de em cromossomo com base nas características do problema de otimização combinatória utilizado.* **Autoria própria**

1.3.3 População

A população é o conjunto de soluções, ou cromossomos, que existem em uma geração específica. Cada cromossomo da população representa uma solução potencial para o problema em questão. A população evolui ao longo de várias gerações, criando novos indivíduos por meio de operadores genéticos como mutação e cruzamento, enquanto os indivíduos menos qualificados são gradualmente eliminados. A população pode explorar o espaço de soluções com essa dinâmica e, idealmente, chegar a uma solução otimizada ao longo do tempo.

Característica	A	B	C	D	E	F
População (Geração x)	0	1	1	1	0	0
	1	1	0	0	1	1
	0	0	1	1	0	1
	0	0	0	0	1	0
	1	0	1	0	0	1
	0	1	0	1	1	0
	1	0	0	1	0	0
	1	0	1	0	0	1
	1	1	1	0	0	1
	0	1	0	1	1	1

Figura 3 — *Visão de uma população de uma geração aleatória com base nas características do problema de otimização combinatória utilizado.* **Autoria própria**

1.3.4 Cruzamento

Os métodos de cruzamento são responsáveis pela reprodução sexuada, que consiste na combinação de genes de cromossomos para produzir um ou mais descendentes, garantindo a diversidade e a evolução constante da população (Koza 1992). Os cromossomos podem se reproduzir de diversas formas, como a divisão dos cromossomos, em que se indica um ponto de “corte” onde o cromossomo “filho” gerado receberá a primeira parte do cromossomo gerador 1 e a segunda parte do cromossomo gerador 2, ou pelo pelo sorteio, onde o valor recebido por cada gene do cromossomo gerado é decidida com base em números aleatórios com 50% de probabilidade para cada cromossomo gerador.

Característica	A	B	C	D	E	F
Cromossomo gerador 1	0	1	0	1	0	0
Pontos de corte		1			2	
Cromossomo gerador 2	0	0	1	1	0	1
Cromossomo gerado	0	1	0	1	0	1

Figura 4 — *Cruzamentos de dois cromossomos por meio de corte.* **Autoria própria**

Característica	A	B	C	D	E	F
Cromossomo gerador 1	0	1	0	1	0	0
Sorteio de corte	0,59	0,61	0,38	0,28	0,88	0,98
Cromossomo gerador 2	0	0	1	1	0	1
Cromossomo gerado	0	0	0	1	0	1

Figura 5 — *Cruzamentos de dois cromossomos por meio de sorteio .* **Autoria própria**

1.3.5 Função de aptidão (Fitness)

Os algoritmos genéticos usam a função de aptidão, também conhecida como função fitness, para avaliar a qualidade de cada cromossomo em uma população. Ela atribui um valor numérico que mostra quão próximo um cromossomo está da solução desejada, avaliando o quão "bom" um cromossomo está em relação ao problema que está tentando resolver. Quando o valor da aptidão aumenta, o cromossomo é considerado mais assertivo para o problema. A função de aptidão pode ser simples (por exemplo, maximizar ou minimizar um valor numérico) ou complexa (por exemplo, envolvendo vários critérios de avaliação).

O algoritmo genético dirige o processo evolutivo usando o valor da aptidão. Os cromossomos mais aptos são mais propensos a serem selecionados para reprodução e podem combinar suas características ou genes para produzir novos cromossomos ou descendentes. A população tende a evoluir dessa forma ao longo das gerações, com cromossomos de alta qualidade se tornando mais comuns. Isso permite que o algoritmo explore eficazmente o espaço de soluções, melhorando continuamente a qualidade dos cromossomos até encontrar uma solução aceitável para o problema.

1.3.6 Mutação

A mutação no algoritmo genético é um processo essencial para manter a diversidade genética dentro da população ao longo das gerações. Ela ocorre quando os cromossomos de um ou mais indivíduos são alterados aleatoriamente, alterando um ou mais genes. Dependendo da representação dos cromossomos, essas mudanças podem incluir a inversão, substituição ou permutação de valores. O objetivo da mutação é evitar que a população se torne homogênea, evitando a convergência prematura das soluções. A mutação permite que o algoritmo explore novas áreas do espaço de soluções ao introduzir novas variações genéticas. Isso aumenta as chances de encontrar a solução ideal ou uma solução mais próxima da ideal.

Característica	A	B	C	D	E	F
Cromossomo Gerado	1	0	0	1	1	0
Mutação por inversão						
Cromossomo Mutado	1	1	1	0	0	0
Mutação por troca						
Cromossomo Mutado	1	0	1	0	1	0

Figura 6 — *Mutação de um cromossomo por meio de inversão e/ou troca* . **Autoria própria**

1.3.7 Critério de Parada

Em algoritmos genéticos, o critério de parada determina quando o algoritmo deve ser interrompido. Ele geralmente para após um número máximo de gerações, quando a melhoria na aptidão das pessoas se estabiliza, quando uma solução é encontrada ou após um tempo de execução pré-estabelecido. Além disso, pode parar se a diversidade da população diminuir significativamente, o que indica que a população ficou uniforme e menos capaz de aproveitar novas soluções. Isso ajuda a equilibrar a eficiência e a qualidade das soluções encontradas.

2.0 Aplicação no problema da mochila binária

2.1 Introdução

O problema da mochila, ou problema da mochila, é um desafio de otimização combinatória. É como encher uma mochila sem exceder um peso certo. Isso é feito para maximizar o valor dos itens incluídos. Richard Karp identificou este problema em 1972 como um dos 21 problemas NP completos. Apesar de sua formulação simples, encontrar uma solução é difícil; na maioria das vezes, isso requer o uso de algoritmos de programação dinâmica. Mas existem alternativas, como técnicas de algoritmos gulosos e procedimentos de separação e evolução (PSE). A análise neste artigo se concentra na utilização da metaheurística do algoritmo genético para descobrir soluções próximas.

O modelo é representado pelo modelo que é frequentemente encontrado em uma variedade de aplicações práticas que buscam soluções computacionais. Considere a existência de um conjunto de arquivos que devem ser gravados em um CD, e o objetivo é maximizar a distribuição desses arquivos para que cada disco ocupe o menor volume possível. Além desse exemplo, ele é usado em vários outros contextos, como investimento de capital, corte e empacotamento, carregamento de veículos e orçamento. Além disso, ele também foi usado na criação do algoritmo de criptografia de chave pública.

2.2 Aplicação do algoritmo

Para resolver o problema da mochila, o algoritmo genético envolve criar soluções iniciais, avaliar essas soluções e evoluir a população para encontrar soluções cada vez melhores. O processo segue passos específicos, como estabelecer uma população inicial, usar uma função objetivo para avaliar a qualidade das soluções e explorar novas oportunidades na área circundante.

No contexto da resolução do problema da mochila binária utilizando o algoritmo genético, o algoritmo será composto por:

- Gene: cada possível item dentro de uma determinada combinação da mochila binária, variando entre 0 (item não incluso na combinação) e 1 (item incluso na combinação).
- Cromossomos: vetores binários que contêm a configuração de uma possível combinação de mochila.
- População: conjunto de possíveis combinações da mochila binária analisada.

2.2.1 População Inicial

O algoritmo se inicia com a construção de uma solução inicial, ou seja, ele gera uma primeira geração de cromossomos da qual irá trabalhar a reprodução e evolução de cada um dos seus indivíduos. Essa população inicial pode ser gerada de maneira aleatória ou pelo uso de um algoritmo guloso, que irá gerar uma solução melhor do que a gerada aleatoriamente, acelerando o processo de evolução dos cromossomos.

2.2.2 Função de aptidão (fitness)

Após gerada a população inicial, é feita a função objetivo, que avalia a qualidade de todas as soluções na população. Essa função é responsável por calcular o valor total dos itens selecionados e garantir que o peso não exceda a capacidade da mochila no caso da mochila. A pontuação mais alta é atribuída a soluções que conseguem maximizar o valor do produto, mantendo o peso dentro do limite permitido.

2.2.3. Cruzamentos e Mutação

Tendo o vetor da população atual formado e ordenado conforme a sua aptidão com o problema (no caso da mochila o valor de cada solução), é iniciada a fase de reprodução da população. Para isso são selecionadas 3 classes dentro da população os cromossomos da elite (que possuem os melhores resultados), os cromossomos normais (que possuem resultados abaixo da média da elite) e o cromossomos da ralé (que possuem os piores resultados da população), sendo os tamanhos atribuídos a cada classe modificados de acordo com a necessidade do uso do algoritmo.

Os cruzamentos começam com os cromossomos da elite, eles serão cruzados entre si, par a par até gerar uma quantidade de cromossomos igual a original da elite.

