An Improved Heuristic Approach for the Interval Immune Transportation Problem

Francesco Carrabs[a], Raffaele Cerulli[a], Ciriaco D'Ambrosio[a,*], Federico Della Croce[b,c], Monica Gentili[d]

[a]*Dipartimento di Matematica, Università degli Studi di Salerno, Via Giovanni Paolo II 132, 84084 Fisciano (SA), Italia*
[b]*Dipartimento di Ingegneria Gestionale e della Produzione, Politecnico di Torino, (TO), Italia*
[c]*CNR, IEIIT, Torino, (TO), Italia*
[d]*Industrial Engineering Department, University of Louisville, KY, USA*

Abstract

We study the problem of determining the bounds of the optimal cost of a transportation problem when the capacity of the suppliers and the demand of the customers vary over an interval. We consider transportation costs such that the transportation paradox does not arise. We design a new heuristic approach based on some polyhedral properties of the problem and provide a novel integer linear programming mathematical formulation to solve it exactly. Our computational results, carried out on benchmark instances from the literature and on some new instances, show that our heuristic algorithm greatly outperforms the best solution approaches currently used.

Keywords: Interval Optimization, Interval RHS, Transportation Paradox, Transportation problem

1. Introduction

The Transportation Problem (TP) is one of the most important problem in operation research and optimization. It has been formulated for the first time in [1],

*Corresponding author
Email addresses: fcarrabs@unisa.it (Francesco Carrabs), raffaele@unisa.it (Raffaele Cerulli), cdambrosio@unisa.it (Ciriaco D'Ambrosio), federico.dellacroce@polito.it (Federico Della Croce), monica.gentili@louisville.edu (Monica Gentili)

more than seventy years ago and it still raises great research interest. It consists of determining the transportation plan for shipping items from suppliers to customers such that the total cost is minimized while meeting customers' demand and respecting capacity constraints at the suppliers. In this research, we examine the transportation problem in which it is assumed that supply and demand vary over specific intervals; this problem is referred to as the Interval Transportation Problem (ITP). Under this scenario, the transportation cost associated with the optimal solutions will also vary over a range. Our goal is to determine the range of the optimal cost of ITP over all possible realizations of the supply and demand parameters. Generally speaking, the addressed problem fits into the field of interval linear programming, where it is assumed that the uncertain coefficients of a linear problem vary independently within given ranges. In this context, the focus of our research leads to solve a max-min problem, rather than finding a stable solution in the worst case (as in robust optimization) which can be addressed by solving a min-max problem. The problem addressed in this paper relates, in particular, to the optimal value range problem [2, 3, 4, 5, 6, 7] which consists of finding the best and the worst optimal values, among the optimal values associated with all the possible data realizations, of an interval linear programming problem. The search for the worst optimum falls into the field of bilevel programming, since it can be seen as a bilevel problem where a leader selects the values to be assigned to the uncertain data, and then the follower solves the related deterministic linear problem. While the majority of the studies in the literature addresses the optimal value range problem for a generic interval linear programming problem, in this paper, we focus on the specific version obtained when the underlying interval linear problem is a transportation problem with interval right-hand-sides. ITP has been explored recently in the literature. Chanas [8] proposed a technique for transforming the ITP into a classical transportation problem and finding the lowest optimal cost value. Liu [9] provided a linear programming formulation to find the best optimal value and a nonlinear programming formulation, based on duality theory, to find the worst optimal value. In this paper, we introduce a mixed integer linear programming formulation inspired by Liu's nonlinear model to provide an exact approach to find the worst optimal value. Juman and Hoque [10] designed two heuristic approaches to find the best optimal value and a lower bound on the worst optimal value. Xie et al. [11] constructed a genetic algorithm to tackle this same problem. Cerulli et al. [12] studied certain properties of the range of the optimal cost; they also developed a Local Search based heuristic to determine a lower bound for the worst value of the optimum. More recently, D'Ambrosio et al. [13] presented the Immune Interval Transportation Problem (IITP), which is a special

case of ITP characterized by a transportation cost matrix that is immune against the transportation paradox. The *transportation paradox* is encountered in the classical transportation problem whenever –paradoxically– by increasing the quantity of goods sent, the transportation cost of the optimal solution decreases [14, 15]. D'Ambrosio et al. [13] studied some properties of IITP and provided a heuristic algorithm, namely, δ-Alg., to solve it.

In this work, from existing results on convex optimization, we present a novel mixed-integer linear model of the problem which can be used to optimally solve it. We also present a dual multi-start efficient solution heuristic. Our solution approaches are tested on a large set of instances and our dual algorithm is shown to greatly outperform the best solution approaches currently used in the literature.

The remaining parts of the paper are organized as follows. Section 2 presents a formal problem definition. Section 3 reviews existing results and provides new properties that are useful for developing our heuristic algorithm and our mathematical model. Section 4 describes the details of the dual algorithm. Section 5 contains a novel mixed-integer model to optimally solve the problem. Experimental results are presented in Section 6. Finally, Section 7 offers some closing remarks.

2. Problem Definition

Let $I = \{1,\ldots,m\}$ represent the set of the suppliers, and $J = \{1,\ldots,n\}$ the set of the customers. We denote the supply level of the supplier i, by s_i, $\forall i \in I$, while the quantity demanded at customer j, by d_j, $\forall j \in J$. Assume that both supply and demand vary over a specified non-negative interval, $[\underline{s_i}, \overline{s_i}]$ and $[\underline{d_j}, \overline{d_j}]$, respectively. Let $c_{ij} \geq 0$ represent the cost of transporting one unit of flow from supplier $i \in I$ to customer $j \in J$. The Interval Transportation Problem (ITP) is formulated as follows:

$$[ITP]: \quad \min \sum_{i\in I, j\in J} c_{ij}x_{ij} \tag{1}$$

$$\sum_{j\in J} x_{ij} \leq [\underline{s_i}, \overline{s_i}] \quad \forall i \in I \tag{2}$$

$$\sum_{i\in I} x_{ij} = [\underline{d_j}, \overline{d_j}] \quad \forall j \in J \tag{3}$$

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

where x_{ij} indicates the quantity of material that is shipped from supplier i to customer j. Constraints (2) ensure that the supply level at each supplier is respected,

and constraints (3) ensure that customer demand is always satisfied. Note that for any particular value of supply and demand (i.e., $s_i, \forall i$, and $d_j, \forall j$) within the specified intervals, the problem becomes the classical transportation problem in which the goal is to determine a transportation plan that results in the minimum total cost while satisfying all capacity and demand constraints. Hence, ITP consists of the complete set of classical transportation problems corresponding to all possible supply-demand realizations. Each of these TPs has an associate optimal cost, and, therefore, the optimal cost for the ITP varies over a range. Our goal is to find the range of these optimal costs, that is, the best and worst values in the entire set. Next, we formally define this problem. Let the pair (s,d) define a *scenario* of ITP. In particular, s is a vector of length m such that $\underline{s_i} \leq s_i \leq \overline{s}_i, \ \ \forall i \in I$, and d is a vector of length n such that $\underline{d_j} \leq d_j \leq \overline{d_j}, \ \ \forall j \in J$. Given any particular scenario (s,d), we denote by TP(s,d) the associated classical transportation problem, which is formulated as follows:

$$[TP(s,d)]: \quad \min \sum_{i\in I, j\in J} c_{ij}x_{ij} \tag{5}$$

$$\sum_{j\in J} x_{ij} \leq s_i \ \ \forall i \in I \tag{6}$$

$$\sum_{i\in I} x_{ij} = d_j \ \ \forall j \in J \tag{7}$$

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

Let $F(s,d)$ denote the feasible region of $TP(s,d)$, and let $z(s,d)$ denote the optimal transportation cost associated with it. Note that $F(s,d) \neq \emptyset$ if and only if $\sum_{i\in I} s_i \geq \sum_{j\in J} d_j$. Let SD denote the set of all *feasible* values of supply and demand (i.e., feasible scenarios), that is:

$$SD = \{(s,d) \in \mathbb{R}_+^{m+n} : \underline{s_i} \leq s_i \leq \overline{s}_i, \forall i \in I; \underline{d_j} \leq d_j \leq \overline{d_j}, \forall j \in J; \sum_{i\in I} s_i \geq \sum_{j\in J} d_j\}$$

