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

Paper base[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 fo resources;
- $d_a^r \in \mathbb{R}$ be the arc $a \in A$' metric resource $r \in R$ consumption;
- $w_i^r = [b_i^r, e_i^r]$ be the node $i \in V$' resource $r \in R$ window.

The ERCSPP consists in finding the shortest s-t-path in D.

Problem definition

$c_a : (d_a^1, d_a^2)$
$w_s^1 = w_s^2 = [0, 0], w_2^1 = [2, 3], w_2^2 = [0, 1], w_3^1 = [1, 5], w_3^2 = [1, 3], w_t^1 = [3, 5], w_t^2 = [0, 2]$

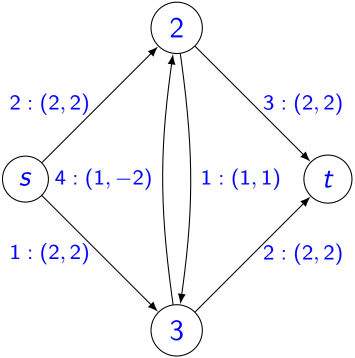

Figure: A ERCSPP instance digraph example.

Problem definition

$c_a : (d_a^1, d_a^2)$
$w_s^1 = w_s^2 = [0,0], w_2^1 = [2,3], w_2^2 = [0,1], w_3^1 = [1,5], w_3^2 = [1,3], w_t^1 = [3,5], w_t^2 = [0,2]$

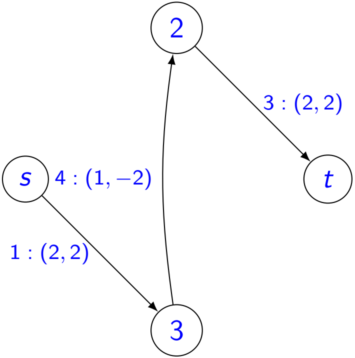

Figure: A ERCSPP instance digraph example.

Properties

Properties

Node state/label I

Let

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

Properties

Node state/label I

$$P_2, S_3 = (L_3 = (2,2), 1), S_2 = (L_2 = (3,0), 5)$$

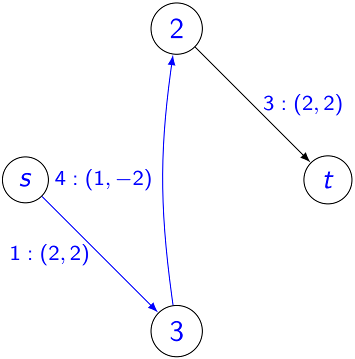

Figure: P_2.

Properties

Dominance I

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 respective labels.

Definition

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

Properties

Node state/label II

Let

- $L_i = (l_i^1, \ldots, l_i^{|R|}, s_i, v_i^1, \ldots, v_i^{|V|})$ be the the new resources state/label, where:
- the number of visited nodes, and the visitation vector by the path P_i.
  - $l_i^r \in \mathbb{R}_+$ is the amount of resource $r \in R$ consumed by the path P_i at $i \in V(P_i)$;
  - $s_i = \sum_{j=1}^{|V|} v_i^j$ is the number of nodes visited by the path P_i at $i \in V(P_i)$; and
  - $v_i^j \in \mathbb{B}$ is equals to 1 if node $j \in V$ is visited by the path P_i at $i \in V(P_i)$ and 0 otherwise.

Properties

Node state/label II

$P_2, S_s = (L_s = (0,0,1,1,0,0,0), 0), S_3 = (L_3 = (2,2,2,1,0,1,0), 1), S_2 = (L_2 = (3,0,3,1,1,1,0), 5)$

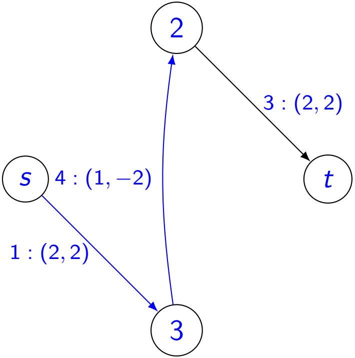

Figure: P_2.

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

Let

- P_i be an elementary path from s to $i \in V$ in D;
- S_i be the path label; and
- $k \in V$ be a digraph node.

Definition

k 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, does not exist any path from i that permits to reach it, due to the digraph metricity.

Properties

Node state/label III

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:
- the number of visited nodes, and the visitation vector by the path P_i.
  - $l_i^r \in \mathbb{R}_+$ is the amount of resource $r \in R$ consumed by the path P_i at $i \in V(P_i)$;
  - $s_i = \sum_{j=1}^{|V|} v_i^j$ is the number of unreachable nodes by the path P_i at $i \in V(P_i)$; and
  - $v_i^j \in \mathbb{B}$ is equals to 1 if node $j \in V$ is unreachable by the path P_i at $i \in V(P_i)$ and 0 otherwise.

