

Aluno: Carla Sanches Nere dos Santos – 15.1.8349

Atividade: Resolução da Prova 1

## 1. Escolha dos algoritmos

Para a criação do algoritmo híbrido, foi utilizado como base o Algoritmo Genético e adicionadas algumas características do Differential Evolution. A justificativa é a de que os dois algoritmos foram os que apresentaram os melhores resultados nas atividades aplicadas em sala. A tentativa foi a de juntar duas vantagens destes algoritmos: o tempo de execução do AG e os bons resultados que o DE proporciona.

## 2. Pseudo-código do algoritmo

```
inicializar(populacao);  
avaliar(populacao);  
melhorSolucao = melhorIndividuo(populacao);  
  
para cada geracao faça  
    para cada individuo da populacao faça  
        selecionar(progenitor1);  
        selecionar(progenitor2);  
        inicializar(descendente);  
  
        descendente = crossover(progenitor1, progenitor2);  
  
        melhorProgenitor = selecionar(progenitor1, progenitor2);  
  
        mutacao(melhorProgenitor, descendente);  
  
        avaliar(descendente);  
    fim para;  
  
    novaPopulacao();  
    definirPopSobrevivente();  
  
    se(melhorIndividuo(novaPopulacao) < melhorSolucao) entao  
        melhorSolucao = melhorIndividuo(novaPopulacao);  
    fim se;  
fim para;  
  
retorne melhorSolucao;
```

Para chegar a este algoritmo, foi combinado o método de crossover do AG com o método de mutação do DE, fazendo as adaptações necessárias. Também foi utilizada a estratégia do DE de sempre adicionar na lista de sobreviventes o melhor indivíduo da geração.

### 3. Casos de Teste

Os casos de teste foram realizados utilizando a seguinte configuração:

- Função de avaliação: *Schwefel's function 7*;
- Variáveis:  $n = 50$ ;
- Representação: Real;
- Inicialização da população: aleatória;
- Tamanho da população: 100 indivíduos;
- Critério de parada: número de gerações = 300;
- Valor mínimo: -500;
- Valor máximo: 500;
- Execuções por caso de teste: 30 execuções;
- Taxa de crossover: 1.0;
- Taxa de mutação: 0.84 – Teste 1 e 0.00001 – Teste 2.

Com esta configuração, a convergência mais rápida ocorreu na geração 173 para a taxa de mutação de 0.84. A partir dos resultados obtidos na saída, representados no gráfico abaixo, foi possível observar que valores de crossover e mutação mais altos geravam melhores resultados.

