C. D'Ambrosio, M. Gentili, R. Cerulli,
The optimal value range problem for the Interval (immune) Transportation Problem,
Omega, Volume 95, 2020, 102059, ISSN 0305-0483,
https://doi.org/10.1016/j.omega.2019.04.002.
(https://www.sciencedirect.com/science/article/pii/S0305048318307448)

The Optimal Value Range Problem for the Interval (Immune) Transportation Problem

C. D'Ambrosio[*], M. Gentili[†], R.Cerulli [‡]

April 1, 2019

Abstract

We address the problem of finding the range of the optimal cost of a transportation problem when supply and demand vary over an interval. We consider the specific version of a transportation problem with supply inequality constraints and demand equality constraints under the assumption that the transportation costs are immune against the transportation paradox. We investigate some theoretical properties of the problem which constitute the basis of a novel solution algorithm. Our results show that the proposed algorithm hugely outperforms the best existing solution approach.

1 Introduction

The Transportation Problem (TP) is a well-known optimization problem which has been studied for decades [1, 2]. TP belongs to the general class of network flow problems [3] and consists of finding the minimum cost transportation plan to ship goods from a set of suppliers to a set of customers while satisfying all customer demands and not exceeding the available capacity at the suppliers. Several contributions in the literature have addressed TP and its variants by assuming supply and demand parameters to be known with certainty [4, 5]. However, there are cases where these parameters may not be known precisely (i.e., level of demand and supply, transportation costs, as well as the underlying network structure, could change over time). Under this assumption, the total transportation cost will also vary over a range. Our focus is on determining the best and worst values of the optimal transportation plan among all the realizations of the uncertain parameters. Under this circumstances it would be useful to determine the lower and the upper bounds of the optimal total costs [6, 7]. Indeed, such an analysis would provide information that give the manager a sense of the risk involved in his/her decision both from an operational and from a strategical perspective. On the

[*]Department of Mathematics, University of Salerno, Italy, cdambrosio@unisa.it
[†]Industrial Engineering Department, University of Louisville, USA, monica.gentili@louisville.edu
Department of Mathematics, University of Salerno, Italy, mgentili@unisa.it
[‡]Department of Mathematics, University of Salerno, Italy, raffaele@unisa.it

one hand, the range of the objective function between the best and the worst optimum values, together with an inspection of corresponding coefficient settings, can provide insights into the likelihood of these extremes. Of course, specific values of the coefficients must be selected in order to provide results for operational use, and the two extreme scenarios can also provide guidance here as well. For example, the specific values of the uncertain coefficients can be chosen to reflect a conservative or a risk-taking strategy. On the other hand, evaluating the worst optimum value can be helpful to strategically choose a transportation provider among different options. Indeed, a company that outsources its transportation processes to a third party logistics (3PL) provider can evaluate the worst transportation cost associated with each of the possible 3PL candidates as part of its decision process.

In this paper, we focus on the transportation problem where supplies and demands are assumed to vary within a prespecified interval; we refer to this problem as the Interval Transportation Problem (ITP). Our focus is on determining the *best* and the *worst* values of the optimal cost of ITP among all the realizations of the uncertain parameters. We address the specific case of the problem which arises when the transportation costs of the problem are immune against the *transportation paradox*. The transportation paradox, also know as the *more-for-less-paradox* may occur on some instances of the transportation problem for which an increase in the amount of goods to be transported may lead to a decrease in the optimal total transportation cost. The analysis of the transportation paradox has been object of some research [8, 9] related to the classical transportation problem, however, none of the existing contributions related to the analysis of the Interval Transportation Problem have considered this issue. To the best of our knowledge, there exist only five contributions in the literature which specifically address the ITP in its general form. Chanas [10] presented a transformation technique to transform the ITP into a classical TP to find the best value of the optimal cost. Liu [11] constructed two mathematical models to find the best and the worst optimal values. Specifically, he provided a linear programming problem formulation whose solution provides the best optimum, and he provided a non-linear mathematical formulation to determine the worst optimum. Juman and Hoque [12] provided a heuristic solution to determine a lower bound to the worst optimum. Xie et al. [13] provided a genetic algorithm for the same problem. Additionally, Xie et al. [13] provided a necessary and sufficient condition under which finding the worst optimum becomes a polynomially solvable problem. Cerulli et al. [14] proved some general properties of the best and worst optimum values and proposed an Iterated Local Search algorithm to find a lower bound on the worst optimum value.

In this study, we analyze the ITP when costs are immune against the transportation paradox and derive theoretical properties of the problem which constitute the basis of a new efficient heuristic we designed to solve ITP. We compared our approach with the two most effective approaches in the literature, namely the Iterated Local Serch developed by Cerulli et al. [14], and the non linear model provided by Liu [11], on an extensive set of benchmark instances.

The remainder of the paper is organized as follows. Section 2 formally describes the

problem under study. The computational complexity of the problem and some analytical properties are shown in Section 3. Our algorithm is described in Section 4, and the computational results discussed in Section 5. Section 6 summarizes the contributions of the paper and the next steps of our research.

2 Problem Definition

Let $I = \{1,2,\ldots,m\}$ denote the set of m suppliers, and $J = \{1,2,\ldots,n\}$ the set of n customers. Available supply at supplier i, and required demand at customer j are denoted by $s_i, \forall i \in I$, and $d_j, \forall j \in J$, respectively. Let $c_{ij} \geq 0$ denote the unit transportaton cost from supplier $i \in I$ to customer $j \in J$. The classical Transportation Problem (TP) consists of finding the minimum cost transportation plan of goods from each supplier to each customer such that capacity at each supplier is not exceeded and each customer demand is satisfied. The mathematical formulation of TP to which we will refer in the remainder of the paper is the following:

$$(TP): \quad \min \sum_{i\in I, j\in J} c_{ij}x_{ij} \tag{1}$$

$$\sum_{j\in J} x_{ij} \leq s_i \quad \forall i \in I \tag{2}$$

$$\sum_{i\in I} x_{ij} = d_j \quad \forall j \in J \tag{3}$$

$$x_{ij} \geq 0 \quad \forall j \in J, \forall i \in I \tag{4}$$

where the variables x_{ij} associated with each supplier-customer pair represent the total amount of flow which is shipped from supplier i to customer j. The interval version of (TP) is obtained by allowing supply and demand to vary in a predefined non-negative range, $[\underline{s_i}, \overline{s_i}]$ and $[\underline{d_j}, \overline{d_j}]$, respectively. The mathematical formulation of the Interval Transportation Problem (ITP) is then directly derived from formulation (1)-(4) by allowing interval values in the supply and demand constraints:

$$(ITP): \quad \min \sum_{i\in I, j\in J} c_{ij}x_{ij} \tag{5}$$

$$\sum_{j\in J} x_{ij} \leq [\underline{s_i}, \overline{s_i}] \quad \forall i \in I \tag{6}$$

$$\sum_{i\in I} x_{ij} = [\underline{d_j}, \overline{d_j}] \quad \forall j \in J \tag{7}$$

