

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
PROGRAMA DE PÓS GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO

**Implementação do algoritmo genético para resolver problemas de
criptoaritmética.**

Elias de Abreu Domingos da Silva

Uberlândia
2020

1. ALGORITMO PARA O PROBLEMA SEND+MORE=MONEY) (ETAPA 1)

O problema SEND+MORE=MONEY é um exemplo de problema de criptoaritmética, o objetivo é atribuir um valor de 0 à 9 a cada uma das letras, não podendo atribuir o mesmo valor para letras diferentes e a mesma letra em locais distintos deve receber o mesmo valor. Ao atribuir os valores devem somar e verificar se o resultado em número ao serem mapeados para as letras será MONEY. Para solucionar o problema foi desenvolvido um algoritmo genético com as seguintes especificações: indivíduo vetor de tamanho 10, criação da população inicial aleatório não permitindo repetições, tamanho da população de 100, taxa de crossover 80%, 50 gerações e método de mutação troca de duas posições do vetor.

Utilizando as especificações acima foram realizadas as implementações com os métodos de seleções: Roleta (com inversão pelo valor do pior indivíduo a cada geração), Torneio simples (Torneio= 3) e Ranking linear. Os métodos implementados no crossover foram o parcialmente combinado (PMX) e o crossover cíclico. Na mutação foram realizados testes com 2%, 10% e 20%. Os métodos acima foram testados com reinserção da população ordenada e Reinserção pura com elitismo de 20%.

Com o objetivo de identificar as melhores configurações de cada etapa entre os métodos acima foram realizados cinco(5) testes com duzentas (200) execuções cada para as possibilidades e calculado a taxa de convergência (Total de execução que identificaram o resultado dividido pelo quantidade de execuções total) de cada uma delas. A linguagem de desenvolvimento utilizada para realizar o trabalho foi o Python na versão 3.7.

1.1 Taxa de mutação 2%

As tabelas abaixo apresenta os resultados aplicados a taxa de mutação 2% e com reinserção ordenada da população, método que pega os melhores entre pais e filhos.

Tabela 1 - Crossover cíclico e PMX com reinserção ordenada, mutação 2%

Crossover Cíclico				Crossover PMX			
Método	de	Convergência	Tempo	Método	de	Convergência	Tempo
Seleção				Seleção			
Torneio simples		34.5%	1 min20s	Torneio simples		65.5	54s
Rank Linear		32%	47s	Rank Linear		74.5%	42 s
Roleta		38.5%	56s	Roleta		72.5%	59s

Fonte: Próprio autor

Os resultados com a mesma taxa de mutação (2%) como o método de reinserção pura com elitismo de 20%, método que a cada geração passa para a próxima com os 80 filhos (taxa de crossover 0.8) e como 20 pais.

Tabela 2 - Crossover cíclico e PMX com reinserção pura com elitismo de 20%, mutação 2%

Crossover Cíclico				Crossover PMX			
Método de Seleção	de	Convergência	Tempo	Método de Seleção	de	Convergência	Tempo
Torneio simples		32%	1 min 27 s	Torneio simples		65.5	57 s
Rank Linear		34.5%	51s	Rank Linear		75.5%	56 s
Roleta		42%	52s	Roleta		87.5	45s

Fonte: Próprio autor

1.2 Taxa de mutação 10%

As tabelas abaixo apresenta os resultados aplicados a taxa de mutação 10% e com reinserção ordenada da população, método que selecionar os melhores entre pais e filhos.

Tabela 3 - Crossover cíclico e PMX com reinserção ordenada, mutação 10%

Crossover Cíclico				Crossover PMX			
Método de Seleção	de	Taxa de Convergência	Tempo	Método de Seleção	de	Convergência	Tempo
Torneio simples		46.5	1 min 16s	Torneio simples		68.5	49s
Rank Linear		54%	55s	Rank Linear		76%	35 s
Roleta		52%	59s	Roleta		75.5%	1 min 1s

Fonte: Próprio autor

Os resultados com a mesma taxa de mutação (10%) como o método de reinserção pura com elitismo de 20%, método que a cada geração passa para a próxima com os 80 filhos (taxa de crossover 0.8) e como 20 pais.

