Multiobjective Minimum Spanning Trees Using Dynamic Programming

Pedro Maristany de las Casas Antonio Sedeño Noda Ralf Borndörfer

EURO 2022, Espoo

July, 06 2022

maristany@zib.de

Multiobjective Minimum Spanning Tree (MO-MST) Problem

Input

Undirected graph $G = (V, E)$, edge cost vectors $c_e \in \mathbb{R}^d_{\geq}$, $d \in \mathbb{N}$.

Multiobjective Minimum Spanning Tree (MO-MST) Problem

Input

Undirected graph $G = (V, E)$, edge cost vectors $c_e \in \mathbb{R}^d_{\geq}$, $d \in \mathbb{N}$.

Dominance

Let $t,\ t'$ be (sub-) trees in G. t *dominates* t' ($c(t) \preceq c(t')$) if

Multiobjective Minimum Spanning Tree (MO-MST) Problem

Input

Undirected graph $G = (V, E)$, edge cost vectors $c_e \in \mathbb{R}^d_{\geq}$, $d \in \mathbb{N}$.

Dominance

Let $t,\ t'$ be (sub-) trees in G. t *dominates* t' ($c(t) \preceq c(t')$) if

- $V(t) = V(t')$

Multiobjective Minimum Spanning Tree (MO-MST) Problem

Input
Undirected graph $G = (V, E)$, edge cost vectors $c_e \in \mathbb{R}^d_{\geq}$, $d \in \mathbb{N}$.

Dominance

Let $t,\ t'$ be (sub-) trees in G. t *dominates* t' ($c(t) \preceq c(t')$) if

- $V(t) = V(t')$
- $c(t) \leq c(t')$ and $c(t) \neq c(t')$

Multiobjective Minimum Spanning Tree (MO-MST) Problem

Input

Undirected graph $G = (V, E)$, edge cost vectors $c_e \in \mathbb{R}^d_{\geq}$, $d \in \mathbb{N}$.

Dominance

Let $t,\ t'$ be (sub-) trees in G. t *dominates* t' $(c(t) \preceq c(t'))$ if

- $V(t) = V(t')$
- $c(t) \leq c(t')$ and $c(t) \neq c(t')$

Efficiency/Optimality

For $V' \subseteq V$, the set $\mathcal{T}^*(V')$ is a minimum complete set of efficient spanning trees.

Multiobjective Minimum Spanning Tree (MO-MST) Problem

Input

Undirected graph $G = (V, E)$, edge cost vectors $c_e \in \mathbb{R}^d_{\geq}$, $d \in \mathbb{N}$.

Dominance

Let $t,\ t'$ be (sub-) trees in G. t *dominates* t' $(c(t) \preceq c(t'))$ if

- $V(t) = V(t')$
- $c(t) \leq c(t')$ and $c(t) \neq c(t')$

Efficiency/Optimality

For $V' \subseteq V$, the set $\mathcal{T}^*(V')$ is a minimum complete set of efficient spanning trees.

Output

The MO-MST problem is to find $\mathcal{T}^*(V)$.

Bellman Condition for MO-MST

Lemma

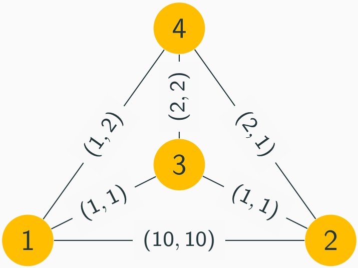
Consider a subset $V' \subseteq V$ *containing* k *nodes. Its efficient spanning trees are given by*

Bellman Condition for MO-MST

Lemma

Consider a subset $V' \subseteq V$ containing k nodes. Its efficient spanning trees are given by

$$\mathcal{T}^*(V') := \min_{\preceq_D} \left\{ t \circ \{u, v\} \mid t \in \mathcal{T}^*(V''),\ \forall V'' \subset V' \ ,\ |V''| = k - 1 \right.$$
$$\left. \text{and } u \in V'',\ v \in V' \setminus V'' \right\}.$$

