

Algoritmos Genético e Hill Climbing aplicados ao Sudoku

Bruno Meurer, Gabriel Vieira e Victor Garritano

Junho de 2017

1 Introdução

Neste relatório apresentaremos os resultados dos algoritmos Genético e Hill Climbing, quando aplicamos ao famoso jogo do Sudoku. Vamos expor os parâmetros básicos e indispensáveis a todos os modelos, a análise dos resultados e por fim uma comparação entre os resultados dos dois algoritmos.

2 Genético

2.1 Experimento básico

Representamos os indivíduos através de uma matriz 9 x 9 em que cada linha era inicializada aleatoriamente com valores de 1 a 9, sendo que cada linha contém apenas uma única ocorrência de um determinado valor.

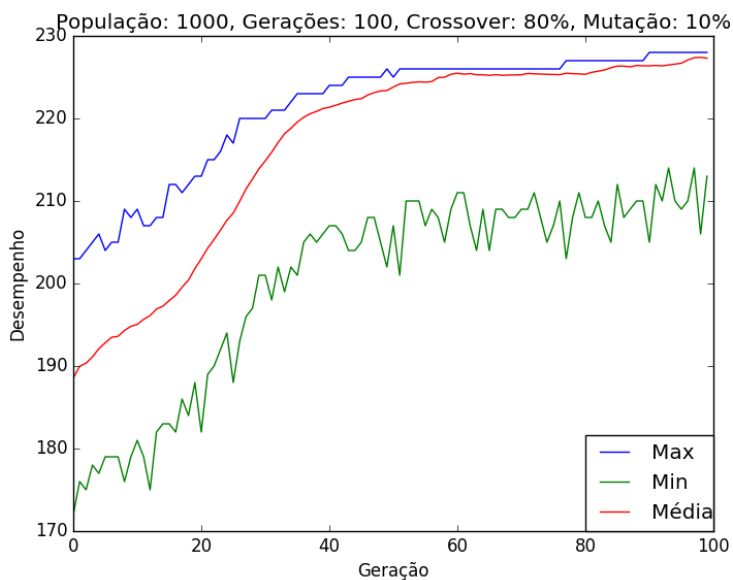
2.2 Função de adaptação

A função utilizada contava o número de elementos únicos por linha, coluna e quadrante. A pontuação máxima é 243 porque, como temos 81 posições no tabuleiro e cada posição (i, j) pode adicionar um total de 3 à pontuação máxima (isso acontece quando nenhuma outra posição pertencente à linha, coluna ou quadrante da posição (i, j) apresenta o mesmo valor de (i, j)), teremos no total uma pontuação máxima de 243.

2.3 Hiper parâmetros do Genético

As probabilidades de *crossover* e mutação consideradas inicialmente foram de 80% e 10%, respectivamente. A população inicial continha 1000 indivíduos e o número de gerações correspondeu a um total de 100. O critério de parada foi o número de gerações e não utilizamos elitismo. Rodamos cada experimento 1 vez.

Figura 1: Resultados do experimento básico

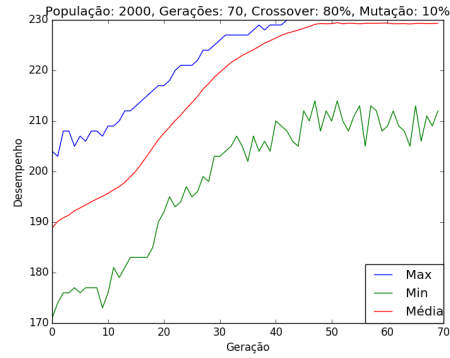


2.4 Avaliação do experimento básico

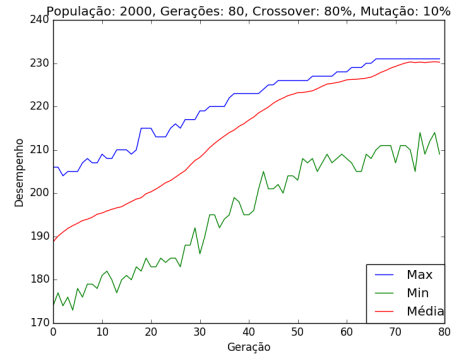
Apresentaremos na Figura 1 o resultado de nosso experimento básico através de um gráfico, onde estão discriminados o mínimo, o máximo e a média de cada geração (nesse experimento, o máximo foi 228, o mínimo foi 213 e a média foi de 227).

