

Projeto de Programação - Otimização de Algoritmos

Leandro Baêta Lustosa Pontes

Prof. Dr. Filipe Wall Mutz

Pesquisa em Computação Aplicada

IFES, Serra, 11/07/2020

1 Problema 1

Para o problema 1 observamos um empate técnico entre os algoritmos Hill-Climbing com restart e o Genetic Algorithm, ambos apresentaram resultados muitos contudentes e encontrando o mínimo global de -10, bem como o Simulated Annealing, no entanto esses dois tiveram 100% das suas execuções terminando abaixo de -9, com um desvio padrão um pouco abaixo do 0.2.

Acredito que o Algoritmo Genético se destaque pela a sua inicialização, que leva ampla vantagem sobre os outros algoritmos, mas que eu entendo como um benefício do algoritmo. Para se ter uma ideia em quanto outros algoritmos iniciam com o valor da função objetivo próximo do máximo global, que é de 105.0366, enquanto o GA tem o seu gráfico de melhor valor, já iniciando abaixo de 0 é um contraste muito grande. Outra vantagem é a rápida convergência dos indivíduos da população para o mínimo global, podemos notar que isso ocorre em geral antes da iteração 100 como podemos verificar na figura 6.

O algoritmo Hill-Climbing com restart é muito eficiente em sua estratégia, pois a cada x iterações, nesse problema a cada 50 iterações ocorre um restart, e com esse reinício certas vezes de fato bem melhores do que as posições anteriores, outras vezes encontra melhores caminhos mesmo com um valor um pouco maior do que o anterior e por muitas vezes também verificamos valores muito piores do que os anteriores, como podemos verificar na figura 3, mas é exatamente essa estratégia que dá tanto destaque para esse algoritmo.

O Simulated Annealing é um algoritmo nervoso que no início aceita muitos valores piores na perspectiva de encontrar um melhor ponto na busca por um mínimo global. Nas dez execuções encontramos apenas 2 valores acima de 0 e mesmo assim são valores muito próximos de 0, ou seja eles não ficaram tão presos a mínimos locais em relação a esse problema. Mas aqui como no Hill-Climbing com restart, vale o destaque para o gráfico de variação do valor atual, para notarmos o quão volátil é até a iteração 810/820 como podemos verificar na figura 5.

O algoritmo Hill-Climbing apresenta grande velocidade na sua execução, pois é um algoritmo bem simples, com as perturbações nos dados sendo geradas com um desvio padrão de 2.0, para esse problema de minimização, verificamos que a maioria das execuções tendem a ficar abaixo de zero e cerca de metade das execuções conseguiram um resultado bem próximo do mínimo global, que é -10. Mas também notamos que muitas vezes o algoritmo fica preso em mínimos locais, nesse último estudo que usamos como referência foram 4 execuções que ficaram presas, com destaque para o valor máximo de 37.798.

1.1 Algoritmo Hill-Climbing

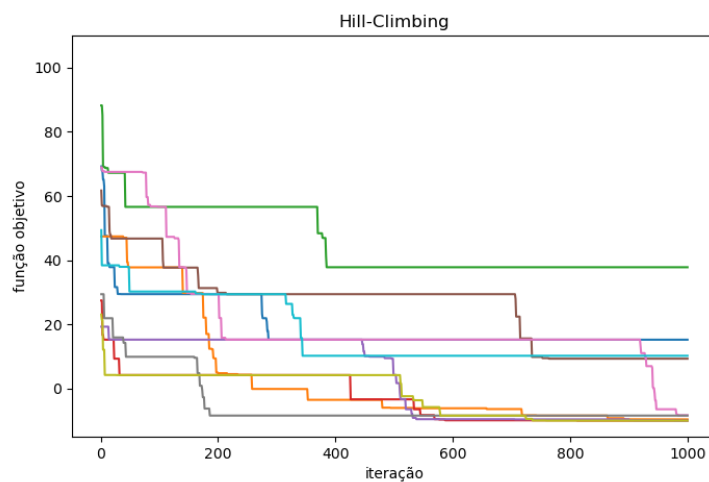


Figure 1: Dados da execução da função objetivo durante as 10 iterações.

