MULTIPLE CRITERIA DECISION MAKING
Vol. 11 2016

Nolberto Munier*

A NEW APPROACH TO THE RANK REVERSAL PHENOMENON IN MCDM WITH THE SIMUS METHOD

DOI: 10.22367/mcdm.2016.11.09

Abstract

When a ranking is obtained for a set of projects, the introduction of a new project, worse than the others, may sometimes perturb the ranking. This is called rank reversal, and happens in most Multi Criteria Decision Making models. The purpose of this paper is to demonstrate that a new method, based on Linear Programming, is immune to rank reversal, which is proved by analyzing the algorithm used to solve the problem. The paper also examines a situation that produces rank reversal when two or more projects have close or identical values.

Keywords: projects, ranking, linear programming, Simplex, rank reversal, SIMUS.

1 Introduction

Given a Multiple Criteria Decision-Making (MCDM) scenario, for instance with four projects A-B-C-D, subject to several criteria and solved by any method, the result indicates preference of some projects over others and this preference/equality constitutes a ranking. For instance, in this case the ranking – obtained using any decision-making method – could be: B $\succcurlyeq$ A $\succcurlyeq$ D $\succcurlyeq$ C. The symbol '$\succcurlyeq$' means is preferred to or equal to, or precede; therefore, B is preferred to A, which is preferred to D, which is preferred to C. Rank Reversal (RR) produces changes in the ranking by altering or even reversing the order of preferences. Rank reversal was observed by Belton and Gear (1983) in the Analytic Hierarchy Process (AHP) (Saaty, 1987). Rank reversal is considered undesirable since it shows weakness in the model used for decision-making. Some authors

* Valencia Polytechnic University, INGENIO, Valencia, Spain, e-mail: nolmunier@yahoo.com.

suggest that a comparison between different models to determine the most appropriate and reliable one – something that has not been achieved yet – could be made by taking into account robustness and strength that is, preserving ranking stability when the original system of projects is modified by changing the number of projects. Wang and Triantaphyllou (2006) and Maleki and Zahir (2012) performed an exhaustive analysis of the occurrence of RR in different models.

Experience shows that several scenarios may alter a ranking, namely:

1. Adding a worse project.
2. Adding a better project.
3. Adding a project which is nearly or entirely identical to another one.
4. Deleting a project.

The addition of a new project E, worse than any in the ranking, can sometimes disrupt the ranking. Common sense and intuition say that if E is worse than all the others, it should go to the end of the ranking, and then the ordering should not be altered. Conversely, if E is better than all the others, it should go to the top of the ranking, but without altering the order. Neither case produces RR but placing the new project in some other, intermediate position in the ranking may do so.

For instance, the ranking above can be written as E ≽ B ≽ A ≽ D ≽ C, if E is the best project, or as B ≽ A ≽ D ≽ C ≽ E if it is the worst one, or as B ≽ A ≽ E ≽ ≽ D ≽ C if it is better than D and C. Observe that the ranking preserves the ordering since it has incorporated only the preference of E over D and C. If E is identical to any other element of the original set, its inclusion will not produce RR, and therefore does not influence the ranking. This is what common sense says, but the real-life situation may be different.

There is no doubt about the necessity of determining the causes for this 'phenomenon' and diverse theories have been developed to explain it. Analysis and discussions have been going on for years and different explanations have been given. Let us start here by analyzing a new project or vector whose components are: 1) its contribution relative to the associated cost or benefit (C_j), and 2) its performance values for the set of criteria (a_{ij}).

In this paper four cases are analyzed. The literature on RR asserts that if a worse project is introduced no changes should be produced in the ranking. But how do we define a worse or a better project? This is a fundamental issue but it is not addressed here. From this author's point of view this is the nub of the question, because on what basis can we assert that a project vector is worse or better than others? Noting that a new project has a lesser cost or a larger benefit (C_j) than others is not enough; the (a_{ij}) values also play a very important if not a larger role than (C_j). Comparing the influence of its performance values (a_{ij}), however, is a more complicated issue and not an obvious one, because for a particular criterion a certain value (a_{ij}) of the new project can be better than the corresponding value of other projects, while for another criterion it could be the opposite, taking into account the action of each criterion, of course.

Assume, as usual in Linear Programming (LP), that columns represent projects and rows represent criteria. It can happen, for instance, that for criterion i_3 the performance value (a_{34}) (i.e. the performance value in the third row or criterion (3) and the fourth column or project (4)), is better than any other performance value for this row, while for criterion i_2 it is the opposite. In addition, most models use weights for criteria, and then it may happen that criterion i_3 has more influence than criterion i_2, which can produce a change in the ranking.

According to Wang and Triantaphyllou (2008), a reliable and stable method for decision-making should not produce RR, when subject to any of the three different tests:

Test number 1*:* ”An effective MCDM method should not change the indication of the best project when a non-optimal project is replaced by another worse project (given that the relative importance of each decision criterion remains unchanged”.