Tabela 4 - Crossover cíclico e PMX com Reinserção pura com elitismo de 20%, mutação 10%

Crossover Cíclico				Crossover PMX			
Método	de	Convergência	Tempo	Método	de	Convergência	Tempo
Seleção				Seleção			
Torneio simples		52.5%	1 min 24s	Torneio simples		79.5	1 min 3s
Rank Linear		62.5%	1 min 30s	Rank Linear		82.5%	1 min 14s
Roleta		75.5	57s	Roleta		89.5	47s

Fonte: Próprio autor

1.3 Taxa de mutação 20%

As tabelas abaixo apresenta os resultados aplicados a taxa de mutação 10% e com reinserção ordenada da população, método que pega os melhores entre pais e filhos.

Tabela 5 -Crossover cíclico e PMX com reinserção ordenada, mutação 20%

Crossover Cíclico				
Método	de	Taxa	de	Tempo
Seleção		Convergência		
Torneio simples		59%		1min 1s
Rank Linear		66%		59s
Roleta		63%		58s

Crossover PMX			
Método	de	Convergência	Tempo
Seleção			
Torneio simples		73%	57s
Rank Linear		79%	45s
Roleta		72.5	1 min 3s

Fonte: Próprio autor

Os resultados com a mesma taxa de mutação (10%) como o método de reinserção pura com elitismo de 20%, método que a cada geração passa para a próxima com os 80 filhos (taxa de crossover 0.8) e como 20 pais.

Tabela 6 - Crossover cíclico e PMX com Reinserção pura com elitismo de 20%, mutação20%

Crossover Cíclico				Crossover PMX			
Método	de	Convergência	Tempo	Método	de	Convergência	Tempo
Seleção				Seleção			
Torneio simples		73.5%	2 min 1s	Torneio simples		80.5	1 min 44
Rank Linear		81%	1 min 54s	Rank Linear		84%	1 min 50s
Roleta		82%	1 min 33s	Roleta		82.5	1 min 27s

Fonte: Próprio autor

2. VARIAÇÕES NO MODELO DO A.G. (ETAPA 2)

Após o fim dos experimentos anteriores notou-se que a melhor configuração obtida na fase anterior foi com o método de seleção roleta (investida pelo pior valor), método de crossover PMX, mutação 10% e a forma de reinserção pura com elitismo de 20%, a qual teve um desempenho médio de 47 segundos para as 200 execuções e obtendo uma taxa de convergência de 89.5%.

Desta forma tal configuração foi selecionada para prosseguir com o trabalho, tendo em vista que a taxa de convergência com maiores índices de acertos e o tempo de execução foi relativamente baixo em comparação com as outras configuração que obtiveram resultados próximos, exemplo a configuração com o método de seleção Rannk Linear, crossover PMX e reinserção pura com elitismo de 20% a qual obteve uma taxa de convergência de 84%, mas com o tempo de 1 minuto e 50 segundos para as 200 execuções.

2.1 Variações nos Números de Gerações

Com a finalidade de obter melhores resultados foram realizadas algumas alterações no número de gerações, foram realizados com 20,40,60,100 e 200, cada configuração foi testada 5 vezes com 200 execuções e ao fim calculado a média. A tabela a seguir apresenta os resultados obtidos:

Tabela 7 - Variações no número de gerações

Número de gerações	Taxa de convergência	Tempo (200 execuções)
20	51,5%	34,2 segundos
40	85,9%	44 segundos
60	91,7%	46,8 segundos
70	91,9%	46 segundos
100	91%	46,8 segundos
200	90.5%	50 segundos

Fonte: Próprio autor

Observando os resultados acima identificou-se que ao diminuir os números de gerações o tempo é reduzido, porém a taxa de convergência tem uma baixa considerável, deste modo os experimentos com o número abaixo de 50 gerações (experimento inicial) não são relevantes para o problema, mesmo diminuindo o tempo de execução. Por outro lado ao passo que aumenta, em quantidades não tão altas, o número de gerações a taxa de convergência adquiriu melhora e o tempo conservou-se, no primeiro experimento (com 50 gerações) a taxa de

convergência média foi de 89,5 com 47 segundos de execução, com as novos números a convergência chegou a 91,9 com 46 segundos.