1.2 Algoritmo Hill-Climbing com Restart

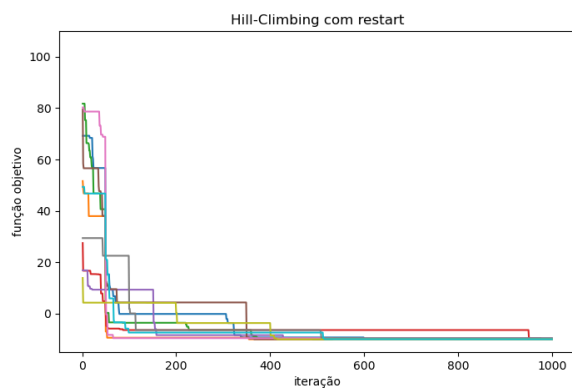


Figure 2: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

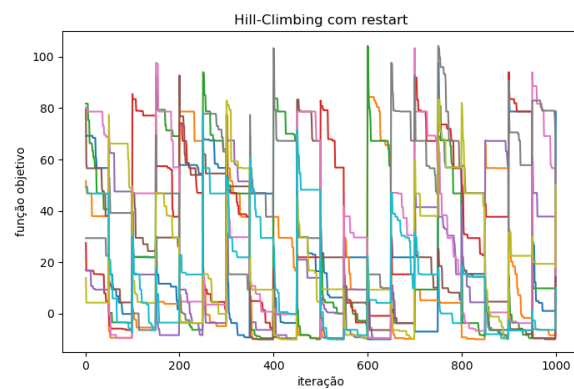


Figure 3: Dados da execução da função objetivo durante as 10 iterações por valor atual.

1.3 Simulated Annealing

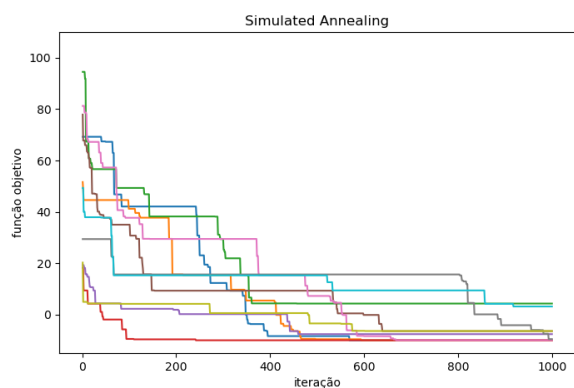


Figure 4: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

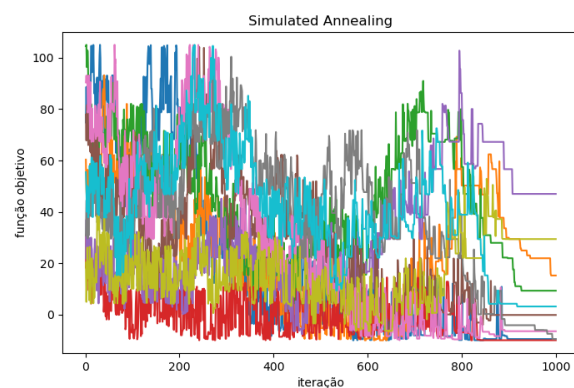


Figure 5: Dados da execução da função objetivo durante as 10 iterações por valor atual.

