

RESOURCES SELECTION FOR AGILE/VIRTUAL ENTERPRISES

PART II - PROTOTYPE/DEMONSTRATOR TOOL

A. Castro, Instituto Superior de Engenharia do Porto, ISEP, 1040122@isep.ipp.pt

A. Pires, Polytechnic Institute of Porto, Portugal, ant@isep.ipp.pt

L. Costa, University of Minho, School of Engineering, Portugal, lac@dps.uminho.pt

G. Putnik, University of Minho, School of Engineering, Portugal, putnikgd@dps.uminho.pt

P. Ávila, Polytechnic Institute of Porto, Portugal, psa@isep.ipp.pt

Abstract: The process of creation/development of an A/V E is not simple. Due to the large number of potential resources is important to make a selection of those who give more guarantees in terms of success in this cooperation and bring more value to A/V E. There are several projects of A/V E with different approaches to the selection process. The Virtual Enterprise Architecture Reference Model (BM_VEARM) project incorporates a model of resource selection. Considering the aspects of the model the main objective is the development of a tool for pre-selection of resources in MATLAB, and its integration with an algorithm for the final selection. The result is the tool named *Selec_E A/V_V2* with the following features: Pre-selection of resources without integration of the Value Analysis (VA), with integration of VA and final selection of resources given to those mentioned cases. This tool will support the development of BM_VEARM selection model and will permit the validation of the importance of VA in the resources pre-selection.

Keywords: agile/virtual enterprises, resources selection, resources pre-selection, value analysis, MATLAB

INTRODUCTION

The main objective is to quantify the performance of the pre-selection with the VA integration, according to the model developed. It is intended to validate that the VA incorporation brings quality and quantity benefits to the overall process of resources selection, particularly in the pre-selection phase. In order to test this assumption, a demonstrator was defined and built. This should support a simulation plan of the pre-selection and consequent final selection with and without the VA integration global model. Certain conditions and restrictions were established in order to identify and quantify the domains of the validation of the model, intended to be as comprehensive as possible.

The main goal of this work is the development of a resources pre-selection tool that will be integrated with the selection model, in order to make the resources selection process quicker and more efficient.

PROBLEM PRESENTATION

The main goal of this work is the creation of a data input tool for the pre-selection of resources to an A/V E which should be capable to integrate within a Branch&Bound algorithm for the final selection of resources [1].

After the analysis on the pre-selection subject, the pseudo-code shown below is created, serving as a guideline for the creation of the code in MATLAB language, which will be integrated with the final selection algorithm.

The problem to be solved is the transposition of the selection of resources model with VA integration to MATLAB code, using the pseudo code developed on Part I of this article.

The main steps of the algorithm model are described below in the form of pseudo code and according to the model conceptual framework and its IDEF0 representation and description [2].

Consider:

$PTP_i = \{TP_1, TP_2, \dots, TP_n\}$ - set of tasks of the processing Task Plan;

P - algorithm parameters;

TP_i - processing task, $i = \{1, 2, \dots, n\}$;

$Rps_1(TP_i)$ - set of pre-selection requisites of level 1 for task TP_i ;

$Rps_2(TP_i)$ - set of pre-selection requisites of level 2 for task TP_i ;

Sma - minimum value

$DS_Rps_1(TP_i)$ - solution domain for pre-selection requisites of level 1 for task TP_i ;

$PR_i(TP_i)$ - proposals for each TP_i , $PR_i = \{1, 2, \dots, n\}$;

PI - information search;

$Drp_{ij}(TP_i) = \{rp_{i1}, rp_{i2}, \dots, rp_{in}\}$ - set of candidate resources to pre-selection of the task TP_i ;

rp_{ij} – candidate resource j to pre-selection of the task TP_i ;
F.O. (Rps_1)(TP_i) – objective function of level 1 requisites associated to the pre-selection systems of task TP_i ;
F.O. (Rps_2)(TP_i) – objective function of level 1 requisites associated to the pre-selection systems of task TP_i ;
F.S.Q. – objective function of quality system;
F.S.S. – objective function of financial system;
F.S.S – objective function of synergies system;
Pps(rp_{ij}) – pre-selection parameters of resource j , that candidates to task TP_i
gf – flexibility
Φ – weighting
Global F.O. – global value of resource candidate

Begin

For PTP:

Define P and continue

%----- Qualitative Phase -----%

For each TP_i do

Define $Rps_1 (TP_i)$

Define $Rps_2 (TP_i)$

For each $Rps_1 (TP_i)$

Define Sma or DS and continue

For the PR_i :

PI and continue

%----- Qualitative Phase -----%

For the $Drp_{ij} (TP_i)$ execute:

Evaluate F.O. $Rps_1 (TP_i)$

If $Pps(rp_{ij}) \neq Sma$ dos $Rps_1 (TP_i)$
 or Se $Pps(rp_{ij}) \neq DS_Rps_1$

Then $Sma = Sma + gf$
 (redefine Sma
 with gf)

Evaluate F.O. $Rps_1 (TP_i)$

Pre-Select rp_{ij} and continue

% ----- End Evaluation level 1 -----%

For the $Drp_{ij} (TP_i)$ fulfilling F.O. $Rps_1 (TP_i)$ execute:

