

Parameter:

$k = \text{\#Nachbarn}$ [hier = 2]

$m = \text{\#Anzahl der zu suchenden Outlier}$ [hier = 3]

Vorgehensweise:

- a) berechne kNN-Graphen (gerichtete Kante von p nach q existiert genau dann, wenn q einer der kNN von p ist) [hier bzgl. eukl. Dist.]
- b) berechne für jede Kante im kNN-Graphen das Gewicht [hier absolute Score-Differenz]
- c) entferne Kanten aus dem Graphen, mit höchstem Gewicht beginnend absteigend
- d) Wenn ein Punkt durch das sukzessive Entfernen von Kanten isoliert wird (keine ein- oder ausgehenden Kanten), ist er ein Outlier
- e) Verfahre so, bis m Outlier gefunden worden sind

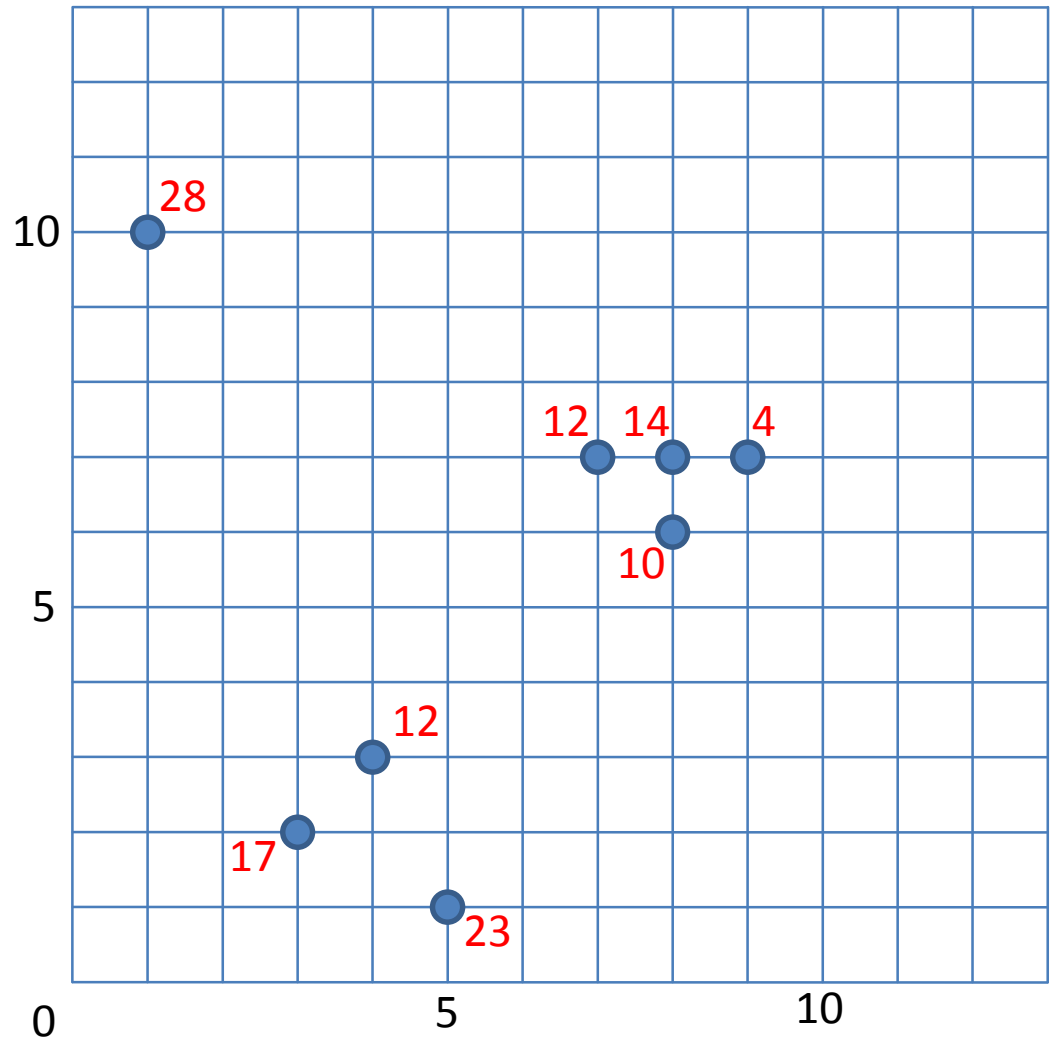
Mögliche Varianten bei Gleichheit der kNN-Distanz:

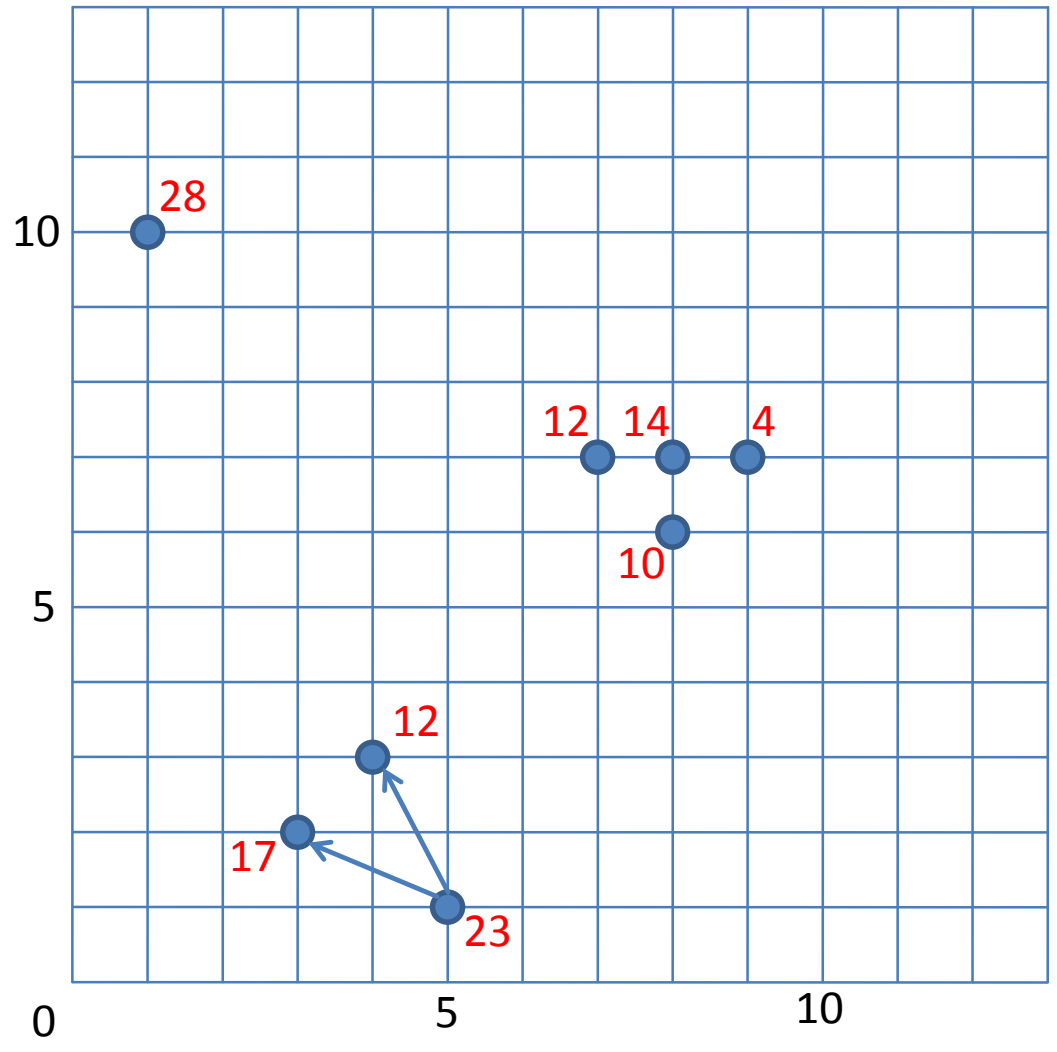
- 1) füge alle Punkte mit gleicher kNN-Distanz der kNN-Menge hinzu (\Rightarrow u.U. $\#kNN \geq k$)
- 2) fähle nichtdeterministisch einen der Punkte mit gleicher kNN-Distanz aus ($\Rightarrow \#kNN = k$)