Let us also assume that ITP is well defined and therefore $SD \neq \emptyset$. We address the problem of determining the two following values:

$$\underline{z} = \{\min z(s,d) : (s,d) \in SD\} \tag{9}$$

$$\overline{z} = \{\max z(s,d) : (s,d) \in SD\} \tag{10}$$

The values $\{\underline{z}, \overline{z}\}$ denote the total cost bounds, the best and the worst optimal costs, respectively, for ITP solved over all the feasible scenarios. Chanas [8] and

Liu [9] presented a method whereby $\underline{z}$ can be found easily via a linear programming formulation. Our focus, therefore, is to determine $\overline{z}$, i.e., solving problem (10). We specifically focus on finding $\overline{z}$ for the Interval Immune Transportation Problem (IITP), that is a specific version of ITP obtained when the transportation cost matrix in the model formulation (1)–(4) is immune against the transportation paradox. The transportation paradox was defined for the equality constrained version of the classical TP and it occurs for some instances for which an increase in the total amount of goods transported leads to a decrease of the corresponding optimal cost. In the remainder of the paper, we will assume that the assumption of immunity holds, i.e., the costs c_{ij} meet the following definition:

Definition 1. *[16] A given $m \times n$ cost matrix $C = \{c_{ij}\}$ of ITP is **immune against the transportation paradox** if, given any pair of scenarios (s^1, d^1) and (s^2, d^2) such that $s_i^2 \geq s_i^1$, $\forall i \in I$, $d_j^2 \geq d_j^1$, $\forall j \in J$, $\sum_i s_i^1 = \sum_j d_j^1$, and $\sum_i s_i^2 = \sum_j d_j^2$, then $z(s^2, d^2) \geq z(s^1, d^1)$.*

We remark that, as explained in [13], problem (10) is a member of the general class of problems involving the maximization of a convex function over a polyhedral set. This entire class of problems has been shown to be NP-hard even for very particular cases [17, 18], however, a formal proof of the computational complexity of IITP is still missing in the literature.

In the next section we recall some properties of problem (10) introduced in [13] and recall some properties from the convex programming literature, useful for developing the solution approaches presented in Sections 4 and 5.

3. IITP Properties

As the analysis carried out in [13], two specific versions of IITP are considered, namely the *supply-surplus* and the *demand-surplus* version, which are defined considering the vectors $\overline{s}$ and $\overline{d}$, that is $s_i = \overline{s}_i \quad \forall i \in I$, and $d_j = \overline{d_j} \quad \forall j \in J$. They are formally defined as follows.

Definition 2. *We define a version of IITP such that $\sum_{i \in I} \overline{s}_i > \sum_{j \in J} \overline{d_j}$, a **supply-surplus** version.*

Definition 3. *We define a version of IITP such that $\sum_{i \in I} \overline{s}_i < \sum_{j \in J} \overline{d_j}$, a **demand-surplus** version.*

The two versions defined above are the computational challenging versions of IITP, since the special case obtained when $\sum_{j \in J} \overline{d_j} = \sum_{i \in I} \overline{s}_i$ is solvable in polynomial time as stated in the following theorem:

Theorem 3.1. *[13] Given an instance of IITP such that* $\sum_{j\in J}\overline{d_j} = \sum_{i\in I}\overline{s}_i$ *then* $\overline{z} = z(\overline{s},\overline{d})$.

Note that the above theorem is true if the costs are immune against the transportation paradox. If such a condition is not satisfied then the theorem may not hold as shown in the following example.

Example 1. Let us consider an instance of ITP with two supply nodes and two demand nodes. The unit transportation costs are $c_{11} = 5$, $c_{12} = 17$, $c_{21} = 18$ and $c_{22} = 6$. The interval extremes of the supplies and of the demands are: $\underline{s}_1 = 7$ and $\overline{s}_1 = 10$ for the first supply, $\underline{s}_2 = 8$ and $\overline{s}_2 = 13$ for the second supply, $\underline{d}_1 = 9$ and $\overline{d}_1 = 11$ for the first demand and $\underline{d}_2 = 8$ and $\overline{d}_2 = 12$ for the second demand. Let us consider the two feasible scenarios in Table 1. The first scenario is such that $s_1^1 = \overline{s}_1$, $s_2^1 = \overline{s}_2$, $d_1^1 = \overline{d}_1$, $d_2^1 = \overline{d}_2$, and the corresponding optimal cost is equal to $z(s^1,d^1) = 140$. The second scenario is such that $s_1^2 = 9$, $s_2^2 = \overline{s}_2$, $d_1^2 = \overline{d}_1$ and $d_2^2 = 11$, and the optimal cost is $z(s^2,d^2) = 147$. Since the cost are not immune against the transportation paradox, decreasing the amount of goods to be transported may result in a higher cost, and the worst optimum may not be attained with the extreme scenario $(\overline{s},\overline{d})$, that is Theorem 3.1, in this example, does not hold.

	s_1	s_2	d_1	d_2	$z(s,d)$
Scenario 1	10	13	11	12	*140*
Scenario 2	9	13	11	11	*147*

Table 1: *Optimal costs for two different scenarios of the ITP instance described in Example 1.*

In the analysis to follow, for clarity of exposition, we will focus on the supply surplus-version of IITP. The analysis of the demand-surplus version of IITP is provided in the Appendix. The following Theorem 3.2 was proved in [13], and it states that the scenario where the optimum value $\overline{z}$ is achieved is such that all the demands d_j are equal to their upper value $\overline{d}_j$.

Theorem 3.2. *[13] Given a supply-surplus version of IITP then the optimal value* $\overline{z}$ *of* $z(s,d)$ *is the solution of the following optimization problem:*

$$\overline{z} = \{\max z(s,\overline{d}) : \underline{s} \le s \le \overline{s}; \sum_{i\in I} s_i \ge \sum_{j\in J} \overline{d}_j\} \tag{11}$$

Let us denote by P_S the polyhedron described by the constraints in (11), that is:

$$P_S = \{s \in \mathbb{R}^m : \underline{s} \leq s \leq \overline{s}; \sum_{i \in I} s_i \geq \sum_{j \in J} \overline{d}_j\} \tag{12}$$

The following corollary describes an additional polynomial case of problem (10):

Corollary 3.3. *Given a supply-surplus version of IITP, if* $\sum_{i \in I} \underline{s}_i \geq \sum_{j \in J} \overline{d_j}$*, then* $\overline{z} = z(\underline{s}, \overline{d})$.

Proof. Since $\sum_{i \in I} \underline{s}_i \geq \sum_{j \in J} \overline{d_j}$ then $\underline{s} \in P_S$, $F(\underline{s}, \overline{d}) \neq \emptyset$, and $TP(\underline{s}, \overline{d})$ admits an optimal solution. Let $z(\underline{s}, \overline{d})$ be the corresponding optimum value. Let us consider any other vector $s \in P_S$ such that $s \neq \underline{s}$. Note that $s_i \geq \underline{s}_i \ \forall i \in I$, and, since $F(\underline{s}, \overline{d}) \subseteq F(s, \overline{d})$, then $z(s, \overline{d}) \leq z(\underline{s}, \overline{d})$. Hence, $z(\underline{s}, \overline{d}) \geq z(s, \overline{d}), \forall s \in P_S$. □

For the supply-surplus version of IITP then the only meaningful version to be analyzed is when $\sum_{i \in I} \underline{s}_i < \sum_{j \in J} \overline{d_j}$. Under such an assumption we can restrict the set of scenarios to be analyzed to those for which the sum of the supplies is equal to the sum of the upper extremes of the demand intervals, as stated in the following lemma:

Lemma 3.4. *Given a supply-surplus version of IITP, if* $\sum_{i \in I} \underline{s}_i < \sum_{j \in J} \overline{d_j}$*, then the optimal value* $\overline{z}$ *of* $z(s, d)$ *is the solution of the following optimization problem:*

$$\overline{z} = \{\max z(s, \overline{d}) : \underline{s} \leq s \leq \overline{s}; \sum_{i \in I} s_i = \sum_{j \in J} \overline{d}_j\} \tag{13}$$

Proof. Since $\sum_{i \in I} \underline{s}_i < \sum_{j \in J} \overline{d_j} < \sum_{i \in I} \overline{s}_i$ then polyhedron $P_S \neq \emptyset$. We want to show that from any vector $s \in P_S$ such that $\sum_{i \in I} s_i > \sum_{j \in J} \overline{d}_j$ we can build another vector $s' \in P_S$ such that $\sum_{i \in I} s'_i = \sum_{j \in J} \overline{d}_j$ and such that $z(s', \overline{d}) \geq z(s, \overline{d})$. Let us consider any vector $s \in P_S$, such that $\delta = \sum_{i \in I} s_i - \sum_{j \in J} \overline{d}_j > 0$. Let us consider the two sets of indices I^1 and I^2 such that $s_i = \underline{s}_i$, $\forall i \in I^1$, and $s_i > \underline{s}_i, \forall i \in I^2$. Note that $|I^2| \geq 1$ since $\sum_{i \in I} \underline{s}_i < \sum_{j \in J} \overline{d}_j$. Let us build a new vector s' such that:

$s'_i = s_i = \underline{s}_i$, $\forall i \in I^1$;

and the following procedure is applied for the remaining components $i \in I^2$:

1: $\delta \leftarrow \sum_{i \in I} s_i - \sum_{j \in J} \overline{d}_j$
2: Randomly select $i \in I^2$

3: while $\delta > s_i - \underline{s}_i$
4: $I^2 \leftarrow I^2 \setminus \{i\}$
5: $s'_i = \underline{s}_i$;
6: $\delta = \delta - (s_i - \underline{s}_i)$;
7: Randomly select $i \in I^2$
8: $s'_i = s_i - \delta$;

Note that $s' \in P_S$, and $\sum_{i \in I} s'_i = \sum_{j \in J} \overline{d}_j$. Moreover, since s' is built such that $s'_i \leq s_i$, $\forall i \in I$, hence $F(s', \overline{d}) \subseteq F(s, \overline{d})$, and $z(s', \overline{d}) \geq z(s, \overline{d})$. □

In [13] the authors proved that $z(s,d)$ is a convex function hence, it is easy to see that function $z(s, \overline{d})$ is convex too. The maximum value of a convex function over a polyhedron is achieved on one of the vertices of the polyhedron [19]. Polytope P_S is known in the literature as the continuous knapsack polytope. A complete characterization of its vertices has been provided in the literature [20, 21]. In what follows, we reinterpret the existing results through the lens of the transportation problem application. These results constitute the basis of the mathematical formulation and of the dual-algorithm presented in the sections to follow. Let us introduce the definition of an extreme scenario and of a supply-quasi-extreme scenario first.

Definition 4. *A scenario (s,d) is an **extreme** scenario if each supply s_i and each demand d_j is equal to either its lower bound or its upper bound, i.e., $s_i \in \{\underline{s}_i, \overline{s}_i\}$, $\forall i \in I$ and $d_j \in \{\underline{d}_j, \overline{d}_j\}$, $\forall j \in J$.*

We focus on supply-quasi-extreme scenarios, that is those scenarios for which *at most one* among all the supplies might *not* be at either its lower or upper bound.

Definition 5. *A scenario (s,d) is a **supply-quasi-extreme** scenario if there exists **at most** one index i_h such that $\underline{s}_{i_h} < s_{i_h} < \overline{s}_{i_h}$, while $s_i \in \{\underline{s}_i, \overline{s}_i\}$ $\forall i \in I \setminus \{i_h\}$ and $d_j \in \{\underline{d}_i, \overline{d}_j\}$, $\forall j \in J$.*

The specific supply that, for a given supply-quasi-extreme scenario, might assume values internal to its corresponding interval is referred to as the *critical* supply of the scenario. The formal definition is given next.

Definition 6. *Given a supply-quasi-extreme scenario we refer to the supply i_h which might not be equal to one of the extremes of its interval as the **critical supply**.*

Existing results on the characterization of the vertices of the polytope P_S lead directly to the following property whose proof is therefore omitted:

Property 3.5. *Given a supply-surplus version of IITP the following statements hold:*

- *the vertices of the polyhedron P_S correspond to supply-quasi-extreme scenarios;*
- *the optimal value $\bar{z}$ of $z(s,d)$ is achieved on a supply-quasi-extreme scenario;*
- *if $\underline{s}$, $\bar{s}$, $\underline{d}$ and $\overline{d}$ have integer components, then the vertices of polyhedron P_S are integer.*

Summarizing, thanks to Lemma 3.4 and Property 3.5, to solve problem (10), we can focus only on the subset of supply-quasi-extreme scenarios such that $\sum_{i\in I} s_i = \sum_{j\in J} \overline{d}_j$. Additionally, when $\underline{s}$, $\bar{s}$, $\underline{d}$ and $\overline{d}$ have integer components, we can restrict further to consider only the supply-quasi-extreme scenarios with integer components.

4. Dual Algorithm

In this section we introduce a dual multistart based algorithm, named *Dual*-Alg., to compute a feasible solution to problem (10). All the discussion is carried out for the supply-surplus version of IITP, however, it can easily be adapted to solve the demand-surplus version as well.

Let us consider a feasible scenario $(s,\overline{d}) \in SD$ and the mathematical formulation $TP(s,\overline{d})$ of the associated classical transportation problem:

$$z(s,\overline{d}) = \min \sum_{i\in I, j\in J} c_{ij}x_{ij} \tag{14}$$

$$\sum_{j\in J} x_{ij} \leq s_i \ \ \forall i \in I \tag{15}$$

$$\sum_{i\in I} x_{ij} = \overline{d}_j \ \ \forall j \in J \tag{16}$$

$$x_{ij} \geq 0 \ \ \forall i \in I, \ \forall j \in J \tag{17}$$

From duality theory, the optimum value $z(s,\overline{d})$ can be found solving the following associated dual formulation $DTP(s,\overline{d})$ [9]:

$$z(s,\overline{d}) = \max \sum_{i\in I} s_i u_i + \sum_{j\in J} \overline{d}_j v_j \tag{18}$$

$$c_{ij} \geq u_i + v_j \quad \forall i \in I,\ \forall j \in J \tag{19}$$

$$u_i \leq 0,\ \ v_j \ \ unsigned, \quad \forall i \in I,\ \forall j \in J \tag{20}$$

where u_i and v_j are the dual variables associates with constraints (15)–(16), respectively. If we consider the supplies s_i, $\forall i \in I$, to be variables, then problem (13), given Lemma 3.4, can be obtained by solving the following non linear optimization model:

$$\max \sum_{i\in I} s_i u_i + \sum_{j\in J} \overline{d}_j v_j \tag{21}$$

$$c_{ij} \geq u_i + v_j \quad \forall i \in I,\ \forall j \in J \tag{22}$$

$$\underline{s}_i \leq s_i \leq \overline{s}_i \qquad \forall i \in I \tag{23}$$

$$\sum_{i\in I} s_i = \sum_{j\in J} \overline{d}_j \tag{24}$$

$$u_i \leq 0,\ \ v_j \ \ unsigned, \quad \forall i \in I,\ \forall j \in J \tag{25}$$