Evaluate F.O. $Rps_2 (TP_i)$

ΦRps_2 for each system

Evaluate F.O. $Rps_2 (TP_i)$

for each system
 and continue

$F.S.Q._rp_{ij} = \sum (\Phi SQ_i * PQ_i_{rp_{ij}})$

$F.S.F._rp_{ij} = \sum (\Phi SF_i * PF_i_{rp_{ij}})$

$F.S.S._rp_{ij} = \sum (\Phi SS_i * PS_i_{rp_{ij}})$

Pre-Select rp_{ij} and continue

% --- End Evaluation level 2 per system---%

Determine Global F.O. of rp_{ij} to the TP_i and continue

$\sum ((\Phi SQ_i * PQ_i_{rp_{ij}}) + (\Phi SF_i * PF_i_{rp_{ij}}) + (\Phi SS_i * PS_i_{rp_{ij}}))$

% ----- End Evaluation level 2 -----%

Pre-Select rp_{ij} for the TP_i

End

DEMONSTRATOR TOOL

The language chosen was the MATLAB program because it adjusted to the desired objectives for our demonstrator. MATLAB is a system for scientific calculation that provides a user-friendly environment with an intuitive yet powerful notation which allows the performance of numerical algorithms on arrays with minimal programming. Also, in MATLAB is possible to create and manipulate arrays without prior scaling and manipulation of variables can be performed interactively. Due to its extension and the scope of the work it will not be represented the programming language of the model that transposes the algorithm referred before [3, 4].

MODULES OF THE TOOL

The tool code is divided in modules in order to permit a simpler understanding and a better implementation, giving the program a flexible base to adjust to subsequent developments.

The Base module will be responsible for the initial data inputs and also for the calling of the other modules (pre-selection, selection, and presentation of the results).

In a high level overview, the base module pseudo code can be presented like this:

Asks the input of data

Minimum requirements

Number of Plans

Number of tasks

Weighting values

For each Plan

Calls the pre-selection module

Calls the selection module (without **VA**)

Calls the presentation of results module

Saves the results of the pre-selection and selection modules (without **VA**)

Calls the selection module (considering **VA**)
 Calls the presentation of results module
 Saves the results of the pre-selection and selection modules (without **VA**)
 Rename the storing variables to keep the data of all cycles.

In the same basis of presentation, the pseudo-code of the pre-selection module is the following one:

- For each task of the Plan in question
 - Generates a matrix with random data of the resources
 - Verification if the generated values are within the limits allowed
 - 1st level evaluation
 - Weighting the values
 - Count the number of resources with positive classification in each system of evaluation
 - 2nd level evaluation
 - Presents results
 - Stores the intermediate matrices

Generation of the Processing Costs array (**CP**)

After the creation of the pseudo codes presented above, there were created two flowcharts that allow an easier understanding of the steps that the algorithm will take. They are presented in figure 1 and figure 2, for the base module and pre-selection module respectively. For the selection module, that was already created, the flowchart is the presented in figure 3.