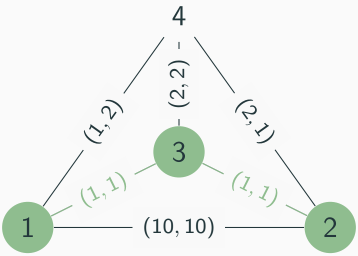


Bellman Condition for MO-MST

Lemma

Consider a subset $V' \subseteq V$ containing k nodes. Its efficient spanning trees are given by

$$\mathcal{T}^*(V') := \min_{\preceq_D} \left\{ t \circ \{u, v\} \mid t \in \mathcal{T}^*(V''),\ \forall V'' \subset V',\ |V''| = k-1 \right.$$
$$\left. \text{and } u \in V'',\ v \in V' \setminus V'' \right\}.$$

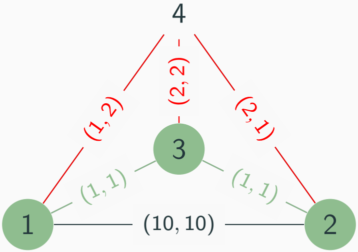


Bellman Condition for MO-MST

Lemma

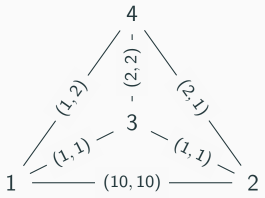
Consider a subset $V' \subseteq V$ containing k nodes. Its efficient spanning trees are given by

$$\mathcal{T}^*(V') := \min_{\preceq_D} \left\{ t \circ \{u, v\} \mid t \in \mathcal{T}^*(V''),\ \forall V'' \subset V',\ |V''| = k - 1 \text{ and } u \in V'',\ v \in V' \setminus V'' \right\}.$$



MO-MST $\rightarrow$ MO Shortest Path

MO-MST instance (G, c) into MOSP instance $(D_G, \bar{c}, \{1\})$

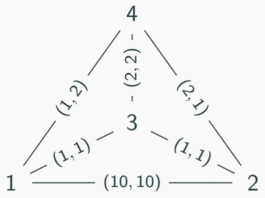


(a) Input (G, c).

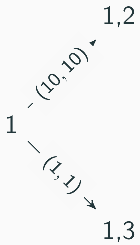
1

(b) Search Graph D_G with costs $\bar{c}$ induced by c.

MO-MST instance (G, c) into MOSP instance $(D_G, \bar{c}, \{1\})$

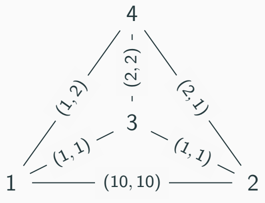


(a) Input (G, c).

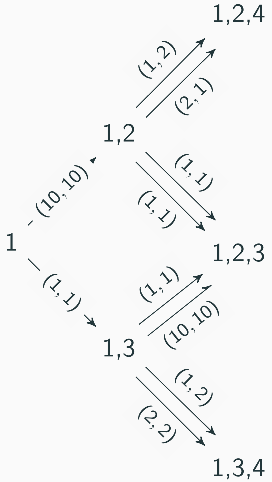


(b) Search Graph D_G with costs $\bar{c}$ induced by c.

MO-MST instance (G, c) into MOSP instance $(D_G, \bar{c}, \{1\})$

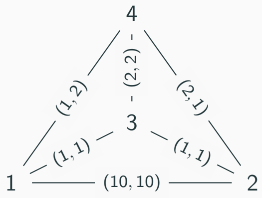


(a) Input (G, c).

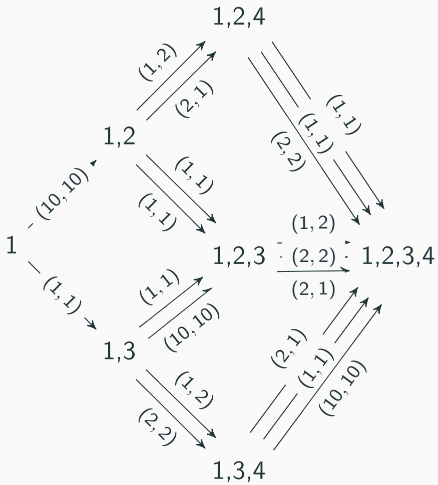


(b) Search Graph D_G with costs $\bar{c}$ induced by c.

