Communications in Combinatorics and Optimization
Vol. ??, No. ?? (????) pp. 1-7
DOI: 10.22049/CCO.2022.27510.1280

CCO
Commun. Comb. Optim.

Short Note

Further study on "an extended shortest path problem: A data envelopment analysis approach"

Javad Tayyebi[1]*, Alireza Amirteimoori[2]

[1]Department of Industrial Engineering, Birjand University of Technology, Birjand, Iran
javadtayyebi@birjandut.ac.ir

[2]Department of Applied Mathematics, Islamic Azad University, Rasht 41335-3516, Iran
ateimoori@iaurasht.ac.ir

Received: 10 November 2021; Accepted: 21 January 2022
Published Online: 25 January 2022

Abstract: Amirteimoori proposed an approach based on data envelopment analysis (DEA) for multi-objective path problems on networks whose arcs contain multiple positive and negative attributes [A. Amirteimoori, An extended shortest path problem: A data envelopment analysis approach, Applied Mathematics Letters 25 (2012) 1839-1843]. The approach is to define a relative efficiency for each arcs using DEA models, and then to solve a longest path problem for obtaining a path with maximum efficiency. In this note, we focus on two drawbacks of the approach and illustrate them using examples. Then, we propose remedies to eliminate them.

Keywords: Data envelopment analysis, Shortest path, Multi attributes

AMS Subject classification: 90C27, 90C29, 90C08

1. Introduction

Shortest path problems lie at the heart of combinatorial optimization problems. A shortest path problem is to send something (e.g., a person, a vehicle, a computer data packet, and a telephone call) between two prescribed points in a network as quickly, as cheaply, or as reliably as possible. The problem has a wide range of applications. Moreover, it appears in many optimization problems as subproblem, such as minimum cost flow problems, maximum flow problems in planar networks, and etc. (see Chapter 5 of [1] for more details).

* *Corresponding Author*

In spite of the fact that shortest path problems is polynomially solvable, their multi-objective versions are NP-hard [7]. This fact together with many resulted applications of these problems cause that different approaches are developed to solve them. As instance, Data Envelopment Analysis (DEA) is used as a special tool to find an efficient path with respect to multiple positive and negative attributes of arcs [2, 3, 5, 6]. This concept is first introduced by Amirteimoori [2]. His approach is to calculate a relative efficiency by DEA models for any arc in comparable with arcs sharing a common endpoint with it, and then to find a longest path as a path with maximum efficiency. This note focuses on two drawbacks of the approach and presents remedies to resolve them.

The rest of this note is organized as follows. Section 2 states the approach presented in [2]. Section 3 presents a drawback which states that the approach requires to a preprocessing. Section 4 focuses on another drawback which can be modified by two remedies. The note ends with some concluding results in Section 5.

2. Approach statement

Suppose that a connected graph $G(\Lambda, \Psi)$ is given in which $\Lambda = \{1, 2, \ldots, m\}$ is the node set and Ψ is the arc set. Without loss of generality, let 1 be an origin and m be a destination. A sequence $v_1 - v_2 - \ldots - v_k$ is referred to as a walk if $(v_i, v_{i+1}) \in \Psi$ for $i = 1, 2, \ldots, k-1$. A such walk is said to be a (simple) path whenever any two nodes of the walk are not the same. In this note, we restrict ourselves only to simple paths connecting the origin node 1 to the destination node m. So, for abbreviation, we call them paths even without mentioning their start and end nodes.

Assume that each arc $(i, j) \in \Psi$ is associated to multiple negative attributes $X = (x_{ij}^{(1)}, x_{ij}^{(2)}, \ldots, x_{ij}^{(t)})$ as well as multiple positive attributes $Y = (y_{ij}^{(1)}, y_{ij}^{(2)}, \ldots, y_{ij}^{(s)})$. Negative attributes are criteria that we would like to minimize them, whereas positive ones have an opposite behaviour. For example, if we think arcs as routes joining two urban area, then traversal time, cost, traffic, and length of any arc are regarded as negative attributes while safety, and pleasant outlook are seemed to be positive attributes.

