EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 1, Article Number 5593
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Handling Degeneracy in Multi-objective Transportation Problem: An Efficient Branch-and-Bound Algorithm

Zineb Mezali[1,*], Mohamed El-Amine Chergui[1]

[1] *Department of Operations Research, Faculty of Mathematics, USTHB University , Algiers, Algeria*

Abstract. In today's competitive market, practically all models are designed to address transportation problems involving multiple, often conflicting, objectives, such as minimizing transportation cost, minimizing transportation time, maximizing reliability, minimizing environmental impact, maximizing social equity, and minimizing product deterioration. In this respect, this paper proposes a method to optimize a multi-objective transportation problem (MOTP). We developed a branch-and-bound-based algorithm coupled with a classical transportation method to find the non-dominated points in the criteria space, in a finite number of steps. The algorithm utilizes reduced costs of all the criteria cost matrices to define the promising regions that may contain non-dominated points. This algorithm is strengthened by efficient bounds allowing us to prune a large number of nodes in the search tree and hence eliminate many dominated points. Efficient bounds further enhance the algorithm by pruning a large number of search tree nodes and eliminating dominated points. The suggested approach effectively addresses the non-degenerate case as well as the degenerate case, the latter of which, to our knowledge, has not been discussed in prior studies on MOTP. To handle degeneracy, we integrated the improved N-tree method into our approach. The effectiveness of our algorithm was assessed by comparing it to Isermann's method for the non-degenerate case, where it was noticed that our approach gives better results. Additionally, computational experiments confirmed its efficiency in handling degenerate cases. Two numerical examples are presented to illustrate the step-by-step application of the proposed method.

2020 Mathematics Subject Classifications: 90C08,90C29

Key Words and Phrases: Transportation problem, multi-objective optimization, non-dominated point, branch-and-bound

1. Introduction

The transportation problem (TP) is a fundamental optimization problem that has been extensively studied since its formulation by Hitchcock in 1941 [17]. It involves determining the optimal allocation of goods from various sources to multiple destinations to minimize transportation costs. Exact algorithms, such as the simplex method [9], and stepping

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v18i1.5593

Email addresses: mezalizineb@gmail.com (Z. Mezali), mchergui@usthb.dz (M.E.A.Chergui)

stone algorithm [8], as well as heuristics ([31],[16], [20], [21], [23], [2]) have been developed to solve the TP efficiently.

Multi-objective transportation problem (MOTP) considers multiple conflicting objectives, such as minimizing transportation costs, delivery times, and maximizing profits.

Various approaches, including exact methods and heuristic algorithms, have been proposed to identify efficient solutions for the MOTP. These approaches can be classified into three main categories. The first category comprises methods designed to identify all efficient solutions, including those based on linear programming ([35], [10], [5], [19], [34], [11]) and dynamic programming ([13], [12]). The second category includes methods that focus on finding a single efficient solution or compromise solution, such as goal programming ([6], [25], [37], [27], [30]), interactive methods [32], lexicographic optimization ([36], [28]), the minimize distance method ([1], [22]), and the decomposition approach ([3],[4]). Finally, the third category encompasses methods that compute a finite approximation of the non-dominated set, such as heuristic approaches ([14], [29], [40], [41], [26], [39]).

While the existing literature on MOTP offers a variety of approaches, a common limitation is the lack of specific strategies to handle degeneracy that can pose significant challenges in identifying non-dominated points. To overcome this limitation, we propose a novel exact method based on the branch-and-bound principle that incorporates a specialized procedure to handle degeneracy. In addition, the branch-and-bound method effectively explores the criteria space by eliminating states that cannot lead to non-dominated points. This is achieved through fathoming rules.

The structure of this article is as follows: in section 2, we provide some definitions and notations, section 3 is devoted to a brief description of the method and a presentation of the algorithm that we named MOTP-Algorithm, which is followed by a numerical example to understand the different steps of the proposed method to solve the MOTP problem. Section 4 is about some theoretical results and proofs. Moreover, in section 5, our algorithm has been tested on a set of non-degenerate and degenerate instances, and then we terminate with a conclusion in section 6.

2. Definitions and notations

We consider the following Mutli-Objective Transportation Problem (MOTP):

$$\begin{cases} min\ Z_k(X) = \sum_{i\in I}\sum_{j\in J} c_{ij}^k x_{ij},\ \forall k \in K & (1) \\ \sum_{j\in J} x_{ij} = a_i,\ \forall i \in I & (2) \\ \sum_{i\in I} x_{ij} = d_j,\ \forall j \in J & (3) \\ x_{ij} \geq 0,\ \forall i \in I,\ j \in J & (4) \end{cases}$$

Let $I = \{1, ..., m\}$, $J = \{1, ..., n\}$, and $K = \{1, ..., r\}$ represent the sets of sources, destinations, and objectives, respectively. The cost matrix c^k, for each objective k is defined as $c^k = (c^k_{ij})\ \forall i \in I, \forall j \in J$. We assume that the supply at source i is $a_i \geq 0$ and the demand at destination j is $d_j \geq 0$. Additionally, the total supply must equal the total demand ($\sum_{i\in I} a_i = \sum_{j\in J} d_j$), which is necessary and sufficient for a transportation problem to have a feasible solution.

Let D be the set of feasible solutions of the MOTP problem, which is assumed to be nonempty and compact, and $Z(D)$ its image set in the criteria space $\mathbb{R}^r$.

Definition 1. *A basic feasible solution to MOTP is a feasible solution that contains no more than m + n – 1 positive basic variables.*

Let $X = (x_{ij}), i \in I, j \in J$ be a basic feasible solution to MOTP, the image of X in the criteria space is denoted by $Z(X)$ such that $Z(X) = \big(Z_1(X), \dots, Z_r(X)\big)$.

Definition 2. *A basic feasible solution $X = (x_{ij}), i \in I, j \in J$, to MOTP is efficient if there is no other basic feasible solution $Y = (y_{ij})$ of MOTP such that: $Z_k(Y) \leq Z_k(X)$ for all criteria $k \in K$, and $Z_k(Y) < Z_k(X)$ for at least one criterion $k \in K$.*

Definition 3. *An efficient solution $X = (x_{ij}), i \in I, j \in J$ is said alternative if there exists an efficient solution $Y = (y_{ij})$ such that: $Z(Y) = Z(X)$.*

Definition 4. *The image of an efficient solution in the criteria space is called a non-dominated point.*

Definition 5. *A non-degenerate solution is a basic feasible solution that has exactly $m + n - 1$ positive basic variables.*

Definition 6. *A degenerate solution is a basic feasible solution that contains less than $m + n - 1$ positive basic variables. In other words, one or more basic variables are zero-valued.*

Throughout this present paper, SND refers to the non-dominated points set in $Z(D)$ and D_E the efficient solutions set in D.
Let $(P0)$ be the transportation problem with $c^0 = (c^0_{ij}), i \in I, j \in J$, being the sum of all criteria cost matrices:

$$(P0)\begin{cases} \min \sum_{i\in I}\sum_{j\in J} c^0_{ij} x_{ij} \\ (x_{ij})_{i\in I, j\in J} \in D \end{cases} \tag{5}$$

The generation of the non-dominated points of MOTP is based on the resolution of the mono-criterion transport problem $(P0)$ with the criterion c^0, knowing that all the operations performed on the cost matrix of $(P0)$ are also applied simultaneously to cost matrices of all the other criteria. A slave program is performed to determine whether a basic feasible solution of $(P0)$ obtained at step l of our MOTP problem's resolution approach, $X_l = (x^l_{ij}), i \in I, j \in J$, is efficient. It is given as follows:

$$(P_{X_l})\begin{cases} \max\ \sum_{k\in K} e_k \\ \sum_{i\in I}\sum_{j\in J} c^k_{ij} x_{ij} + e_k = \sum_{i\in I}\sum_{j\in J} c^k_{ij} x^l_{ij} \quad \forall k \in K \\ (x_{ij})_{i\in I, j\in J} \in D \\ e_k \geq 0\ \forall k \in K \end{cases} \tag{6}$$

