Arc-partitioning and vertex-ordering problems

Nóra Anna Borsik

Supervisor: Péter Madarasi

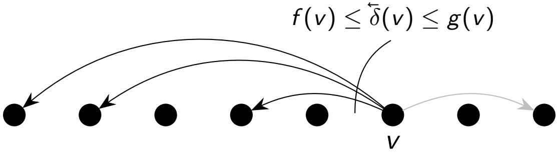
Math project III., 2024

First two semesters: The $(f, g, \sum w)$-FAS problem

$(f, g; \sum w)$-FAS problem

Let us given a digraph $D = (V, A)$ with a weight function $w : A \to \mathbb{R}_+$, and lower and upper bound functions $f : V \to \mathbb{R}_+$ and $g : V \to \mathbb{R}_+$. Does there exist an order of the vertices such that the left weighted out-degree of each vertex v is between $f(v)$ and $g(v)$?

In the unweighted case, the problem is called the (f, g)-FAS problem.



Only upper bounds

Theorem

The $(-\infty, g, \sum w)$-FAS problem, where only upper bounds are given on the vertices, is polynomial-time solvable.

Algorithm: In each step, fix a vertex v which does not violate the upper bound $g(v)$ on the last free place, and delete v from D.
If no such vertex exists at some point, then no solution exists.

Only upper bounds

Theorem

The $(-\infty, g, \sum w)$-FAS problem, where only upper bounds are given on the vertices, is polynomial-time solvable.

Algorithm: In each step, fix a vertex v which does not violate the upper bound $g(v)$ on the last free place, and delete v from D.
If no such vertex exists at some point, then no solution exists.

The case when only lower bounds are given can be solved similarly.

Complexity

The previously discussed solvable cases become NP-complete after small modifications.

Theorem

The (f, g)-FAS problem is NP-complete if only upper bounds are given for all vertices except for a single vertex v for which $f(v) = g(v)$.

Complexity

The previously discussed solvable cases become NP-complete after small modifications.

Theorem

The (f, g)-FAS problem is NP-complete if only upper bounds are given for all vertices except for a single vertex v for which $f(v) = g(v)$.

Theorem

The $(-\infty, g; \sum w)$-FAS problem is NP-complete if negative arc-weights are allowed.

Partitioning into an in-branching and an acyclic subgraph

Theorem

A digraph can be partitioned into an in-branching and an acyclic subgraph if and only if the $(-\infty, g)$-FAS problem with $g \equiv 1$ is solvable.

Partitioning into an in-branching and an acyclic subgraph

Theorem

A digraph can be partitioned into an in-branching and an acyclic subgraph if and only if the $(-\infty, g)$-FAS problem with $g \equiv 1$ is solvable.

Theorem

It is NP-hard to find a minimum-size in-branching whose complement is acyclic.

Rank aggregation

Consider a competition, where different judges give complete rankings of the candidates.

Goal: Find a “fair” common ranking.

Rank aggregation

Consider a competition, where different judges give complete rankings of the candidates.

Goal: Find a "fair" common ranking.

For a candidate v, let $d(v)$ denote the number of the candidates whom v precedes by the majority of the judges, but not in the common ranking.

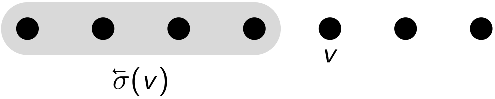
We can minimize the maximum distance $d(v)$ over all candidates by a reduction to the $(-\infty, g)$-FAS ordering problem.

Generalization

$(f, g; h)$-ordering problem

For a finite ground set V, let us given a non-decreasing set-function $h_v : 2^{V-v} \to \mathbb{R}$ for each item $v \in V$, and lower and upper bound functions $f : V \to \mathbb{R}$ and $g : V \to \mathbb{R}$.
Does there exists an order σ of V such that, for each item v, the set $\overleftarrow{\sigma}(v)$ of the items preceding v satisfies $f(v) \leq h_v(\overleftarrow{\sigma}(v)) \leq g(v)$?

The cases where only upper or only lower bounds are given are polynomial-time solvable by modifying the algorithms given for the $(f, g; \sum w)$-FAS problem.



Sceduling: $1|g\text{-prec}|L_{max}$

Let us given some jobs with

- $p(v)$: processing time of v
- $d(v)$: due date of v
- „precedence" constraints (not necessarily acyclic)
- $g(v)$: upper bound for the number of the violated precedence constraints for v.