Outra observação importante é quando há um aumento considerável no número de gerações, exemplo os testes com 100 e 200, a taxa de convergência ficou aparelhada com as com 60 e 70 gerações, porém existindo a possibilidade de aumento em alguns segundos no tempo de execução. Por fim, analisando os resultados notou-se que para o problema proposto o número de 70 gerações obteve melhores índices.

2.2 Repetições de Pais no mesmo Crossover

A implementação da primeira parte do trabalho há a possibilidade de repetição do mesmo pai na operação de crossover, desta forma caso entre indivíduo repetido na operação gerará dois filhos idênticos ao pai. Para avaliar se é pertinente evitar a repetição foram realizadas novas configurações no algoritmo genético, os experimentos foram realizados com as configurações consideradas mais pertinentes na primeira etapa, modificando apenas quantidade de gerações de 50 para 70 por motivo dos resultados da seção anterior.

Após a referida configuração foram realizados dez (10) novos testes e calculado a média, os resultados obtidos foram os seguintes:

Tabela 8 - Repetição de pais no crossover

	Taxa de Convergência	Tempo de Execução
Roleta com Repetições	91,9	46 segundos
Roleta sem Repetições	92,1	51,5 segundos
Roleta com três Tentativa	90,8	48,2 segundos

Fonte: Próprio autor

A tabela acima mostra que eliminando a repetição teve um pequeno aumento na taxa de convergência, porém a execução também teve um aumento, desta forma foi realizado uma nova configuração com o objetivo de verificar apenas três (3) vezes se é o mesmo indivíduo que foi selecionado, essa implementação evita que o laço de repetição fique em busca de um novo indivíduo por muito tempo, o resultado é apresentado na terceira linha.

2.3 Aplicação de dois Métodos de Seleção Diferentes

Com base na análise dos resultados obtidos na primeira etapa foi observado que alguns métodos de seleção têm uma pressão seletiva maior enquanto outros têm a pressão seletiva menor, por esse motivo foram realizados experimentos utilizando dois métodos de seleção diferentes, um para o primeiro pai outro para o segundo pai, com o objetivo de obter uma

pressão seletiva mediana. As configurações utilizadas foram: população de 100, 70 gerações, crossover PMX, mutação 10%, reinserção pura com elitismo de 20% e com três tentativas de selecionar indivíduos diferentes.

Tabela 9 - Combinação de métodos de seleção

Métodos de seleção	Taxa de convergência	Tempo
Roleta com Torneio Simples	82,2	1 min 2 45 segundos
Roleta com Rank Linear	88,6	47,2 segundos

Fonte: Próprio autor

Com base nos resultados obtidos identificou-se que a combinação de métodos de seleção para o problema SEND+MORE=MONEY não impulsionou melhoras, desta forma não prosseguimos com essa configuração.

2.4 Variações no Tamanho da População

Os experimentos realizados até este ponto do trabalho foram realizados com o tamanho da população com 100 indivíduos, com a finalidade de buscar novas configurações relevantes foram executados novos testes com os tamanhos de população de 20,50,80,100,120,150,180.

As configurações utilizadas foram: 70 gerações, método de seleção Roleta (investida pelo valor do pior indivíduo), taxa de crossover 0.8, crossover PMX, mutação 10%, reinserção pura com elitismo de 20% e com três tentativas de selecionar indivíduos diferentes. Os resultados são apresentados abaixo.

Tabela 10 - Tamanho da População

Tamanho da População	Taxa de Convergência	Tempo
20	89,3	44,2 segundos
50	90,4	42,6 segundos
80	91,1	44,2 segundos
100	91	44,4 segundos
120	92,1	48,4 segundos
150	91,7	49,4 segundos
180	90,9	49,6 segundos

Fonte: Próprio autor

Os resultados mostram que o tamanho da população afeta em pequenas frações na taxa de convergência e no tempo, identifica-se que com o aumento do tamanho o tempo de execução

também aumenta. Como os indivíduos são formados de números aleatórios a variação em pequenas quantidades é esperada, desta forma iremos prosseguir, mesmo com um pequeno aumento no tempo, com o tamanho da população com 120, o qual obteve uma pequena vantagem na taxa de convergência.

