Elementary Resource Constrained Shortest Path Problem (ERCSPP)

Matheus Diógenes Andrade
matheusdiogenesandrade@gmail.com
Fábio Luiz Usberti
fusberti@ic.unicamp.br
Rafael Kendy Arakaki
rafaelkendyarakaki@gmail.com

Institute of Computing - University of Campinas

2021

Topics

Presentation based on[1].

1. Problem definition;
2. Properties;
3. Algorithm;

[1] Dominique Feillet et al. "An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems". In: *Networks* 44.3 (2004), pp. 216–229. DOI: 10.1002/net.20033.

Problem definition

Problem definition

Let

- $D(V = \{s\} \cup V^+ \cup \{t\}, A)$ be the digraph, where
  - s is the source node; and
  - t is the target node.
- $c_a \in \mathbb{R}$ be the arc $a \in A$ cost;
- R be the set of resources;
- $d_a^r \in \mathbb{R}$ be the metric resource $r \in R$ consumption of the arc $a \in A$;
- $w_i^r = [b_i^r, e_i^r]$ be the resource $r \in R$ window of the node $i \in V$.

Problem definition

Let

- P be an elementary resource constrained s-t-path in D; and
- S_i^r be the resource $r \in R$ consumption of node $i \in V(P)$, such that $\forall (i,j) \in A(P) (\forall r \in R (S_j^r = \max\{S_i^r + d_{ij}^r, b_j^r\} \leqslant e_j^r))$.

The ERCSPP consists in finding the shortest elementary resource constrained s-t-path P in D.

Problem definition

$c_a : (d_a^1, d_a^2)$

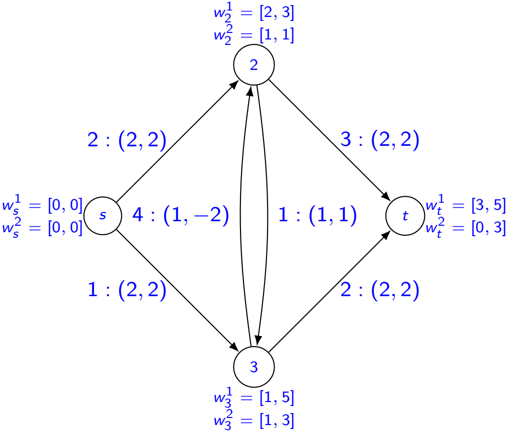

Figure: A ERCSPP instance digraph example.

Problem definition

$c_a : (d_a^1, d_a^2)$

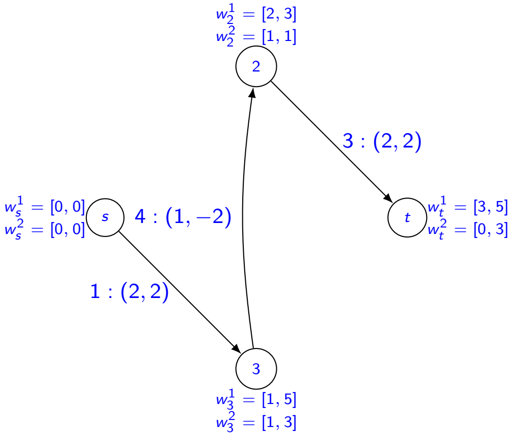

Figure: The solution of the previous instance.

Properties

Properties

Node state/label I

Let

- $i \in V$ be a node in D;
- P_i be an elementary resource constrained s-i-path in D; and
- $S_i = (L_i = (l_i^1, ..., l_i^{|R|}), C_i)$ be the state or label of node i in P_i, where:
  - $l_i^r \in \mathbb{R}$ is the amount of resource $r \in R$ consumed by P_i; and
  - $C_i \in \mathbb{R}$ is the cost of path P_i.