Goal: minimize the maximum lateness.

Sceduling: $1|g\text{-prec}|L_{max}$

Let us given some jobs with

- $p(v)$: processing time of v
- $d(v)$: due date of v
- „precedence" constraints (not necessarily acyclic)
- $g(v)$: upper bound for the number of the violated precedence constraints for v.

Goal: minimize the maximum lateness.

Consider the $(-\infty, g; h)$-ordering problem with
$h_v(V') = M(\delta(v, V') - g(v))^+ + (\sum_{u \in V'} p(u) + p(v) - d(v))^+$
$g' \equiv K$.
We can decide in polynomial time, whether there exists a schedule with maximum lateness K.

This semester: Egalitarian vertex-ordering problems

Let us given an undirected graph G. Our goal is to find a vertex-order of G such that the left degree vector is as "equitable" as possible.

This semester: Egalitarian vertex-ordering problems

Let us given an undirected graph G. Our goal is to find a vertex-order of G such that the left degree vector is as "equitable" as possible.

- dec-min problem: The goal is to lexicographically minimize the non-increasingly ordered left degree sequence.

This semester: Egalitarian vertex-ordering problems

Let us given an undirected graph G. Our goal is to find a vertex-order of G such that the left degree vector is as "equitable" as possible.

- dec-min problem: The goal is to lexicographically minimize the non-increasingly ordered left degree sequence.
- inc-max problem: The goal is to lexicographically maximize the non-decreasingly ordered left degree sequence.

This semester: Egalitarian vertex-ordering problems

Let us given an undirected graph G. Our goal is to find a vertex-order of G such that the left degree vector is as "equitable" as possible.

- dec-min problem: The goal is to lexicographically minimize the non-increasingly ordered left degree sequence.
- inc-max problem: The goal is to lexicographically maximize the non-decreasingly ordered left degree sequence.
- $\min_{v \in V} \sum_{v \in V} h(\overleftarrow{d}(v))$ problem: For a given strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$, the goal is to minimize the $\sum_{v \in V} h(\overleftarrow{d}(v))$ function.

This semester: Egalitarian vertex-ordering problems

Let us given an undirected graph G. Our goal is to find a vertex-order of G such that the left degree vector is as “equitable” as possible.

- dec-min problem: The goal is to lexicographically minimize the non-increasingly ordered left degree sequence.
- inc-max problem: The goal is to lexicographically maximize the non-decreasingly ordered left degree sequence.
- $\min_{v \in V} \sum_{v \in V} h(\overleftarrow{d}(v))$ problem: For a given strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$, the goal is to minimize the $\sum_{v \in V} h(\overleftarrow{d}(v))$ function.

Each vertex-ordering problem above can be equivalently rephrased as an acyclic orientation problem.

Egalitarian *orientation* problems

Let us given an undirected graph G. Our goal is to find an orientation D of G such that the indegree vector of D is as "equitable" as possible.

- dec-min problem: The goal is to lexicographically minimize the non-increasingly ordered indegree sequence.
- inc-max problem: The goal is to lexicographically maximize the non-decreasingly ordered indegree sequence.
- $\min_{v \in V} \sum_{v \in V} h(\varrho(v))$ problem: For a given strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$, the goal is to minimize the $\sum_{v \in V} h(\varrho(v))$ function.

Theorem (Frank, Murota (2022))

The optimal orientations coincide for the dec-min, the inc-max and the $\min \sum_{v \in V} h(\varrho(v))$ *orientation problems.*

Egalitarian *orientation* problems

Let us given an undirected graph G. Our goal is to find an orientation D of G such that the indegree vector of D is as "equitable" as possible.

- dec-min problem: The goal is to lexicographically minimize the non-increasingly ordered indegree sequence.
- inc-max problem: The goal is to lexicographically maximize the non-decreasingly ordered indegree sequence.
- $\min_{v \in V} \sum_{v \in V} h(\varrho(v))$ problem: For a given strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$, the goal is to minimize the $\sum_{v \in V} h(\varrho(v))$ function.