Test number 2: ”The rankings of projects by an effective MCDM method should follow the transitivity property”.

Test number 3: ”For the same decision problem and when using the same MCDM method, after combining the rankings of the smaller problems that an MCDM problem is decomposed into, the new overall ranking of the projects be identical to the original overall ranking of the un-decomposed problem”.

Other researchers believe that the most difficult situation appears when two projects have very close performances (or are nearly identical), or when they are identical (see Saaty, 1987; Belton and Gear, 1983). Cascales and Lamata (2012), even assert that ”It is well known that when the projects are very close the order between them can depend on the method used on their evaluation” (see also Li, 2010).

As an example in the case of a maximization criterion, the new project may have a performance that is worse than all of the others with respect to that criterion, or better, or in between. Consequently, stating that a new project vector is worse than those already existent, we mean that all performances with respect to all criteria, as well as the corresponding (C_j), must be worse than the others which in reality is possible but uncommon. Some authors (Wang and Triantaphyllou, 2008) try to analyze this issue by using random numbers in a simulation, which certainly may correspond to reality for a new project vector.

This author's opinion is that there could be situations where the existence of better performances can lead to a change of the ranking – but not to a RR. That is, if a raking shows a D > B > A > C > E > F, the introduction of a new alternative **G**, which is better than **C,** or **G > C**, means that the new ranking will be D > B > A > **G > C >** E > F. As seen, the new ranking does not show changes in the other precedence.

The objective of this paper is to demonstrate that a new model called SIMUS – Sequential Interactive Model of Urban Systems (Munier, 2011a and 2011b; Teames International, 2014), is not subject to RR. To prove this assertion it is necessary to know how this model works, and this is briefly explained in Section 2.

2 The SIMUS model

It is assumed that the reader has some knowledge of the LP technique (Kantorovich, 1939); see MIT (2016) for very clear explanation and examples, as well as Romero and Balteiro (2013). LP is taught in most undergraduate courses on MCDM, and therefore it is not explained here. Instead we provide here a detailed explanation of how SIMUS works. When LP is applied to an initial decision matrix, with the purpose of maximizing or minimizing an objective function, it uses the Simplex algorithm (Dantzig, 1963), which identifies the best solution. This is Pareto efficient, and consequently cannot be improved, that is, it is optimal. The Simplex algorithm is solved, for instance, by the 'Solver' software (FrontLineSystems, 2015), which is used in SIMUS.

As an example, consider three projects subject to five criteria as shown in the initial matrix of Figure 1, a problem that will be solved via the SIMUS model, in order to explain its functioning.

To understand this model it is necessary to take into account that for SIMUS, objective functions and criteria are equivalent, because both are linear functions, and both are subject to maximization, minimization or equalization. Consequently, in the initial matrix all criteria are at some moment used as objective functions. A thorough explanation of SIMUS with many examples can be found in TEAMES International (2004), and downloaded as free fully operational software from decisionmaking.esy.es.

SIMUS starts by using the first criterion as the objective function, by removing it from the decision matrix, and the Simplex algorithm determines the best solution or project, if such a solution exists. This preference is visualized by comparing values or scores that the algorithm assigns to each project (the higher the better). Thus, when the first objective is processed, the result is saved in a matrix called Efficient Results Matrix (ERM) and indicates that project 1 has the score of 0.57, project 2 has the score of 0.91, while project 3 has the score of 0, meaning that this last project is not a part of the solution. Consequently, according to this first objective, the best solution is project 2, although the two scores are Pareto efficient or optimal.

When the second criterion is used as objective function it appears that only project 3 is selected with the score of 1, while the lack of positive values in the other two projects indicates that these are not selected by this objective. The same procedure is followed for criteria 3, 4 and 5 and the respective scores are saved in the ERM matrix. Since criteria may have different units they have to be normalized, and then the normalized Efficient Result Matrix (ERM) is built. Any normalization system can be used, and SIMUS allows to choose from Total sum in a row, Maximum value in a row, Euclidean formula and Min-Max. Whatever the system chosen, the results or ranking are not changed.

The next stage is to add up all values in each column (SC), which gives, for instance, the score of 2.27 for project 2. Note that projects 2 and 3 satisfy three criteria each, while project 1 satisfies only two; their relation with the total number of criteria constitutes the Participation Factor (PF). That is, project 1 has a participation of 2/5 while projects 2 and 3 participate with 3/5 each. This participation is used as a weight for projects since a large number of PF means that the corresponding project satisfies more criteria. These (PF) are then normalized resulting in the Normalized Participation Factor (NPF). This ratio is obtained taking into account the number of values and the number of criteria as mentioned above, thus, for instance, for project 1 it is 2/5 = 0.4.

