The Complexity of Foremost Coverage of Time-varying Graphs

Eric Aaron[1] Danny Krizanc[2] Elliot Meyerson[3]

[1]Computer Science Department, Vassar College

[2]Department of Mathematics & Computer Science, Wesleyan University

[3]Department of Computer Science, UT Austin

Venice, Jan 28, 2015

Foremost Coverage in a Dynamic Environment

Goal: Given a map, a set of agents and set of critical locations

- navigate agents so that every location is visited at least once
- complete coverage as soon as possible
- map may change during navigation

Application domains: inspection, surveillance, coverage, etc. by robots in a disaster-prone or hostile domain, virtual agents in a mobile network, etc.

Our approach:

- use time-varying graphs (TVGs) to model dynamics
- analyze DMVP across central chain of TVG classes
- consider centralized, offline complexity

Time-varying graphs (TVGs)

A *TVG* [Casteigts et al. '12] is a five-tuple $\mathcal{G} = (V, E, \mathcal{T}, \rho, \zeta)$

- $G = (V, E)$ is called the *underlying graph* of $\mathcal{G}$
- $\mathcal{T} \subseteq \mathbb{T}$ is the *lifetime* of the system
- *presence function* $\rho(e, t) = 1 \iff$ edge $e \in E$ is available at time $t \in \mathcal{T}$
- *latency function* $\zeta(e, t)$ gives the time it takes to cross e if starting at time t (given that it is available)

We work with the unweighted discrete case:

- $\mathbb{T} = \mathbb{N}$
- $\zeta(e, t) = 1, \ \forall \ e, t$

Related concepts: Delay-tolerant networks, Evolving graphs, T-interval connected graphs, etc.

Journeys

TVG analog of path in (static) graphs is a journey:

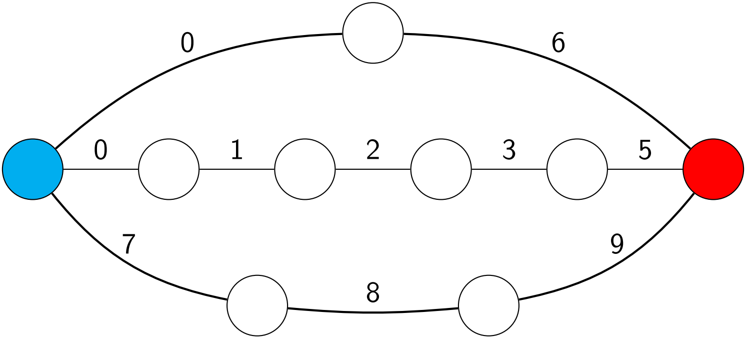
$\mathcal{J} = \{(e_1, t_1), \ldots, (e_k, t_k)\}$ is a *journey* iff

- $\{e_1, \ldots, e_k\}$ is a walk in G
- $\rho(e_i, t_i) = 1$ and $t_{i+1} > t_i$ for all $i < k$

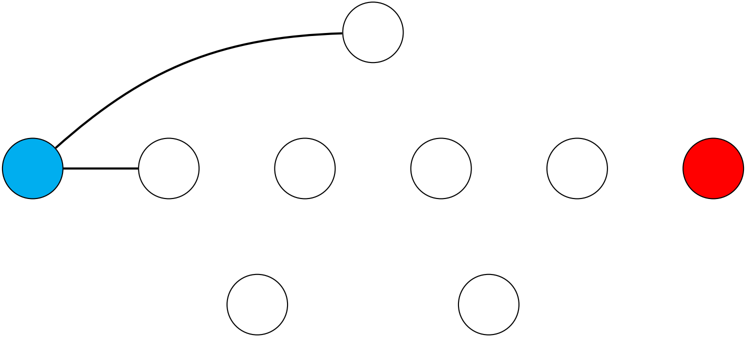
Types of minimal journeys starting on or after a given date t:

- *shortest*: k is minimal
- *fastest*: $(t_k + 1) - t_1$ is minimal
- **foremost**: $t_k + 1$ is minimal (counting from t=0)

