Solutions to the Exam "Discrete Optimization"

Monday, January 12, 2015, 14:00 – 17:00

1. Independent Set

First variant. We show that the set of independent sets in graphs of degree at most Δ form a Δ-troid. This implies that the greedy algorithm is a Δ-approximation. We have unit weights. Hence, no sorting is required. We start with $U = \emptyset$ and go through the set of nodes. For every node, we check if it is available. If yes, then we put it into U and mark all its neighbors as unavailable. The running-time is immediate.

It is clear that independent sets form an independent set system. We have to prove that if U and Z are independent sets with $|Z| > \Delta \cdot |U|$, then there exists some node $v \in Z \setminus U$ such that $U \cup \{v\}$ is an independent set.

For every $v \in Z \cap U$, we have $N(v) \cap Z = \emptyset$ since Z is an independent set. Let $\ell = |Z \cap U| \in \{0, 1, \ldots, |U|\}$, and let $U' = U \setminus Z$. Thus, $|U'| = |U| - \ell$.

Then $|Z \setminus (U \cup N(U))| \geq |Z| - \ell - \Delta \cdot |U'| = |Z| - \ell - \Delta \cdot (|U| - \ell) \geq |Z| - \Delta \cdot |U| \geq 1$. Hence, there exists a node $v \in Z \setminus U$ with $v \notin N(U)$. Hence, $U \cup \{v\}$ is an independent set with one more node.

Second variant. For the analysis of the greedy method, we construct two sets U and F. The set F contains all nodes of $U^\star$ (the independent set of maximum cardinality) that cannot be added to U anymore either because they are already part of U or because a neighbor is contained in U. We claim that $|F| \leq \Delta \cdot |U|$ throughout the execution of the greedy algorithm.

In the beginning, this is true because both sets are empty. For any node u that is added to U by the greedy method, one of the following cases happens:

- $u \in U^\star$. Then either $u \in U^\star$. Then only u is added to F because none of u's neighbors is in $U^\star$ because $U^\star$ is an independent set.
- $u \notin U^\star$. Since u has at most Δ neighbors, at most Δ vertices have to be added to F.

In either case, the invariant $|F| \leq \Delta \cdot |U|$ is maintained. Since $F = U^\star$ in the end (otherwise, the greedy algorithm would add more vertices to U), the approximation ratio is proved.

Ratio $\Delta + 1$. The proof that the set of independent sets in graphs of degree at most Δ form a $(\Delta + 1)$-troid is similar but considerably simpler than the proof that it is a Δ-troid. If $|Z| > (\Delta + 1) \cdot |U|$, then there exists a node $v \in Z \setminus (N(U) \cup U)$, and any such node can be added to U to obtain a larger independent set.

Similarly, we can choose F to be the set of all nodes (including the nodes of U) that cannot be taken anymore. Then $|F| \leq (\Delta + 1) \cdot |U|$ throughout the execution of the algorithm. In the end, we have $F = V$. Since $U^\star \subseteq V$, we obtain an approximation ratio of $\Delta + 1$.

2. Shortest Paths

We use the following algorithm:

1: $d(v_1) \leftarrow 0$
2: $d(v_i) \leftarrow \infty$ for all $i \in \{2, 3, \ldots, n\}$
3: **for** $i = 1$ **to** n **do**
4:     **for all** j with $(v_i, v_j) \in E$ **do**
5:         RELAX(v_i, v_j)
6:     **end for**
7: **end for**

The running-time of the algorithm is $O(n + m)$ as each vertex is used once as v_i, and each edge is considered exactly once.

We have to prove the correctness of the algorithm. We observe that the input graph does not contain any cycles.

The correctness proof is by induction on the index i of the vertices. For $i = 1$, the algorithm is clearly correct is v_1 is not part of any cycle, thus not part of any negative cycle.

Now assume that $d(v_1), \ldots, d(v_{i-1})$ are correct when considered in the loop in line 3. We know from the lecture that $\delta(v_i) \leq d(v_i)$. We have to prove that $\delta(v_i) = d(v_i)$. Consider a shortest v_1-v_i path P, and let (v_j, v_i) be the last edge of P. By induction hypothesis, $d(v_j) = \delta(v_j)$. Hence, $d(v_i) = \delta(v_j) + c(v_j, v_i) = \delta(v_i)$. The vertex v_i is only later considered in the for loop in line 3. The induction statement holds for all i, and the correctness follows.

3. Spanning Trees

(a) Since $T^\star$ is a minimum spanning tree, we have $w(T^\star) \leq w(Y)$. Let e be any edge of $Y \setminus T^\star$, which exists since $|Y| = |T^\star|$ and $T^\star \triangle Y \neq \emptyset$. The graph $T^\star \cup \{e\}$ contains a cycle C since $T^\star$ is a tree. There must be some edge $e^\star \in C$ with $e^\star \notin Y$ since Y is a tree.

We must have $w(e^\star) \leq w(e)$ since $T^\star$ is a minimum spanning tree. Otherwise, we could obtain a lighter tree $(T^\star \setminus \{e^\star\}) \cup \{e\}$.