$$x_{ij} \geq 0 \quad \forall j \in J, \forall i \in I \tag{8}$$

We define the pair $(\hat{s}, \hat{d})$ to be a *scenario* of (ITP). Specifically, $\hat{s}$ is an m-dimensional vector such that $\underline{s_i} \leq \hat{s}_i \leq \overline{s_i}, \ \forall i \in I$, while $\hat{d}$ is an n-dimensional vector such that $\underline{d_j} \leq \hat{d}_j \leq \overline{d_j}, \ \forall j \in J$. Given a scenario $(\hat{s}, \hat{d})$, we denote by TP$(\hat{s}, \hat{d})$ the corresponding

classical transportation problem, whose formulation is the following:

$$TP(\hat{s},\hat{d}): \quad \min \sum_{i\in I, j\in J} c_{ij}x_{ij} \tag{9}$$

$$\sum_{j\in J} x_{ij} \leq \hat{s}_i \quad \forall i \in I \tag{10}$$

$$\sum_{i\in I} x_{ij} = \hat{d}_j \quad \forall j \in J \tag{11}$$

$$x_{ij} \geq 0 \quad \forall j \in J, \forall i \in I \tag{12}$$

Hence, ITP can be seen as the collection of all classical transportation problems $TP(\hat{s},\hat{d})$ each associated with a scenario. We denote by $F(\hat{s},\hat{d})$ the feasible region of $TP(\hat{s},\hat{d})$, and by $z(\hat{s},\hat{d})$ its optimal transportation cost. Note that $F(\hat{s},\hat{d}) \neq \emptyset$ if and only if $\sum_{i\in I}\hat{s}_i \geq \sum_{j\in J}\hat{d}_j$. Let us denote by SD the region of all the admissible values of the supply and demand vectors, that is

$$SD = \{(s,d) \in R_+^{m+n} : \underline{s_i} \leq s_i \leq \overline{s_i}, \forall i \in I; \underline{d_j} \leq d_j \leq \overline{d_j}, \forall j \in J; F(s,d) \neq \emptyset\}$$

We are interested in determining the following two values:

$$\underline{z} = \{\min z(s,d) : (s,d) \in SD\} \tag{13}$$

$$\overline{z} = \{\max z(s,d) : (s,d) \in SD\} \tag{14}$$

The values $\{\underline{z},\overline{z}\}$ constitute the best and the worst optimal costs of ITP computed among all the feasible scenarios, respectively. Chanas [10] and Liu [11] showed that $\underline{z}$ can be determined in polynomial time by solving an appropriate linear programming problem. Hence, our focus is in determining $\overline{z}$. We specifically address problem (14) assuming the transportation costs are immune against the transportation paradox. In the next section we recall some properties of problem (14) which were already proved in [14], and prove some additional ones which hold specifically when the transportation paradox does not arise. The results proved in the next section are useful for the development of our solution approach described in Section 4.

3 Properties of $\overline{z}$

In this section we prove some properties of problem (14). Theorem 3.1 and properties 3.2 - 3.5 hold for the general version of ITP, that is, when the transportation costs are not immune against the paradox. Theorems 3.6 -3.9 are instead specific for the case under study.

Theorem 3.1 to follow, proves that the optimal value function $z(s,d)$ is a convex function, and hence problem (14) consists of maximizing a convex function over a polyedral set. This class of problems is NP-hard even in very special cases [15, 16].

Theorem 3.1. *Function $z(s,d)$ is convex.*

Proof. Let us consider two feasible scenarios $(s^1,d^1) \in SD$ and $(s^2,d^2) \in SD$, and let x^1 and x^2 be the optimal solutions of $TP(s^1,d^1)$ and $TP(s^2,d^2)$, respectively; that is $z(s^h,d^h) = \sum_{i\in I, j\in J} c_{ij}x^h_{ij}$, $h = 1,2$. We want to show that for each $0 \leq \lambda \leq 1$ the following holds:

$$z(\lambda s^1 + (1-\lambda)s^2, \lambda d^1 + (1-\lambda)d^2) \leq \lambda z(s^1,d^1) + (1-\lambda)z(s^2,d^2)$$

Let us consider the scenario (s^3,d^3) such that vector $s^3 = \lambda s^1 + (1-\lambda)s^2$ and $d^3 = \lambda d^1 + (1-\lambda)d^2$. Since the set SD is convex then $(s^3,d^3) \in SD$. Additionally, it is easy to verify that $x^3 = \lambda x^1 + (1-\lambda)x^2$ is a feasible solution of $TP(s^3,d^3)$. Hence we have the following:

$z(\lambda s^1 + (1-\lambda)s^2, \lambda d^1 + (1-\lambda)d^2) \leq \sum_{i\in I, j\in J} c_{ij}x^3_{ij}$

$= \lambda \sum_{i\in I, j\in J} c_{ij}x^1_{ij} + (1-\lambda)\sum_{i\in I, j\in J} c_{ij}x^2_{ij}$

$= \lambda z(s^1,d^1) + (1-\lambda)z(s^2,d^2).$

□

Two polynomially solvable cases of problem (14) arise when the extreme of the supply and of the demand intervals satisfy specific conditions. Specifically, properties 3.2 and 3.3 state the conditions under which $\overline{z}$ is achieved on the extreme scenarios $(\overline{s},\underline{d})$ and $(\underline{s},\overline{d})$, respectively.

Property 3.2. *[14] Given ITP such that* $\sum_{i\in I}\overline{s_i} = \sum_{j\in J}\underline{d_j}$, *then* $\overline{z} = z(\overline{s},\underline{d})$.

Property 3.3. *[13, 14] Given ITP such that* $\sum_{i\in I}\underline{s_i} \geq \sum_{j\in J}\overline{d_j}$ *then* $\overline{z} = z(\underline{s},\overline{d})$.

Properties 3.4 and 3.5 to follow are useful to analyze how the function $z(s,d)$ varies when moving from one scenario to another either changing only the supply or only the demand levels, respectively. Specifically, property 3.4 states that if we consider scenarios with the same demand values, then the optimal value of the objective function increases when the level of the supply decreases. While, property 3.5 states that if we consider scenarios with the same supply values, then the optimal value of the objective function decreases when the level of the demand decreases.

Property 3.4. *[14] If* $(\hat{s}^1,\hat{d})$ *and* $(\hat{s}^2,\hat{d})$ *are two scenarios of ITP such that* $\hat{s}^1_i \leq \hat{s}^2_i$, $\forall i \in I$, *then* $z(\hat{s}^1,\hat{d}) \geq z(\hat{s}^2,\hat{d})$.

Property 3.5. *[14] If* $(\hat{s},\hat{d}^1)$ *and* $(\hat{s},\hat{d}^2)$ *are two scenarios of ITP such that* $\hat{d}^2_j \geq \hat{d}^1_j$, $\forall j \in J$, *then* $z(\hat{s},\hat{d}^2) \geq z(\hat{s},\hat{d}^1)$.

Observe that, without the assumptions stated in properties 3.4 and 3.5, we cannot a-priori derive how the optimal values of two scenarios relate with each other as shown in the following example.