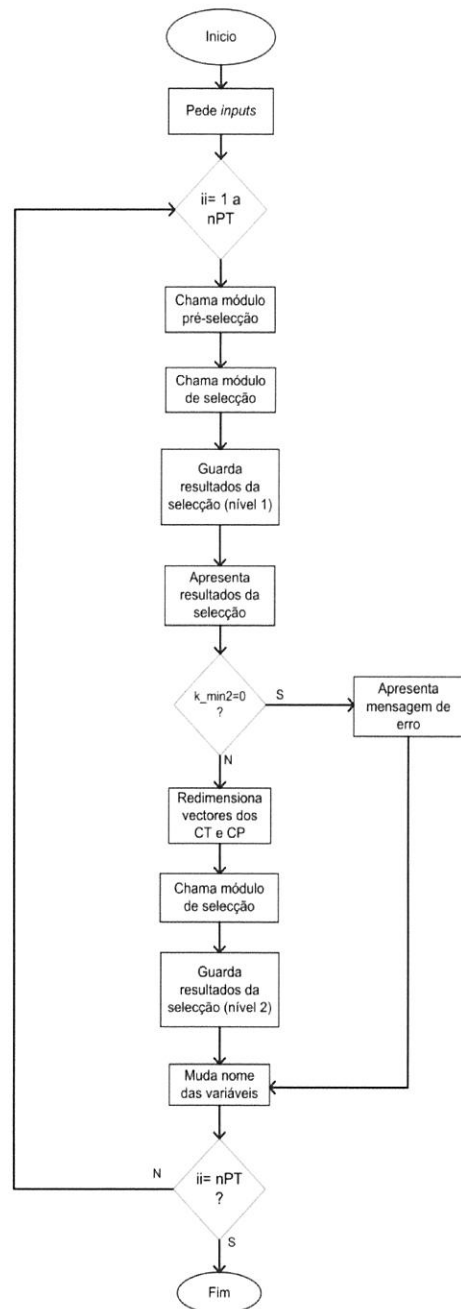


Fig. 1 - Base module flowchart

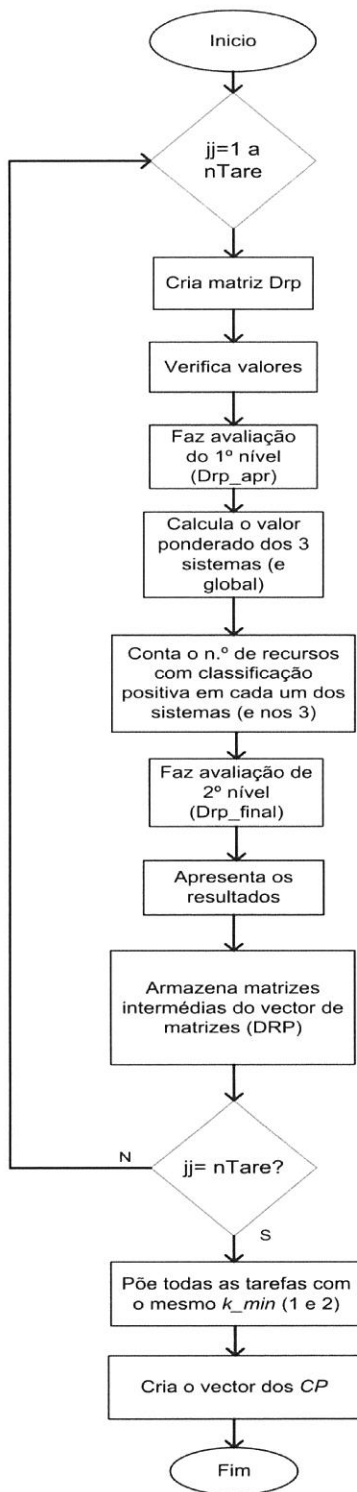


Fig. 2 - Pre-selection flowchart

The selection module will make the generation of transport costs (CT) for the resources that passed the pre-selection and will select the resources based on an algorithm for cost optimization of type Branch and Bound.

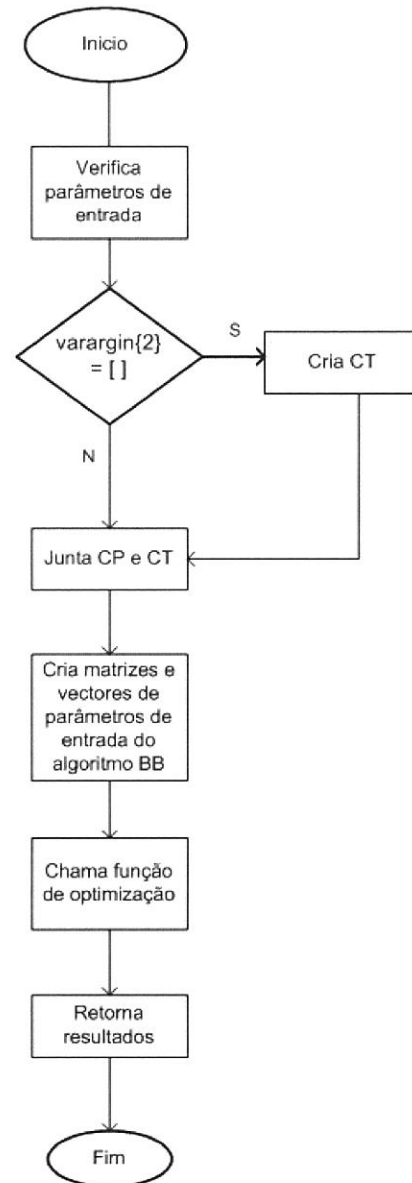


Fig. 3 - Selection flowchart

OUTPUTS OF THE TOOL

The Base module is responsible for the data inputs, calling the other modules, and storing the processed data.

The inputs are the number of Plans, the number of tasks of each Plan and the number of candidate resources for each task. Also on this phase will be inputted the weighting values for each system to be calculated. After this the program enters in self mode, not requiring human intervention.

```

Número de plano de tarefas a analisar: 1

Para o plano de tarefas 1 indique número de tarefas
N.º de tarefas: 4

Plano de tarefas n.º 1
O número de recursos é igual para todas as tarefas? (s/n): s

Introduza o número de recursos candidatos às tarefas: 10

Introduza os valores de ponderação para os 3 sistemas em % (Se todos iguais prima enter):

Introduza os 5 valores de ponderação da qualidade em % (Se todos iguais prima enter):

Introduza os 5 valores de ponderação financeiros em % (Se todos iguais prima enter):

Introduza os 5 valores de ponderação das Sinergias em % (Se todos iguais prima enter):
    
```

Fig. 4 - Inputs

As it can be seen in figure 1, after the inputs, the Base module will call the pre-selection module. In the pre-selection module, figure 2, the process will be task by task. This module will receive entry parameters (the number of tasks of the Plan being analysed, the number of candidate resources that is equal for all tasks of the Plan, the minimum requirements mentioned above and the weighting ratios).

In the first place, a matrix (table 1), called Drp, will be created using distribution functions since in this phase it's only a matter of simulation in order to prove the importance of the Value Analysis (VA) and to make sure that the program works properly. Afterwards it's expected that the values will be obtained throw surveys to the candidate resources.

The values will be verified to make sure that they are within the defined limits (between zero and ten).

The 1st level evaluation of the pre-selection will eliminate the resources that don't achieve the requirements to pass to the next phase. For simulation effects, only those who have the value "1" in all off the 1st level requirements (red box of the table) will pass.