The optimal objective value of (P_{X_l}) is 0 if and only if X_l is efficient (refer [7]).
We associate the following parameters with any basic feasible solution $X_l = (x^l_{ij}), i \in I, j \in J$:
B_l: the set of indices of basic variables.
N_l: the set of indices of non-basic variables.
H_l: the set of the descent directions of the criteria at X_l, except the criterion c^0 which is initially at the minimum.
$H^1_l = \{(i, j) \in N_l \mid \hat{c}^0_{ij} \geq 0\}$.
$H^2_l = \{(i, j) \in N_l \mid \exists k \in K \mid \hat{c}^k_{ij} < 0\}$.
$H^3_l = \{(i, j) \in (H^1_l \cap H^2_l) \mid \exists (s, t) \in (H^1_l \cap H^2_l) \mid \sum_{k\in K} \hat{c}^k_{st} x_{st} \leq \sum_{k\in K} \hat{c}^k_{ij} x_{ij} \ \forall k \in K \wedge \exists k \in K \mid \sum_{k\in K} \hat{c}^k_{st} x_{st} < \sum_{k\in K} \hat{c}^k_{ij} x_{ij}\} \cup \{(i, j) \in (H^1_l \cap H^2_l) \mid \exists (s, t) \in (H^1_l \cap H^2_l) \mid \sum_{k\in K} \hat{c}^k_{st} x_{st} = \sum_{k\in K} \hat{c}^k_{ij} x_{ij} \ \forall k \in K\}$.
$H_l = (H^1_l \cap H^2_l) \setminus H^3_l$.
$\hat{c}^k_{ij}$: the reduced costs of the non-basic variables determined using the following equations:

$$\hat{c}^k_{ij} = c^k_{ij} - (u^k_i + v^k_j), \ \forall (i, j) \in N_l, \ \forall k \in \{0\} \cup K \tag{7}$$

(u^k_i, v^k_j): the dual variables of the constraints (2) and (3) respectively for the basis B_l so that:

$$(u^k_i + v^k_j) = c^k_{ij}, \ \forall (i, j) \in B_l, \ \forall k \in \{0\} \cup K \tag{8}$$

The system of equations (8) is solved by substituting : $u^k_1 = 0, \ \forall k \in \{0\} \cup K$.

MOTP is represented by a table (refer Table 1) having m rows and n columns. Each cell (i, j) at the intersection of row i and column j contains a vector of the criteria c^k_{ij}, the value of the variable x_{ij}, referred to as allocation, and a vector of the reduced costs $\hat{c}^k_{ij}$. A cell with a positive x_{ij} value is called an occupied cell, while a cell with a zero x_{ij} value is called an unoccupied cell. Furthermore, the MOTP table is delimited by a column containing the supplies ai, a row containing the demands dj, and the last row and column containing the vector of u^k_i values and v^k_j values respectively, for all $k \in \{0\} \cup K$.

	d_1	$\dots$	d_n	$(u_i^0, ..., u_i^r)$
a_1	$(c_{11}^0, ..., c_{11}^r)$ x_{11} $(\hat{c}_{11}^0, ..., \hat{c}_{11}^r)$	$\dots$	$(c_{1n}^0, ..., c_{1n}^r)$ x_{1n} $(\hat{c}_{1n}^0, ..., \hat{c}_{1n}^r)$	$(u_1^0, ..., u_1^r)$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
a_m	$(c_{m1}^0 \dots c_{m1}^r)$ x_{m1} $(\hat{c}_{m1}^0, ..., \hat{c}_{m1}^r)$	$\dots$	$(c_{mn}^0, ..., c_{mn}^r)$ x_{mn} $(\hat{c}_{mn}^0, ..., \hat{c}_{mn}^r)$	$(u_m^0, ..., u_m^r)$
$\begin{pmatrix} v_j^0 \\ \vdots \\ v_j^r \end{pmatrix}$	$\begin{pmatrix} v_1^0 \\ \vdots \\ v_1^r \end{pmatrix}$	$\dots$	$\begin{pmatrix} v_n^0 \\ \vdots \\ v_n^r \end{pmatrix}$	

Table 1: MOTP table

Using the vector notations:
$x = \left(x_{11}, x_{12}, \dots x_{1n}, x_{21}, \dots, x_{2n}, \dots, x_{m1}, \dots, x_{mn}\right)^T,$
$$c = \begin{pmatrix} c_{11}^1 & c_{12}^1 & \dots & c_{1n}^1 & c_{21}^1 & \dots & c_{2n}^1 & \dots & c_{m1}^1 & \dots c_{mn}^1 \\ c_{11}^2 & c_{12}^2 & \dots & c_{1n}^2 & c_{21}^2 & \dots & c_{2n}^2 & \dots & c_{m1}^2 & \dots c_{mn}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ c_{11}^r & c_{12}^r & \dots & c_{1n}^r & c_{21}^r & \dots & c_{2n}^r & \dots & c_{m1}^r & \dots c_{mn}^r \end{pmatrix},$$
$b = \left(a_1, a_2, \dots a_m, d_1, d_2, \dots d_n\right)^T$
MOTP can be written as follows:

$$\begin{cases} min\ cx \\ Ax = b \\ x \geq 0 \end{cases} \tag{9}$$

A^{B_l}: regular matrix of columns of A corresponding to basic variables.
A^{N_l}: matrix of columns of A corresponding to non-basic variables.
$P(B_l)$: pivot tableau of the form shown in Table 2, where W denotes the matrix $(A^{B_l})^{-1}A^{N_l}$ and x symbolizes the basic feasible solution, i.e., $x = (A^{B_l})^{-1}b$.

B_l	N_l	x
	W	

Table 2: pivot tableau $P(B_l)$

Definition 7. *Let x be a degenerate solution, the column $W.j = (A^{B_l})^{-1}A^{N_l}.j$ of the pivot tableau $P(B_l)$ is called a transition column if it has positive entries in the rows where x equals zero and has at least one positive entry in a row where x is non-zero and therefore allows the degenerate solution to be left by a pivot step.*

Definition 8. *A pivot tableau with at least one transition column is called a transition tableau.*

3. Method and Algorithm

This section is devoted to the illustration of the ideas developed in the framework of the exact method that we propose for the MOTP problem, exploiting the particularities of the classical transport problem.

3.1. Principle of the method

A branch-and-bound-based method is described to find the set SND of non-dominated points of the MOTP problem. At the beginning of the search tree, the algorithm starts with the optimization of the cost matrix criterion c^0, which corresponds to solving the problem $(P0)$, through the Modified Distribution Stepping-Stone (MODI) method, also known as the method of multipliers (refer [38]), to obtain an optimal solution X_0 for $(P0)$, which is an efficient solution for the MOTP problem. The same MODI operations are alson applied to all cost matrix criteria to evaluate them at X_0 , and the set SND is updated. Using the reduced cost matrices, we define the set H_0 of the descent directions of the criteria at X_0, except the criterion $k = 0$, which must increase, knowing that it was initially at the minimum. In this step, $|H_0|$ nodes are created. A basic feasible solution X_l is generated at each node l of the search tree by introducing a non-basic variable x_{ij} into the current basis such that $(i, j) \in H_f$ and f is the parent node of node l, using the transportation problem's pivoting procedure (refer [38]). If at any node l, X_l is degenerate, we solve the degeneracy as described in section 3.2 below. Finally, the efficiency test stated in (6) is used, the SND set is then updated, and the process is repeated until all nodes that have been created are fathomed.