Example 1. *Let us consider the following instance of ITP with two suppliers and two customers (refer to Table 1). The available supply ranges are as follows:* $\underline{s}_1 = 9$, $\overline{s}_1 = 12$, *for the first supplier, and* $\underline{s}_2 = 10$, $\overline{s}_2 = 15$ *for the second supplier. The demand*

ranges are $\underline{d}_1 = 11$, $\overline{d}_1 = 12$ *for the first customer, and* $\underline{d}_2 = 11$, $\overline{d}_2 = 14$ *for the second customer. Unit transportation costs are:* $c_{11} = 10$, $c_{12} = 34$, $c_{21} = 36$, $c_{22} = 11$. *Let us consider the three scenarios* (s^1, d^1), (s^2, d^2), *and* (s^3, d^3) *shown in Table 1. The first scenario is such that* $s_1^1 = 10$, $s_2^1 = 15$, $d_1^1 = 12$, $d_2^1 = 13$, *and the optimal cost is* $z^1 = 315$. *The second scenario is such that* $s_1^2 = 11$, $s_2^2 = 15$, $d_1^2 = 13$, $d_2^2 = 13$, *and the corresponding optimal cost is* $z^2 = 325$. *Finally, the third scenario is such that* $s_1^3 = 11$, $s_2^3 = 15$, $d_1^3 = 12$, $d_2^3 = 14$, *and the corresponding optimal cost is* $z^3 = 300$. *Note that supply and demand of scenario two are greater than or equal to the supply and demand of scenario one, and the optimal cost of scenario two is greater than the optimal cost of scenario one. On the other hand, supply and demand of scenario three are greater than or equal to supply and demand of scenario one, but the optimal cost of scenario three is less than the optimal cost of scenario one.*

Table 1: *Optimal costs for three different scenarios of an instance of ITP with two suppliers and two customers as described in Example 1.*

	$\hat{s}_1$	$\hat{s}_2$	$\hat{d}_1$	$\hat{d}_2$	$z(\hat{s},\hat{d})$
Scenario 1	10	15	12	13	*315*
Scenario 2	11	15	13	13	*325*
Scenario 3	11	15	12	14	*300*

Cases as the one described in example 1 arise when the transportation costs c_{ij} are not immune against the transportation paradox. Formally, we define the costs c_{ij} to be immune against the transportation paradox if, given two scenarios $(\hat{s}^1, \hat{d}^1)$ and $(\hat{s}^2, \hat{d}^2)$ such that $\hat{s}_i^2 \geq \hat{s}_i^1$, $\forall i \in I$ and $\hat{d}_j^2 \geq \hat{d}_j^1$, $\forall j \in J$, then $z(\hat{s}^2, \hat{d}^2) \geq z(\hat{s}^1, \hat{d}^1)$. Note that, checking whether a cost matrix $C_{m\times n}$ is immune against the transportation paradox requires a polynomial number of steps [9]. When the costs c_{ij} are immune against the transportation paradox then we can prove a more general version of property 3.3. Let us introduce some definitions first.

Definition 1. *For a given value k, we define a* ***demand-restricted*** *worst optimum* $\overline{z}_k$ *the value of the worst optimum obtained among all the scenarios where the sum of the demands is equal to k, that is* $\overline{z}_k(s) = max\{z(s,d) : F(s,d) \neq \emptyset, \sum_{j\in j} d_j = k\}$

Definition 2. *For a given value k, we define a* ***supply-restricted*** *worst optimum* $\overline{z}_k$ *the value of the worst optimum obtained among all the scenarios where the sum of the supplies is equal to k, that is* $\overline{z}_k(d) = max\{z(s,d) : F(s,d) \neq \emptyset, \sum_{i\in I} s_i = k\}$

Theorem 3.6 to follow is a generalization of Property 3.3, when comparing two scenarios.

Theorem 3.6. *Let* $(\hat{s}^1, \hat{d}^1)$ *and* $(\hat{s}^2, \hat{d}^2)$ *be two feasible scenarios of ITP such that* $\hat{d}_j^2 \geq \hat{d}_j^1$, $\forall j \in J$. *If the costs* c_{ij} *are immune against the transportation paradox then* $\overline{z}_{k^1} \leq \overline{z}_{k^2}$, *where* $k^l = \sum_{j\in J} d_j^l, l = 1,2$.

Proof. Let $\sum_{j\in j} d_j^1 = k^1$, $\sum_{j\in j} d_j^2 = k^2$. Since from the hypothesis, $\hat{d}_j^2$ dominates $\hat{d}_j^1$, we have that $k^1 \leq k^2$. From property 3.2, it follows that $\overline{z}_{k^1} = z(s^{*1}, d^1)$ where $\sum_{i\in I} s_i^{*1} = k^1$. Similarly, $\overline{z}_{k^2} = z(s^{*2}, d^2)$ where $\sum_{i\in I} s_i^{*2} = k^2$. Since $k^1 \leq k^2$, it is possible to build from s^{*1} a new set of supply $\tilde{s}$, such that $\tilde{s}_j \geq s_j^{*1}$ and $\sum_{i\in I} \tilde{s}_i = k^2$. Let $\tilde{z}$ be the optimal solution value of the scenario $(\tilde{s}, d^2)$. Note that, $\tilde{z} \leq \overline{z}_{k^2}$, and since the costs are immune against the transportation paradox we also have that $\overline{z}_{k^1} \leq \tilde{z}$. Then, the thesis follows. □

As a consequence of Theorem 3.6 we can (i) determine one additional special instance of ITP which can be solved in polynomial time (a stated in Corollary 3.7 to follow), and (ii) restrict the search of $\overline{z}$ only to a subset of scenarios (as stated in Theorems 3.8 and 3.9 to follow).

Corollary 3.7. *Given ITP such that* $\sum_{i\in I} \overline{s_i} = \sum_{j\in J} \overline{d_j}$ *and such that the costs* c_{ij} *are immune against the transportation paradox then* $\overline{z} = z(\overline{s}, \overline{d})$.

Theorem 3.8. *Given ITP such that* $\sum_{i\in I} \overline{s_i} > \sum_{j\in J} \overline{d_j}$, $\sum_{j\in J} \overline{d_j} = k$ *and such that the costs* c_{ij} *are immune against the transportation paradox then*

$$\overline{z} = \max_{\underline{s}\leq s\leq \overline{s}} \overline{z}_k(s)$$

Proof. Due to Theorem 3.6, the optimum value $\overline{z}$ is obtained, on a scenario (s,d) such that $d = \overline{d}$. Due to Property 3.2, among all the scenarios $(s, \overline{d})$, $\overline{z}$ is obtained on a scenario where $\sum_{i\in I} s_i = k$. □

By applying a similar argument, the following theorem holds too.