2.5 Taxa de Crossover e Reinscrição com Elitismo

A taxa de crossover, responsável para identificar a quantidade de filhos a cada geração, foi implementada em todos os experimentos com 80%, esta parte do trabalho destina-se a realizar novos experimentos realizando alterações na taxa de crossover em consequência na taxa de reinscrição pura com elitismo, a soma das duas taxas deve ser igual a 100%.

As configurações utilizadas foram: população de 120 indivíduos, 70 gerações, método de seleção Roleta (investida pelo valor do pior indivíduo), crossover PMX, mutação 10%. Os resultados são apresentados na tabela abaixo.

Tabela 11 - variação na taxa de crossover

Taxa de Crossover	Reinscrição	Convergência	Tempo
50%	50%	85,5%	39,8 segundos
70%	30%	89,3	39,4 segundos
90%	10%	89,2	1 min e 34 segundos

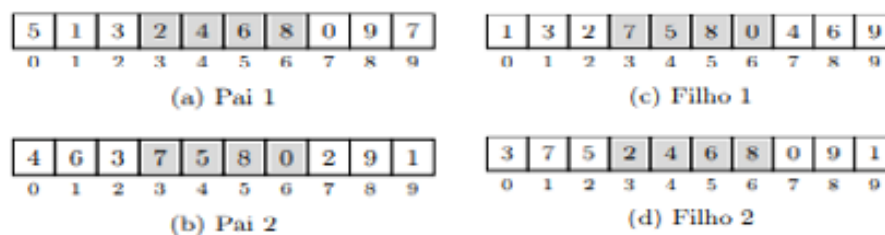
Fonte: Próprio autor

Analisando os dados acima constata que entre os parâmetros testados a melhor configuração para o problema proposto é a implementação inicial, taxa de crossover 80% e reinscrição com elitismo 20%, a qual obteve resultado de 92,1 de convergência com 48,4 segundos.

2.6 Crossover ORDER (OX)

O operador OX produz dois indivíduos filhos utilizando uma subsequência dos indivíduos pais. No exemplo abaixo, o Filho 1 recebe a subsequência do Pai 2 enquanto o Filho 2 recebe a subsequência do Pai 1. Após receber a subsequência, as outras posições dos vetores filhos são preenchidas com relação aos seus respectivos pais. Caso um elemento do pai já estiver no vetor do filho, o próximo elemento do vetor pai será considerado, respeitando a ordem em que aparece no indivíduo (DORNAS et al., 2017).

Figura 1 - Exemplo do cruzamento OX (Ordered Crossover)



Fonte: Adaptada de (DORNAS ET AL.,2017)

O método de crossover foi aplicado ao problema com as configurações descritas na seção 3.4 - população de 120, 70 gerações, método de seleção Roleta (investida pelo valor do pior indivíduo), taxa de crossover 0.8, mutação 10%, reinserção pura com elitismo de 20% e com três tentativas de selecionar indivíduos diferentes.

Tabela 12 - Crossover OX

Taxa de convergência	Tempo
83,5%	1 minuto 52 segundos

Fonte: Próprio autor

O resultado com o método foi de uma taxa de convergência de 83,5 com o tempo de 1 minuto e 53 segundos, sendo inferiores ao operador PMX, o qual com as mesmas configurações alcançou uma taxa de convergência de 92,1% com o tempo de 48,4 segundos.

2.7 Taxa de mutação 15%

Na primeira etapa foram realizados testes com a taxa de mutação de 2%, 10% e 20%, os melhores resultados foram com 10% (89.5% de convergência) e 20% (82.5%), ambos com o método de reinserção com elitismo de 20%. Com a finalidade de obter melhores resultados foram realizados novas execuções com a taxa de convergência de 15% (médias das duas melhores), os resultados foram 88.5% com o tempo de 1 minuto e 6 segundos, sendo inferior ao de 10%.

Tabela 13 - Mutação 15%

Taxa de convergência	Tempo
88,5%	1 minuto 6 segundos

Fonte: Próprio autor

2.8 Aplicação do Método de Seleção Torneio Estocástico

O método de seleção Torneio Estocástico escolhe um conjunto de indivíduo (do tamanho do Tuor) para participar do torneio, essa escolha é realizado com base na probabilidade da avaliação, em seguida o método seleciona o com maior aptidão dentro desse conjunto.