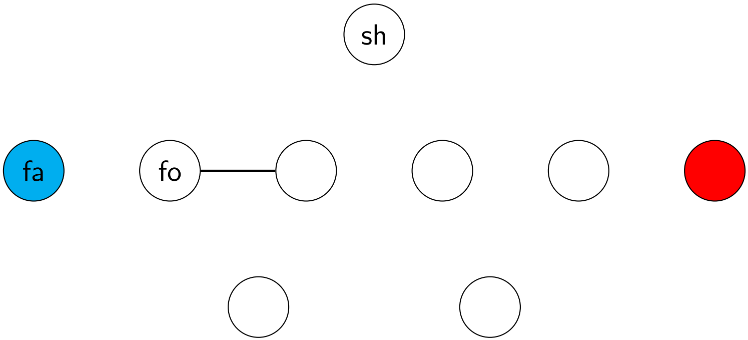
Example: Journeys in $\mathcal{G}$



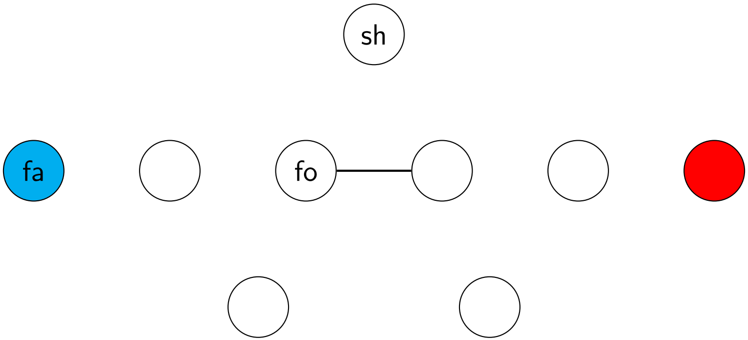
Example: Journeys in $\mathcal{G}$, $t = 0$



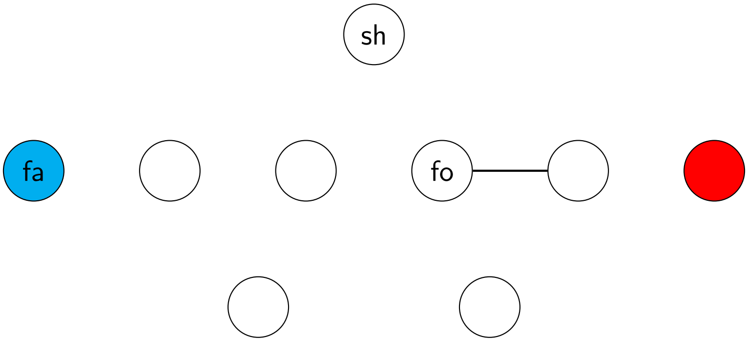
Example: Journeys in $\mathcal{G}$, $t = 1$



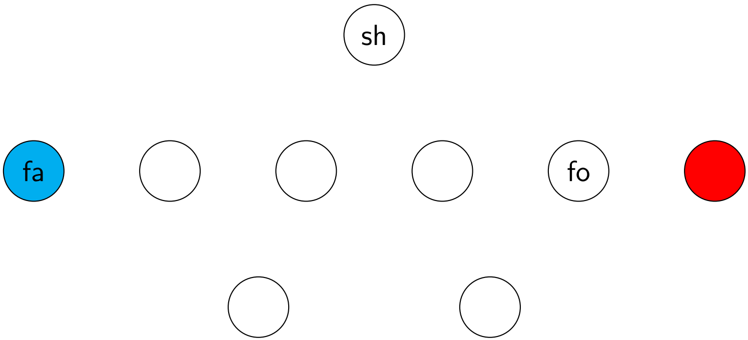
Example: Journeys in $\mathcal{G}$, $t = 2$



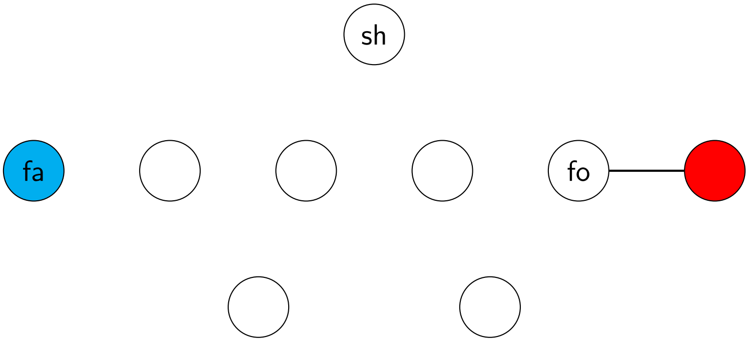
Example: Journeys in $\mathcal{G}$, $t = 3$



