**Solução para buscas**

Alexandre Silva 624071

Gabriela Stela

Hugo Hakamada

Wagner Silva

Uma vez que o Hill Climbing é propenso à parar no ótimo local, poderíamos fazer o algoritmo executar até encontrar uma solução ou a solução mais próxima do global ótima. Como sabemos que o global ótima se ao alcançar o marco de 0 ataques, podemos fazer o algoritmo executar *n* vezes e armazenar os resultados. Ao final, podemos avaliar os resultados e pegar a configuração que teve a melhor performance perante as outras.

De Maneira geral, poderíamos modelar o problema como sendo a funçãoo algoritmo hill climbing, o estado atual, o tamanho do passo e o número mínimo de ataques que o agente alcançou. Em uma representação gráfica, o estado atual é posto sob as abscissas e o número final de ataques é o valor em .

[grafico]

Assim, nossa modificação executaria o hill climbing original vezes com estados iniciais e passos randomicos no intervalo de

, e quando encontrada essa solução e salva e uma nova posição aleatória é dada para ele, sendo executado $n$ vezes. Assim, ao final das $n$ execuções teremos uma lista de $n$ combinações para o ótimo global, e então podemos compara e encontrar o mais proximo do ótimo global.

Esse não seria o método mais eficiente, sendo $O(nm)$ do qual $n$ é a quantidade de iterações e $m$ o valor máximo de combinações de ataques.