

Modelling III: truck loading



Loading boxes onto trucks

A business sells many types of items.

Each type:

- has a fixed price per item,
- generates a known profit per item,
- has a known weight,
- has a current stock level.

The truck has a max weight it can carry.



Choose items to load onto the truck, without exceeding the weight capacity of the truck.

Either

- (i) ensure the profit is above some threshold, or
- (ii) maximise the profit.

Model this problem as a CSP, and then implement it in Choco

The Knapsack problem

- fundamental problem in combinatorial optimisation
- known to be NP-hard
- many important applications in resource management, sequence and scheduling, task assignment, financial portfolio management, and computer science
- the knapsack problem forms the basis of knapsack cryptosystems

The knapsack global constraint

18.23 knapsack

The *knapsack* constraint involves: - an array of integer variables *OCCURRENCES*, - an integer variable *TOTAL_WEIGHT*, - an integer variable *TOTAL_ENERGY*, - an array of integers *WEIGHT* and - an array of integers *ENERGY*.

It formulates the Knapsack Problem: to determine the count of each item to include in a collection so that the total weight is less than or equal to a given limit and the total value is as large as possible.

- $\sum OCCURRENCES[i] \times WEIGHT[i] \leq TOTAL_WEIGHT$ and
- $\sum OCCURRENCES[i] \times ENERGY[i] = TOTAL_ENERGY$.

86

Chapter 18. Constraints over integer variables

API:

```
Constraint knapsack(IntVar[] OCCURRENCES, IntVar TOTAL_WEIGHT, IntVar TOTAL_ENERGY,  
                   int[] WEIGHT, int[] ENERGY)
```

Example

```
1      IT[2] = VF.bounded("IT_2", 0, 1, solver);  
2      IntVar WE = VF.bounded("WE", 0, 8, solver);  
3      IntVar EN = VF.bounded("EN", 0, 6, solver);  
4      int[] weights = new int[]{1, 3, 4};  
5      int[] energies = new int[]{1, 4, 6};  
6      solver.post(ICF.knapsack(IT, WE, EN, weights, energies));  
7      Chatterbox.showSolutions(solver);  
8      solver.findAllSolutions();  
-
```

The knapsack global constraint

- the knapsack global constraint is significantly faster on larger hard problems
- can be combined with other arbitrary constraints to model realistic problems
- general principle for effective solving in Choco:
 - check to see if your problem contains any standard sub-patterns that may have been modelled as global constraints
 - if so, use them
 - but be prepared to experiment to find the best settings

Next lecture ...

More modelling: bin packing