Properties

Nondominated paths

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

Proof: Let's consider two elementary paths P_i and $P_i^{'}$ from s to $i \in V$, along with its labels S_i and $S_i^{'}$, such that P_i dominates $P_i^{'}$. Also, let's consider an arc $(i,j) \in A : \forall r \in R(l_i^{'r} + d_{ij}^r \leqslant e_j^r) \wedge v_i^{'j} = 0$. 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;
- $$f(S_i) = \begin{cases} \text{true}, & \text{if } \forall r \in R(l_i^r \leqslant e_i^r) \wedge \forall j \in V(v_i^j \leqslant 1) \\ \text{false}, & \text{otherwise} \end{cases}$$

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

Algorithm

Code

Algorithm 1 Labeling algorithm

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

$\Lambda_s \leftarrow \{(\{0\}^{|R|+1+|V|}, 0)\}$;
$\Lambda_i \leftarrow \emptyset \quad \forall i \in V \backslash \{s\}$;
$N \leftarrow \{s\}$;
while $N \neq \emptyset$ **do**
    Let $i \in N$ be a N arbitrary node;
    for all $j \in \delta^+(i)$ **do**
        $F_{ij} \leftarrow \emptyset$;
        for all $S_i \in \Lambda_i$ **do**
            if $v_i^j = 0 \wedge f(E(S_i, j))$ **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
    $N \leftarrow N \backslash \{i\}$;
end while

Algorithm

Example

$c_a : (d_a^1, d_a^2)$
$w_s^1 = w_s^2 = [0,0], w_2^1 = [2,3], w_2^2 = [0,1], w_3^1 = [1,5], w_3^2 = [1,3], w_t^1 = [3,5], w_t^2 = [0,2]$

State before line 3:

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

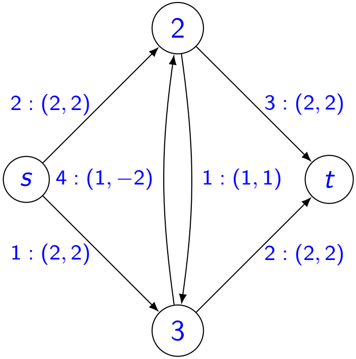

Figure: A ERCSPP instance digraph example.

Algorithm

Example

$c_a : (d_a^1, d_a^2)$
$w_s^1 = w_s^2 = [0,0], w_2^1 = [2,3], w_2^2 = [0,1], w_3^1 = [1,5], w_3^2 = [1,3], w_t^1 = [3,5], w_t^2 = [0,2]$

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

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

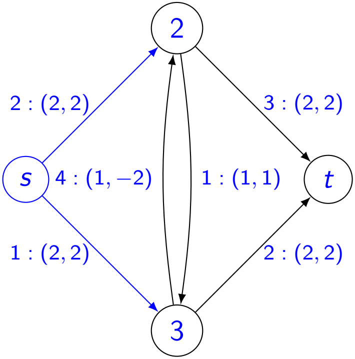

Figure: Arcs explored the for loop at line 5 when $i = s$.

Algorithm

Example

$c_a : (d_a^1, d_a^2)$
$w_s^1 = w_s^2 = [0, 0], w_2^1 = [2, 3], w_2^2 = [0, 1], w_3^1 = [1, 5], w_3^2 = [1, 3], w_t^1 = [3, 5], w_t^2 = [0, 2]$

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

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

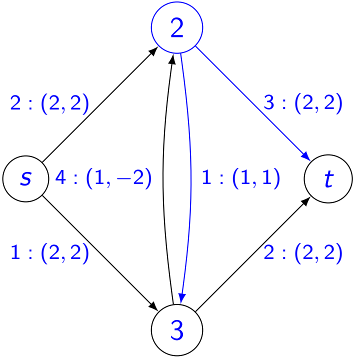

Figure: Arcs explored the for loop at line 5 when $i = 2$.

Algorithm

Example

$c_a : (d_a^1, d_a^2)$
$w_s^1 = w_s^2 = [0,0], w_2^1 = [2,3], w_2^2 = [0,1], w_3^1 = [1,5], w_3^2 = [1,3], w_t^1 = [3,5], w_t^2 = [0,2]$

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

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

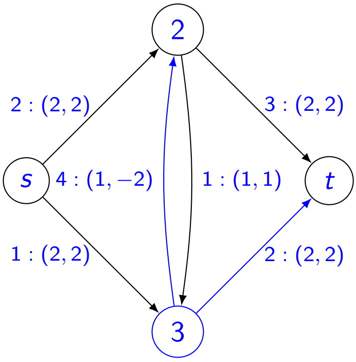

Figure: Arcs explored the for loop at line 5 when $i = 3$.

Algorithm

Example

$c_a : (d_a^1, d_a^2)$
$w_s^1 = w_s^2 = [0,0], w_2^1 = [2,3], w_2^2 = [0,1], w_3^1 = [1,5], w_3^2 = [1,3], w_t^1 = [3,5], w_t^2 = [0,2]$

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

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

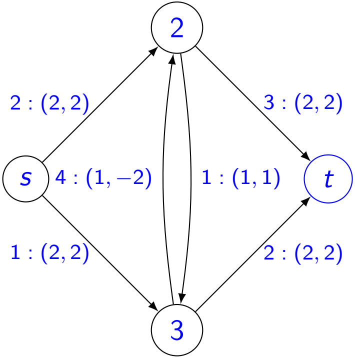

Figure: Arcs explored the for loop at line 5 when $i = t$.

Algorithm

Example

$c_a : (d_a^1, d_a^2)$
$w_s^1 = w_s^2 = [0,0], w_2^1 = [2,3], w_2^2 = [0,1], w_3^1 = [1,5], w_3^2 = [1,3], w_t^1 = [3,5], w_t^2 = [0,2]$

State at the end of 5th iteration of the while at line 3:

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

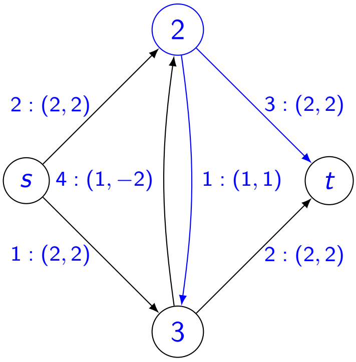

Figure: Arcs explored the for loop at line 5 when $i = 2$.

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.