1.4 Algoritmo Genético

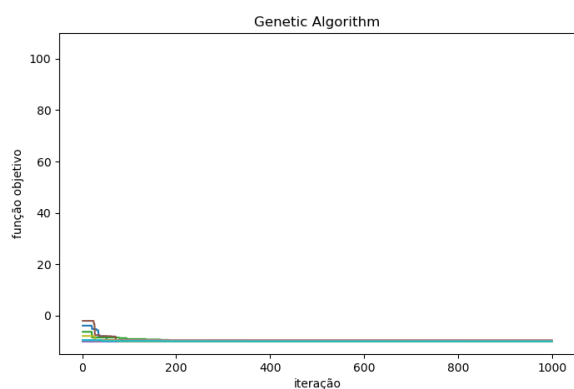


Figure 6: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

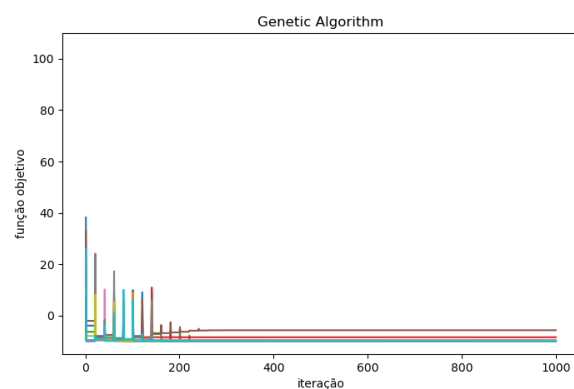


Figure 7: Dados da execução da função objetivo durante as 10 iterações por valor atual.

1.5 Resumo

Algoritmo	Max	Min	Média	Desvio Padrão
Hill-Climbing	37.798	-9.992	1.623	16.181
Hill-Climbing com restart	-9.589	-10.0	-9.91	0.169
Simulated Annealing	4.3	-10.0	-6.245	5.463
Genetic Algorithm	-9.606	-10.0	-9.878	0.188

Table 1: Tabela com dados consolidados dos algoritmos

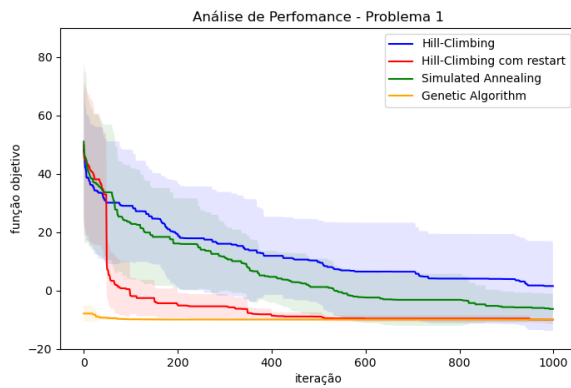


Figure 8: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

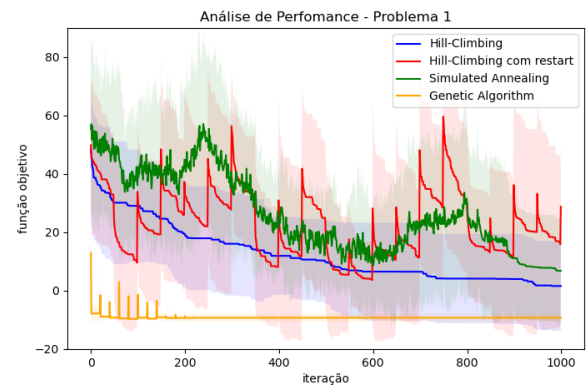


Figure 9: Dados da execução da função objetivo durante as 10 iterações por valor atual.

2 Problema 2

Nesse problema tivemos uma situação um pouco diferente, pois podemos observar uma rápida convergência de todos os algoritmos para o mínimo global, isso pode ser notado nas figuras 6, 2, 1 e 4. Eu acredito que isso tenha ocorrido pela distribuição dos valores iniciais para esse problema, mas também verifica-se que os valores das funções variam muito mesmo com vizinhos próximos, diferentemente do que ocorre com o problema anterior. Logo, para esse problema vamos declarar que os 4 algoritmos se destacaram e tiveram resultados muito próximos do mínimo global.