Let us consider formulation (21)–(25) and assume the value u_i are given (i.e., $u_i = u_i^*$), the corresponding mathematical formulation becomes then:

$$\max \sum_{i\in I} s_i u_i^* + \sum_{j\in J} \overline{d}_j v_j \tag{26}$$

$$c_{ij} \geq u_i^* + v_j \quad \forall i \in I,\ \forall j \in J \tag{27}$$

$$\underline{s}_i \leq s_i \leq \overline{s}_i \qquad \forall i \in I \tag{28}$$

$$\sum_{i\in I} s_i = \sum_{j\in J} \overline{d}_j \tag{29}$$

$$v_j \ \ unsigned, \qquad \forall j \in J \tag{30}$$

This formulation can be simplified by observing that variables v_j can be set such that $v_j = \min_i(c_{ij} - u_i^*)$, $\forall i$, and hence constraints (27) can be removed from the model. As a result, the model can then be easily solved by using the greedy procedure presented in Algorithm 1. Specifically, the procedure initially sets all the values $s_i = \underline{s}_i$. Successively, it sorts all the supplies in non increasing order with respect to the corresponding value u_i^*, and set them to their upper bound $\overline{s}_i$ until constraint (29) is respected.

Algorithm 1: Constructive procedure to solve formulation (26) – (30)

Input: $u^*, C_{m\times n}, [\underline{d}, \overline{d}], [\underline{s}, \overline{s}]$;

1 $s_i \leftarrow \underline{s}_i$;
2 $supply \leftarrow sort(s_i, u_i^*)$; // sort s_i, in non increasing order, according to u_i^*
3 $\Delta \leftarrow \sum_{j\in J} \overline{d}_j - \sum_{i\in I} s_i$;
4 **for** $i = 1$ *to* m **do**
5 $\quad s_k \leftarrow supply[i]$;
6 $\quad$ **if** $\Delta \geq \overline{s}_k - s_k$ **then**
7 $\quad\quad \Delta \leftarrow \Delta - (\overline{s}_k - s_k)$;
8 $\quad\quad s_k \leftarrow \overline{s}_k$;
9 $\quad$ **else**
10 $\quad\quad s_k \leftarrow s_k + \Delta$;
11 $\quad\quad \Delta \leftarrow 0$;
12 $\quad\quad$ break;
13 $v_j \leftarrow \min_i(c_{ij} - u_i^*), \forall i \in I, \forall j \in J$;
14 **return** $\sum_i s_i u_i^* + \sum_j \overline{d}_j v_j$;

Our dual multistart algorithm randomly generates a supply-quasi extreme scenario $(s, \overline{d})$, solves the dual formulation $DTP(s, \overline{d})$ (i.e., model (18) – (20)) and determines the corresponding optimal values u_i^*. Then, *Dual*-Alg. uses the optimal values u_i^* of the dual variables as input of Algorithm 1 to solve formulation (26) – (30) to further increase the objective function value. The process is iterated a predefined number of times.

More specifically, let us consider the pseudo-code given in Algorithm 2. Line 1 initializes the value of the current best solution. If we are under the condition of Corollary 3.3, then the optimum value $\overline{z}$ is found by solving a single transportation problem (Line 3). An initial random supply-quasi-extreme scenario is generated by randomly assigning to all but one supply either value $\underline{s}_i$ or value $\overline{s}_i$ so that the sum of the supplies is equal to the sum of the demands. This is obtained by steps 6–17. Specifically, initially (Line 7) all the supplies s_i are set equal to their lower bound, and (Line 8) the supply surplus Δ is computed. The loop in lines 8–17 aims at changing the value of some of the supplies from their lower bound to their upper bound until $\sum_{i\in I} s_i = \sum_{j\in J} \overline{d}_j$.At each iteration of the loop (Lines 8–17), each supply is selected uniformly at random (*randomSupply* procedure, line 10) over the supplies not assigned yet.

Then, model (18) – (20) is solved considering the randomly generated supply-quasi-extreme scenario and the best u_i^*'s are obtained (Line 21).

Algorithm 2: Pseudocode of *Dual*-Alg. for the supply-surplus version of IITP

Input: $C_{m\times n}, [\underline{d}, \overline{d}], [\underline{s}, \overline{s}], IT_{max}$;

1 $BestSol \leftarrow 0$;
2 **if** $\sum_{i\in I} \underline{s}_i \geq \sum_{j\in J} \overline{d_j}$ **then**
3 $BestSol \leftarrow z(\underline{s}, \overline{d})$;
4 **else**
5 **for** $h = 1$ *to* IT_{max} **do**
6 $IterationBestSol \leftarrow 0$;
7 $s_i \leftarrow \underline{s}_i \ \forall i \in I$;
8 $\Delta \leftarrow \sum_{j\in J} \overline{d}_j - \sum_{i\in I} s_i$;
9 **do**
10 $k \leftarrow randomSupply(1 \dots m)$;
11 **if** $\Delta \geq \overline{s}_k - s_k$ **then**
12 $\Delta \leftarrow \Delta - (\overline{s}_k - s_k)$;
13 $s_k \leftarrow \overline{s}_k$;
14 **else**
15 $s_k \leftarrow \underline{s}_k + \Delta$;
16 $\Delta \leftarrow 0$;
17 **while** $\Delta > 0$;
18 $DualSol \leftarrow 0$;
19 **do**
20 **if** $DualSol > IterationBestSol$ **then**
21 $IterationBestSol \leftarrow DualSol$;
22 Solve model (18) - (20), and let u^* be the optimal values of the decision variables;
23 $DualSol \leftarrow Algorithm1(u^*, C_{m\times n}, [\underline{d}, \overline{d}], [\underline{s}, \overline{s}])$;
24 **while** $DualSol > IterationBestSol$;
25 **if** $IterationBestSol > BestSol$ **then**
26 $BestSol \leftarrow IterationBestSol$;
27 **return** $BestSol$;

Finally (Line 23), model (26) – (30) is solved considering the fixed values u_i^* to further increase the objective function value. The whole process is iterated for a given number IT_{max} of iterations.

5. An integer programming formulation

In this section, we provide an exact method to solve problem (13). The approach is based on a mixed integer linear programming formulation (MILP) in-

spired by Liu's bilinear model. Unlike Liu's model, our approach explores only scenarios based on Property 3.5, that is quasi-extreme-scenarios. The idea behind our method is to enumerate all quasi-extreme-scenarios. In particular, we consider, in turn, each supply as the critical supply h. Then, for each integer value in $[\underline{s}_h, \bar{s}_h]$, we solve the MILP formulation to decide the values of the remaining supplies, that is to determine which supply has to be equal to its lower bound and which one to its upper bound. Let us now describe our exact approach in more details.

Let us consider a feasible scenario (s,d) and the two following versions of the classical transportation problem:

$$z_1 = \min \sum_{i\in I, j\in J} c_{ij}x_{ij} \tag{31}$$

$$\sum_{j\in J} x_{ij} \leq s_i \ \ \forall i \in I \tag{32}$$

$$\sum_{i\in I} x_{ij} = d_j \ \ \forall j \in J \tag{33}$$

$$x_{ij} \geq 0 \ \ \forall i \in I, \ \ \forall j \in J \tag{34}$$

$$z_2 = \min \sum_{i\in I, j\in J} c_{ij}x_{ij} \tag{35}$$

$$\sum_{j\in J} x_{ij} \geq s_i \ \ \forall i \in I \tag{36}$$

$$\sum_{i\in I} x_{ij} \geq d_j \ \ \forall j \in J \tag{37}$$

$$x_{ij} \geq 0 \ \ \forall i \in I, \ \ \forall j \in J \tag{38}$$

The following Lemma shows that, under some conditions, the optimum value of model (31)–(34) and the optimal value of model (35)–(38) are the same.

Lemma 5.1. *Given a scenario (s,d) such that $\sum_{i\in I} s_i = \sum_{j\in J} d_j$ and such that the costs are immune against the transportation paradox then $z_1 = z_2$.*

Proof. Let x^* be the optimal solution of model (31)–(34) and let the corresponding value of the objective function be z_1. Since $\sum_i s_i = \sum_j d_j$, then constraints (32) are all binding and x^* is also feasible for (35)–(38). Note that, x^* is also optimum for (35)–(38). Indeed, let us assume by contradiction that the optimal solution for

(35)–(38) is a vector $\hat{x}$ different from x^* and whose corresponding optimal value is $\hat{z} < z_1$. Since $\hat{x} \neq x^*$, this would imply that at least one of the constraints (36)–(37) is not binding. Hence, we would have $\sum_{ij} \hat{x}_{ij} > \sum_{ij} x^*_{ij}$, however, since we assumed $\hat{z} < z_1$, this is a contradiction, since costs are immune against the transportation paradox. Hence, x^* is optimum for (35)-(38) and the claim follows. □

Now let us consider the dual of the optimization model (35)–(38) for a generic scenario $(s, \overline{d})$, which is as follows:

$$\max \sum_{i \in I} s_i u_i + \sum_{j \in J} \overline{d}_j v_j \tag{39}$$

$$c_{ij} \geq u_i + v_j \quad \forall i \in I,\ \forall j \in J \tag{40}$$

$$u_i \geq 0 \qquad \forall i \in I \tag{41}$$

$$v_j \geq 0 \qquad \forall j \in J \tag{42}$$

If we consider the supplies s_i, $\forall i \in I$, to be variables, then problem (13), given Lemma 3.4 and Lemma 5.1, consists in solving the following non linear optimization model:

$$\max \sum_{i \in I} s_i u_i + \sum_{j \in J} \overline{d}_j v_j \tag{43}$$

$$c_{ij} \geq u_i + v_j \quad \forall i \in I,\ \forall j \in J \tag{44}$$

$$\underline{s}_i \leq s_i \leq \overline{s}_i \qquad \forall i \in I \tag{45}$$

$$\sum_{i \in I} s_i = \sum_{j \in J} \overline{d}_j \tag{46}$$

$$u_i \geq 0,\ \ s_i \geq 0 \qquad \forall i \in I \tag{47}$$

$$v_j \geq 0 \qquad \forall j \in J \tag{48}$$

We know that the optimum of problem (13) occurs at a supply-quasi-extreme scenarios. Let us assume we know the critical supply h such that $\underline{s}_h \leq s_h \leq \overline{s}_h$ and its exact value. The following mathematical programming formulation can be used to determine the optimum value $\overline{z}$, by optimally setting the remaining supplies values s_i, $i \in I \backslash \{h\}$ to be equal to either their lower bound $\underline{s}_i$ or their upper bound $\overline{s}_i$. Let us introduce m binary variables y_i where $y_i = 1$ implies $s_i = \overline{s}_i$, and $s_i = \underline{s}_i$ otherwise. Then, model (43)–(48) can be rewritten as follows:

$$\max\ s_h u_h + \sum_{i\in I,\, i\neq h} \underline{s}_i u_i + (\overline{s}_i - \underline{s}_i) y_i u_i + \sum_{j\in J} \overline{d}_j v_j \tag{49}$$

$$c_{ij} \geq u_i + v_j \quad \forall i \in I,\ \forall j \in J \tag{50}$$

$$s_h + \sum_{i\in I,\, i\neq h} \underline{s}_i + (\overline{s}_i - \underline{s}_i) y_i = \sum_{j\in J} \overline{d}_j \tag{51}$$

$$u_i \geq 0 \qquad \forall i \in I \tag{52}$$

$$v_j \geq 0 \qquad \forall j \in J \tag{53}$$

$$y_i \in \{0,1\} \qquad \forall i \in I \tag{54}$$

Model (49)–(54) is still non linear but an equivalent integer linear programming formulation can be determined as follows. Let $c_i^{min} = \min_{j\in J}\{c_{ij}\}\ \forall i \in I$, that is c_i^{min} is the minimum cost coefficient c_{ij} among all destinations d_j connected to supply s_i. Consider introducing a set of m variables t_i with the aim of having $\sum_{i\in I,\, i\neq h} t_i = \sum_{i\in I,\, i\neq h} (\overline{s}_i - \underline{s}_i) y_i u_i$. We guarantee that t_i is not superior to $(\overline{s}_i - \underline{s}_i) y_i u_i$ whatever the value of y_i, by introducing the following constraints:

$$t_i \leq (\overline{s}_i - \underline{s}_i) u_i \quad \forall i \in I \tag{55}$$

and, since $c_{ij} \geq u_i + v_j \geq u_i$, we add also the following constraints:

$$t_i \leq (\overline{s}_i - \underline{s}_i) c_i^{min} y_i \quad \forall i \in I \tag{56}$$

This final set of big-M constraints guarantees that the following ILP model reaches the same solution of model (49)–(54) when we assume to know the critical supply h and its optimum value s_h:

$$\text{Max}\ s_h u_h + \sum_{i\in I,\, i\neq h} \underline{s}_i u_i + t_i + \sum_{j\in J} \overline{d}_j v_j \tag{57}$$

$$c_{ij} \geq u_i + v_j \quad \forall i \in I,\ \forall j \in J \tag{58}$$

$$t_i \leq (\overline{s}_i - \underline{s}_i) u_i \qquad \forall i \in I \tag{59}$$

$$t_i \leq (\overline{s}_i - \underline{s}_i) c_i^{min} y_i \qquad \forall i \in I \tag{60}$$

$$s_h + \sum_{i\in I,\, i\neq h} \underline{s}_i + (\overline{s}_i - \underline{s}_i) y_i = \sum_{j\in J} \overline{d}_j \tag{61}$$

$$u_i \geq 0 \qquad \forall i \in I \tag{62}$$

$$v_j \geq 0 \qquad \forall j \in J \tag{63}$$

$$t_i \geq 0 \qquad \forall i \in I \tag{64}$$

$$y_i \in \{0,1\} \qquad \forall i \in I \tag{65}$$

The above formulation (57)–(65) can be used to find the optimal value $\bar{z}$ when we assume the vectors $\underline{s}$, $\bar{s}$, $\underline{d}$ and $\overline{d}$ have integer components. The main idea is to solve model (57)–(65) multiple times by changing the selection of the critical supply and its value as shown in Algorithm 3. Line 1 initializes the value of the current best solution. If we are under the condition of Corollary 3.3 (Line 2), then the optimum value $\bar{z}$ is found by solving a single transportation problem (Line 3). Line 5–9 contain two embedded for loops. The external loop fixes a supply to be the critical supply, and the inner loop solves model (57)–(65) for each possible integer value that the critical supply can assume inside its interval. The best value among all is then returned. Given the results in the previous section, and under the assumption of integrality of $\underline{s}$, $\bar{s}$, $\underline{d}$, and $\overline{d}$, Algorithm 3 returns the optimum value $\bar{z}$.

Algorithm 3: Pseudocode of our exact algorithm for the supply-surplus version of IITP, when $\underline{s}$, $\bar{s}$, $\underline{d}$, and $\overline{d}$ are integer values.

Input: $C_{mxn}, [\underline{d}, \overline{d}], [\underline{s}, \bar{s}]$;

1 $BestSol \leftarrow 0$;
2 **if** $\sum_{i \in I} \underline{s}_i \geq \sum_{j \in J} \overline{d_j}$ **then**
3 $\quad Bestsol \leftarrow z(\underline{s}, \overline{d})$;
4 **else**
5 $\quad$ **for** $h \in I$ **do**
6 $\quad\quad$ **for** $s_h \in [\underline{s}_h, \ldots, \bar{s}_h] \cap \mathbb{Z}$ **do**
7 $\quad\quad\quad$ Solve model (57)–(65) and let z_h be its optimal solution value;
8 $\quad\quad\quad$ **if** $z_h > Bestsol$ **then**
9 $\quad\quad\quad\quad Bestsol \leftarrow z_h$;
10 **return** $BestSol$;

6. Experimental Results

In this section, we describe the results of *Dual*-Alg. obtained during our computational test phase. Our algorithm was implemented in C++ and the mathematical models were solved using the IBM ILOG CPLEX 12.10 solver. The nonlinear model is solved by using CPLEX with its default parameters' values which require also the definition of an optimality target [22]. In particular, when solving

Liu's model, the only optimality target that guarantees a successful solution of the model is the "first-order optimality" target (all the other options return wrongly either "Infeasible Status" or "Unbounded Status").

All tests were performed in single thread mode on a workstation with an Intel Xeon CPU E5-2650 v3 @ 2.30 GHz and 128 GB of RAM.
The computational tests are carried out on two different data sets:

Data set 1: This set contains the benchmark instances proposed in [13]. The data set is composed of 330 instances and it contains supply-surplus instances (that is, instances for which $\sum_{i\in I}\overline{s}_i > \sum_{j\in J}\overline{d}_j$) of scenarios having a size ranging from 5×5 to 100×100, and having supply and demands intervals widths equal to 5, 10 and 20. Each scenario is composed of 10 instances generated in a random way, such that the extremes of the supply and demand intervals assume integer values. The costs are generated so that the transportation paradox does not arise. To this end, we applied the following theorem [16]:

Theorem 6.1. *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:*

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

In order to satisfy the conditions of the theorem, cost values can be randomly generated in $[\frac{k}{2},k]$, for any $k>0$. Instances in this data set were generated by setting $k=30$, therefore the resulting costs vary in the range $[15,30]$.

Data set 2: We generated a new set of random supply-surplus instances with wider interval width and a different way of generating costs. Specifically, the size of the new scenarios ranges from 10×10 to 100×100. The supply and demand intervals widths are equal to 10, 20 and 30. Each scenario is composed of 10 instances generated in a random way, according to the previous parameters, for a total of 300 instances. Again, instances are generated such that the extremes of the supply and demand intervals assume integer values. The costs c_{ij} are generated in a prespecified range of values $[c_{min},c_{max}]$ according to the following three steps procedure:

Step 1: Generation of the minimum cost values
Randomly generate values α_p, $p=1,2,\ldots,m$ such that: (i) $\alpha_p\geq c_{min}$, $p=1,\ldots,m$, and (ii) $\alpha_p+\alpha_q\leq c_{max}$, $p=1,\ldots,m$, $q=1,\ldots,m$.

Step 2: Assignment of the minimum cost values to rows and columns of the cost matrix

Randomly associate the generated values α_p, $p = 1, \ldots, m$ to rows and columns of the cost matrix such that exactly one value is assigned to row i and exactly one values is associated with each column j of the cost matrix. Let us define $\alpha^{i\cdot}$ and $\alpha^{\cdot j}$, the values associated with row i and column j, respectively.

Step 3: Generation of the remaining costs

For each element in the cost matrix which was not associated to any value in the previous step, randomly generate a cost c_{ij} such that $\max\{\alpha^{i\cdot}, \alpha^{\cdot j}\} \leq c_{ij} \leq \alpha^{i\cdot} + \alpha^{\cdot j}$.

The above procedure ensures that the generated costs satisfy the conditions stated in Theorem 6.1, and therefore they are immune against the transportation paradox. The generated instances are such that $c_{min} = 10$ and $c_{max} = 50$. All the instances are available upon request to the authors.

Table 2 and Table 3 show the results of our experiments on data set 1 and data set 2, respectively. Each row of the tables shows average values of the results obtained on 10 instances. The first two columns report the number of nodes (*Size*) and the width of the intervals (*Width*). The remaining eight columns report the objective value (*Obj*) and the computational time in "seconds" (*Time*) of our formulation (MIP), of the non linear model presented by Liu [9] (*NLP-Liu*), of the algorithm proposed in [13] (δ-*Alg.*), and of our dual algorithm (*Dual-Alg.*).

The non-linear model proposed by Liu [9] has a non-linear objective function and linear constraints. It builds a scenario by using supplies and demands variables defined over the non-negative intervals and the dual variables derived from the classical transportation problem. The NLP-Liu formulation, used for the computational test, is described in details in the appendix of [13] . The δ-Alg. proposed in [13] is a constructive algorithm that builds a feasible scenario by starting from an infeasible one. It takes as input an IITP instance and it returns as output the built scenario and the solution value of the corresponding transportation problem. In the case of a supply-surplus instance, δ-Alg. starts by considering a scenario where each demand is fixed at its upper bound and each supply at its lower bound. Iteratively, the algorithm selects a supply and increases its value by a given predefined quantity δ. The procedure stops when the total amount of the supply is equal to the total amount of the demand.

The maximum number of iterations (IT_{max}) of *Dual*-Alg. was set to 20, while we set $\delta = 1$ for the δ-Alg. A time limit of three hours is fixed for both NLP-Liu and MIP. Whenever NLP-Liu reaches the time limit or stops with the NumBest