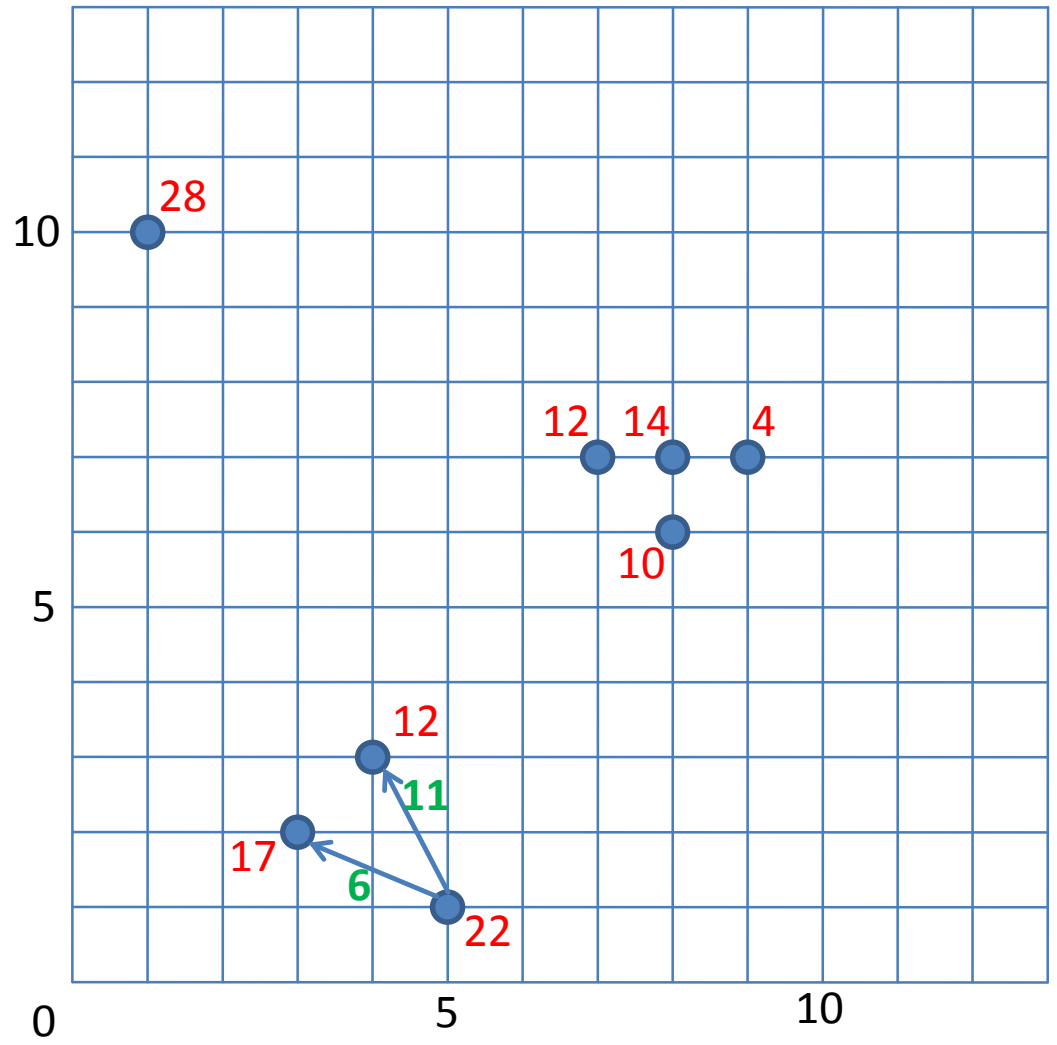
Mögliche Varianten bei Gleichheit der Gewichtungen der Kanten im kNN-Graphen:

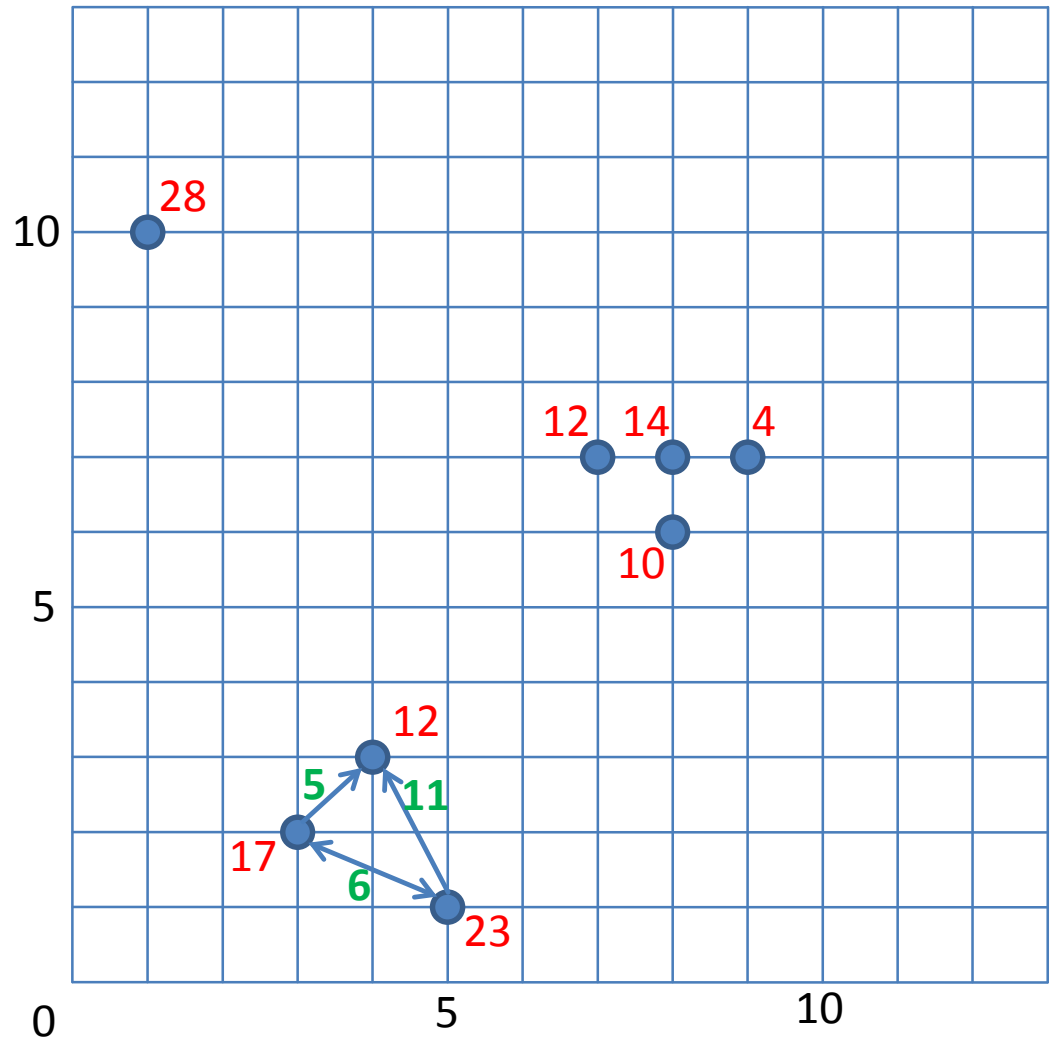
- 1) entferne alle Kanten mit gleichem Gewicht auf einmal
- 2) entferne eine der Kanten nichtdeterministisch

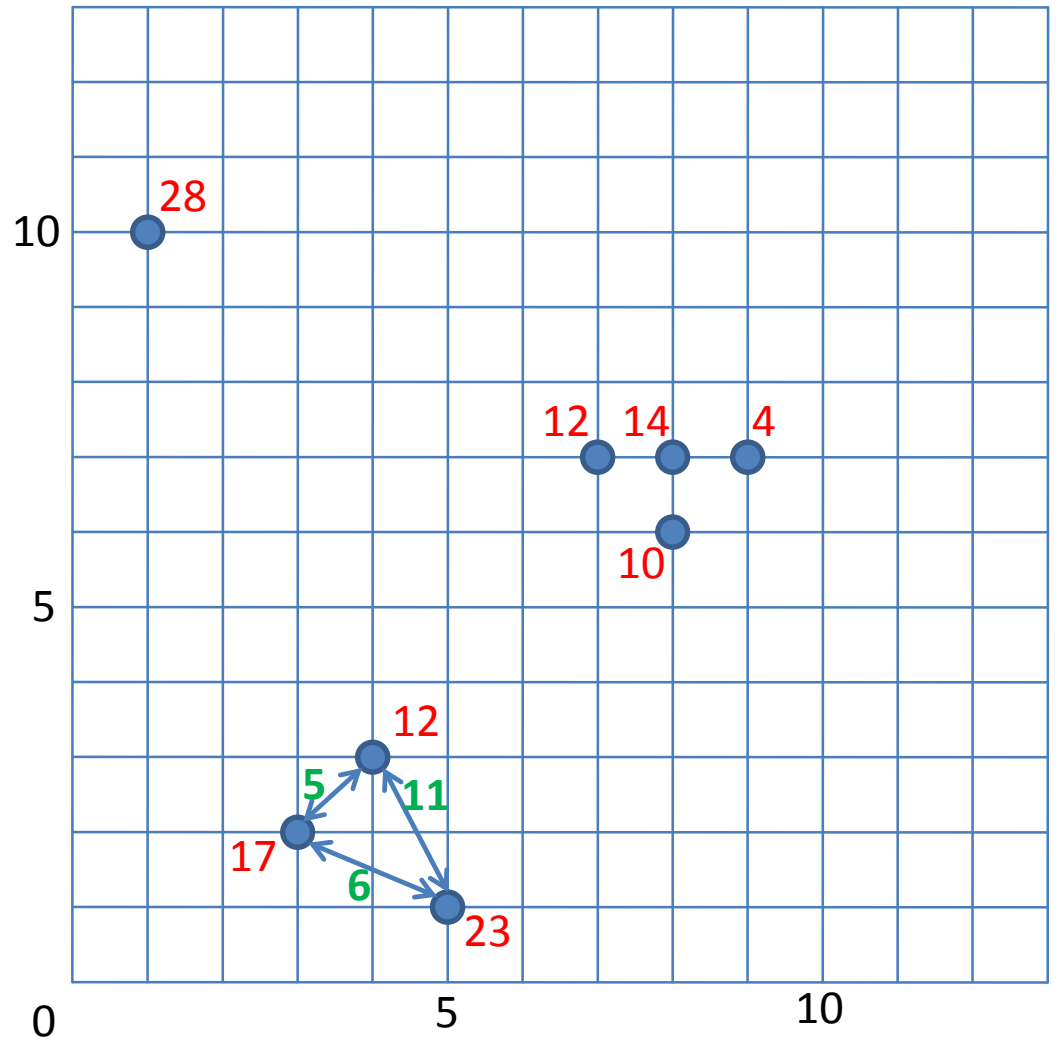
Wir wählen Strategie 1) und 1)