For each project or column, the (NPF) is then multiplied by the column sum (SC) and its product constitutes the score for that project, as can be seen in the boxed row. The higher the better, consequently, the best project is 2 followed by projects 3 and 1. This allows for building the ERM ranking as depicted. Thus, this result was obtained taking into account for each project its values for all criteria.

In the second stage SIMUS considers the values by row in the ERM matrix, that is, it analyzes for each criterion its values for all projects. Here the model finds the differences with all the other values in the same row, starting from the highest value in the first row. The result is saved in a new square matrix formed by the projects. This new matrix is called Project Dominance Matrix or PDM. The process is repeated for the same row for the next highest value and this procedure is repeated with all the values. That is, the model finds the degree by which a project dominates or outranks another.

Next, all values in a row are added; the result gauges the dominance of a project in that row. Thus, project 1 has the dominance value of 1.9. The same addition is applied to each column, and the model finds the degree by which a project is outranked by or subordinated to another. In this case, project 1 has the subordinate value of 3.2. The net difference for the same project gives the net value as a score. Thus, the score for project 1 is 1.9 – 3.2 = -1.3.

SIMUS orders them in decreasing order and constructs a ranking. Even when scores are different for the same project in ERM and PDM, their rankings coincide, that is: Ranking from ERM = Ranking from PDM.

Consequently, the same problem is solved by **two different procedures** and **the same ranking is obtained.**

Initial Matrix

	Project 1	Project 2	Project 3
Criterion 1	0.23	0.91	
Criterion 2			0.63
Criterion 3		0.50	
Criterion 4	1.18	0.56	1.18
Criterion 5	0.29	1.89	1

After running Simplex with all objectives this is the resulting efficient matrix

Efficient Results Matrix (ERM)

	Project 1	Project 2	Project 3
Criterion 1	0.57	0.91	
Criterion 2			1
Criterion 3		1	
Criterion 4	3.96		0.74
Criterion 5		1	0.53

Efficient Results Matrix (ERM) Normalized

	Project 1	Project 2	Project 3	
Criterion 1	0.38	0.62		
Criterion 2			1	
Criterion 3		1		
Criterion 4	0.84		0.16	
Criterion 5		0.66	0.34	
Sum of Column (SC)	1.22	2.27	1.50	Number of criteria
Participation Factor (PF)	2	3	3	5
Norm. Participation Factor (NPF)	0.40	0.60	0.60	
Final Result (SC x NPF)	**0.49**	**1.36**	**0.90**	

ERM Ranking **Project 2 - Project 3 - Project 1**

Project Dominance Matrix (PDM)

	Subordinated projects				
Dominant proj.	Project 1	Project 2	Project 3	Row sum of dominant projects	Net dominance
Project 1		0.8	1.1	1.9	-1.3
Project 2	1.9		1.9	3.8	1.8
Project 3	1.3	1.2		2.5	-0.5
Column sum of subordinated projects	3.2	2.0	3.0		

PDM Ranking **Project 2 - Project 3 - Project 1**

Figure 1. Initial matrix, and ERM and PDM matrices in the SIMUS method

3 Why SIMUS does not produce rank reversal?

The simple and straight answer is: because it is based on the Simplex algorithm that does not allow it. To understand this very important algorithm, consider the following problem:

Table 1 shows the initial data of an example, which consists in selecting the best project of a renewable energy power plant using one of two sources of renewable energy: Solar energy (x_1) and Photovoltaic (x_2). Its elements are:

Table 1: Initial data

	Projects or projects				
	Solar energy x_1	Photovoltaic x_2			
Unit cost (Cj)	0.72	0.68	**Objective function Z = 0.72 x_1 + 0.68 x_2 (MIN)**		
Criteria	Project's or projects' contributions (aij) or performances		Action	Action operator	Constant value (B)
Efficiency index	0.85	0.75	MAX	≤	1
Financial index	0.78	0.98	MIN	≥	0.84
Land use index	0.92	0.65	MAX	≤	0.94
Generation index	0.99	0.60	MIN	≥	0.80

Z is the objective function minimizing the total unit cost. Its equation is $Z = 0.72, x_1 + 0.68\ x_2$.
Cj: Unit cost related to each project.
0.72: (C_1), unit cost for project x_1.
0.68: (C_2), unit cost for project x_2.
a_{ij}: Values corresponding to alternatives x_1 and x_2 for all criteria. The problem consists in determining the values of x_1 and x_2 that satisfy the objective function.

The simplex algorithm starts with this initial matrix arranged in the form of a tableau as shown in Table 2.