O algoritmo Hill-Climbing apresentou uma escalada muito abrupta já nas primeiras iterações e um pouco antes das 200 iterações encontrou alguns mínimos locais, já abaixo de 5 e que persistiram até quase o final da sua execução.

O algoritmo Hill-Climbing com restart teve a sua execução muito parecida com o Hill-Climbing, diferença essa que só se vê nos detalhes e na apreciação do gráfico baseado em valores atuais visto na figura 12.

O algoritmo Simulated Annealing teve uma curva ainda mais acentuada, antes mesmo da iteração 100 o algoritmo já estava bem próximo dos mínimos globais, como pode ser visto na figura 4. Mas algo que se destacou no Simulated Annealing foram os picos máximos observados próximos das iterações 400 e 600, onde o primeiro pico atingiu um valor próximo de 350 e o segundo 250, como pode ser visto na figura 5.

O Algoritmo Genético apresenta grande similaridade com o Simulated Annealing no que se refere a rápida escalada, também encontrando valores próximos dos mínimos globais um pouco antes da iteração 100, como pode ser vista na figura 15.

2.1 Algoritmo Hill-Climbing

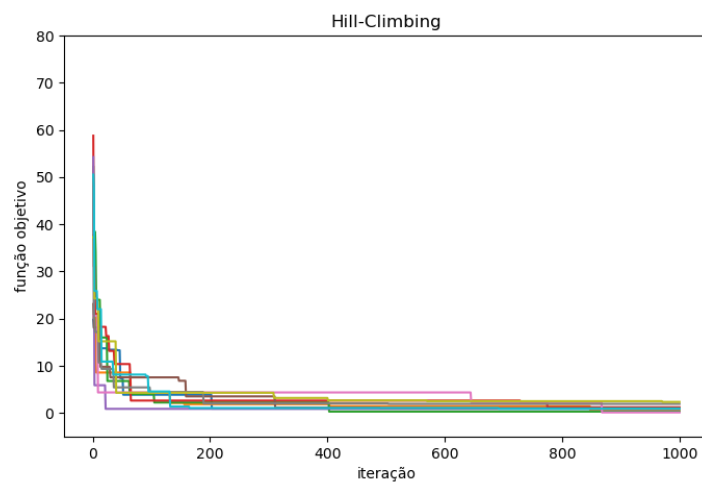


Figure 10: Dados da execução da função objetivo durante as 10 iterações.

2.2 Algoritmo Hill-Climbing com Restart

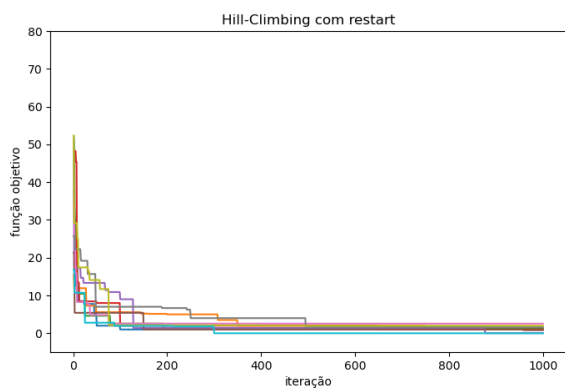


Figure 11: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

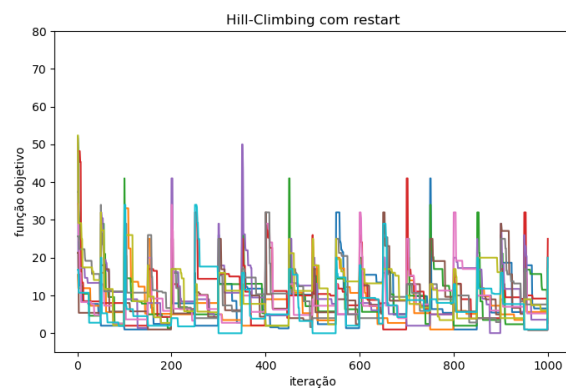


Figure 12: Dados da execução da função objetivo durante as 10 iterações por valor atual.