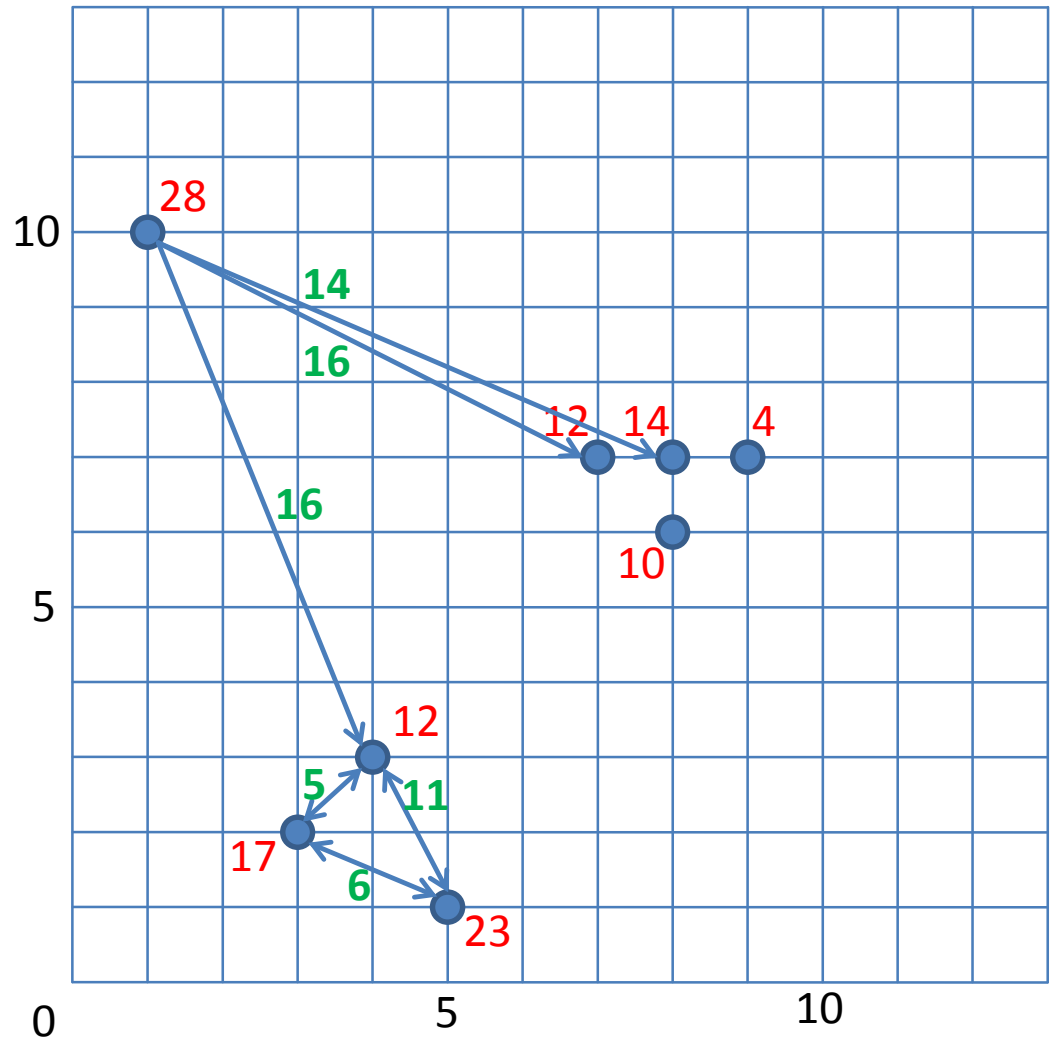


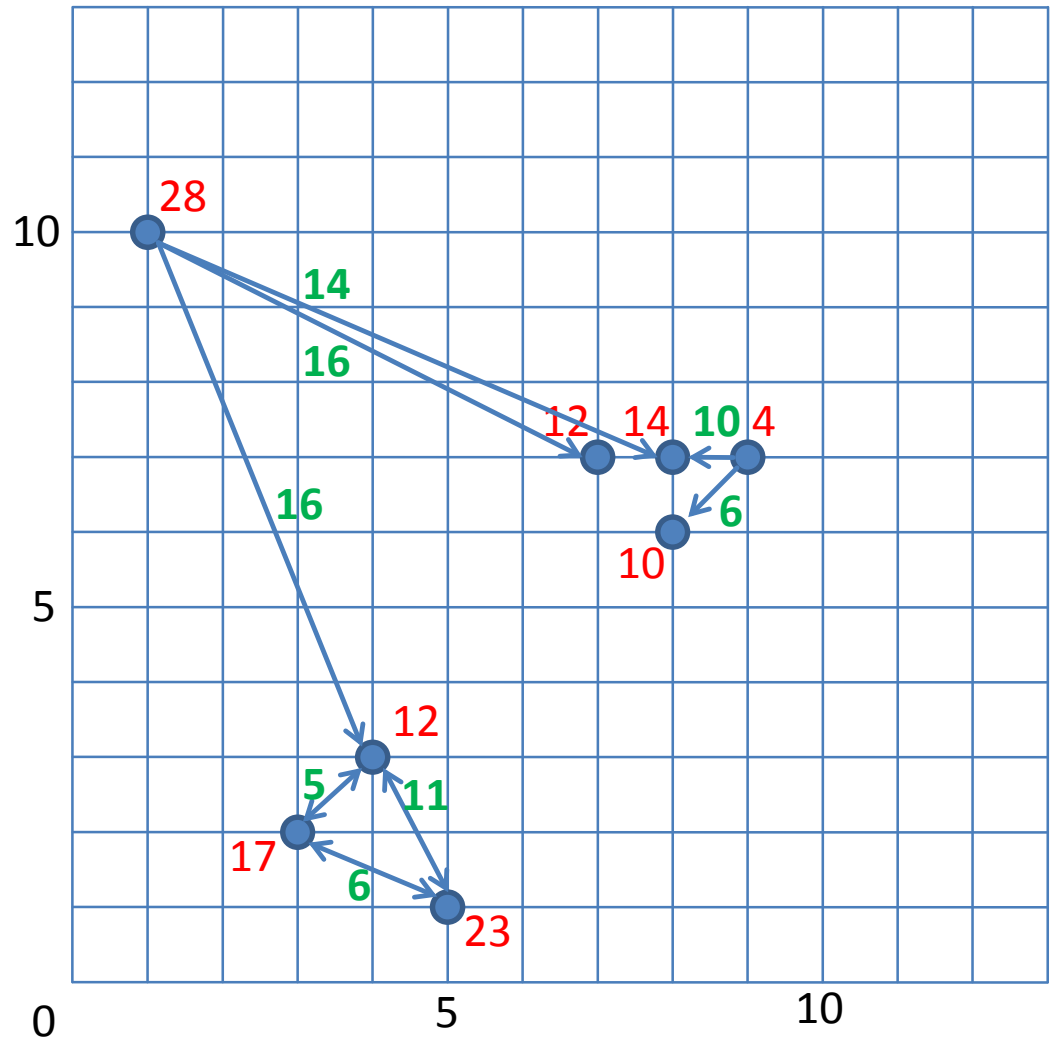


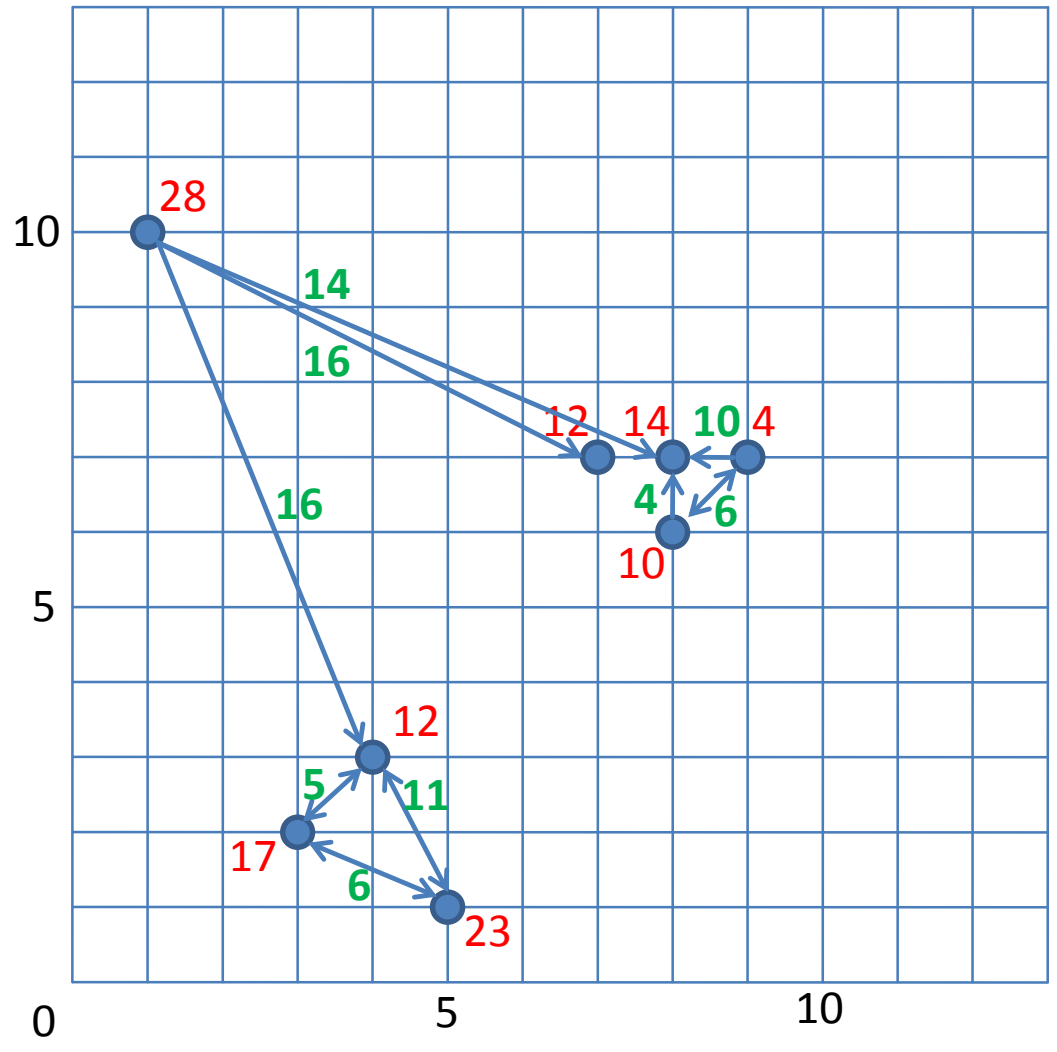