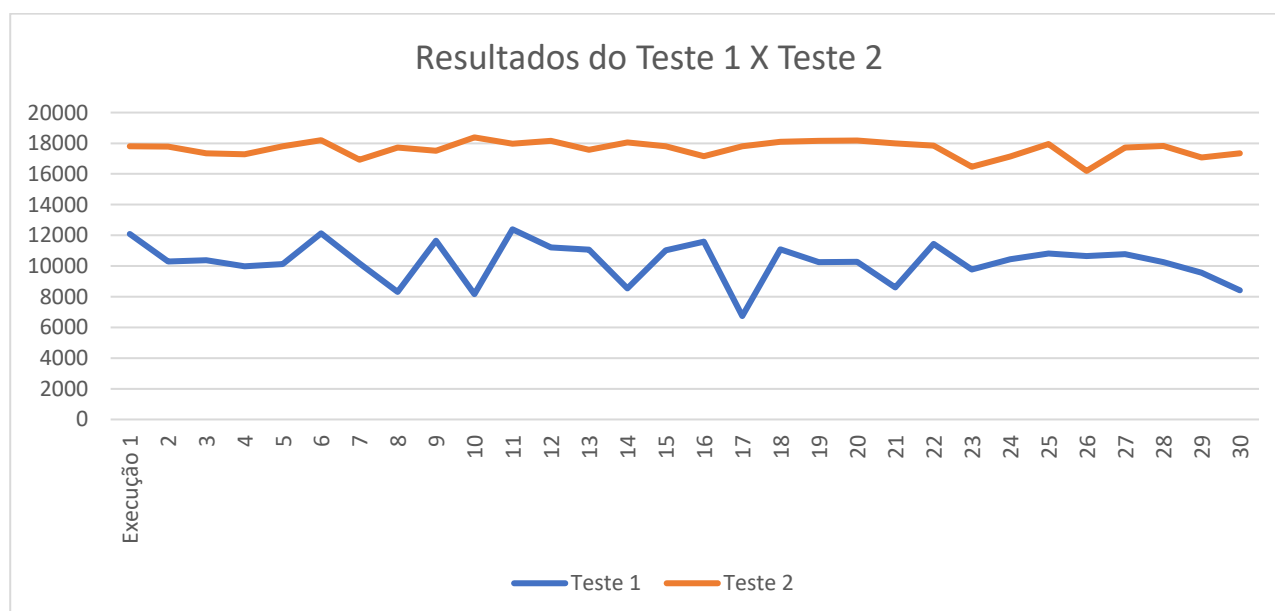


Figura 1 - Resultados dos casos de teste do algoritmo híbrido

Também foram realizados testes utilizando a função de avaliação proposta nos algoritmos AG e DE a fim de comparar os resultados com o algoritmo híbrido e verificar sua eficácia. Foram utilizadas as mesmas configurações recomendadas, modificando somente as taxas de mutação e crossover para a obtenção dos melhores resultados.

A Figura 2 apresenta a comparação dos 30 melhores resultados de cada algoritmo. As configurações para o Algoritmo Genético foram uma taxa de mutação de 0.001 e taxa de crossover de 0.013. Para o Differential Evolution foi utilizada uma taxa de mutação de 0.000035 e taxa de crossover de 1.0.