O método foi aplicado ao problema SEND+MORE=MONEY com a finalidade de comparar com o método Roleta (o qual tem maior taxa de convergência até o momento). As configurações aplicadas foram: população de 120, 70 gerações, Tuor de três (3), taxa de crossover 0.8, mutação 10%, reinserção pura com elitismo de 20% e com três tentativas de selecionar indivíduos diferentes. A taxa de convergência média obtida (5 etapas de 200 execuções) foi de 89.5 com o tempo médio de 1 minuto e 36 segundos, portanto resultados inferiores ao método roleta.

Tabela 14 - Torneio Estocástico

Taxa de convergência	Tempo
89,5%	1 minuto 36 segundos

Fonte: Próprio autor

2.9 Verificação antes de aplicar a mutação

O algoritmo executado nas seções anteriores não verifica se o indivíduo é o resultado antes de aplicar a mutação, por este motivo foi realizado uma nova configuração para que antes de fazer a mutação verificar se ele é ou não o resultado, caso seja o resultado a execução terminará e incrementará a contagem de convergido, ao contrário será executado a mutação. O resultado alcançou uma taxa de convergência de 99% com o tempo de 48,2 segundos, em comparação com o resultado anterior de 92,1 em 48,8 segundos, desta forma a configuração será mantida nas próximas etapas.

Tabela 15 - Verificação antes da mutação

Taxa de convergência	Tempo
48,2 segundos	99%

Fonte: Próprio autor

2.10 Considerações da Etapa 2

Analisando todos as configurações realizadas nesta etapa conclui-se que a melhor configuração obtida foi com a população de 120 indivíduo, 70 gerações, taxa de crossover 0.8, método de seleção Roleta invertida pelo pior valor e com três tentativas de selecionar indivíduos

diferentes, mutação 10%, reinserção pura com elitismo de 20% e a verificação antes de aplicar a mutação, a qual conseguiu uma taxa de convergência de 99% com 48,8 segundos.

Ao compararmos o resultados obtidos nessa etapa com o melhor resultado obtido na etapa anterior percebemos que houve ganhos consideráveis na taxa de convergência, a qual na etapa 1 foi de 89,5 e a aumentou ao fim da etapa 2 para 99%. O tempo de execução também teve um pequeno aumento, na etapa 1 foi de 47 segundos e na etapa dois chegou a 48,8 segundos. A convergência teve um aumento de 9.5% com as configurações, portanto conclui-se que as configurações foram relevantes para o trabalho, mesmo com o aumento do tempo em 1,8 segundos.

3. APLICAÇÃO DO ALGORITMO EM OUTROS PROBLEMAS DE CRIPTOARITMÉTICA (ETAPA 3)

O algoritmo com as melhores configurações obtidas na fase anterior foi aplicado aos outros quatros (4) problemas propostos de criptoaritmética, os resultados são expostos a seguir:

Tabela 16 - Aplicação do Algoritmo aos outros problemas

Problema	Convergência	Tempo
EAT+THAT=APPLE	29,8%	2min 27s
CROSS+ROADS=DANGER	1,3%	2 min 47s
COCA+COLA=OASIS	20,3	2 min 21s
DONALD+GERALD=ROBERT	7,4%	2 min 33s

Fonte: Próprio autor

Observando os dados acima percebe-se que o algoritmo, embora obtendo boa eficiência para o problema de SEND+MORE+MONEY, não obteve resultados positivos para os outros problemas, os quais tiveram baixas na convergência e aumentos no tempo.

3.1 Variação no Tamanho da População

Os resultados com os outros problemas apresentaram baixa taxa de convergência e o tempo elevado, como tentativa de diminuir o tempo de execução e buscando melhoras na convergência foram realizados novos testes com o tamanho da população inferior a 120 (melhor execução da etapa anterior). A primeira tabela apresenta os resultados com o tamanho da população igual a 100 e a segunda com o tamanho de 80, foram executas cinco (5) vezes com 200 execuções cada configuração e calculado a média.

