Facultad de Ingeniería
ISSN: 0121-1129
Universidad Pedagógica y Tecnológica de Colombia

Aliaga-Benavides, Adriel Alejandro; Santana-Pacheco, Yudisel;
Mustelier-Hechavarría, Antonio; Guzmán-Bazán, Yuniel

Computer Algorithm for comparing and ordering models based on indicators and user value

Facultad de Ingeniería, vol. 26, no. 46, 2017, pp. 61-69
Universidad Pedagógica y Tecnológica de Colombia

DOI: https://doi.org/10.19053/01211129.v26.n46.2017.7318

# Computer Algorithm for comparing and ordering models based on indicators and user value

## Algoritmo computacional para comparar y ordenar modelos basados en indicadores y valor de usuario

## Algoritmo computacional para comparar e ordenar modelos baseados em indicadores e valor de usuário

Adriel Alejandro Aliaga-Benavides[*]
Yudisel Santana-Pacheco[**]
Antonio Mustelier-Hechavarría[***]
Yuniel Guzmán-Bazán[****]

**Fecha de recepción:** 12 de octubre de 2016
**Fecha de aceptación:** 3 de julio de 2017

## Abstract

Sometimes a list of models can be compared using a set of indicators, ordered by the user's criterion, given that it is known whether the growth of the indicators value affect the user's interests. This paper proposes a computational algorithm that compares models, assigning them a numerical value that corresponds to the priority given to each model indicator by the user. The proposed algorithm allows to rank the models as other known algorithms do, such as the vector space model; however, it takes into account user comparison priorities. When working with lineal functions, the margin of error in the algorithm calculations is null. The algorithm is validated by the software Ambiens v1.0, whose purpose is to manage the waste control information, showing relevant results in the comparison of models. The algorithm essence is the need to identify the best model of the group, according to the user's criterion per indicator.

**Keywords:** Algorithm; Comparison; Indicators; Models; Ordering.

## Resumen

En ocasiones se tiene un listado de modelos que se deben comparar mediante un grupo de indicadores, ordenados según el criterio de un usuario, y, dado que se conoce si el crecimiento del valor de los indicadores afectaría o no los intereses del usuario. Se muestra un algoritmo computacional capaz de comparar y asignar un valor numérico a cada modelo, de tal forma que el valor asignado se corresponda con la prioridad establecida por el usuario a cada indicador de los modelos. El algoritmo propuesto permite establecer un *ranking* entre los modelos, al igual que otros algoritmos conocidos, como el Modelo Espacio Vectorial; sin embargo, este algoritmo considera las prioridades de comparación del usuario. Al trabajar estrictamente con funciones lineales, el margen de errores de cálculo es nulo. El algoritmo ha sido validado por el software Ambiens v1.0 (dedicado a la gestión de residuos)

[*] M. Sc. Universidad de Granma (Bayamo, Cuba). aaliagab@udg.co.cu. ORCID: 0000-0002-1682-1490.
[**] M. Sc. Universidad de Granma (Bayamo, Cuba). ysantanap@udg.co.cu.
[***] M. Sc. Universidad de Granma (Bayamo, Cuba). tonym@udg.co.cu. ORCID: 0000-0002-3474-955X.
[****]M. Sc. Universidad de Granma (Bayamo, Cuba). yguzmanb@udg.co.cu.

y muestra resultados relevantes en la comparación de modelos. La propuesta cobra su esencia en la necesidad de conocer, de entre un grupo de modelos, el mejor de ellos, según el criterio de usuario por indicadores.

**Palabras clave:** Algoritmo; Comparación; Indicadores; Modelos; Ordenamiento.

## Resumo

Em ocasiões tem-se um listado de modelos que devem comparar-se mediante um grupo de indicadores, ordenados segundo o critério de um usuário, e deste modo conhece-se se o crescimento do valor dos indicadores afetaria ou não os interesses do usuário. Mostra-se um algoritmo computacional capaz de comparar e assignar um valor numérico a cada modelo, de tal forma que o valor assignado corresponda-se com a prioridade estabelecida pelo usuário a cada indicador dos modelos. O algoritmo proposto permite estabelecer um *ranking* entre os modelos, ao igual que outros algoritmos conhecidos, como o Modelo Espaço Vetorial; porém, este algoritmo considera as prioridades de comparação do usuário. Ao trabalhar estritamente com funções lineais, a margem de erros de cálculo é nula. O algoritmo tem sido validado pelo software Ambiens v1.0 (dedicado à gestão de resíduos) e mostra resultados relevantes na comparação de modelos. A proposta é relevante na necessidade de conhecer, entre um grupo de modelos, o melhor deles, segundo o critério de usuário por indicadores.

**Palavras chave:** Algoritmo; Comparação; Indicadores; Modelos; Ordenamento.

# I. Introduction

The increase of production costs in the industry, due to the loss of unused raw material has led to the concern of experts who study waste generation from industrial processes. Such concern is sustained by the law of conservation of mass discovered by Lavoisier, which states that the total mass of reacting substances is equal to the mass of products formed in the reaction [1]; from this, it is derived that the mass of the matter before the reaction is equal to that of the matter after the reaction. The matter after the reaction will be composed of the final product and wastes. This confirms that minimizing the waste generation increase the production; therefore, Reverse Logistics (RL) appears in order to achieve such minimization.

In essence, RL deals with the recovery and recycling of containers, packaging of dangerous wastes, and also with the processes of return of inventory excess, customer returns, obsolete products, and season inventory [2]. Several studies have been conducted in this field, such as [3] that proposed a genetic algorithm to solve the problem of decisions related to the location of treatment plants, collecting and transfer places; [4], which proposed an evolutionary algorithm to solve the NP-hard problem of a remanufacturing net; and [5], which solved the problem of programming ground international transportation with a metaheuristic algorithm. Other studies, focused on mathematical models such as Schrady's (considered the father of the RL models), can be found in the literature; examples of them are [6], which optimized the requests to the repairable item warehouse and to the main supplier; [7], which proposed a model to optimize production and refabrication, and [8], whose model analyzed the optimal expression that the return of items must have.

The problem is that the models do not reflect comparison criteria with other similar models. Even if there were any criterion of comparison, an importance user value for a given model over the rest of the models is not calculated. For a better understanding of these issues, this paper shows an example based on the model $(1, R)$ published by [9], given that this paper revolutionized the RL. This deterministic model allows to find optimal lots of requests to the external supplier and the process of recovery [10]. The reason is that when there are N suppliers requiring different conditions, N different models arise. Then, the use of comparison indicators becomes necessary, so that according to a given order of priority, it is possible to determine the proper model based on any criterion of user value (UV) assignment.

In this paper, results and discussion describe algorithms for comparing models, based either on criteria or on numerical methods of high codification complexity. The main result of this paper is a computational algorithm that easily solves the problem of comparison among models through UV assignment by positioning, taking into account that the UV assigned to each model corresponds to the priority set by the user for each indicator.

Therefore, the advantage of the algorithm is that, given a set of N indicators $(I_1, I_2, \dots . I_N)$ with an established priority and a set of M models $(m_1, m_2, \dots . m_M)$, where a list of N values $I_i$ corresponds to each $m_j$, it allows to order the models according to the priority set by the indicators, and at the same time, to calculate a UV for each model that corresponds to the given priority. For that reason, the algorithm is named "Comparison for ordering models based on indicators and UV (COMIUV)."

# II. Materials and methods

In order to simplify the understanding of the algorithm, we analyzed the Schrady's model, published in [9], as a case study, for which some indicators with different values for each model have been defined. Furthermore, we simulated and visualized the proposed computational algorithm, and provided the generalized version of the algorithm.