Theorem 3.9. *Given ITP such that* $\sum_{i\in I} \overline{s_i} < \sum_{j\in J} \overline{d_j}$, $\sum_{i\in I} \overline{s_i} = k$, *and such that the costs* c_{ij} *are immune against the transportation paradox then*

$$\overline{z} = \max_{\underline{d}\leq d\leq \overline{d}} \overline{z}_k(d)$$

The heuristic presented in the next section finds a lower bound to $\overline{z}$ under the assumptions that the transportation costs are immune against the transportation paradox. The core of the algorithm is based on the results presented in this section. In Section 5 we compare the solution of our heuristic with the solution obtained by solving the nonlinear optimization model provided by Liu in [11]. We modified Liu's mathematical model to account for the case when the transportation costs are immune against the transportation paradox, the details are provided in the Appendix.

4 Our algorithm

Our heuristic is a constructive heuristic, which starts from an infeasible initial scenario and it iterates until this scenario becomes feasible. The procedure takes as input an immune cost matrix C_{mxn}, the interval demands $[\underline{d}, \overline{d}]$, the interval supplies $[\underline{s}, \overline{s}]$, and a parameter δ. It returns as output a feasible scenario and the corresponding optimal

value.
Given the results in Section 3 the algorithm considers three cases:

> ***Case 1.*** If $\sum_{i\in I}\bar{s}_i = \sum_{j\in J}\overline{d}_j$ then the algorithm returns the value $z(\bar{s},\overline{d})$, which is optimum according to Corollary 3.7.
>
> ***Case 2.*** If $\sum_{i\in I}\bar{s}_i > \sum_{j\in J}\overline{d}_j$, then the algorithm starts from the initial infeasible scenario obtained by setting all the demands at their upper values $\overline{d}_j, \forall j \in J$, and all the supplies at their lower values $\underline{s}_i, \forall i \in I$. The algorithm explores only demand-restricted scenarios where the demands are fixed at their upper bounds. At each iteration an origin is appropriately selected and its supply increased by δ. This operation is repeated until a feasible scenario $(s,\overline{d})$ is obtained. The algorithm returns the corresponding optimal solution $z(s,\overline{d})$ which is a lower bound to the worst optimum $\bar{z}$.
>
> ***Case 3.*** If $\sum_{i\in I}\bar{s}_i < \sum_{j\in J}\overline{d}_j$ then the algorithm starts from the initial infeasible scenario obtained by setting all the demands at their lower values $\underline{d}_j, \forall j \in J$, and all the supply at their upper values $\bar{s}_i, \forall i \in I$. The algorithm explores only supply-restricted scenarios where the supply are fixed at their upper bounds. At each iteration a destination is appropriately selected and its demand increased by δ. This operation is repeated until a feasible scenario $(\bar{s},d)$ is obtained. The algorithm returns the corresponding optimal solution $z(\bar{s},d)$ which is a lower bound to the worst optimum $\bar{z}$.

We describe how our algorithm works for Case 2 (similar steps are performed for Case 3). The description to follow refers to the pseudocode reported in Algorithm 1.

At iteration $k = 1$, an initial infeasible scenario $(s^{(k)},\overline{d})$ such that $s_i^{(k)} = \underline{s}_i, \forall i \in I$ is determined (line 2), and a value Δ is computed as the difference between the total demands and the total supplies of the current scenario (line 3).

From $(s^{(k)},\overline{d})$ a transportation problem with a dummy supply is build (procedure $BuildDTP()$, line 4) as follows. A dummy supply $m+1$ is added with supply value equal to $s_{m+1}^{(k)} = \Delta$. The dummy supply is connected to each destination j with unit transportation cost equal to the maximum among all the transportation costs from any origin to destination j, that is $c_{m+1,j} = \max_{i\in I} c_{ij}$.

The transportation problem with the dummy supply is solved (procedure $SolveDTP()$, line 5), and the corresponding solution $x_{ij}^{(k)}$ is analyzed to obtain a new scenario $(s^{(k+1)},\overline{d})$ (procedure $Update()$, line 6). If the new scenario is feasible then the algorithm stops and returns the corresponding optimal solution, otherwise the algorithm iterates by solving a new dummy transportation problem starting from $(s^{(k+1)},\overline{d})$.

The $Update()$ routine makes use of two sets, namely A_Γ and B. The set A_Γ, $\Gamma \subseteq J$, is a subset of the set I of the suppliers computed as follows. For each destination $j \in \Gamma$, we define by α_j the origin which is connected to j with maximum cost, that is

$\alpha_j = argmax_{i \in I} c_{ij}$, and the set A_Γ is such that[1] $A_\Gamma = \cup_{j \in \Gamma} \alpha_j$. The set $B = \{i_1, \ldots, i_m\}$ is an ordered set containing all the origins. Specifically, the origins in B are ordered increasingly with respect to the cost $c_i = max_{j \in J} c_{ij}$. The $Update()$ routine generates the scenario $(s^{(k+1)}, \overline{d})$, from the previous scenario $(s^{(k)}, \overline{d})$ by increasing the supply of one selected origin by a quantity δ. Specifically, let $x_{ij}^{(k)}$ be the solution of the transportation problem (with the dummy supply) obtained on line 5, and let $\Gamma = \{j_1, j_2, \ldots, j_h\}$ be the set of the destinations such that $x_{m+1,j}^{(k)} > 0$, and $A_\Gamma = \{\alpha_{j_1}, \alpha_{j_2}, \ldots, \alpha_{j_h}\}$ the corresponding associated origins. Without loss of generality, let us assume these sets are ordered in decreasing order of the cost $c_{m+1,j}$, that is $c_{m+1,j_1} \geq c_{m+1,j_1} \geq \ldots c_{m+1,j_h}$. The $Update()$ routine selects the first non saturated origin in A_Γ, that is the first origin whose corresponding supply is not equal to its upper bound. Let α^* be such an origin. The new scenario $(s^{(k+1)}, \overline{d})$ is obtained as follows: $s_{\alpha^*}^{(k+1)} = s_{\alpha^*}^{(k)} + \delta$ and $s_i^{(k+1)} = s_i^{(k)}, \forall i \neq \alpha^*$. When the set A_Γ is empty, then the new non saturated origin is selected from the set B.

Algorithm 1: Pseudocode of our algorithm for the case when $\sum_{i \in I} \overline{s}_i > \sum_{j \in J} \overline{d}_j$

Input: $C_{mxn}, [\underline{d}, \overline{d}], [\underline{s}, \overline{s}], \delta$;
Output: $z(s, \overline{d})$

1 $k \leftarrow 1$;
2 $s_i^{(k)} \leftarrow \underline{s}_i, \forall i \in I$;
3 $\Delta \leftarrow \sum_j \overline{d}_j - \sum_i s_i^{(k)}$;
4 $DTP(s^{(k)}, \overline{d}) \leftarrow BuildDTP(s^{(k)}, \overline{d})$;
5 $x_{ij}^{(k)} \leftarrow SolveDTP(s^{(k)}, \overline{d})$;
6 $s \leftarrow Update(s^{(k)}, x_{ij}^{(k)}, \delta)$;
7 **if** $\sum_j \overline{d}_j - \sum_i s_i \geq 0$ **then**
8 $\quad k \leftarrow k + 1$;
9 $\quad s_i^{(k)} \leftarrow s_i, \forall i \in I$;
10 $\quad$ **go to** 5;
11 **else**
12 $\quad$ **return** $z(s, \overline{d})$;

The computational complexity of our algorithm is $O(\frac{\Delta}{\delta} \times m)$. The total number of iterations is equal to $\frac{\Delta}{\delta}$. Indeed, at each iteration the procedure reduces the value Δ of a quantity δ by increasing the supply of a selected origin by δ, until Δ becomes equal to zero. Also, at each iteration the $Update()$ procedure selects the first non saturated origin from the set A_Γ in at most $O(|A_\Gamma|)$ steps, since the set A_Γ is a subset of the set I of the suppliers, then its size is at most m.