Recurso	Requisitos 1º nivel					Classificação nos requisitos do sistema da Qualidade					Classificação nos requisitos do sistema Financeiro					Classificação nos requisitos do sistema das Sinergias					CP
	1	1	1	1	1	3,9595	5,1060	6,2193	8,2575	5,4201	8,5231	6,9508	8,3482	6,2539	4,6864	3,0372	6,3110	7,6371	5,4148	4,9184	
1	1	1	1	1	1	3,0338	3,9595	5,1060	6,2193	8,2575	5,4201	8,5231	6,9508	8,3482	6,2539	4,6864	3,0372	6,3110	7,6371	5,4148	4,9184
2	1	1	1	1	1	6,0917	5,8724	7,2227	6,2186	9,6280	6,6240	9,6090	4,5538	7,0531	5,4795	7,2003	7,1879	1,6280	3,3459	3,1180	6,8037
3	1	1	1	1	1	3,8967	6,7949	4,9622	9,0325	5,9349	9,2720	5,1499	7,1789	5,8744	1,9561	4,0357	7,2250	5,8902	3,7625	4,7472	6,4990
4	1	1	1	1	1	3,8403	6,3984	2,9579	4,5527	4,8135	6,8027	7,8843	6,6010	5,2539	7,6310	7,5978	6,2404	7,1425	6,8256	4,0261	7,5191
5	1	1	1	1	1	5,7364	7,1907	8,0937	5,6041	6,6554	5,5234	6,4592	6,8800	4,7663	6,5497	7,2022	6,1846	9,4597	4,7829	4,5259	2,5002
6	0	1	1	1	1	6,3674	6,5816	6,2259	6,8799	6,2033	10,0000	3,6667	3,7186	3,8133	5,1328	5,6631	5,5629	7,0827	6,7785	7,5025	
7	0	1	1	1	1	7,8039	2,3287	6,1335	6,0710	10,0000	5,8616	4,9854	6,4716	6,4916	6,1401	4,7828	3,5548	6,6330	3,3143	3,9356	8,6624
8	1	1	1	1	1	8,0587	5,3099	8,0256	7,2587	5,7440	4,2686	3,9138	5,4599	5,1237	7,9671	5,4046	8,2874	4,9368	7,9451	4,9555	
9	1	1	1	1	1	10,0000	7,9020	5,1360	7,9979	5,2798	7,4118	8,8317	2,7910	8,0577	8,9159	6,0949	9,4925	6,3108	3,5258	1,6130	5,3332
10	1	1	1	1	1	6,2880	2,7227	4,4798	4,3624	7,0395	5,9717	3,6889	5,9810	4,6204	4,6666	7,7283	6,2268	6,7967	7,7679	6,3605	7,1017

Table 1 data

Before the 2nd level evaluation, the five evaluation criteria of each considered system must be weighted in order to obtain a classification for each system and the global weighting (ratio between the three systems), that can be seen in table 2. In the 2nd level evaluation, only those who obtain positive classification (five or more) on the three systems will pass, table 3.

Being

r_{ij} ; ($j = 1, n$), ($i = 1, k$): candidate resource j for task i

F.S.Q.: Objective Function of Quality System

ΦSQ_i ; ($i = 1$ a 5): Weighting of requirements i of Quality System

PQ_{i_rij} : parameter of candidate resource j to task i for the requirements Q_i .

$$(\Phi SQ1 * PQ1_rij + \Phi SQ2 * PQ2_rij + \Phi SQ3 * PQ3_rij + \Phi SQ4 * PQ4_rij + \Phi SQ5 * PQ5_rij) = F.S.Q._rij .$$

Doing the substitution by the table 1 values, and having in consideration that the ratios are equal to all the criteria of the system we obtain,

$$(0,2 * 3,0338 + 0,2 * 3,9595 + 0,2 * 5,1060 + 0,2 * 6,2193 + 0,2 * 8,2575) = 5,3152 .$$

Which proves that the calculus is being well performed.

Table 2 – Matrix Drp_final , with resource's evaluation (1st level and systems weighting)

Recurso	V.S.Q.	V.S.F.	V.S.S.	C.Pond	CP
1	5,3152	7,0992	5,4173	5,9439	4,9184
2	7,0067	6,6639	4,4960	6,0555	6,8037
3	6,0311	5,8862	5,1322	5,6832	6,4990
5	6,6560	6,0357	6,4311	6,3743	2,5002
8	6,8454	4,7897	6,9082	6,1811	4,9555
9	7,1231	7,2016	5,4074	6,5774	5,3332
10	4,9785	4,9857	6,9761	5,6467	7,1017

Table 3 – Matrix Drp_final , with resource's evaluation (1st and 2nd level)

Recurso	V.S.Q.	V.S.F.	V.S.S.	C.Pond	CP
1	5,3152	7,0992	5,4173	5,9439	4,9184
3	6,0311	5,8862	5,1322	5,6832	6,4990
5	6,6560	6,0357	6,4311	6,3743	2,5002
9	7,1231	7,2016	5,4074	6,5774	5,3332

The lines of the matrix will be ordered by the global classification of the resources (5th column)

After that, the presentation of the results of this phase and the storage of all the matrixes (initial, intermediate and final) in a cell array, DRP , figure 5.

