

Optimization Methods

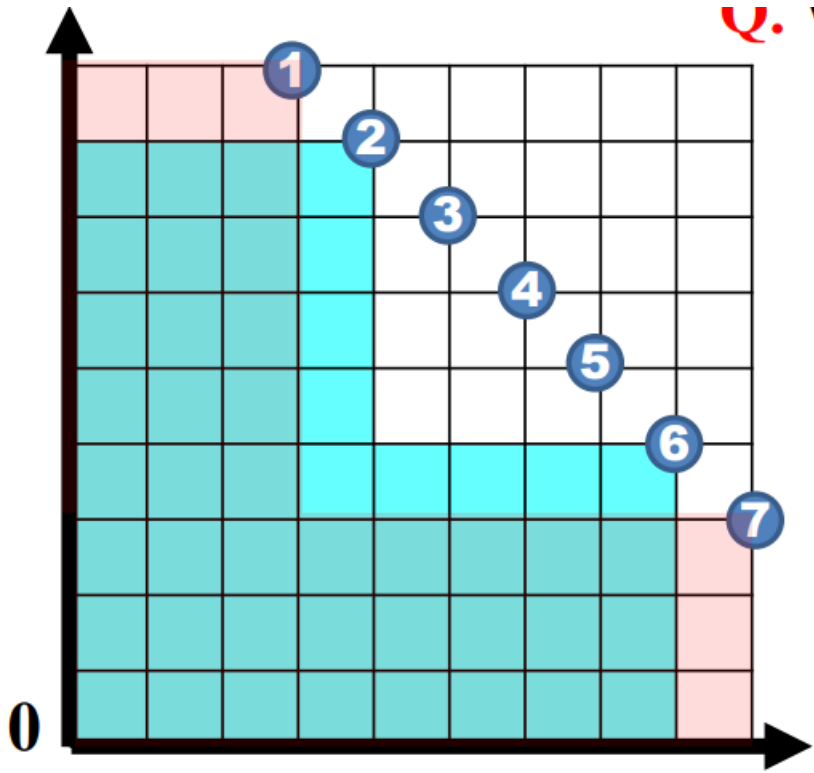
Lab 8 Session



By Shi Jimao



Task1



When the candidate size is too large, update the contribution of all items is time consuming. Hence if we don't update the contribution frequently, it will be more effectively.

At first, we calculate contribution of all items and select the largest one.

And then we repeat the following steps:

1. select the best item in the remain set
2. update this item's contribution(since the recent contribution is in previous selected set)
3. **if** (the selected item is still the best item after updating)
 - add this item to the selected set**else**
 - update the selected item in the remain set and go to step1

Task2

First, I calculate all items' contribution initially. Instead of randomly selecting a item in each iteration, I randomly choose a set of size k from the remaining set, and choose the best item from the subset.

Hence the contribution is not for current solution, so we should update contribution of all remaining items with a constant frequency f . Usually, I set f equal to $(\text{total training rounds} / k)$.

This idea is similar with task1 that lazy updating and using historical information to get the best solution.