[1] To keep notation simple, we did not consider the case when a destination has associated multiple suppliers with the same maximum cost. If such a case occurs, then all of them are added to A_Γ.

5 Experimental Analysis

We compared our algorithm (δ-alg.) with the non linear program provided by Liu [11] (NLP-Liu) and the Iterated Local Search provided in Cerulli et al. [14] (ILS) on a set of random generated instances. The two approaches were chosen, among the existing ones, since, on the one hand, the non linear model returns the best quality solutions while, on the other hand, the Iterated Local Search shows the best trade-off between solution quality and computational time. We ran the algorithms considering different values of the input parameter δ, namely, $\delta = 0.1, 0.5, 1$. We considered different scenarios with multiple sizes $m \times n = 5 \times 5, 10 \times 10, 20 \times 20, 30 \times 30, 40 \times 40, 50 \times 50, 60 \times 60$, 70×70, 80×80, 90×90, and 100×100. For each size, we considered supply and demand interval with three different widths: $5, 10$ and 20. We generated 10 random instances for each combination of the parameters, for a total of 330 instances. The cost matrices were generated immune against the transportation paradox by applying the following condition stated in [9]:

Theorem 5.1. *A given $m \times n$ cost matrix $C = \{c_{ij}\}$ is immune against the transportation paradox if and only if, for all integers q, r, s, t with $1 \leq q$, $s \leq m$, $1 \leq r$, $t \leq n$, $q \neq s$, $r \neq t$ the following inequality is satisfied:*

$$c_{qr} \leq c_{qt} + c_{sr}$$

Therefore, in order to generate a cost matrix satisfying the above stated condition we can randomly generate values in the range $[\frac{k}{2}, k]$ for any given integer value $k > 0$. We fixed $k = 30$ and generated cost matrices with costs in the range $[15, 30]$. All the instances were generated such that $\sum_{i \in I} \overline{s}_i > \sum_{j \in J} \overline{d}_j$ and are available upon request to the authors. Our algorithm is coded in C++ and runs on a PC with Intel Core i5 2.4GHz and 8.00 GB of RAM. We solved each linear and nonlinear model with Cplex 12.8.

Figures 1, 2, and 3 show the performance of our algorithm when varying the value of the δ parameter for the class of scenarios grouped with respect to the width of the interval, that is, width = 5, 10 and 20, respectively. Each figure shows, for each class of scenarios, the average value of the percentage gap of the objective function value and the average value of the computational time, over the 10 instances of the scenario. Specifically, the average percentage gap between the solution value returned by our algorithm with respect to the solution value obtained by solving the non linear model is shown on the primary y-axis, and the corresponding average computational time (in seconds) on the secondary y-axis. The percentage gap was computed as $\frac{NLP - Delta}{NLP} \times 100$ where NLP is the optimal solution value obtained by solving the non linear model, and $Delta$ is the solution value returned by our algorithm. It appears evident from the figures that lower values of the parameter δ do not affect, on average, the quality of the returned solution, while hugely increase the computational time. The worsening of the computational time was an expected result given that it is inversely proportional to the value of the parameter δ.

Given these insights, the results of the non linear model and of the Iterated Local Search are compared with the results of our algorithm obtained when $\delta = 1$. Table 2 shows

Table 2: *Average running time, average obj. function value, and average percentage gap of the non linear model and of our algorithm (when $\delta = 1$) on all the scenarios.*

		Time (secs)		***Obj. function***		
Width	***m* × *n***	**NLP-Liu**	**δ-alg.**	**NLP-Liu**	**δ-alg.**	**% Gap**
5	5×5	0.01	0.01	3,084.00	3,070.10	0.45
	10×10	0.01	0.02	6,025.90	5,993.20	0.57
	20×20	0.16	0.08	13,951.50	13,922.00	0.22
	30×30	2.27	0.17	21,854.00	21,813.10	0.19
	40×40	10.07	0.44	37,599.40	37,561.00	0.10
	50×50	60.39	0.81	52,674.00	52,648.30	0.05
	60×60	1,137.09	1.25	63,806.47	63,745.50	0.09
	70×70	218.15	2.75	80,494.90	80,473.10	0.03
	80×80	825.88	4.39	115,106.90	115,079.80	0.02
	90×90	1,149.67	5.18	140,545.80	140,534.30	0.01
	100×100	2,791.77	7.24	159,866.50	159,856.80	0.01
10	5×5	0.00	0.01	3,700.20	3,674.70	0.75
	10×10	0.01	0.02	6,946.80	6,908.70	0.55
	20×20	0.16	0.09	14,700.40	14,632.00	0.47
	30×30	1.38	0.35	23,903.80	23,825.90	0.32
	40×40	246.57	0.79	39,995.34	39,864.30	0.33
	50×50	751.24	1.50	55,589.54	55,526.20	0.11
	60×60	558.49	1.99	69,584.49	69,507.30	0.11
	70×70	955.73	4.41	87,384.03	87,341.00	0.05
	80×80	613.58	6.87	125,948.50	125,923.40	0.02
	90×90	1,257.09	9.40	138,773.20	138,764.50	0.01
	100×100	2,950.46	10.08	160,993.70	160,979.40	0.01
20	5×5	0.01	0.02	4,515.50	4,461.50	1.19
	10×10	0.02	0.05	9,570.30	9,440.00	1.41
	20×20	0.18	0.23	18,085.50	17,990.80	0.51
	30×30	39.14	0.63	28,556.58	28,448.40	0.38
	40×40	6.50	1.71	47,460.40	47,311.70	0.32
	50×50	142.15	3.03	61,423.50	61,290.40	0.22
	60×60	501.45	3.79	74,371.50	74,278.70	0.12
	70×70	689.52	8.43	98,027.21	97,955.70	0.07
	80×80	1,897.72	15.58	138,745.80	138,686.00	0.04
	90×90	2,413.51	21.50	160,476.70	160,451.20	0.02
	100×100	2,951.50	25.98	183,589.10	183,568.60	0.01

the comparison with the non linear model, while Table 3 shows the comparison with the Iterated Local Search. Table 2 reports the average results (computed among the 10 instances of each scenario) of the non linear model and of our algorithm (when $\delta = 1$) for all the scenarios. The first two columns report the characteristics of the scenario, that is, the width (column Width) and the size of the instance (column $m \times n$), respectively. Average computational times (in seconds) of the two approaches are reported in the third and fourth columns. The last three columns report the average value of the objective function of the non linear model, of the objective function value of the solution returned by our algorithm, and the corresponding percentage gap, respectively. The computational times of both the approaches increase with the size of the instances, as it was expected. The gap between the computational times of our algorithm and that of the solver hugely increases with the size of the instance. For small instances (that is, instances of size up to 20×20), both the approaches require, on average, less than one second to solve an instance. For medium size instances (that is, instances of size between 30×30 and 60×60), the computational time of the δ-alg. ranges between 0.17 and 3.79 second, while the computational time of the solver ranges between 1.38 and 1137.09 secs, and it reaches the one hour time limit on 7 instances out of 40. Finally, on large instances (that is, instances of size between 70×70 and 100×100), the computational time of our algorithm ranges between 2.75 and 25.98 seconds, while the computational time of the solver ranges between 218.15 and 2951.50 seconds, and it reaches the one hour time limit on 21 instances out of 40.