Table 2: First Simplex tableau

Cj			0.72	0.68	0	0	0	M	M	Ratio	
	Basic	bi	x1	x2	s1	s2	s3	A1	A2		
0	s1	1	0.85	**0.75**	1					1.18	
M	A1	0.84	0.78	0.98		-1		1		0.93	
0	s3	0.94	0.92	0.65			1			1.02	
M	A2	0.8	**0.99**	**0.6**					1	0.81	Key ro
Objective function	Zj	**1.64M**	1.77M	1.35M	0	0	0	M	M		
Index row	Cj - Zj		0.72-1.77M	0.68-1.75M	0	0	0	0	0		
			Key column								

The tableau includes artificial variables Aj for minimization (using the ≥ operator in the corresponding equation), with a very large cost value M, and slack variables sj for maximization (using the ≤ operator), to convert the inequalities to equations, and with cost values equal to '0'. At the beginning of the computation the objective function Z = 1.77M, that is (0.78M + 0.99M), which is extremely high and corresponds to artificial variables or projects A1 and A2, both of which constitute the initial solution of the problem. This is the starting point for the computation. To improve this performance the Simplex algorithm uses two indexes: The Index row (Cj-Zj) and the Key row (bi/aij), where b_i is the right hand side value for the ith criterion.

The first index selects the variable to be entered into the system to improve the solution, that is, to decrease the cost. This is obtained by selecting the most negative value in the index row (0.72-1.77M); in this case the most negative value is related to alternative or project x_1 (Solar energy). The corresponding column (shaded), is called Key column. To preserve the dimensions of the problem (this is a two dimensional problem, because we have two projects), it will be now necessary to eliminate one of the artificial projects. This is done by using the key row (shaded), which indicates that A2 must be eliminated – see Chinneck (2000), for a justification. In the next step the algorithm recalculates the complete matrix, because the basis has changed, and we get the second Simplex tableau shown in Table 3.

Table 3: Second Simplex tableau

Cj →			0.72	0.68	0	0	0	M	M	Ratio	
↓	Basic	bi	x_1	x_2	s_1	s_2	s_3	A_1	A_2		
0	s_1	0.31		0.23	1				-0.86	0.36	
M	A_1	0.21		**0.51**		-1		1	-0.79	0.27	← Key row
0	s_3	0.00		0.09			1		-0.93	-0.19	
0.72	x_1	0.81	1	0.61					1.01	-0.80	
Objective function	Zj	**0.21M**	0.72	0.51M+0.44	0	-M	0	M	0.79M		
Index row	Cj - Zj		0	0.44-0.51M	0	M	0	0	0.21M		
				↑ Key column							

Note that project x_1 is now a unit vector, and therefore is in the basis. The objective function is now: Zj = 0.51M + 0.44, that is (0.51xM + 0.61 x 0.72), which is still a very high value, but considerably less than the first one. Therefore the cost has been reduced. The process is now repeated, i.e, the algorithm looks for the most negative Cj-Zj value and finds that it corresponds to project 2 (Photovoltaic) (0.44-0.51M). The key row index is applied again and then the artificial project A_1 is removed. The process continues until there is no more negative Cj-Zj, as shown in Table 4. As can be seen there are no more negative Cj-Zj and this indicates that the final and optimal solution has been reached with x_1 (Solar) = 0.56 and x_2 (PV) = 0.41.

Table 4: Third Simplex tableau

Cj ↓	Basic	bi	0.72 x_1	0.68 x_2	0 s_1	0 s_2	0 s_3	M A_1	M A_2
0	s_1	0.22			1	0.46		-0.46	
0.68	x_2	**0.41**		1		-1.97		1.97	-2
0	s_3	0.00				0.18	1	-0.09	
0.72	x_1	**0.56**	1			1.19		-1.19	
Objective function	Zj	**0.68**	0.72	0.68	0	-0.48	0.00	0.68	0.73
Index row		Cj - Zj	0	0	0	0.48	0	M-0.68	M-0.73

The process has been explained in some detail to show how the Simplex always selects a better project, based on its Cj and its aij values (from Zj). In a more complicated scenario, the number of projects and criteria is irrelevant**; the Simplex will select only those projects that improve previous solutions**; consequently, it is impossible to select a project that does not satisfy this condition.

3.1 Adding a new project

Let us see now how the system reacts when a new project is introduced about which we do not know if it is better or worse than the existing projects. Of course, with the introduction of this new project, the **original problem with *n* projects has changed, and so it is a new one.** The new problem will have $n + 1$ projects, but the same rules apply. Assume that to our original problem with two projects we add a third one (x_3). If we apply the Simplex to this new problem the algorithm will perform as before when there were two projects. Consequently, if C_3-Z_3 of the new project is positive respecting to C_1-Z_1 and C_2-Z_2, this **new project will never be selected.** This is the reason why no rank reversal can be produced in SIMUS. However, if the cost of opportunity of x_3 is better than the cost of opportunity of x_1 and x_2, then the new project will be selected as the best project in the ranking. Naturally, this is not rank reversal, but the result of introducing a new project that is better than the existent ones. However, even in this last case the original order in the ranking must be preserved. A complete and thorough explanation of the Simplex Tableau is found in Kothari (2009) and in MIT (2016).