## *A. Schrady's model, a case study*

It is important to highlight that this model does not belong to the area of informatics, but it is reflected in an industrial engineering problem. Therefore, in order to understand the essence of the model and the relation among the indicators proposed herein [9].

Suppose we have indicators $(I_1, I_2, I_3, I_4, I_5, I_6)$ by which Schrady's models $(m_1, m_2, m_3, m_4, m_5)$ must be compared, and that those indicators have been ordered by an user from greater to smaller priority, resulting in the form $(I_4, I_1, I_5, I_2, I_3, I_6)$. Also, it is known that $(I_4, I_1, I_5, I_2)$ are good for the user, which means that the greater their value, the greater the benefit for the user, and that $(I_3, I_6)$ are bad for the user, that is, the greater their value, the smaller the benefit for the user. As an example, Table 1 shows the distribution of the indicators values for each model.

### Table 1
### Example of a distribution of values

| | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ | good | priority |
|---|---|---|---|---|---|---|---|
| $I_1$ | 22.45 | 33.12 | 09.64 | 54.23 | 45.03 | true | 2 |
| $I_2$ | 0.43 | 0.15 | 0.75 | 0.59 | 0.22 | true | 4 |
| $I_3$ | 77.55 | 66.88 | 90.36 | 45.77 | 54.97 | false | 5 |
| $I_4$ | 3425 | 4526 | 3004 | 6814 | 3024 | true | 1 |
| $I_5$ | 2351 | 964 | 1142 | 857 | 1028 | true | 3 |
| $I_6$ | 5776 | 5490 | 4146 | 7671 | 4052 | false | 6 |

**Step 0:** Table 1 must be adjusted according to the priority given by the user (Table 2).

### Table 2
### Matrix of indicators adjusted according to priority

| | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ | good |
|---|---|---|---|---|---|---|
| $I_4$ | 3425 | 4526 | 3004 | 6814 | 3024 | true |
| $I_1$ | 22.45 | 33.12 | 09.64 | 54.23 | 45.03 | true |
| $I_5$ | 2351 | 964 | 1142 | 857 | 1028 | true |
| $I_2$ | 0.43 | 0.15 | 0.75 | 0.59 | 0.22 | true |
| $I_3$ | 77.55 | 66.88 | 90.36 | 45.77 | 54.97 | false |
| $I_6$ | 5776 | 5490 | 4146 | 7671 | 4052 | false |

**Step 1:** Conversion of the matrix of value of indicators (MVI) into a vector matrix, where each indicator value per model is transformed by a vector of two components: $< B, C >$ (B is the model identifier, and C is the indicator value for model B; if the indicator is bad for the user, C is assigned $-C$). The objective is to keep the reference of the values of the indicators for each model after ordering in step 2 (Table 3).

### Table 3
### Vector matrix of indicators (VMI)

| | | | | |
|---|---|---|---|---|
| <$m_1$,3425> | <$m_2$,4526> | <$m_3$,3004> | <$m_4$,6814> | <$m_5$,3024> |
| <$m_1$,22.45> | <$m_2$,33.12> | <$m_3$,09.64> | <$m_4$,54.23> | <$m_5$,45.03> |
| <$m_1$,2351> | <$m_2$,964> | <$m_3$,1142> | <$m_4$,857> | <$m_5$,1028> |
| <$m_1$,0.43> | <$m_2$,0.15> | <$m_3$,0.75> | <$m_4$,0.59> | <$m_5$,0.22> |
| <$m_1$,$-77.55$> | <$m_2$,$-66.88$> | <$m_3$,$-90.36$> | <$m_4$,$-45.77$> | <$m_5$,$-54.97$> |
| <$m_1$,$-5776$> | <$m_2$,$-5490$> | <$m_3$,$-4146$> | <$m_4$,$-7671$> | <$m_5$,$-4052$> |

**Step 2:** To move the VMI, six orderings are performed, one for each row of the VMI, so that each row is ordered from greater to smaller, according to the C value of each vector placed in the entry $[\mathrm{i}][\mathrm{j}]$ (i corresponds to the row, and j to the column, where the first row or column has a value equal to 0). Any ordering algorithm that best suits can be used (Table 4).

### Table 4
### VMI moved by row ordering

| | | | | |
|---|---|---|---|---|
| <$m_4$,6814> | <$m_2$,4526> | <$m_1$,3425> | <$m_5$,3024> | <$m_3$,3004> |
| <$m_4$,54.23> | <$m_5$,45.03> | <$m_2$,33.12> | <$m_1$,22.45> | <$m_3$,09.64> |
| <$m_1$,2351> | <$m_3$,1142> | <$m_5$,1028> | <$m_2$,964> | <$m_4$,857> |
| <$m_3$,0.75> | <$m_4$,0.59> | <$m_1$,0.43> | <$m_5$,0.22> | <$m_2$,0.15> |
| <$m_4$,$-45.77$> | <$m_5$,$-54.97$> | <$m_2$,$-66.88$> | < $m_1$,$-77.55$> | <$m_3$,$-90.36$> |
| <$m_5$,$-4052$> | <$m_3$,$-4146$> | <$m_2$,$-5490$> | <$m_1$,$-5776$> | <$m_4$,$-7671$> |

After step 2, we are in conditions to define the proper order of the models. Selecting according to the first row is enough, and in case of a tie, the decision will be based on the next row. In this case, the order $m_4, m_2, m_1, m_5, m_3$ can be observed (Table 4). The problem is that perhaps the intentions do not lead to a simple ordering, but to an assignation of a value to each model, which corresponds to the previous ordering from greater to smaller. At the same time, this value must integrate the positioning of each vector with a value, whose meaning will be named as user value (UV), which must substitute the C value of the vector, so that the total UV (sum of each UV of the vectors corresponding with the model) comply with the desired ordering. In order to achieve the UV assignment in C for each vector, it is necessary, even in the worst circumstances of comparison, to have a proper ordering, that is, the winning model in the worst way (it wins because its indicator of greater priority is the best among all the models, but the rest of its indicators are the worst) is better than the second best model (its indicator of greater priority is the second among all the models, but the rest of its indicators win among all the models); and it is required that equation (1) holds for any entry $[i][j]$, according to Table 5.

$$C_{ij} + \sum_{i=k}^{N-1} C_{(i+1)(M-1)} > C_{i(j+1)} + \sum_{i=k}^{N-1} C_{(i+1)j} \qquad (1)$$

**Table 5**

An example of the sums to be compared by similitude of color

| | | | |
|---|---|---|---|
| ……………………………………………………………………………… | | | |
| ⋮ | $< \mathrm{B}, \mathrm{C}_{\mathrm{ij}} >$ | $< \mathrm{B}, \mathrm{C}_{\mathrm{i(j+1)}} >$ | ……………………………………………… |
| | $< \mathrm{B}, \mathrm{C}_{\mathrm{(i+1)j}} >$ | ……………………………………………… | $< \mathrm{B}, \mathrm{C}_{\mathrm{(i+1)(M-1)}} >$ |
| ⋮ | ⋮ | | ⋮ |
| | | ……………………………………………… | |
| ⋮ | ⋮ | | ⋮ |
| | $< \mathrm{B}, \mathrm{C}_{\mathrm{(N-1)j}} >$ | ……………………………………………… | $< \mathrm{B}, \mathrm{C}_{\mathrm{(N-1)(M-1)}} >$ |
| ⋮ | | | |
| | | ……………………………………………… | |
| ⋮ | | | |

Note that the used logic for this methodology of assigning value to the C component is sustained by a logic ruled provided by the method of "proof by contradiction" or by "strong reduction to an absurdity". If the questioning of a determinable statement is followed by the acceptance and questioning of a statement incompatible with its denial, then the initial statement is accepted [11]. If it is possible to assign an UV in C of each vector, such that even in the worst circumstances the model with the indicator of greater priority with greater value still remains the winner, in any other circumstance where the conditions of this indicator of greater priority prevail and the conditions of values in the rest of the indicators become better, the model having this indicator with a greater value will always win.

**Step 3:** Conversion of the VMI into a matrix of user value (MUV) (Table 6). This step consists in moving along all the entries $[i][j]$ of the VMI and replacing the C component of each vector with a value that holds the previous condition. In order to achieve this, it is proposed that the C value for each vector, placed in the position $[i][j]$, be modified according to the following rules (where N is the number of indicators and M the number of models):

Rule 1: if $i = N - 1$ do (2), else if $j = M - 1$ do (3).

$C_{(N\text{-}1)j} = M\text{-}j$ (2)

$C_{i(M-1)} = C_{(i-1)0} + 1$ (3)

Rule 2: if rule 1 does not hold do (4):

$$C_{ij} = C_{i(j+1)} + \sum_{i=1}^{N-1} C_{(i+1)j} \quad (4)$$

TABLE 6

MATRIX OF USER VALUE

| | | | | |
|---|---|---|---|---|
| <$m_4$,7145> | <$m_2$,4978> | <$m_1$,3457> | <$m_5$,2399> | <$m_3$,1667> |
| <$m_4$,1666> | <$m_5$,1165> | <$m_2$,809> | <$m_1$,560> | <$m_3$,388> |
| <$m_1$,387> | <$m_3$,273> | <$m_5$,190> | <$m_2$,131> | <$m_4$,90> |
| <$m_3$,89> | <$m_4$,64> | <$m_1$,45> | <$m_5$,31> | <$m_2$,21> |
| <$m_4$,20> | <$m_5$,15> | <$m_2$,11> | <$m_1$,8 > | <$m_3$,6> |
| <$m_5$,5> | <$m_3$,4> | <$m_2$,3> | <$m_1$,2> | <$m_4$,1> |

**Step 4:** Finding the UV of each model. After assigning an UV in C for each vector, the total UV corresponding to each model can be calculated as the sum of the C value of the vectors corresponding to each model. The final result of the algorithm is shown in Table 7 and Fig. 1.

TABLE 7

ORDERING FROM GREATER TO SMALLER WITH THE RELATION MODEL- TOTAL UV

| Models | $m_4$ | $m_2$ | $m_1$ | $m_5$ | $m_3$ |
|---|---|---|---|---|---|
| Total UV | 8986 | 5953 | 4459 | 3805 | 2427 |

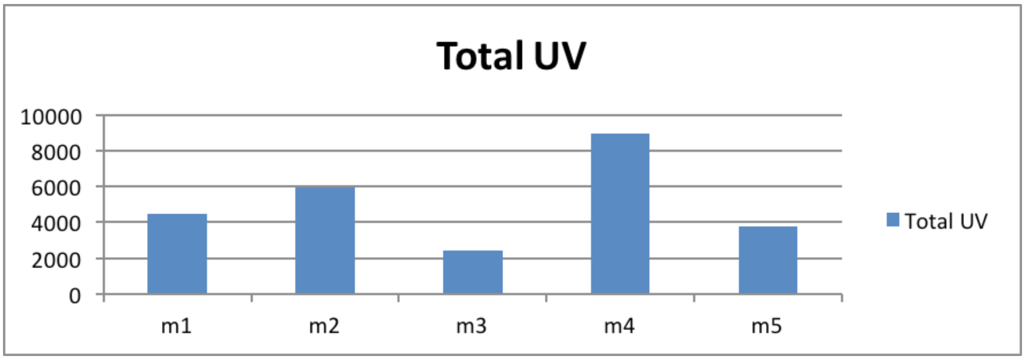

FIG. 1. Relation model - Total UV shown to the user.

### *B. Computational algorithm: Model ordering based on indicators and user value (COMIUV). Generalized version*

In this section, a generalized version of the COMIUV algorithm, proposed here, is detailed, implemented, and tested in the software Ambiens v1.0 for waste management. RL and the selection of the proper model are implemented in this software, according to the priorities given by the user to the indicators for model comparison, where one of them is the Schrady's model $(1, R)$ visualized in the previous example. It is worthy of remark that this generalized version applies to any case of applied models.

**Situation:** We want to compare M models, taking into account the priority order given by the user to N indicators. At the end, the user value must be set for each model, where the greater UV, the better the model.

**Entry:** The algorithm entry is a matrix of N rows and M+2 columns, where each entry [i][j] relates the M model to the N values of its indicators. The information about whether the increase of the indicator is good for the user is found in the next to the last column, and the priority of each indicator (row), in the last column.

**Step 0:** First, the matrix must be adjusted according to the priority given by the user (*i.e.,* ordering the rows from smaller to greater by the priority number, where the smaller the number, the greater the priority), and the last column must be deleted, so that the matrix remains with a dimension of $N\ x\ (M+1)$, $N\ x\ (M+1)$, only with the data necessary for the next steps.

**Step 1:** Conversion of the matrix of indicators into a vector matrix, where each indicator value per model becomes a vector of two components: B and C (B is the identifier of the model, and C is the indicator value for model B; if the indicator is bad for the user, C is assigned $-C$). The vectors would remain as:

$$V = < B, C > \ (5)$$

**Step 2:** To move the MVI, N orderings will be performed, one for each row of the MVI, so that each row is ordered from greater to smaller, according to the C value of each vector placed in the entries $[i][j]$.

**Step 3:** Converting the MVI into a matrix of user value (MUV) consists in moving along all the entries $[i][j]$ of the VMI and replacing the C component of each vector with a value that holds condition (1). In order to achieve this, it is proposed to modify the C value for each vector placed in the position $[i][j]$, according to the following rules (where N is the number of indicators, and M the number of models):

***Rule 1:*** if $i = N - 1$ do (2), else if $j = M - 1$ do (3).

***Rule 2:*** if rule 1 does not hold do (4).

**Step 4:** Finding the UV of each model as the sum of the C values of the vectors corresponding to each model. The final result of the algorithm would be an ordered relation, from greater to smaller, among models and their corresponding UV expressed as follows:

$$UV(Mj) = \sum_{i=0}^{N-1} (C\ of\ vector\ with\ model\ j) \ (6)$$

## III. Results and discussion

There are computational algorithms for comparing models such as Markov chain Monte Carlo (MCMC), which is a general set of techniques for simulated sampling that use distributions achieved through Markov chains, in order to generate independent samples based on the distribution of interest [12]; the Metropolis-Hastings algorithm is a general MCMC method for finding a conditional density $g(.|\theta)$ or an easily sampled distribution, and then, generating observations of the proposed distribution, in order to decide whether they correspond to the distribution of $\theta|x$ [13]. Both algorithms depend on using complex numerical methods, which makes their implementation more complex than implementing COMIUV. However, it would be fair to compare COMIUV with other algorithms with similar features, such as entry parameters and main objective.

It is obvious that COMIUV sets a ranking among models. Due to the entry features, models may be seen as objects that can be measured by a number N of indicators. The vector space model (VSM) is a well-known method in the theory of information retrieval that seeks to rank documents, taking into account the weights of the terms found in each document [14]. In this case, the similitude to give any ranking solution for this second algorithm lies in that M documents must be considered for evaluation, taking into account N indicators that set the weights of the terms. It is necessary to consider the models as vectors, whose components values are the weights of each indicator. With the values from table 1, the values in Table 8 can be considered as the entry for the VSM.