MO-MST instance (G, c) into MOSP instance $(D_G, \bar{c}, \{1\})$



(a) Input (G, c).



(b) Search Graph D_G with costs $\bar{c}$ induced by c.

MO-MST instance $(G, c) \Leftrightarrow$ MOSP instance $(D_G, \bar{c}, \{1\})$

Lemma

The implicit search graph D_G of a MO-MST instance (G, c) is a directed, acyclic multigraph with $\mathcal{O}(2^{n-1})$ nodes, one for every subset $\{v_1\} \cup V'$, $V' \subseteq \{v_2, \ldots, v_n\}$.

MO-MST instance $(G, c) \Leftrightarrow$ MOSP instance $(D_G, \bar{c}, \{1\})$

Lemma

The implicit search graph D_G of a MO-MST instance (G, c) is a directed, acyclic multigraph with $\mathcal{O}(2^{n-1})$ nodes, one for every subset $\{v_1\} \cup V'$, $V' \subseteq \{v_2, \ldots, v_n\}$.

Theorem

Efficient $\{v_1\}$-V-paths in D_G uniquely correspond to efficient spanning trees of G.

MO-MST instance $(G, c) \Leftrightarrow$ MOSP instance $(D_G, \bar{c}, \{1\})$

Lemma

The implicit search graph D_G of a MO-MST instance (G, c) is a directed, acyclic multigraph with $\mathcal{O}(2^{n-1})$ nodes, one for every subset $\{v_1\} \cup V'$, $V' \subseteq \{v_2, \ldots, v_n\}$.

Theorem

Efficient $\{v_1\}$-V-paths in D_G uniquely correspond to efficient spanning trees of G.

Proof.

Follows from the Bellman Condition for MO-MST. $\square$

MO-MST $\rightarrow$ MO Shortest Path

Tailored MOSP algorithm for MO-MST instances

Designing an efficient MOSP algorithm for MO-MST

- Preprocessing: eliminate *red* edges and contract *blue* edge components.

Designing an efficient MOSP algorithm for MO-MST

- Preprocessing: eliminate *red* edges and contract *blue* edge components.
- Pruning by Chen et al., 2007: Remove parallel dominates edges.

Designing an efficient MOSP algorithm for MO-MST

- Preprocessing: eliminate *red* edges and contract *blue* edge components.
- Pruning by Chen et al., 2007: Remove parallel dominates edges.

- Early pruning of subtrees (not today)

Designing an efficient MOSP algorithm for MO-MST

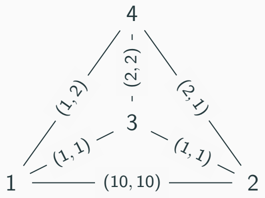
- Preprocessing: eliminate *red* edges and contract *blue* edge components.
- Pruning by Chen et al., 2007: Remove parallel dominates edges.

- Early pruning of subtrees (not today)
- Dimensionality reduction (e.g. Pulido et al., 2014)

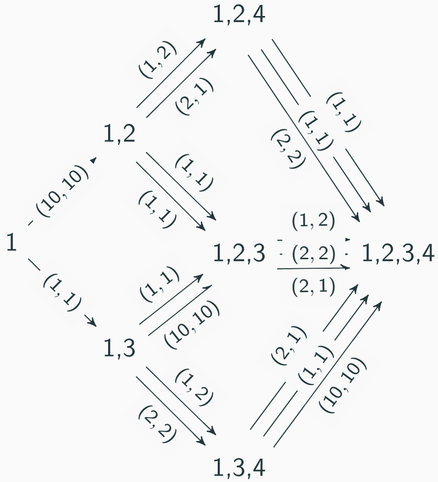
MO-MST $\rightarrow$ MO Shortest Path

Preprocessing Edges

Red/Blue Edge Coloring

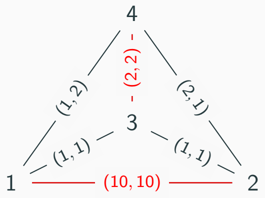


(a) Input (G, c).