3.2. Solving degeneracy in MOTP problem

In the case of a single objective transportation problem with m origins and n destinations, if a basic feasible solution X is degenerate, the problem is said to be a degenerate transportation problem. To remove degeneracy, we utilize a fictitious quantity e, this quantity is assigned to one or more unoccupied cells with the minimum transportation costs, to make $(m + n - 1)$ allocations (refer [33]). However, in the case of the MOTP problem, to find the non-dominated points neighboring a degenerate solution X, we must first solve the degeneracy, i.e., find out how to determine the cells to which e can be assigned to compute the reduced costs and identify the set H_l. A naive approach would be to enumerate all the possible cases to allocate e which is equivalent to treating all the possible bases associated with X and it can be very costly in time and memory space because one degenerate solution has more than one basis (refer [24]), and some of these bases don't lead to a distinct solution. Starting from the pivot tableau of a degenerate solution, the idea suggested to solve the degeneracy in MOTP problem is to use the improved N-tree algorithm (refer [15]) to find the transition tableaux associated with it, i.e.,

tableaux that possess at least one pivot that will yield a distinct solution and thus enables us to leave the degenerate solution. Next, each degenerate solution associated with each of the transition tableaux is put in the MOTP table by replacing the null basic variables of that solution with e, then we proceed as the non-degenerate case.
All the ideas developed above are structured in the following algorithm named MOTP-Algorithm.

3.3. MOTP-Algorithm

The steps of the proposed MOTP-Algorithm are as follows:

Algorithm 1. *MOTP-Algorithm*
Input: *MOTP table.*
Output: *SND (the non-dominated points set), DE (the efficient solutions set)*

Step 0:
Initialize: $DE := \emptyset$, $SND := \emptyset$, $l := 0$.

Step 1:
Find X_0*, the optimal solution to* (P_0)*, and put it in the MOTP table.*
Update: $DE := DE \cup \{X_0\}$, $SND := SND \cup \{Z(X_0)\}$.

Step 2:
If X_l *is degenerate, go to Step 3. Else, go to Step 4.2.*

Step 3:
Apply the improved N-tree Algorithm (refer to [15]) to determine the set T_l *of transition tables associated with* X_l.

Step 4:
For each transition table $T_l^p \in T_l$, $\forall p = 1, ..., |T_l|$:

- *4.1 Let* X_l^p *be the basic feasible solution associated with* T_l^p *and assign the value* e *to each null basic variable, then put the resulting solution in the MOTP table.*
- *4.2 Determine the set* H_l *and create* $|H_l|$ *nodes.*
- *4.3 While there exists a non-fathomed node:*
  - *4.3.1 Set* $l := l + 1$*, use Depth First Search (DFS) to select node* l*, and let* x_{ij} *be a non-basic variable such that* $(i, j) \in H_{l-1}$.
  - *4.3.2 Consider the MOTP table of node* $l - 1$ *and enter the non-basic variable* x_{ij} *into the basis. Let* X_l *be the obtained basic feasible solution at node* l.
  - *4.3.3* ***Efficiency Test of*** X_l***:*** *If there exists* $k \in \{1, ..., r\}$ *such that* $\hat{c}_{ij}^k > 0$, $\forall (i, j) \in N_l$ *(i.e.,* X_l *is the unique optimal solution according to criterion* k*), then* X_l *is efficient. Otherwise, solve the linear problem* (P_{X_l}).
  - *4.3.4 If* X_l *is efficient, go to Step 4.3.5. Else, node* l *is fathomed.*
  - *4.3.5 If* $X_l \in DE$ *or* $H_l = \emptyset$*, node* l *is fathomed.*
  - *4.3.6 If* X_l *has not been visited, update:* $DE := DE \cup \{X_l\}$, $SND := SND \cup \{Z(X_l)\}$.

Step 5:
If X_l is degenerate, go to Step 3. Else, go to Step 4.2.

In the following, we unroll our MOTP-Algorithm in the non-degenerate and degenerate cases.

Example 1. *Let's consider the following non-degenerate instance for the MOTP problem with* $(m,n,r) = (3,3,3)$, $c^1 = \begin{pmatrix} 3 & 10 & 2 \\ 2 & 9 & 4 \\ 10 & 8 & 2 \end{pmatrix}$, $c^2 = \begin{pmatrix} 1 & 7 & 3 \\ 4 & 6 & 6 \\ 7 & 5 & 8 \end{pmatrix}$, $c^3 = \begin{pmatrix} 8 & 7 & 4 \\ 6 & 2 & 1 \\ 8 & 6 & 2 \end{pmatrix}$, $a = \begin{pmatrix} 1 \\ 9 \\ 2 \end{pmatrix}$, $d = (5,3,4)$. *Then, the matrix c^0 is given by :* $c^0 = \begin{pmatrix} 12 & 24 & 9 \\ 12 & 17 & 11 \\ 25 & 19 & 12 \end{pmatrix}$.
Step 0 *:* $D_E = \varnothing, SND = \varnothing$.
At node 0, the optimal solution X_0 according to c^0 is presented in Table 3.

	$d_1 = 5$	$d_2 = 3$	$d_3 = 4$	$(u_i^0, u_i^1, u_i^2, u_i^3)$
$a_1 = 1$	$(12,3,1,8)$ 0 $(2,3,0,-1)$	$(24,10,7,7)$ 0 $(9,3,4,2)$	$(9,2,3,4)$ 1	$(0,0,0,0)$
$a_2 = 9$	$(12,2,4,6)$ 5	$(17,9,6,2)$ 3	$(11,4,6,1)$ 1	$(2,2,3,-3)$
$a_3 = 2$	$(25,10,7,8)$ 0 $(12,10,1,1)$	$(19,8,5,6)$ 0 $(1,1,-3,3)$	$(12,2,8,2)$ 2	$(3,0,5-2)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \\ v_j^3 \end{pmatrix}$	$\begin{pmatrix} 10 \\ 0 \\ 1 \\ 9 \end{pmatrix}$	$\begin{pmatrix} 15 \\ 7 \\ 3 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 9 \\ 2 \\ 3 \\ 4 \end{pmatrix}$	$Z(X_0) = (47,63,45)$ $X_0 = \begin{pmatrix} 0 & 0 & 1 \\ 5 & 3 & 1 \\ 0 & 0 & 2 \end{pmatrix}$

Table 3: Optimal solution X_0 of $(P0)$

X_0 is an efficient solution because it's a unique optimal solution according to c^1.
$D_E := D_E \cup \{X_0\}$, $SND := SND \cup \{Z(X_0)\}$, $H_0 = \{(1,1),(3,2)\}$. *So x_{11} enters the basis and we obtain a basic feasible solution X_1 at node 1, in Table 4.*

X_1 is an efficient solution because it's a unique optimal solution according to c^3.
$D_E := D_E \cup \{X_1\}, SND := SND_1 \cup \{Z(X_1)\}, H_1 = \{(3,2)\}$. *The variable x_{32} enters the basis and we obtain a basic feasible solution X_2 at node 2, in Table 5.*

By solving $P(X_2)$, we find that X_2 is efficient, $D_E := D_E \cup \{X_2\}, SND := SND \cup \{Z(X_2)\}$ *and* $H_2 = \{(3,1)\}$.

So x_{31} enters the basis and we obtain a basic feasible solution X_3 at node 3, in Table 6. By solving $P(X_3)$, we find that X_3 is not efficient, node 3 is then fathomed.
The variable x_{32} enters the basis and we obtain at node 4 a basic feasible solution X_4 in Table 7.

By solving $P(X_4)$, we find that X_4 is efficient, $D_E := D_E \cup \{X_4\}$, $SND := SND \cup \{Z(X_4)\}$ *and* $H_4 = \{(1,1),(3,1)\}$. *If x_{11} enters the basis, we obtain the solution X_5 such that $X_5 = X_2$ at node 5 and since this efficient solution is already visited, node 5 is fathomed. Hence, x_{31} enters the basis and we obtain at node 6 a basic feasible solution X_6 in Table 8.*

	$d_1 = 5$	$d_2 = 3$	$d_3 = 4$	$(u_i^0, u_i^1, u_i^2, u_i^3)$
$a_1 = 1$	$(12, 3, 1, 8)$ 1	$(24, 10, 7, 7)$ 0 $(7, 0, 4, 3)$	$(9, 2, 3, 4)$ 0 $(-2, -3, 0, 1)$	$(0, 0, 0, 0)$
$a_2 = 9$	$(12, 2, 4, 6)$ 4	$(17, 9, 6, 2)$ 3	$(11, 4, 6, 1)$ 2	$(0, -1, 3, -2)$
$a_3 = 2$	$(25, 10, 7, 8)$ 0 $(12, 10, 1, 1)$	$(19, 8, 5, 6)$ 0 $(1, 1, -3, 3)$	$(12, 2, 8, 2)$ 2	$(1, -3, 5 - 1)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \\ v_j^3 \end{pmatrix}$	$\begin{pmatrix} 12 \\ 3 \\ 1 \\ 8 \end{pmatrix}$	$\begin{pmatrix} 17 \\ 10 \\ 3 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 11 \\ 5 \\ 3 \\ 3 \end{pmatrix}$	$Z(X_1) = (50, 63, 44)$ $X_1 = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 3 & 2 \\ 0 & 0 & 2 \end{pmatrix}$

Table 4: Solution X_1 at node 1

	$d_1 = 5$	$d_2 = 3$	$d_3 = 4$	$(u_i^0, u_i^1, u_i^2, u_i^3)$
$a_1 = 1$	$(12, 3, 1, 8)$ 1	$(24, 10, 7, 7)$ 0 $(7, 0, 4, 3)$	$(9, 2, 3, 4)$ 0 $(-2, -3, 0, 1)$	$(0, 0, 0, 0)$
$a_2 = 9$	$(12, 2, 4, 6)$ 4	$(17, 9, 6, 2)$ 1	$(11, 4, 6, 1)$ 4	$(0, -1, 3, -2)$
$a_3 = 2$	$(25, 10, 7, 8)$ 0 $(11, 9, 4, -2)$	$(19, 8, 5, 6)$ 2	$(12, 2, 8, 2)$ 0 $(-1, -1, 3, -3)$	$(2, -2, 2, 2)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \\ v_j^3 \end{pmatrix}$	$\begin{pmatrix} 12 \\ 3 \\ 1 \\ 8 \end{pmatrix}$	$\begin{pmatrix} 17 \\ 10 \\ 3 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 11 \\ 5 \\ 3 \\ 3 \end{pmatrix}$	$Z(X_2) = (52, 57, 50)$ $X_2 = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 4 \\ 0 & 2 & 0 \end{pmatrix}$

Table 5: Solution X_2 at node 2

By solving $P(X_6)$, we find that X_6 is not efficient, node 6 is then fathomed.

The MOTP-Algorithm terminates since all created nodes are fathomed and the set of non-dominated points is: $SND = \{(47, 63, 45), (50, 63, 44), (52, 57, 50), (49, 57, 51)\}$.

The search tree corresponding to example 1 is shown in Figure 1.

	$d_1 = 5$	$d_2 = 3$	$d_3 = 4$	$(u_i^0, u_i^1, u_i^2, u_i^3)$
$a_1 = 1$	$(12, 3, 1, 8)$ 1	$(24, 10, 7, 7)$ 0 $(7, 0, 4, 3)$	$(9, 2, 3, 4)$ 0 $(-2, -3, 0, 1)$	$(0, 0, 0, 0)$
$a_2 = 9$	$(12, 2, 4, 6)$ 2	$(17, 9, 6, 2)$ 3	$(11, 4, 6, 1)$ 4	$(0, -1, 3, -2)$
$a_3 = 2$	$(25, 10, 7, 8)$ 2	$(19, 8, 5, 6)$ 0 $(-11, -9, -4, 2)$	$(12, 2, 8, 2)$ 0 $(-12, -10, -1, -1)$	$(13, 7, 6, 0)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \\ v_j^3 \end{pmatrix}$	$\begin{pmatrix} 12 \\ 3 \\ 1 \\ 8 \end{pmatrix}$	$\begin{pmatrix} 17 \\ 10 \\ 3 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 11 \\ 5 \\ 3 \\ 3 \end{pmatrix}$	$Z(X_3) = (70, 65, 46)$ $X_3 = \begin{pmatrix} 1 & 0 & 0 \\ 2 & 3 & 4 \\ 2 & 0 & 0 \end{pmatrix}$

Table 6: Solution X_3 at node 3

	$d_1 = 5$	$d_2 = 3$	$d_3 = 4$	$(u_i^0, u_i^1, u_i^2, u_i^3)$
$a_1 = 1$	$(12, 3, 1, 8)$ 0 $(2, 3, 0 - 1)$	$(24, 10, 7, 7)$ 0 $(9, 3, 4, 2)$	$(9, 2, 3, 4)$ 1	$(0, 0, 0, 0)$
$a_2 = 9$	$(12, 2, 4, 6)$ 5	$(17, 9, 6, 2)$ 1	$(11, 4, 6, 1)$ 3	$(2, 2, 3, -3)$
$a_3 = 2$	$(25, 10, 7, 8)$ 0 $(11, 9, 4, -2)$	$(19, 8, 5, 6)$ 2	$(12, 2, 8, 2)$ 0 $(-1, -1, 3, -3)$	$(4, 7, 6, 0)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \\ v_j^3 \end{pmatrix}$	$\begin{pmatrix} 10 \\ 0 \\ 1 \\ 9 \end{pmatrix}$	$\begin{pmatrix} 15 \\ 7 \\ 3 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 9 \\ 2 \\ 3 \\ 4 \end{pmatrix}$	$Z(X_4) = (49, 57, 51)$ $X_4 = \begin{pmatrix} 0 & 0 & 1 \\ 5 & 1 & 3 \\ 0 & 2 & 0 \end{pmatrix}$

Table 7: Solution X_4 at node 4

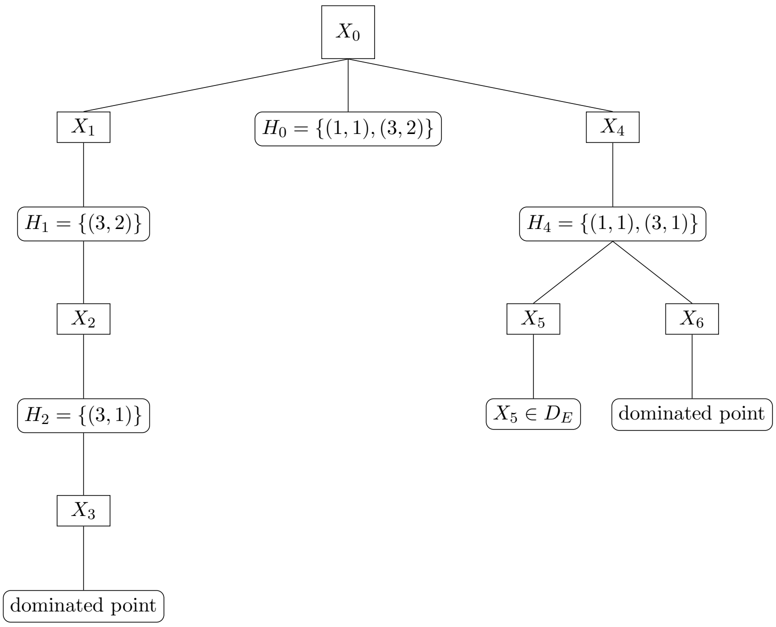

Figure. 1. Search tree 1

	$d_1 = 5$	$d_2 = 3$	$d_3 = 4$	$(u_i^0, u_i^1, u_i^2, u_i^3)$
$a_1 = 1$	$(12, 3, 1, 8)$ 0 $(2, 3, 0 - 1)$	$(24, 10, 7, 7)$ 0 $(9, 3, 4, 2)$	$(9, 2, 3, 4)$ 1	$(0, 0, 0, 0)$
$a_2 = 9$	$(12, 2, 4, 6)$ 3	$(17, 9, 6, 2)$ 3	$(11, 4, 6, 1)$ 3	$(2, 2, 3, -3)$
$a_3 = 2$	$(25, 10, 7, 8)$ 2	$(19, 8, 5, 6)$ 0 $(-11, -9, -4, 2)$	$(12, 2, 8, 2)$ 0 $(-12, -10, -1, -1)$	$(15, 10, 6, -1)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \\ v_j^3 \end{pmatrix}$	$\begin{pmatrix} 10 \\ 0 \\ 1 \\ 9 \end{pmatrix}$	$\begin{pmatrix} 15 \\ 7 \\ 3 \\ 5 \end{pmatrix}$	$\begin{pmatrix} 9 \\ 2 \\ 3 \\ 4 \end{pmatrix}$	$Z(X_6) = (67, 65, 47)$ $X_6 = \begin{pmatrix} 0 & 0 & 1 \\ 3 & 3 & 3 \\ 2 & 0 & 0 \end{pmatrix}$