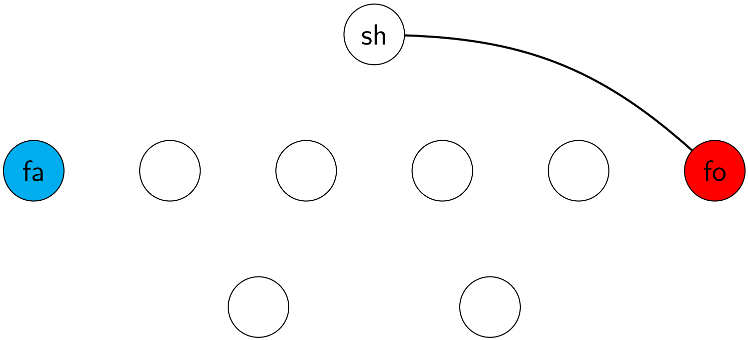
Example: Journeys in $\mathcal{G}$, $t = 4$



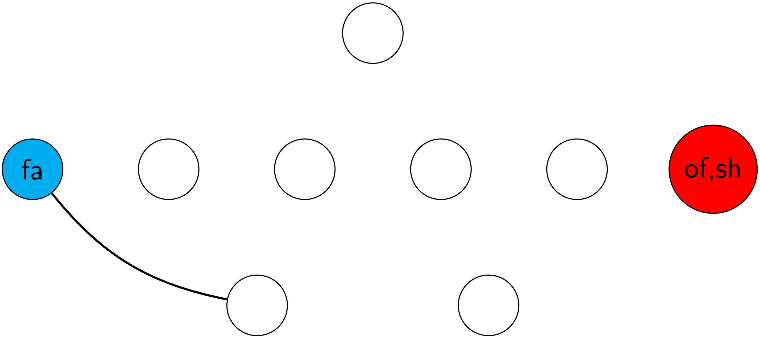
Example: Journeys in $\mathcal{G}$, $t = 5$



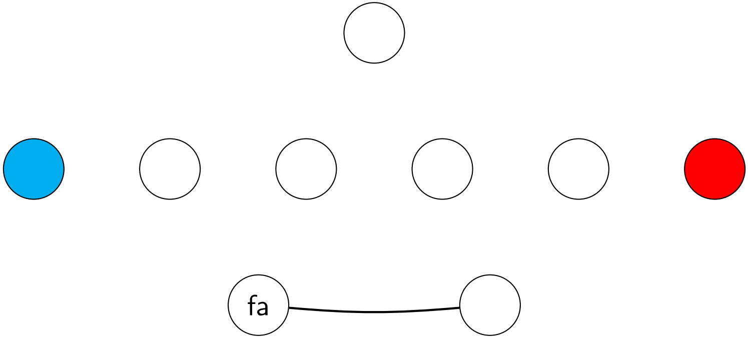
Example: Journeys in $\mathcal{G}$, $t = 6$



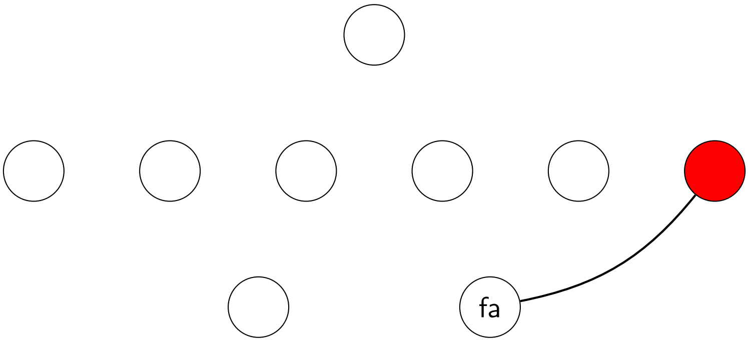
Example: Journeys in $\mathcal{G}$, $t = 7$



Example: Journeys in $\mathcal{G}$, $t = 8$



Example: Journeys in $\mathcal{G}$, $t = 9$



TVG classes: $\mathcal{R} \supset \mathcal{B} \supset \mathcal{P}$ [Casteigts et al. '12]

For each, assume underlying graph G is connected.

- (Edge-recurrent) $\mathcal{R}$ is the class of all TVG's $\mathcal{G}$ such that $\forall e \in E, \forall t \in \mathcal{T}, \exists t' > t$ s.t. $\rho(e, t') = 1$.