Após isso, são cruzados os cromossomos normais, eles são cruzados sempre com um cromossomo aleatório da elite, até gerar um número de cromossomos igual ao original dos normais. Por fim, os cromossomos da ralé são completamente eliminados (para evitar resultados ruins) e são gerados novos cromossomos aleatoriamente.

2.2.4 Vizinhança utilizada

A "vizinhança utilizada", dentro do algoritmo genético, refere-se a aleatorização dos resultados, para evitar qualquer tipo de vício no algoritmo. Ela é feita de 2 maneiras diferentes: na troca dos cromossomos pertencentes a ralé, pois assim adicionam se novas combinações completamente diferentes das anteriormente geradas, e na mutação dos novos indivíduos gerados, onde os resultados são levemente alterados (troca de genes) nos cromossomos gerados, garantindo que os cromossomos normais e da elite não fiquem presos nos mesmos resultados.

2.2.3. Critério de parada

Por fim, ao término da geração de uma população nova, é verificado se a população cumpre o critério de aceitação, esse critério pode ser baseado em vários fatores, como o número máximo de gerações, onde o algoritmo pára após um número pré definido de iterações; a estagnação, que ocorre quando não há melhora significativa na aptidão das melhores soluções ao longo de várias gerações consecutivas; ou quando uma solução satisfatória é encontrada, isto é, uma solução que atenda ou supere um limite específico de aptidão previamente definido. Esse critério é essencial para garantir que o algoritmo não continue indefinidamente e que a solução obtida seja considerada suficientemente boa em termos de maximização do valor dos itens dentro das restrições de capacidade da mochila.

2.2.5 Estrutura de sequência do algoritmo

```
# Algoritmo Genético Mochila
// Inicialização
População <- GerarPopulaçãoInicialAleatória()
AvaliarPopulação(População)
DividirPopulação(População, Elite, Normais, Ralé)
// Loop principal
Enquanto (condição de parada não atingida) fazer
    NovaPopulação <- []
    // Reprodução Elite x Elite
    Para cada indivíduo da Elite[] fazer
    Filho <- Cruzamento(SelecionaPai(Elite), SelecionaPai(Elite))
    Filho <- Mutação(Filho)
  NovaPopulação.adicionar(Filho)
    Fim para
    // Reprodução Elite x Normais
    Para cada indivíduo da Normais[] fazer
    Filho <- Cruzamento(SelecionaPai(Elite), SelecionaPai(Normal))
    Filho <- Mutação(Filho)
  NovaPopulação.adicionar(Filho)
    Fim para
```

```
        //Substituição da Ralé por Aleatórios
        Para cada indivíduo na Ralé fazer
        Filho <- GerarIndividuoAleatorio()
      NovaPopulação.adicionar(Filho)
        Fim para
        // Avaliação da nova população
        AvaliarPopulação(NovaPopulação)
        DividirPopulação(NovaPopulação, Elite, Normais, Ralé)
        População <- NovaPopulação
Fim enquanto
// Retornar a melhor solução encontrada
Retornar MelhorIndivíduo(População)
Fim Algoritmo
```

2.3 Resultados obtidos

Os resultados do algoritmo genético fornecem soluções potenciais para o problema da mochila binária, indicando quais itens devem ser colocados na mochila para maximizar o valor total sem exceder a capacidade. Um cromossomo mostra todos os resultados, com cada gene representando um bit de zero ou um que representa um item específico. Um valor de 1 indica que o item está incluído na mochila, enquanto um valor de 0 indica que o item está fora da mochila. O algoritmo genético evolui esses cromossomos ao longo dos anos, combinando e mutando genes para explorar diferentes combinações de itens.

O objetivo final é encontrar o cromossomo que representa a combinação de itens que oferece o maior valor possível, mantendo a capacidade da mochila. Assim, os resultados são uma variedade de combinações de itens; a combinação que maximiza o valor total usando a capacidade mais pequena é a melhor solução.

2.3.1 Resultados obtidos em pesquisa externa

Para parametrização e entendimento do funcionamento e dos resultados obtidos pelo algoritmo genético, vamos demonstrar uma pesquisa feita e analisada pelo engenheiro de software Vitor Emanuel Batista em sua pesquisa “Algoritmo Genético para o Problema da Mochila” em 2020.

O espaço amostral usado, compreende os vetores de pesos e valores de cada item da mochila, como mostrado na figura 7, além do valor de peso máximo da mochila, sendo de 13Kg. Na mesma figura é possível observar a população inicial gerada na pesquisa com o uso da criação aleatória.

