Efficient Computing of Disaster-Disjoint Paths: Greedy and Beyond

Balázs Vass*, Erika Bérczi-Kovács†, Péter Gyimesi†, János Tapolcai*
*Budapest University of Technology and Economics, and †Eötvös Loránd University

Model and greedy approaches

Model assumptions:

- Input graph $G = (V, E)$ is planar
- Incident edges for each node are given in clockwise order (rotation system).
- List of disasters is encoded as a list of link sets $\mathcal{R} \subseteq 2^E$
- Each disaster causing the outage is a connected destruction area; consequently, the corresponding dual edges of each disaster $r \in \mathcal{R}$, form a connected subgraph in the dual graph G^* [6]

The aim is to find a maximum number of disaster-disjoint st-paths (Problem 1). This is is $\mathcal{NP}$-hard by [3].

Greedy approaches:

- ***Simple Greedy***: we are given an st-path P_1. Then, for each $i \in \{2, 3, \dots\}$, P_i is the nearest *clockwise disaster-disjoint* path to P_{i-1} (formally defined in [6]).

Problem 1: Maximum number of disaster-disjoint st-paths

Input: A planar graph $G = (V, E)$, rotation system, nodes $s, t \in V$, disasters/regions $\mathcal{R} \subset 2^E$

Output: A maximum number of disaster-disjoint st-paths $P_1, P_2 \dots, P_k$

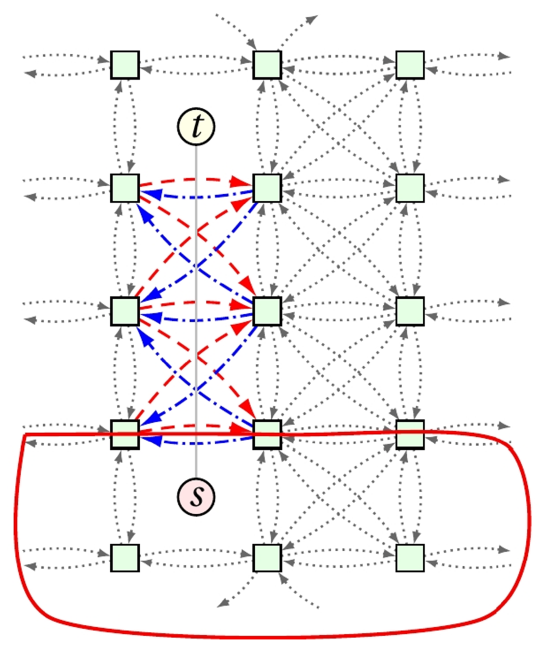
Theorem 1. *The Simple Greedy doesn't always find an optimal solution for Problem 1 even supposing node-disjointness.*

Fig. (f) is an example where the Simple Greedy does not find a disaster-disjoint path pair.

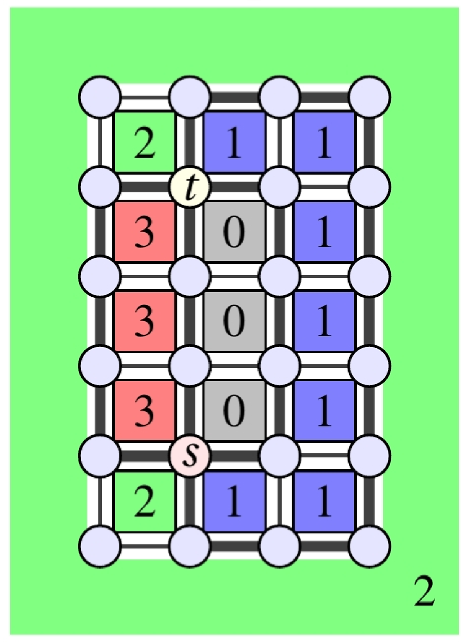
- ***Dervish*** [6] adds a second rule: searching for the k^{th} disaster-disjoint path, it starts with disaster-disjoint paths $P_1, \dots, P_{k-1}$, with $P_0 := P_{k-1}$. A new path P_l should be (non-strictly) clockwise to P_{l-k}. In presence of the node failures, the Dervish is guaranteed to solve Problem 1 with node-disjoint paths in polynomial time, in practice, squared in $|V|$. E.g., on (c), (d), when searching for the 3rd disaster-disjoint path, starting from P_1, P_2, Dervish generates paths: $P_3^D = \{s, v_3, v_4, t\}$, $P_4^D = \{s, v_6, t\}$, $P_5^D = \{s, v_2, t\}$, $P_6^D = \{s, v_4, t\}$. Note that P_4^D, P_5^D, and P_6^D are pairwise node- and disaster-disjoint.