Table 8: Solution X_6 at node 6

Example 2. *Let consider the following instance for the MOTP problem where degeneracy occurs with* $(m, n, r) = (\,3,\,3,\,2\,)$, $c^1 = \begin{pmatrix} 3 & 3 & 3 \\ 5 & 7 & 2 \\ 6 & 8 & 3 \end{pmatrix}$, $c^2 = \begin{pmatrix} 4 & 1 & 1 \\ 5 & 3 & 10 \\ 6 & 9 & 8 \end{pmatrix}$, $a = \begin{pmatrix} 5 \\ 2 \\ 3 \end{pmatrix}$, $d = (\,2,\,4,\,4\,)$.
Let $c^0 = \begin{pmatrix} 7 & 4 & 4 \\ 10 & 10 & 12 \\ 12 & 17 & 11 \end{pmatrix}$.
Step 0: $D_E = \varnothing$, $SND = \varnothing$.
At node 0 the optimal solution X_0 *is presented in Table 9.*

	$d_1 = 2$	$d_2 = 4$	$d_3 = 4$	(u_i^0, u_i^1, u_i^2)
$a_1 = 5$	$(7, 3, 4)$ 1	$(4, 3, 1)$ 4	$(4, 3, 1)$ 0 $(0, 3, -8)$	$(0, 0, 0)$
$a_2 = 2$	$(10, 5, 5)$ 1	$(10, 7, 3)$ 0 $(4, 2, 1)$	$(12, 2, 10)$ 1	$(3, 2, 1)$
$a_3 = 3$	$(12, 6, 6)$ 0 $(3, 0, 3)$	$(17, 8, 9)$ 0 $(12, 2, 9)$	$(11, 3, 8)$ 3	$(2, 3, -1)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \end{pmatrix}$	$\begin{pmatrix} 7 \\ 3 \\ 4 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 9 \\ 0 \\ 9 \end{pmatrix}$	$Z(X_0) = (31, 47)$ $X_0 = \begin{pmatrix} 1 & 4 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 3 \end{pmatrix}$

Table 9: Optimal solution X_0 of $(P0)$

Check degeneracy: X_0 *is non degenerate.*
By solving $P(X_0)$, *we find that* X_0 *is efficient,* $D_E := D_E \cup \{X_0\}$, $SND := SND \cup \{Z(X_0)\}$ *and*
$H_0 = \{(1, 3)\}$. *So* x_{13} *enters the basis and we get at node 1, the basic feasible solution* $X_1 = \begin{pmatrix} 0 & 4 & 1 \\ 2 & 0 & 0 \\ 0 & 0 & 3 \end{pmatrix}$, $Z(X_1) = (34, 39)$.
By solving $P(X_1)$, *we find that* X_1 *is efficient,* $D_E := D_E \cup \{X_1\}$, $SND := SND \cup$

$\{Z(X_1)\}$.

Check degeneracy: *X_1 is degenerate then using the improved N-tree Algorithm, we find the transition tables shown in Table 10, hence $T_1 = \{T_1^1, T_1^2\}$.*

B_1^1	x_{11}	x_{22}	x_{31}	x_{32}	x_1^1
x_{12}	0	1	0	1	4
x_{13}	1	−1	0	−1	1
x_{21}	1	0	1	0	2
x_{23}	−1	1	−1	0	0
x_{33}	0	0	1	1	3

(a) T_1^1

B_1^2	x_{11}	x_{23}	x_{31}	x_{32}	x_1^2
x_{12}	1	−1	1	1	4
x_{13}	0	1	−1	−1	1
x_{21}	1	0	1	0	2
x_{22}	−1	1	−1	0	0
x_{33}	0	0	1	1	3

(b) T_1^2

Table 10: Resulting transition tables from N-tree Algorithm

In T_1^1, the basic variable $x_{23} = 0$, then we assign the value e to the cell $(2,3)$ and the resulting solution X_1^1 is shown in Table 11.

	$d_1 = 2$	$d_2 = 4$	$d_3 = 4$	(u_i^0, u_i^1, u_i^2)
$a_1 = 5$	$(7,3,4)$ 0 $(0,-3,8)$	$(4,3,1)$ 4	$(4,3,1)$ 1	$(0,0,0)$
$a_2 = 2$	$(10,5,5)$ 2	$(10,7,3)$ 0 $(4,5,-7)$	$(12,2,10)$ e	$(3,-1,9)$
$a_3 = 3$	$(12,6,6)$ 0 $(3,0,3)$	$(17,8,9)$ 0 $(12,5,1)$	$(11,3,8)$ 3	$(2,0,7)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \end{pmatrix}$	$\begin{pmatrix} 7 \\ 6 \\ -4 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 9 \\ 3 \\ 1 \end{pmatrix}$	$Z(X_1^1) = (34,39)$ $X_1^1 = \begin{pmatrix} 0 & 4 & 1 \\ 2 & 0 & e \\ 0 & 0 & 3 \end{pmatrix}$

Table 11: Solution X_1^1 at node 1

$H_1^1 = \{(1,1)\}$. If x_{11} enters the basis we obtain at node 2, the solution $X_2 = X_0$ which is already visited, then node 2 is fathomed.
In T_1^2, the basic variable $x_{22} = 0$, then we assign the value e to the cell $(2,2)$ and the resulting solution X_1^2 is shown in Table 12.

$H_1^2 = \{(3,1)\}$, so x_{31} enters the basis and we obtain at node 3 the basic feasible solution X_3 in Table 13.

Checking degeneracy: *X_3 is non degenerate. Since X_3 is a unique optimal solution according to c^2, then X_3 is efficient, $D_E := D_E \cup \{X_3\}$, $SND := SND \cup \{Z(X_3)\}$, $H_3 = \varnothing$, then node 3 is pruned.*
The MOTP-Algorithm terminates because all created nodes are fathomed and the set of

	$d_1 = 2$	$d_2 = 4$	$d_3 = 4$	(u_i^0, u_i^1, u_i^2)
$a_1 = 5$	$(7, 3, 4)$ 0 $(4, 2, 1)$	$(4, 3, 1)$ 4	$(4, 3, 1)$ 1	$(0, 0, 0)$
$a_2 = 2$	$(10, 5, 5)$ 2	$(10, 7, 3)$ e	$(12, 2, 10)$ 0 (-4,-5,7)	$(3, -1, 9)$
$a_3 = 3$	$(12, 6, 6)$ 0 $(7, 5, -4)$	$(17, 8, 9)$ 0 $(12, 5, 1)$	$(11, 3, 8)$ 3	$(2, 0, 7)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \end{pmatrix}$	$\begin{pmatrix} 7 \\ 6 \\ -4 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 9 \\ 3 \\ 1 \end{pmatrix}$	$Z(X_1^2) = (34, 39)$ $X_1^2 = \begin{pmatrix} 0 & 4 & 1 \\ 2 & e & 0 \\ 0 & 0 & 3 \end{pmatrix}$

Table 12: Solution X_1^2 at node 1

	$d_1 = 2$	$d_2 = 4$	$d_3 = 4$	(u_i^0, u_i^1, u_i^2)
$a_1 = 5$	$(7, 3, 4)$ 0 $(-3, -3, 5)$	$(4, 3, 1)$ 2	$(4, 3, 1)$ 3	$(0, 0, 0)$
$a_2 = 2$	$(10, 5, 5)$ 0 $(-7, -5, 4)$	$(10, 7, 3)$ 2	$(12, 2, 10)$ 0 $(-4, -5, 7)$	$(7, 4, 2)$
$a_3 = 3$	$(12, 6, 6)$ 2	$(17, 8, 9)$ 0 $(12, 5, 1)$	$(11, 3, 8)$ 1	$(2, 0, 7)$
$\begin{pmatrix} v_j^0 \\ v_j^1 \\ v_j^2 \end{pmatrix}$	$\begin{pmatrix} 10 \\ 6 \\ -1 \end{pmatrix}$	$\begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 9 \\ 3 \\ 1 \end{pmatrix}$	$Z(X_3) = (44, 31)$ $X_3 = \begin{pmatrix} 0 & 2 & 3 \\ 0 & 2 & 0 \\ 2 & 0 & 1 \end{pmatrix}$