- (Time-bounded edge-recurrent) $\mathcal{B}$ is the class of all TVG's $\mathcal{G}$ such that $\forall e \in E, \forall t \in \mathcal{T}, \exists t' \in [t, t + \Delta)$ s.t. $\rho(e, t') = 1$, for some Δ.

- (Edge-periodic) $\mathcal{P}$ is the class of all TVG's $\mathcal{G}$ such that $\forall e \in E, \forall t \in \mathcal{T}, \forall k \in \mathbb{N}, \rho(e, t) = \rho(e, t + kp)$ for some p. p is called the *period* of $\mathcal{G}$.

Note: $\mathcal{G}$ can be disconnected at any moment.

Separations in $\mathcal{R} \supset \mathcal{B} \supset \mathcal{P}$

[Casteigts et al. '10] established a number of separations between the TVG classes $\mathcal{R}, \mathcal{B}, \mathcal{P}$.

They studied shortest, fastest and foremost broadcast with termination detection in the distributed on-line setting.

Their main result is to show how knowledge of n, Δ and p effects the feasibility of broadcast.

In particular they show:

$$\mathcal{R}_n \supsetneq \mathcal{B}_\Delta \supsetneq \mathcal{P}_p$$

where $\mathcal{X}_K$ is TVG class $\mathcal{X}$ with knowledge of K.

What about the offline complexity of exploration?

DMVP (Dynamic Map Visitation Problem)

Problem: Given a TVG $\mathcal{G}$ and a set of starting locations S for k agents in G find journeys for each of these k agents such that

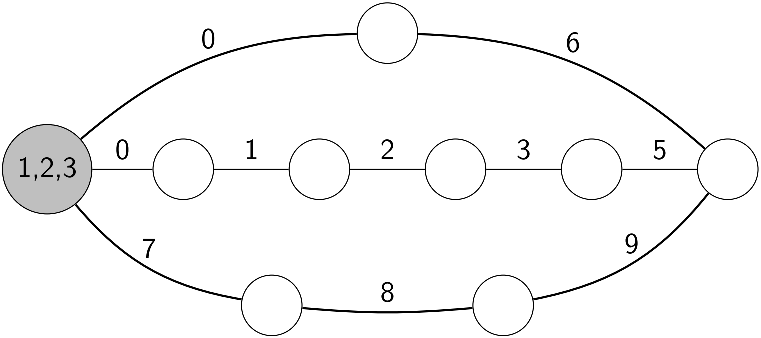
1. every node in V is in some journey
2. the maximum temporal length among all k journeys is minimized (starting at $t = 0$)

Decision variant: max temporal length $\leq t$.

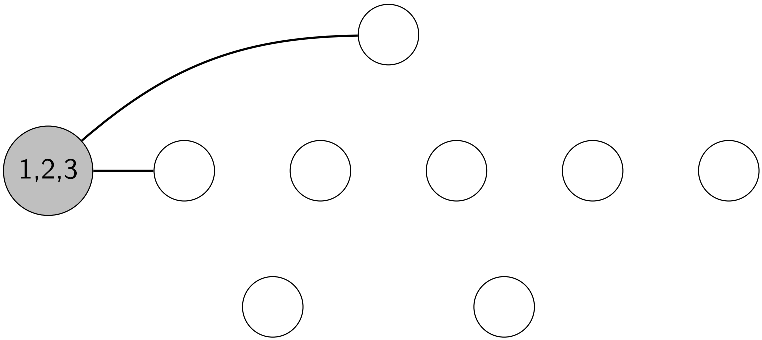
Essentially k-TSP with following distinctions:

- agents potentially start at different nodes (multi vs single depot)
- agents need not return to depot (with or without return)
- graph is a TVG

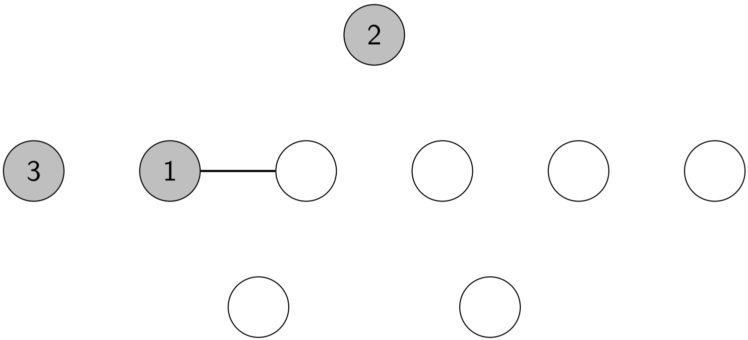
Example: DMVP over $\mathcal{G}$



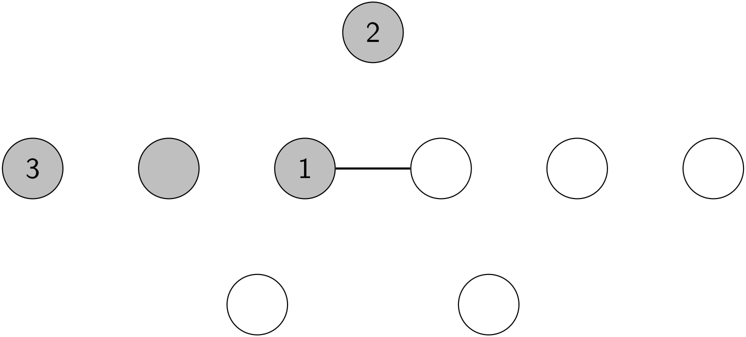
Example: DMVP over $\mathcal{G}$, $t = 0$



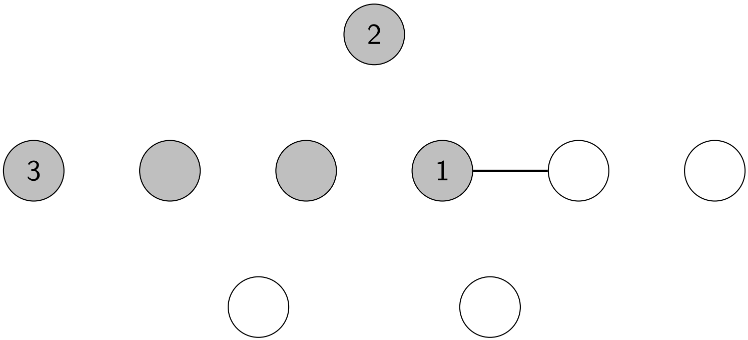
Example: DMVP over $\mathcal{G}$, $t = 1$



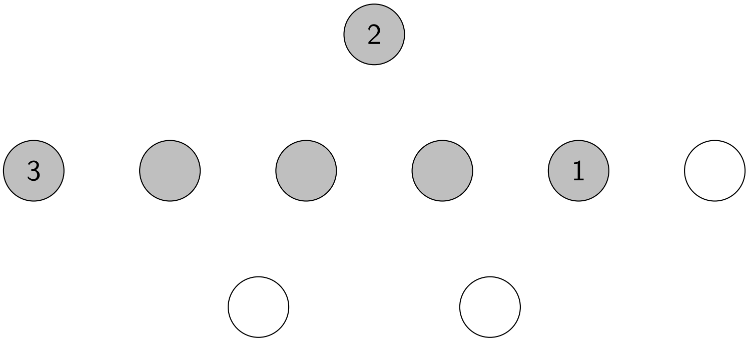
Example: DMVP over $\mathcal{G}$, $t = 2$



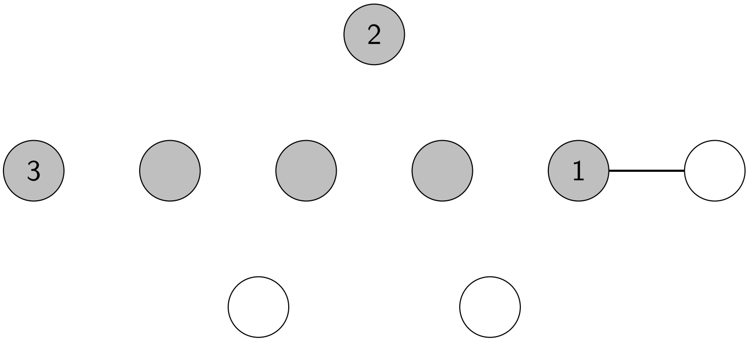
Example: DMVP over $\mathcal{G}$, $t = 3$



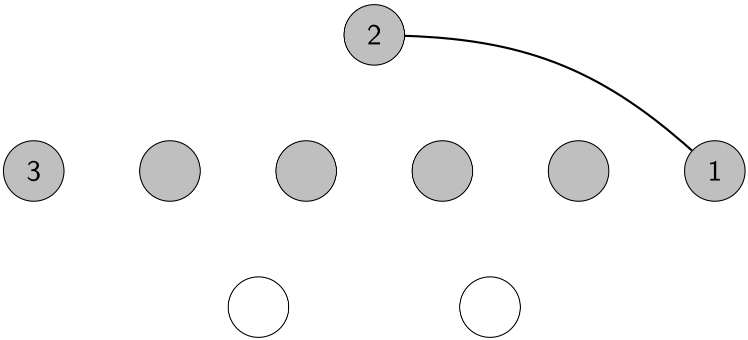
Example: DMVP over $\mathcal{G}$, $t = 4$



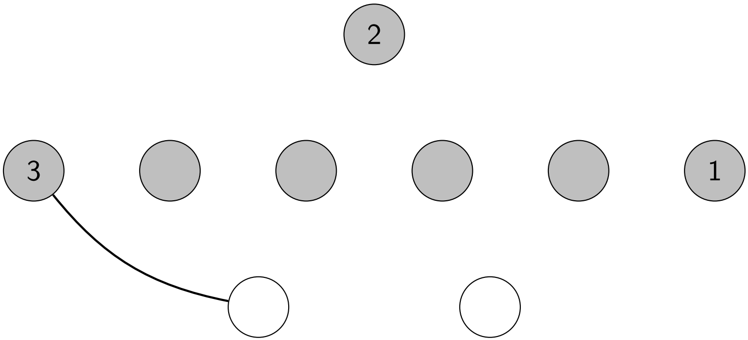
Example: DMVP over $\mathcal{G}$, $t = 5$



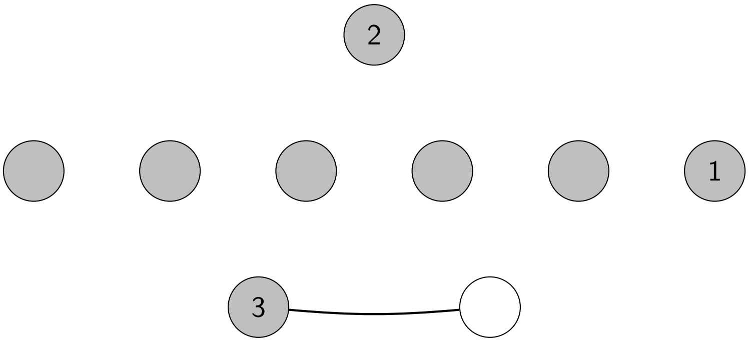
Example: DMVP over $\mathcal{G}$, $t = 6$



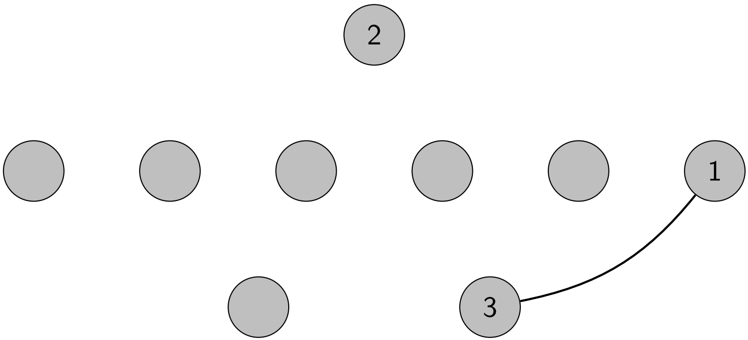
Example: DMVP over $\mathcal{G}$, $t = 7$



Example: DMVP over $\mathcal{G}$, $t = 8$



Example: DMVP over $\mathcal{G}$, $t = 9$



Presentation of $\mathcal{G}$ for offline case

Input: $\mathcal{G} = (G_1, t_1), (G_2, t_2), \ldots, (G_m, t_m)$ [Ferreira '04]

- G_i a static graph of G, t_i the duration of G_i.
- $T = \sum_{i=1}^{m} t_i$.

We think of $\mathcal{G}$ as being the prefix of some possibility infinite object.

Dealing with exponentiality of T:

- *Observation*: It is not necessary to consider each static temporal subgraph (G_i, t_i) for more than $2n - 3$ time steps.
- $T' = \sum_{i=1}^{m} \min(t_i, 2n - 3) < 2nm - 3m$
- We can think of T as T', thereby avoiding the exponential nature of T.
- Do this with $O(T')$ preprocessing step.
- Does not affect asymptotic runtimes.

Related results for offline case

- [Bui-Xuan, Ferreira and Jarry 2003] give polytime algorithms for computing shortest, fastest and foremost journeys in TVGs
- [Ferreira 2004] shows that deciding if there exists a strongly-connected component of size k in a TVG is NP-complete.
- [Mans and Mathieson 2013] show that testing certain properties of TVGs is fixed parameter tractable for graphs of bounded (local) tree-width.
- [Michail and Spirakis 2014] present polytime constant approximations for TSP with edge weights 1 and 2 on TVGs (shown to be APX-hard).

Our results

Lower bounds for a single agent:

- Hard to approximate to within any factor in $\mathcal{R}$ even on stars or degree 3 trees. (cf. [Michail and Spirakis 2014].)
- Hard to approximate to within better than Δ in $\mathcal{B}$ even on spiders or degree 3 trees.
- NP-complete in $\mathcal{P}$ for general graphs.

Upper bounds for a k agents:

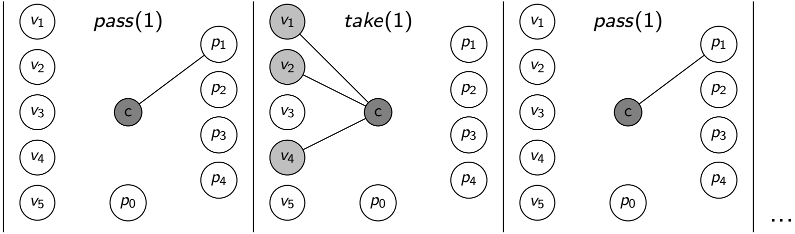
- $O(Tn)$ for a path; $O(Tn^2/k)$ for a cycle in $\mathcal{R}$.
- $O(Tn^{6c+1})$ for planar region with constant c subregions in $\mathcal{R}$.
- Fixed parameter algorithm for m-leaf c-almost trees in $\mathcal{R}$.

Upper bounds for 2 agents:

- $O(n^3)$ Δ-approximation for trees (tight) in $\mathcal{B}$.
- $O(n^5)$ algorithm on trees in $\mathcal{P}$ with $p = 2$.
- Linear time $\frac{12\Delta}{5}$-approximation for general graphs in $\mathcal{B}$.

Inapproximability in $\mathcal{R}$

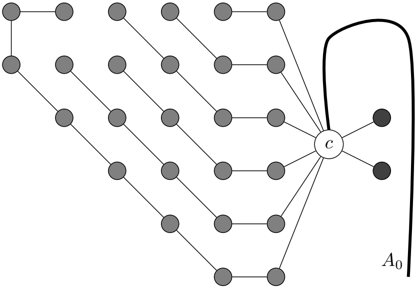
Thm. DMVP in $\mathcal{R}$ is NP-hard to approximate within any factor, even over stars.



Set cover: $U = \{1, 2, 3, 4, 5\}$, $S = \{\{1, 2, 4\}, \{2, 4\}, \{3, 4\}, \{3, 5\}\}$, $k = 2$

Inapproximability in $\mathcal{B}$

Thm. DMVP in $\mathcal{B}$ is NP-hard to approximate within any factor $< \Delta$, even over spiders.



3-partition input: $S = \{2, 3, 4, 4, 5, 8\}$

Complexity in $\mathcal{P}$

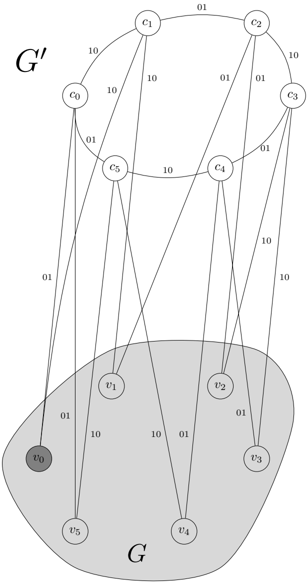
Thm. DMVP in $\mathcal{P}$ is NP-complete.

- $p = 1$ is static case, essentially TSP.

Thm. $O(n)$ over a tree with $p = 2$.

- Can only enter each subtree once.
- Partition subtrees into equivalence classes.

Thm. $\exists$ graphs such that $p = 1$ is trivial but $p = 2$ is NP-hard.