Properties

Node state/label I

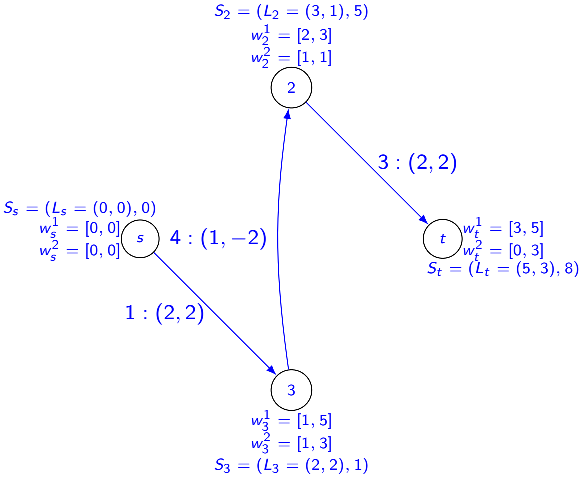

Figure: P_s, P_3, P_2, and P_t.

Properties

Dominance I

Let

- P_i and P_i' be two distinct elementary resource constrained s-i-paths in D; and
- S_i and S_i' be their respective labels.

Definition

P_i dominates $P_i' \leftrightarrow S_i \leqslant S_i'$.

Properties

Node state/label II

Let

- $L_i = (l_i^1, ..., l_i^{|R|}, s_i, v_i^1, ..., v_i^{|V|})$ be the the new resources state/label, where:
  - $s_i = \sum_{j=1}^{|V|} v_i^j = |V(P_i)|$ is the number of nodes visited by P_i; and
  - $v_i^j \in \mathbb{B}$ is equals to 1 if node $j \in V(P_i)$, i.e., j is visited by P_i, such that $j \in V$, and 0 otherwise.

Properties

Node state/label II

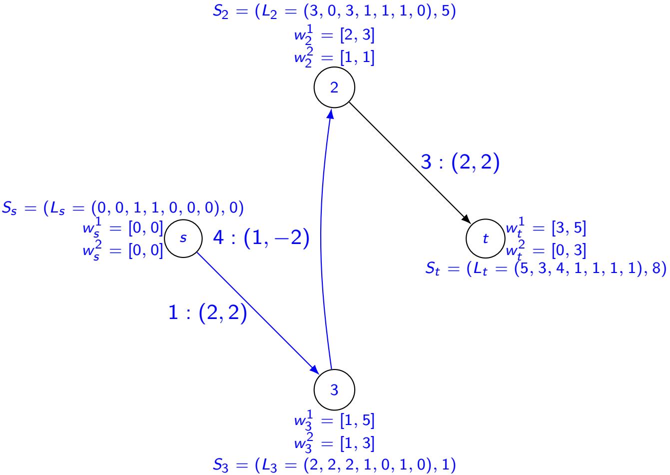

Figure: P_s, P_3, P_2, and P_t with new labels.

Properties

Dominance II

Let

- P_i and P_i' be two distinct elementary paths from s to $i \in V$ in D; and
- S_i and S_i' be the paths respective labels.

Definition

P_i dominates $P_i' \leftrightarrow S_i \leqslant S_i'$.

Properties

Unreachable nodes

Definition

$k \in V$ is said to be unreachable by $P_i \rightarrow k \in V(P_i) \vee$
$\exists r \in R(l_i^r + d_{ik}^r > e_k^r)$.

Note that, if $k \in V$ is unreachable by P_i, then $\nexists$ elementary resource constrainted i-j-path in D, due to the digraph metricity.

Properties

Node state/label III

Let

- $s_i = \sum_{j=1}^{|V|} v_i^j$ be the number of unreachable nodes by the path P_i at $i \in V(P_i)$; and
- $v_i^j \in \mathbb{B}$ be equals to 1 if node $j \in V$ is unreachable by the path P_i, and 0 otherwise.

Properties

Nondominated paths

Claim: In order to find an ERCSPP optimal solution, suffices to consider only nondominated paths.