Table 8
Entry for VSM

| | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ | good |
|---|---|---|---|---|---|---|
| $I_1$ | 22.45 | 33.12 | 09.64 | 54.23 | 45.03 | true |
| $I_2$ | 0.43 | 0.15 | 0.75 | 0.59 | 0.22 | true |
| $I_3$ | 77.55 | 66.88 | 90.36 | 45.77 | 54.97 | false |
| $I_4$ | 3425 | 4526 | 3004 | 6814 | 3024 | true |
| $I_5$ | 2351 | 964 | 1142 | 857 | 1028 | true |
| $I_6$ | 5776 | 5490 | 4146 | 7671 | 4052 | false |

If the entry is adjusted by the criterion "good", as in COMIUV, multiplying the weights of the indicators that are not good for the user by -1, we obtain the values shown in Table 9.

### TABLE 9

ENTRY ADJUSTED BY THE CRITERION "GOOD" OF THE INDICATOR

| | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ |
|---|---|---|---|---|---|
| $I_1$ | 22.45 | 33.12 | 09.64 | 54.23 | 45.03 |
| $I_2$ | 0.43 | 0.15 | 0.75 | 0.59 | 0.22 |
| $I_3$ | -77.55 | -66.88 | -90.36 | -45.77 | -54.97 |
| $I_4$ | 3425 | 4526 | 3004 | 6814 | 3024 |
| $I_5$ | 2351 | 964 | 1142 | 857 | 1028 |
| $I_6$ | -5776 | -5490 | -4146 | -7671 | -4052 |

MVS works with distances between the entry vectors and a desired vector in the search. The priority of the user in the indicators is not considered in the VSM's philosophy. In this sense, it is logical to consider that for the user, it would be optimal to compare with the best possible vector, which is the one with the greater weight achieved for each indicator in each component, named as "query" in the study case. Then, the definitive entry to take into account is:

$$m_1 = (22.45, 0.43, -77.55, 3425, 2351, -5776\,) \qquad (7)$$

$$m_2 = (33.12, 0.15, -66.88, 4526, 964, -5490\,) \qquad (8)$$

$$m_3 = (09.64, 0.75, -90.36, 3004, 1142, -4146\,) \qquad (9)$$

$$m_4 = (54.23, 0.59, -45.77, 6814, 857, -7671\,) \qquad (10)$$

$$m_5 = (45.03, 0.22, -54.97, 3024, 1028, -4052) \qquad (11)$$

$$query = (54.23, 0.75, -45.77, 6814, 2351, -4052\,) \qquad (12)$$

The resulted VSM ranking after entering these data is shown in Table 10.

### TABLE 10

RESULTS OF VSM FOR ENTRY DATA

| Models | $\mathrm{m}_4$ | $\mathrm{m}_2$ | $\mathrm{m}_5$ | $\mathrm{m}_3$ | $\mathrm{m}_1$ |
|---|---|---|---|---|---|
| Value of ranking | 0.9341 | 0.9323 | 0.9245 | 0.9210 | 0.8884 |

Comparing to Table 7, we observe a coincidence in the order of the models $m_4$ *and* $m_2$, which in the case of COMIUV (Annex 1) were winners in the same order. However, the results corresponding to the rest of the models are different, probably due to the fact that VSM does not take into account the priority of the indicators, because only the ideal vector (query) is of interest to compare the vectors that represent the documents (in this case the models). Table 11 summarizes the comparison between both algorithms.

### TABLE 11

COMPARISON BETWEEN THE COMIVU AND VSM ALGORITHMS

| Algorithm | Ranking setting? | Low implementation Complexity? | Work with indicator priority? | Ranking value? | Error in calculation (Language) |
|---|---|---|---|---|---|
| COMIUV | yes | Yes | yes | yes | 0 (All) |
| MVS | yes | Yes | No | yes | $10^{-16}10^{-16}$<br>(Python) |