2.3 Simulated Annealing

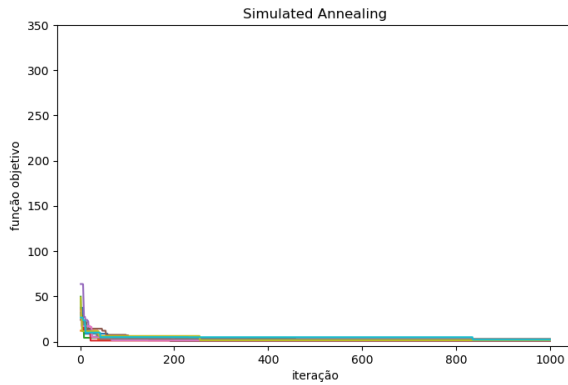


Figure 13: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

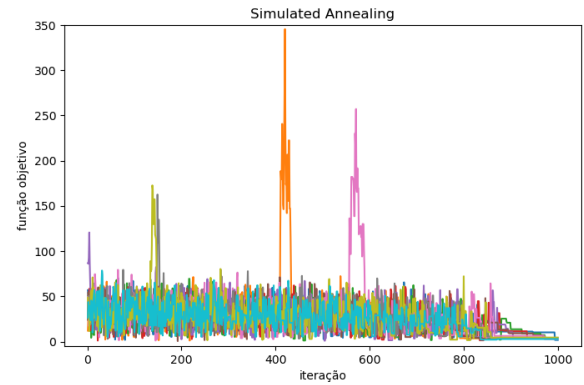


Figure 14: Dados da execução da função objetivo durante as 10 iterações por valor atual.

2.4 Algoritmo Genético

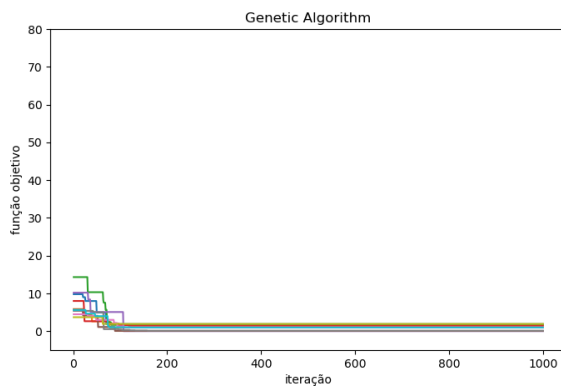


Figure 15: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

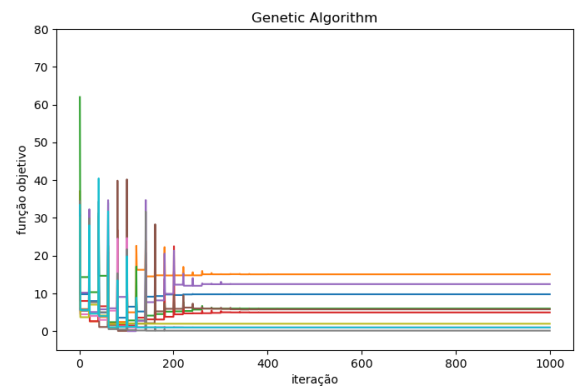


Figure 16: Dados da execução da função objetivo durante as 10 iterações por valor atual.