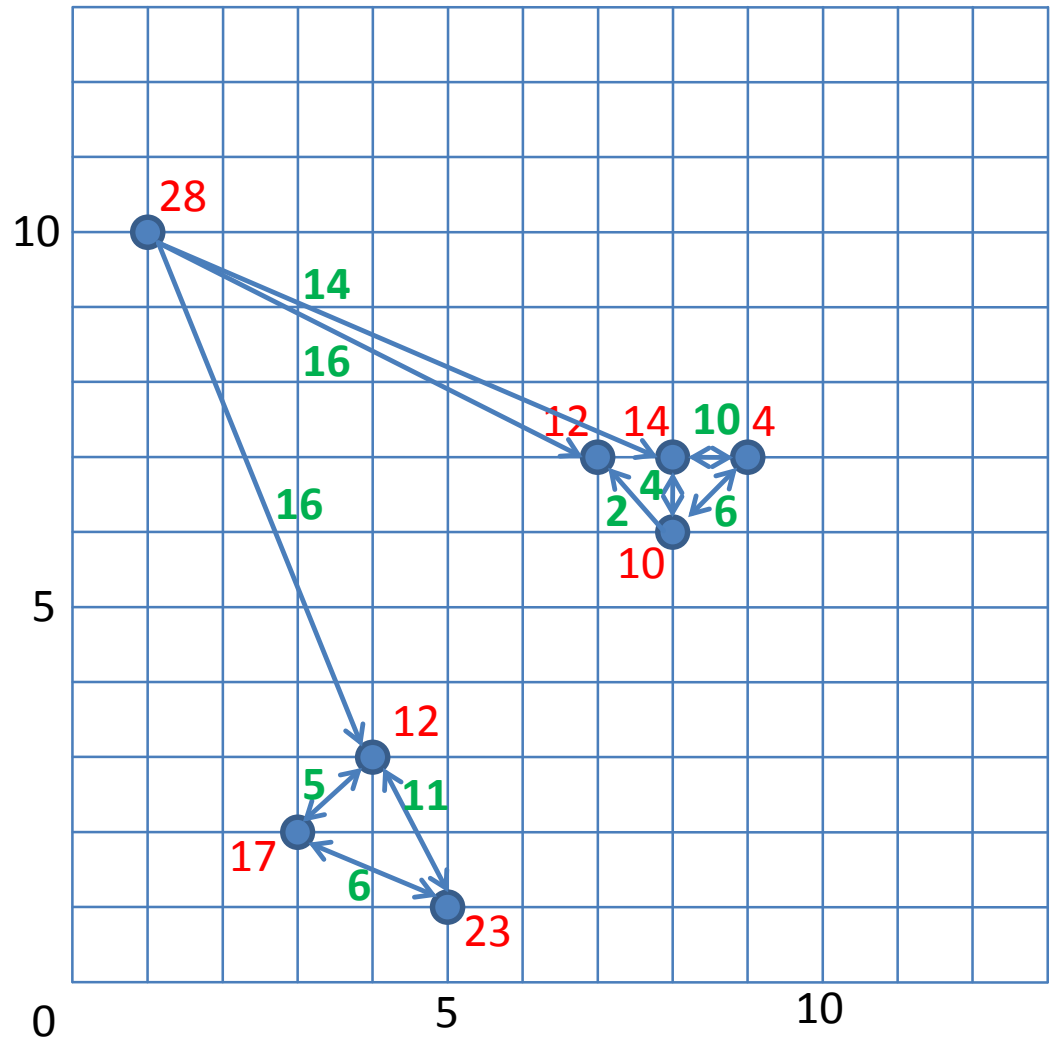




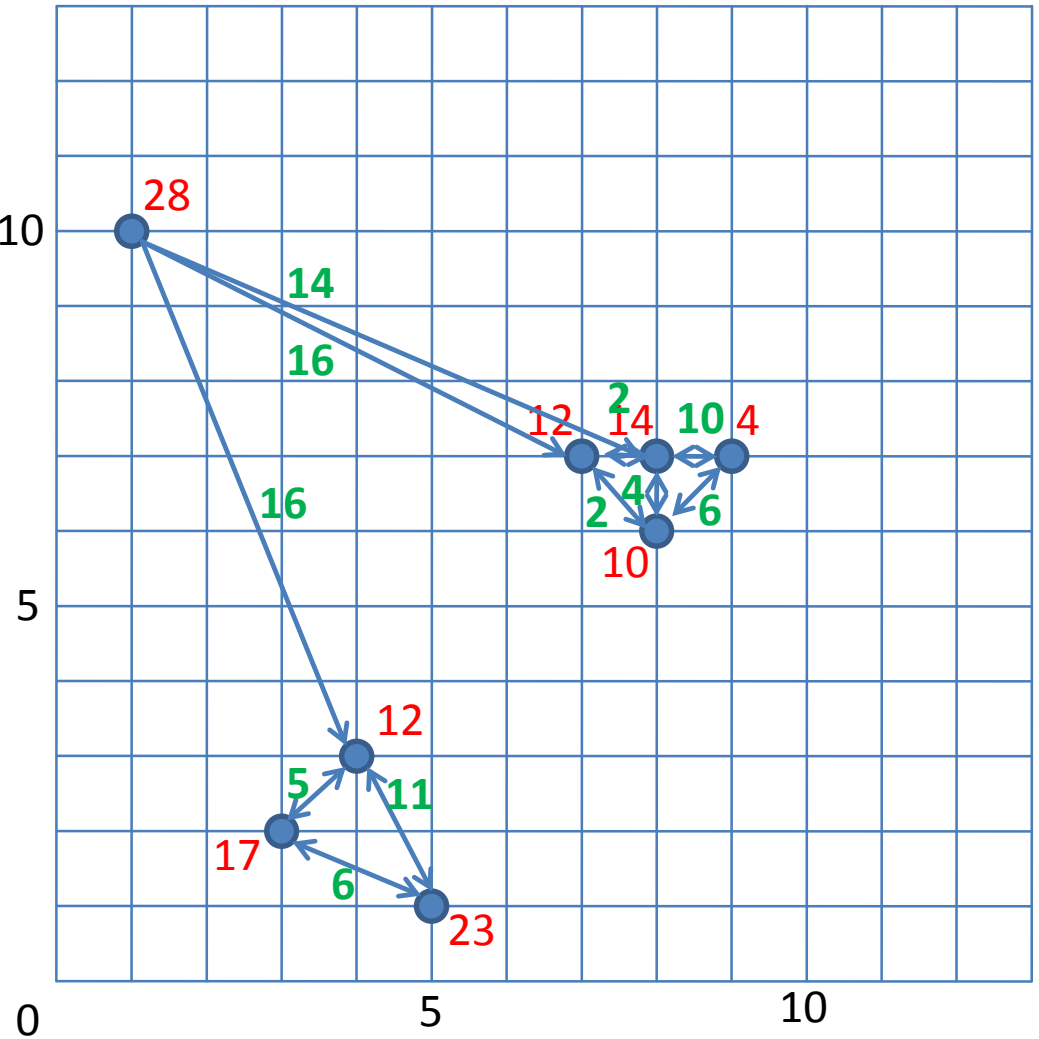