Table 13: Solution X_3 at node 3

non-dominated points is: $SND = \{(31, 47), (34, 39), (44, 31)\}$ *and the search tree corresponding to example 2 is shown in Figure 2.*

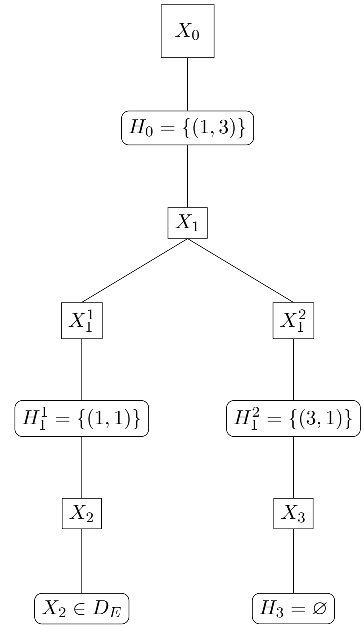

Figure. 2. Search tree 2

4. Theoretical Results

The following results are established to support different steps of the MOTP-Algorithm.

Lemma 1. *Let X_l be a basic feasible solution of MOTP at a node l of the MOTP-Algorithm's search tree, if $Z(X_l)$ is a dominated point then node l is fathomed.*

Proof. Each couple of efficient solutions of MOTP is joined by a chain (refer [19]), this means that there is a sequence of intermediate efficient solutions (forming a chain) connecting any two efficient solutions, hence if $Z(X_l)$ is a dominated point, any non-explored efficient solution could be reached by following the chain of intermediate efficient solutions. This concludes the proof.

Theorem 1. *Every optimal solution for problem $(P0)$ is an efficient solution for MOTP.*

Proof. Assume that X is an optimal solution of $(P0)$, but not an efficient solution for MOTP. This implies that there exists a feasible solution Y that dominates X, that is, $Z_i(Y) \leq Z_i(X),\ \forall i \in \{1, ..., r\}$ and there exists $j \in \{1, ..., r\}$ such that $Z_j(Y) < Z_j(X)$. When we sum up both sides of these inequalities, we get $Z_1(Y) + \ldots + Z_r(Y) < Z_1(X) + \ldots + Z_r(X)$, this contradicts the fact that X is optimal for $(P0)$. Hence, X is an efficient solution for MOTP.

Theorem 2. *If X_l^1 is a basic feasible solution of MOTP at a node l of the MOTP-Algorithm's search tree, and H_l^1 is empty, then the node l is pruned.*

Proof. If H_l^1 is empty, then $\hat{c}_{ij}^0 < 0,\ \forall (i,j) \in N_l$. The criterion c^0 will decrease in all directions in the set N_l. Knowing that c^0 was initially at the optimum and since we only consider increasing directions of c^0, then after pivoting, the resulting basic feasible solution at the child node $l+1$ will already be visited, therefore node l is pruned.

Theorem 3. *If X_l is a basic feasible solution of MOTP at a node l of the MOTP-Algorithm's search tree, and H_l^2 is empty, then node l is pruned.*

Proof. Suppose that H_l^2 is empty and let X_m be a basic feasible solution obtained at the child node m, $m = l+1$, then there are two cases:
Case 1: If $\hat{c}_{ij}^k \geq 0,\ \forall k \in K$ with at least a strict inequality, then we have $Z(X_m) = Z(X_l) + \sum_{k\in K} \hat{c}_{ij}^k x_{ij}$, where $\sum_{k\in K} \hat{c}_{ij}^k x_{ij} > 0$. Then, $Z(X_l) \leq Z(X_m)$ with at least a strict inequality, so $Z(X_m)$ is a dominated point and by the lemma 1, the node l is fathomed.
Case 2: If $\hat{c}_{ij}^k = 0,\ \forall k \in K$ then $Z(X_m) = Z(X_l)$ which means that X_m is an alternative solution to X_l.

And a consequence of Theorem 2 and Theorem 3 is the statement in the next corollary.

Corollary 1. *If the set H_l is empty, then corresponding feasible solutions set at the node l doesn't contain efficient solutions.*

Proof. If H_l is empty, then $H_l^1 \cap H_l^2$ is empty. The case corresponding to one of the two sets is empty is proved in Theorems 2 and 3.
If both sets are empty then, $\exists (i,j) \in N_l$ such that $\hat{c}_{ij}^0 < 0$ in which case any pivoting in this direction brings us back to a vertex already visited (refer Theorem 2), and $\exists (i,j) \in N_l$ such that $\hat{c}_{ij}^k >= 0,\ \forall k \in K$ in which case any pivoting in this direction leads us to a vertex which is not efficient (refer Theorem 3).

Theorem 4. *Let X_l be a basic feasible solution of MOTP at a node l of the MOTP-Algorithm's search tree, then any basic feasible solution obtained at the child node $l+1$ by entering a non-basic cell$(i,j) \in H_l^3$ into basis is either not efficient or an alternative solution.*

Proof. Suppose that $(i,j) \in H_l^3$ and let X_l be a basic feasible solution corresponding to a parent node l then using the definition of the set H_l^3, we distinguish two cases:
Case 1: let $Z(X_{ij})$ and $Z(X_{st})$ be the criterion vector of obtained basic feasible solution if (i,j) and (s,t) enters the basis, respectively. Then $Z(X_{ij}) = Z(X_l) + \sum_{k\in K} \hat{c}_{ij}^k x_{ij}$ and $Z(X_{st}) = Z(X_l) + \sum_{k\in K} c_{st}^k x_{st}$. As $\sum_{k\in K} \hat{c}_{st}^k x_{st} \leq \sum_{k\in K} \hat{c}_{ij}^k x_{ij}, \forall k \in K$, with at least a strict inequality, then $Z(X_{st}) \leq Z(X_{ij})$, with at least a strict inequality, hence X_{ij} is not an efficient solution.
Case 2: If $\hat{c}_{st}^k x_{st} = \hat{c}_{ij}^k x_{ij},\ \forall k \in K$, then $Z(X_{st}) = Z(X_{ij})$, hence X_{ij} is an alternative solution.

Theorem 5. *MOTP-Algorithm converges toward the set of non-dominated points of MOTP in a finite number of steps.*

Proof. D is a compact feasible region with a finite number of basic feasible solutions. In addition, at each node l of the search tree, three fathoming rules were applied: the first when a dominated point is found, the second when the set H_l is empty in other words the solution X_l is an ideal point since no criterion can be minimized and the third when an efficient solution has already been determined, hence MOTP's non-dominated points set is returned after a finite number of iterations.

5. Computational results and comparative study

This section contains a summary of the experimental results achieved while implementing both approaches, MOTP-Algorithm and Isermann's method, on an Intel(R) Core (TM) i5-7300U CPU @ 2.60GHz 2.70GHz and 8GB RAM processor computer, we carried out the numerical implementation using MATLAB R2015 software. None of the optimization packages are used and all of the involved functions are programmed.

5.1. Data structure

Assuming that $m = n$, ten instances of each triplet (m, n, r) are randomly generated, with m and n values in{10,11,12,15,20,25,30}, the number of criteria r in {3,4,5}, the c^k coefficients are in in $[1, 100]$ and the coefficients of a and d are in $[1, 30]$.
For the non-degenerate case, MOTP-Algorithm is compared to Isermann's method (refer [19]). The obtained results are shown in Table 14.