3.2 Adding an exact copy of an existing project

According to some researchers the most likely scenario for RR is when two projects (or the existing one and a new vector) are nearly or entirely identical. In this section we analyze this case and demonstrate that SIMUS is immune to this phenomenon. For instance, we can introduce a new project x_3 identical to x_1 to our solar power problem (both are shaded in Table 5).

Table 5: First Simplex tableau – introducing x_3 identical to x_1

Cj	Basic	bi	x_1	x_2	x_3	s_1	s_2	s_3	A_1	A_2	Ratio	
			0.72	0.68	0.72	0	0	0	0	M		
0	s_1	1	0.85	0.75	0.85	1					1.18	
M	A_1	0.84	0.78	0.98	0.78		-1			1	0.93	
0	s_3	0.94	0.92	0.65	0.92			1			1.02	
M	A_2	0.8	0.99	0.6	0.99				-1		0.81	Key row
Objective function	Zj	**1.64M**	1.77M	1.35M	1.77M	0	0	0	-M	M		
Index row	Cj - Zj	0.72-1.77M	0.68-1.75M	0.72-1.77M	0	0	0	M	0			

Key column

This demonstrates that if we have two identical vectors as projects, the system considers only one of them and rejects the other one. Consequently the ranking is preserved.

According to the rule it will be now logical to introduce x_1 or x_3 since both have the largest negative value. We can introduce either, because when transformed they will both be basic variables, but only one of them will be in the solution (see Table 6).

Table 6: Second Simplex tableau

Cj	Basic	bi	x_1	x_2	x_3	s_1	s_2	s_3	A_1	A_2	Ratio	
			0.72	0.68	0.72	0	0	0	0	M		
0	s_1			0.23		1				-0.86	0.36	
M	A_1			**0.51**			-1		1	-0.79	0.27	Key row
0	s_3			0.09				1		-0.93	-0.19	
0.72	x_1		1	0.61	1					1.01	-0.80	
Objective function	Zj	**0.21M**	0.72	0.51M+0.44	0.72	0	-M	M	M	0.79M		
Index row	Cj - Zj		0	0.44-0.51M	0	0	M	0	0	0.21M		

Continuing with the process we must select x_2 as the entering variable and criterion 2 (A_1) (shaded), as the leaving variable. The transformation values are in Table 7.

Table 7: Third Simplex tableau

Cj	Basic	bi	x_1	x_2	x_3	s_1	s_2	s_3	A_1	A_2
			0.72	0.68	0.72	0	0	0	M	M
0	s_1					1	0.46		-0.46	
0.68	**x_2**	**0.41**		1			-1.97		1.97	-2
0	s_3						0.18	1	-0.09	
0.72	**x_1**	**0.56**	1		1		1.32		-1.19	
Objective function	Zj	**0.68**					-0.48	0	0.08	1
Index row	Cj - Zj		0	0	0	0.48	0	0	M-0.68	M-0.73

The outcome has the same values as before and the same ranking.

3.3 Demonstration of absence of RR in SIMUS when more than a single project is added

Starting from an initial problem several scenarios are proposed. Note that these involve much stricter conditions as those found in the literature on RR where, in general only one scenario is examined at a time, while here we are using more than one and even mixing different scenarios.

3.4 Solving a problem with SIMUS software

Assume the initial matrix shown in Table 8 with five projects (in bold case) is given. Projects 6, 7 and 8 are added later. The system uses Euclidean normalization but, as mentioned before, any other can be used. This case is solved using SIMUS software and the result is shown on the last screen (Figure 2). Observe that SIMUS provides two solutions in its ERM and PDM matrices. The ERM solution is found in the solid black row while the PDM solution is in the solid black PDM column. Note, however, that both ERM and PDM rankings are identical.

Table 8: Initial decision matrix with five projects

Initial decision matrix Added project						**Added project**	**Added projects**	
Project 1	**Project 2**	**Project 3**	**Project 4**	**Project 5**	Project 6 **worse** than any original	Project 7 **better** than any original	Project 8 = Project 3	**Action**
6200	**6050**	**4800**	**5100**	**3800**	3600	6500	4800	**MAX**
3	**4.2**	**2.5**	**6.1**	**3.10**	2.4	6.5	2.5	**MAX**
20	**20**	**21**	**30**	**32**	35	18	21	**MIN**
4	**3**	**2.5**	**3**	**5**	2.4	5.5	2.5	**MAX**