Tabela 17 - População com 100 indivíduos

Problema	Convergência	Tempo
EAT+THAT=APPLE	25,8%	2 minutos 12 segundos
CROSS+ROADS=DANGER	0,9%	2 minutos 36 segundos
COCA+COLA=OASIS	17,5%	2 minutos 18 segundos
DONALD+GERALD=ROBERT	8,9%	2 minutos 39 segundos
SEND+MORE = MONEY	98,5	45,4 segundos

Fonte: Próprio autor

Tabela 18 - População com 80 indivíduos

Problema	Convergência	Tempo
EAT+THAT=APPLE	24,9	2 minutos 9 segundos
CROSS+ROADS=DANGER	1,1%	2 minutos 37 segundos
COCA+COLA=OASIS	17,5%	2 minutos 13 segundos
DONALD+GERALD=ROBERT	9,3%	2 minutos 34 segundos
SEND+MORE = MONEY	98,1	43,6 segundos

Fonte: Próprio autor

Os resultados demonstra que com o tamanho de população de 100 há uma baixa no tempo, porém há também uma pequena baixa na convergência, o mesmo ocorre com o tamanho de 80 em comparação os demais.

3.2 Taxa de mutação e Número de Gerações

Como a taxa de mutação de 20% teve alguns resultados até melhores que alguns de 10% na primeira etapa, embora o melhor foi de 10%, os cinco (5) problemas foram expostos a tal configuração (mutação de 20%). Cada problema foi testado cinco (5) vezes com as 200 execuções.

Tabela 19 - Mutação de 20% com 70 gerações

Problema	Convergência	Tempo
EAT+THAT=APPLE	31,2%	3 minutos 38 segundos
CROSS+ROADS=DANGER	2%	4 minutos 36 segundos
COCA+COLA=OASIS	26,4%	3 minutos 39 segundos
DONALD+GERALD=ROBERT	10,8%	3 minutos 57 segundos
SEND+MORE = MONEY	99,3%	1 minuto 12 segundos

Fonte: Próprio autor

A taxa de mutação de 20% apresentou melhoras na convergência, porém o tempo teve acréscimo considerável, para tentar equilibrar foram realizados novos experimentos com a taxa de 20%, mas com alteração no número de gerações, o qual estava com 70 e foi alterada para 50.

Tabela 20 - Mutação de 20 com 50 gerações

Problema	Convergência	Tempo
EAT+THAT=APPLE	27,7%	2 minutos 9 segundos
CROSS+ROADS=DANGER	1,3%	2 minutos 31 segundos
COCA+COLA=OASIS	23%	2 minutos 9 segundos
DONALD+GERALD=ROBERT	7%	2 minutos 26 segundos
SEND+MORE = MONEY	98,7	56,6 segundos

Fonte: Próprio autor

Os resultados demonstram que houve melhoras consideráveis no tempo, tanto na comparação com a população de 120, 70 gerações e mutação de 20% quanto na comparação com os resultados iniciais (população de 120, taxa de mutação 10% com 70 gerações), em contra partida teve uma baixa pequena na taxa de convergência comparado com ambos.

3.3 Crossover Cíclico

O método crossover cíclico foi aplicado aos problemas, no entanto não obteve melhoras em comparação ao PMX, assim prosseguiremos com o método PMX.

Tabela 21 - Aplicação do crossover cíclico nos cinco problemas

Problema	Convergência	Tempo
EAT+THAT=APPLE	14,5%	2 minutos 8 segundos
CROSS+ROADS=DANGER	0,4	2 minutos 29 segundos
COCA+COLA=OASIS	15,5	2 minutos 30 segundos
DONALD+GERALD=ROBERT	3,1%	2 minutos 28 segundos
SEND+MORE = MONEY	87,8	1 minuto 3 segundos

Fonte: Próprio autor

3.4 Método de seleção Rank Linear

O método de seleção Rank Linear obteve bons resultados na primeira etapa, embora abaixo da Roleta, para verificar a eficiência nos outros problemas foram realizados cinco (5) testes com 200 execuções, os resultados são apresentados abaixo.

Tabela 22 - Aplicação do método de seleção Rank Linear aos problemas

Problema	Convergência	Tempo
EAT+THAT=APPLE	16,1%	3 minutos 6 segundos
CROSS+ROADS=DANGER	1,2	3 minutos 47 segundos

COCA+COLA=OASIS	15,6%	3 minutos 6 segundos
DONALD+GERALD=ROBERT	5,3%	3 minutos 31 segundos
SEND+MORE = MONEY	95,4%	52,2 segundos

Fonte: Próprio autor

3.5 Reinscrição Ordenada

A reinscrição apresentou uma melhora considerável no tempo para todos os problemas, todavia a taxa de convergência também teve uma baixa, dessa forma a reinscrição com elitismo continuará nas próximas configurações.