Efficient algorithm

Paper [1] introduces the so-called *regional dual graph* $G^*_{\mathcal{R}}$ with related weighting c_k (see below). There are k non-crossing disaster-disjoint st-paths exactly if c_k is conservative. This can be decided and the paths can be computed in sub-squared worst-case, or w.h.p. in near-linear expected time.



(g) Regional dual $D^*_{\mathcal{R}}$. Cost c_k of black-and-dotted, red-and-dashed, and blue-and-dashdotted arcs is 1, $1-k$, and $1+k$, resp. For $c_{k\geq 5}$, the red closed arc encodes a negative cycle.



(h) Topology G, with the regions being exactly the nodes $v \in V \setminus \{s, t\}$. Numbers on the faces form a feasible potential for $c_{k=4}$. Boundaries between values form paths.

References

[1] E. Bérczi-Kovács *et al.*, "Efficient Algorithm for Region-Disjoint Survivable Routing in Backbone Networks," in *IEEE INFOCOM*, 2024.

[2] B. Vass *et al.*, "Efficient computing of disaster-disjoint paths: Greedy and beyond," in *IEEE INFOCOM WKSHPS*, 2024.

[3] D. Bienstock, "Some generalized max-flow min-cut problems in the plane," *Mathematics of Op. Res.*, 1991.

[4] Y. Kobayashi *et al.*, "Max-flow min-cut theorem and faster algorithms in a circular disk failure model," in *IEEE INFOCOM 2014*, April 2014.

[5] S. Neumayer *et al.*, "Assessing the vulnerability of the fiber infrastructure to disasters," *IEEE/ACM ToN.*, vol. 19, 2011.

[6] B. Vass *et al.*, "Polynomial-time algorithm for the regional SRLG-disjoint paths problem," in *Proc. IEEE INFOCOM*, May 2022.

