Multi-criteria logistics distribution network design using SAS/OR

William Ho* and Ali Emrouznejad

Operations and Information Management Group
Aston Business School, Aston University
Birmingham B4 7ET, United Kingdom

E-mail: w.ho@aston.ac.uk; Tel: +44 (0)121 2043342

Abstract

This paper explores the use of the optimization procedures in SAS/OR software with application to the contemporary logistics distribution network design using an integrated multiple criteria decision making approach. Unlike the traditional optimization techniques, the proposed approach, combining analytic hierarchy process (AHP) and goal programming (GP), considers both quantitative and qualitative factors. In the integrated approach, AHP is used to determine the relative importance weightings or priorities of alternative warehouses with respect to both deliverer oriented and customer oriented criteria. Then, a GP model incorporating the constraints of system, resource, and AHP priority is formulated to select the best set of warehouses without exceeding the limited available resources. To facilitate the use of integrated multiple criteria decision making approach by SAS users, an ORMCDM code was implemented in the SAS programming language. The SAS macro developed in this paper selects the chosen variables from a SAS data file and constructs sets of linear programming models based on the selected GP model. An example is given to illustrate how one could use the code to design the logistics distribution network.

Keywords: SAS; SAS/OR; Multiple criteria decision making; Analytic hierarchy process; Goal programming; Logistics distribution network

* Corresponding author.

1. Introduction

The logistics distribution problem is to allocate a number of points of consumption to a number of points of supply, including suppliers, manufacturers, warehouses, distribution centers, and customers. The connection of these various logistics stakeholders by a mean of transportation facilities is regarded as the logistics distribution network. Logistics distribution network design is one of the major decision problems arising in contemporary supply chain management. There are mainly two inadequacies in the traditional approaches for the problem. First, a single criterion was focused only. The objective was either to minimize the total logistics cost (Su, 1998; Wasner and Zäpfel, 2004; Hwang, 2005) or total delivery time (Su, 1999). Second, only quantifiable data were considered in the optimization techniques. Some qualitative factors, which are mainly customer oriented, were not considered.

Multiple criteria decision making (MCDM) techniques have been used in recent years. One of the most prevalent techniques is analytic hierarchy process (AHP) (Ho, 2008). Some researchers (Korpela and Lehmusvaara, 1999; Korpela et al., 2001a–b; 2002) applied the combined AHP-mixed integer linear programming (MILP) model approach for the network design, whereas another group of researchers (Chan and Chung, 2004a–b; 2005; Chan et al., 2004; 2005; 2006) applied the combined AHP-genetic algorithm (GA) approach to solve the problem. For the combined AHP-MILP approach, the selection of distribution network was simply based on the customer satisfaction priorities instead of minimizing the total logistics cost or maximizing the total profit. Therefore, it is believed that the selected distribution network may not be cost effective. For the combined AHP-GA approach, the evaluation criteria used in AHP are all quantitative such as total cost, total delivery day, effectiveness of capacity utilization for warehouses, and so on. Some qualitative factors such as flexibility of capacity and value-added services were neglected. These factors are crucial in the integrated logistics system because they affect the customer satisfaction directly.

To overcome the drawbacks, this paper develops a systematic and prominent MCDM technique, combining AHP and goal programming (GP), to design an optimal logistics distribution network. The combined AHP-GP approach considers both quantitative and qualitative factors and also aims at maximizing the benefits of deliverer and customers.

SAS is recognized as one of the lead packages for statistical analysis and as a powerful tool for data base systems in many organizations, both in public and private sectors. SAS users come from every major industry (banking to pharmaceuticals, manufacturing to telecommunications, and so on). All with the same basic needs to make better strategic decisions and to gain a competitive edge (Emrouznejad, 2005).

There are many applications in SAS that the users recognized as powerful tools in organizational management. For example, the SAS/OR system has numerous optimization procedures which handle the standard problems such as linear programming (LP) and nonlinear programming (NLP) with all types of constraints. In addition to the standard procedures available in SAS system applications such as neural network, simulation, and control project management are introduced.

This paper aims to introduce a new application in SAS system for yielding an optimal solution to a multi-criteria logistics distribution problem by solving the combined AHP-GP model.

2. MCDM techniques

MCDM techniques are generally divided into two categories: multiple attribute decision making (MADM) and multiple objective decision making (MODM). MADM techniques aim at selecting the best solution from a population of feasible alternatives which characterized by multiple attributes. One of the commonly used MADM techniques is AHP (Saaty, 1980). MODM techniques are a special extension of linear programming. A model is defined as a linear programming when the single objective function and the constraints involve linear expressions, and the decision variables are continuous. But, in MODM techniques, multiple objective functions are incorporated into the model simultaneously. GP (Charnes and Cooper, 1961) is an example of the MODM techniques.

In this paper, we apply AHP to evaluate the relative importance weightings of alternative warehouses with respect to some qualitative-based evaluating criteria. Then, a GP model is formulated to select the optimal set of warehouses while considering the AHP priorities of warehouses and the quantitative-based limitations of resources.

2.1. Prioritization of warehouses using AHP

In this paper, five criteria are proposed to evaluate the performance of warehouses. They include total lead time, reliability of order fulfillment, quality, flexibility of capacity, and value-added services. Total lead time comprises the time of handling inventory in warehouses, the time of storing/loading inventory in warehouses, and the time of delivering products from warehouses to customers. Reliability of order fulfillment consists of the accuracy of quantity fulfillment, the accuracy of due date fulfillment, and reliability of delivery time. Quality involves the commitment of deliverer to provide high-quality products and the condition of products received by customers. Flexibility of capacity refers to the ability of warehouses to respond to fluctuation in volume of customer orders. Value-added services refer to any activities that facilitate customers (e.g., track-and-trace and 24-hour customer hotline) and the responsiveness of warehouses to customer special requests (e.g., secure packaging and urgent delivery).

The first step of AHP for evaluating the performance of warehouses is to develop a hierarchy of the decision problem as shown in Figure 1. After constructing the hierarchy, two criteria are compared at a time with respect to the goal. Once the pairwise comparisons have been made for the five criteria, each alternative warehouse is compared against each other alternative with respect to the corresponding criterion at a time. After completion of all pairwise comparisons, the relative priority of each criterion (from Table 1) and each alternative (from Table 2) are synthesized. The judgments are acceptable because the consistency ratios are all below the maximum 0.10 level. The overall priority ranking of warehouses is: $wp_1 = 0.446$, $wp_2 = 0.180$, $wp_3 = 0.239$, and $wp_4 = 0.135$. The AHP priorities are used to determine the priority level of the AHP priority constraints in the GP model.

2.2. Logistics distribution network design in GP

Consider a typical logistics distribution network which consists of m warehouses denoted as $i = \{1, 2, \ldots, m\}$ and n customers denoted as $j = \{1, 2, \ldots, n\}$. Each warehouse has a maximum throughput, Q_i, a minimum throughput, q_i, a fixed cost, fc_i, and a unit inventory holding cost, hc_i. Each customer has a unique order volume, D_j. When warehouse i is assigned to serve customer j, it costs dc_{ij} dollars per unit for delivery. If the total amount of products assigned to warehouse i (i.e., $\sum_j x_{ij}, \forall i$) is less than q_i, this is regarded as impractical allocation because it is not cost-effective to set up a warehouse for processing only a few orders. To avoid low effectiveness of warehouse utilization, penalty cost, pc_i, is considered in the model, which is incurred if $0 < \sum_j x_{ij} < q_i$. The problem here is to determine an optimal distribution network, which refers to the allocation of orders to the best warehouses.

In the model, there are four types of decision variables:

x_{ij} = amount of products delivered from warehouse i to customer j

$$u_i = \begin{cases} 1 & \text{if total allocation of products to warehouse } i \text{ is less than } q_i \\ 0 & \text{otherwise} \end{cases}$$

$$v_i = \begin{cases} 1 & \text{if warehouse } i \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

$$w_i = \begin{cases} 1 & \text{if both } u_i \text{ and } v_i \text{ equal to one} \\ 0 & \text{otherwise} \end{cases}$$

In the model, there are three types of constraints: system constraints, resource constraints, and AHP priority constraints. System constraints are ordinary linear programming constraints, in which there is no deviation variable. This type of constraints cannot be violated, and thus they are called hard constraints. Resource constraints are goal equations or soft constraints, in which there are deviation variables. AHP priority constraints are akin to resource constraints. In this type of constraints, there are deviation variables of which the priority levels are dependent on the overall AHP priority ranking.