Size	Width	MIP		NLP-Liu		δ-Alg.		*Dual*-Alg.	
		Obj	**Time**	**Obj**	**Time**	**Obj**	**Time**	**Obj**	**Time**
5x5	5	**3,084.00**	0.04	**3,084.00**	0.01	3,070.10	0.01	**3,084.00**	0.04
10x10		**6,025.90**	0.58	**6,025.90**	0.01	5,993.20	0.03	**6,025.90**	0.05
20x20		**13,951.60**	132.07	13,951.50	0.16	13,922.00	0.10	**13,951.60**	0.14
30x30				21,854.00	1.89	21,813.10	0.22	**21,854.10**	0.32
40x40				**37,599.40**	8.09	37,560.40	0.51	37,599.00	0.58
50x50				**52,674.00**	31.69	52,646.80	1.01	**52,674.00**	0.92
60x60				**63,786.17***	3,308.37	63,745.60	1.61	63,785.50	1.45
70x70				**80,494.90**	161.99	80,472.00	3.52	**80,494.90**	2.01
80x80				**115,106.90***	1,478.80	115,081.30	5.50	**115,106.90**	2.48
90x90				**140,545.80**	1,028.75	140,535.20	7.08	**140,545.80**	3.25
100x100				159,866.70*	4,742.66	159,857.50	9.23	**159,867.10**	4.25
5x5	10	**3,700.20**	0.07	**3,700.20**	0.01	3,674.70	0.02	**3,700.20**	0.04
10x10		**6,946.80**	0.26	**6,946.80**	0.01	6,908.70	0.03	**6,946.80**	0.05
20x20		**14,702.20**	150.06	14,700.40	0.14	14,632.00	0.10	**14,702.20**	0.16
30x30				23,903.80	2.89	23,821.80	0.43	**23,904.10**	0.34
40x40				**39,963.11***	224.75	39,861.00	0.93	39,962.50	0.62
50x50				55,563.18*	1,588.82	55,526.20	1.88	**55,590.50**	0.94
60x60				**69,584.50***	1,451.09	69,511.70	2.46	69,583.70	1.41
70x70				87,381.41*	3,436.85	87,335.70	5.32	**87,384.00**	2.08
80x80				**125,948.50***	1,574.59	125,924.20	8.66	**125,948.50**	2.52
90x90				**138,773.20**	1,055.02	138,760.90	11.87	**138,773.20**	3.19
100x100				**160,993.70***	5,822.09	160,978.90	12.36	**160,993.70**	4.04
5x5	20	**4,519.40**	0.11	4,515.50	0.01	4,461.50	0.04	**4,519.40**	0.04
10x10		**9,571.20**	1.20	9,570.30	0.02	9,440.00	0.08	**9,571.20**	0.05
20x20		**18,086.90**	358.01	18,085.50	0.18	17,987.20	0.28	**18,086.90**	0.15
30x30				28,556.58*	39.60	28,453.70	0.76	**28,563.10**	0.34
40x40				47,460.40	6.21	47,312.60	2.14	**47,462.20**	0.64
50x50				61,423.50	72.21	61,287.70	3.74	**61,423.90**	0.92
60x60				**74,371.23***	2,264.45	74,275.40	4.66	74,369.30	1.38
70x70				**98,027.45***	2,388.65	97,958.40	10.03	98,025.80	1.90
80x80				**138,745.80***	4,701.01	138,687.60	19.09	138,744.30	2.70
90x90				**160,476.70***	4,975.37	160,447.40	26.17	**160,476.70**	2.78
100x100				**183,589.10***	3,256.76	183,568.60	33.59	183,588.40	3.61
Avg					*1,321.91*		*5.26*		*1.38*
#Best				*20*		*0*		*25*	

Table 2: *Computational results on data set 1 instances.*

CPLEX status on at least one of the 10 instances of the scenario, the related solution value is marked with a "$*$" symbol. Rows with no values in the MIP columns correspond to scenarios where the solver has not been able to find the optimal solution for all the 10 instances of the scenario within the time limit. The best value, on each row, is reported in bold. The last two rows report the total average computational time (*Avg*) and the number of times the best solution was found (*#Best*) by NLP-Liu, δ-Alg. and *Dual*-Alg., respectively. From the results of Table 2, it is evident that *Dual*-Alg. is the most effective and fastest algorithm. Indeed, it finds the best solution in 25 out of 33 scenarios, and it always finds the optimal

solution, where this value is known (i.e., on the instances up to 20x20). NLP-Liu finds the best solution on 20 out of 33 scenarios, while δ-Alg. is the least effective algorithm with zero best solution found. When comparing the computational time, *Dual*-Alg. is the fastest one with an average computational time equal to 1.38 seconds and a peak equal to 4.25 seconds. It is worth noting that for the instances up to 50x50 the time of *Dual*-Alg. is lower than one second making it particularly suitable to quickly find high quality solutions. δ-Alg. is slower than *Dual*-Alg with an average time equal to 5.26 seconds and a peak equal to 33.59 seconds. Finally, the slowest approach is NLP-Liu which, on average, requires 1,321 seconds and, in the worst case, it reaches the time limit of three hours.

In Table 3 the results of the algorithms, on the data set 2, are reported. The computational times are approximately doubled for all the instances, revealing that the instances in this data set are more challenging than those on data set 1. However, the trend of the results, observed in Table 2, is confirmed also for these instances. Indeed, *Dual*-Alg. finds the best solution on 27 out of 30 scenarios, while NLP-Liu finds the best solution on 12 scenarios. Again δ-Alg. never finds the best solution. With 2.21 seconds, on average, *Dual*-Alg. remains the fastest algorithm and the gap from the computational time of the δ-Alg. increases on these instances. For NLP-Liu the computational time significantly increases, with respect to the computational time required to solve instances of the data set 1, with an average of 2,772 seconds. Summarizing, the results of Table 2 and 3 show that *Dual*-Alg. strongly outperforms both state of the art algorithms with particular emphasis on quality with respect to δ-alg (*Dual*-Alg. reaches worse solutions on all scenarios) and on computational time with respect to NLP-Liu (on the average NLP-Liu is three order of magnitude slower).

7. Conclusions

In this paper we focused on finding the best and worst values of the optimal cost for a transportation problem when all the supplies and demands range over an interval and the cost matrix is such that the transportation paradox does not arise (referred to as the Immune Interval Transportation Problem -IITP-). We specifically focused on determining the worst optimal cost, since finding the best one can be done in polynomial time. We looked at the case of a transportation problem with inequality constraints for supply and equality constraints for demand. Based on existing results on convex optimization, we introduced the definition of quasi-extreme scenarios as those scenarios whose corresponding optimal transportation cost is a candidate to be the worst optimal cost among all the data re-

Size	Width	MIP		NLP-Liu		δ-Alg.		*Dual*-Alg.	
		Obj	**Time**	**Obj**	**Time**	**Obj**	**Time**	**Obj**	**Time**
10x10	10	**3,628.80**	0.28	3,628.50	0.01	3,596.20	0.04	**3,628.80**	0.07
20x20		**7,090.70**	13.75	**7,090.70***	3.68	6,945.00	0.16	**7,090.70**	0.22
30x30		**10,814.70**	760.04	10,814.20	3.46	10,512.80	0.45	**10,814.70**	0.47
40x40				**14,132.90***	180.63	13,803.70	1.05	**14,132.90**	0.81
50x50				18,429.10*	757.90	18,216.70	2.24	**18,429.20**	1.23
60x60				21,675.90*	2,383.19	20,889.00	4.13	**21,676.20**	1.82
70x70				**26,114.10***	1,285.22	25,572.40	6.80	**26,114.10**	2.82
80x80				**28,629.80***	5,296.98	27,978.80	10.86	**28,629.80**	3.76
90x90				32,529.36*	7,179.82	31,576.50	16.71	**32,529.50**	4.90
100x100				**36,523.70***	7,736.79	35,735.30	18.46	**36,523.70**	6.10
10x10	20	**5,227.40**	0.32	**5,227.40**	0.02	5,151.80	0.07	**5,227.40**	0.07
20x20		**10,518.50**	13.15	10,516.60	0.29	10,243.90	0.32	**10,518.50**	0.22
30x30		**15,887.20**	587.78	**15,887.20**	30.22	15,484.10	0.96	**15,887.20**	0.46
40x40				21,499.34*	188.10	20,777.80	2.25	**21,499.40**	0.78
50x50				26,375.10*	1,078.46	25,705.00	4.66	**26,375.20**	1.27
60x60				32,667.95*	2,293.90	31,853.50	8.41	**32,673.70**	1.88
70x70				**37,589.87***	5,590.16	36,696.50	14.34	37,589.70	2.66
80x80				41,900.76*	5,812.89	41,042.00	23.33	**41,943.60**	3.50
90x90				47,957.96*	6,032.02	46,827.40	35.54	**47,958.00**	5.03
100x100				**52,658.70***	7,402.78	50,628.10	42.50	**52,658.70**	6.34
10x10	30	**6,818.90**	0.39	6,806.80	0.03	6,696.90	0.10	6,818.60	0.06
20x20		**13,901.60**	13.15	**13,901.60**	0.14	13,739.60	0.49	**13,901.60**	0.21
30x30		**20,762.50**	468.08	20,761.90	1.44	20,068.70	1.41	**20,762.50**	0.47
40x40				28,197.40	14.25	27,686.50	3.33	**28,198.40**	0.83
50x50				**34,584.30***	3,000.16	33,466.60	7.12	**34,584.30**	1.24
60x60				41,872.59*	3,733.95	40,857.30	13.26	**41,873.80**	1.83
70x70				50,540.46*	4,508.15	49,301.20	21.70	**50,552.10**	2.64
80x80				**56,417.85***	5,743.83	55,196.80	35.50	56,415.70	3.72
90x90				63,630.43*	5,314.93	60,740.40	54.45	**63,630.60**	4.82
100x100				71,549.89*	7,614.09	70,030.60	72.80	**71,549.90**	6.14
Avg					*2,772.92*		*13.45*		*2.21*
#Best				*12*		*0*		*27*	

Table 3: *Computational results on data set 2 instances.*

alizations. This characterization forms the basis for a new mathematical mixed integer model to solve the problem optimally and for a new heuristic algorithm that strongly outperforms the current state of the art procedures.