2.5 Demais experimentos

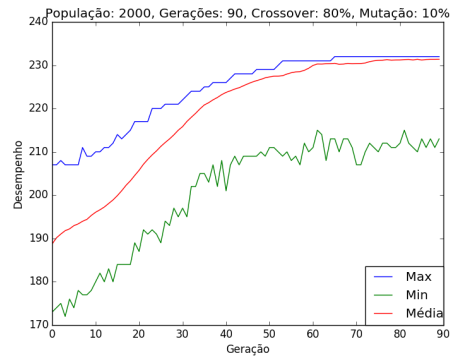
A partir da figura 2, apresentaremos o desempenho dos demais experimentos que realizamos, variando a quantidade de gerações, a probabilidade de mutação e de *crossover*, com uma população inicial de 2000 indivíduos (os parâmetros estão descritos no título de cada gráfico). Repetimos os mesmos experimentos com populações iniciais de 500 e 1000 indivíduos, porém omitiremos os gráficos dos resultados, para que o relatório não fique sobrecarregado de imagens. Comentaremos sobre o efeito da quantidade de indivíduos na população inicial quando analisarmos os resultados.



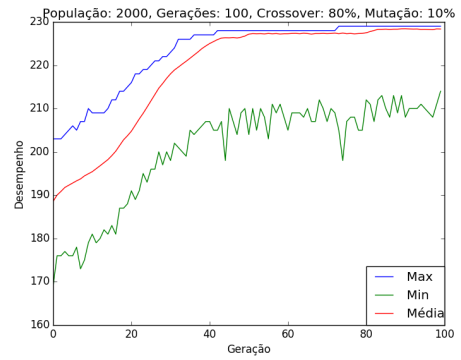
(a) max=230, min=212, avg=229



(b) max=231, min=209, avg=230

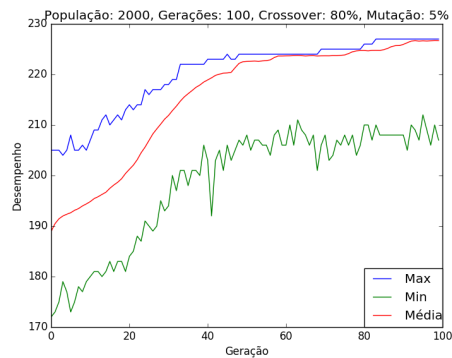


(c) max=232, min=213, avg=231

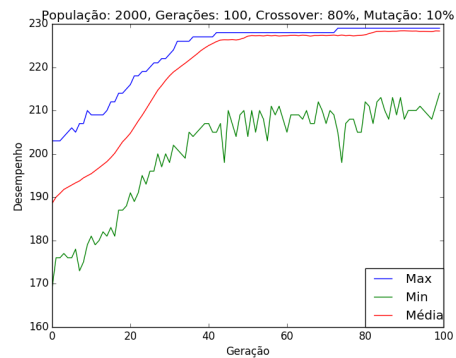


(d) max=229, min=214, avg=228

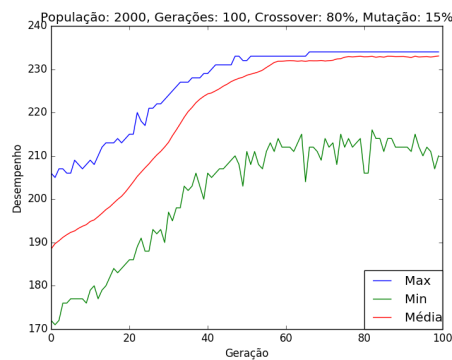
Figura 2: Variando a quantidade de gerações