Standard multi-criteria logistics distribution model

System constraints

$$\sum_i v_i \le m \qquad (1)$$

$$\sum_j x_{ij} + Mu_i \ge q_i \qquad \forall i \qquad (2)$$

$$\sum_j x_{ij} - Mv_i \le 0 \qquad \forall i \qquad (3)$$

$$w_i - u_i - v_i = -1 \qquad \forall i \qquad (4)$$

Resource constraints

Priority 1 (P_1):

$$\sum_j x_{ij} - d_i^+ + d_i^- = Q_i \qquad \forall i \qquad (5)$$

$$\sum_i x_{ij} - d_{j+m}^+ + d_{j+m}^- = D_j \qquad \forall j \qquad (6)$$

Priority 2 (P_2):

$$\sum_i \sum_j (hc_i + dc_{ij}) x_{ij} + \sum_i fc_i v_i - d_{m+n+1}^+ + d_{m+n+1}^- = TC \qquad (7)$$

Priority 3 (P_3):

$$\sum_i pc_i w_i - d^+_{m+n+2} + d^-_{m+n+2} = 0 \qquad (8)$$

AHP priority constraints

Priorities 4 to 4 + m – 1 (P_4 to P_{4+m-1}):

$$v_i - d^+_{m+n+2+i} + d^-_{m+n+2+i} = 1 \qquad \forall i \qquad (9)$$

Objective function

$$\text{Minimize } z = \sum_k P_k\left(d_r^+ + d_r^-\right) + \sum_k P_k\left(d_s^+ + d_s^-\right) \qquad \forall r, \forall s \qquad (10)$$

Constraint set (1) ensures that the number of warehouses selected must be equal to or less than the number of warehouses available. Constraint set (2) determines which warehouse(s) has/have allocation of products that less than minimum warehouse throughput. Constraint set (3) determines which warehouse(s) is/are selected. Constraint set (4) determines which warehouse(s) incur(s) penalty cost. Constraint set (5) allocates products to warehouses while the amount must not exceed the maximum warehouse throughput. Constraint set (6) allocates products to warehouses while the amount must equal to that demanded by the customers. Constraint set (7) ensures that the total cost, including the inventory holding cost, delivery cost, and fixed cost associated with warehouse selection, must not exceed the targeted amount. Constraint set (8) ensures that allocation of products to warehouses incurring penalty cost is not allowed. Constraint set (9) is to select warehouse *i*. Objective function (10) is to minimize the total deviations from the goals.

3. ORMCDM: SAS code for multi-criteria logistics distribution network design

The ORMCDM introduced in this paper provides a powerful decision making tool for yielding an optimal logistics distribution network design. To enhance the model there are several parameters. The user can select the desired parameters according to the particular model that is required. Users familiar with SAS can add their own features. Users not familiar with SAS need only to run the program with their model specification prior to running the system.

The ORMCDM requires an initial data set that contains the name of variable and data file for warehouses and customers. The data is saved as SAS data set. SAS has the ability also to read from a text (Tab delimited) or Excel format. The program has the ability to accommodate unlimited number of warehouses and customers. The only limitation is the memory of computer used to run the ORMCDM.

The ORMCDM software then converts data set to a GP model. Based on the data and parameters specified in the ORMCDM, the code first creates the usual linear program, then use “PROC LP” to solve the model. The results will then be transferred to report files. The ORMCDM produces a table listing values of decision variables and deviation variables. It also supplies a report summarizing the status of goal achievement at each stage (or priority level).

In the rest of this paper the procedure of implementation of ORMCDM together with an example are explained. Figure 2 illustrates the data flow in the ORMCDM.

4. Definition of terms and typographical conventions

In the rest of this paper and particularly in the ORMCDM code we will see several types of styles used. Style conventions are summarized below:

- `Courier font`: is used to show example of SAS statements. In most cases, this paper uses lowercase type for SAS code. SAS users can enter their own SAS code in lowercase, uppercase or a mixture of the two. Enter any titles and footnotes exactly as you want them to appear on the printout.
- _Underscore_: Variable name that are surround by "_" are specifically used as parameters to the SAS. In all case these variables must be used without any change.
- _Underscore: Variable name that are started with "_" are specifically used as parameters to the ORMCDM.

The ORMCDM runs four macros for data handling (%data), model building (%model1 and %model2), and report writing (%report).

5. Illustration of ORMCDM

This section presents a simple example of four warehouses and seven customers for illustration of ORMCDM. The resource data includes:

- QQi: maximum throughput of warehouse *i*.
- qi: minimum throughput of warehouse *i*.
- hci: unit inventory holding cost associated with warehouse *i*.
- fci: fixed cost associated with warehouse *i*.
- pci: penalty cost associated with warehouse *i*.
- cost: unit delivery cost from warehouse *i* to customer *j*.
- demand: unique order volume requested by the customers.

6. Data handling (%data)

This part of ORMCDM prepares the data to a suitable format that can be used in PROC LP. The ORMCDM requires one data set containing name of "warehouses", name of "customers", "cost matrix", and "demand". The data set should be a ".txt" file, which is saved as "text tab delimited". The warehouses' and customers' names must start with a letter and may contain up to 50 characters. The variable names must be listed in the first row of the data file. The warehouses' names should be listed in the first column. Note that the demand should be given in the last row of the data set. Other information, including QQi, qi, hci, fci, and pci should also be given in the data set, but the order of the columns is not important. An example of data file is:

File: data.txt

Wareh	Cust1	Cust2	Cust3	Cust4	Cust5	Cust6	Cust7	QQi	qi	hci	fci	pci
w1	1	1	2	4	4	3	6	30000	6000	5	30000	7500
w2	2	6	9	3	7	8	4	26000	5200	3	25000	6250
w3	8	4	3	6	3	2	4	22000	4400	3	20000	5000
w4	8	8	9	3	5	7	2	18000	3600	2	15000	3750
demand	12000	9000	10000	8000	6000	11000	7000					

There is only one parameter prior to calling the data macro:

- _data: indicates the name and location of the data file.

SAS procedure for data handling is presented in Appendix A.1.

7. Model building (%model)

This part of ORMCDM calls PROC LP to solve the model. There are two SAS macros for model building. Macro %model1 creates a suitable data set that can be used for PROC LP. There are four parameters prior to calling the model macro:

- _nWareh: defines the number of warehouses.
- _nCustomer: defines the number of customers.
- _TotalCost: defines the total cost.
- _myupperbnd: defines the upper bound value for integer variables.

Macro %model2 defines seven data sets that could be used to solve the multi-criteria logistics distribution problem in seven stages. At each stage, a data set is created, then PROC LP solves the model and results are saved in three data sets for future analysis. These three output data sets are for primal solution, dual solution, and tabulated solution of each stage. There are two parameters before calling this macro.

- _myLarge: defines an optional large number.
- _nIterations: defines an optional value for number of integer iterations.

The parameter _myLarge is an optional parameter that is needed for LP, and in this case, we set it to 100000.

SAS procedure for models %model1 and %model2 building are presented in Appendices A.2 and A.3, respectively.

8. Report writing (%report)

The outputs from ORMCDM include one report for each stage. This report contains all information regarding the summary and solution of each stage of the problem, in this case, seven stages. All these information are also saved in SAS data sets. The users can define appropriate names for each of these data sets before calling %ormcdm macro as follows.

- _outprimal: identifies the name of the SAS output file for primal solution.
- _outdual: identifies the name of the SAS output file for dual solution.
- _outtable: identifies the name of the SAS output file for tabulated solution.

Another parameter needs to be set before calling this macro is the _title as follows:

- _title: gives a title in the output of the SAS.

SAS procedure for report writing is presented in Appendix A.4.

9. ORMCDM macro (%ormcdm)

To make the system as ease as possible, the “%ormcdm macro” put all the above code together.

```
* A SAS macro for the multi-criteria logistics distribution network design;
%macro ormcdm;
 %data;
 %model1;
 %do i=1 %to &_nWareh+3;    * Number of stages in GP model;
 %model2(&i);
 %report(&i);
 %end;
%mend ormcdm;
```

In the above code, the “%ORMCDM macro” is used to manage all previously explained codes, including data handling, model building, and report writing. To get the result, user needs to set up the parameters and run only one statement:

```
%ORMCDM;
```

10. Instruction of how to use ORMCDM macro

This section presents SAS code for the earlier example of the multi-criteria logistics distribution problem with four warehouses and seven customers. The data is saved in file “data.txt”.

