Article

A proposed extended version of the Hadi-Vencheh model to improve multiple criteria ABC inventory classification

Hung-Chieh Chang [1] and Pei-Chun Lin [2,*]

[1] Department of Mathematics, Southwestern Oklahoma State University 1; hungchieh.chang@swosu.edu
[2] Department of Transportation and Communication Management Science, National Cheng Kung University 2; peichunl@ncku.edu.tw
* Correspondence: peichunl@ncku.edu.tw (Tel: +886-62757575 ext 53222)

Featured Application: Inventory Management.

Abstract: In this paper, we present an extended version of the Hadi-Vencheh model for multiple criteria ABC inventory classification. The proposed model is a nonlinear weighted product model (WPM) which determines a common set of weights for all the items. Our proposed nonlinear WPM incorporates multiple criteria with different measure units, without converting the performance of each inventory item in terms of each criterion into a normalized attribute value, thereby providing an improvement over the model proposed by Hadi-Vencheh. Our study mainly includes various criteria for ABC classification, and demonstrates an efficient algorithm for solving nonlinear programming problems in which the feasible solution set does not have to be convex. The algorithm presented in this study improves the solution efficiency of the Canonical Coordinates Method (CCM) algorithm substantially when applied to large scale, nonlinear programming problems. The modified algorithm was tested to compare our proposed model results to the results derived using the Hadi-Vencheh model and demonstrate the algorithm's efficacy. The practical implications of the study are to develop an efficient nonlinear optimization solver by optimizing the quality of existing solutions, thus improving time and space efficiency.

Keywords: non-linear programming 1; Hadi-Vencheh model 2; multiple criteria ABC inventory classification 3; multiple criteria ABC inventory classification 4.

---

1. Introduction

To succeed in managing a growing number of stock-keeping units (SKU), inventory managers have found that inventory classification systems provide an essential context for evaluating inventory management. The ABC analysis is one of the most frequently used inventory classification techniques. Raw materials, sub-assemblies, intermediate products, parts and end product can be divided into three classes, including A (very important items), B (medium important items), and C (relatively unimportant items). The ABC classification problem is solved as a ranking problem by most current classification models [1-3]—that is, a group of inventory items is expressed according to its overall weighted score of criteria in descending order. The idea of the ABC analysis has been applied to inventory management by General Electric during the 1950s. This approach is based on Pareto's famous theory of inequality in the distribution of incomes. A conventional ABC study is conducted on the basis of one criterion, the annual dollar usage (value of an item times its annual usage) of SKUs.

In Pareto's theory, all items are ranked based only on a single criterion; for inventory management, dollar usage has been deemed as the only criterion for managers to classify items into the A, B, and C categories. However, in reality managers sometimes want to consider more attributes of an item when classifying goods; many item characteristics could influence inventory control policy and must be considered. Flores [2] noted that other vital criteria can be adopted in addition to dollar usage, such as commonality, reparability, substitutability, lead time and commonality. For instance, an enterprise must pursue operating in an efficient way that can both minimize total costs and maximize satisfaction brought to their customers. If SKUs are only classified based on the single criterion of dollar usage, an item with less dollar usage but long lead time and high criticality may be misclassified into the C category, resulting in serious damage for the company if the item suffers from stock out.

There are a detailed literary research on multi-choice programming (MCP) theories and applications. MCP is a branch of multiple objective programming which descends from the multiple criteria decision making (MCDM). MCDM tests in various areas several overlapping criteria in decision-making [4-5]. The multiple criteria inventory classification (MCIC) can be viewed as an application of the multiple criteria decision analysis [6-7]. To solve the MCIC problem, the joint-criteria matrix [8] is a simple and easy-to-understand tool, but it is not practical for more than two criteria and involves too much subjectivity. The analytic hierarchy process (AHP) is a popular methodology but it involves subjectivity as well. Methods for solving the ABC inventory classification problem have been systematically and thoroughly reviewed and discussed in relevant literature [9-12]. A number of methods were suggested in order to achieve the multiple criteria classification of SKUs. These methods contribute much to the classification of items and help improve the efficiency and performance of a firm through better inventory management. However, these approaches contain some shortcomings, such as involving too much subjectivity or being overly complicated.

For better allocations of priorities of items and further classification, it is worth developing a model that can accommodate multiple criteria to create guidelines for inventory control. The study builds a proper model for categorizing SKUs and demonstrates an efficient algorithm for solving the nonlinear programming model in which the feasible solution set does not have to be convex. The rest of this paper is as follows: Section 2 provides the details of model development. The solution algorithm and its improvement is presented in Section 3. Section 4 details the results of the model conducted herein, with comparisons to previous studies with a benchmark data set. Conclusions and recommendations for future research are offered in the final section.

2. The HV-model and the WPM

Hadi-Vencheh [13] proposed a multiple criteria weighted nonlinear model for ABC inventory classification. The proposed model, hereafter referred to as the HV-model, is an extension of the Ng-model [1]. The Ng-model transforms the inventory object to a scalar value in all parameters steps. The grouping according to the measured values is then applied according to the ABC theory. Hadi-Vencheh extended the Ng-model to resolve the condition in which the score is independent of the weights from the model for each item. Despite the improvement in maintaining the influences of weights in the final score, one notable problem remains: the HV-model calculates the scores assigned to each item using the weight sum method (WSM) for criteria with different measure units which therefore requires converting the performance of single inventory item in terms of every criterion into a normalized attribute value. Triantaphyllou [14] contend that, if the problem involves criteria with different measure units, the weighted product method (WPM) would be a more suitable tool to calculate the scores given to each item. To avoid an erroneous extreme value leading to inventory items misclassification, we propose the following nonlinear WPM to model the classification problem involving criteria with different measure units. This study therefore presents an broad version of the HV-model, taking weight values into account in the ABC inventory classification for multiple criteria using the WPM, which applies multiplication weights and forms a nonlinear optimization problem. To solve the nonlinear optimization problem efficiently, the canonical coordinates method (CCM) algorithm is used to calculate the weights of the criteria for each inventory item.

Suppose that I inventory items are present, and that the items must be graded as A, B or C based on their results according to J criteria. In particular, let the output of the i th inventory item be referred to as $y_{i,j}$ with respect to each criterion. For simplicity, all parameters are beneficial; in other words, they are positively connected with the degree of value of an item. The goal is to combine many performance scores in the subsequent ABC inventory classification with regard to different parameters into a single score. In both the Ng and HV-models, a nonnegative weight $w_{i,j}$ is the weight of performance contribution of the i th item under the j th criteria to the score of the item. The parameters are supposed to be listed in descending order such that $w_{i,1} \geq w_{i,2} \geq \cdots \geq w_{i,J}$ for all items i. The proposed model by Hadi-Vencheh [13] is as follows:

$$\begin{aligned} \max \quad & S_i = \prod_{j=1}^{J} y_{i,j}{}^{w_{i,j}} \\ \text{s.t.} \quad & \sum_{j=1}^{J} w_{i,j}{}^2 = 1 \\ & w_{i,j} - w_{i,j+1} \geq 0, \quad j = 1,2,\ldots,J-1 \\ & w_{i,j} \geq 0, \quad j = 1,2,\ldots,J \end{aligned} \tag{1}$$

In the HV-model, the performance in each criterion of the i th inventory item $y_{i,j}$ is further normalized to $s_{i,j}$, and the objective function of the nonlinear programming (NLP) model (1) is found to be:

$$\max \quad S_i = \sum_{j=1}^{J} s_{i,j} w_{i,j}$$

Ng [1] indicated that the normalization scaling involves extreme measurement values and would thus have an effect on all normalized measurements if the extremes change. To avoid an invalid extreme value leading to inventory items misclassification, we propose the following nonlinear WPM to model the classification problem involving criteria with different measure units:

$$\begin{aligned} \max \quad & S_i = \prod_{j=1}^{J} y_{i,j}{}^{w_{i,j}} \\ \text{s.t.} \quad & \sum_{j=1}^{J} w_{i,j}{}^2 = 1 \\ & w_{i,j} - w_{i,j+1} \geq 0, \quad j = 1,2,\ldots,J-1 \\ & w_{i,j} \geq 0, \quad j = 1,2,\ldots,J \end{aligned} \tag{2}$$

3. The Solution Algorithm

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation as well as the experimental conclusions that can be drawn.

3.1. Nomenclature

3.1.1. Notation of the weighted product method for ABC classification

- I : set of inventory items
- J : set of evaluation criteria
- $y_{i,j}$: the i th inventory item in terms of the j th criteria
- $w_{i,j}$: the weight of performance contribution of the i th item under the j th criteria
- S_i : score of the item i.

3.1.2. Notation of the CCM Algorithm

- $\Re$: the set of all real numbers
- ξ : decision variables
- ξ^0 : feasible initial solution
- $\phi_i(\xi)$: set of constraints, $i = 1,\ldots,m$
- $f(\xi)$: the objective function
- ξ^* : the optimal solution.

3.2. The CCM Algorithm

This section presents the Canonical Coordinates Method (CCM) algorithm [15-16] applied to solve nonlinear programming problems in which the feasible solution set does not have to be convex. Convexity is a strong property that often replaces differentiability as a desirable property in most constrained optimization problems. However, the CCM is an efficient algorithm to deal with continuous search spaces and benefit from the low computational cost for solving constrained optimization. A set containing nonlinear constraints may or may not be convex. This study mainly demonstrates an efficient algorithm for solving nonlinear programming problems in which the feasible solution set does not have to be convex. The main difference between linear and nonlinear programming is that linear programming helps find the best solution from a set of parameters or requirements that have a linear relationship, whereas nonlinear programming helps find the best solution from a set of parameters or requirements that have a nonlinear relationship. The prerequisite for applying the CCM algorithm is that the theorem of Implied Function can be used in any feasible set. That is, at any point in the feasible set, one can find m variables, say $z = (z_1,\ldots,z_m)$, in such a way that the Jacobian matrix of the constraint functions $\phi = (\phi_1,\ldots,\phi_m)$ with respect to z is nonsingular. From the Implicit Function Theorem there exist functions g_j such that $z_j = g_j(x_1,\ldots,x_n)$, $j = 1,\ldots,m$. We describe the CCM algorithm below:

Input: The nonlinear program

$$\max\left\{ f(\xi) \middle| \phi_i(\xi) = 0, i = 1,\ldots,m \right\}$$

with given differentiable functions $f, \phi_i : \Re^{m+n} \to \Re, i = 1,\ldots,m$ and a feasible point $\xi^0 \in \Re^{m+n}$ satisfying $\phi_i(\xi^0) = 0, i = 1,\ldots,m$.

Output: A critical point ξ^* of f satisfying $\phi(\xi^*) = 0$.

Steps:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2020 doi:10.20944/preprints202010.0527.v1

1. ξ^0 is partitioned into $\xi^0 = \left(x^0, z^0\right)$, where $x^0 \in \Re^n$, $z^0 \in \Re^m$ such that $\det\left(J(\phi, z)\Big|_{\left(x^0, z^0\right)} \right) \neq 0$.
2. For $i, j = 1, \ldots, m$, and $k = 1, \ldots, n$, we calculate the following partial derivatives at point $\xi^0 = \left(x^0, z^0\right)$: $\partial f/\partial x_k$, $\partial f/\partial z_j$, $\partial \phi_i/\partial x_k$, $\partial \phi_i/\partial z_j$.
3. We then calculate the $m \times n$ matrix for the implicit function g: $(\partial g/\partial x) = -(\partial \phi/\partial z)^{-1}(\partial \phi/\partial x)$, and then find the direction $D^0 := (\partial f/\partial x) = (\partial f/\partial x_1, \ldots, \partial f/\partial x_n)$ and $D^0 = (\partial f/\partial x)^T + (\partial f/\partial z)(\partial g/\partial x)$.
4. We perform a line search along the ray through x^0 with the direction $D^0 = D\left(x^0\right)$; that is to say, we find a one-dimensional local optimal t^* of: $\hat{F}(t) := f\left(x^0 + tD^0, z(t)\right)$, $t \geq 0$.

   To do so, we need to solve for $z(t)$, which is done by solving the following system of ordinary differential equations: $\begin{cases} z(0) = z^0 \\ (\partial \phi/\partial x)(\partial F/\partial x)^T + (\partial \phi/\partial z)(dz/dt)^T = 0 \end{cases}$.

   Set $x^* \leftarrow x^0 + t^* D^0$.
5. We compute $z_j^* = g_j\left(x^*\right), j = 1, \ldots, m$ using Taylor polynomial approximation and then apply Newton's method to solve the system of ordinary differential equations at $t = t^*$ above.
6. If $\nabla f\left(x^*, z^*\right) \approx 0$, then we have found a local optimal point. Otherwise, we replace $\left(x^0, z^0\right)$ with $\left(x^*, z^*\right)$ and repeat the procedure.

The CCM algorithm helps us identify the local optimal points of an NLP that the feasible set fulfills the requirements of the Implicit Function Theorem. The problem can then be turned into an NLP problem on a subspace $\Re^n$ of the original space $\Re^{m+n}$.

3.3. Improvement of the Algorithm Using Efficient Selection of Bases

Step 1 in the CCM algorithm is to find a subset of m variables among the $(m+n)$ variables so that a resulting Jacobian matrix is non-singular [8]. This is equivalent to finding a subset of column vectors in the original $m \times (m+n)$ matrix that is linearly independent. Let $\phi_1, \cdots, \phi_m$ be differentiable functions in $(m+n)$ variables:

$$A = \begin{bmatrix} \partial \phi_1/\partial x_1 & \partial \phi_1/\partial x_2 & \cdots & \partial \phi_1/\partial x_{m+n} \\ \partial \phi_2/\partial x_1 & \partial \phi_2/\partial x_2 & \cdots & \partial \phi_2/\partial x_{m+n} \\ \vdots & \vdots & \vdots & \vdots \\ \partial \phi_m/\partial x_1 & \partial \phi_m/\partial x_2 & \cdots & \partial \phi_m/\partial x_{m+n} \end{bmatrix}$$

In order to find a subset of m columns of A that is linearly independent, the original method in Chang and Prabhu [15] was to choose any m subset of the $(m+n)$ columns to check if it

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2020 doi:10.20944/preprints202010.0527.v1

qualifies. There are two drawbacks to doing so: first, there are C_m^{m+n} many choices of such subsets; and second, each choice will require a calculation of the determinant of an $m \times m$ matrix, which has the same complexity as a Gaussian Elimination process. We will show that, by using Gaussian Elimination on A to reach its reduced row echelon form, we can find one subset of columns of A that is linearly independent. The process of Gaussian Elimination is to perform a sequence of row operations to a given matrix to reach its reduced row echelon form. Each type of row operation corresponds to a type of elementary matrix, all of which are nonsingular; each time a row operation is performed, it is equivalent to multiplying the original matrix by an elementary matrix on the left. We can also see that in an $m \times (m+n)$ matrix with rank m, there is a subset of column vectors that is linearly independent. Now let us state the proposition that yields the discovery of the desired linearly independent subset of columns of A.

Proposition: Let m and n be positive integers, A be a $m \times (m+n)$ matrix with rank m, and $u_1, \ldots, u_{m+n}$ be the column vectors of A. Suppose B is the reduced row echelon form of A, and that $v_1, \ldots, v_{m+n}$ are the columns of B. Then, there exist integers $1 \le j_1 < j_2 < \cdots < j_m \le (m+n)$ so that $\left[v_{j_1}, \cdots, v_{j_m}\right] = I_{m\times m}$ forms the $m \times m$ identity matrix. Moreover, the corresponding subset of columns of A, $\left[u_{j_1}, \cdots, u_{j_m}\right]$ is non-singular.

Proof: Matrix B must also have rank m because it is the reduced row echelon form of A, whose rank is m. Thus there are m columns of B that form the $m \times m$ identity matrix. That is, there exist integers $1 \le j_1 < j_2 < \cdots < j_m \le (m+n)$ so that $\left[v_{j_1}, \cdots, v_{j_m}\right] = I_{m\times m}$. During the process of Gaussian Elimination to obtain the reduced row echelon form of A, we can find elementary matrices $E_1, E_2, \cdots, E_p$ that:

$$B = E_p \cdots E_2 E_1 A$$

Note that the k-th column of B is also obtained from performing the same row operations on the k-th column of A. Thus:

$$v_{j_k} = E_p \cdots E_2 E_1 u_{j_k}, \quad k = 1, 2, \cdots, m$$

and

$$E_p \cdots E_2 E_1 \left[u_{j_1}, \cdots, u_{j_m}\right] = \left[v_{j_1}, \cdots, v_{j_m}\right] = I .$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2020 doi:10.20944/preprints202010.0527.v1

Since all the elementary matrices $E_1, E_2, \cdots, E_p$ and the identity matrix I are nonsingular, then $\left[u_{j_1}, \cdots, u_{j_m}\right]$ must also be nonsingular. Now we can simply apply the Gaussian Elimination to the matrix and find a linearly independent subset of column vectors that allows the Implicit Function Theorem and the CCM algorithm to be applied.

3.4. Accuracy Improvement

A line search in Step 4, whereby a system of nonlinear ordinary differential equations with initial values is to be resolved [15], must be carried out in the implementation of the CCM algorithm. The desired unidimensional direction can be approximated numerically from any line search, but its explicit functional expression can not be calculated. This drawback impedes the output of the points found in any line search system. We present a modification of the CCM algorithm used by Chang and Prabhu [15], which adopted the gradient method to determine the next point without any line search.

Suppose the feasible set $S = \left\{u \in \mathbf{R}^{m+n} \middle| \phi_i(u) = 0, i = 1, \cdots, m\right\}$ satisfies the condition of the Implicit Function Theorem. That is, $\phi_i(u)$ is a holomorphic function in the $m+n$ variables if one treats the $m+n$ variables as complex variables such that one of its Jacobians is nonsingular. Therefore, one can find an m subset of the $m+n$ variables, say $z_1, \cdots, z_m$, so that the corresponding Jacobian matrix $(\partial\phi/\partial z)$ is nonsingular. Furthermore there exist implicit functions $g_j, j = 1, \cdots, m$ in terms of the remaining n variables, say $x_1, \cdots, x_n$, such that $z_j = g_j(x_1, \ldots, x_n)$, $j = 1, \cdots, m$. The original NLP can now be viewed as the following induced NLP:

$$\begin{array}{ll} \text{Maximize} & F(x) \\ \text{Subject to} & x \in U \end{array}$$

where $F(x) = f\left(x, g_1(x), \cdots, g_m(x)\right)$ and U is a neighborhood of the point $x^0 \in \mathbf{R}^n$ that the Implicit Function Theorem holds. Because U contains an open subset of $\mathbf{R}^n$, the induced NLP can be viewed as a locally non-constrained NLP. One important benefit is that moving along the induced gradient direction $D = (\partial F/\partial x)$ will stay in U if the distance is small enough.

One common issue with solving an NLP using the gradient method is that it is likely to leave S by travelling along the gradient direction of a feasible point. This causes a big problem in keeping the NLP feasibility. Applying the CCM algorithm does not present such a problem as every iteration

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2020 doi:10.20944/preprints202010.0527.v1

remains within the feasible region. This is because the gradient of the inducted objective function F with regard to the selected $x_1,\cdots,x_n$, will locally move inside the feasible set U, if we selected carefully. When using the CCM algorithm in a small-scale NLP, one can conduct a line search along the gradient direction of the induced objective function. The relation between the induced line search and the movement along the original feasible set is illustrated in Figure 1.

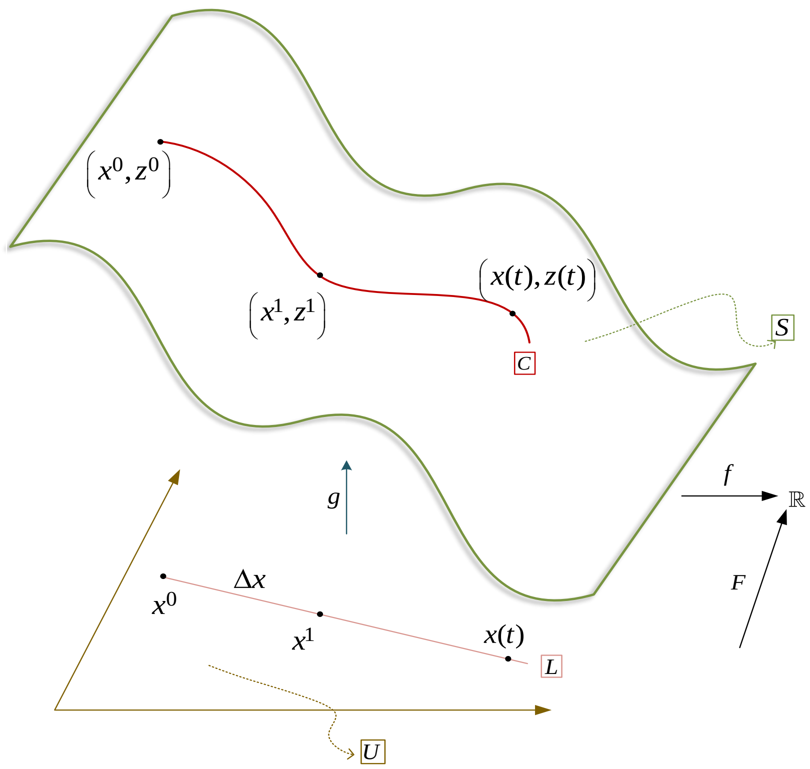

Figure 1. The relationship between induced line search and movement along the original feasible set

Because there is a one to one mapping between U and a neighborhood of $\left(x^0, z^0\right)$ in S, there is a one dimensional curve C in S corresponding to the line $L=\left\{x(t)\middle| t\geq 0\right\}$ in U such that:

$$C=\left\{x(t),z(t)\middle| x(t)\in L, z(t)=g\left(x(t)\right)\right\}.$$

The problem can be viewed as 'lifting' a straight line in $\mathbf{R}^n$ to a curve in $\mathbf{R}^{m+n}$. In performing a line search, one has to find a one-dimensional local optimal point on such a curve with only the knowledge of the projection of the curve while the other coordinates are unknown. Fortunately we also know the explicit objective and constraint functions, so we can approximate the change of the unknown coordinates Δz with the derivatives dz/dt. That is, we can approximate Δz by:

$$\Delta z \sim \left(\frac{dz}{dt}\right)\Delta t.$$

Let $D^0 = D\left(x^0\right) = \left(d_1,\cdots,d_n\right)$. Then:

 doi:10.20944/preprints202010.0527.v1

$$\frac{dz_j}{dt} = \frac{\partial g_j}{\partial x_1}\frac{dx_1}{dt} + \cdots + \frac{\partial g_j}{\partial x_n}\frac{dx_n}{dt}$$

and

$$\left(\frac{dz}{dt}\right) = \left(\frac{\partial g}{\partial x}\right)\begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix} = -\left(\frac{\partial \phi}{\partial z}\right)^{-1}\left(\frac{\partial \phi}{\partial x}\right)\begin{pmatrix} d_1 \\ \vdots \\ d_n \end{pmatrix}.$$

We can now move the previous point along its gradient direction $(\Delta x, \Delta z)$, where $\Delta x = D^0 \Delta t$, D^0 is the exact induced gradient of the induced objective function in the projection space, and Δz is the change in z by the above approximation. We can choose Δt carefully to avoid any line search. There are two reasons to avoid a line search.

First, it requires the knowledge of the 'lifted' curve for any $z > 0$. When an NLP is a small scale problem and the corresponding system of ordinary differential equations is possible to solve, the CCM algorithm can locate or approximate the exact lifted curve C at any $t > 0$. To do so, we may need to apply some ODE methods such as the Euler and the Runge-Kutta methods. As t gets larger, the feasibility of the point $(x(t), z(t))$ is likely disappearing.

Second, we use numerical data to approximate the partial derivatives at all the points involved. The ODE problem in Step 4 is a 'point-by-point' case without an explicit global expression for the coefficients $(\partial\phi/\partial x)(\partial F/\partial x)^T$ and $\partial\phi/\partial z$ in it. Thus, it might be impossible to solve for it in practice. We have chosen to avoid any line search. Instead, as mentioned above, we move a point to the next one in its gradient direction $(\Delta x, \Delta z)$ with a chosen Δt. This way, we have control over staying as close to the feasible set as we need. Not only is the feasibility better kept, but the calculation is also reduced since we are not solving for the system of ordinary differential equations globally.

4. Illustrative Example

We applied the WPM to the same problem of multiple-criteria inventory classification problem as reported in the referenced literature [1, 6, 7, 13, 17]. Following Ng [1] and Hadi-Vencheh [13] we considered three criteria for inventory classification: annual dollar usage (ADU), average unit cost (AUC), and lead time (LT) and assumed the importance of the criteria to be, in descending order, ADU, AUC, and LT. All the criteria were positive for the inventory item score

4.1. Quality of Solutions

The 47 inventory items' optimal scores and weights are shown in Table 1. As can be seen, the optimal scores derived by using the CCM algorithm to solve the WPM of multicriteria ABC classification were as good as those derived using LINGO [18], the off-the-shelf optimization software. If we look carefully at the weights derived using LINGO, it is obvious that most items' weights for ADU and AUC are identical and that some items' (4, 25, 27, 30) weights for LT are zero. This is because the primary underlying technique used by LINGO's nonlinear solver is to get to a feasible solution for nonlinear models quickly. The weight values derived by using the CCM better fit the assumption that the criteria are graded in descending, such that $w_{i,1} \geq w_{i,2} \geq \cdots \geq w_{i,J}$ for all

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2020 doi:10.20944/preprints202010.0527.v1

items i. Therefore, the CCM is superior to LINGO in terms of solution quality in this illustrative example.

Table 1. Measures of inventory items, the optimal scores and weights.

Item Parameter				LINGO				CCM			
				Objective value	Decision variable			Objective value	Decision variable		
	ADU	AUC	LT	Score	ADU weight	AUC weight	LT weight	Score	ADU weight	AUC weight	LT weight
1	5840.64	49.92	2	8.92	7.050E-01	7.050E-01	7.770E-02	8.92	7.050E-01	7.049E-01	7.790E-02
2	5670	210	5	10.02	6.979E-01	6.979E-01	1.606E-01	10.02	6.980E-01	6.978E-01	1.608E-01
3	5037.12	23.76	4	8.38	6.974E-01	6.974E-01	1.654E-01	8.38	6.974E-01	6.973E-01	1.658E-01
4	4769.56	27.73	1	8.34	7.071E-01	7.071E-01	0.000E+00	8.34	7.072E-01	7.071E-01	2.360E-04
5	3478.8	57.98	3	8.71	7.015E-01	7.015E-01	1.262E-01	8.71	7.015E-01	7.014E-01	1.264E-01
6	2936.67	31.24	3	8.15	7.007E-01	7.007E-01	1.347E-01	8.15	7.007E-01	7.006E-01	1.350E-01
7	2820	28.2	3	8.05	7.005E-01	7.005E-01	1.364E-01	8.05	7.005E-01	7.004E-01	1.367E-01
8	2640	55	4	8.52	6.977E-01	6.977E-01	1.627E-01	8.52	6.977E-01	6.976E-01	1.630E-01
9	2423.52	73.44	6	8.73	6.921E-01	6.921E-01	2.051E-01	8.73	6.921E-01	6.920E-01	2.053E-01
10	2407.5	160.5	4	9.20	6.990E-01	6.990E-01	1.507E-01	9.20	6.991E-01	6.989E-01	1.509E-01
11	1057.2	5.12	2	6.12	7.026E-01	7.026E-01	1.133E-01	6.12	7.025E-01	7.024E-01	1.145E-01
12	1043.5	20.87	5	7.24	6.894E-01	6.894E-01	2.222E-01	7.24	6.894E-01	6.893E-01	2.226E-01
13	1038	86.5	7	8.30	6.874E-01	6.874E-01	2.346E-01	8.30	6.875E-01	6.873E-01	2.347E-01
14	883.2	110.4	5	8.28	6.936E-01	6.936E-01	1.944E-01	8.28	6.938E-01	6.934E-01	1.945E-01
15	854.4	71.2	3	7.87	7.002E-01	7.002E-01	1.397E-01	7.87	7.003E-01	7.000E-01	1.399E-01
16	810	45	3	7.51	6.995E-01	6.995E-01	1.463E-01	7.51	6.996E-01	6.994E-01	1.465E-01
17	703.68	14.66	4	6.68	6.917E-01	6.917E-01	2.075E-01	6.68	6.917E-01	6.916E-01	2.081E-01
18	594	49.5	6	7.49	6.866E-01	6.866E-01	2.391E-01	7.49	6.867E-01	6.865E-01	2.393E-01
19	570	47.5	5	7.39	6.902E-01	6.902E-01	2.177E-01	7.39	6.902E-01	6.900E-01	2.178E-01
20	467.6	58.45	4	7.36	6.944E-01	6.944E-01	1.885E-01	7.36	6.946E-01	6.942E-01	1.887E-01
21	463.6	24.4	4	6.74	6.920E-01	6.920E-01	2.056E-01	6.74	6.920E-01	6.919E-01	2.058E-01
22	455	65	4	7.41	6.946E-01	6.946E-01	1.871E-01	7.41	6.948E-01	6.944E-01	1.873E-01
23	432.5	86.5	4	7.57	6.952E-01	6.952E-01	1.830E-01	7.57	6.954E-01	6.949E-01	1.832E-01
24	398.4	33.2	3	6.80	6.978E-01	6.978E-01	1.616E-01	6.80	6.979E-01	6.977E-01	1.618E-01
25	370.5	37.05	1	6.74	7.071E-01	7.071E-01	0.000E+00	6.74	7.071E-01	7.071E-01	2.570E-04
26	338.4	33.84	3	6.70	6.975E-01	6.975E-01	1.640E-01	6.70	6.975E-01	6.975E-01	1.642E-01
27	336.12	84.03	1	7.25	7.071E-01	7.071E-01	0.000E+00	7.25	7.071E-01	7.071E-01	2.860E-04
28	313.6	78.4	6	7.37	6.859E-01	6.859E-01	2.431E-01	7.37	6.859E-01	6.859E-01	2.433E-01
29	268.68	134.34	7	7.67	6.840E-01	6.840E-01	2.537E-01	7.67	6.840E-01	6.840E-01	2.539E-01
30	224	56	1	6.67	7.071E-01	7.071E-01	0.000E+00	6.67	7.071E-01	7.071E-01	2.880E-04
31	216	72	5	7.01	6.882E-01	6.882E-01	2.295E-01	7.01	6.882E-01	6.882E-01	2.297E-01
32	212.08	53.02	2	6.63	7.032E-01	7.032E-01	1.045E-01	6.63	7.032E-01	7.032E-01	1.048E-01
33	197.92	49.48	5	6.69	6.864E-01	6.864E-01	2.404E-01	6.69	6.864E-01	6.864E-01	2.406E-01

 doi:10.20944/preprints202010.0527.v1

34	190.89	7.07	7	5.46	6.606E-01	6.606E-01	3.567E-01	5.46	6.606E-01	6.606E-01	3.595E-01
35	181.8	60.6	3	6.67	6.975E-01	6.975E-01	1.647E-01	6.67	6.975E-01	6.975E-01	1.649E-01
36	163.28	40.82	3	6.32	6.963E-01	6.963E-01	1.738E-01	6.32	6.963E-01	6.963E-01	1.740E-01
37	150	30	5	6.16	6.826E-01	6.826E-01	2.612E-01	6.16	6.826E-01	6.826E-01	2.613E-01
38	134.8	67.4	3	6.54	6.971E-01	6.971E-01	1.680E-01	6.54	6.971E-01	6.971E-01	1.683E-01
39	119.2	59.6	5	6.47	6.849E-01	6.849E-01	2.486E-01	6.47	6.849E-01	6.849E-01	2.488E-01
40	103.36	51.68	6	6.33	6.782E-01	6.782E-01	2.831E-01	6.33	6.782E-01	6.782E-01	2.833E-01
41	79.2	19.8	2	5.25	7.009E-01	7.009E-01	1.321E-01	5.25	7.009E-01	7.009E-01	1.323E-01
42	75.4	37.7	2	5.67	7.018E-01	7.018E-01	1.223E-01	5.66	7.018E-01	7.018E-01	1.227E-01
43	59.78	29.89	5	5.53	6.765E-01	6.765E-01	2.908E-01	5.53	6.765E-01	6.765E-01	2.910E-01
44	48.3	48.3	3	5.59	6.933E-01	6.933E-01	1.964E-01	5.59	6.933E-01	6.933E-01	2.000E-01
45	34.4	34.4	7	5.37	6.590E-01	6.590E-01	3.625E-01	5.37	6.590E-01	6.590E-01	3.639E-01
46	28.8	28.8	3	4.88	6.889E-01	6.889E-01	2.252E-01	4.88	6.889E-01	6.889E-01	2.291E-01
47	25.38	8.46	5	4.12	6.510E-01	6.510E-01	3.903E-01	4.12	6.510E-01	6.510E-01	3.931E-01

Next, the maximal overall scores were sorted in descending order and inventory classification was conducted based on the WPM (shown in Table 2). For comparison purposes, we maintained the same distribution of class A, B and C items as in studies in the cited literature [1, 13, 17]; that is, 10 class A, 14 class B, and 23 class C items. The ABC analysis using the Ng [1], Hadi-Vencheh [13], and Zhou and Fan models [17] is also shown in Table 2. Ten items (8, 29, 15, 16, 27, 33, 39, 40, 34 and 45) did not have the same classification in the WPM model as in the Ng, HV and ZF models. The difference in classification was due to the difference in score computation' schemes. Of the 10 class A items identified in the WPM, only item 8 was recognized as a class B item in the Ng, ZF and HV models. Moreover, in these models item 29 was classified as a group A item while the WPM re-classified it as a class B item. Comparing items 29 and 8, item 8 was superior to item 29 in terms of ADU value (>>). Although item 29 outperformed item 8 in AUC (<) and LT (<), the differences were not significant; therefore, based on the most important consideration of the value of annual consumption of inventory items (ADU) in a year, the WPM provided a more reasonable classification.

In regards to the 14 class B items in the HV-model, eight items (6, 7, 23, 18, 19, 28, 12 and 31) were retained in class B when the WPM was adopted, five of the class B items (33, 39, 40, 34 and 45) were re-classified as C, while the remaining one (item 8) was moved up to the class A. Out of the 23 class C items, 18 items were retained as such, whereas the remaining five (15, 16, 22, 20 and 27) were moved up to class B. Items 33, 39, 40, 34, and 45, classified as class B items in Ng, HV and ZF models but re-classified as class C items using the WPM (see Table 3), had relatively higher LT measures (more than 4), but lower performance in terms of AUC and ADU, the two more important criteria. However, the maximum ADU value (197.92) of these five items was much less than the minimum ADU value (336.12) of items 15, 16 and 27, re-classified as class C items by the WPM, while their AUC is about even. Therefore, the WPM provided a more reasonable ranking of items.

Table 2. Comparison of ABC classification using optimal WPM, ZF-model, Ng-model and HV-model inventory score.

Item	Optimal Score (CCM)	ADU	AUC	LT	WPM Model (CCM)	WPM Model (LINGO)	HV-model	Ng-model	ZF-model
2	10.0222	5670	210	5	A	A	A	A	A
10	9.20134	2407.5	160.5	4	A	A	A	A	A

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2020 doi:10.20944/preprints202010.0527.v1

1	8.92424	5840.64	49.92	2	A	A	A	A	A
9	8.73406	2423.52	73.44	6	A	A	A	A	A
5	8.70633	3478.8	57.98	3	A	A	A	A	B
8	8.51791	2640	55	4	A	A	B	B	B
3	8.38316	5037.12	23.76	4	A	A	A	A	A
4	8.33829	4769.56	27.73	1	A	A	A	A	C
13	8.29587	1038	86.5	7	A	A	A	A	A
14	8.28055	883.2	110.4	5	A	A	A	B	A
6	8.15406	2936.67	31.24	3	B	B	B	A	C
7	8.05394	2820	28.2	3	B	B	B	B	C
15	7.86615	854.4	71.2	3	B	B	C	C	C
29	7.67051	268.68	134.34	7	B	B	A	A	A
23	7.57315	432.5	86.5	4	B	B	B	B	B
16	7.50776	810	45	3	B	B	C	C	C
18	7.49247	594	49.5	6	B	B	B	B	A
22	7.40991	455	65	4	B	B	C	C	B
19	7.39402	570	47.5	5	B	B	B	B	B
28	7.36954	313.6	78.4	6	B	B	B	B	A
20	7.35514	467.6	58.45	4	B	B	C	C	B
27	7.24598	336.12	84.03	1	B	B	C	C	C
12	7.24393	1043.5	20.87	5	B	B	B	B	B
31	7.01167	216	72	5	B	B	B	B	B
24	6.79949	398.4	33.2	3	C	C	C	C	C
21	6.74367	463.6	24.4	4	C	C	C	C	C
25	6.73618	370.5	37.05	1	C	C	C	C	C
26	6.69892	338.4	33.84	3	C	C	C	C	C
33	6.69389	197.92	49.48	5	C	C	B	B	B
17	6.67996	703.68	14.66	4	C	C	C	C	C
30	6.67213	224	56	1	C	C	C	C	C
35	6.67164	181.8	60.6	3	C	C	C	C	C
32	6.63135	212.08	53.02	2	C	C	C	C	C
38	6.53696	134.8	67.4	3	C	C	C	C	C
39	6.47356	119.2	59.6	5	C	C	B	B	B
40	6.32774	103.36	51.68	6	C	C	B	B	B
36	6.32156	163.28	40.82	3	C	C	C	C	C
37	6.16169	150	30	5	C	C	C	C	B
11	6.11791	1057.2	5.12	2	C	C	C	C	C
42	5.66488	75.4	37.7	2	C	C	C	C	C
44	5.59151	48.3	48.3	3	C	C	C	C	C
43	5.5337	59.78	29.89	5	C	C	C	C	C

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 October 2020 doi:10.20944/preprints202010.0527.v1

34	5.45517	190.89	7.07	7	C	C	B	B	B
45	5.36813	34.4	34.4	7	C	C	B	B	B
41	5.24818	79.2	19.8	2	C	C	C	C	C
46	4.87682	28.8	28.8	3	C	C	C	C	C
47	4.12265	25.38	8.46	5	C	C	C	C	C

Table 3. The 10 items re-classified using the WPM.

Item	ADU	AUC	LT	WPM	HV-model	Ng-model	ZF-model
8	2640	55	4	A	B	B	B
29	268.68	134.34	7	B	A	A	A
15	854.4	71.2	3	B	C	C	C
16	810	45	3	B	C	C	C
27	336.12	84.03	1	B	C	C	C
33	197.92	49.48	5	C	B	B	B
39	119.2	59.6	5	C	B	B	B
40	103.36	51.68	6	C	B	B	B
34	190.89	7.07	7	C	B	B	B
45	34.4	34.4	7	C	B	B	B

4.2. Elapsed runtime and Iterations

The efficiency of solving a problem is also an important criterion for algorithm comparisons; a good algorithm should solve problems within acceptable time. This section compares the number of iterations of implementing the CCM and LINGO to solve the WPM of multi criteria ABC classification. The main difference between implementing the CCM algorithm and LINGO solver is that we do not have to specify a starting point or moving step size. When the solution region is a polyhedron, determine the first basic solution (the starting point) would be vital as the local optimal solution is usually located near the basic feasible solution (BFS); the quality of solution is highly related to the location of BFS. An unsuitable starting point would lead to a worse local optimal solution. The search region of an algorithm is related to the step size of search and the search region decides whether a feasible solution can be found, the step size also determines the quality of the final solution. Large step size of line search may “jump over” the optimal solution and whereas smaller search distances may “trap in” and require a significant amount of time to reach the local optimal. A good starting point and step size of search can help reach acceptable solutions in less time. Table 4 illustrates the process of tuning step size in order to reach a feasible solution. The CCM algorithm provides flexibility than other solvers of commercial package software, which means a higher probability of finding better solutions. In this study, using CCM to solve the problem requires more time to achieve the local optimal solution than LINGO. Because the step size of line search determines whether the CCM can find feasible solutions. In this study, LINGO requires only seconds to find a feasible solution, whereas CCM requires a longer time—sometimes nearly a minute.

Table 4. Tuning the step size to reach a feasible solution.

	Step size	$\sum_{j=1}^{J} w_j^2$		Step size	$\sum_{j=1}^{J} w_j^2$
Item 4	0.0003	1.000009	Item 5	0.00045	1.000062
	0.000295	0.999993		0.000448	1.000059
	0.000298	1.000003		0.00044	1.000046

	0.000297	1		0.00041	0.999997
				0.000413	1.000002
				0.000412	1

When looking at the number of iterations each algorithm needs, CCM takes more iterations to solve the problem, even with larger step size. As presented in Table 5, LINGO can solve most problems in only 60 iterations whereas CCM may need approximately one thousand iterations. In conclusion, LINGO is more efficient than CCM in this study, which contradicts the results of the previous study that CCM can solve nonlinear problems more quickly than other software packages. The result of the current study might stem from the fact that the problem in this study is too simple to show the power of CCM.

Table 5. The number of iteration for LINGO and CCM to find the local optima.

item	LINGO	CCM	item	LINGO	CCM	item	LINGO	CCM	Item	LINGO	CCM
1	60	1720	**13**	60	1505	**25**	35	2482	**37**	60	1367
2	60	1866	**14**	55	1852	**26**	60	1646	**38**	50	2369
3	60	1207	**15**	60	1911	**27**	35	3213	**39**	50	1888
4	35	1867	**16**	59	1673	**28**	55	1741	**40**	50	1657
5	60	1627	**17**	60	1090	**29**	50	2024	**41**	58	1903
6	60	1419	**18**	59	1377	**30**	35	3045	**42**	50	2452
7	60	1388	**19**	57	1456	**31**	55	1900	**43**	55	1496
8	60	1495	**20**	60	1729	**32**	55	2397	**44**	75	555
9	58	1428	**21**	60	1287	**33**	55	1645	**45**	55	215
10	57	1995	**22**	55	1802	**34**	60	1158	**46**	75	492
11	60	1086	**23**	55	2009	**35**	50	2223	**47**	60	2712
12	60	1105	**24**	60	1612	**36**	55	1932			