Also $Z = (Y \setminus \{e\}) \cup \{e^\star\}$ is a spanning tree by the choice of e and $e^\star$. We have $w(Z) = w(Y) - w(e) + w(e^\star) \leq w(Y)$. Since $T^\star$ is a minimum-weight spanning tree, we have $w(T^\star) \leq w(Z)$. By construction, $|Z \triangle T^\star| = |Y \triangle T^\star| - 2$.

(b) Let $T^\star$ be a minimum-weight spanning tree. Consider any second-lightest tree S. By applying the above statement iteratively to $T^\star$ and S, we arrive at a spanning tree S' with $|T^\star \triangle S'| = 2$. By construction, $w(T^\star) \leq w(S') \leq w(S) \leq w(T)$ for all spanning trees $T \neq T^\star$ of G.

(c) We compute a minimum spanning tree. According to Part (b), there exists a tree S as desired with $|T^\star \triangle S| = 2$. For all edges $e \notin T^\star$, we add e to $T^\star$ and remove the heaviest edge of the unique cycle of the graph thus obtained. This yields a tree S_e. We select as S the tree S_e that minimizes $c(S_e)$.

The correctness follows from Part b and the choice of the edge for removal and the choice of S.

The running-time is $O(n^3)$ or $O(mn)$.

4. NP-Completeness

We show HamiltonPath $\preceq$ BoundedMST. Let $G = (V, E)$ be an instance of HamiltonPath. We construct a graph $H = (U, F)$ as follows:

- $U = V \cup \{v' \mid v \in V\}$.
- $F = E \cup \{\{v, v'\} \mid v \in V\}$.

Clearly, the mapping $G \mapsto H$ is polynomial-time computable. We have to prove that $G \in$ HamiltonPath if and only if $H \in$ BoundedMST.

"$\Rightarrow$": Let $G \in$ HamiltonPath. There there exists a simple path P that connects all vertices in V. We claim that $T = P \cup \{\{v, v'\} \mid v \in V\}$ is a bounded-degree spanning tree of H:

- Every two vertices of U are connected. This follows by the construction of T.
- T has a maximum degree of 3: Since P is a simple path, every node of V has a maximum degree of 2 in P. Exactly one edge is added to each $v \in V$, thus all vertices in V have a maximum degree of 3 in T. All vertices v' with $v \in V$ have a degree of 1 in T by construction of T.

We conclude that T is a bounded-degree spanning tree. Thus, $H \in$ BoundedMST.

"$\Leftarrow$": Let T be a bounded-degree spanning tree of H. Since every v' with $v \in V$ must be connected to some other vertex and there is exactly one edge in H incident to v', we have $\{v, v'\} \in T$ for all $v \in V$. We remove all these edges and call the resulting graph P. We claim that P is a Hamilton path of G. P is a collection of paths since it has a maximum degree of 2 since we have removed exactly one edge of each vertex and T has a maximum degree of 3. Since we have only removed leaves from the tree T by construction of H, P connects all vertices of V. Thus, P is a Hamiltonian path. We conclude that $G \in$ HamiltonPath.

What remains to be proved is that BoundedMST $\in \mathcal{NP}$. If $G \in$ BoundedMST, then there exists a bounded-degree spanning tree T of G. Given such a tree, we can check in polynomial time that T is indeed a spanning tree and has a maximum degree of 3. If $G \notin$ BOUNDEDMST, then no such tree exists.

Since have proved that BOUNDEDMST is $\mathcal{NP}$-hard and in $\mathcal{NP}$, we conclude that BOUNDEDMST is $\mathcal{NP}$-complete.

5. Minimum Cost Flows

"(I) $\Rightarrow$ (II)". Assume that G_f contains a cycle C with $c(C) \leq 0$. We augment along this cycle to obtain a flow f'. If $c(C) = 0$, then $c(f) = c(f')$, and f cannot be the unique optimal flow. If $c(C) < 0$, then $c(f') < c(f)$ and f is not optimal.

"(II) $\Rightarrow$ (I)". Let f' be any other flow with $f' \neq f$. We know from the lecture that f' can be obtained from f by augmenting along cycles $C_1, \ldots, C_k$ in G_f for some $k \in \mathbb{N}$. By assumption, $c(C_i) > 0$. Since $f' \neq f$, we augment along at least one cycle. Hence, $c(f') > c(f)$. Since f' is arbitrary, f is the unique optimal flow.

6. Questions

(a) Yes. By the lecture, there exists an optimal integral flow. Since the costs are integral, its costs are integral as well. Hence, every optimal flow has integral costs.

(b) No. Knapsack is $\mathcal{NP}$-complete. Hence, 3SAT $\preceq$ Knapsack.

(c) Yes. PERFECTMATCH $\in \mathcal{P} \subseteq \mathcal{NP}$.

(d) Yes. Consider any shortest path from s to some node t. Then the first vertex after s on this path is such a vertex by the subtour property.