A user needs to set the parameters as required and run the following code.

```
options nodate;
%let _title='An example of MCDM';
%let _dataMCDM='c:/sasor/data.txt';
%let _data='c:\sasor\MCDM.txt';
%let _nWareh=4;
%let _nCustomer=7;
%let _TotalCost=425000;
%let _outprimal=outprimal;
%let _outdual=outdual;
%let _outtable=outtable;
%let _myLarge=100000;
%let _nIterations=100000;
%let _myupperbnd=12000;
%ormcdm;
```

The above code manages to get the result based on the specified parameters and the cost matrix saved in the text file. The above code also produces a macro variable (_ORLP_) at termination. Users can examine the result of this macro variable, examine whether PROC LP ran correctly, and examine what error or difficulty it encountered. Summary of information, including the objective value at optimum level and status of _ORLP_ can be seen in the log file as shown in Figure 3.

11. Sample results from ORMCDM macro: output from SAS

In the GP model, there are 40 integral decision variables, 34 deviation variables, 30 constraints, and seven goals (or stages). Some of the results of running the above code are presented in Figure 4 and Figure 5, which summarize the solutions in stage 5 and stage 6, respectively. The solution, shown in Figure 4, is feasible because the allocation does not exceed the maximum throughput of warehouses, does satisfy the volume requirement of customers, does not exceed the total cost budget, and does not incur any penalty cost. When stage 6 was found to be unachievable (i.e., dN16 or d_{16}^{-} = 1), shown in Figure 5, the optimization process was terminated. So, the solution, satisfying the first five priority levels (i.e., P_1 to P_5), is an optimal solution of the problem, shown in Figure 4. The values of decision variables v_i show that three warehouses were selected, including warehouse 1 (v_1 = 1), warehouse 3 (v_3 = 1), and warehouse 4 (v_4 = 1). The total cost spent in setting up these three warehouses, holding inventory in the warehouses, and delivering products from the warehouses to their assigned customers is $425000 with no slack. Besides, the total penalty cost incurred is zero.

12. Conclusions

Nowadays, there is a plenty of software that can be used for operations research. The major reason why we selected SAS is because it has various optimization tools that can be used in a wide range of problems in operations research. In particular, optimization procedures in SAS/OR are exposed to the user in a variety of places such as “PROC LP” and “PROC NLP”. ORMCDM as introduced in this paper is a new application in SAS/OR that is a powerful decision making tool for designing an optimal logistics distribution network. The ORMCDM application implemented in this paper has no limitation on the number of nodes (warehouses and customers). The only limitation is the memory and disk space of the computer used. Besides, SAS has strong data management capabilities that can handle very large datasets efficiently, and it can work with multiple datasets simultaneously. In cases where SAS users want to solve another problem or change the data, they can simply revise a single data set file and run a single statement, %ormcdm. The ORMCDM report’s files can also directly feed to other SAS routines for further analysis.

References

Chan FTS, Chung SH. Multi-criteria genetic optimization for distribution network problems. International Journal of Advanced Manufacturing Technology 2004a; 24; 517–532.

Chan FTS, Chung SH. A multi-criterion genetic algorithm for order distribution in demand driven supply chain. International Journal of Computer Integrated Manufacturing 2004b; 17; 339–351.

Chan FTS, Chung SH, Wadhwa S. A heuristic methodology for order distribution in a demand driven collaborative supply chain. International Journal of Production Research 2004; 42; 1–19.

Chan FTS, Chung SH, Wadhwa S. A hybrid genetic algorithm for production and distribution. Omega 2005; 33; 345–355.

Chan FTS, Chung SH. Multicriterion genetic optimization for due date assigned distribution network problems. Decision Support Systems 2005; 39; 661–675.

Chan FTS, Chung SH, Choy KL. Optimization of order fulfillment in distribution network problems. Journal of Intelligent Manufacturing 2006; 17; 307–319.

Charnes, A. and Cooper, W. W. (1961), *Management Models and Industrial Applications of Linear Programming*, John Wiley & Sons, New York.

Emrouznejad A. Measurement efficiency and productivity in SAS/OR. Computers & Operations Research 2005; 32, 1665–1683.

Ho W. Integrated analytic hierarchy process and its applications – a literature review. European Journal of Operational Research 2008; 186, 211–228.

Hwang HS. An integrated distribution routing model in multi-supply center system. International Journal of Production Economics 2005; 98; 136–142.

Korpela J, Lehmusvaara A. A customer oriented approach to warehouse network evaluation and design. International Journal of Production Economics 1999; 59; 135–146.

Korpela J, Lehmusvaara A, Tuominen M. Customer service based design of the supply chain. International Journal of Production Economics 2001a; 69; 193–204.

Korpela J, Kyläheiko K, Lehmusvaara A, Tuominen M. The effect of ecological factors on distribution network evaluation. International Journal of Logistics: Research and Applications 2001b; 4; 257–269.

Korpela J, Kyläheiko K, Lehmusvaara A, Tuominen M. An analytic approach to production capacity allocation and supply chain design. International Journal of Production Economics 2002; 78; 187–195.

Saaty TL. The Analytic Hierarchy Process. McGraw-Hill: New York; 1980.

Su CT. Locations and vehicle routing designs of physical distribution systems. Production Planning & Control 1998; 9; 650–659.

Su CT. Dynamic vehicle control and scheduling of a multi-depot physical distribution system. Integrated Manufacturing Systems 1999; 10; 56–65.

Wasner M, Zäpfel G. An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. International Journal of Production Economics 2004; 90; 403–419.

Appendix A

A.1. SAS procedure for data handling

```
* The data handling macro;
%macro data;
* Import text tab delimited data file to SAS data file;
 proc import
  datafile=&_dataMCDM
  out=dataC
  dbms=tab
  replace;
  getnames=yes;
 run;
%mend data;
```