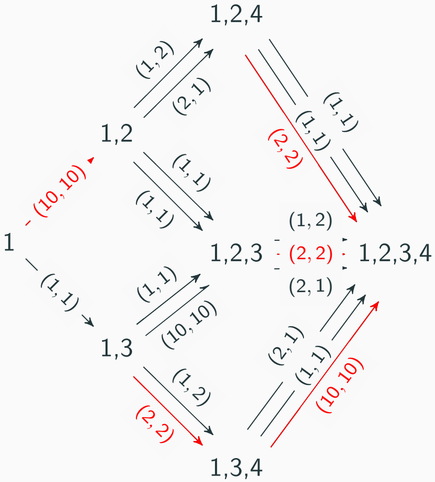


(b) Search Graph D_G with costs $\bar{c}$ induced by c.

Red/Blue Edge Coloring

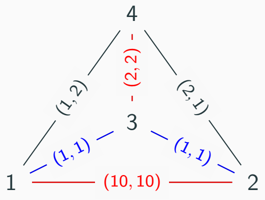


(a) Input (G, c).

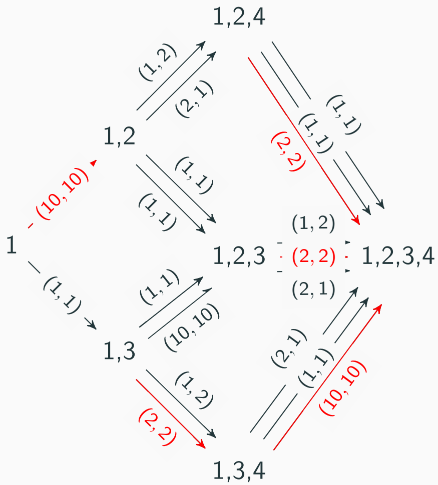


(b) Search Graph D_G with costs $\bar{c}$ induced by c.

Red/Blue Edge Coloring

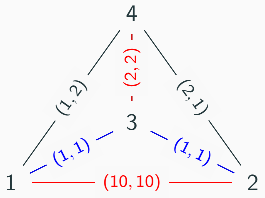


(a) Input (G, c).

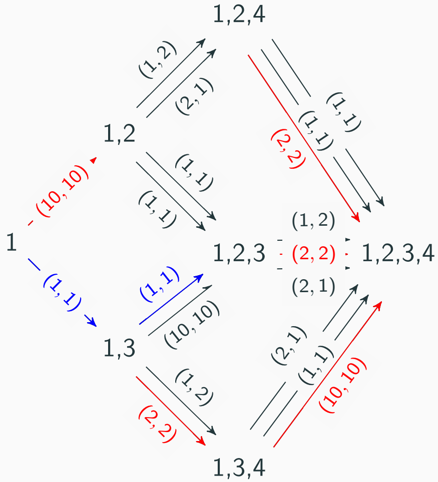


(b) Search Graph D_G with costs $\bar{c}$ induced by c.

Red/Blue Edge Coloring



(a) Input (G, c).



(b) Search Graph D_G with costs $\bar{c}$ induced by c.

Cut and Cycle Optimality Conditions

Red/Blue coloring inspired by...

- Tarjan, 1983. Data Structures and Network Algorithms.
- Corley, 1985. Hamacher and Ruhe, 1994. Ehrgott, 2005.
- Sourd and Spanjaard, 2008. Multiobjective B&B: Application to Bi-MST.

Cut and Cycle Optimality Conditions

Red/Blue coloring inspired by...

- Tarjan, 1983. Data Structures and Network Algorithms.
- Corley, 1985. Hamacher and Ruhe, 1994. Ehrgott, 2005.
- Sourd and Spanjaard, 2008. Multiobjective B&B: Application to Bi-MST.