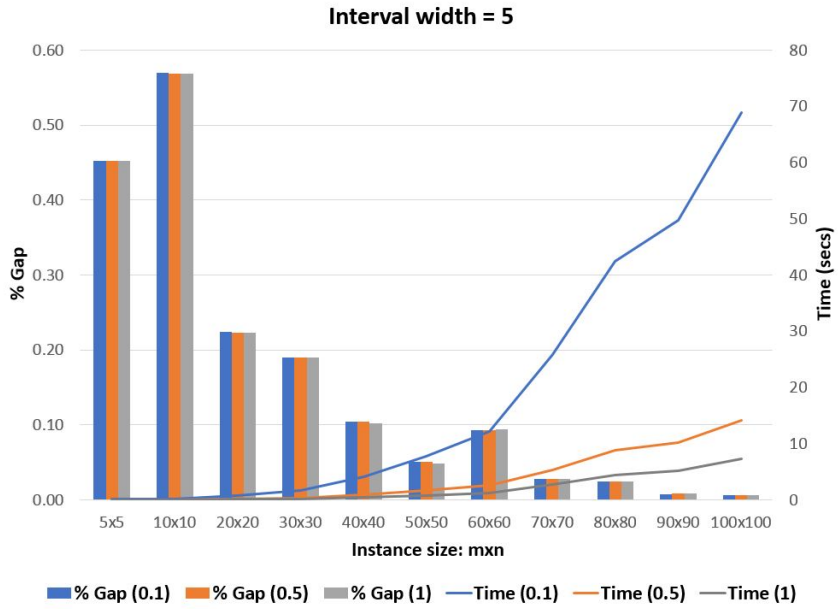
Figure 4 shows the plot of the average percentage gap on all the scenarios. The average percentage gap is always less that 1.5%. It decreases when the size of the instance increases, and it increases when the width of the intervals increases. Specifically, the gap ranges between 0.006% and 0.569% on instance with interval width equal to 5, between 0.006% and 0.754% when the interval width is equal to 10, and between 0.011% and 1.407% on instances with interval width equal to 20. The search space of the problem increases with the width of the intervals, this explains why the percentage gap increases as well. However, for a fixed width, even if the search space increases with the size of the instances, the gap decreases. We believe this is due to the greedy nature of the algorithm. The initial choices made by the algorithm (that is, the non saturated origins which are chosen at each iteration with the $Update()$ routine) limit the subsequent choices: if a wrong choice is made at the beginning the algorithm cannot overcome this mistake. This is more evident with small size instances, while with bigger instances the influence of initial wrong choices is mitigated. This explains, in our opinion, also why the percentage gap for big instances is almost the same no matter what the width of the intervals is.

Table 3 reports the results on each of the ten instances of a selected subset of scenarios of our delta algorithm with the Iterated Local Search in [14]. A similar behavior is shown for the remaining instances which are not reported here. The gap in the table was computed both for the objective function value (*%Gap - Obj*) and for the computational time (*%Gap - Time*) as $\frac{ILS-Delta}{Delta} \times 100$. As it can be seen, our δ-algorithm always finds a very similar solution in much less time, this is of course due to the fact that the ILS is a general purpose heuristic and it does not take into account the properties of the problem, hence it spends time exploring not good candidate solutions.

Table 3: *Running time, Obj. function value, and percentage gap of the Iterated Local Search and of our algorithm (when $\delta = 1$) on the set of ten instances for a subset of scenarios.*

				ILS		**δ - alg.**		***%Gap***	
Width	***m***	***n***	***Instance ID***	***Obj***	***Time (secs.)***	***Obj***	***Time (secs.)***	***Obj***	***Time***
5	40	40	1	41,035	809.67	41,018	0.728	0.000	1110.60
			2	37,179	199.19	37,148	0.168	0.001	1185.88
			3	32,521	22.48	32,511	0.041	0.000	553.99
			4	41,546	870.04	41,534	0.756	0.000	1149.09
			5	38,879	180.34	38,838	0.138	0.001	1304.11
			6	34,798	419.83	34,719	0.312	0.002	1343.76
			7	37,494	338.28	37,410	0.253	0.002	1333.63
			8	36,720	873.33	36,698	0.725	0.001	1203.81
			9	40,068	675.98	40,001	0.557	0.002	1211.69
			10	35,740	777.69	35,733	0.676	0.000	1149.79
5	50	50	1	58,090	1,561.35	58,070	0.999	0.000	1561.30
			2	52,143	1,833.07	52,107	1.154	0.001	1587.30
			3	59,679	1,968.20	59,650	1.318	0.000	1492.71
			4	52,720	302.50	52,694	0.237	0.000	1276.98
			5	52,764	2,125.77	52,729	1.388	0.001	1530.52
			6	49,247	925.26	49,231	0.682	0.000	1356.34
			7	54,920	2,102.11	54,886	1.255	0.001	1673.77
			8	46,704	84.92	46,690	0.092	0.000	919.39
			9	53,304	1,487.81	53,272	0.918	0.001	1619.07
			10	47,160	49.86	47,154	0.072	0.000	693.56
10	40	40	1	35,637	276.59	35,544	0.301	0.003	917.39
			2	40,838	700.20	40,663	0.812	0.004	861.82
			3	36,658	504.36	36,556	0.560	0.003	899.79
			4	38,397	366.95	38,316	0.422	0.002	868.43
			5	43,433	825.59	43,427	1.396	0.000	590.24
			6	43,060	728.46	42,906	0.879	0.004	827.30
			7	37,151	817.19	37,078	1.099	0.002	742.39
			8	40,775	404.16	40,732	0.507	0.001	795.38
			9	41,600	843.66	41,520	1.199	0.002	702.69
			10	42,022	567.67	41,901	0.679	0.003	834.79
10	50	50	1	56,175	2,113.97	56,096	2.331	0.001	905.93
			2	52,644	68.77	52,633	0.096	0.000	714.83
			3	56,791	1,777.09	56,720	1.782	0.001	995.99
			4	55,307	2,178.73	55,252	2.161	0.001	1007.20
			5	56,836	2,452.64	56,765	2.475	0.001	990.09
			6	55,699	829.27	55,660	0.748	0.001	1108.31
			7	53,491	993.14	53,446	1.105	0.001	897.41
			8	55,193	1,360.19	55,137	1.463	0.001	928.99
			9	57,481	1,834.13	57,399	1.964	0.001	932.94
			10	56,233	947.39	56,154	0.896	0.001	1055.94
20	40	40	1	44,595	995.88	44,519	2.514	0.002	395.20
			2	49,712	778.21	49,591	1.895	0.002	409.59
			3	49,266	1,012.31	49,134	2.514	0.003	401.73
			4	50,587	897.31	50,544	2.071	0.001	432.36
			5	46,279	329.46	46,145	0.632	0.003	520.03
			6	53,835	995.77	53,760	2.259	0.001	439.70
			7	44,686	410.86	44,578	0.903	0.002	454.09
			8	42,896	379.08	42,809	0.796	0.002	474.96
			9	44,803	715.74	44,621	1.620	0.004	440.69
			10	47,506	806.13	47,416	1.865	0.002	431.35
20	50	50	1	66,281	2,365.39	66,187	4.287	0.001	550.71
			2	64,253	2,452.99	64,249	5.323	0.000	459.82
			3	69,610	2,277.43	69,582	5.273	0.000	430.93
			4	49,424	513.55	49,287	0.641	0.003	800.12
			5	70,673	2,456.45	70,610	4.462	0.001	549.51
			6	48,467	362.81	48,357	0.540	0.002	671.48
			7	69,936	1,763.88	69,840	3.376	0.001	521.49
			8	58,901	1,066.52	58,831	1.818	0.001	585.65
			9	56,347	465.67	56,205	0.684	0.003	680.09
			10	59,880	2,154.09	59,756	3.851	0.002	558.43