Fara a Tarefa 1

Referente ao nível 1 do algoritmo (sem avaliação do valor dos recursos)

Candidataram-se 10 recursos, dos quais foram aprovados 7 no nível 1 do algoritmo de pré-seleção

Referente ao nível 2 do algoritmo (com avaliação do valor dos recursos)

6 recursos tiveram nota positiva no S.O.
 5 recursos tiveram nota positiva no S.F.
 6 recursos tiveram nota positiva no S.S.
 4 recursos tiveram nota positiva nos 3 sistemas

Recurso	V.S.O.	V.S.F.	V.S.S.	C.P.
9	7,1231	7,2016	5,4074	6,5774
5	6,6560	6,0357	6,4311	6,3743
1	5,3152	7,0992	5,4173	5,9439
3	6,0311	5,8862	5,1322	5,6832

Fig. 5 - Results presentation (task 1)

The program will then redo all this cycle until it analyses all the tasks of the plan. After the analysis of all the tasks, the program will normalise the number of resources that passes on each level. For example, let's consider that a given Plan have 4 tasks. The number of candidate resources is the same for all the tasks. Let's assume that for the 1st task, the number of resources that passes the 1st level is eight. For the 2nd and 3rd tasks, the number of resources approved is eleven and for the 4th is six. In this example, and making the normalization, all the 4 tasks will have the same number of resources approved considering the minimum of all four, which means that the number of resources approved is going to be six.

The same concept is applied to the 2nd level analyses.

$DRP =$

[10x21 double]	[10x21 double]	[10x21 double]	[10x21 double]
[7x17 double]	[6x17 double]	[7x17 double]	[6x17 double]
[7x6 double]	[6x6 double]	[7x6 double]	[6x6 double]
[6x6 double]	[6x6 double]	[6x6 double]	[6x6 double]
[4x6 double]	[3x6 double]	[4x6 double]	[5x6 double]
[3x6 double]	[3x6 double]	[3x6 double]	[3x6 double]

Fig. 6 - Another example of the normalization of the number of resources in both levels (At red, the resources approved at each level and at blue the normalization)

The next step is the creation of the Processing Costs array (CP). To do this, the program will access to the last column of the matrix of each task and copy the values to a new array. This array (CP) will be one of the entry parameters for the selection module.

```

>> FT(4,1)
ans =
    9.0000    7.1231    7.2016    5.4074    6.5774    5.3332
    5.0000    6.6560    6.0357    6.4311    6.3743    2.5002
    8.0000    6.8454    4.7897    6.8082    6.1811    4.9555
    2.0000    7.0067    6.6639    4.4960    6.0555    6.8037
    1.0000    5.3152    7.0992    5.4173    5.9439    4.9184
    3.0000    6.0311    5.8862    5.1322    5.4832    6.4990

>> FT(4,2)
ans =
    4.0000    4.8691    6.4578    7.5724    6.2998    8.1269
    9.0000    4.9902    6.3518    5.3414    6.2308    2.8990
   10.0000    5.4388    6.6651    6.0950    6.0663    6.8022
    5.0000    5.1126    5.6815    5.9594    5.5825    3.2295
    6.0000    7.0753    4.9980    4.4014    5.5883    7.5591
    3.0000    3.9220    6.2440    6.2969    5.4877    2.7454

>> CP
CP =
Columns 1 through 10
    5.3332    2.5002    4.9555    6.8037    4.9184    6.4990    8.1269    2.8990    6.8022    3.2295
Columns 11 through 20
    7.5591    2.7454    4.3805    3.8938    6.8184    5.5909    5.3534    7.0817    5.1015    4.7628
    
```

Fig. 7 - Creation of the Processing Costs array

In the example of figure 7, are presented two tasks. The values surrounded in the 1st matrix by the red line consist in the processing costs of the resources of the 1st task that passed the level 1 evaluation. They are the first ones of the CP array that is being showed in the last part of figure 7, also surrounded by a red line. In a similar way, the values of the 2nd matrix, surrounded by a blue line, represents the processing costs for the 2nd task and are the second ones to be part of the array.

The pre-selection module will return to the base module the cell array with all the matrixes and the CP array.

Back to the base module, the number of resources that passes the 1st level of the pre-selection, *k_min1*, will be checked and will be another of the entry parameters of the selection module, as the CP array and the number of tasks of the Plan in analysis.

The outputs of the selection module (without Value Analysis) will be the array *x1*, which will allows us to identify the selected resources and the related Transportation Costs (CT), the variable *fx1* which will give us the total cost of the system, *exitflag1* to know if it's the optimal solution or not and a structure, *output1*, with different fields as the number of iterations realized to achieve the solution and the time used, among others.

It's inside this module that the CT is created for the first time with the dimensions of (n.º of tasks-1)**k_min1***k_min1*. After that, an array, *f*, will be created, being the binding of the CP and CT arrays.

One of the outputs will be the array *x1*, which has the dimensions of the referred array *f*, being in this case a Boolean type. This means that the

array *x1* will be filled only with ones or zeros and the ones means approval (selected resource). Having in consideration the case that is being presented where *k_min1* is six the CP array has a dimension of 24 (*ntasks***k_min1*). This means that each "1" found on the firsts 24 positions of *x1* array corresponds to a Processing Cost that is related to a resource that has been selected. The "1" found in the remaining positions are the Transportation Costs between the selected resources. This means that the first "1" found after the first 24 positions is the Transportation Cost between the 1st and the 2nd task, the second is the cost between the 2nd and 3rd task and so on.