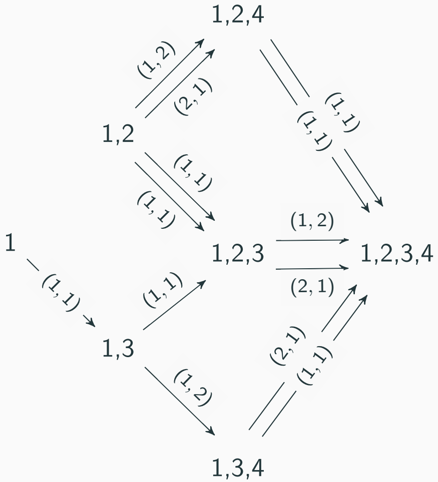
Preprocessing

Red and Blue edges can be determined running two DFS searches on G. Cycle and Cut Optimality conditions are checked.

MO-MST $\rightarrow$ MO Shortest Path

Remove dominated parallel arcs

Delete dominated parallel arcs – Chen et al., 2007

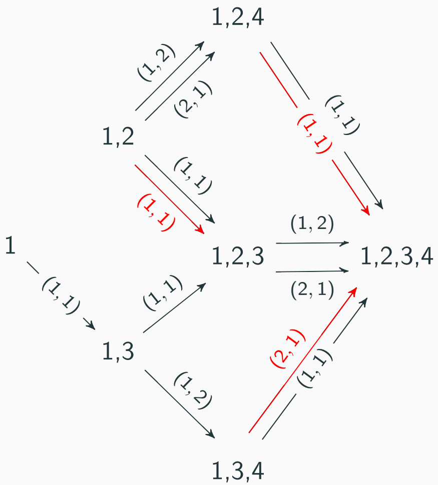


(a) Search Graph D_G with costs $\bar{c}$ induced by c.

Efficient Deletion

Use lex. sorting of outgoing arcs.

Delete dominated parallel arcs – Chen et al., 2007



(a) Search Graph D_G with costs $\bar{c}$ induced by c.

Efficient Deletion

Use lex. sorting of outgoing arcs.

What MOSP algorithm to choose?

Multiobjective Dijkstra Algorithm (MDA). M. et al., 2021

Consider the d-dimensional MOSP instance $(D_G = (\bar{V}, A),\ \bar{c},\ \{1\})$.

What MOSP algorithm to choose?

Multiobjective Dijkstra Algorithm (MDA). M. et al., 2021

Consider the d-dimensional MOSP instance $(D_G = (\bar{V}, A),\ \bar{c},\ \{1\})$.

- N – sum of all efficient s-v-paths in D for all $v \in \bar{V}$.
- $N_{\max}$ – max. number of efficient s-v-paths, $v \in \bar{V}$.

What MOSP algorithm to choose?

Multiobjective Dijkstra Algorithm (MDA). M. et al., 2021

Consider the d-dimensional MOSP instance $(D_G = (\bar{V}, A),\ \bar{c},\ \{1\})$.

- N – sum of all efficient s-v-paths in D for all $v \in \bar{V}$.
- $N_{\max}$ – max. number of efficient s-v-paths, $v \in \bar{V}$.

Then, the MDA runs in $\mathcal{O}\left(d\left(N \log(n) + N_{\max}^2 |A|\right)\right)$.

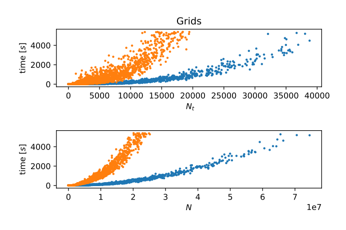
What MOSP algorithm to choose?

Multiobjective Dijkstra Algorithm (MDA). M. et al., 2021

Consider the d-dimensional MOSP instance $(D_G = (\bar{V}, A),\ \bar{c},\ \{1\})$.

- N – sum of all efficient s-v-paths in D for all $v \in \bar{V}$.
- $N_{\max}$ – max. number of efficient s-v-paths, $v \in \bar{V}$.

Then, the MDA runs in $\mathcal{O}\left(d\left(N\log(n) + N_{\max}^2|A|\right)\right)$.



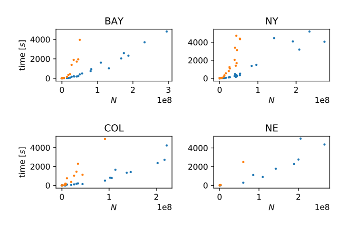


Figure 4: MDA vs. (Martins, 1984) algorithm on 3-d MOSP instances.

