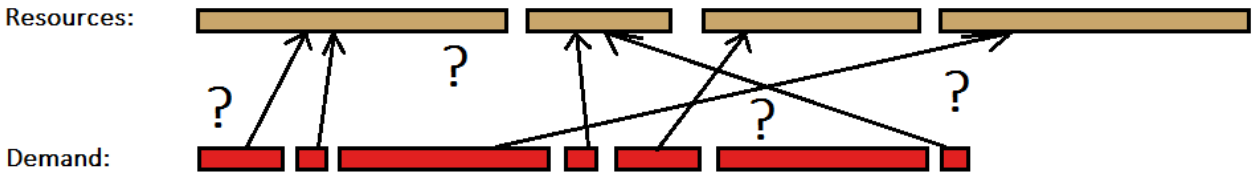
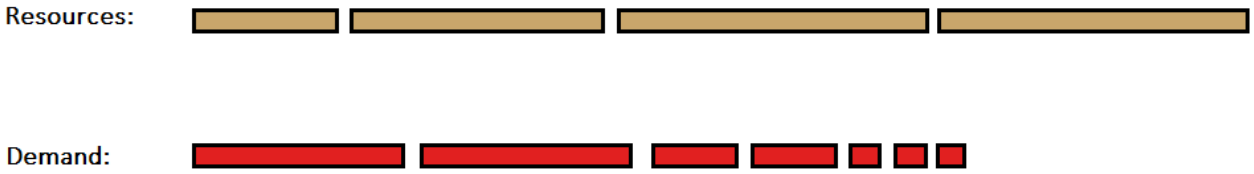
Wood: how it works

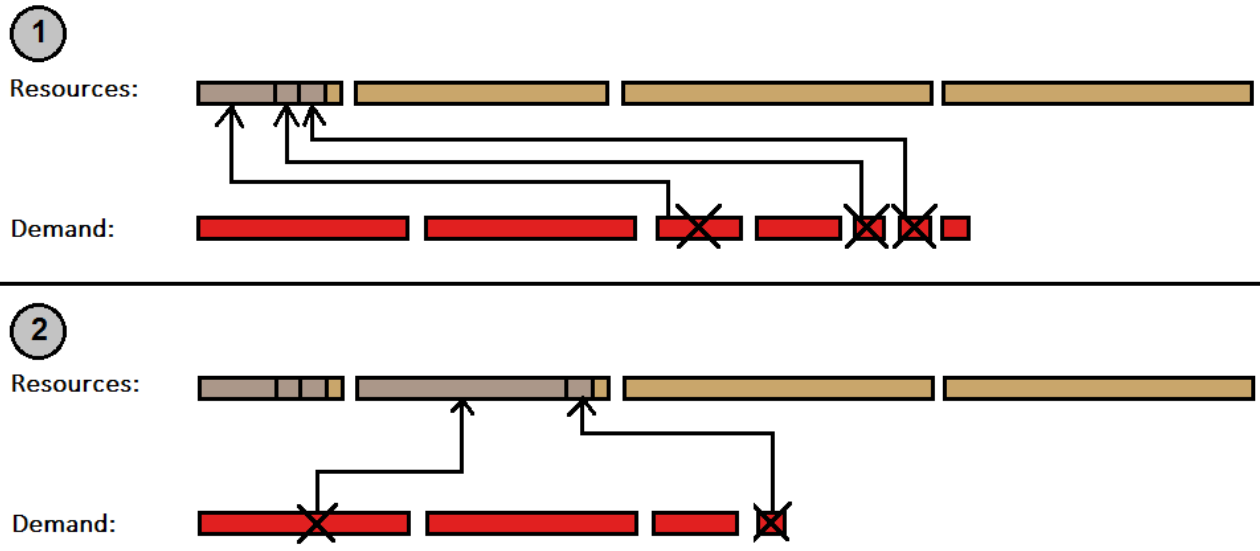
Problem: How should we allocate demanded pieces to wood resources in an efficient way?



Solution: Instead of using pricy recursion based functions, we can harness greedy principles that will guarantee a possible solution (if exists). Due to their nature, a relatively good efficiency is gained.
The algorithm will sort the resources in an ascending order and the demanded pieces in a descending order:

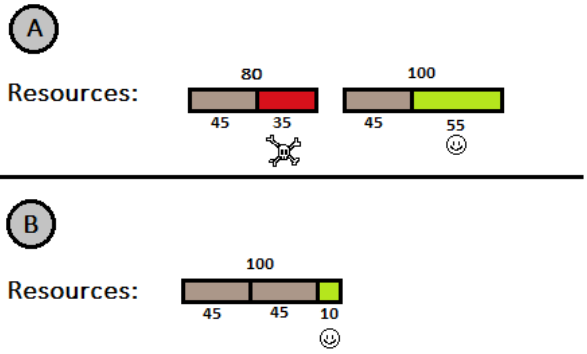


Now, in each iteration the algorithm will choose the shortest (weakest) available resource and will try to fill it with the longest (most problematic) demanded pieces that it can contain. The longest demanded pieces are the "bottleneck" of the problem and by prioritizing their handling, a possible solution must appear if exists (while we try to preserve the more powerful longer resources):



Wood: optimizations

Singular optimizations: An optimization that may prove useful for small projects.
sometimes the algorithm will output us a solution that will contain an ineffective division of at list one resource while the last (longest) resource may be mostly unused. By eliminating that inefficient resource and recalculating, we may get a better solution with a cheap computing time:



Greedy efficiency optimization: an optimization that allows only waste efficient-wise solutions.
By pre-defining criteria, the algorithm is able to real-time reject any resource that didn't get an efficient pieces allocation, releasing its temporary allocated pieces for the next resource iteration.
If the resource is "factory sized" and doesn't match the efficiency criteria – a demanded piece will be reallocated from it leaving us with a usable larger resource for future projects (no rejection since we don't expect any upcoming iterations to gain better efficiency for the same settings).
This optimization is better for "the long run" and may require a larger resource pool. In the diagram A was calculated without using the optimization while B did. The efficiency criteria were roughly set to reject remaining resources with sizes of 20-40: