HOW TO GUIDE A PRESENT-BIASED AGENT THROUGH PRESCRIBED TASKS?

ARTUR IGNATIEV
YURIY DEMENTIEV

TATIANA BELOVA
FEDOR V. FOMIN
PETR A. GOLOVACH

27th European Conference on Artificial Intelligence ECAI 2024
19-24 October, 2024

PRESENT-BIAS PLANNING

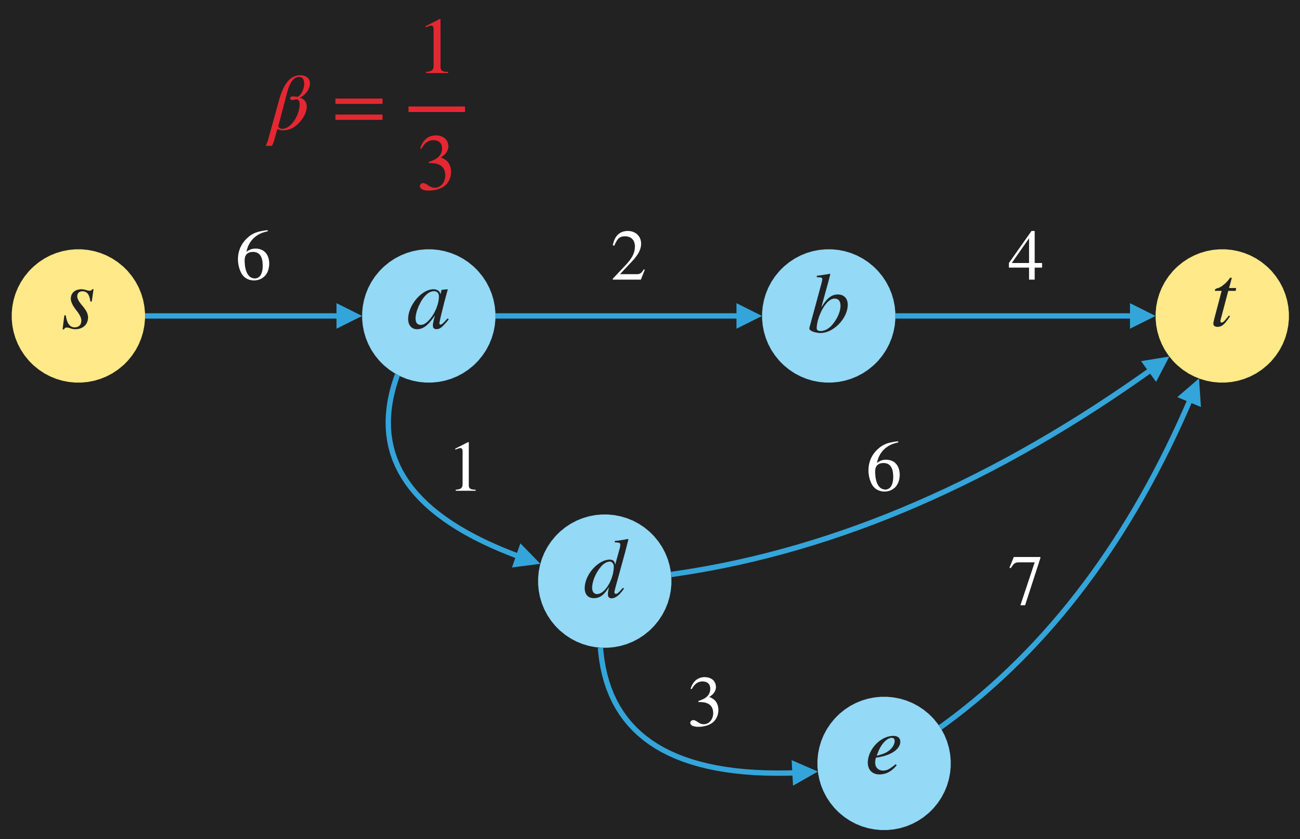
- Algorithmic Game Theory and Behavioral Economic.
- Study the impact of the gap between the anticipated costs of future actions and their real costs.
- Time-inconsistent planning: procrastination, abandonment, etc.
- **Akerlof (1991):** Graph theoretical model, where the cost of an action in the future is assumed to be β times smaller than its actual cost, for some $\beta < 1$.

KLEINBERG–OREN'S MODEL (EC 2014)

ECAI 50th ANNIVERSARY

5-tuple $M = (G, w, s, t, \beta)$, where:

- $G = (V(G), E(G))$ – DAG
- $w : E(G) \to \mathbb{N}$ – cost-function
- $s \in V(G)$ – start vertex
- $t \in V(G)$ – target vertex
- $\beta \leq 1$ – agent's present-bias parameter.

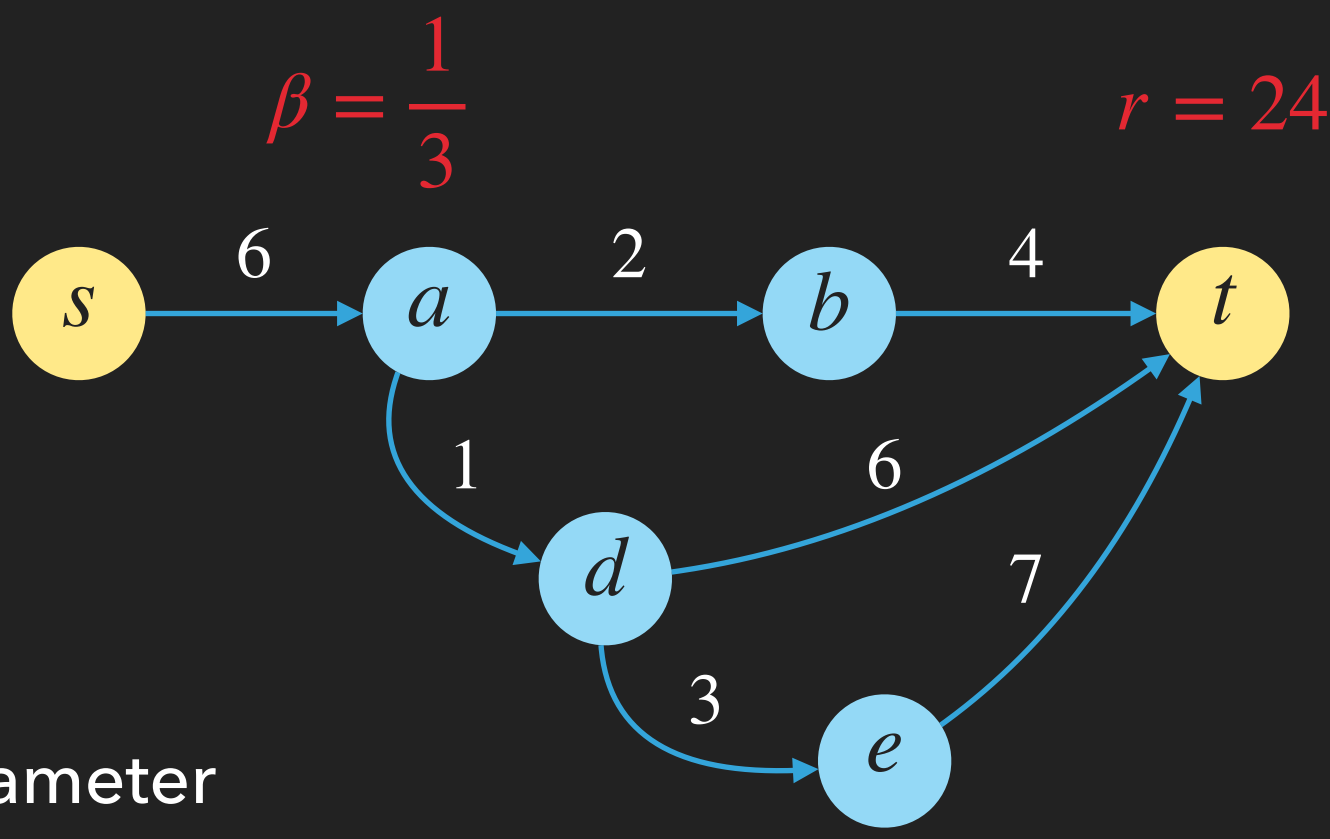


In vertex v agent evaluates a path $P \subseteq G$ with edges $e_1, e_2, \ldots, e_p$ to cost $\zeta_M(P) = w(e_1) + \beta \cdot \sum_{i=2}^{p} w(e_i)$.

MODEL WITH REWARD (KLEINBERG-OREN)

6-tuple $M = (G, w, s, t, \beta, r)$, where:

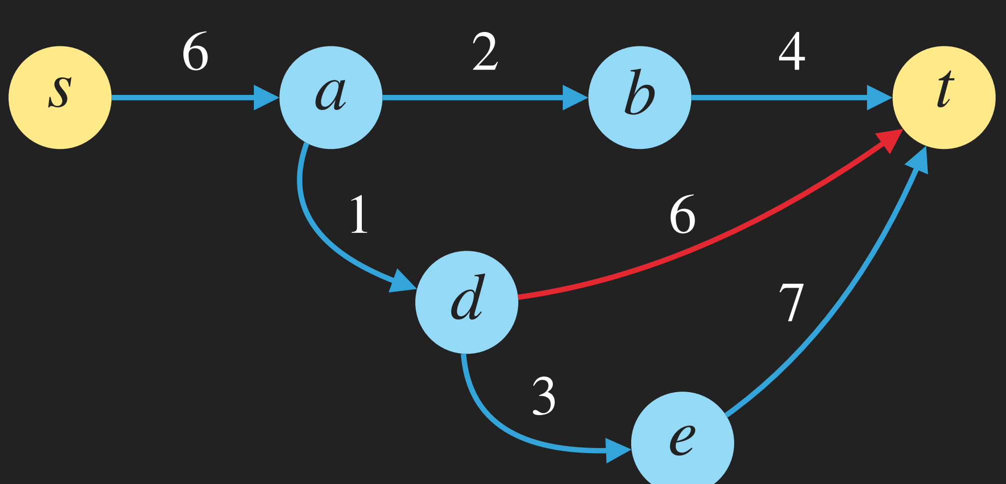
- $G = (V(G), E(G))$ – DAG
- $w : E(G) \to \mathbb{N}$ – cost-function
- $s \in V(G)$ – start vertex
- $t \in V(G)$ – target vertex
- $\beta \leq 1$ – agent's present-bias parameter
- r – reward the agent receives by reaching t



If for the agent in some vertex v perceived cost $\zeta_M(P)$ every path P exceeds a threshold $\beta \cdot r$, the agent will abandon the whole project.

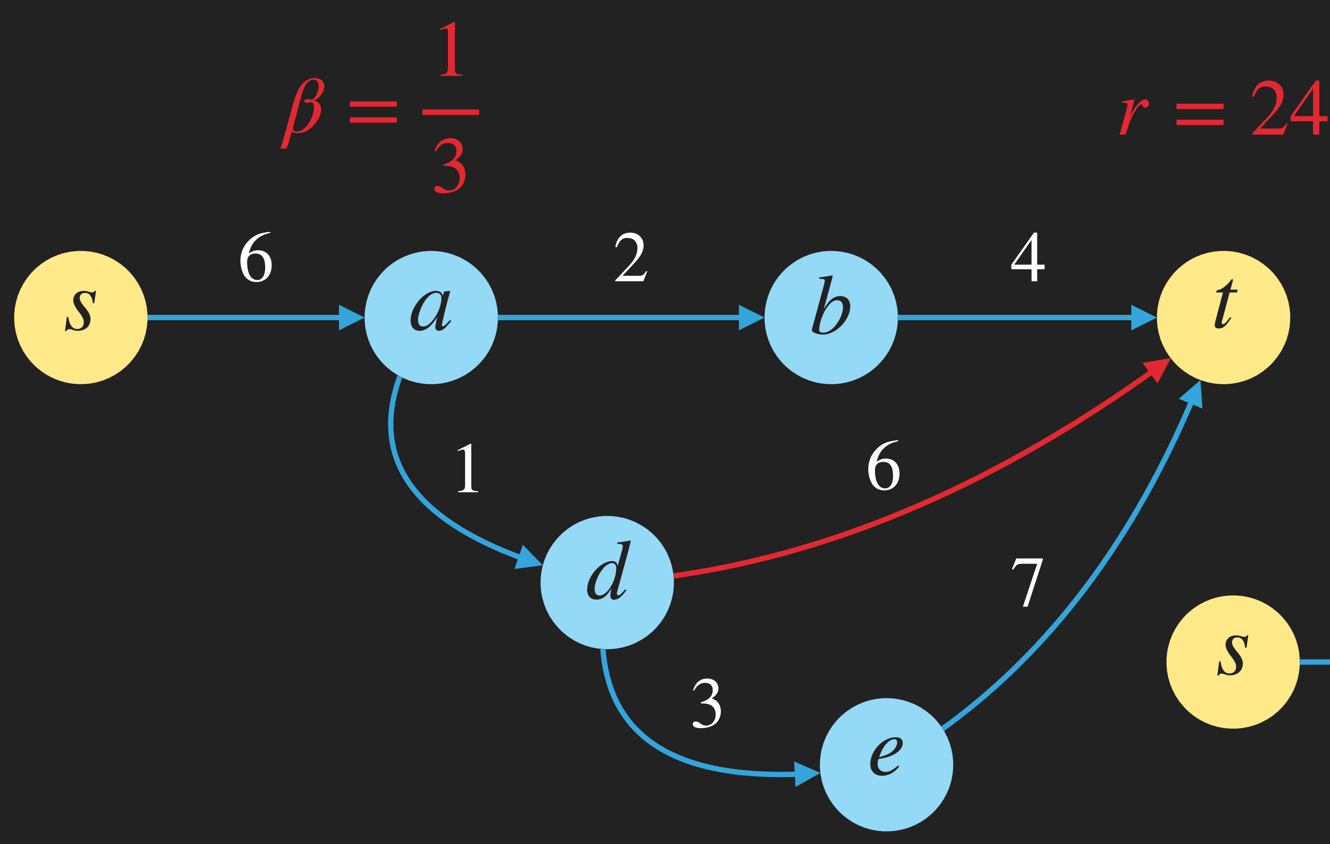
MOTIVATION

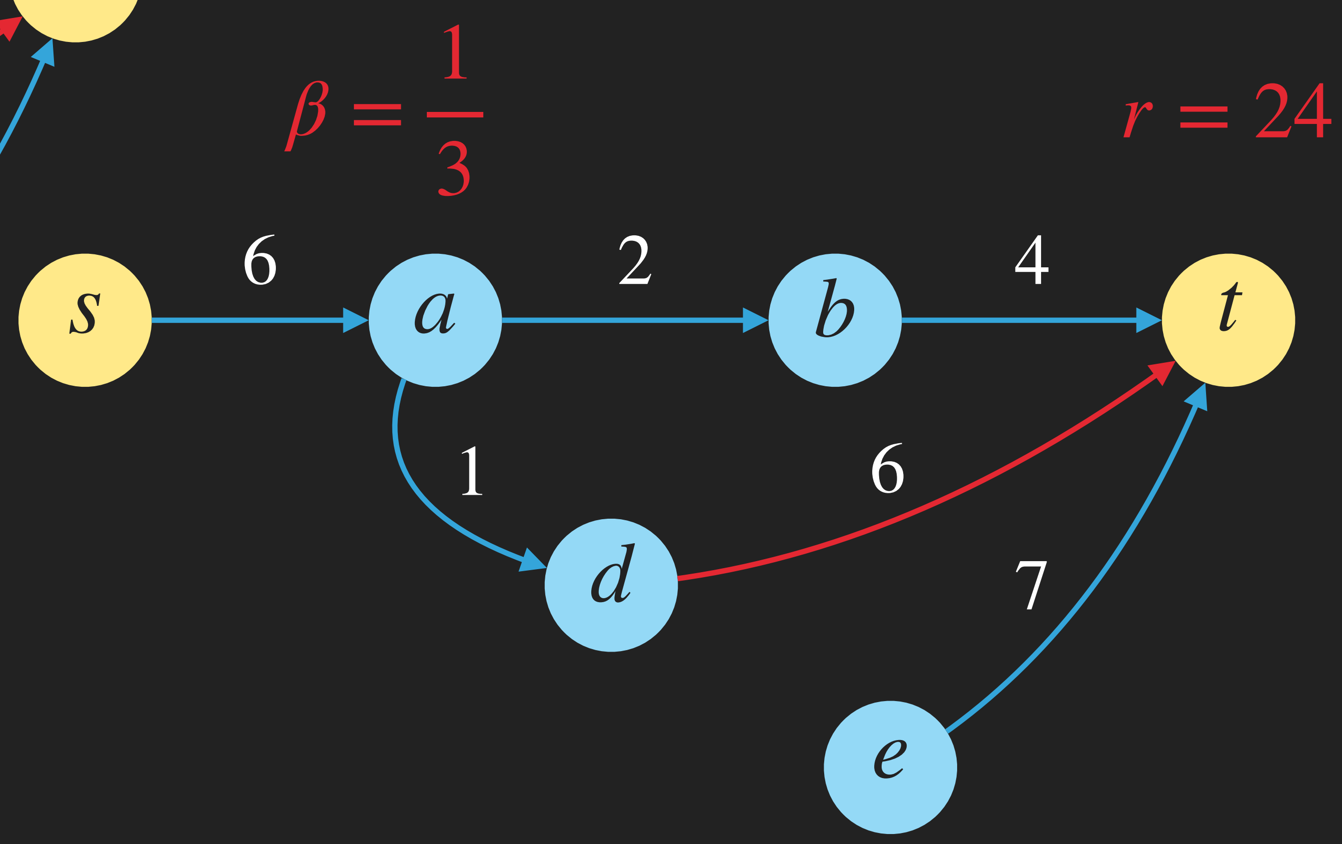
- Alice is PhD student in Santiago. 
- She has to accomplish several research projects to obtain her PhD.
- Bob is her advisor. 
- Bob wants her to finish her studies, he has additional interests too.
- The task corresponding to the arc dt is the most exciting part of the whole project.



EXAMPLE

- Bob can remove some tasks from Alice's roadmap.
- Bob decided to remove the arc de.





PROBLEMS

T-path-Deletion

Input: $M = (G, w, s, t, \beta, r)$, integer k and a set of arcs $T \subseteq E(G)$.

Task: Find a subset of arcs $D \subseteq E(G)$ of size at most k (or prove that no such set exists), such that after removing D from M, the agent will follow a T-path.

T-path-Addition

Input: $M = (G, w, s, t, \beta, r)$, integer k and a set of arcs $T \subseteq E(G)$, and a set of additional weighted arcs $A \subset V \times V$.

Task: Find a set S of at most k arcs from A (or prove that no such set exists), such that after adding these arcs to G the agent will follow a T-path.

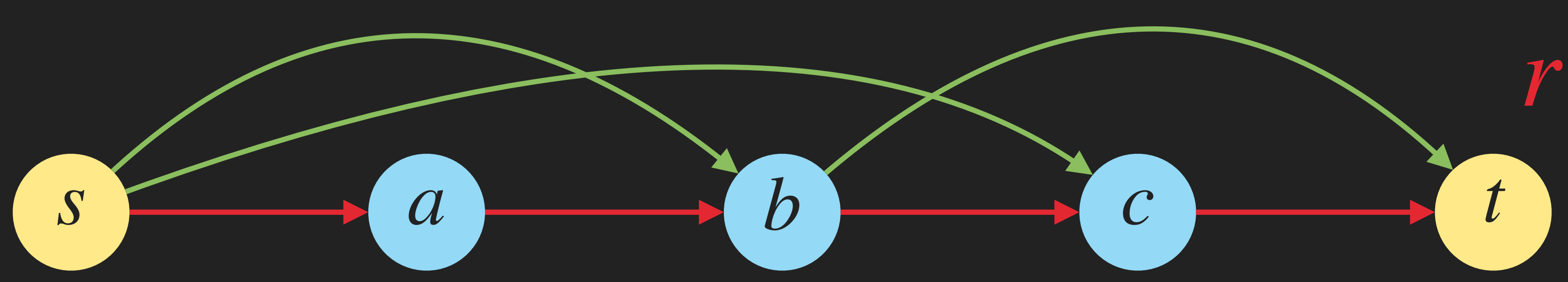
RELATED PROBLEMS

ECAI 50th ANNIVERSARY

- Finding a motivating subgraph (in our model T is empty).
- Tang, Teng, Wang and et al. show that this problem is NP-complete.
- Alberts and Kraft show approximation for reward.
- Fomin and Strømme studied the parameterized complexity of computing a motivating subgraph.
- Oren and Soker studied P-motivating subgraph problem.
- Is there a subgraph of G, such that in this subgraph, the agent will follow along path P?
- In our model, the prescribed arcs T form the edge set of P.

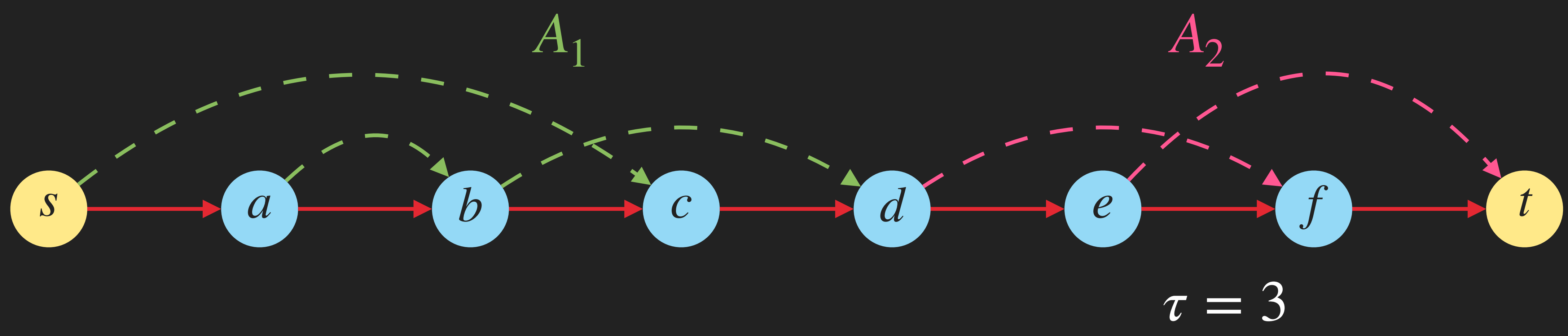
HARDNESS

- T-path-Deletion is W[1]-hard parameterized by k for any $\beta \leq 1$ even when T consists of a single arc.
- W[1]-hard parameterized by k for any $|T|$.
- T-path-Deletion is NP-hard with 2 different weights on the arcs, constant reward, unique T-path, cost of any T-paths no more than 6, any path from s to t contains at most 8 arcs.
- T-path-Addition problem on the path with detours instances is NP-hard and W[1]-hard parameterized by k.



ALGORITHMS

- T-path-Deletion problem is solvable in time $O(m^{2k}) \cdot poly(n)$.
- T-path-Deletion admits a polynomial kernel parameterized by the size of a feedback edge set.
- T-path-Addition problem on paths with detours can be solved in time $2^{\tau} n^{O(1)}$, where τ is the size of maximum intersection component of A.



OPEN PROBLEMS

- T-path problem with changing the weights of the arcs.
- What happens to the complexity of the problems when you can put intermediate rewards to the vertices.
- Algorithms relative to other parameters.
- Kernel is bounded by the size of vertex cover (even exponential) in graph G.

Contact

Full paper