Experiments

Choice Justification

Fernandes et al., 2019. Empirical study of exact algorithms for the multi-objective spanning tree.

Setup

Choice Justification

Fernandes et al., 2019. Empirical study of exact algorithms for the multi-objective spanning tree.

	BN algo. original	Our BN algo.	New algo.
Coloring	No	Yes	Yes
Arc ordering	Head/Tail Ids	Head/Tail Ids	Lex.
Lower bound pruning	No	Yes	Yes
Dim. reduction	No	Yes	Yes
MOSP algo.	(Martins, 1984)	(Martins, 1984)	MDA

Choice Justification

Fernandes et al., 2019. Empirical study of exact algorithms for the multi-objective spanning tree.

	BN algo. original	Our BN algo.	New algo.
Coloring	No	Yes	Yes
Arc ordering	Head/Tail Ids	Head/Tail Ids	Lex.
Lower bound pruning	No	Yes	Yes
Dim. reduction	No	Yes	Yes
MOSP algo.	(Martins, 1984)	(Martins, 1984)	MDA

Table 1: Original BN vs. Our BN on Santos et al., 2018 instances with 14 nodes.

Instance dimension	BN algo. original	Our BN algo.
2	1.12s	0.09s
3	86.33s	44.34s
4	3656.91s	1823.48s

Experiments

Biobjective $(d = 2)$ MST

Solutions on complete graphs with 2 anticorrelated objectives

n	TIME NEW [s]	TIME BN [s]	SPEEDUP	$\mathcal{T}^*(V)$ cardinality
10	0.05	0.05	1	143
12	0.57	0.58	1	218
15	19.97	20.14	1	359
17	163.06	163.47	1	453
20	677.97	712.44	1.05	374
22	1877.95	1978.14	1.05	289
25	4192.325	-	-	398

Two phase approaches for BO-MST problems

Graph	n	m'	blue	LS	Sort	Shaving	BnB	Total
Clique	50	156.5	1.4	0.62	0.05	0.05	0.75	1.54
	100	339.8	2.8	4.34	0.56	0.23	3.17	8.56
	150	534.3	3.0	12.71	1.89	0.40	3.87	19.68
	200	727.0	3.5	28.22	4.10	0.45	4.10	38.64
	250	905.3	4.8	52.26	7.06	0.46	2.28	65.74
	300	1075.9	5.2	87.94	9.47	0.43	7.61	105.82
	350	1181.8	8.7	137.92	13.27	0.58	4.87	154.01
	400	1650.4	7.4	208.94	-	-	1.56	225.68
	450	1839.3	10.1	308.38	-	-	0.44	331.68
	500	1980.9	13.6	438.75	-	-	1.88	473.34

Figure 5: Extremely fast Biobjective algorithm by Sourd and Spanjaard.

References

- Sourd and Spanjaard, 2008
- Amorosi and Puerto, 2022

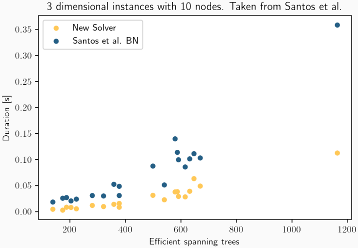
Experiments

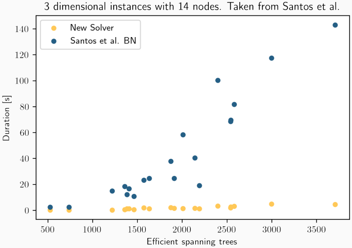
Random graphs from Santos et al., 2018

Random graphs from Santos et al., 2018

Table 2: 3-d instances with anticorrelated edge costs.