The result is: **4 ≽ 5 ≽ 3 ≽ 2 ≽ 1.**

Efficient Results Matrix (ERM) Normalized

	Project 1	Project 2	Project 3	Project 4	Project 5	
Target 1		0.32	0.68			
Target 2				1.00		
Target 3						
Target 4					1.00	
Sum of Column (SC)	0.00	0.32	0.68	1.00	1.00	Number of targets
Participation Factor (PF)	0	1	1	1	1	4
Norm. Participation Factor (NPF)	0.00	0.25	0.25	0.25	0.25	
Final Result (SC x NPF)	**0.00**	**0.08**	**0.17**	**0.25**	**0.25**	

ERM Ranking **Project 4 - Project 5 - Project 3 - Project 2 - Project 1**

Project Dominance Matrix (PDM)

	Dominated projects						
Dominant proj.	Project 1	Project 2	Project 3	Project 4	Project 5	Row sum of dominant projects	Net dominance
Project 1						0.0	-3.0
Project 2	0.3			0.3	0.3	0.9	-1.4
Project 3	0.7	0.4		0.7	0.7	2.4	0.4
Project 4	1.0	1.0	1.0		1.0	4.0	2.0
Project 5	1.0	1.0	1.0	1.0		4.0	2.0
Column sum of dominated projects	3.0	2.4	2.0	2.0	2.0		

PDM Ranking **Project 4 - Project 5 - Project 3 - Project 2 - Project 1**

Figure 2. The original problem – the final screen of SIMUS showing the results for the initial set of projects (5)

3.4.1 Adding project 6 'worse' than the others

Now we add project 6, which is obviously worse than any other since its performances are lower in maximization and higher in minimization. Figure 3 shows the result, which as can be seen replicates the ranking of the situation with only five projects. Project 6 is added but with '0' score' in the ERM matrix, meaning that it is not considered.

Output: Original ranking preserved.

Efficient Results Matrix (ERM) Normalized

	Project 1	Project 2	Project 3	Project 4	Project 5	Project 6	
Target 1		0.32	0.68				
Target 2				1.00			
Target 3							
Target 4					1.00		
Sum of Column (SC)	0.00	0.32	0.68	1.00	1.00	0.00	Number of targets
Participation Factor (PF)	0	1	1	1	1	0	4
Norm. Participation Factor (NPF)	0.00	0.25	0.25	0.25	0.25	0.00	
Final Result (SC x NPF)	0.00	0.08	0.17	0.25	0.25	0.00	

ERM Ranking Project 4 - Project 5 - Project 3 - Project 2 - Project 1 - Project 6

Project Dominance Matrix (PDM)

	Dominated projects							
Dominant proj.	Project 1	Project 2	Project 3	Project 4	Project 5	Project 6	Row sum of dominant projects	Net dominance
Project 1							0.0	-3.0
Project 2	0.3			0.3	0.3	0.3	1.3	-1.1
Project 3	0.7	0.4		0.7	0.7	0.7	3.1	1.1
Project 4	1.0	1.0	1.0		1.0	1.0	5.0	3.0
Project 5	1.0	1.0	1.0	1.0		1.0	5.0	3.0
Project 6							0.0	-3.0
Column sum of dominated projects	3.0	2.4	2.0	2.0	2.0	3.0		

PDM Ranking Project 4 - Project 5 - Project 3 - Project 2 - Project 1 - Project 6

Figure 3. The original problem with 'worse' project 6 added

3.4.2 Adding project 7 keeping project 6 and with $x_3 = x_6 = x_7$

Now we keep project 6 identical to project 3 and add project 7 also identical to 3 and 6 (see Figure 4). The result is again the same ranking: $4 \succcurlyeq 5 \succcurlyeq 3 \succcurlyeq 2 \succcurlyeq 1$. Note that project 6 and 7 have '0' scores.

Output: Original ranking preserved.

Efficient Results Matrix (ERM) Normalized

	Project 1	Project 2	Project 3	Project 4	Project 5	Project 6	Project 7
Target 1		0.32	0.68				
Target 2				1.00			
Target 3							
Target 4					1.00		
Sum of Column (SC)	0.00	0.32	0.68	1.00	1.00	0.00	0.00
Participation Factor (PF)	0	1	1	1	1	0	0
Norm. Participation Factor (NPF)	0.00	0.25	0.25	0.25	0.25	0.00	0.00
Final Result (SC x NPF)	0.00	0.08	0.17	0.25	0.25	0.00	0.00

ERM Ranking Project 4 - Project 5 - Project 3 - Project 2 - Project 1 - Project 6 - Project 7

Project Dominance Matrix (PDM)

Dominant proj.	Project 1	Project 2	Project 3	Project 4	Project 5	Project 6	Project 7	Row sum of dominant projects	Net dominance
	Dominated projects								
Project 1								0.0	-3.0
Project 2	0.3			0.3	0.3	0.3	0.3	1.6	-0.8
Project 3	0.7	0.4		0.7	0.7	0.7	0.7	3.8	1.8
Project 4	1.0	1.0	1.0		1.0	1.0	1.0	6.0	4.0
Project 5	1.0	1.0	1.0	1.0		1.0	1.0	6.0	4.0
Project 6								0.0	-3.0
Project 7								0.0	-3.0
Column sum of dominated projects	3.0	2.4	2.0	2.0	2.0	3.0	3.0		