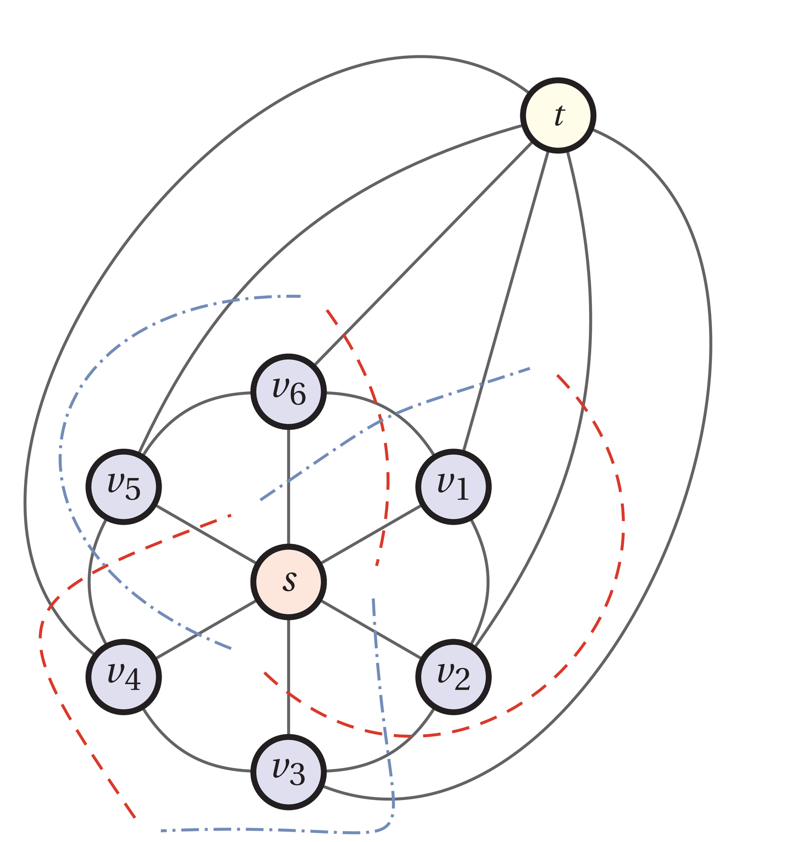
Contact Information

Web lendulet.tmit.bme.hu, egres.elte.hu

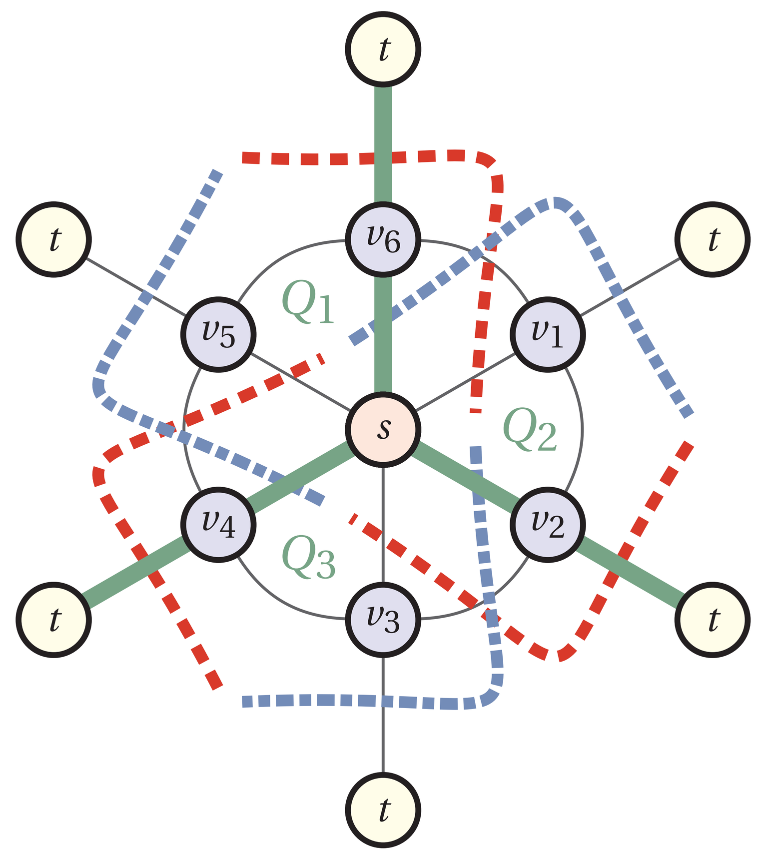
Email {vb,tapolcai}@tmit.bme.hu, erika.berczi-kovacs@ttk.elte.hu, peti1234@student.elte.hu