Tabela 23 - Reinscrição ordenada

Problema	Convergência	Tempo
EAT+THAT=APPLE	12,4%	1 minuto 6 segundos
CROSS+ROADS=DANGER	0,5	1 minuto 37 segundos
COCA+COLA=OASIS	11,5%	57,2 segundos
DONALD+GERALD=ROBERT	2,9%	1 minuto 16 segundos
SEND+MORE = MONEY	86,2%	30,2 segundos

Fonte: Próprio autor

3.6 Taxa de Crossover

A taxa de crossover original de 0.8 com reinscrição pura com elitismo de 20% obteve os melhores resultados para o problema SEND+MORE=MONEY, porém para os outros não obteve a mesma performance, por este motivo foram realizados novos testes com uma taxa de crossover menor e outra maior. Os resultados são expostos a seguir.

Tabela 24 - Taxa de crossover 70% com reinscrição pura com elitismo de 30%

Problema	Convergência	Tempo
EAT+THAT=APPLE	22,5%	1 minutos 42 segundos
CROSS+ROADS=DANGER	1,8%	2 minutos 2 segundos
COCA+COLA=OASIS	18,6%	1 minuto 40 segundos
DONALD+GERALD=ROBERT	4,9%	1 minutos 56 segundos
SEND+MORE = MONEY	96,1%	47,8 segundos

Fonte: Próprio autor

Tabela 25 - Taxa de crossover 90% com reinserção pura com elitismo de 10%

Problema	Convergência	Tempo
EAT+THAT=APPLE	29,8%	3 minutos 58 segundos
CROSS+ROADS=DANGER	1,3	4 minutos 19 segundos
COCA+COLA=OASIS	28,5%	3 minutos 41 segundos
DONALD+GERALD=ROBERT	8,2%	4 minutos 9 segundos
SEND+MORE = MONEY	98,4%	1 minuto 30 segundos

Fonte: Próprio autor

A taxa de crossover de 90% obteve melhoras na convergência de alguns dos problemas, entretanto o tempo de execução teve, na maioria, aumento acima de 50%, já a taxa crossover de 70% obteve melhoras no tempo, porém a taxa de convergência teve baixa considerável, dessa forma nenhuma das configurações serão mantidas.

3.7. Método de Mutação (Troca pelo Próximo)

O método de mutação utilizado até essa parte do trabalho permuta dois genes, as posições são sorteadas aleatórias, foram realizadas algumas pesquisas e não foi encontrada nenhuma forma de mutação que troca apenas o genes com o da próxima posição, desta forma foi construído um método que sorteia uma posição e troca o genes com o da próxima posição, caso o sorteia a última posição ela será trocada pela posição zero. Os resultados não foram favoráveis, como são apresentados na tabela abaixo.

Tabela 26 - Método de mutação que realiza a troca pela próxima posição

Problema	Convergência	Tempo
EAT+THAT=APPLE	24,4%	2 minutos 10 segundos
CROSS+ROADS=DANGER	1,3	2 minutos 37 segundos
COCA+COLA=OASIS	21,6%	2 minutos 20 segundos
DONALD+GERALD=ROBERT	7,1%	2 minutos 26 segundos
SEND+MORE = MONEY	98%	1 minuto 2 segundos

Fonte: Próprio autor

3.8 Mutação Apenas nos Melhores Avaliados

Nas etapas anteriores as escolhas dos indivíduos para a aplicação da mutação foram realizadas de forma aleatória, pensando em dar prioridades aos indivíduos com melhores aptidão foi realizado uma nova configuração que realiza a mutação apenas nos indivíduos com

melhores avaliações, o primeiro elemento a sofrer a mutação é o melhor classificado, depois o segundo e assim sucessivamente até atingir a taxa de mutação.

Tabela 27 - Mutação apenas nos melhores avaliados

Problema	Convergência	Tempo
EAT+THAT=APPLE	20,3%	2 minutos 29 segundos
CROSS+ROADS=DANGER	0,8%	2 minutos 53 segundos
COCA+COLA=OASIS	16,6%	2 minutos 23 segundos
DONALD+GERALD=ROBERT	5,6%	2 minutos 37 segundos
SEND+MORE = MONEY	95,8	56,4 segundos

Fonte: Próprio autor

Os resultados apresentam que a configuração não é adequada para os problemas, sendo assim não será prosseguida com a mesma.