Theorem (Frank, Murota (2022))

The optimal orientations coincide for the dec-min, the inc-max and the $\min \sum_{v \in V} h(\varrho(v))$ *orientation problems.*

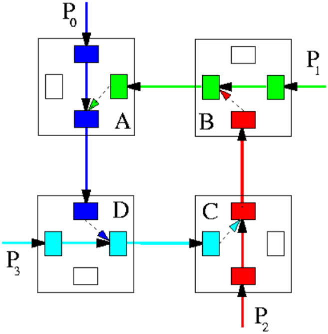
The $\min \sum_{v \in V} h_v(\varrho(v))$ orientation problem is also solvable.

Application: Routing protocol

Let us given a G=(V,E) undirected network, with capacities on the vertices, and packets with start and end destinations.

Such networks can suffer form deadlock if the packets are waiting in a cycle in which every vertex reached its capacity.

Routing protocol: Consider an order of the vertices, and forbid the packets to use the length-two paths formed by edges that are going to the left from the same vertex.

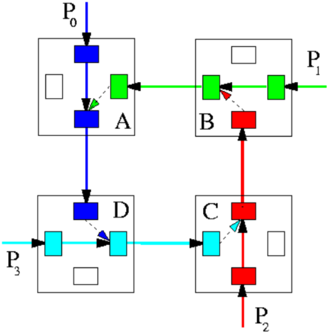


Application: Routing protocol

Let us given a G=(V,E) undirected network, with capacities on the vertices, and packets with start and end destinations.

Such networks can suffer form deadlock if the packets are waiting in a cycle in which every vertex reached its capacity.

Routing protocol: Consider an order of the vertices, and forbid the packets to use the length-two paths formed by edges that are going to the left from the same vertex.



- dec-min: Minimizes the maximum number of forbidden length-two paths for a vertex.
- $\min \sum_{v \in V} \overleftarrow{d}^2(v)$: Minimizes the number of forbidden length-two paths.

The complexities of the dec-min and inc-max problems

We call an order k-bounded, if every left degree is at most k (i.e. it is an $(-\infty, g)$-FAS order for $g \equiv k$).

Theorem (Borradaile, Iglesias, Migler, Ochoa, Wilfong, Zhang (2017))

The dec-min ordering problem is NP-hard, even in case of k-bounded orders for every odd $k \geq 5$.

The complexities of the dec-min and inc-max problems

We call an order k-bounded, if every left degree is at most k (i.e. it is an $(-\infty, g)$-FAS order for $g \equiv k$).

Theorem (Borradaile, Iglesias, Migler, Ochoa, Wilfong, Zhang (2017))

The dec-min ordering problem is NP-hard, even in case of k-bounded orders for every odd $k \geq 5$.

Theorem

The dec-min and the inc-max ordering problems are NP-hard, even in case of k-bounded orders for all $k \geq 3$.

The complexities of the dec-min and inc-max problems

We call an order k-bounded, if every left degree is at most k (i.e. it is an $(-\infty, g)$-FAS order for $g \equiv k$).

Theorem (Borradaile, Iglesias, Migler, Ochoa, Wilfong, Zhang (2017))

The dec-min ordering problem is NP-hard, even in case of k-bounded orders for every odd $k \geq 5$.

Theorem

The dec-min and the inc-max ordering problems are NP-hard, even in case of k-bounded orders for all $k \geq 3$.

The dec-min and the inc-max ordering problems coincide in case of 2-bounded orders, but the complexity remains open.

The min $\sum_{v \in V} h(\overleftarrow{d}(v))$ ordering problem

Let us given a strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$ and an undirected graph $G = (V, E)$.
Goal: Finding an order that minimizes the objective function $\sum_{v \in V} h(\overleftarrow{d}(v))$.

Important strictly convex functions:

- $h(z) = |V|^z$: dec-min ordering problem,
- $h(z) = |V|^{-z}$: inc-max ordering problem,
- $h(z) = z^2$.

The complexity of the min $\sum_{v \in V} h(\overleftarrow{d}(v))$ ordering problem

Theorem

For any strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$, it is NP-hard to find a vertex order minimizing $\sum_{v \in V} h(\overleftarrow{d}(v))$ in case of multigraphs.

The complexity of the min $\sum_{v\in V} h(\overleftarrow{d}(v))$ ordering problem

Theorem

For any strictly convex function $h : \mathbb{Z}_+ \to \mathbb{R}$, it is NP-hard to find a vertex order minimizing $\sum_{v\in V} h(\overleftarrow{d}(v))$ in case of multigraphs.