Examples

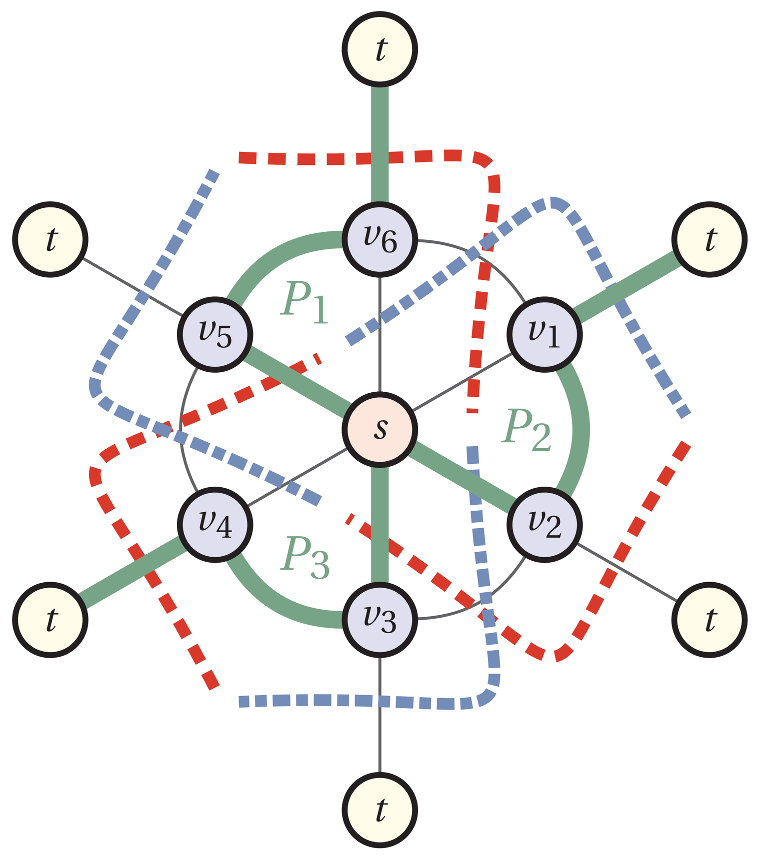
While subfigures a) and e) show the input graph, in the rest of the subfigures, for easier visualization, t is drawn in multiple copies. b) shows three disaster-disjoint paths, c) and d) combined show a cycle of paths the simple greedy generates, in which no 3 consecutive is disaster-disjoint. Paths traversing vertices s-v_1-t and s-v_2-t are *non-crossing*, while those traversing s-v_1-v_2-t and s-v_2-v_1-t are *crossing*. Finally, f) shows an example where there exist disaster-disjoint path pairs (e.g., $s - v_1 - t$ and $s - v_4 - t$), but, starting from an unfortunate path, the Simple Greedy does not find such a pair: the red-orange-brown-green is a cycle of non-disaster-disjoint paths; this means the Simple Greedy is not necessarily suitable for computing routing for 1+1 protection.