A.2. SAS procedure for model building (%model1)

```
* The model building macro;
%macro model1;
data  _null_;
FILE &_data;

array Customer(&_nCustomer)  Cust1-Cust&_nCustomer;
array NCustom(&_nCustomer) $;
array NWareh(&_nWareh) $;
array QQ(&_nWareh);
array q(&_nWareh);
array hc(&_nWareh);
array fc(&_nWareh);
array pc(&_nWareh);

array Demand(&_nCustomer);
array cost(&_nWareh, &_nCustomer);
array u(&_nCustomer);
ds='xxxxx';
myst='xxxxxxxxx';
do i1=1 to &_nWareh;
Link ReadDataC;
NWareh(i1)=Wareh;
do j1=1 to &_nCustomer;
cost(i1,j1)=Customer[j1];
end;
QQ[i1]=QQi;
q[i1]=qi;
hc[i1]=hci;
fc[i1]=fci;
pc[i1]=pci;
end;

Link ReadDataC;
do j1=1 to &_nCustomer;
Demand(j1)=Customer[j1];
end;

myst='_row_ ';
put myst @@;
myst0='X';
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
   ds=myst0||put(i,1.)||put(j,1.);
   put ds @@;
end;
end;
do i=1 to &_nWareh;
   ds='w'||put(i,1.); put ds @@;
   ds='u'||put(i,1.); put ds @@;
   ds='v'||put(i,1.); put ds @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i<10 then ds='dN'||put(i,1.); else  ds='dN1'||put(i-10,1.);put ds
@@;
   if i<10 then ds='dP'||put(i,1.); else  ds='dP1'||put(i-10,1.);put ds
@@;
end;
put '_type_ _rhs_';
```

```
myst='binary';
put myst @@;
myInt=0;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
put '. ' @@;
end;
end;
do i=1 to &_nWareh;
    myInt=myInt+1;
   put myInt @@;
   myInt=myInt+1;
   put myInt @@;
   myInt=myInt+1;
   put myInt @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i<=&_nWareh+&_nCustomer then mycoefP=1; else mycoefP=0;
   if &_nWareh<i<=&_nWareh+&_nCustomer then mycoefN=1; else mycoefN=0;
   put '. ' @@;
   put '. ' @@;
end;
put 'binary .';

myst='integer';
put myst @@;
myInt=0;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
myInt=myInt+1;
put myInt @@;
end;
end;
do i=1 to &_nWareh;
    put '. ' @@;
    put '. ' @@;
    put '. ' @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i<=&_nWareh+&_nCustomer then mycoefP=1; else mycoefP=0;
   if &_nWareh<i<=&_nWareh+&_nCustomer then mycoefN=1; else mycoefN=0;
   put '. ' @@;
   put '. ' @@;
end;
put 'integer .';

myst='upperbound';
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
myUpp=&_myupperbnd;
put myUpp @@;
end;
end;
do i=1 to &_nWareh;
    put '. ' @@;
    put '. ' @@;
    put '. ' @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i<=&_nWareh+&_nCustomer then mycoefP=1; else mycoefP=0;
   if &_nWareh<i<=&_nWareh+&_nCustomer then mycoefN=1; else mycoefN=0;
   put '. ' @@;
   put '. ' @@;
end;
put 'upperbd .';

myst='objP1';
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
put '0 ' @@;
end;
end;
do i=1 to &_nWareh;
   put '0 ' @@;
   put '0 ' @@;
   put '0 ' @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i<=&_nWareh+&_nCustomer then mycoefP=1; else mycoefP=0;
   if &_nWareh<i<=&_nWareh+&_nCustomer then mycoefN=1; else mycoefN=0;
   put mycoefN @@;
   put mycoefP @@;
end;
put 'min .';

myst='objP2';
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
put '0 ' @@;
end;
end;
do i=1 to &_nWareh;
   put '0 ' @@;
   put '0 ' @@;
   put '0 ' @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i=&_nWareh+&_nCustomer+1 then mycoef=1; else mycoef=0;
   put '0 ' @@;
   put mycoef @@;
end;
put 'min .';

myst='objP3';
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
 put '0 ' @@;
end;
end;
do i=1 to &_nWareh;
   put '0 ' @@;
   put '0 '  @@;
   put '0 '  @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i=&_nWareh+&_nCustomer+2 then mycoef=1; else mycoef=0;
   put '0 ' @@;
   put mycoef @@;
```

```
end;
put 'min .';

do k=1 to &_nWareh;
myst='objP'||put(&_nWareh+k-1,1.);
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
put '0 ' @@;
end;
end;
do i=1 to &_nWareh;
   put '0 ' @@;
   put '0 '  @@;
   put '0 '  @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i=&_nWareh+&_nCustomer+2+k then mycoef=1; else mycoef=0;
   put mycoef @@;
   put mycoef @@;
end;
put 'min .';
end;

do k=1 to 2*&_nWareh+&_nCustomer+2;
if k<=&_nWareh+&_nCustomer then myst1='P1';
else if k<=&_nWareh+&_nCustomer+1 then myst1='P2';
else if k<=&_nWareh+&_nCustomer+2 then myst1='P3';
else myst1='P'||put(k-(&_nCustomer+3),1.);
if (myst1='P1' and 1<=k<=4) or (myst1='P2') or (myst1='P3') then;
else do;
   if k<10 then myst='dZrN'||myst1||put(k,1.);
     else myst='dZrN'||myst1||put(k,2.);
   put myst @@;
   do i=1 to &_nWareh;
   do j=1 to &_nCustomer;
   put '0 ' @@;
   end;
   end;
   do i=1 to &_nWareh;
       put '0 ' @@;
       put '0 ' @@;
       put '0 ' @@;
   end;
   do i=1 to 2*&_nWareh+&_nCustomer+2;
       if i=k then mycoef=1; else mycoef=0;
       put mycoef @@;  put '0 ' @@;
   end;
   put 'eq 0';
   end;
end;

do k=1 to 2*&_nWareh+&_nCustomer+2;
   if k<=&_nWareh+&_nCustomer then myst1='P1';
   else if k<=&_nWareh+&_nCustomer+1 then myst1='P2';
   else if k<=&_nWareh+&_nCustomer+2 then myst1='P3';
   else  myst1='P'||put(k-(&_nCustomer+3),1.);
   if k<10 then myst='dZrP'||myst1||put(k,1.);
     else myst='dZrP'||myst1||put(k,2.);
   put myst @@;
   do i=1 to &_nWareh;
   do j=1 to &_nCustomer;
   put '0 ' @@;
   end;
   end;
   do i=1 to &_nWareh;
       put '0 ' @@;
       put '0 ' @@;
       put '0 ' @@;
   end;
   do i=1 to 2*&_nWareh+&_nCustomer+2;
       if i=k then mycoef=1; else mycoef=0;
       put '0 ' @@;    put mycoef @@;
   end;
   put 'eq 0';
end;

myst='v ';
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
   put '0 ' @@;
end;
end;
do i=1 to &_nWareh;
   put '0 ' @@;
   put '0 ' @@;
   put '1 ' @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   put '0 ' @@;
   put '0 ' @@;
end;
put 'le 4' ;

do k=1 to &_nWareh;
myst='wuv'||put(k,1.);
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
   put '0 ' @@;
end;
end;
do i=1 to &_nWareh;
if i=k then mycoefP=1; else mycoefP=0;
    mycoefN=-mycoefP;
   put mycoefP @@;
   put mycoefN @@;
   put mycoefN @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   put '0 ' @@;
   put '0 ' @@;
end;
put 'eq -1' ;
end;

do k=1 to &_nWareh;
myst='XU'||put(k,1.);
put myst @@;
do i=1 to &_nWareh;
if i=k then mycoef=1; else mycoef=0;
```

```
do j=1 to &_nCustomer;
put mycoef @@;
end;
end;
do i=1 to &_nWareh;
   if i=k then mycoef=&_myLarge; else mycoef=0;
   put '0 ' @@;
   put  mycoef @@;
   put '0 ' @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   put '0 ' @@;
   put '0 ' @@;
end;
put 'ge '  @@;
put q[k];
end;

do k=1 to &_nWareh;
myst='XV'||put(k,1.);
put myst @@;
do i=1 to &_nWareh;
if i=k then mycoef=1; else mycoef=0;
do j=1 to &_nCustomer;
put mycoef @@;
end;
end;
do i=1 to &_nWareh;
   if i=k then mycoef=-&_myLarge; else mycoef=0;
   put '0 ' @@;
   put '0 '  @@;
   put mycoef @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   put '0 ' @@;
   put '0 ' @@;
end;
put 'le '  @@;
put '0';
end;

do k=1 to &_nWareh;
myst='XP1d'||put(k,1.);
put myst @@;
do i=1 to &_nWareh;
if i=k then mycoef=1; else mycoef=0;
do j=1 to &_nCustomer;
put mycoef @@;
end;
end;
do i=1 to &_nWareh;
   put '0 ' @@;
   put '0 ' @@;
   put '0 ' @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i=k then mycoef=1; else mycoef=0;
   mycoefN=-mycoef;
   put mycoef @@;
   put mycoefN @@;
end;
```

```
put 'eq '  @@;
put QQ[k];
end;

do k=&_nWareh+1 to &_nCustomer+&_nWareh;
if k<10 then myst='P1d'||put(k,1.); else myst='P1d'||put(k,2.);
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
if j+&_nWareh=k then mycoef=1; else mycoef=0;
put mycoef @@;
end;
end;
do i=1 to &_nWareh;
   put '0 ' @@;
   put '0 ' @@;
   put '0 ' @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i=k then mycoef=1; else mycoef=0;
   mycoefN=-mycoef;
   put mycoef @@;
   put mycoefN @@;
end;
put 'eq '  @@;
put demand[k-&_nWareh];
end;

myst='P2';
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
mycoef=cost[i,j]+hc[i];
put mycoef @@;
end;
end;
do i=1 to &_nWareh;
   put '0 ' @@;
   put '0 '  @@;
   put fc[i] @@;
end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i=&_nWareh+&_nCustomer+1 then mycoef=1; else mycoef=0;
   mycoefN=-mycoef;
   put mycoef @@;
   put mycoefN @@;
end;
put 'eq '  @@;
mycost=&_TotalCost;
put mycost ;

myst='P3 ';
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
   put '0 ' @@;
end;
end;
do i=1 to &_nWareh;
   put pc[i] @@;
   put '0 ' @@;
   put '0 ' @@;
end;
   do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i=&_nWareh+&_nCustomer+2 then mycoef=1; else mycoef=0;
   mycoefN=-mycoef;
   put mycoef @@;
   put mycoefN @@;
   end;
put 'eq 0';

do k=1 to &_nWareh;
myst='P'||put(&_nWareh+k-1,1.);
put myst @@;
do i=1 to &_nWareh;
do j=1 to &_nCustomer;
put '0 ' @@;
end;
end;
do i=1 to &_nWareh;
if k =1 or k=4 then do;
   if i=k then mycoef=1; else mycoef=0;
   put '0 ' @@;
   put '0 '  @@;
   put mycoef @@;
   end;
if k =2 then do;
   if i=k+1 then mycoef=1; else mycoef=0;
   put '0 ' @@;
   put '0 '  @@;
   put mycoef @@;
   end;
if k =3 then do;
   if i=k-1 then mycoef=1; else mycoef=0;
   put '0 ' @@;
   put '0 '  @@;
   put mycoef @@;
   end;

end;
do i=1 to 2*&_nWareh+&_nCustomer+2;
   if i=&_nWareh+&_nCustomer+2+k then mycoef=1; else mycoef=0;
   mycoefN=-mycoef;
   put mycoef @@;
   put mycoefN @@;
end;
put 'eq '  @@;
put '1';
end;

ReadDataC: set dataC; return;
run;

proc import
  datafile=&_data
  out=dataMCDM
  dbms=dlm
  replace;
  getnames=yes;
 run;
%mend model1;
```