To find an efficient path, the main concept of the approach presented in [2] is that we can consider any arc as a decision making unit (DMU) having negative attributes as its inputs and positive attributes as its outputs. Using this concept, the approach first compares any DMU (i, j) with all DMUs whose tail node is i to obtain a relative efficiency $e_{ij}^{(1)}$ by solving the following CCR model.

$$\max \quad \sum_{r=1}^{s} y_{ij}^{(r)} u_r \tag{1a}$$

$$s.t. \quad \sum_{l=1}^{t} x_{ij}^{(l)} v_l = 1, \tag{1b}$$

$$\sum_{r=1}^{s} y_{ih}^{(r)} u_r - \sum_{l=1}^{t} x_{ih}^{(l)} v_l \leq 0 \quad h \in \Lambda_i, \tag{1c}$$

$$u_r, v_l \geq \epsilon \quad \forall r, l \tag{1d}$$

where $\epsilon > 0$ is a non-Archimedean infinitesimal quantity to enforce strict positivity of the weights, and $\Lambda_i = \{h \in \Lambda : (i,h) \in \Psi\}$.
In a similar manner, it computes a relative efficiency $e_{ij}^{(2)}$ by comparing DMU (i,j) with all DMUs whose head node is j. $e_{ij}^{(1)}$ and $e_{ij}^{(2)}$ are the quantities that evaluate the efficiency of (i,j). So, the problem is converted into a biobjective optimization. To obtain an efficient solution, a customary approach is to use the weighting method. Since there is no any justification which quantity is preferred to the other, the arithmetic mean can be used to aggregate two efficiency values, that is, $e_{ij} = \frac{e_{ij}^{(1)}+e_{ij}^{(2)}}{2}$ for every arc (i,j). Finally, it uses the following linear programming model to introduce a longest path problem for determining a path with maximum efficiency.

$$\max \quad z = \sum_{(i,j)\in\Psi} e_{ij}x_{ij} \tag{2a}$$

$$s.t. \quad \sum_{j:(i,j)\in\Psi} x_{ij} - \sum_{j:(j,i)\in\Psi} x_{ji} = \begin{cases} 1 & i=1 \\ 0 & i \neq 1, m \\ -1 & i = m \end{cases} \quad \forall i \in \Lambda, \tag{2b}$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in \Psi. \tag{2c}$$

3. First Drawback

The concept of introducing the arc efficiency using DEA models is interesting and considerable because an arc (i,j) is compared with all arcs competing with it at nodes i and j for selecting an efficient path. So, at a glance, computing the relative efficiency of (i,j) seems to be reasonable in this fashion. However there is a vital problem for a special situation. The situation occurs whenever there are some arcs not belonging to any path from the origin to the destination. Although these arcs do not appear in any path, they may have a significant effect in computing the relative efficiency of other arcs. The following example clarifies this situation.

Example 1. Consider the instance of the efficient path problem shown in Figure 1.a. Its arcs have a negative and a positive attributes. Notice that the arcs of path $P_1 : 1-4-5-8$ is dominated by the arcs of path $P_2 : 1-2-3-8$. So P_1 cannot be an efficient path. However, based on arc efficiencies computed in Figure 1.b, this path is selected by the approach. The main problem is that arcs $(2,6)$ and $(7,3)$ decrease the relative efficiency of $(2,3)$ and on the other hand, none of them belongs to a path from the origin to the destination. Hence, such arcs play a dummy role.

According to this argument, we propose that the dummy arcs are removed from the graph. For this purpose, one can perform a simple preprocessing to find them. The preprocessing uses a traversal algorithm, like Depth-First-Search or Breadth-First-Search, to find the set A^+ of arcs being accessible from the origin. Similarly, it applies a backward traversal algorithm to find the set A^- of arcs including a path