Figure 1: *Average running time and average percentage gap between the solution value returned by the δ-alg. and the solution value of the non linear model solved using cplex, for instances with interval width = 5.*

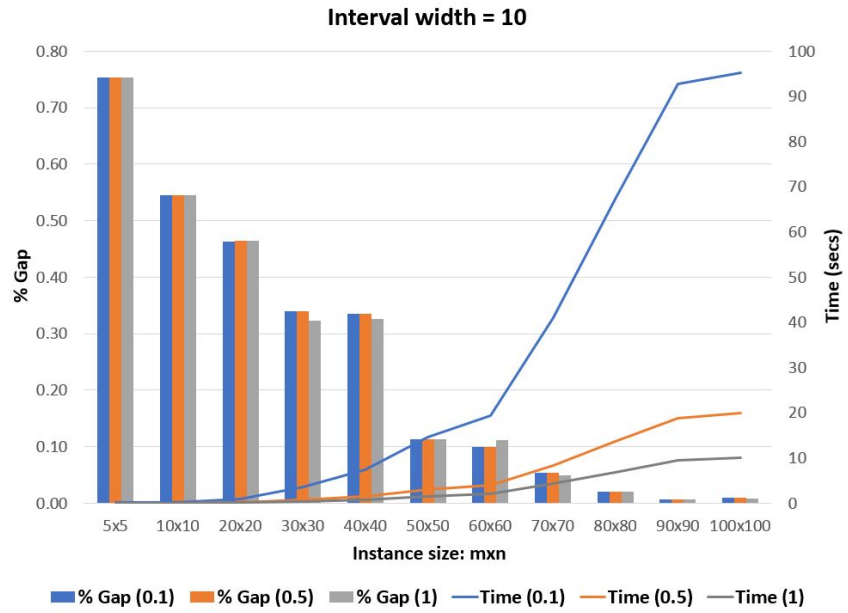


6 Conclusion

We addressed the problem of finding the range of the optimal cost of a transportation problem when supply and demand vary over an interval. We consider the specific version of a transportation problem with supply inequality constraints and demand equality constraints under the assumption that the transportation costs are immune against the transportation paradox. We investigated some theoretical properties of the problem of finding the worst optimum value which constitute the basis of a novel solution algorithm. Our results show that the proposed algorithm hugely outperforms the best existing solution approach in terms of computational time while returning a solution which is at most 1.5% worst.

We are planning to investigate several interesting aspects of the problem. First of all, we are planning to focus on different versions of the ITP which arise when either constraints on supply and demand are of the equality form or when there are supply equality constraints and demand inequality constraints. From an algorithmic perspective, we are planning to use our delta algorithm to design more efficient solution approaches to find both approximate and exact solutions of the problem.

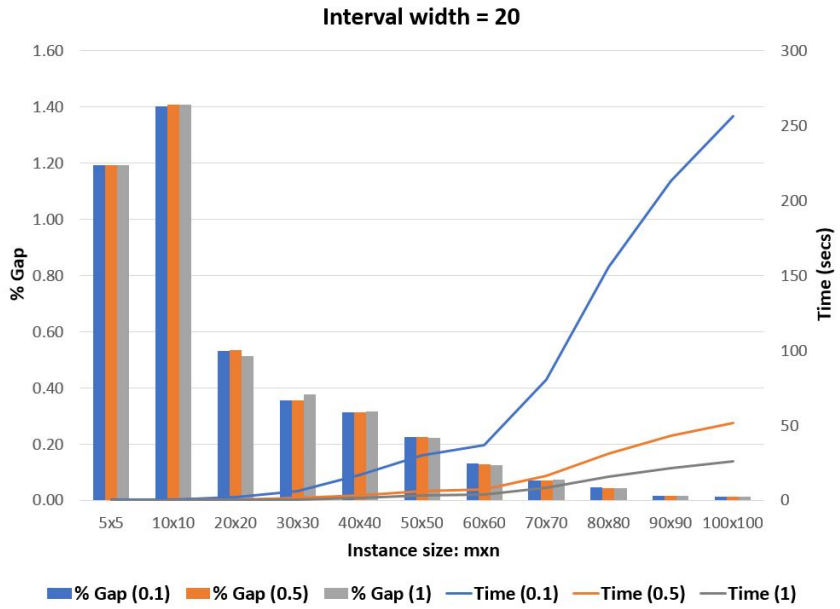
Figure 2: *Average running time and average percentage gap between the solution value returned by the δ-alg. and the solution value of the non linear model solved using cplex, for instances with interval width = 10.*



Acknowledgments

The authors would like to thank the two anonymous referees whose comments and suggestions helped improving the quality and readability of the paper.

Figure 3: *Average running time and average percentage gap between the solution value returned by the δ-alg. and the solution value of the non linear model solved using cplex, for instances with interval width = 20.*



References

[1] Abraham Charnes and William W Cooper. The stepping stone method of explaining linear programming calculations in transportation problems. *Management Science*, 1(1):49–69, 1954.