Theorem

For any linear functions $h_v : \mathbb{Z}_+ \to \mathbb{R}$ the problem of finding a vertex order minimizing $\sum_{v\in V} h_v(\overleftarrow{d}(v))$ is polynomial-time solvable.

Exact dynamic program for $\min \sum_{v \in V} h_v(\overleftarrow{d}(v))$

Let $f(V')$ denote the optimum for the graph induced by the vertex set V'.

$$f(V') = \min_{v \in V'} \{ f(V' - v) + h_v(d(v, V')) \}$$

Theorem

We can compute the value of $f(V')$ for every subset $\emptyset \neq V' \subseteq V$ in non-decreasing order by $|V'|$ in $O(2^{|V|} poly(|V|, |E|))$ time, provided that the function h_v can be evaluated in polynomial time for each $v \in V$.

The min $\sum_{v \in V} \overleftarrow{d}^2(v)$ ordering problem

Greedy algorithm: Fix a vertex with minimum degree at the last free place in the order and delete it from G until no vertex remains.

Theorem

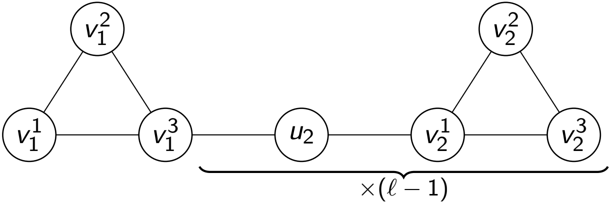
The greedy order gives a $\min\{4H_n, k_{\min}\}$*-approximate order, where* H_n *is the* n^{th} *harmonic number and* $k_{\min}$ *denotes the smallest integer* k *for which the graph has a* k*-bounded order.*

The min $\sum_{v \in V} \overleftarrow{d}^2(v)$ ordering problem

Greedy algorithm: Fix a vertex with minimum degree at the last free place in the order and delete it from G until no vertex remains.

Theorem

The greedy order gives a $\min\{4H_n, k_{\min}\}$*-approximate order, where* H_n *is the* n^{th} *harmonic number and* $k_{\min}$ *denotes the smallest integer* k *for which the graph has a* k*-bounded order.*



A simple graph, for which the greedy order is $\frac{9}{7}$-approximate if $\ell \to \infty$.

“Balanced” vertex orders

The goal is to find a vertex order for which $\overleftarrow{d}(v) \approx \overrightarrow{d}(v)$.
A vertex order is perfectly balanced if $|\overleftarrow{d}(v) - \overrightarrow{d}(v)| \leq 1 \ \forall v \in V$.

"Balanced" vertex orders

The goal is to find a vertex order for which $\overleftarrow{d}(v) \approx \overrightarrow{d}(v)$.
A vertex order is perfectly balanced if $|\overleftarrow{d}(v) - \overrightarrow{d}(v)| \leq 1 \ \forall v \in V$.

Theorem (Biedl, Chan, Ganjali, Hajiaghayi, Wood (2005) and Kára, Kratochvíl, Wood (2005))

It is NP-hard to find a perfectly balanced vertex order, even for graphs with maximum degree 4.

Theorem (Biedl, Chan, Ganjali, Hajiaghayi, Wood (2005))

For graphs with maximum degree 3, we can find a perfectly balanced vertex order in polynomial time.

The max $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$ problem

Goal: Find a vertex order maximizing $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$.

Corollary

A max $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$ *problem is NP-hard, even for graphs with maximum degree 4.*

Theorem

For graphs with maximum degree 3, we can find a vertex order maximizing $\sum_{v \in V} \overleftarrow{d}(v)\overrightarrow{d}(v)$ *in polynomial time.*

Approximation

Theorem

A random vertex order is a 3-approximate solution in expectation.

This estimate is sharp for a star with 3 leaves.

We gave a de-randomized 3-approximating algorithm.

Open questions

- Partitioning a digraph into an in-arborescence and an acyclic subgraph.
- The complexity of the 2-bounded dec-min and inc-max problems.
- The complexity of the min $\sum_{v \in V} h(\overleftarrow{d}(v))$ problem
  - for simple graphs, and
  - for special graph classes (e.g. planar graphs).
- The approximation ratio of the greedy order for minimizing the square-sum of the left degrees.