## IV. Conclusions

The COMIVU algorithm proposed for model comparison is able to relate levels of priority given by a user to a set of comparison indicators, whose relation materializes in the factor "user value", taking into account the positioning of the vectors. In COMIUV, calculation errors do not occur because it only works with sums depending on the entry data. On the other hand, VSM may have a small calculation error depending on the used programming language, because VSM works with vector distances, where a division by the product of the norms of the compared vectors takes place. Therefore, if the interest is to take into account the priority of the indicators, we suggest using COMIUV, otherwise, if the interest is just to set a ranking, both algorithms are totally useful. The shown COMIUV pseudocode exemplifies its implementation, and the given example illustrate its easy development. The developed algorithm made it possible to set a number named user value (UV), considering that the greater the user value, the better the model. Finally, the proposed algorithm guarantees, as a result, that all models will be ordered with their corresponding UV.

## Author's contributions

Adriel Alejandro Aliaga Benavides developed the algorithm and main content of this paper. Yudisel Santana Pacheco and Yuniel Guzmán Bazán recognized the problematical situation and reviewed the literature. Antonio Mustelier Echavarría reviewed the literature and translated the paper into English.

## References

[1] J. Dalton, *El atomismo en Química*. Spain: Alicante University, 2012.

[2] H. R. Lezama, *et al.*, "Logística Inversa: una alternativa para la gestión de productos fuera de uso (PFU)," *Rev. Inter. La Nueva Gest. Org.,* vol. 9, pp. 13-23, 2010.

[3] M. Ortega, and J. Delgado, "Uso de algoritmos genéticos para resolver el modelo determinista y estocástico para el diseño de una red de recogida de residuos", in *International Conference on Industrial Engineering and Industrial Management*, Madrid, Spain, 2007, pp. 1443-1451.

[4] E. Santibanez, G. Robson, *et al.*, "Um Algoritmo evolucionário combinado com otimização para resolver o problema de projecto de uma supply chain de remanufatura," in *Anais XLIV Simpósio Brasileiro de Pesquisa Operacional.* Rio de Janeiro, Brasil, 2012.

[5] P. Ruiz, "Implementación de un algoritmo metaheurístico para la solución de un problema de programación de transporte terrestre internacional," M.S. Thesis, Universidad Nacional, Medellín, Colombia, 2013.

[6] M. P. Gato, *et al.*, "Revisión de modelos de gestión de inventarios para repuestos reparables," in *4th International Conference On Industrial Engineering and Industrial Management*. Donostia San Sebastián, Spain, 2010, pp. 1329-1335.

[7] I. Konstantaras, "Optimal Control of Production and Remanufacturing in a Reverse Logistics Model with Backlogging," *Mathematical Problems in Engineering,* vol. 2010.

[8] S. R. Singh, and N. Saxena, "An Optimal Returned Policy for a Reverse Logistics Inventory Model with Backorders," *Advances in Decision Sciences*, vol. 2012, 2012.

[9] D. Schrady, "Mathematical models of the reparable item inventory system," Naval Postgraduate School, Monterey, California, 1967.

[10] J. P. Sepúlveda, F. Baesler, and D. A. Núñez, "Benefits of using ad-hoc inventory management models in the presence of return flows," *Revista Facultad de Ingeniería Universidad de Antioquia*, Medellín, Colombia, 2010, no. 55, pp. 210-218.

[11] M. Sierra, "Lógica Básica Paraconsistente y Paracompleta con Negación Clásica," *Revista Universidad EAFIT*, vol. 38 (126), pp. 81-97, 2012.

[12] R. A. Mendoza, "Métodos de inferencia estadística para entrenamiento de modelos ocultos de Markov," *Elementos,* vol. 1 (1), pp. 57-70, 2011.

[13] L. E. Vargas, "Cálculo de la distribución predictiva en un modelo Tar", Master Thesis, Universidad Nacional de Colombia**,** Bogotá, Colombia, 2012.

[14] Y. C. Fonseca, "Recuperación de la información: taxonomía de sus modelos," Revista Cubana de Ciencias Informáticas, vol. 6 (2), pp. 1-8, 2012.