NODES	TIME NEW [s]	TIME BN [s]	SPEEDUP
10	0.03	0.08	3.44
11	0.06	0.22	3.56
12	0.18	1.63	7.45
13	0.49	7.69	14.35
14	1.76	44.34	25.33

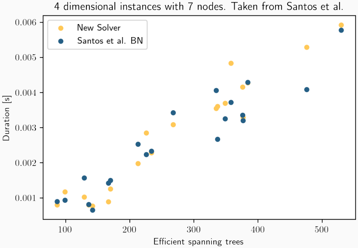


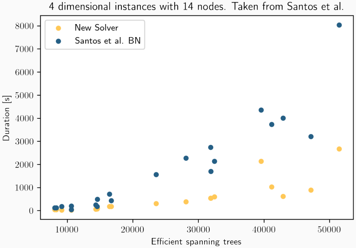


Random graphs from Santos et al., 2018

Table 3: 4-d instances with anticorrelated edge costs.

NODES	TIME NEW [s]	TIME BN [s]	SPEEDUP
10	1.35	1.48	1.22
11	4.90	7.39	1.50
12	35.00	82.87	2.35
13	102.79	311.90	2.86
14	523.48	1823.48	3.96





Complete graphs

Table 4: 4-d instances with anticorrelated edge costs.

NODES	TIME NEW [s]	TIME BN [s]	SPEEDUP
7	0.01	0.02	2
10	8.36	150.75	18.03
12	272.45	4044.89	14.85

Multiobjective Minimum Spanning Trees Using Dynamic Programming

Pedro Maristany de las Casas   Antonio Sedeño Noda   Ralf Borndörfer

EURO 2022, Espoo

July, 06 2022

maristany@zib.de

ZIB
ZUSE INSTITUTE BERLIN

Freie Universität Berlin

References

Amorosi, L., & Puerto, J. (2022). Two-phase strategies for the bi-objective minimum spanning tree problem. *International Transactions in Operational Research*, *29*(6), 3435–3463. https://doi.org/10.1111/itor.13120

Chen, G., Chen, S., Guo, W., & Chen, H. (2007). The multi-criteria minimum spanning tree problem based genetic algorithm. *Information Sciences*, *177*(22), 5050–5063. https://doi.org/https://doi.org/10.1016/j.ins.2007.06.005

- Corley, H. W. (1985). Efficient spanning trees. *Journal of Optimization Theory and Applications*, *45*(3), 481–485. https://doi.org/10.1007/bf00938448
- Ehrgott, M. (2005). *Multicriteria optimization*. Springer-Verlag. https://doi.org/10.1007/3-540-27659-9
- Fernandes, I. F. C., Goldbarg, E. F. G., Maia, S. M. D. M., & Goldbarg, M. C. (2019). Empirical study of exact algorithms for the multi-objective spanning tree. *Computational Optimization and Applications*, *75*(2), 561–605. https://doi.org/10.1007/s10589-019-00154-1
- Hamacher, H. W., & Ruhe, G. (1994). On spanning tree problems with multiple objectives. *Annals of Operations Research*, *52*(4), 209–230. https://doi.org/10.1007/bf02032304

M., P., Sedeño-Noda, A., & Borndörfer, R. (2021). An improved multiobjective shortest path algorithm. *Computers and Operations Research*, *135*, 105424. https://doi.org/https://doi.org/10.1016/j.cor.2021.105424

Martins, E. Q. V. (1984). On a multicriteria shortest path problem. *European Journal of Operational Research*, *16*(2), 236–245. https://www.sciencedirect.com/science/article/pii/0377221784900778

Pulido, F. J., Mandow, L., & de la Cruz, J. L. P. (2014). Multiobjective shortest path problems with lexicographic goal-based preferences. *European Journal of Operational Research*, *239*(1), 89–101. https://doi.org/10.1016/j.ejor.2014.05.008

Santos, J., Di Puglia Pugliese, L., & Guerriero, F. (2018). A new approach for the multiobjective minimum spanning tree. *Computers And Operations Research*, *98*, 69–83. https://doi.org/https://doi.org/10.1016/j.cor.2018.05.007

Sourd, F., & Spanjaard, O. (2008). A Multiobjective Branch-and-Bound Framework: Application to the Biobjective Spanning Tree Problem. *INFORMS Journal on Computing*, *20*(3), 472–484. https://doi.org/10.1287/ijoc.1070.0260

Tarjan, R. E. (1983). *Data structures and network algorithms*. Society for Industrial; Applied Mathematics.