Item	1	2	3	4	5	6	7
Peso (kg)	6	3	1	7	4	2	5
Valor (R$)	2	7	3	4	5	2	6

População 01	Indivíduo 01	0	1	0	1	0	1	0
	Indivíduo 02	1	0	0	0	1	0	1
	Indivíduo 03	1	0	1	0	0	1	0

Peso (kg)	Valor (R$)
12	13
15	13
9	7

Figura 7 — Aplicação do algoritmo genético no problema da mochila. **Criado por: Vitor E. Batista, 2020**

O passo seguinte foi feito a avaliação de aptidão fitness na qual é calculado o peso e valor de cada indivíduo e verificando se atende os critérios definidos anteriormente. Após avaliação, é realizado o procedimento de seleção dos melhores indivíduos com o intuito de gerar uma nova população que possuirá suas características herdadas no processo de cruzamento (criação) da nova população.E no algoritmo em questão foram utilizados dois modelos diferentes de seleção, conforme descrito abaixo:

- **Indivíduo Único:** No procedimento *first_elite*, independentemente do número de indivíduos aptos, apenas o melhor é selecionado para gerar uma nova população.
- **Grupo de Indivíduos**: No procedimento *all_elite*, todos os indivíduos aptos são selecionados para formar uma nova população.

Após selecionar os melhores indivíduos para perpetuar para a próxima geração, é feito o cruzamento dos indivíduos restantes, utilizando todos os indivíduos que foram aceitos no critério de aceitação. Para o cruzamento foi utilizado o método de *crossover*, separando a primeira metade dos genes do indivíduo 1 e a segunda metade do indivíduo 2 para a formação de um novo indivíduo.

Para finalizar a criação da nova população, ele utilizou de dois métodos de mutação. O primeiro método foi o *fix_mutate*, onde o segundo quarto e o terceiro quarto do gene são alterados aleatoriamente (altera-se os *n* genes do meio do cromossomo, sendo $n = ½$). O segundo método foi selecionar cromossomos aleatórios e trocar o valor de um gene escolhido aleatoriamente por um número 0 ou 1, também escolhido aleatoriamente.

Isso marca o fim da criação da nova população, essa rotina será repetida até que o critério de parada seja aceito, no caso da pesquisa em questão é critério é um número de gerações criadas, de 5, 10, 15 e 20 gerações.

Como forma de avaliar o resultado o algoritmo foi posto a teste com outro algoritmo mas nesse caso o algoritmo segue o método da força bruta, que tem como premissa na verificação de todas as combinações possíveis na busca pelo conjunto ideal (possui maior valor e atenda às restrições de peso). Resumindo,trata-se de um algoritmo com complexidade (2^n), sendo “*n*” o número de itens disponíveis para alocação de mochila.

Search this file…

	Itens	Força Bruta (seg.)	Algoritmo Genético (seg.)	Gerações
2	5	0.005731	0.013566	5
3	10	0.229117	0.335321	161
4	15	5.376002	1.458588	611
5	20	165.451445	10.469887	3733

knapsack-results.csv hosted with ❤ by GitHub view raw

Tabela 1 — Comparativo de performance do algoritmo genético e de força bruta. **Criado por: Vitor E. Batista, 2020**

Com base nos resultados das comparações, encontrados acima, é possível observar que o algoritmo de força bruta sofre um aumento de tempo exponencial conforme o aumento do número de itens da mochila, o que já era esperado, dado que o algoritmo de força bruta é um algoritmo de complexidade de tempo exponencial. Já o algoritmo genético não se mostrou prejudicado pela quantidade de itens em relação ao tempo, porém o mesmo precisou de uma quantidade muito maior de gerações para chegar a um resultado ótimo.

Esses dados nos mostram que o algoritmo genético apresenta, em geral, um resultado superior ao algoritmo de força bruta, principalmente quanto maior forem os resultados.