Proof: Let's consider two elementary resource constrained s-i-paths P_i and $P_i^{'}$, along with its labels S_i and $S_i^{'}$, such that P_i dominates $P_i^{'}$. Also, let's consider an arc $(i, j) \in \delta^+(i) : \forall r \in R$ $(l_i'^r + d_{ij}^r \leqslant e_j^r) \wedge v_i'^j = 0$. Note that $\forall r \in R (l_i^r + d_{ij}^r \leqslant e_j^r) \wedge v_i^j = 0 \wedge C_i \leqslant C_i^{'}$. ■

Algorithm

Algorithm

Notations

Let

- Λ_i be the list of labels on node $i \in V$;
- N be the list of nodes to be processed;
- $$f(S_i) = \begin{cases} \text{true}, & \text{if } \forall r \in R(l_i^r \leqslant e_i^r) \land \forall j \in V(v_i^j \leqslant 1) \\ \text{false}, & \text{otherwise} \end{cases}$$

  be a function that says whether a label is feasible;
- $E(S_i, j) = (\max\{l_i^r + d_{ij}^r, b_j^r\} : r \in R) \cup (s_i + 1, v_i^1, ..., v_i^j + 1, ..., v_i^{|V|})$ be the function that returns the label resulting from the extension of a path P_i by a node j;
- F_{ij} be the set of labels extended from i to $j \in V$; and
- $N_D(\Lambda) = \{S_i \in \Lambda : \nexists S_i' \in \Lambda(S_i' \neq S_i \land S_i' \leqslant S_i)\}$ be the function that returns all nondominated labels from Λ.

Algorithm

Code

Algorithm 1 Labeling algorithm

Require: $D(V, A), c_a \quad \forall a \in A, d_a^r \quad \forall a \in A, r \in R$

$\Lambda_s \leftarrow \{(\{0\}^{|R|} \cup (1, 1) \cup \{0\}^{|V|-1}, 0)\}$;
$\Lambda_i \leftarrow \emptyset \quad \forall i \in V \setminus \{s\}$;
$N \leftarrow \{s\}$;
while $N \neq \emptyset$ **do**
    Let $i \in N$ be a N arbitrary node;
    for all $(i, j) \in \delta^+(i)$ **do**
        $F_{ij} \leftarrow \emptyset$;
        for all $S_i \in \Lambda_i$ **do**
            if $f(E(S_i, j))$ **then**
                $F_{ij} \leftarrow F_{ij} \cup \{E(S_i, j)\}$;
            end if
        end for
        $\Lambda_j \leftarrow N_D(F_{ij} \cup \Lambda_j)$;
        if Λ_j has changed **then**
            $N \leftarrow N \cup \{j\}$;
        end if
    end for
    $N \leftarrow N \setminus \{i\}$;
end while
return $\min_{(L_t, C_t) \in \Lambda_t} \{C_t\}$ if $\Lambda_t \neq \emptyset$ else ∞;

Algorithm
Example

State before line 3:

- $\Lambda_s = \{((0,0,1,1,0,0,0),0)\}$;
- $\Lambda_2 = \Lambda_3 = \Lambda_t = \emptyset$;
- $N = \{\{s\}\}$;

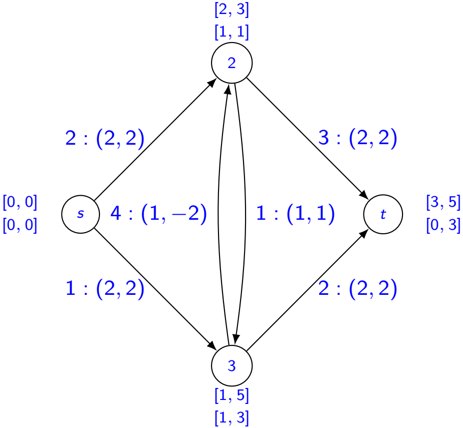

Figure: A ERCSPP instance digraph example.

Algorithm

Example

State at the end of 1st iteration of the while at line 3:

- $\Lambda_s = \{((0,0,1,1,0,0,0),0)\}$;
- $\Lambda_2 = \emptyset$;
- $\Lambda_3 = \{((2,2,2,1,0,1,0),1)\}$;
- $\Lambda_t = \emptyset$;
- $N = \{\{3\}\}$;

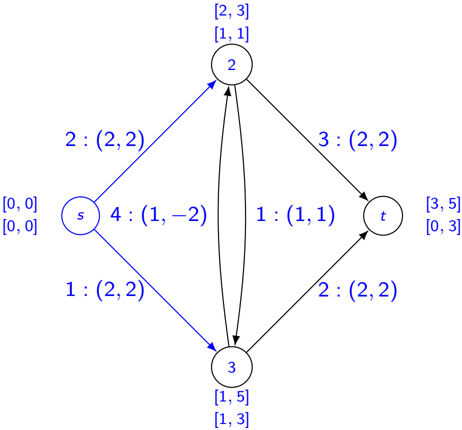

Figure: Arcs explored the for loop at line 5 when $i = s$.

Algorithm

Example

State at the end of 2rd iteration of the while at line 3:

- $\Lambda_s = \{((0,0,1,1,0,0,0),0)\}$;
- $\Lambda_2 = \{((3,0,3,1,1,1,0),5)\}$;
- $\Lambda_3 = \{((2,2,2,1,0,1,0),1)\}$;
- $\Lambda_t = \emptyset$;
- $N = \{\{2\}\}$;

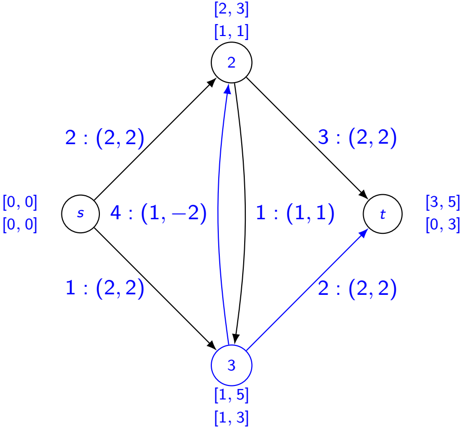

Figure: Arcs explored the for loop at line 5 when $i = 3$.

State at the end of 3nd iteration of the while at line 3:

- $\Lambda_s = \{((0,0,1,1,0,0,0),0)\}$;
- $\Lambda_2 = \{((3,0,3,1,1,1,0),5)\}$;
- $\Lambda_3 = \{((2,2,2,1,0,1,0),1)\}$;
- $\Lambda_t = \{((5,2,4,1,1,1,1),8)\}$;
- $N = \{\{t\}\}$;

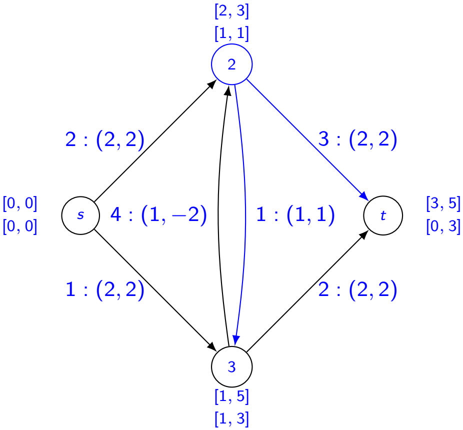

Figure: Arcs explored the for loop at line 5 when $i = 2$.

Algorithm

Example

State at the end of 4th iteration of the while at line 3:

- $\Lambda_s = \{((0,0,1,1,0,0,0),0)\}$;
- $\Lambda_2 = \{((3,0,3,1,1,1,0),5)\}$;
- $\Lambda_3 = \{((2,2,2,1,0,1,0),1)\}$;
- $\Lambda_t = \{((5,2,4,1,1,1,1),8)\}$;
- $N = \emptyset$;

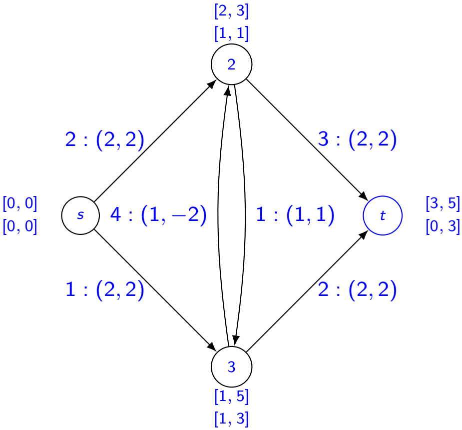

Figure: Arcs explored the for loop at line 5 when $i = t$.

Dominique Feillet et al. "An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems". In: *Networks* 44.3 (2004), pp. 216–229. DOI: 10.1002/net.20033.