3.9 Mutação por Inversão Simples (SIM)

O operador de mutação simples (simple inversion operator - SIM) seleciona aleatoriamente dois pontos de corte, e reverte a sequência entre os pontos (MALAQUILAS, 2006. p.65). Exemplo o cromossomo 9, 6, 2, 3, 5, 1, 8, 7, 4, 0 e o primeiro ponto de corte é 2 e o segundo 5, a mutação resultaria no cromossomo 9,6,1,5,3,2,8,7,4,0.

Figura 2 - Mutação por Inversão Simples (SIM)



Fonte: MALAQUIAS 2006

Tabela 28 - Mutação por inversão simples (SIM)

Problema	Convergência	Tempo
EAT+THAT=APPLE	33%	2 minutos 51 segundos
CROSS+ROADS=DANGER	1,3%	3 minutos 17 segundos
COCA+COLA=OASIS	24,7	2 minutos 47 segundos
DONALD+GERALD=ROBERT	8%	3 minutos 9 segundos
SEND+MORE = MONEY	98,6%	1 minuto 6 segundos

Fonte: Próprio autor

3.10 Considerações da Etapa 3

Analisando os dados obtidos em cada configuração conclui-se que a única configuração que obteve vantagem foi a mutação por inversão simples, onde a taxa de convergência teve um aumento, o aspecto ruim dessa configuração é o tempo, pois também teve um aumento, porém não alcançou 50% a mais do que os experimentos do início da fase. Os resultados apresentados na tabela 15 também tiveram melhoras, em dois dos problemas até uma convergência maior do que a configuração escolhida, porém em alguns casos o tempo aumentou em mais de 50%. Portanto a configuração escolhida é: população 120, 70 gerações, método de seleção Roleta, crossover PMX, taxa de crossover 0.8, reinserção pura com elitismo de 20% e mutação por inversão simples.

4. CONCLUSÕES

O algoritmo desenvolvido no presente trabalho obteve bons resultados na solução do problema $SEND + MODE = MONEY$, atingindo uma taxa de acerto em torno de 99%, contudo quando aplicado aos outros problemas não obteve a mesma performance, pois em todos os casos a convergência ficou abaixo de 40% e tendo caso que não chegou a 5%. A criação de uma nova função de avaliação (Etapa 4) poderia solucionar os problemas, entretanto a elaboração dessa função não foi alcançada.

O tempo de execução é outro aspecto a se levar em consideração, no primeiro problema atingiu por volta de 50 segundos para as 200 execuções, o qual já é um tempo elevado, e quando aplicado aos outros problemas esse tempo chegou a atingir 3 minutos e 17 segundos. Portanto conclui-se que o algoritmo não é um bom revolvedor de criptaritmética.

REFERÊNCIAS

A.J. UMBARKAR¹ e P.D. SHETH. **Crossover Operators in Genetic Algorithms: A Review**. International Journal of Computer Applications, 2017. Disponível em: https://www.researchgate.net/publication/288749263_CROSSOVER_OPERATORS_IN_GENETIC_ALGORITHMS_A_REVIEW. Acesso em: 15 de setembro de 2020.

DORNAS et al. **Algoritmo genético para o Problema de Dimensionamento de Lotes Multi-item Capacitado**. XIII Brazilian Congress on Computational Intelligence, Rio de Janeiro, Brazil from 30 October to 1 November 2017. Disponível em: <http://cbic2017.org/papers/cbic-paper-100.pdf>. Acesso em 22 de setembro de 2020:

MALAGUIAS, N. G. L. **Uso de Algoritmo Genético para a Otimização de Rotas de Distribuição**. 2006. 113 f. Dissertação (Mestrado em Engenharia Elétrica) - Faculdade de Engenharia Elétrica, Universidade Federal de Uberlândia, Uberlândia, 2006. Disponível em: <https://repositorio.ufu.br/bitstream/123456789/14632/1/NGLMalaquiasDISPRT.pdf>. Acesso em: 26 de setembro.