A.3. SAS procedure for model building (%model2)

```
%macro model2(p);
data dataMCDM&p;
set dataMCDM;
myP='objP'||put(&p,1.);
if _type_='min' and _row_ ne myP then delete;
if substr(_row_,1,5)='dZrNP' and substr(_row_,1,6)>='dZrNP'||put(&p,1.)
then delete;
if substr(_row_,1,5)='dZrPP' and substr(_row_,1,6)>='dZrPP'||put(&p,1.)
then delete;
run;

proc lp
   data=dataMCDM&p
   dualout=&_outdual&p
   primalout=&_outprimal&p
   tableauout=&_outtable&p
   imaxit=&_nIterations;

run;
%put &_orlp_;
%mend model2;
```

A.4. SAS procedure for report writing

```
* The report writing macro;
%macro report(p);
title  &_title '(P=' &p ')';
proc print
   data=&_outprimal&p;
proc print
   data=&_outdual&p;
run;
proc print
   data=&_outtable&p;
run;
%mend report;
```

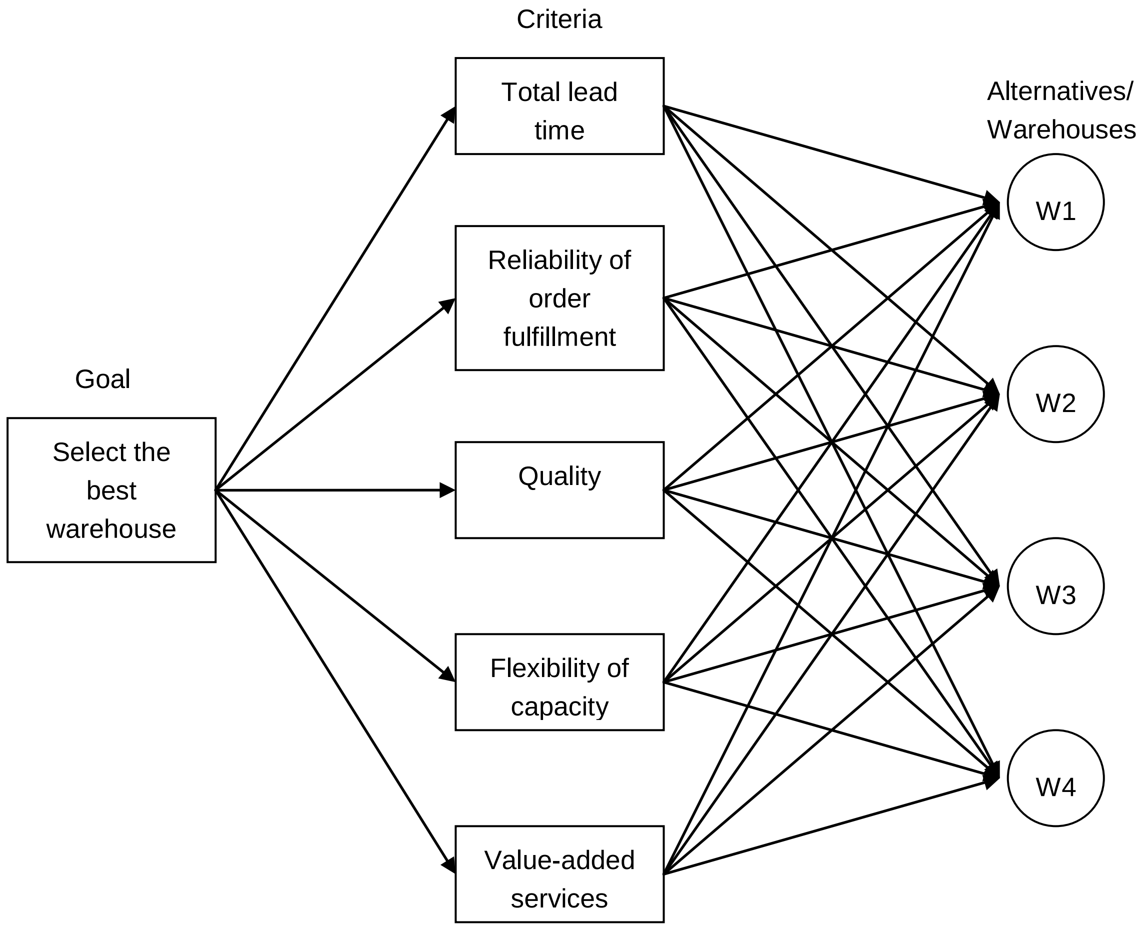