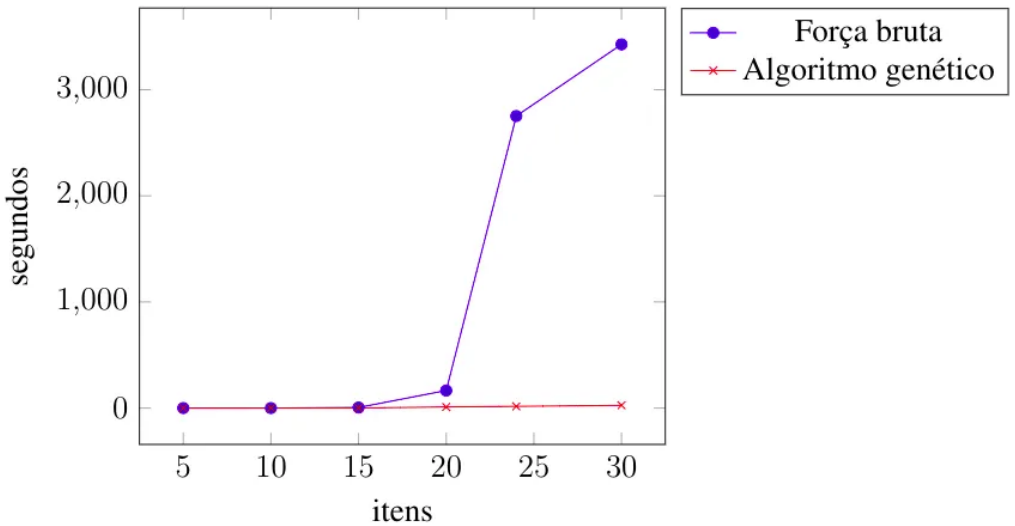

Figura 8 — Gráfico comparativo entre Força Bruta e o Algoritmo Genético. **Vitor E. Batista, 2020**

Como visto no gráfico da figura 8 , é notório que o algoritmo de força bruta se

distancia de forma exponencial do algoritmo genético na questão de duração de tempo, à medida que a população vai aumentando, isso demonstra a superioridade do algoritmo genético na obtenção de resultado em cenários com maior número de indivíduos/objetos.

Search this file…

	Itens	Força Bruta	Algoritmo Genético	Aproximação
2	5	306	306	100%
3	10	848	844.8	99.62%
4	15	1458	1444.8	99.09%
5	20	12585215	11688599	92.87%

knapsack-results2.csv hosted with ❤ by GitHub view raw

Tabela 2 — Aproximação do resultado ótimo pelo algoritmo genético. **Criado por Vitor E. Batista**

Em uma segunda análise foi comparada a aproximação dos resultados do algoritmo genético com o melhor resultado obtido pelo algoritmo de força bruta, nela pode-se observar que quanto maior a quantidade de itens da mochila, maior a distância entre o resultado encontrado e o resultado ótimo. De acordo com Vitor, isso se dá pela “calibração” do algoritmo, já que o número de gerações e indivíduos por geração se manteve fixo em todos os testes. Ainda segundo ele, seria necessário configurar esses valores como produtos do tamanho da lista de itens, o que aumentaria a complexidade do algoritmo.

Por fim, uma terceira análise, focada especificamente no algoritmo genético, foi baseada na distribuição da população em cada nova geração, visualizada através de um gráfico de dispersão. Os parâmetros de configuração do algoritmo genético foram os mesmos utilizados na análise anterior. No gráfico, o eixo x representa o peso da mochila, com uma linha destacando o peso máximo permitido, enquanto o eixo y indica o valor atribuído a cada solução específica:

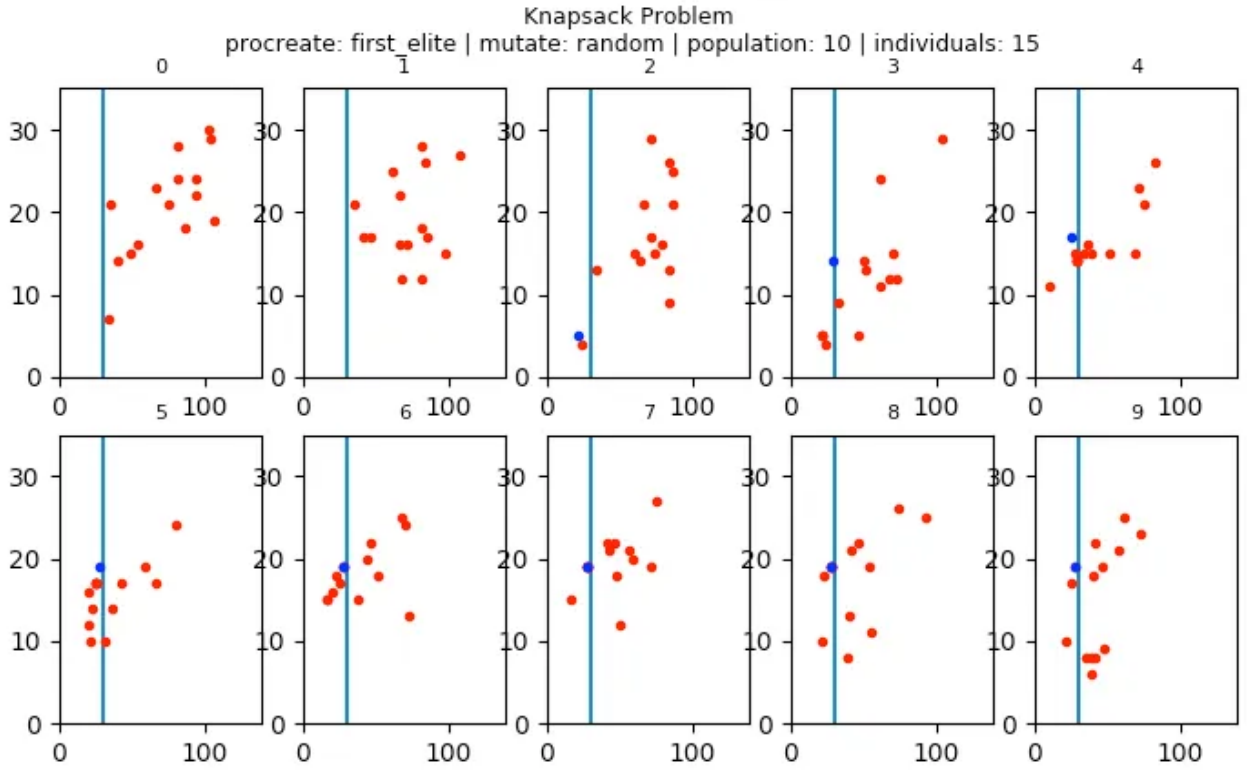

Figura 9 — Distribuição visual da população para gerações sucessivas. **Criado por Vitor E. Batista**

Os pontos azuis representam os indivíduos selecionados para a próxima geração. Nota-se uma tendência de aumento gradual do valor da melhor solução e aproximação do limite de peso, sem ultrapassá-lo. A diversidade da população se mantém através dos processos de cruzamento e mutação, evidenciando o funcionamento do algoritmo genético.

3.0 Referências

Algoritmos Genéticos. Disponível em: <https://sites.icmc.usp.br/andre/research/genetic/>. Acesso em: 25 ago. 2024.

Algoritmos Genéticos Aplicados Ao Problema Da Mochila Binária. Disponível em: <https://pt.scribd.com/document/715911860/Algoritmos-Geneticos-Aplicados-Ao-Problema-Da-Mochila-Binaria>. Acesso em: 25 ago. 2024.

BATISTA, V. E. **Algoritmo Genético para o Problema da Mochila** - Vitor Emanuel Batista. Disponível em: <https://vitorebatista.medium.com/algoritmo-gen%C3%A9tico-para-o-problema-da-mochila-5910f90f9488>. Acesso em: 25 ago. 2024.

CARVALHO, Rubens. **Problema da Mochila**. Disponível em: <https://www.ime.unicamp.br/~mac/db/2015-1S-122181-1.pdf>. Acesso em: 25 ago. 2024.

PACHECO, Marco Aurélio Cavalcanti. **Algoritmos genéticos: princípios e aplicações**. Disponível em: <https://inf.ufsc.br/~mauro.roisenberg/ine5377/Cursos-ICA/CE-intro_apost.pdf>. Acesso em: 25 ago. 2024.

DOS REIS, J. VON A. **Meta-heurística Algoritmo Genético aplicada no Problema da Mochila Binária**. Disponível em: <https://www.youtube.com/watch?v=ZCSbi3BnLB8>. Acesso em: 25 ago. 2024.

EDUARDO, P. **Algoritmo Genético E o Problema da Mochila – Trilhas Python**. Disponível em: <http://www.trilhaspython.com.br/2020/12/09/algoritmo-genetico-e-o-problema-da-mochila/>. Acesso em: 25 ago. 2024.

Genetic algorithms. Disponível em: <https://www.geeksforgeeks.org/genetic-algorithms/>. Acesso em: 25 ago. 2024.

ÓTIMO, Ponto **Algoritmo Genético aplicado ao Problema da Mochila.** Disponível em: <https://www.youtube.com/watch?v=FYF6lS_BHKA>. Acesso em: 25 ago. 2024.

SANTOS, Philippe Leal Freire dos. **"Heurísticas para o Problema da Partição Cromática de Custo Mínimo"**. 2018. Disponível em: https://site.ic.uff.br/wp-content/uploads/2021/09/872.pdf. Acesso em: 16 fev. 2023.