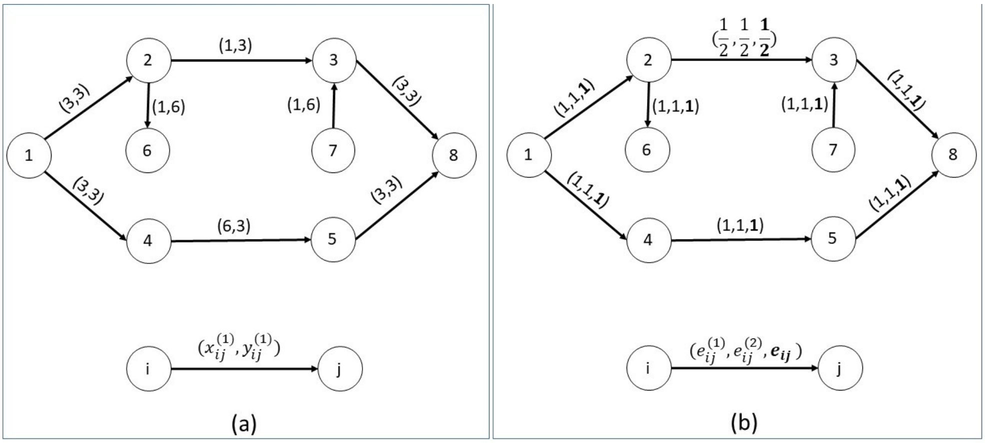

Figure 1. **An example illustrating the first drawback**

toward the destination. Obviously all arcs not belonging to $A^- \cap A^+$ are dummy and have to be removed from the graph.

4. Second Drawback

In [2], we recall that the author proposes either to solve model (2) or to use shortest path algorithms, such as Dijkstra, for finding a path with maximum efficiency. In this regard, he states that *"The foregoing problem* (model (2)) *is a variety of the classical shortest path problem with nonnegative costs and it can be solved by using the modified Dijkstra's Algorithm or labeling Algorithm".* Even though the use of longest path concept is admissible to find a path with maximum efficiency, we have to state that model (2) and shortest path algorithms cannot be applied to obtain it. Let us explain it in the further details.
Although it seems that shortest and longest path problems have a close relationship, they have completely different behaviours. Shortest path problems can be solved in polynomial time by several algorithms such as Dijkstra, and Bellman-Ford, while longest path problems are NP-hard [4], and consequently, they cannot admit any polynomial-time algorithm, assuming $P \neq NP$ [1]. So this claim of the author that "*it can be solved by using the modified Dijkstra's Algorithm or labeling Algorithm*" is not correct. As another significant point, model (2) with the "minimization" objective function is a well-known formulation of shortest path problems even if the binary constraint $x_{ij} \in \{0, 1\}$ is relaxed to $x_{ij} \geq 0$ for every $(i, i) \in \Psi$ [1]. However this formulation cannot be applied to find a longest path from 1 to m because it is possible that the optimal solution is a walk, instead of a path. Let us state a simple example in this regard.

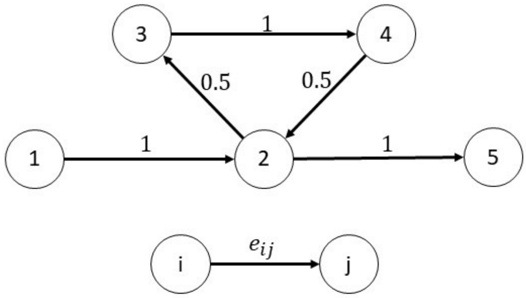

Figure 2. **An example illustrating the second drawback**

Example 2. Consider the instance of a longest path problem shown in Figure 2 in which 1 and 5 are respectively the origin and the destination. It is obvious that the longest path is the only path $1-2-5$. On the other hand, model (2) cannot find it. Indeed, model (2) has the unbounded optimal value $z^* = +\infty$ because the solution corresponding to the walk $1-2-3-4-2-5$ is feasible and its only cycle can be traversed repeatedly to obtain an objective value as large as possible.

So finding a path with maximum efficiency cannot be performed neither by solving model (2) nor by using Dijkstra and labeling algorithm.

Remark 1. Since any walk is a path in acyclic graphs, the results used in [2] for finding a longest path remains valid for acyclic graphs.

Example 2 shows that model (2) can guarantee that its optimal solution is a walk, but not a path. However if the following constraints are added to model (2), then this drawback is solved.

$$\sum_{j:(i,j)\in\Psi} x_{ij} \leq 1 \quad \forall i \in \Lambda. \tag{3}$$

These constraints restrict any node to contain at most one leaving arc in any feasible solution. So the optimal walk is essentially a path. A similar way is to add the following constraints, instead of (3).

$$\sum_{j:(j,i)\in\Psi} x_{ij} \leq 1 \quad \forall i \in \Lambda. \tag{4}$$

Despite the fact that model (2) with one of constraints (3) and (4) solves longest path problem, obtaining a maximum efficiency path is not possible for large scale graphs because longest path problems are NP-hard in general. Here, we would like to propose