5. Conclusions

In this paper we have presented an extended version of the HV-model to improve multiple criteria ABC inventory classification. Our proposed nonlinear weighted product model (WPM) incorporates multiple criteria with different measure units, without converting the performance of each inventory item in terms of each criterion into a normalized attribute value, an improvement over the model proposed by Hadi-Vencheh. The WPM could also be viewed as providing a more reasonable classification for inventory items from the illustrative example presented and used to compare our model with the HV-model. In this paper, we also presented the improved CCM algorithm for solving the WPM, where nonconvex nonlinearity is present in both the objective function and the constraints. The strategy presented here consisted of greatly reducing the steps in choosing m variables among $(m+n)$ variables, such that the corresponding $m \times m$ Jacobian matrix is nonsingular. In the improved algorithm, we applied the Gaussian elimination to the original matrix to determine which m variables to choose. Our second improvement was to remove solving nonlinear differential equations system that occurs in the line search method of the CCM algorithm. The paper demonstrates an efficient algorithm for solving nonlinear programming problems in which the feasible solution set does not have to be convex. The practical implication of the study is to further improve the efficient nonlinear optimization solver based on the CCM by optimizing the quality of existing solutions, thus improving time and space efficiency.

Future works must continue to investigate the feasibility to implement this proposed CCM algorithm on discrete-domain issues for engineering applications in order to decide if the algorithm could be superior to off-shelf software. Future studies could apply the CCM to other nonlinear programs arisen in practice. For instance, autonomous vehicles is one of many developments that will influence future mobility needs and planning needs. Traffic assignment models seek the same objective as route guidance strategies and provide the turning points with information for implementing control strategies of route guidance. Faster algorithms developed specifically for traffic assignment can be adapted and used in vehicle route guidance systems. The minimization of total travel time is a common goal both globally and from a traffic administration perspective. The current road network manages more traffic by achieving system optimization. Some researchers have focused their efforts on dynamic traffic assignment because of the unrealistic assumptions of static traffic assignment. The difficulties encountered by the dynamic model result from the route calculation being related to the traveling time on arc, which is also dependent on the traffic along the route. It is difficult to solve such relationships analytically in a dynamic circumstance. In response to the difficulties of dynamic traffic modelling, Jahn et al. [19] therefore developed model in which flow represents the traffic patterns in a steady state and the results as the bound for the total travel time. However, Jahn et al.'s [19] algorithm only solves problems with convex nonlinear objective functions and linear constraints. To avoid this restriction, future studies could adopt the CCM to solve nonlinear optimization models and provide strategies for route guidance.

Author Contributions: Conceptualization, Hung-Chieh Chang and Pei-Chun Lin; methodology, Hung-Chieh Chang; software, Hung-Chieh Chang and Pei-Chun Lin; validation, Hung-Chieh Chang.; data curation, Pei-Chun Lin; writing—original draft preparation, Pei-Chun Lin; writing—review and editing, Pei-Chun Lin; visualization, Pei-Chun Lin; funding acquisition, Pei-Chun Lin. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by NAME OF FUNDER, grant number XXX.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ng, W.L. A simple classifier for multiple criteria abc analysis. Eur J Oper Res. 2007, 177, 344-353.
2. Flores, B.E.; Whybark, D.C. Multiple Criteria ABC Analysis. Int J Oper Prod Man. 1986, 6, 38-46.
3. Teunter, R.H; Babai, M.Z.; Syntetos, A.A. ABC classification: service levels and inventory costs. Prod Oper Manag. 2010, 19, 343-352.
4. Harirchian, E.; Jadhav, K.; Mohammad, K.; Aghakouchaki Hosseini, S. E.; Lahmer, T. A Comparative Study of MCDM Methods Integrated with Rapid Visual Seismic Vulnerability Assessment of Existing RC Structures. Appl. Sci., 2020, 10, 6411.
5. Carneiro, J.; Martinho, D.; Alves, P.; Conceição, L.; Marreiros, G.; Novais, P. A Multiple Criteria Decision Analysis Framework for Dispersed Group Decision-Making Contexts. Appl. Sci., 2020, 10, 4614.
6. Flores, B.E.; Olson, D.L.; Dorai, V. Management of multicriteria inventory classification. Math Comput Model. 1992, 16, 71-82.
7. Ramanathan, R. ABC inventory classification with multiple-criteria using weighted linear optimization. Comput Oper Res. 2006, 33, 695-700.
8. Flores, B.E.; Whybark, D.C. Implementing multiple criteria ABC analysis. J Oper Manag. 1987, 7, 79-85.
9. Adida, E.; Perakis, G. A nonlinear continuous time optimal control model of dynamic pricing and inventory control with no backorders. Nav Res Log. 2007, 54, 767-795.
10. Aggarwal, V. A closed-from approach for multi-item inventory grouping. Nav Res Log. 1983, 30, 471-485.
11. Leap, N.J.; Bauer, K.W. A confidence paradigm for classification systems. Nav Res Log. 2011, 58, 236-254.
12. Liang, X.; Yan, H. Inventory control and replenishment with flexible delivery-time upgrade. Nav Res Log. 2014, 61, 418-426.
13. Hadi-Vencheh, A. An improvement to multiple criteria abc inventory classification. Eur J Oper Res. 2010, 201, 962-965.
14. Triantaphyllou, E. Multi-criteria Decision Making Methods: A Comparative Study. Boston, MA. Springer; 2000. Chapter 2, Multi-Criteria Decision Making Methods; p. 5-21.

doi:10.20944/preprints202010.0527.v1

15. Chang, H.-C.; Prabhu, N. Canonical coordinates method for equality-constrained nonlinear optimization. Appl Math Comput. 2003, 140, 135-158.
16. Chang, H.-C.; Lin, P.-C. A demonstration of the improved efficiency of the canonical coordinates method using nonlinear combined heat and power economic dispatch problems. Eng Optimiz. 2014, 46, 261-269.
17. Zhou, P.; Fan, L. A note on multi-criteria abc inventory classification using weighted linear optimization. Eur J Oper Res. 2007, 182, 1488-1491.
18. Lindo Ssystem Inc. Extended lingo (release 8.0) [computer software], Chicago, IL., 2003.
19. Jahn, O.; Möhring, R.; Schulz, A., et al. System-optimal routing of traffic flows with user constraints in networks with congestion. Oper Res. 2005, 53, 600-616.