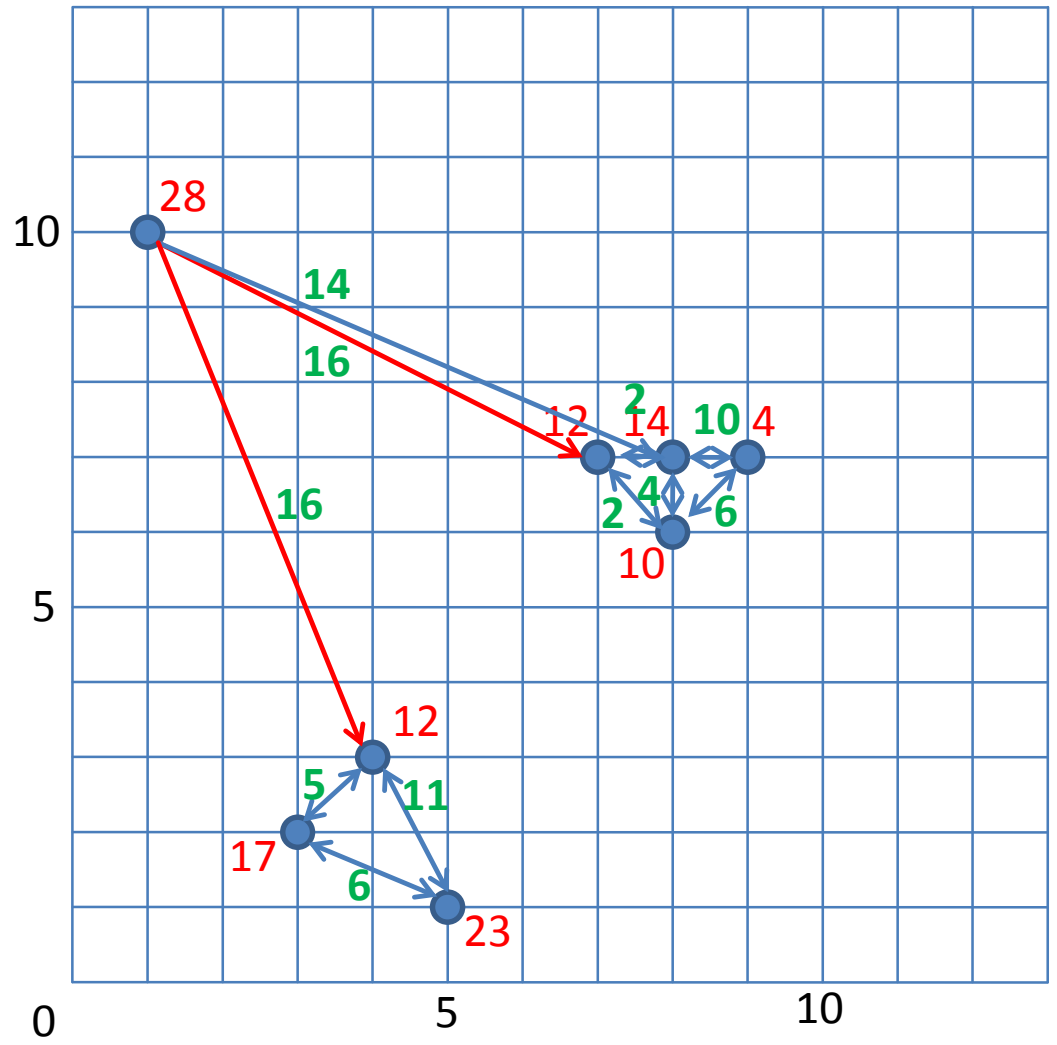


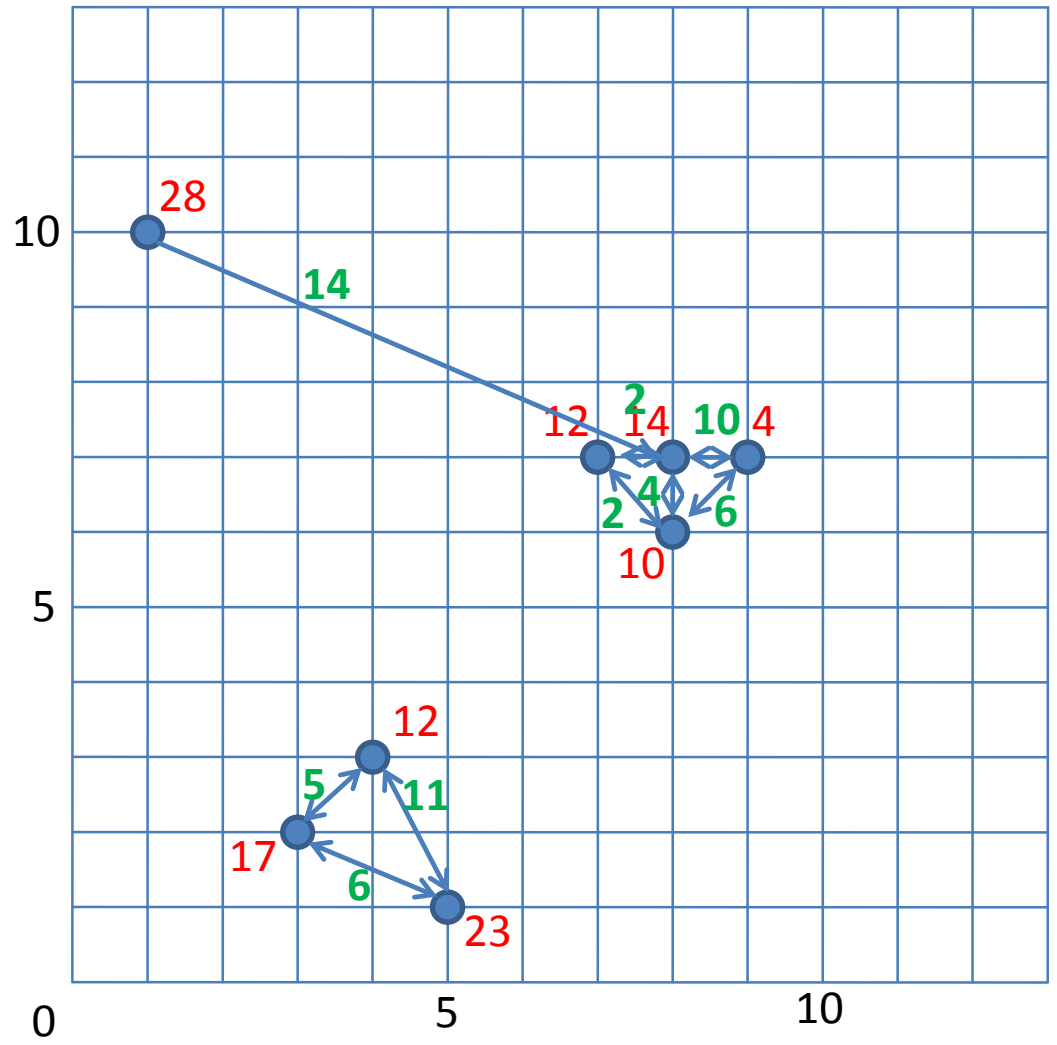


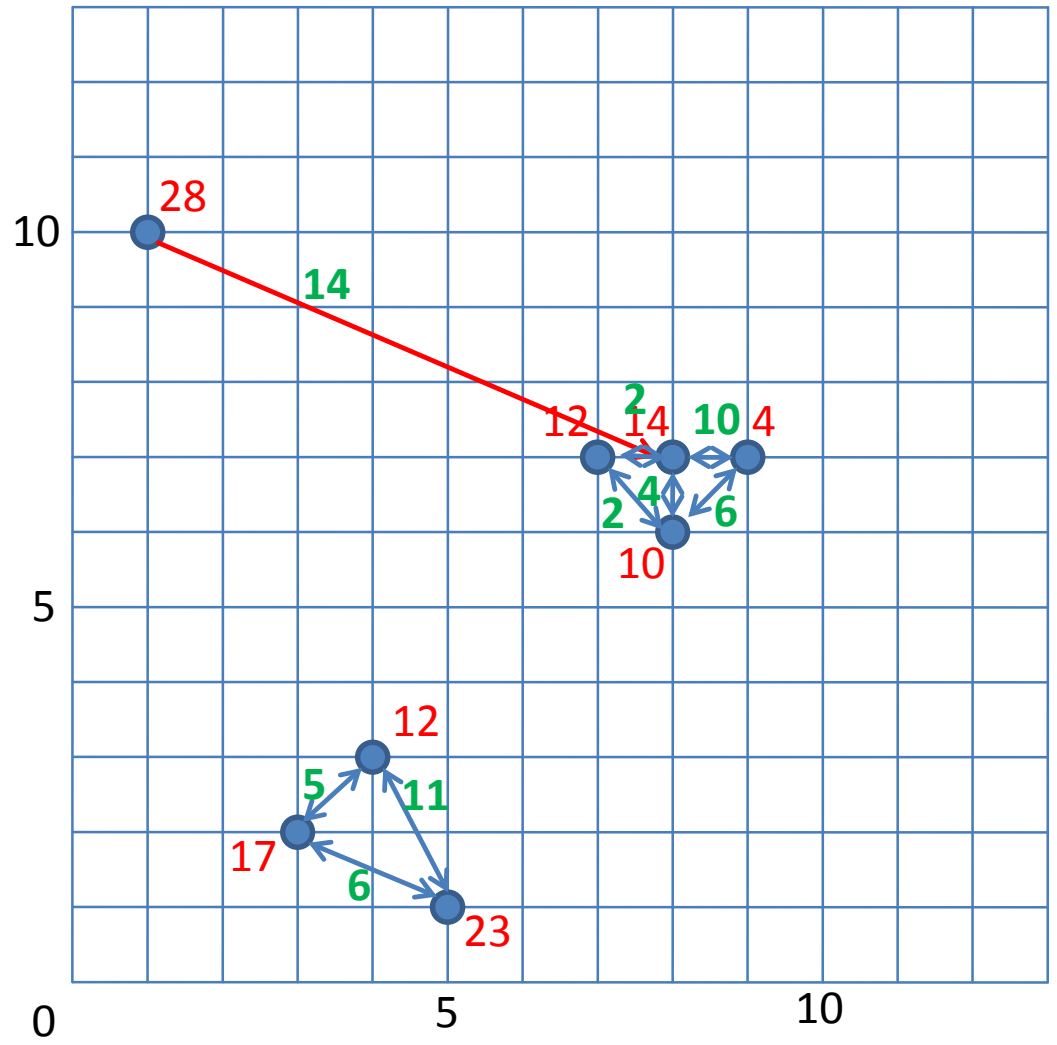