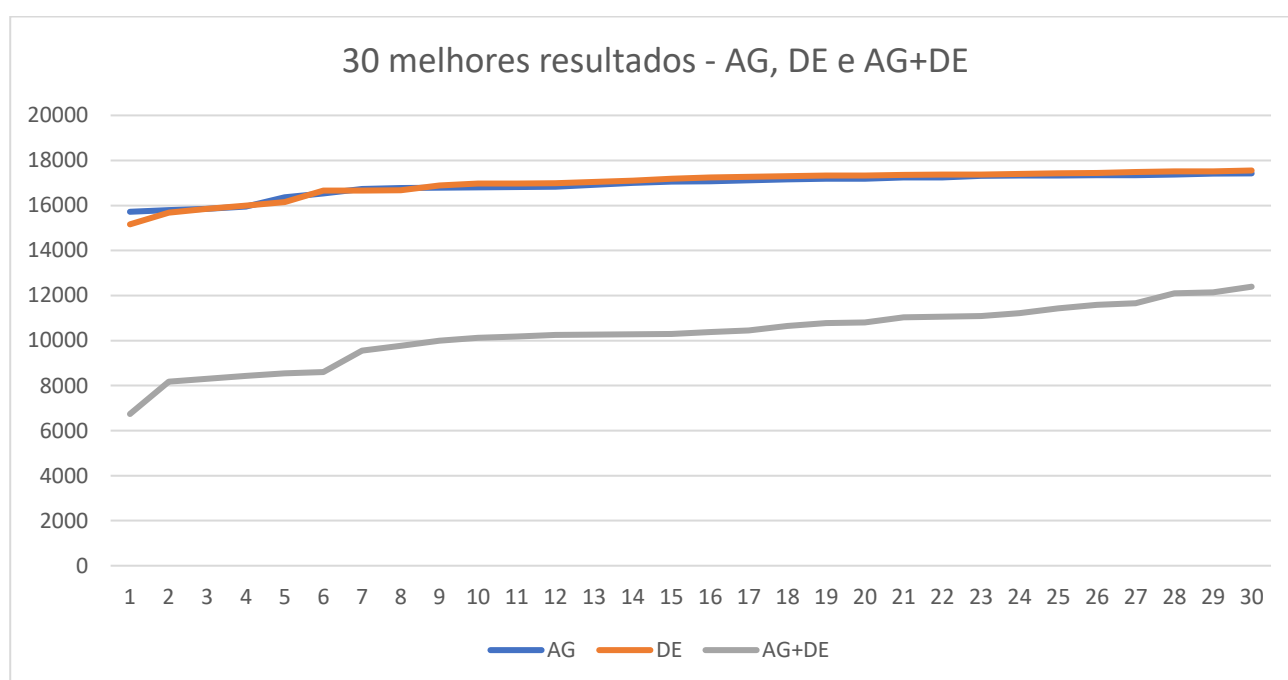


Figura 2 - comparação dos 30 melhores resultados para os algoritmos AG, DE e híbrido.

A Figura 3 apresenta a comparação do tempo de execução em milissegundos dos três algoritmos para os 30 melhores resultados de cada um. Apesar de o AG apresentar tempos bem menores, o algoritmo híbrido foi executado mais rapidamente que o DE.

#### 4. Considerações Finais

Ao combinar métodos do Algoritmo Genético com o Differential Evolution, fazendo as adaptações necessárias, foi possível obter bons resultados. O algoritmo se mostrou eficiente para o cenário proposto ao apresentar resultados melhores em um tempo de execução razoável. O melhor resultado obtido foi de 6738.269, o pior resultado foi 18381.46 e o tempo médio de execução foi de 152.9667 ms.

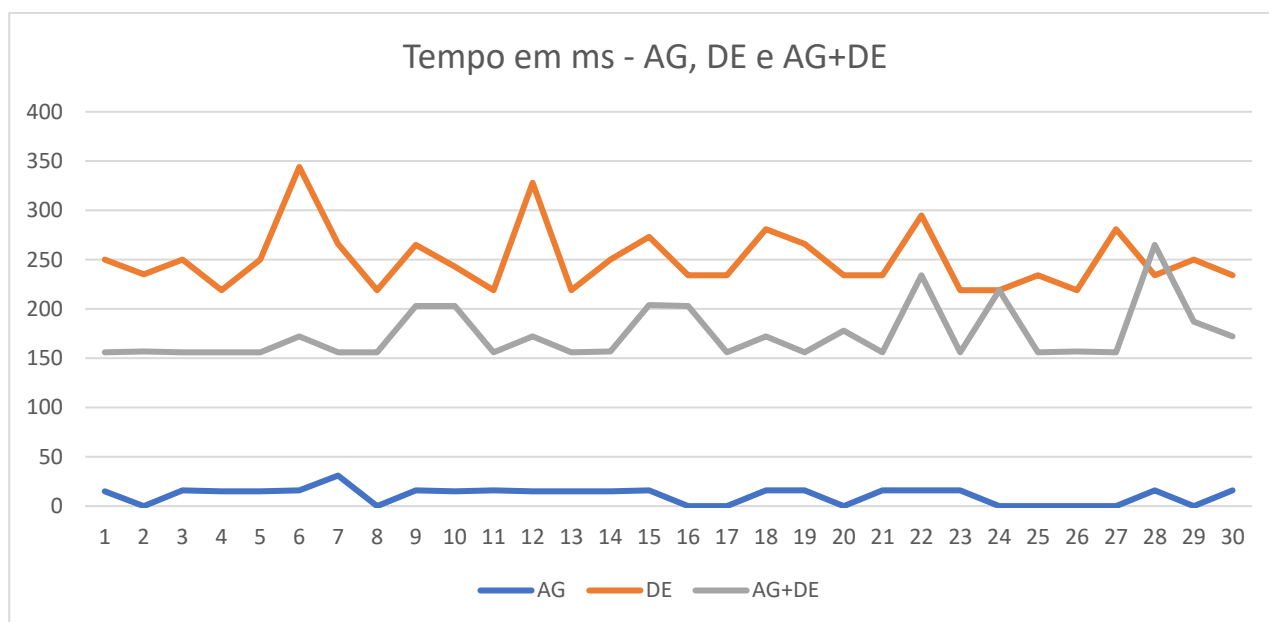


Figura 3 - Tempo de execução em ms para os 30 melhores resultados dos algoritmos AG, DE e híbrido.