Using a function of MATLAB (*find*) we can easily obtain the positions of *x1* array that where filled with "1". Therefore is created a new array, *locali*, which contains the positions where the "1" are located, figure 8.

```

>> locali'
ans =
    2     8    14    24    32    68   128
    
```

Fig. 8 - Locali array

elements, (8, 14 and 24) it's necessary to subtract the (number of task minus 1) multiplied by *k_min1*. Using the example, for the second task, we need to subtract 1**k_min1* to the value 8, obtaining the value 2. This value corresponds to the 2nd resource of the 2nd task matrix (column 2, line 4 of cell array DRP). For the 4th number, the selected resource will be the number 6 of the matrix (24 - 3*6 = 6). The positions of the resources at their respective positions can be seen on figure 9.

After the positions of each selected resource being found, the number of the resource, the final classification (global weighting) and CP associated will be stored in the 1st cell of another cell array, *valor*. At each iteration the position of the resource is identified and the data is copied to the 1st cell of the cell array *valor*, figure 10.

```
>> DRP(4,1)

ans =

    9.0000    7.1231    7.2016    5.4074    6.5774    5.3332
    5.0000    6.6560    6.0357    6.4311    6.3743    2.5002
    8.0000    6.8454    4.7897    6.9082    6.1811    4.9555
    2.0000    7.0067    6.6639    4.4960    6.0555    6.8037
    1.0000    5.3152    7.0992    5.4173    5.9439    4.9184
    3.0000    6.0311    5.8862    5.1322    5.6832    6.4990

>> DRP(4,2)

ans =

    4.0000    4.8691    6.4578    7.5724    6.2998    8.1269
    9.0000    6.9982    6.3518    5.3414    6.2305    2.8990
    10.0000    5.4388    6.6651    6.0950    6.0663    6.8022
    5.0000    5.1126    5.6815    5.9534    5.5825    3.2295
    6.0000    7.0753    4.9980    4.6014    5.5583    7.5591
    3.0000    3.9220    6.2440    6.2969    5.4877    2.7454

>> DRP(4,3)

ans =

    7.0000    7.3641    6.5484    6.1106    6.6743    4.3805
    10.0000    6.2799    7.7712    4.5040    6.1851    3.8938
    5.0000    6.4435    5.9543    5.9823    6.1267    6.8184
    6.0000    6.2969    5.7011    6.0554    6.0178    5.5909
    8.0000    6.6797    5.6699    5.2741    5.8746    5.3534
    2.0000    4.6783    5.6014    6.6621    5.6473    7.0817

>> DRP(4,4)

ans =

    8.0000    6.2636    6.1886    6.7580    6.4034    5.1015
    9.0000    5.9717    6.4221    6.0769    6.1569    4.7628
    5.0000    6.2932    5.9528    5.6323    5.9594    7.1062
    10.0000    6.2918    6.2620    5.2899    5.9479    4.6905
    1.0000    6.2242    5.2225    5.6303    5.6923    6.5391
    6.0000    4.8990    5.0826    4.9355    4.9724    6.0347
```

Fig. 9 - Selected resources

```
>> valor

valor =

    [4x3 double]
    [      23.7622]
    [1x24 double]
    [1x108 double]
    [      27.2808]

>> valor(1)

ans =

    5.0000    6.3743    2.5002
    9.0000    6.2305    2.8990
    10.0000    6.1851    3.8938
    6.0000    4.9724    6.0347
```

Fig. 10 - Valor cell array and matrix of the selected resources

The 2nd cell is the value of the system, which is the sum of the global weighting of each selected resource identified by the green box. The 3rd and 4th cells are the CP and CT arrays respectively and the 5th cell is the total cost of the selected system (CP+CT) given by the selection module output, *fx1*.

The next phase is the presentation of the results of this selection that doesn't contemplate the Value Analysis. The module of the presentation of the results is responsible for

showing the analysed information in a more user friendly form.

It will show the number of tasks of the Plan, the number of initial candidate resources, the number of resources that were approved in the 1st level, the selected resource system, the cost of the system and his parcels, the value of the system, the number of resources that were selected, having although negative classification, the information if the solution was optimal, the number of iterations that the program used and the time spend.

```
-----
Resultados da selecção final para o Plano de Tarefas 1

N.º de tarefas: 4
N.º inicial de recursos candidatos: 10
k mínimo do nível 1: 6

Sem análise de valor

Sistema de recursos: R1,5; R2,9; R3,10; R4,6;

Custo do sistema: 27.2808
CP: 15.3277
CP R1,5= 2.5002;
CP R2,9= 2.899;
CP R3,10= 3.8938;
CP R4,6= 6.0347;

CT: 11.9531
CT 1->2= 4.1147
CT 2->3= 3.8495
CT 3->4= 3.9889

Valor do sistema: 23.7622
Valor R1,5= 6.3743;
Valor R2,9= 6.2305;
Valor R3,10= 6.1851;
Valor R4,6= 4.9724;

N.º de recursos seleccionados com classificação negativa: 1

Foi encontrada a solução óptima

N.º de iterações efectuadas: 807

Tempo da simulação = 0 horas,0 minutos e 13 segundos
```

Fig. 11 - Presentation of the selection (without VA)

After the presentation of the results, and back do the Base module, the number of resources that passes the 2nd level will be checked, *k_min2*. If it's only one Plan being analysed, and the *k_min2* is equal to zero, the program ends here, otherwise it's necessary to resize the CP and CT arrays. In the resize of the CP array, there's no difficulty since the CP is always attached with the respective resource. The procedure is the same as the presented for the first creation of CP array, but this time the matrixes that are going to be accessed are the ones that retain the 2nd level evaluation.

