Programming Exercise 2 - Combinatorial Optimization

Ivana Koroman
Lucas Keiler
Noah Schaumburg

January 2, 2021

Algorithm 1 Minimum Mean Weight Cycle

procedure MinMeanWeightCycle
  if G has no cycle **then**
    return
  $\gamma \leftarrow max\{c(e) : e \in E(G)\}$
  $T \leftarrow \emptyset$
  $c'(e) := c(e) - \gamma$
loop:
  $J \leftarrow$ *MinWeightEmptyJoin()*
  if $c'(J) = 0$ **then**
    return $0 - c'$−weight-cycle
  else
    $\gamma' \leftarrow \frac{c'(J)}{|J|}$
    $c'(e) \leftarrow c'(e) - \gamma', \forall e \in E(G)$

Algorithm 2 Minimum Weight $\emptyset-$Join

procedure MinWeightEmptyJoin
  $T \leftarrow V^-$
  $G_i \leftarrow G(T)$, with cost function $d(e) := |c(e)|$
  $\overline{G}_i \leftarrow$ `MetricClosure`(G_i)
  $M \leftarrow$ `MinWeightPerfectMatching(`G_i`)`, $\forall i$
  $J \leftarrow P_{\{x_1,y_1\}} \triangle \ldots \triangle P_{\{x_m,y_m\}}$, where $P_{x,y}$ is the min x-y-path
  `return` $J \triangle E^-$

Algorithm 1 was in exercise 7.4. Algorithm 2 is (almost explicitly) stated in Theorems 51 and 52 of the lecture notes. For Algorithm 2, E^- is the set of edges with negative weight, V^- the set of vertices that are incident with an odd number of edges in $E-$