2.5 Resumo

Algoritmo	Max	Min	Média	Desvio Padrão
Hill-Climbing	2.369	0.144	1.05	0.682
Hill-Climbing com restart	2.556	0.0	1.137	0.795
Simulated Annealing	3.089	0.546	1.48	0.711
Genetic Algorithm	1.949	0.037	1.027	0.737

Table 2: Tabela com dados consolidados dos algoritmos

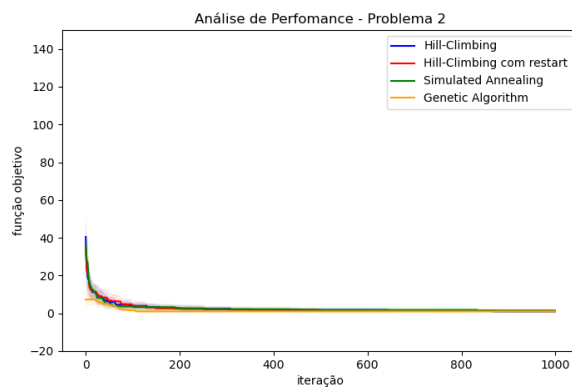


Figure 17: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

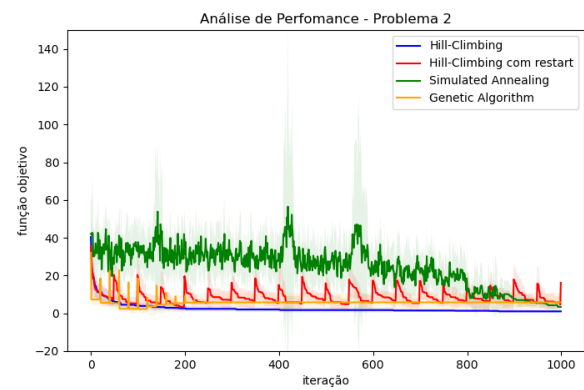


Figure 18: Dados da execução da função objetivo durante as 10 iterações por valor atual.

3 Problema 3

Nesse problema do caixeiro viajante, o Algoritmo Genético se destacou, apresentando os melhores resultados entre todos os demais algoritmos, enquanto todos os demais ficaram presos no mínimo local próximo de 19000, o GA chegou alcançar o mínimo local de 15538, ainda que seja um valor muito longe do mínimo global, que é de 6656, observamos que esse algoritmo ainda não havia chegado ao seu potencial máximo, pois os valores de novas populações ainda estavam oscilando muito, como pode ser visto na figura 25, se compararmos com as figuras dos problemas anteriores 7 e 16, vemos claramente que mesmo após 100k iterações os valores atuais ainda não haviam convergido para um valor único, enquanto nos demais problemas essa convergência ocorreu pouco após a iteração 200.

O algoritmo genético apesar de possuir um peso computacional muito elevado, apresentou uma eficiência muito maior em encontrar novos mínimos locais, acreditamos que após algumas análises sobre esse problema que a aplicação de algumas heurísticas para melhorar a eficácia não só desse mas de todos os demais seja um caminho mais adequado, por exemplo evitando que path's cruzem com outros, verificamos que em geral a rota que segue o perímetro mais externo dos pontos, possui a distância mais curta.

O algoritmo Hill-Climbing apresentou boa performance para esse problema e conseguiu ao longo das iterações fugir de mínimos locais.

Os algoritmos Hill-Climbing com restart e Simulated Annealing apresentaram um comportamento muito parecido com números muito similares também.