Figure 1: A hierarchy of the warehouse prioritization

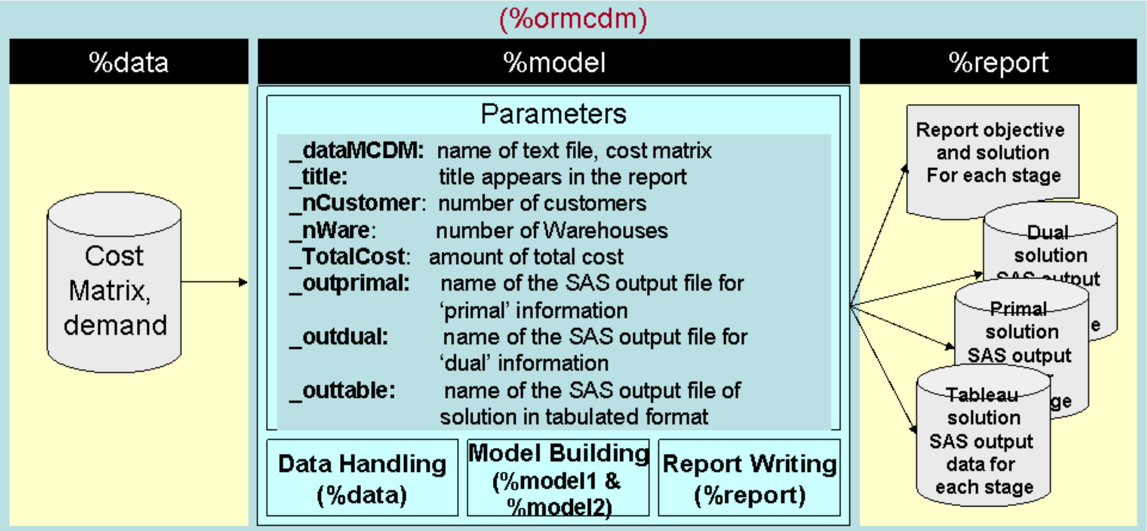
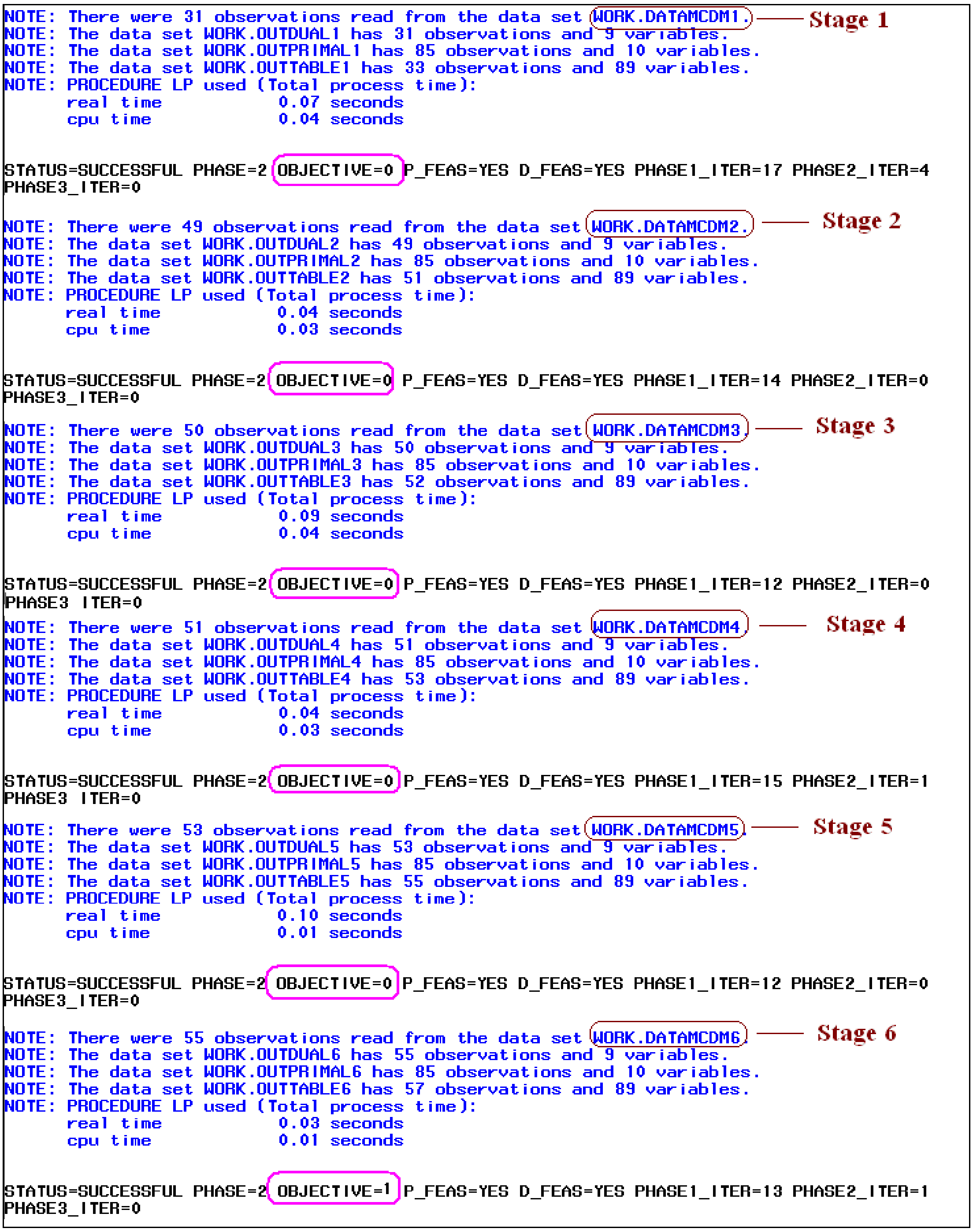
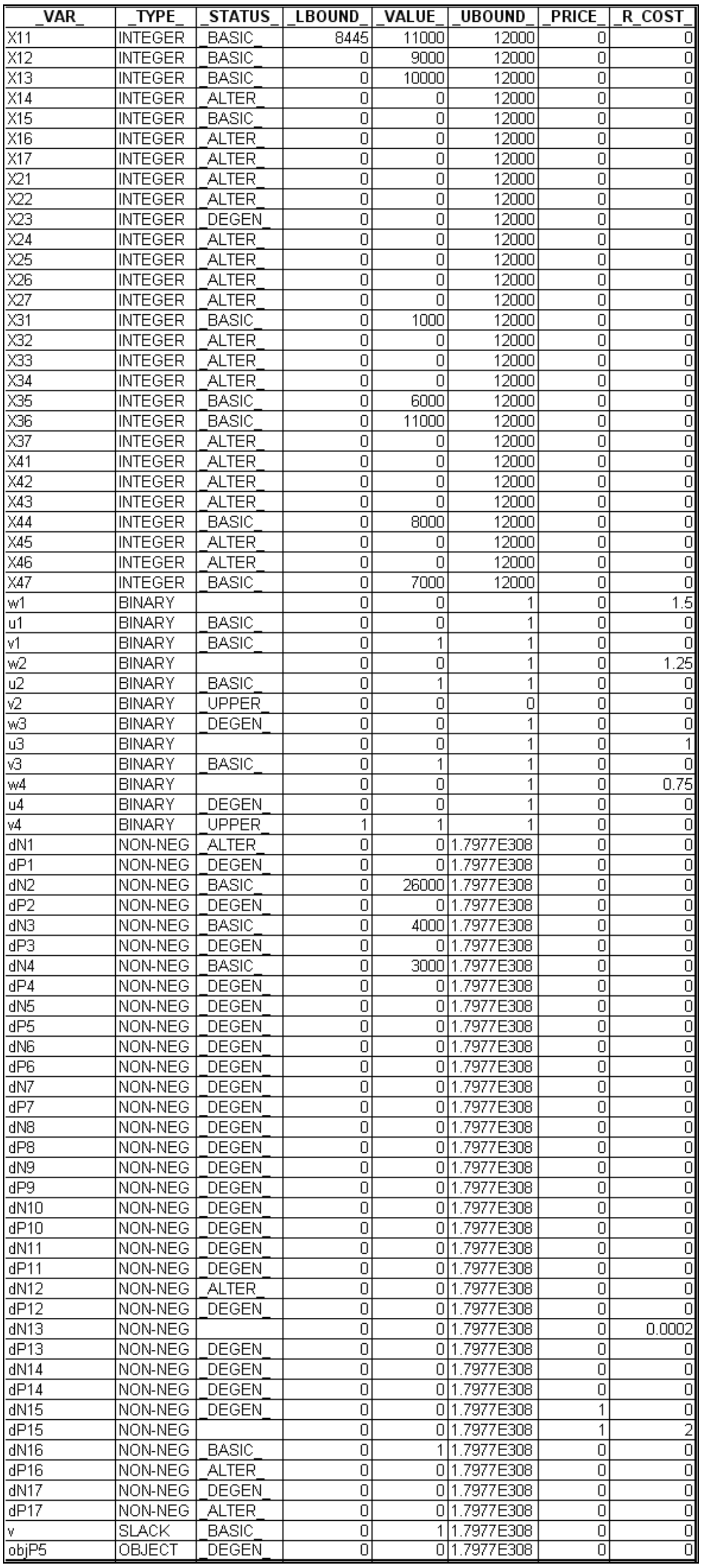

(%ormcdm)
%data
%model
%report
Cost Matrix, demand
Parameters
_dataMCDM: name of text file, cost matrix
_title: title appears in the report
_nCustomer: number of customers
_nWare: number of Warehouses
_TotalCost: amount of total cost
_outprimal: name of the SAS output file for 'primal' information
_outdual: name of the SAS output file for 'dual' information
_outtable: name of the SAS output file of solution in tabulated format
Data Handling (%data)
Model Building (%model1 & %model2)
Report Writing (%report)
Report objective and solution For each stage
Dual solution
e
Primal solution SAS output
ge
Tableau solution SAS output data for each stage