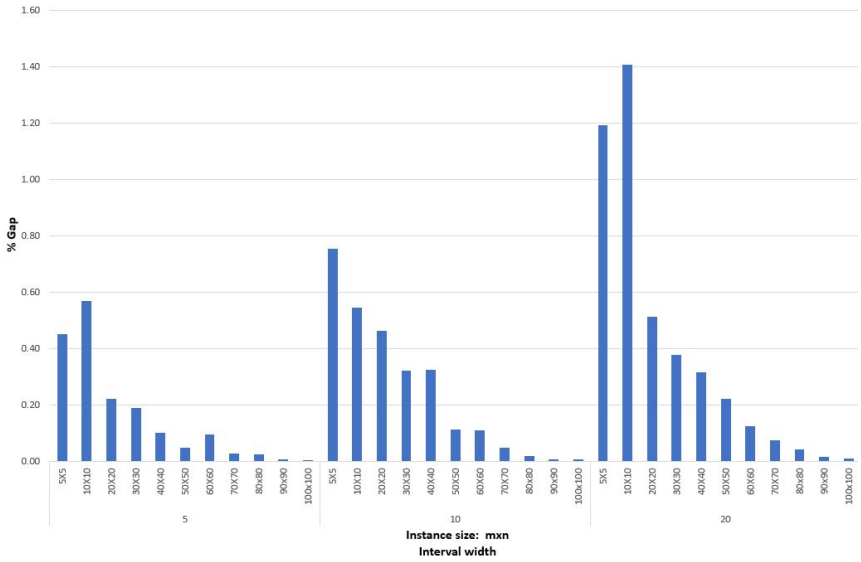
[2] Frank L Hitchcock. The distribution of a product from several sources to numerous localities. *Studies in Applied Mathematics*, 20(1-4):224–230, 1941.

[3] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network flows*. Elsevier, 2014.

[4] Ocotlán Díaz-Parra, Jorge A Ruiz-Vanoye, Beatriz Bernábe Loranca, Alejandro Fuentes-Penna, and Ricardo A Barrera-Cámara. A survey of transportation problems. *Journal of Applied Mathematics*, 2014, 2014.

[5] GM Appa. The transportation problem and its variants. *Journal of the Operational Research Society*, 24(1):79–99, 1973.

Figure 4: *Average percentage gap between the solution value returned by the δ-alg. and the solution value of the non linear model solved using cplex.*



[6] Milan Hladık. Interval linear programming: A survey. *Linear programming-new frontiers in theory and applications*, pages 85–120, 2012.

[7] Mohsen Mohammadi and Monica Gentili. Bounds on the worst optimal value in interval linear programming. *Soft Computing*, Dec 2018.

[8] Wlodzimierz Szwarc. The transportation paradox. *Naval Research Logistics (NRL)*, 18(2):185–202, 1971.

[9] Deineko Vladimir G, Bettina Klinz, and Gerhard J Woeginger. Which matrices are immune against the transportation paradox? *Discrete Applied Mathematics*, 130(3):495–501, 2003.

[10] S Chanas, M Delgado, JL Verdegay, and MA Vila. Interval and fuzzy extensions of classical transportation problems. *Transportation Planning and Technology*, 17(2):203–218, 1993.

[11] Shiang-Tai Liu. The total cost bounds of the transportation problem with varying demand and supply. *Omega*, 31(4):247–251, 2003.

[12] ZAMS Juman and MA Hoque. A heuristic solution technique to attain the minimal total cost bounds of transporting a homogeneous product with varying demands and supplies. *European Journal of Operational Research*, 239(1):146–156, 2014.

[13] Fanrong Xie, Muhammad Munir Butt, Zuoan Li, and Linzhi Zhu. An upper bound on the minimal total cost of the transportation problem with varying demands and supplies. *Omega*, 68:105–118, 2017.

[14] R. Cerulli, C. D'Ambrosio, and M. Gentili. Best and worst values of the optimal cost of the interval transportation problem. *Springer Proceedings in Mathematics and Statistics*, 217:367–374, 2017.

[15] Altannar Chinchuluun, Enkhbat Rentsen, and Panos M Pardalos. A numerical method for concave programming problems. In *Continuous Optimization*, pages 251–273. Springer, 2005.

[16] Panos M Pardalos and Georg Schnitger. Checking local optimality in constrained quadratic programming is np-hard. *Operations Research Letters*, 7(1):33–35, 1988.

Appendix

In this appendix we provide the bilinear mathematical formulation proposed in [11] opportunely modified to account for transportation costs which are immune against the transportation paradox. Let's recall problem (14):

$$\overline{z} = \{\max z(s,d) : (s,d) \in SD\}$$

which is equivalent to the following optimization problem:

$$\overline{z} = \max_{(s,d)\in SD} \min_{x} \sum_{i\in I, j\in J} c_{ij}x_{ij} \tag{15}$$

$$\sum_{j\in J} x_{ij} \leq s_i \quad \forall i \in I \tag{16}$$

$$\sum_{i\in I} x_{ij} = d_j \quad \forall j \in J \tag{17}$$

$$x_{ij} \geq 0 \quad \forall j \in J, \forall i \in I \tag{18}$$

where SD describes the set of feasible scenarios, that is:

$$SD = \{(s,d) \in R_+^{m+n} : \underline{s_i} \leq s_i \leq \overline{s_i}, \forall i \in I; \underline{d_j} \leq d_j \leq \overline{d_j}, \forall j \in J; F(s,d) \neq \emptyset\}$$

By applying duality theory, model (15)-(18) can be rewritten as follows:

$$\overline{z} = \max_{(s,d)\in SD} \max_{v,w} -\sum_{i\in} s_i v_i + \sum_{j\in J} d_j w_j \tag{19}$$

$$-v_i + w_j \leq c_{ij} \quad \forall i \in I, \forall j \in J \tag{20}$$

$$v_i \geq 0 \quad \forall i \in I \tag{21}$$

$$w_j \quad \textit{unrestricted} \quad \forall j \in J \tag{22}$$

Finally, model (19)-(22) can be rewritten as the following bilinear optimization model:

$$\overline{z} = \max_{v,w} -\sum_{i\in} s_i v_i + \sum_{j\in J} d_j w_j \tag{23}$$

$$-v_i + w_j \leq c_{ij} \qquad \forall i \in I, \forall j \in J \tag{24}$$

$$\underline{s_i} \leq s_i \leq \overline{s_i} \qquad \forall i \in I \tag{25}$$

$$\underline{d_j} \leq d_j \leq \overline{d_j} \qquad \forall j \in J \tag{26}$$

$$\sum_{i\in I} s_i \geq \sum_{j\in J} d_j \tag{27}$$

$$v_i \geq 0 \qquad \forall i \in I \tag{28}$$

$$w_j \quad unrestricted \qquad \forall j \in J \tag{29}$$

Consider the case that transportation costs are immune against the transportation paradox. Under the assumption that $\sum_{i\in I} \overline{s_i} > \sum_{j\in J} \overline{d_j}$ then, by Theorem 3.8, variables d_j in model (23)-(29) can be eliminated since they can be set to their upper bound $\overline{d}_j, \forall j \in J$. Similarly, when $\sum_{i\in I} \overline{s_i} < \sum_{j\in J} \overline{d_j}$ then, by Theorem 3.9, variables s_i can be eliminated since they can be set to their upper bound $\overline{s}_i, \forall i \in I$.