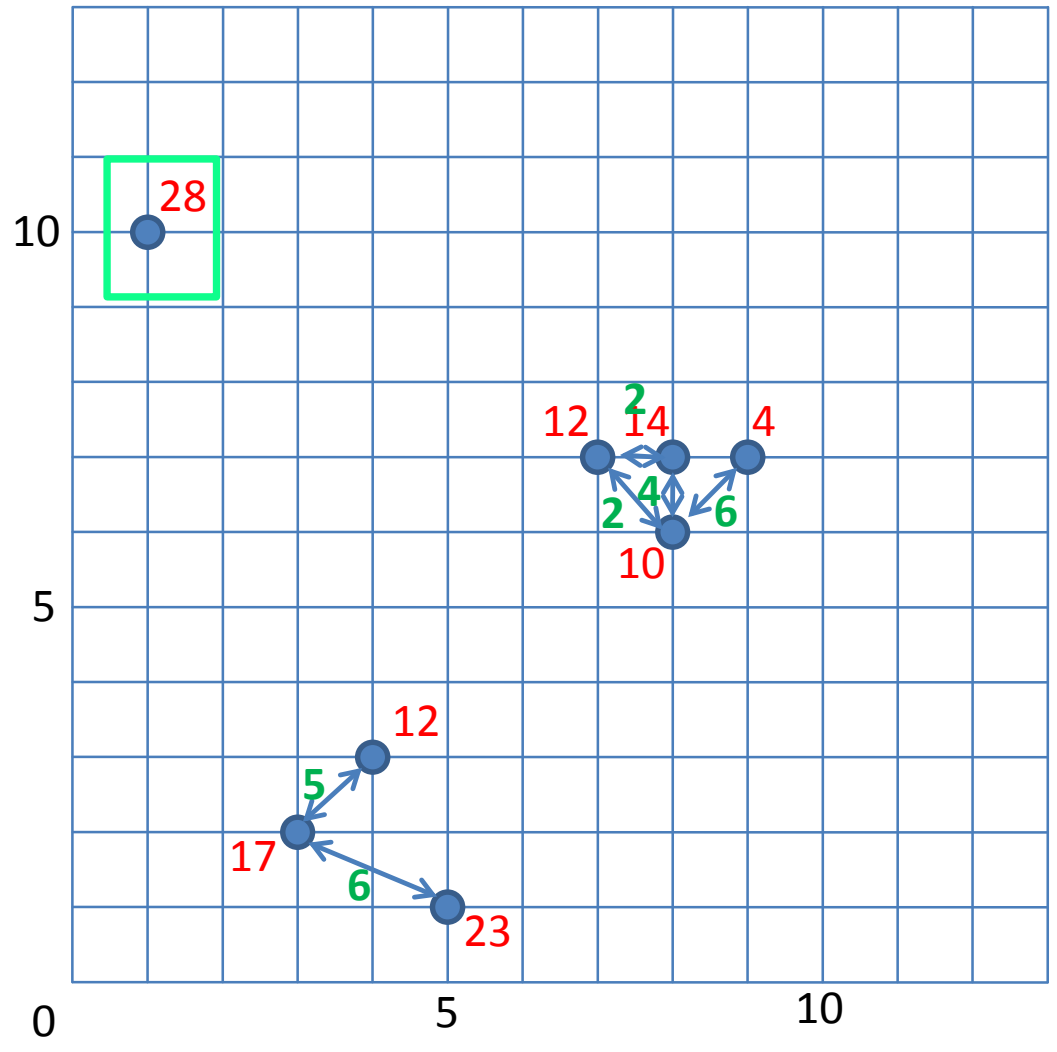
Nachbarschaftsgraph aufgebaut.
Entferne nun der Reihe nach
(absteigend) die Kanten mit dem
höchsten Gewicht (Score-Differenz) aus dem Graphen.











Die drei stärksten Outlier sind somit bestimmt.