Instance	MOTP-Algorithm CPU(s)			Isermann method CPU(s)			SND		
m n r	average	min	max	average	min	max	average	min	max
10 10 3	0.07	0.04	0.14	0.14	0.06	0.29	20.20	11.00	33.00
4	0.74	0.24	1.30	1.40	0.41	3.70	66.80	30.00	106.00
5	7.35	4.47	11.75	15.94	12.93	17.61	420.80	302.00	628.00
11 11 3	0.09	0.03	0.18	0.20	0.05	0.30	15.60	10.00	22.00
4	1.92	1.03	2.90	3.68	0.67	8.13	145.00	92.00	198.00
5	16.40	7.28	28.09	26.28	11.77	39.84	649.40	393.00	950.00
12 12 3	0.15	0.03	0.27	0.43	0.04	1.40	31.80	9.00	58.00
4	1.97	0.64	4.27	2.17	0.26	5.87	136.60	74.00	252.00
5	33.97	12.19	72.46	85.88	21.16	252.84	871.20	450.00	1440.00
15 15 3	0.15	0.03	0.27	0.26	0.03	0.64	25.00	7.00	39.00
4	4.47	3.47	6.23	8.75	6.28	10.30	213.80	180.00	265.00
5	310.46	186.66	678.24	492.60	359.50	836.53	2444.20	1755.00	4033.00
20 20 3	1.54	0.18	6.76	3.03	0.32	13.60	123.40	21.00	513.00
4	64.62	17.65	192.83	362.26	77.59	471.60	874.00	407.00	2058.00
5	4013.60	1179.10	7469.1	4679.35	1336.64	8538.04	7025.40	4014.00	9795.00
25 25 3	16.94	0.39	81.39	70.00	0.52	345.73	472.20	31.00	2165.00
4	1635.78	206.35	6963.50	5409.66	288.24	13466.44	4066.00	1495.00	12663.00
30 30 3	157.50	96.96	199.02	163.22	97.68	229.37	147.00	55.00	199.00

Table 14: Results for non-degenerate instances

The approach is also tested for the degenerate case such that for each triplet (m, n, r), five instances were generated where both m and n values are in $\{4, 5, ..., 15\}$, the number of criteria r in $\{3, 4, 5\}$ and the c^k coefficients are in $[1, 100]$, the coefficients of a and d are in $[1, 30]$, with $a < m * n$ (see Theorem 1 in [18]). Table 15 summarizes the acquired results.

5.2. Results analysis for the non-degenerate case

We can notice in general that the $CPU(s)$ time and the number of non-dominated points of the MOTP-Algorithm increase on average with the size of the instance (m, n, r), especially with the criteria number r, for non-degenerate instances. Due to this, we did not perform the computations for $r = 5$ from $(m, n, r) = (25, 25, 5)$:

- For $r = 3$:
  - minimum average: $CPU = 0.07$ s and $SND = 20.20$ for $(m, n) = (10, 10)$,
  - maximum average: $CPU = 157.50$ s and $SND = 147$ for $(m, n) = (30, 30)$.
- For $r = 4$:
  - minimum average: $CPU = 0.74$ s and $SND = 66.80$ for $(m, n) = (10, 10)$,
  - maximum average: $CPU = 1635.78$ s and $SND = 4066$ for $(m, n) = (25, 25)$.
- For $r = 5$:
  - minimum average: $CPU = 7.35$ s and $SND = 420.80$ for $(m, n) = (10, 10)$,
  - maximum average: $CPU = 4013.60$ s and $SND = 7025.40$ for $(m, n) = (20, 20)$.

Among the 180 instances solved for the non-degenerate case, the results show that MOTP-Algorithm outperforms Isermann's method for $r = 3$, $r = 4$ and $r = 5$. The gap between the two methods exhibits a range from 8.59 seconds to over 600 seconds for $r = 5$ and from 0.07 seconds to 53.06 seconds for $r = 3$. The average gap for instances with r = 4 varies from 0.19 seconds to 3773.88 seconds, where the highest gap is recorded for the instance with (m, n) values of $(25, 25)$ and a number of non-dominated points ranging from 1495 to 12663.
we observe that the gap between the two methods, MOTP-Algorithm and Isermann's method, varies notably depending on the number of criteria r and the problem instance (m, n), as the problem size increases the search tree grows, hence the significantly longer CPU time can be explained by two main factors:

- Higher SND : Finding a greater number of solutions requires more computational efforts and, consequently, increases the CPU time significantly.
- Higher Number of Variables: Additionally, the increase in computation time can be linked to the higher number of variables in the problem instances. As the number of variables increases, the complexity of the optimization problem grows exponentially, leading to longer execution times.

5.3. Results analysis for the degenerate case

Instance	MOTP-Algorithm CPU(s)			SND		
m n r	*average*	*min*	*max*	*average*	*min*	*max*
4 4 3	0.39	0.14	1.10	14.60	11.00	17.00
4	0.42	0.20	1.26	16.60	12.00	23.00
5	0.85	0.27	1.64	29.60	14.00	47.00
5 5 3	1.20	0.82	2.32	23.40	15.00	29.00
4	1.42	0.91	1.89	46.60	32.00	65.00
5	5.59	2.68	8.36	164.60	81.00	256.00
6 6 3	3.23	2.07	5.54	53.40	41.00	73.00
4	5.15	3.10	8.00	94.80	60.00	119.00
5	82.30	40.23	163.27	537.60	273.00	874.00
7 7 3	400.15	2.21	1982.12	166.40	46.00	613.00
4	100.83	5.84	197.23	280.20	87.00	475.00
5	1165.62	11.64	4431.39	511.60	118.00	975.00
8 8 3	222.32	6.79	1080.90	234.40	36.00	903.00
4	466.63	93.08	1412.21	347.80	212.00	522.00
5	1521.75	158.61	4966.90	718.00	150.00	1396.00
9 9 3	31.39	11.00	48.45	111.20	83.00	135.00
10 10 3	286.80	68.27	699.11	207.80	169.00	289.00
11 11 3	819.45	60.02	3389.23	173.40	114.00	208.00
12 12 3	76.32	9.97	278.86	89.00	16.00	166.00
13 13 3	1405.90	194.80	3365.21	232.40	44.00	344.00
14 14 3	2711.20	507.56	5265.53	312.00	276.00	375.00
15 15 3	3246.42	1503.65	6057.79	277.20	110.00	424.00

Table 15: Results for degenerate instances

The degenerate case of the MOTP problem is much trickier to answer than the traditional transportation problem since there are so many different combinations that might be examined in this case, one degenerate vertex has more than one basis (refer [24]) and some of these basis doesn't lead to a distinct adjacent vertex. It takes a lot of time and memory to process all of the basis connected to a degenerate vertex since the set of non-dominated points of the MOTP problem is based on set H_l, which depends on basis.
The handling of degenerate instances revealed that the MOTP-Algorithm is adversely affected by the additional calculations of the improved N-tree approach required in this situation. Consequently, instances of types $(m, n, 5)$ have been omitted from (9,9,5).

6. Conclusion

In this paper, a branch-and-bound approach for finding all non-dominated points for the multi-objective transportation problem is described for both degenerate and non-degenerate cases. The separation idea is based on the criteria's improving directions at every possible extreme point. Additionally, numerous tests have been developed to fathom nodes whose corresponding domains do not contain efficient solutions allowing entire branches of the search tree to be pruned. The experiment's findings demonstrate that our algorithm can solve non-degenerate instances up to (30,30,3) in a fair amount of time; the smaller r, the faster the algorithm is. Solving degenerate instances has been made possible by using the appropriate approach that has been recommended for this purpose in the literature. Also, a comparative study concluded that our algorithm performs better than Isermann's method. All of these factors make it seem conceivable to modify the strategy to deal with other difficult multi-objective transport problems. However, we do not guarantee that problems with large dimensions may be solved in a fair amount of time; in these situations, suitable approximation techniques must be used.

Although the existing literature on Multi-Objective Transportation Problems (MOTP) presents a range of methodologies, a notable limitation is the absence of robust strategies to effectively address degeneracy, which can significantly complicate the identification of non-dominated points. To address this gap, we propose a novel exact method grounded in the branch-and-bound principle, enhanced by a tailored procedure specifically designed to manage degeneracy. Additionally, our approach integrates a cutting-plane mechanism to efficiently eliminate non-dominated feasible points, ensuring more precise and reliable outcomes.