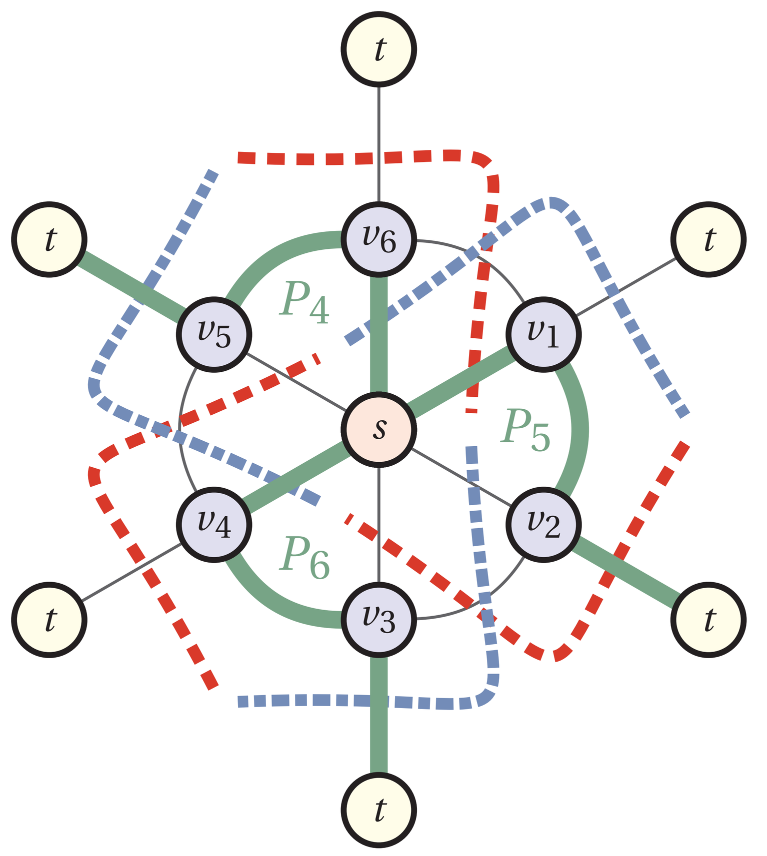
(a) Graph G and disasters



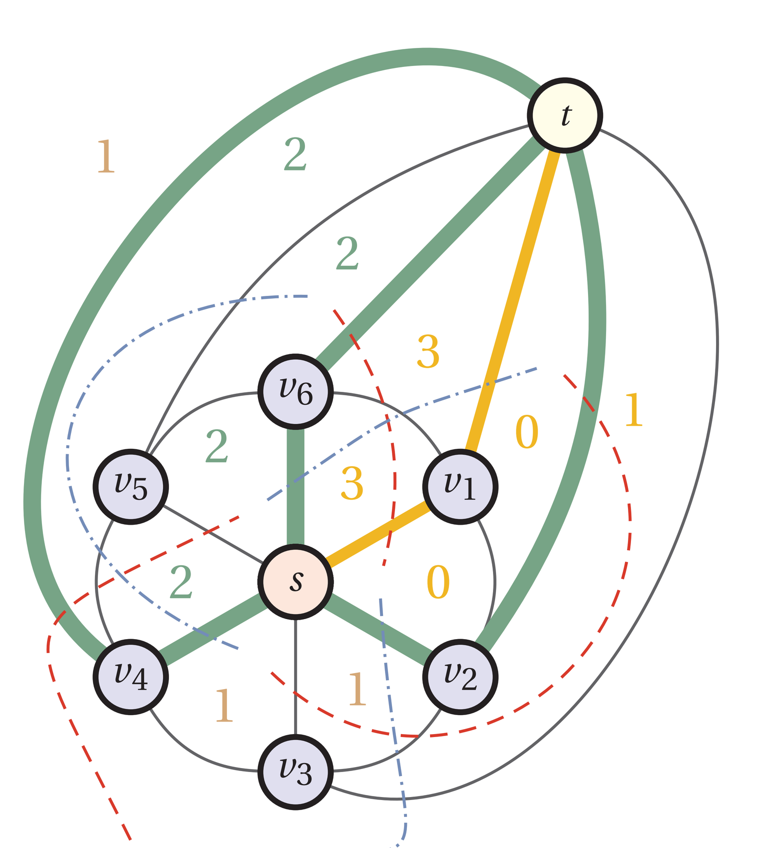
(b) Paths Q_1, Q_2, Q_3



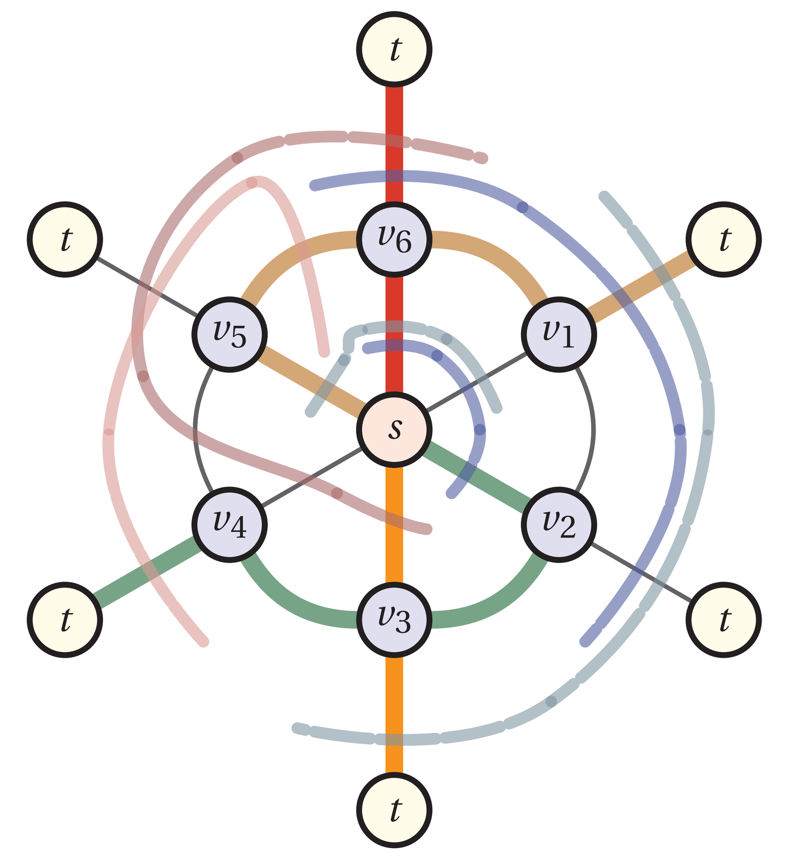
(c) Paths P_1, P_2, P_3



(d) Paths P_4, P_5, P_6



(e) Distances computed in [1]



(f) Simple Greedy bad cycle for 2 paths