3.1 Algoritmo Hill-Climbing

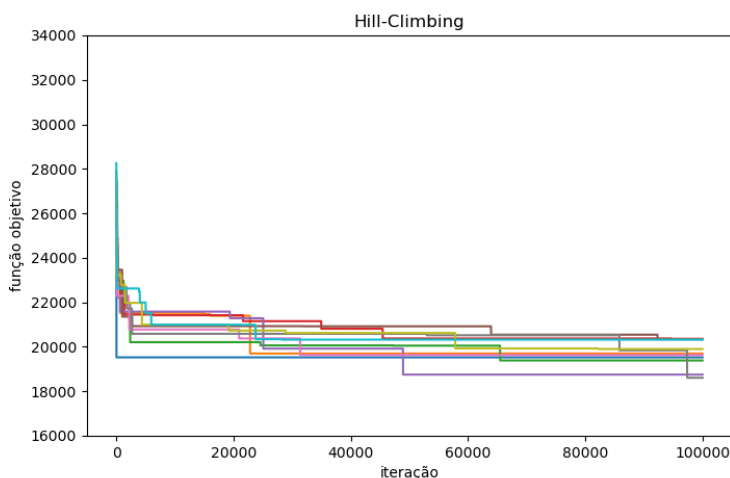


Figure 19: Dados da execução da função objetivo durante as 10 iterações.

3.2 Algoritmo Hill-Climbing com Restart

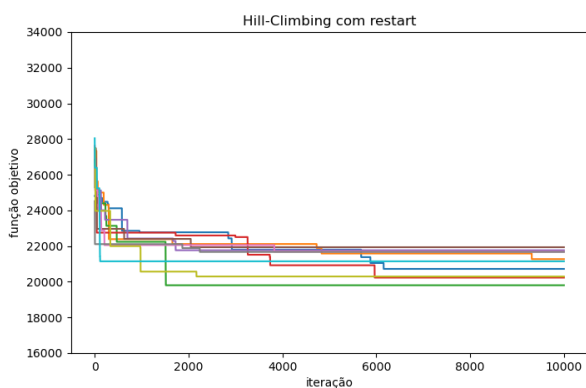


Figure 20: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

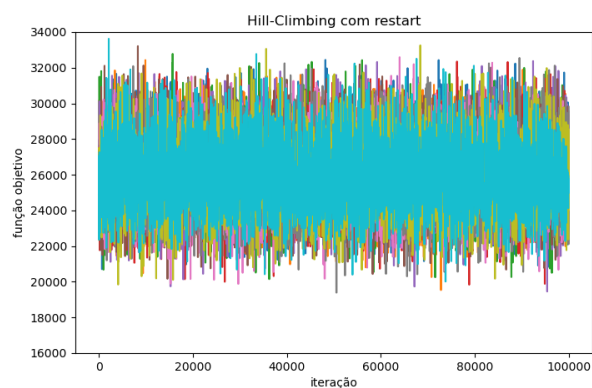


Figure 21: Dados da execução da função objetivo durante as 10 iterações por valor atual.

3.3 Simulated Annealing

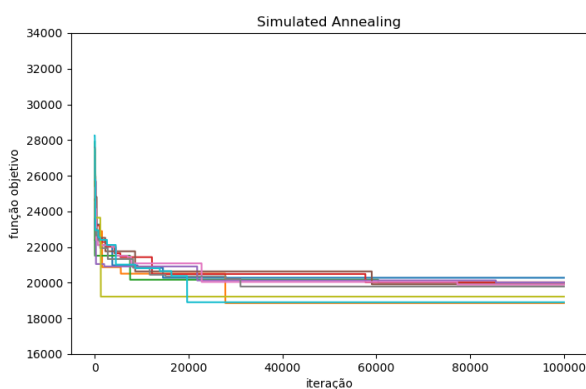


Figure 22: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

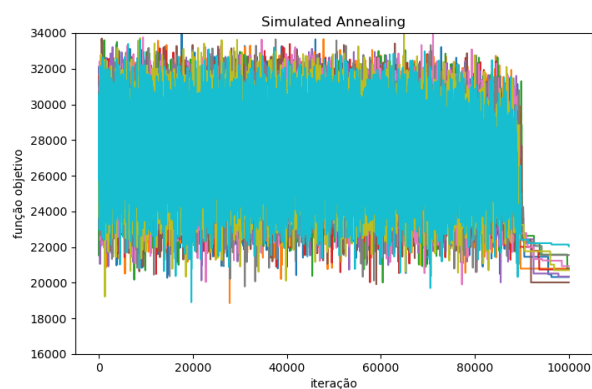


Figure 23: Dados da execução da função objetivo durante as 10 iterações por valor atual.