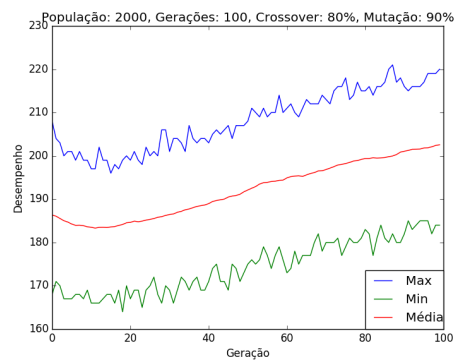
(a) max=227, min=207, avg=227



(b) max=229, min=214, avg=228



(c) max=234, min=210, avg=233



(d) max=220, min=184, avg=202

Figura 3: Variando a probabilidade de mutação

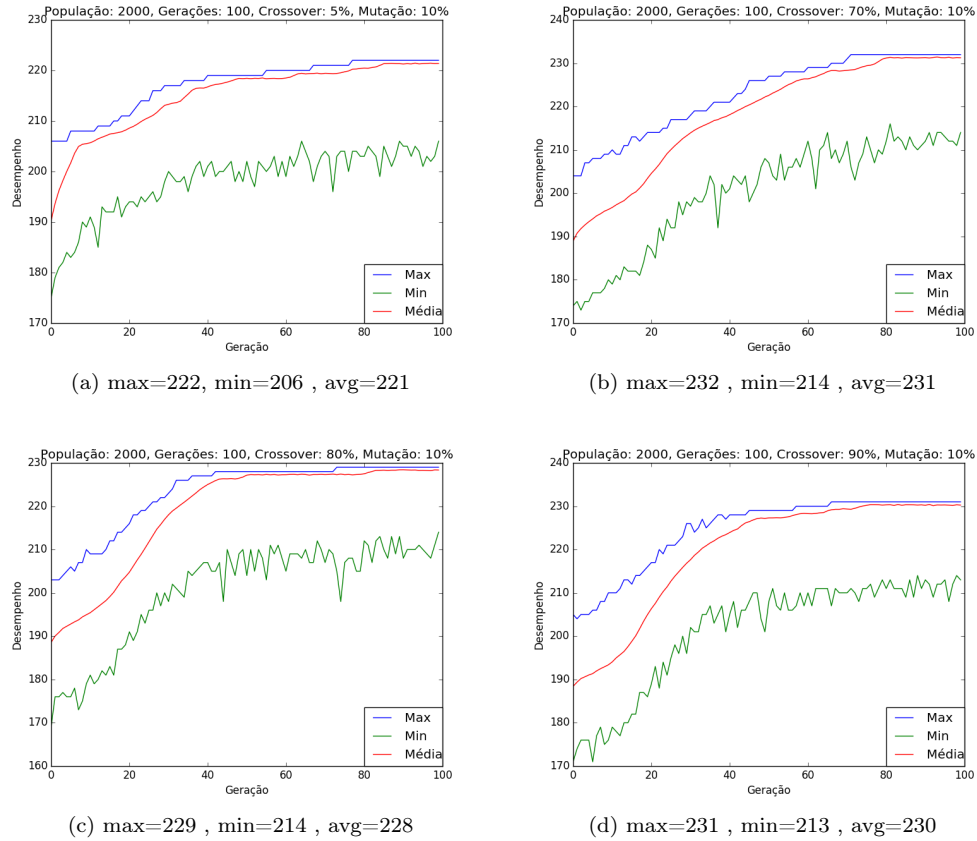


Figura 4: Variando a probabilidade de crossover

2.6 Análise dos resultados

Nenhum dos experimentos conseguiu alcançar a pontuação máxima de 243, o que nos leva a concluir que o algoritmo genético não foi capaz de resolver totalmente o Sudoku.

Com relação ao experimento inicial, podemos constatar que o quantidade de gerações que foi utilizada por padrão em nossos experimentos (100) não era o ideal, pois todos os experimentos utilizando um número de gerações menor apresentaram um máximo e uma média de indivíduos superiores aos encontrados no experimentos básico. Em particular, o experimento com 90 gerações apresentou o melhor resultado.

Alterando a probabilidade de mutação, percebemos que mais uma vez a configuração inicial não foi a que apresentou os melhores resultados, porém diferentemente do caso anterior em que alteramos o número de gerações, o valor inicial desse parâmetro foi o segundo a apresentar os melhores resultados. O

primeiro foi o experimento com probabilidade de mutação igual a 15 %.

Observamos também que uma probabilidade de mutação muito alta piorou muito os resultados, o que indica que, em uma tarefa como o Sudoku, utilizar uma taxa de mutação muito “agressiva” não é um bom caminho a ser considerado, tendo como objetivo encontrar uma solução para o jogo.

Por fim podemos verificar que, empregando probabilidades altas de *crossover* (em torno de 80% ~ 90%), os experimentos obtiveram os melhores resultados. Há uma diferença significativa entre o experimentos com apenas 5% de probabilidade de *crossover* e os outros, que possuem esse valor muito mais elevado.

Baseado nos resultados que foram omitidos nesse relatório, um grande população inicial (2000) apresentou resultados melhores do que populações menores (500 e 1000), o que pode indicar que um espaço inicial de busca maior facilita o algoritmo no que diz respeito a encontrar o melhor caminho, que é aquele que leva à maior pontuação.