Figure 2: Data flow in ORMCDM



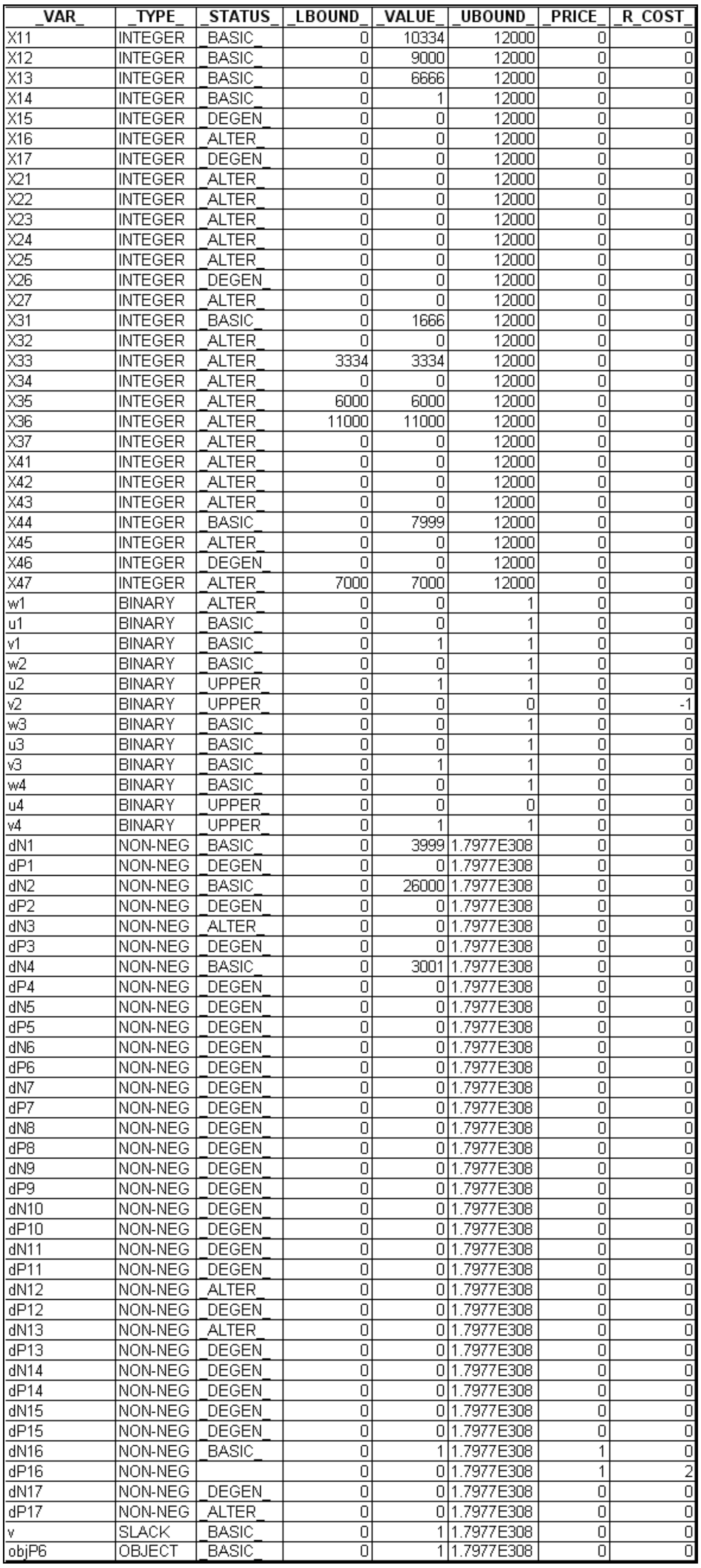
```
NOTE: There were 31 observations read from the data set WORK.DATAMCDM1.
NOTE: The data set WORK.OUTDUAL1 has 31 observations and 9 variables.
NOTE: The data set WORK.OUTPRIMAL1 has 85 observations and 10 variables.
NOTE: The data set WORK.OUTTABLE1 has 33 observations and 89 variables.
NOTE: PROCEDURE LP used (Total process time):
      real time           0.07 seconds
      cpu time            0.04 seconds


STATUS=SUCCESSFUL PHASE=2 OBJECTIVE=0 P_FEAS=YES D_FEAS=YES PHASE1_ITER=17 PHASE2_ITER=4
PHASE3_ITER=0

NOTE: There were 49 observations read from the data set WORK.DATAMCDM2.
NOTE: The data set WORK.OUTDUAL2 has 49 observations and 9 variables.
NOTE: The data set WORK.OUTPRIMAL2 has 85 observations and 10 variables.
NOTE: The data set WORK.OUTTABLE2 has 51 observations and 89 variables.
NOTE: PROCEDURE LP used (Total process time):
      real time           0.04 seconds
      cpu time            0.03 seconds


STATUS=SUCCESSFUL PHASE=2 OBJECTIVE=0 P_FEAS=YES D_FEAS=YES PHASE1_ITER=14 PHASE2_ITER=0
PHASE3_ITER=0

NOTE: There were 50 observations read from the data set WORK.DATAMCDM3
NOTE: The data set WORK.OUTDUAL3 has 50 observations and 9 variables.
NOTE: The data set WORK.OUTPRIMAL3 has 85 observations and 10 variables.
NOTE: The data set WORK.OUTTABLE3 has 52 observations and 89 variables.
NOTE: PROCEDURE LP used (Total process time):
      real time           0.09 seconds
      cpu time            0.04 seconds


STATUS=SUCCESSFUL PHASE=2 OBJECTIVE=0 P_FEAS=YES D_FEAS=YES PHASE1_ITER=12 PHASE2_ITER=0
PHASE3 ITER=0
NOTE: There were 51 observations read from the data set WORK.DATAMCDM4
NOTE: The data set WORK.OUTDUAL4 has 51 observations and 9 variables.
NOTE: The data set WORK.OUTPRIMAL4 has 85 observations and 10 variables.
NOTE: The data set WORK.OUTTABLE4 has 53 observations and 89 variables.
NOTE: PROCEDURE LP used (Total process time):
      real time           0.04 seconds
      cpu time            0.03 seconds


STATUS=SUCCESSFUL PHASE=2 OBJECTIVE=0 P_FEAS=YES D_FEAS=YES PHASE1_ITER=15 PHASE2_ITER=1
PHASE3 ITER=0

NOTE: There were 53 observations read from the data set WORK.DATAMCDM5.
NOTE: The data set WORK.OUTDUAL5 has 53 observations and 9 variables.
NOTE: The data set WORK.OUTPRIMAL5 has 85 observations and 10 variables.
NOTE: The data set WORK.OUTTABLE5 has 55 observations and 89 variables.
NOTE: PROCEDURE LP used (Total process time):
      real time           0.10 seconds
      cpu time            0.01 seconds


STATUS=SUCCESSFUL PHASE=2 OBJECTIVE=0 P_FEAS=YES D_FEAS=YES PHASE1_ITER=12 PHASE2_ITER=0
PHASE3_ITER=0

NOTE: There were 55 observations read from the data set WORK.DATAMCDM6.
NOTE: The data set WORK.OUTDUAL6 has 55 observations and 9 variables.
NOTE: The data set WORK.OUTPRIMAL6 has 85 observations and 10 variables.
NOTE: The data set WORK.OUTTABLE6 has 57 observations and 89 variables.
NOTE: PROCEDURE LP used (Total process time):
      real time           0.03 seconds
      cpu time            0.01 seconds


STATUS=SUCCESSFUL PHASE=2 OBJECTIVE=1 P_FEAS=YES D_FEAS=YES PHASE1_ITER=13 PHASE2_ITER=1
PHASE3_ITER=0
```

Figure 3: Log for %ORMCDM



VAR	_TYPE_	_STATUS_	_LBOUND_	_VALUE_	_UBOUND_	_PRICE_	_R_COST_
X11	INTEGER	_BASIC_	8445	11000	12000	0	0
X12	INTEGER	_BASIC_	0	9000	12000	0	0
X13	INTEGER	_BASIC_	0	10000	12000	0	0
X14	INTEGER	_ALTER_	0	0	12000	0	0
X15	INTEGER	_BASIC_	0	0	12000	0	0
X16	INTEGER	_ALTER_	0	0	12000	0	0
X17	INTEGER	_ALTER_	0	0	12000	0	0
X21	INTEGER	_ALTER_	0	0	12000	0	0
X22	INTEGER	_ALTER_	0	0	12000	0	0
X23	INTEGER	_DEGEN_	0	0	12000	0	0
X24	INTEGER	_ALTER_	0	0	12000	0	0
X25	INTEGER	_ALTER_	0	0	12000	0	0
X26	INTEGER	_ALTER_	0	0	12000	0	0
X27	INTEGER	_ALTER_	0	0	12000	0	0
X31	INTEGER	_BASIC_	0	1000	12000	0	0
X32	INTEGER	_ALTER_	0	0	12000	0	0
X33	INTEGER	_ALTER_	0	0	12000	0	0
X34	INTEGER	_ALTER_	0	0	12000	0	0
X35	INTEGER	_BASIC_	0	6000	12000	0	0
X36	INTEGER	_BASIC_	0	11000	12000	0	0
X37	INTEGER	_ALTER_	0	0	12000	0	0
X41	INTEGER	_ALTER_	0	0	12000	0	0
X42	INTEGER	_ALTER_	0	0	12000	0	0
X43	INTEGER	_ALTER_	0	0	12000	0	0
X44	INTEGER	_BASIC_	0	8000	12000	0	0
X45	INTEGER	_ALTER_	0	0	12000	0	0
X46	INTEGER	_ALTER_	0	0	12000	0	0
X47	INTEGER	_BASIC_	0	7000	12000	0	0
w1	BINARY		0	0	1	0	1.5
u1	BINARY	_BASIC_	0	0	1	0	0
v1	BINARY	_BASIC_	0	1	1	0	0
w2	BINARY		0	0	1	0	1.25
u2	BINARY	_BASIC_	0	1	1	0	0
v2	BINARY	_UPPER_	0	0	0	0	0
w3	BINARY	_DEGEN_	0	0	1	0	0
u3	BINARY		0	0	1	0	1
v3	BINARY	_BASIC_	0	1	1	0	0
w4	BINARY		0	0	1	0	0.75
u4	BINARY	_DEGEN_	0	0	1	0	0
v4	BINARY	_UPPER_	1	1	1	0	0
dN1	NON-NEG	_ALTER_	0	0	1.7977E308	0	0
dP1	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN2	NON-NEG	_BASIC_	0	26000	1.7977E308	0	0
dP2	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN3	NON-NEG	_BASIC_	0	4000	1.7977E308	0	0
dP3	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN4	NON-NEG	_BASIC_	0	3000	1.7977E308	0	0
dP4	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN5	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP5	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN6	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP6	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN7	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP7	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN8	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP8	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN9	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP9	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN10	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP10	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN11	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP11	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN12	NON-NEG	_ALTER_	0	0	1.7977E308	0	0
dP12	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN13	NON-NEG		0	0	1.7977E308	0	0.0002
dP13	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN14	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP14	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN15	NON-NEG	_DEGEN_	0	0	1.7977E308	1	0
dP15	NON-NEG		0	0	1.7977E308	1	2
dN16	NON-NEG	_BASIC_	0	1	1.7977E308	0	0
dP16	NON-NEG	_ALTER_	0	0	1.7977E308	0	0
dN17	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP17	NON-NEG	_ALTER_	0	0	1.7977E308	0	0
v	SLACK	_BASIC_	0	1	1.7977E308	0	0
objP5	OBJECT	_DEGEN_	0	0	1.7977E308	0	0