PDM Ranking Project 4 - Project 5 - Project 3 - Project 2 - Project 1 - Project 6 - Project 7

Figure 4. Adding projects 6 and 7 identical to project 3 simultaneously

3.4.3 Adding a new project identical to another one and simultaneously adding one regarded as the best

We add project 6 which is identical to project 3 and also add project 7 which is regarded as the best of all (see Figure 5). The result is: $4 \succcurlyeq 5 \succcurlyeq 3 \succcurlyeq 2 \succcurlyeq 1$.

Output: Original ranking preserved.

Efficient Results Matrix (ERM) Normalized

	Project 1	Project 2	Project 3	Project 4	Project 5	Project 6	Project 7
Target 1		0.04	0.96				
Target 2				1.00			
Target 3							
Target 4					1.00		
Sum of Column (SC)	0.00	0.04	0.96	1.00	1.00	0.00	0.00
Participation Factor (PF)	0	1	1	1	1	0	0
Norm. Participation Factor (NPF)	0.00	0.25	0.25	0.25	0.25	0.00	0.00
Final Result (SC x NPF)	0.00	0.01	0.24	0.25	0.25	0.00	0.00

ERM Ranking Project 4 - Project 5 - Project 3 - Project 2 - Project 1 - Project 6 - Project 7

Project Dominance Matrix (PDM)

Dominant proj.	Project 1	Project 2	Project 3	Project 4	Project 5	Project 6	Project 7	Row sum of dominant projects	Net dominance
	Dominated projects								
Project 1								0.0	-3.0
Project 2	0.0			0.0	0.0	0.0	0.0	0.2	-2.7
Project 3	1.0	0.9		1.0	1.0	1.0	1.0	5.7	3.7
Project 4	1.0	1.0	1.0		1.0	1.0	1.0	6.0	4.0
Project 5	1.0	1.0	1.0	1.0		1.0	1.0	6.0	4.0
Project 6								0.0	-3.0
Project 7								0.0	-3.0
Column sum of dominated projects	3.0	2.9	2.0	2.0	2.0	3.0	3.0		

PDM Ranking Project 4 - Project 5 - Project 3 - Project 2 - Project 1 - Project 6 - Project 7

Figure 5. Project 6 identical to project 3 and project 7 regarded as the best

3.4.4 Deleting a project from the original portfolio

We are deleting Project 3 (see Figure 6). The result is: 4 ≽ 5 ≽ 2 ≽ 1.

Output: Original ranking preserved.

Efficient Results Matrix (ERM)

	Project 1	Project 2	Project 3	Project 4	Project 5
Target 1	0.35	1.21			
Target 2				1.22	
Target 3					
Target 4					1.63

Efficient Results Matrix (ERM) Normalized

	Project 1	Project 2	Project 3	Project 4	Project 5
Target 1	0.22	0.78			
Target 2				1.00	
Target 3					
Target 4					1.00
Sum of Column (SC)	0.22	0.78	0.00	1.00	1.00
Participation Factor (PF)	1	1	0	1	1
Norm. Participation Factor (NPF)	0.25	0.25	0.00	0.25	0.25
Final Result (SC x NPF)	**0.06**	**0.19**	**0.00**	**0.25**	**0.25**

ERM Ranking Project 4 - Project 5 - Project 2 - Project 1 - Project 3

Project Dominance Matrix (PDM)

Dominant proj.	Project 1	Project 2	Project 3	Project 4	Project 5	Row sum of dominant projects	Net dominance
	Dominated projects						
Project 1			0.2	0.2	0.2	0.7	-1.9
Project 2	0.6		0.8	0.8	0.8	2.9	0.9
Project 3						0.0	-3.0
Project 4	1.0	1.0	1.0		1.0	4.0	2.0
Project 5	1.0	1.0	1.0	1.0		4.0	2.0
Column sum of dominated projects	2.6	2.0	3.0	2.0	2.0		

PDM Ranking Project 4 - Project 5 - Project 2 - Project 1 - Project 3

Figure 6. Deleting project 3

4 Summary of scenarios and results

Table 9 summarizes our findings.

Table 9: Summary of results for different scenarios

		Ranking	Comments	Result referred to ranking
Original project				
	Original result taking into account only **five** projects or projects	**4 ≽ 5 ≽ 3 ≽ 2 ≽ 1 Figure 2**		Initial ranking
Adding one project to original scenario				
Adding project 6 ‘worse’ than others	Project added has worse values in all criteria than all other projects	**4 ≽ 5 ≽ 3 ≽ 2 ≽** 1 ≽ 6 **Figure 3**		Ranking preserved

Table 9 cont.