3.4 Algoritmo Genético

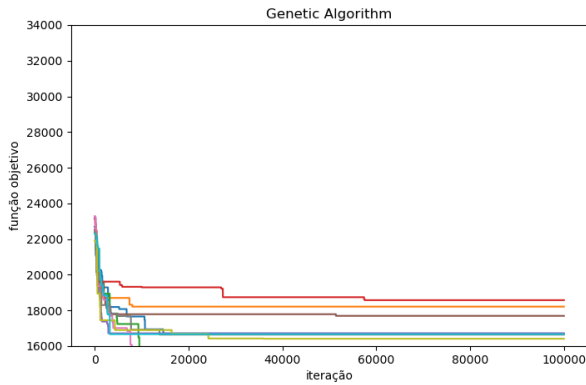


Figure 24: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

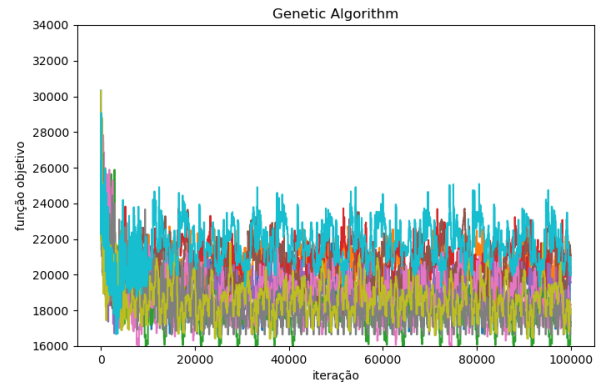


Figure 25: Dados da execução da função objetivo durante as 10 iterações por valor atual.

3.5 Resumo

Algoritmo	Max	Min	Média	Desvio Padrão
Hill-Climbing	20360.232	18608.415	19652.804	621.752
Hill-Climbing com restart	20708.751	19393.573	19898.23	394.147
Simulated Annealing	20280.727	18859.27	19694.674	505.547
Genetic Algorithm	18577.383	15538.484	16879.374	1006.5

Table 3: Tabela com dados consolidados dos algoritmos

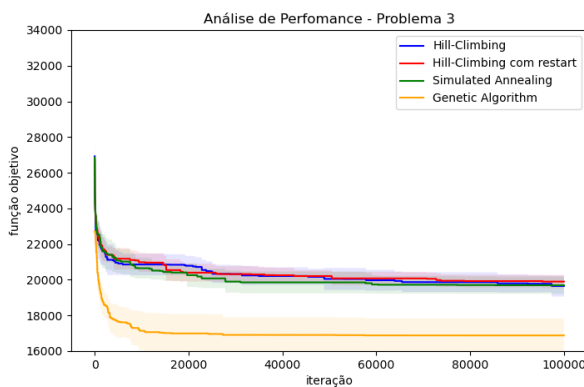


Figure 26: Dados da execução da função objetivo durante as 10 iterações por melhor valor.

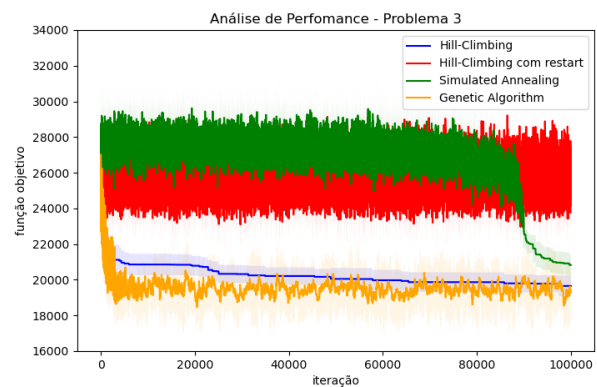


Figure 27: Dados da execução da função objetivo durante as 10 iterações por valor atual.