Figure 4: Result of %ORMCDM, primal output of stage 5



VAR	_TYPE_	_STATUS_	_LBOUND_	_VALUE_	_UBOUND_	_PRICE_	_R_COST_
X11	INTEGER	_BASIC_	0	10334	12000	0	0
X12	INTEGER	_BASIC_	0	9000	12000	0	0
X13	INTEGER	_BASIC_	0	6666	12000	0	0
X14	INTEGER	_BASIC_	0	1	12000	0	0
X15	INTEGER	_DEGEN_	0	0	12000	0	0
X16	INTEGER	_ALTER_	0	0	12000	0	0
X17	INTEGER	_DEGEN_	0	0	12000	0	0
X21	INTEGER	_ALTER_	0	0	12000	0	0
X22	INTEGER	_ALTER_	0	0	12000	0	0
X23	INTEGER	_ALTER_	0	0	12000	0	0
X24	INTEGER	_ALTER_	0	0	12000	0	0
X25	INTEGER	_ALTER_	0	0	12000	0	0
X26	INTEGER	_DEGEN_	0	0	12000	0	0
X27	INTEGER	_ALTER_	0	0	12000	0	0
X31	INTEGER	_BASIC_	0	1666	12000	0	0
X32	INTEGER	_ALTER_	0	0	12000	0	0
X33	INTEGER	_ALTER_	3334	3334	12000	0	0
X34	INTEGER	_ALTER_	0	0	12000	0	0
X35	INTEGER	_ALTER_	6000	6000	12000	0	0
X36	INTEGER	_ALTER_	11000	11000	12000	0	0
X37	INTEGER	_ALTER_	0	0	12000	0	0
X41	INTEGER	_ALTER_	0	0	12000	0	0
X42	INTEGER	_ALTER_	0	0	12000	0	0
X43	INTEGER	_ALTER_	0	0	12000	0	0
X44	INTEGER	_BASIC_	0	7999	12000	0	0
X45	INTEGER	_ALTER_	0	0	12000	0	0
X46	INTEGER	_DEGEN_	0	0	12000	0	0
X47	INTEGER	_ALTER_	7000	7000	12000	0	0
w1	BINARY	_ALTER_	0	0	1	0	0
u1	BINARY	_BASIC_	0	0	1	0	0
v1	BINARY	_BASIC_	0	1	1	0	0
w2	BINARY	_BASIC_	0	0	1	0	0
u2	BINARY	_UPPER_	0	1	1	0	0
v2	BINARY	_UPPER_	0	0	0	0	-1
w3	BINARY	_BASIC_	0	0	1	0	0
u3	BINARY	_BASIC_	0	0	1	0	0
v3	BINARY	_BASIC_	0	1	1	0	0
w4	BINARY	_BASIC_	0	0	1	0	0
u4	BINARY	_UPPER_	0	0	0	0	0
v4	BINARY	_UPPER_	0	1	1	0	0
dN1	NON-NEG	_BASIC_	0	3999	1.7977E308	0	0
dP1	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN2	NON-NEG	_BASIC_	0	26000	1.7977E308	0	0
dP2	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN3	NON-NEG	_ALTER_	0	0	1.7977E308	0	0
dP3	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN4	NON-NEG	_BASIC_	0	3001	1.7977E308	0	0
dP4	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN5	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP5	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN6	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP6	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN7	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP7	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN8	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP8	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN9	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP9	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN10	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP10	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN11	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP11	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN12	NON-NEG	_ALTER_	0	0	1.7977E308	0	0
dP12	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN13	NON-NEG	_ALTER_	0	0	1.7977E308	0	0
dP13	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN14	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP14	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN15	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP15	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dN16	NON-NEG	_BASIC_	0	1	1.7977E308	1	0
dP16	NON-NEG		0	0	1.7977E308	1	2
dN17	NON-NEG	_DEGEN_	0	0	1.7977E308	0	0
dP17	NON-NEG	_ALTER_	0	0	1.7977E308	0	0
v	SLACK	_BASIC_	0	1	1.7977E308	0	0
objP6	OBJECT	_BASIC_	0	1	1.7977E308	0	0

Figure 5: Result of %ORMCDM, primal output of stage 6

Table 1: Priorities of criteria with respect to goal

	C1	C2	C3	C4	C5	Priorities	λ_{max}	CI	RI	CR
C1	1	1	2	3	4	0.315				
C2	1	1	2	3	4	0.315				
C3	1/2	1/2	1	3	4	0.211				
C4	1/3	1/3	1/3	1	1	0.086				
C5	1/4	1/4	1/4	1	1	0.073				
					Total	1.000	5.088	0.022	1.120	0.020

C1: Total lead time; C2: Reliability of order fulfillment; C3: Quality; C4: Flexibility of capacity; C5: Value-added services.

Table 2: Priorities of alternatives with respect to criteria

	W1	W2	W3	W4	Priorities	λ_{max}	CI	RI	CR
(*w.r.t. C1: Total lead time*)									
W1	1	3	3	5	0.518				
W2	1/3	1	1	4	0.214				
W3	1/3	1	1	3	0.196				
W4	1/5	1/4	1/3	1	0.072				
				Total	1.000	4.076	0.025	0.900	0.028
(*w.r.t. C2: Reliability of order fulfillment*)									
W1	1	3	2	4	0.467				
W2	1/3	1	1/2	2	0.160				
W3	1/2	2	1	3	0.277				
W4	1/4	1/2	1/3	1	0.095				
				Total	1.000	4.031	0.010	0.900	0.011
(*w.r.t. C3: Quality*)									
W1	1	3	2	3	0.455				
W2	1/3	1	1/2	1	0.141				
W3	1/2	2	1	2	0.263				
W4	1/3	1	1/2	1	0.141				
				Total	1.000	4.010	0.003	0.900	0.004
(*w.r.t. C4: Flexibility of capacity*)									
W1	1	1/3	1/2	1/5	0.086				
W2	3	1	3	1/2	0.299				
W3	2	1/3	1	1/3	0.140				
W4	5	2	3	1	0.474				
				Total	1.000	4.065	0.022	0.900	0.024
(*w.r.t. C5: Value-added services*)									
W1	1	4	2	3	0.470				
W2	1/4	1	1/3	1	0.114				
W3	1/2	3	1	2	0.280				
W4	1/3	1	1/2	1	0.136				
				Total	1.000	4.031	0.010	0.900	0.010

W1: Warehouse 1; W2: Warehouse 2; W3: Warehouse 3; W4: Warehouse 4.

Author/s:
Ho, W; Emrouznejad, A

Title:
Multi-criteria logistics distribution network design using SAS/OR

Date:
2009-04-01

Citation:
Ho, W. & Emrouznejad, A. (2009). Multi-criteria logistics distribution network design using SAS/OR. EXPERT SYSTEMS WITH APPLICATIONS, 36 (3), pp.7288-7298. https://doi.org/10.1016/j.eswa.2008.09.012.

Persistent Link:
http://hdl.handle.net/11343/118678

File Description:
Accepted version