In the case of the CT array, it's a little more complicated. In this array we have all the possible transport combinations between the tasks. As presented in figure 12, only for the first two tasks, we have 36 possible transport combinations.

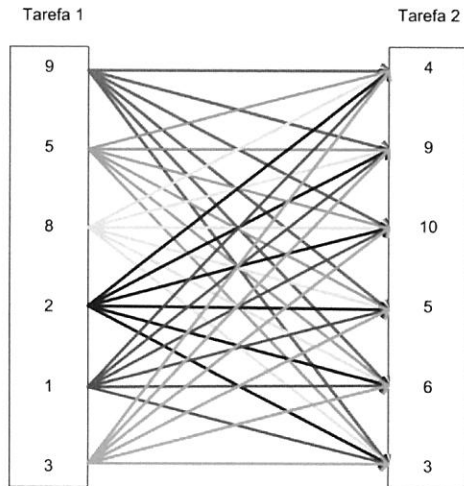


Fig. 12 - Transport combinations between the first 2 tasks

In the 2nd level of the pre-selection some of the resources were eliminated and, therefore, the connections are also eliminated. As the number of resources that passes the 2nd level is 3, the number of possible combinations reduces considerably passing from 36 to only 9 possible connections between 2 tasks, figure 13.

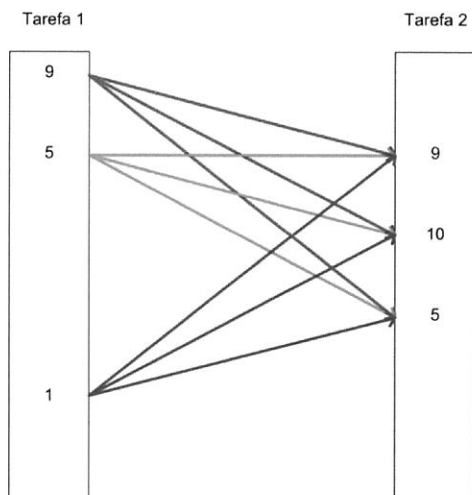


Fig. 13 - Transport combinations between the first 2 tasks after 2nd level evaluation

In the creation of the new CT array, it is necessary a special attention to keep the coherency of the values of the connections. For that, the method found is a restructuration of the values, correspondent to the 2 tasks, into a matrix of dimensions k_{min1} by k_{min1} , or in our example 6 by 6, as in the next figure.

```
R>> m
m =
  5.4073  3.0062  4.1903  5.1916  4.5484  4.2670
  5.1843  4.1147  8.6838  4.0231  9.4359  5.2511
  3.0965  4.7626  7.8690  8.1119  6.3205  6.5748
  7.2655  3.0819  4.8366  2.3397  5.1018  7.8985
  7.4349  10.5757  6.3335  1.6870  9.3788  8.5646
  4.8347  6.4452  7.5589  6.7694  7.3927  5.7746

R>> CT
CT =
  Columns 1 through 10
  5.4073  3.0062  4.1903  5.1916  4.5484  4.2670  5.1563  4.1147  8.6838  4.0231
  Columns 11 through 20
  9.4359  5.2511  3.0965  4.7626  7.8690  8.1119  6.3205  6.5748  7.2655  3.0819
  Columns 21 through 30
  4.8366  2.3397  5.1018  7.8985  7.4349  10.5757  6.3335  1.6870  9.3788  8.5646
  Columns 31 through 40
  4.8347  6.4452  7.5589  6.7694  7.3927  5.7746  5.9224  6.1762  4.4207  8.8459
  Columns 41 through 50
```

Fig. 14 - Restructuration of the first 36 values in a matrix

This way, and doing a resumed explanation, each line of the matrix corresponds to a resource of the first task (1st line corresponds to the CT of resource 9, 2nd line to resource 5 and so on) and each column corresponds to a resource of the second task (1st column is relative to resource 4, 2nd column to resource 9 and so on). It will be done a comparison between the 2 levels of each task to verify which resources were eliminated of one level to another. After that comparison the positions of the resources that were eliminated will be known and therefore the lines and columns related to them will be filled with zeros. The remaining values are the ones that will be part of the new CT array, CT2.

```
R>> m
ans =
  0  3.0062  4.1903  5.1916  0  0
  0  4.1147  8.6838  4.0231  0  0
  0  0  0  0  0  0
  0  0  0  0  0  0
  0  10.5757  6.3335  1.6870  0  0
  0  0  0  0  0  0

R>> CT2
CT2 =
  3.0062  4.1903  5.1916  4.1147  8.6838  4.0231  10.5757  6.3335  1.6870
```

Fig. 15 - Matrix with the remaining values and 1st part of CT2 array

The filling procedure for the rest of the array is similar to the explained above. In the end, the CT2 array will have 27 values while the CP2 array will have 12 values and it's now possible to proceed with the program, calling again the selection module in order to obtain the system with VA.

The steps for this are the same as the explained above for the selection without the VA. The only difference is that the name of the variables is going to be different to avoid conflicts and losing the previous data.

At the end of the presentation of the new results, the total time of simulation is showed as we can see in the figure below.