3 Hill Climbing

3.1 Experimento básico

A forma de representação dos indivíduos é a mesma do algoritmo genético, uma matriz 9 x 9.

3.2 Função de objetivo

A função de objetivo é igual à função de adaptação do algoritmo Genético. Porém no *Hill Climbing* mudamos de estado (configuração do tabuleiro do Sudoku) trocando 2 posições de uma mesma linha. A cada busca pelo próximo estado, obtemos a pontuação de todos estados possíveis do tabuleiro. Embora isso seja custoso computacionalmente, acreditamos que dessa maneira conseguimos evitar que o algoritmo fique estagnado em um máximo local.

3.3 Resultados

Vamos apresentar agora os resultados do algoritmo. Realizamos algumas execuções e, em cada execução, o algoritmo pode recomeçar a busca até 10 vezes, por isso cada gráfico possui 10 curvas plotadas. Exibiremos agora algumas delas.

Figura 5: 1ª execução - Max = 227

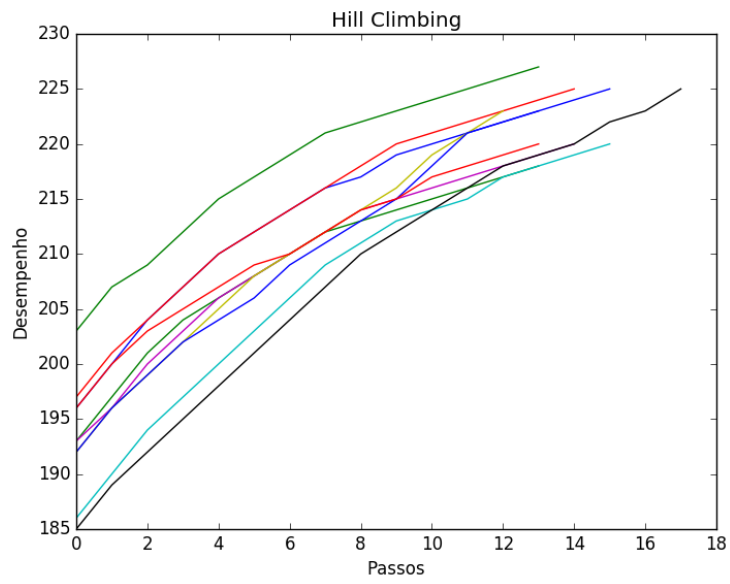


Figura 6: 2ª execução - Max = 224

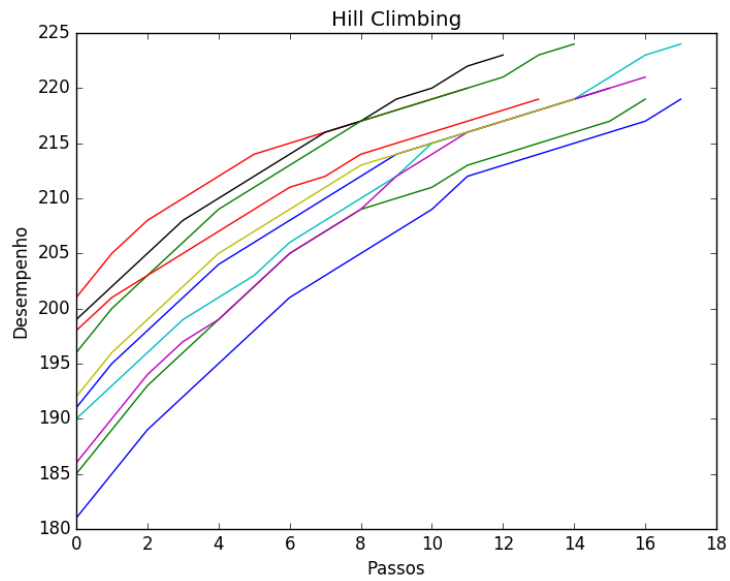


Figura 7: 3ª execução - Max = 225

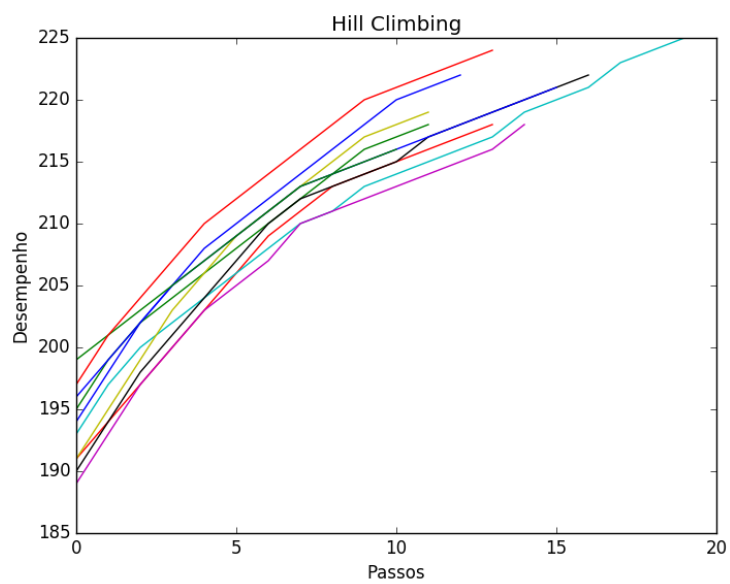


Figura 8: 4ª execução - Max = 229

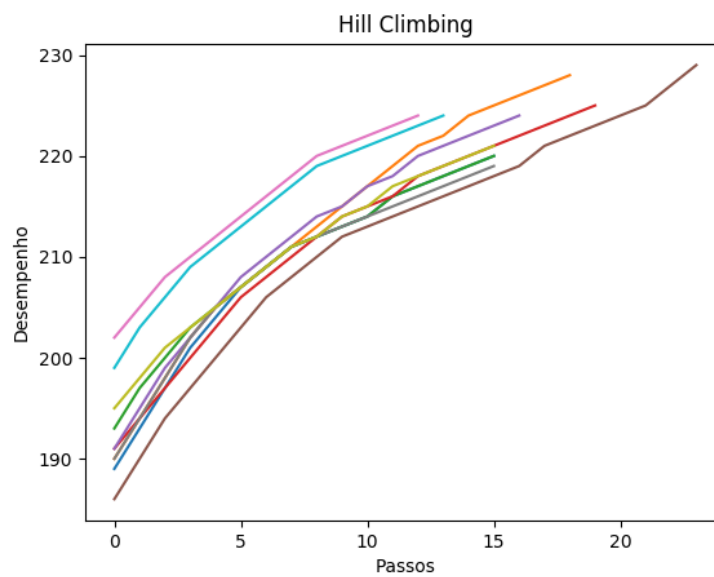


Figura 9: 5ª execução - Max = 227

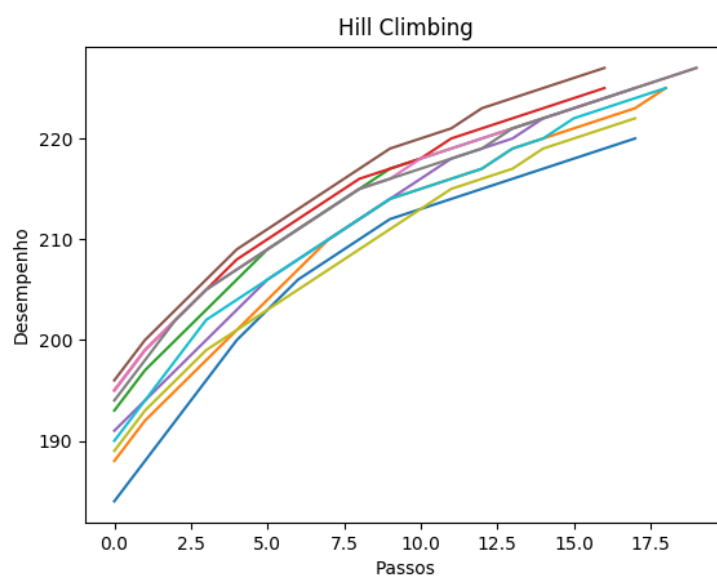
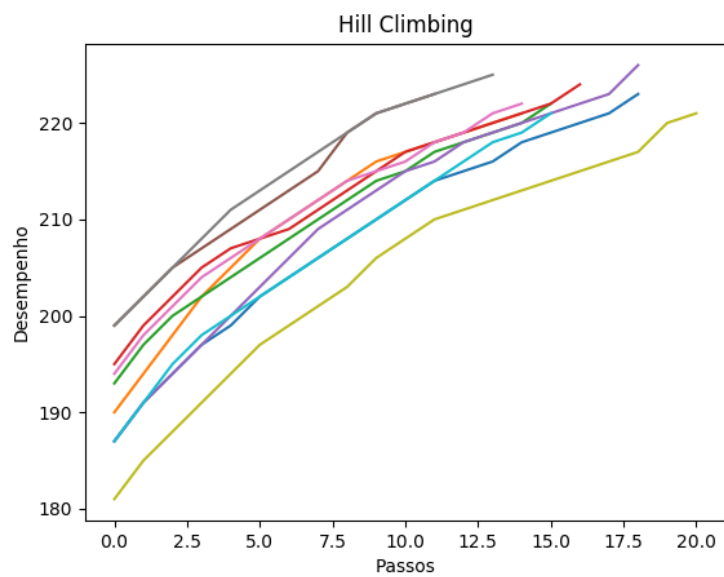