Paths/Cycles in $\mathcal{R}$

Observe:

- "No crossing" lemma
- Need only consider case where no two agents start at the same node

Together these yield an $O(Tn)$ dynamic programming algorithm for path.

For cycle:

- There exist two agents $\leq \frac{n}{k}$ apart
- Split cycle at each of these $\frac{n}{k}$ positions and apply path algorithm

Yields an $O(T\frac{n^2}{k})$ algorithm.

Planar region with constant c subregions in $\mathcal{R}$

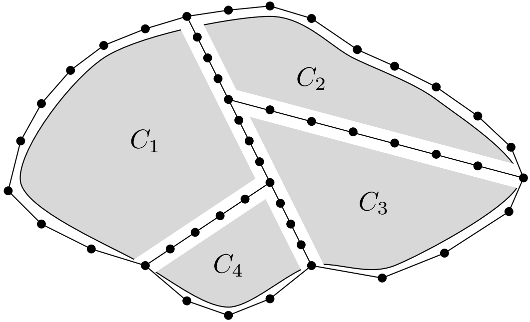


Figure: Border coverage graph extracted from a planar region subdivided into four subregions.

Planar regions in $\mathcal{R}$

- Graph consists of $O(c)$ paths of length $O(n)$ between $O(c)$ nodes of degree > 2.
- Each path starts with either 0 or > 0 agents.
- A path starting with 0 agents is covered by either one agent traveling its full length or by two agents entering from either end.
- A path starting with > 0 agents is either covered entirely by its starting agents or partially covered by these agents with either or both of the ends covered by at most two external agents.

Planar regions in $\mathcal{R}$

Algorithm idea:

- For each path guess:
  - how many external agents required (0, 1, 2),
  - which agents they are,
  - what portion do they cover.
- For each external agent we are left with a problem of covering a tree with $O(c)$ leaves which can be solved using an $O(Tn^3 + c^2 2^{O(c)})$ algorithm.
- For the internal agents we apply the path algorithm.
- Dominated by the cost of guessing where to cut the paths: $O(n^{6c+1})$.
- Similar approach gives fixed parameter tractable algorithm for m-leaf c-almost trees.

Two agents on a tree

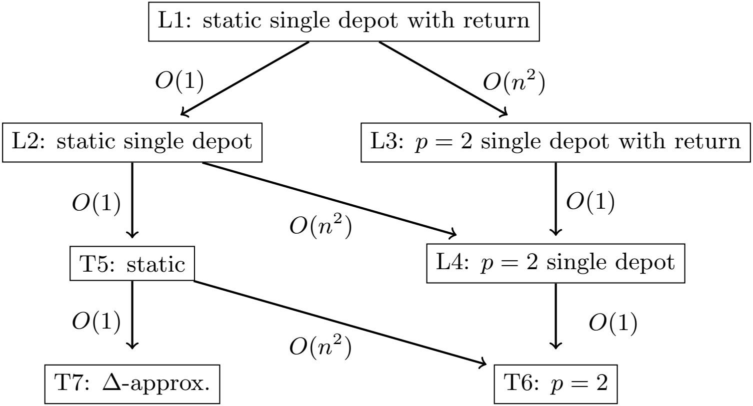


Figure: Poset of results leading to solutions for two-agent DMVP on a tree; arrows indicate increasing factors of complexity as constraints are loosened. L1 implied by [Dynai et al., 2006], [Xu et al., 2013].

Two agents on general graphs

Algorithm idea:

- Chose a spanning tree, T, of general graph G.
- Use Euler tour C of T of length $2n-1$.
- Observe that on a (static) cycle of length n there exists solution where neither agent visits more than $\frac{3n}{5}$ of the nodes
- Further observe that optimal coverage must take at least $\frac{n-1}{2}$ steps

Thm. $O(n)$-time $\frac{12\Delta}{5}$-approximation algorithm for general graphs.

Open Problems

Potential generalizations of above:

- More agents on a tree (pseudo-polytime algorithm)
- Fixed $p > 2$
- Larger underlying graph classes (e.g., bounded max-leaf number, poly number of spanning trees)

Open problems:

- Relation between $\mathcal{B}$ and $\mathcal{R}$: Is there a graph class C such that DMVP over C is tractable in $\mathcal{B}$, but NP-hard in $\mathcal{R}$? $\mathcal{B}$ with $\Delta = 2$ hard on a star?
- Apply TVGs to related problems
- Markovian TVGs