This tool has already done a set of simulations as a form of support the advantages of a model with a VA in pre-selection phase. Through these simulations some assumptions were proved, as the higher value of the selected system and the guarantee that all the selected resources have a positive classification in all the considered systems. Another conclusion is that amount of time used by the algorithm is smaller with the integration of the VA in the pre-selection phase.

```

-----
N.º inicial de recursos candidatos: 10
k mínimo do nível 1: 6
k mínimo do nível 2: 3

Com análise de valor

Sistema de recursos: R1,5; R2,9; R3,7; R4,5;

Custo do sistema: 30.3663
CP: 16.8859
CP R1,5= 2.5002;
CP R2,9= 2.899;
CP R3,7= 4.3805;
CP R4,5= 7.1062;

CT: 13.4804
CT 1->2= 4.1903
CT 2->3= 4.2901
CT 3->4= 4.9999

Valor do sistema: 25.2385
Valor R1,5= 6.3743;
Valor R2,9= 6.2305;
Valor R3,7= 6.6743;
Valor R4,5= 5.9594;

Foi encontrada a solução ótima

N.º de iterações efectuadas: 183

Tempo da simulação = 0 horas,0 minutos e 1 segundos
-----
Tempo total da simulação = 0 horas,0 minutos e 14 segundos

```

Figure 16 - Presentation of the selection (without VA)

CONCLUSIONS

Based on BM_VEARM model, the selection process consists in two phases, the pre-selection and the selection of resources. The main goal of this work was the development of a pre-selection tool and its integration with a selection module. For that, was necessary an overall comprehension of the pre-selection process for distributed enterprises, develop the pre-selection pseudo code, implement the algorithm in MATLAB, and after the integration with the selection module, the simulation and validation of the tool

Despite the existence of several programs that could be used to develop this tool, the chosen one was MATLAB, due to the mentioned characteristics and due to the fact that the selection module had been already developed

using this language making the integration process easier.

The tool was validated having already realized a simulation plan in order to validate the pre-selection model with VA integration.

The optimization of the code can be seen as future development, since that in the beginning of the development of the tool, the knowledge of the language used was low. The creation of a graphic interface, even more user-friendly, the possibility of choosing more than one selection algorithm (besides Branch&Bound) and the importation and exportation of data from/to Excel sheets can also be seen as future development.

References

- [1] Castro, A. (2010), Desenvolvimento de uma Ferramenta de Pré-selecção de Recursos para Empresas Ágeis/Virtuais, Dissertação de Mestrado em Eng. Electrotécnica e de Computadores, ISEP.
- [2] Pires A. (2011), Integração de Análise do Valor no Processo de Configuração de Empresas Ágeis/Virtuais, Unpublished PhD Thesis, Universidade do Minho.
- [3] Golub, G. and Van Loan, C., 1996. Matriz Computations. 3rd ed. Johns Hopkins University Press.
- [4] Ruggiero, M. and Lopes, V., 1996. Cálculo Numérico - Aspectos Teóricos e Computacionais. São Paulo: MacGraw-Hill.

BUSINESS SUSTAINABILITY 2.0

Management, Technology and Learning
for Individuals, Organisations and Society in Turbulent
Environments

Edited by:

Goran D. Putnik

Paulo Ávila



Chaos and Sustainability – the 2nd Order 2.0



Universidade do Minho

isep Instituto Superior de
Engenharia do Porto

About the authors:

Goran D. Putnik

Prof. Dr. Goran D. Putnik is (Full) Professor at Department of Production and Systems Engineering, University of Minho, Portugal. His scientific and engineering interests are production systems and enterprises design and management theory and implementations: CIM, intelligent production systems and enterprises, machine learning as a design theory model, concurrent and collaborative engineering, information systems management, formal theory of production systems and enterprises, distributed, agile and virtual enterprises, and complexity management in organizations. He is supervising a number of PhD projects as well. He regularly publishes and participates on international scientific conferences. His publishing record comprises more than 200 publications in international and national journals and conferences proceedings, and 9 books, of which the "Encyclopedia of Networked and Virtual Organizations" is distinguished. He serves as a member of Editorial Board for several International Journals. He was also invited lecturer on a number of universities.

Presently, he is the member of the following professional societies and scientific networks:

- Member of the CIRP – "The International Academy for Production Engineering"
- Member of the IFIP WG 5.5 – "COoperation infrastructure for Virtual Enterprises and electronic business – COVE"
- Member of the ECCON – "European Chaos and Complexity in Organisations Network – ECCON"

Paulo Ávila

Prof. Dr. Paulo Ávila is Coordinator Professor in the Department of Mechanical Engineering, at the School of Engineering – Polytechnic of Porto, Portugal, and the Director of the same Department. He received his Dipl.Eng. from the University of Coimbra in the domain of Mechanical Engineering, his MSc from the University of Minho in the domain of Computer Integrated Manufacturing and his PhD from the University of Minho in the area of Agile and Virtual Enterprises. His scientific and engineering interests cover the subjects of Production System Organization and Management, Computer Integrated Manufacturing (CIM), Total Quality Management (TQM) and Virtual Enterprises. He regularly publishes in international and national scientific conferences proceedings, journals and books. He is a consultant in several enterprises in the field of Industrial Organization and Management.



Sponsored by



ISBN 978-972-8692-66-7

ISBN 978-989-95907-3-1