Future work could explore the extension of the proposed method to larger-scale problems by adapting it to handle more complex instances of the multi-objective transportation problem (MOTP), thereby broadening its applicability to a wider range of real-world scenarios.

References

[1] H. A. Abdelwali, M. Alardhi, A. M. Khalfan, and M. H. Abdelati. Using the minimize distance method to find the best compromise solution of multi-objective transportation problem with case study. *International Journal of Traffic and Transportation Engineering*, 11(2), 2022.

[2] M.M. Ahmed, A.R. Khan, M.S. Uddin, and F. Ahmed. A new approach to solve transportation problems. *Open Journal of Optimization*, 5(1):22–30, 2016.

[3] J.A. Alkhulaifi, H.A. Abdelwali, M. Al-Ardhi, E.E.M. Ellaimony, and Khalid. An algorithm for solving bi-criteria large scale transshipment problems. *The Global Journal of Researches in Engineering*, 14(4), 2014.

[4] J. Alrajhi, H.A. Abdelwali, E.E.M. Ellaimony, and S.A.M. Swilem. A decomposition algorithm for solving a class of bi-criteria multistage transportation problem with case study. *International Journal of Innovative Research in Science, Engineering and Technology*, 2(9), 2013.

[5] Y.P. Aneja and K.P. Nair. Bicriteria transportation problem. *Management Science*, 25(1):73–78, 1979.

[6] P. Anukokila, B. Radhakrishnan, and M. Rajeshwari. Multi-objective transportation problem by using goal programming approach. *International Journal of Pure and Applied Mathematics*, 117(11):393–403, 2017.

[7] H.P. Benson. Optimization over the efficient set. *Journal of Mathematical Analysis and Applications*, 98(2):562–580, 1984.

[8] A. Charnes and WW. Cooper. The stepping stone method of explaining linear programming calculations in transportation problems. *Management science*, 1(1):49–69, 1954.

[9] G.B. Dantzig. A proof of the equivalence of the programming problem and the game problem. *Activity analysis of production and allocation*, 13, 1951.

[10] J.A. Diaz. Solving multiobjective transportation problem. *Ekonomicko-Matematicky Obzor.*, 14:267–274, 1978.

[11] A. Elazeem, A. Mousa, M. El-Shorbagy, S. Elagan, and Y. Abo-Elnaga. Detecting all non-dominated points for multi-objective multi-index transportation problems. *Sustainability*, 13(3):1372, 2021.

[12] E.M. Ellaimony, H.A. Abdelwali, J.M. Alrajhi, M.S. Al-Ardhi, and Y. Alhouli. Solution of a class of bi-criteria multistage transportation problem using dynamic programming technique. *International Journal of Traffic and Transportation Engineering*, 4:115–122, 2015.

[13] R.J. Gallagher and O.A. Saleh. Constructing the set of efficient objective values in linear multiple objective transportation problems. *European Journal of Operational Research*, 73(1):150–163, 1994.

[14] M. Gen, Y. Li, and K. Ida. Solving multi-objective transportation problem by spanning tree-based genetic algorithm. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 82(12):2802–2810, 1999.

[15] F. Geue. An improved n-tree algorithm for the enumeration of all neighbors of a degenerate vertex. *Annals of Operations Research*, 46:361–391, 1993.

[16] M. Hakim. An alternative method to find initial basic feasible solution of a transportation problem. *Annals of pure and applied mathematics*, 1(2):203–209, 2012.

[17] F.L. Hitchcock. The distribution of a product from several sources to numerous localities. *Journal of mathematics and physics*, 20(1-4):224–230, 1941.

[18] M.S. Hung, W.O. Rom, and A.D. Waren. Degeneracy in transportation problems. *Discrete applied mathematics*, 13(2-3):223–237, 1986.

[19] H. Isermann. The enumeration of all efficient solutions for a linear multiple-objective transportation problem. *Naval Research Logistics Quarterly*, 26(1):123–139, 1979.

[20] M.A. Islam, A.R. Khan, M.S. Uddin, and M.A. Malek. Determination of basic feasible solution of transportation problem: a new approach. *Jahangirnagar University Journal of Science*, 35(1):101–108, 2012.

[21] Z. Juman and M. Hoque. An efficient heuristic to obtain a better initial feasible solution to the transportation problem. *Applied Soft Computing*, 34:813–826, 2015.

[22] L. Kaur, S. Singh, A. Bhandari, S. Singh, and M. Ram. An effective approach for solving multi-objective transportation problem. *Journal of Reliability and Statistical Studies*, pages 153–170, 2023.

[23] A.R. Khan, A.Vilcu, N.Sultana, and S.S. Ahmed. Determination of initial basic feasible solution of a transportation problem: a tocm-sum approach. *Buletinul Institutului Politehnic Din Iasi, Romania, Secţia Automatica si Calculatoare*, 61(1):39–49, 2015.

[24] H.J. Kruse. *Degeneracy graphs and the neighbourhood problem.* 2012.

[25] S.M. Lee and L.J. Moore. Optimizing transportation problems with multiple objectives. *AIIE transactions*, 5(4):333–338, 1973.

[26] L. Lemarchand, D. Massé, P. Rebreyend, and J. Hakansson. Multiobjective optimization for multimode transportation problems. *Advances in Operations Research*, page 13, 2018.

[27] G. Maity and S.K. Roy. Solving two-stage multi-objective transportation problem using goal programming and its application to sustainable development. In *Computational Intelligence Methodologies Applied to Sustainable Development Goals.* Springer, Switzerland, 2022.

[28] D. Mardanya, G. Maity, and S. Roy. The multi-objective multi-item just-in-time transportation problem. *Optimization*, 71(16):4665–4696, 2021.

[29] A.A. Mousa. Using genetic algorithm and topsis technique for multiobjective transportation problem: A hybrid approach. *International Journal of Computer Mathematics*, 87(13):3017–3029, 2010.

[30] M.A. Nomani, I. Ali, and A. Ahmed. A new approach for solving multi-objective transportation problems. *International Journal of Management Science and Engineering Management*, 39:165–173, 2016.

[31] N.V. Reinfeld and W.R. Vogel. Mathematical programming. *(No Title)*, 1958.

[32] J.L. Ringuest and D.B. Rinks. Interactive solutions for the linear multiobjective transportation problem. *European Journal of operational research*, 32(1):96–106, 1987.

[33] A. Shafaat and S. Goyal. Resolution of degeneracy in transportation problems. *Journal of the Operational Research Society*, pages 411–413, 1988.

[34] Y. Shi. A transportation model with multiple criteria and multiple constraint levels. *Mathematical and computer modelling*, 21(4):13–28, 1995.

[35] V. Srinivasan and G.L. Thompson. Algorithms for minimizing total cost, bottleneck time and bottleneck shipment in transportation problems. *Naval Research Logistics Quarterly*, 23(4):567–595, 1976.

[36] T. Stewart and H.W. Ittmann. Two-stage optimization in a transportation problem. *Journal of the Operational Research Society*, 30(10):897–904, 1979.

[37] C. Sutcliffe, J. Board, and P. Cheshire. Goal programming and allocating children to secondary schools in reading. *Journal of the Operational Research Society*, 35(8):719–730, 1984.

[38] H.A. Taha. *Operations research: an introduction.* Pearson Education India, Delhi, 2013.

[39] J. Wu, C. Pu, S. Ding, G. Cao, C. Xia, and P. Pardalos. Multi-objective optimization of transport processes on complex networks. *IEEE Transactions on Network Science and Engineering*, 10(2):780–794, 2023.

[40] S.A. Zaki, A.M. Abd Allah, H.M. Geneedi, and A.Y. Elmekawy. Efficient multiobjective genetic algorithm for solving transportation, assignment, and transshipment problems. 2012.

[41] Y.J. Zheng and S.Y. Chen. Cooperative particle swarm optimization for multiobjective transportation planning. *Applied intelligence*, 39:202–216, 2013.