Adding identical projects to original scenario				
Projects 6 and 7 identical to project 3	Simultaneous addition plus identity with an existing project	**4** ≽ **5** ≽ **3** ≽ **2** ≽ 1 ≽ 6 ≽ 7 **Figure 4**	Project 6 and 7 are not considered since their score is '0'	Ranking preserved
Adding an identical project and at the same time adding another one which is regarded as the best				
Project 6 identical to project 3 and project 7 regarded as the best	Simultaneous **addition** of one project identical to another existent one plus addition of another project regarded as the best	**4** ≽ **5** ≽ **3** ≽ **2** ≽ 1 ≽ 6 ≽ 7 **Figure 5**	Project 6 and 7 are not considered since their score is '0'	Ranking preserved
Projects deletion from the original scenario				
Delete project 3		**4** ≽ **5** ≽ **2** ≽ **1** ≽ 3 **Figure 6**	Project 3 is eliminated since its value is '0'	Ranking preserved

5 Conclusion

The goal of this paper is to demonstrate that when LP is used for decision-making no RR occurs. This was shown by examining the original algebraic procedure of the Simplex algorithm created by Dantzig (1963). It clearly reveals that the incorporation of a new project regarded as worse than the existing projects cannot alter the ranking because the algorithm takes into account both the contribution (cost or benefit) and the performances of the new project. To put it simply, the algorithm works by analyzing and comparing opportunity costs, minimizing or maximizing them. It is a well-known fact that RR occurs also when a project is deleted from the scenario, or when two projects are nearly or entirely identical. These two scenarios were also examined in this paper by modifying the original problem and solving each using SIMUS. Four different scenarios were considered with more than one project added at the time and also including projects with identical data. The author believes that the algebraic analysis performed and the examples proposed validate our claim.

References

Belton V., Gear T. (1983), *On a Short-coming of Saaty's Method of Analytic Hierarchies*, Omega, 11, 228-230.

Cascales S., Lamata M.-T. (2012), *On Rank Reversal on TOPSIS Method*, Mathematical and Computer Modeling, 5, 123-132.

Chinneck J. (2000), *Practical Optimization and Gentle Introduction*, http//www.sce.carleton.ca/faculty/chinneck/po.html (accessed April 04, 2016).

Dantzig G. (1963), *Linear Programming and Extensions*, Princeton University Press and the RAND Corporation.

FrontLineSystems (2015), http://www.solver.com/about-our-company (accessed December 12, 2015).

Kantorovich L.V. (1939), *Mathematical Methods of Organizing and Planning Production*, Management Science, Vol. 6, No. 4 (Jul., 1960), 366-422.

Khotari C. (2009), *Quantitative Techniques*, Vikas Publishing House Pvt Ltd.

Li S. (2010), *Rank Reversal in Properties in Multicriteria Decision Making Models*, Ph.D. Thesis, University of Birmingham.

Maleki H., Zahir S. (2013), *A Comprehensive Literature Review of the Rank Reversal Phenomenon in the Analytic Hierarchy Process*, Journal of Multicriteria Decision Analysis. Vol. 20, No. 3-4, 141-155.

MIT (2016), *Mathematical Programming – An Overview*, http://web.mit.edu/15.053/www/AMP-Chapter-01.pdf (accessed March 19, 2016).

Munier N (2011a), *Procedimiento fundamentado en la Programación Lineal para la selección de alternativas en proyectos de naturaleza compleja y con objetivos múltiples*, Universidad Politécnica de Valencia, España (Ph.D. Thesis).

Munier N. (2011b), *A Strategy for Using Multicriteria Analysis in Decision-Making: A Guide for Simple and Complex Environmental Projects*, Springer, The Netherlands.

Romero C., Balteiro L. (2013), *Métodos de análisis basados en el paradigma de la decisión multicriterio en: La Sostenibilidad de la Agricultura Española*, ed. José A. Gómez-Limón y Ernest Reich, Fundación Cajamar, Almeria, 321-352.

Saaty T. (1987), *Rank Generation, Preservation, and Reversal in the Analytic Hierarchy Process*, Decision Sciences, 18, 157-177.

Teames International (2014), *Software para el método Simus*, https://sites.google.com/site/decision makingmulticriteria/manual-simus-system (accessed December 27, 2016).

Wang X., Triantaphyllou E. (2008), *Ranking Irregularities When Evaluating Projects by Using Some ELECTRE Methods*, Omega 36, 45-63.

Wang X., Triantaphyllou E. (2006), *Ranking Irregularities When Evaluating Projects by Using Some Multi-Criteria Decision Analysis Methods* [in:] *Handbook of Industrial and Systems Engineering*, ed. A. Badiru, CRC Press, Taylor & Francis Group, Boca Raton, FL, U.S.A., Chapter 27, 27-1–27-12.