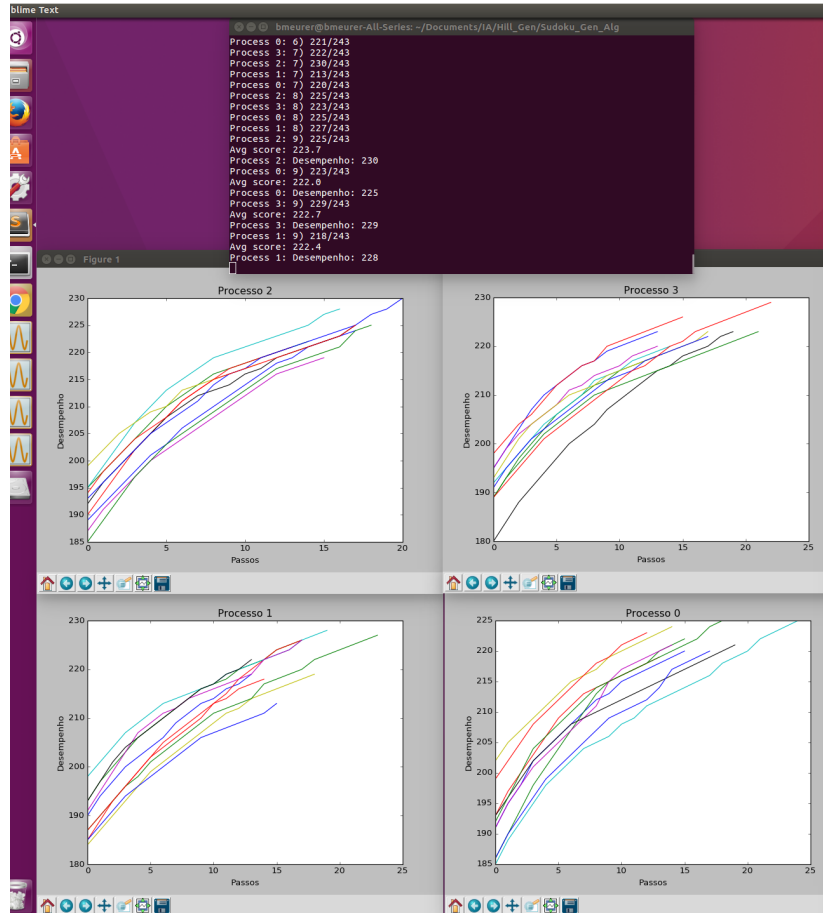
Figura 10: 6ª execução - Max = 226



3.4 Versão Paralela

Implementamos uma versão que roda várias instâncias do algoritmo *Hill Climbing* paralelamente, para verificar se conseguiríamos encontrar a solução ótima mais rapidamente. Bom, depois de rodarmos algumas vezes, chegamos a conclusão de que chegar na solução ótima não é apenas uma questão de tempo.

Figura 11: Várias instâncias do *Hill Climbing* rodando paralelamente



3.5 Análise dos resultados e comparação entre os dois algoritmos

Os resultados do *Hill Climbing* foram um pouco inferiores aos resultados do algoritmo genético. Como podemos perceber, cada execução do algoritmo percorre em média 18 estados antes de chegar a um máximo (local, já que nenhum deles conseguiu a pontuação máxima, que seria o máximo global).

Nenhum dos dois algoritmos conseguiu encontrar a solução ótima para o problema do Sudoku. O Algoritmo Genético apresentou um tempo de execução maior que o *Hill Climbing*, e essa diferença no tempo de execução crescia à medida que a população inicial e o número de gerações aumentava.