Um breve resumo do Algoritmo Optimizer

O algoritmo funciona por meio de recursividade, percorrendo uma árvore de tentativas, testando todas as possibilidades de arrumação das caixas no container. E escolhe aquela que apresenta melhor resultado, considerado o maior volume alocado dentro do container.

A aplicação inicia com a função *optimizer*, que apenas declara constantes e variáveis e realiza a primeira chamada à função *solutionSeeker*, que é quem vai fazer a maior parte do trabalho.

A função solutionSeeker se utiliza de um mapa do chão do container (portanto em 2D, o que reduzir a necessidade de processamento). Nesse mapa, as projeções no plano XY de cada caixa que vai sendo colocada no container na posição z = 0 (no chão) vão sendo registradas (no mapa).

A solutionSeeker vai percorrendo os lugares vagos desse mapa. Após percorrer todo o chão do container, o mapa é atualizado para a altura da caixa de menor altura, liberando os espaços dessas caixas mais baixas do mapa. E assim sucessivamente até chegar no teto do container ou acabarem as caixas ou não couber mais nenhuma.

Seguindo pelo mapa, a cada lugar livre encontrado, a *solutionSeeker* chama a função *tryWithThisBox* para cada um dos três tipos de caixa. Nesse momento, é criado um nó de bifurcação e a execução segue para cada uma das três possibilidades, uma de cada vez. Em seguida, cada uma das três possibilidades anteriores se bifurca em outras três possibilidades, criando um novo nó, assim sucessivamente. Até que as caixas acabem ou até que não caiba mais nenhuma caixa no container.

Nesse momento, cada uma das estâncias vai retornando seus valores obtidos aos nós que as chamaram. A *solutionKeeper* comparando os valores obtidos por cada caminho percorrido e escolhe somente a que possui maior valor de volume alocado. Assim sucessivamente, até retornar o valor da melhor solução para a primeira função que havia a chamado em primeiro lugar: a função *optimizer*. Por tentar todas as possibilidades, o programa consegue encontrar a melhor arrumação possível.