Acknowledgements

The authors thank the three anonymous referees whose comments and suggestions helped to improve the quality of the paper. This work has been partially supported by "Ministero dell'Istruzione, dell'Università e della Ricerca" (MIUR):

- F. Carrabs, R. Cerulli, C. D'Ambrosio and M. Gentili with the program PRIN 2015, project "SPORT - Smart PORt Terminals", code 2015XAPRKF_005
- F. Della Croce with Award "TESUN-83486178370409, finanziamento dipartimenti di eccellenza CAP. 1694 TIT. 232 ART. 6"
- C. D'Ambrosio with the program PON "Ricerca e Innovazione" 2014-2020, Azione 1.2 "Mobilità dei Ricercatori" (AIM "Attraction and International Mobility"-LINEA 1), POC R&I 2014-2020.

References

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

[2] Elif Garajová, Milan Hladík, and Miroslav Rada. Interval linear programming under transformations: optimal solutions and optimal value range. *Central European Journal of Operations Research*, 27(3):601–614, 2019.

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

[4] Milan Hladík. Optimal value range in interval linear programming. *Fuzzy Optimization and Decision Making*, 8(3):283–294, 2009.

[5] John W Chinneck and K Ramadan. Linear programming with interval coefficients. *Journal of the operational research society*, 51(2):209–220, 2000.

[6] Mohsen Mohammadi and Monica Gentili. Bounds on the worst optimal value in interval linear programming. *Soft Computing*, 23(21):11055–11061, 2019.

[7] Mohsen Mohammadi and Monica Gentili. The outcome range problem. *arXiv preprint arXiv:1910.05913*, 2019.

[8] Stefan Chanas, Miguel Delgado, José Luis Verdegay, and María Amparo Vila. Interval and fuzzy extensions of classical transportation problems. *Transportation Planning and Technology*, 17(2):203–218, 1993.

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

[10] Zeinul Abdeen Mohamed Silmi Juman and Mohammad Abdul 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.

[11] 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.

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

[13] Ciriaco D'Ambrosio, Monica Gentili, and Raffaele Cerulli. The optimal value range problem for the interval (immune) transportation problem. *Omega*, 95:102059, 2020.

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

[15] Sverre Storøy. The transportation paradox revisited. *Optimization Online http://www. optimization-online. org/DB_FILE/2007/09/1763. pdf*, 2007.

[16] 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.

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

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

[19] Fabio Tardella. On a class of functions attaining their maximum at the vertices of a polyhedron. *Discrete applied mathematics*, 22(2):191–195, 1988.

[20] Stefan M Stefanov. *Separable programming: Theory and methods*, volume 53. Springer Science & Business Media, 2013.

[21] Stefan M Stefanov. Characterization of the optimal solution of the convex separable continuous knapsack problem and related problems. *Journal of Information and Optimization Sciences*, pages 1–16, 2019.

[22] `https://www.ibm.com/support/knowledgecenter/SSSA5P_12.10.0/ilog.odms.cplex.help/CPLEX/Parameters/topics/OptimalityTarget.html`.

Appendix

Properties of the demand-surplus version of IITP

Results similar to those shown in section 3 can be obtained for the demand-surplus version of IITP. The following theorem, proved in [13], states that the optimum values $\overline{z}$ of the demand-surplus version of IITP is achieved on a scenario such that all the supplies s_i are equal to their upper value $\overline{s}_i$:

Theorem 7.1. *[13] Given a demand-surplus version of IITP then the optimal value $\overline{z}$ of $z(s,d)$ is the solution of the following optimization problem:*

$$\overline{z} = \{\max z(\overline{s},d) : \underline{d} \leq d \leq \overline{d}; \sum_{j\in J} d_j \leq \sum_{i\in I} \overline{s}_i\} \tag{66}$$

Let us denote by P_D the polyhedron described by the constraints in (66), that is:

$$P_D = \{d \in \mathbb{R}^n : \underline{d} \leq d \leq \overline{d}; \sum_{j\in J} d_j \leq \sum_{i\in I} \overline{s}_i\}$$

We can distinguish three different cases of problem (66) as stated in the following three lemmas.

Lemma 7.2. *Given a demand-surplus version of IITP, if $\sum_{j\in J} \underline{d}_j > \sum_{i\in I} \overline{s}_i$, then $P_D = \emptyset$.*

Proof. Under the given assumption any scenario $(\overline{s},d)$ is such that $F(\overline{s},d) = \emptyset$. □

Lemma 7.3. *Given a demand-surplus version of IITP, if $\sum_{j\in J} \underline{d}_j = \sum_{i\in I} \overline{s}_i$, then $\overline{z} = z(\overline{s},\underline{d},)$.*

Proof. Under the given assumption scenario $(\overline{s},\underline{d})$ is the only feasible scenario, that is, $SD = \{(\overline{s},\underline{d})\}$, and hence the claim follows. □

Lemma 7.4. *Given a demand-surplus version of IITP, if* $\sum_{j\in J}\underline{d}_j < \sum_{i\in I}\overline{s}_i$*, then the optimal value* $\overline{z}$ *of* $z(s,d)$ *is the solution of the following optimization problem:*

$$\overline{z} = \{\max z(\overline{s},d) : \underline{d} \leq d \leq \overline{d}; \sum_{j\in J} d_j = \sum_{i\in I}\overline{s}_i\} \tag{67}$$

The proof of this lemma is omitted since it can be derived by using similar arguments used for the supply-surplus version of the problem.
As for the supply-surplus version of IITP, vertices of polyhedron P_D, can be associated to demand-quasi-extreme scenarios. Let us formally introduce the definition of demand-quasi-extreme scenarios.

Definition 7. *A scenario* (s,d) *is a* ***demand-quasi-extreme*** *scenario if* $s_i \in \{\underline{s}_i, \overline{s}_i\}$*,* $\forall i \in I$*, and there exists* ***at most*** *one index* j_h *such that* $\underline{d}_h < d_{j_h} < \overline{d}_h$ *and* $d_j \in \{\underline{d}_j, \overline{d}_j\}$*,* $\forall j \in J\backslash\{j_h\}$*.*

Demand-quasi-extreme scenarios are those scenarios where a most one among the demands might not be neither at its lower or upper bound, and all the supplies are either at their lower or upper bounds. The following property follows directly from known results in convex optimization [20, 21] and its proof is omitted:

Property 7.5. *Given a demand-surplus version of IITP the following statements hold:*

- *the vertices of the polyhedron* P_D *correspond to supply-quasi-extreme scenarios;*
- *if* $\underline{s}$*,* $\overline{s}$*,* $\underline{d}$ *and* $\overline{d}$ *have integer components, then the vertices of polyhedron* P_D *are integer.*