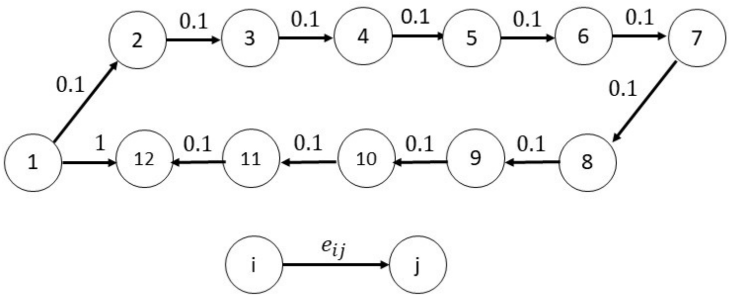

Figure 3. **An example comparing additive form and productive form**

another method for finding a maximum efficiency path. The method is to find a path which the production of its arc efficiencies is maximum among other paths, i.e.,

$$\max_{P\in\mathcal{P}} \prod_{(i,j)\in P} e_{ij}. \tag{5}$$

in which $\mathcal{P}$ is the set of all paths. Based on efficiency notion, we believe that the productive form of model (5) is better than the additive form of model (2) to find a maximum efficiency path. As an example, consider the graph shown in Figure 3. Paths $1-2-3-\ldots-12$ and $1-2$ are the only paths from 1 to 12. It is obvious that the $1-2-3-\ldots-12$ is an inefficient path because it is longer than $1-2$ and moreover, the efficiency value of its arcs are very low. However, the additive approach finds $1-2-3-\ldots-12$ because the additive term of $1-2-3-\ldots-12$ is greater than that of $1-2$, but the multiplicative approach finds $1-2$, correctly.

Remark 2. Due to the productive form of model (5), one can use the geometry mean, instead of the arithmetic mean, to compute arc efficiency as $e_{ij} = \sqrt{e_{ij}^{(1)} e_{ij}^{(2)}}$.

Since $e_{ij} \leq 1$, it follows that $\ln e_{ij} \leq 0$. Consequently, we can convert model(5) to the following shortest path problem.

$$\min_{P\in\mathcal{P}} \sum_{(i,j)\in P} \ln e_{ij}. \tag{6}$$

This method has this major advantage that polynomial-time shortest path algorithms can be used now.

5. Concluding remarks

Amirteimoori [2] presented a DEA-based approach to solve efficient path problems. In this note, we showed that there are two drawbacks in his proposed approach. The first drawback is the important efficient of some dummy arcs in computing the efficiency of other arcs which is solved by pruning them. The second drawback is about this fact that his proposed model cannot find a maximum efficiency path, correctly. This problem is solved by modifying his proposed model and furthermore, by proposing a novel approach.

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, PHI Englewood Cliffs NJ, 1993.

[2] A. Amirteimoori, *An extended shortest path problem: A data envelopment analysis approach*, Applied Math. Lett. **25** (2012), no. 11, 1839–1843.

[3] F. Ghandi, *Solving multi-objective shortest path problem using data envelopment analysis*, Journal of Operational Research and its Applications (Applied Mathematics) - Lahijan Azad University **18** (2021), no. 1, 21–38.

[4] F. Keshavarz-Kohjerdi, A. Bagheri, and A. Asgharian-Sardroud, *A linear-time algorithm for the longest path problem in rectangular grid graphs*, Discrete Appl. Math. **160** (2012), no. 3, 210–217.

[5] S. Kordrostami and M.J.S. Noveiri, *Solving a fuzzy shortest path problem with multiple inputs and outputs by using data envelopment analysis*, 2013 13th Iranian Conference on Fuzzy Systems (IFSC), IEEE, 2013, pp. 1–4.

[6] G.H. Shirdel and S. Ramezani-Tarkhorani, *A DEA-based approach for finding a favorable multi-objective shortest path*, Croat. Oper. Res. Rev. **9** (2018), no. 2, 149.

[7] Y. Wang, X. Li, and R. Ruiz, *A fast algorithm for finding the bi-objective shortest path in complicated networks*, 2018 IEEE 22nd International Conference on Computer Supported Cooperative Work in Design ((CSCWD)), IEEE, 2018, pp. 104–109.