

International Conference  
on Industrial Engineering and Systems Management

**IESM' 2009**

May 13 - 15, 2009

MONTREAL - CANADA

# Activity-on-Arc Network Graphic Editor

Anabela Tereso\*, João Paulo Cardoso, José Pedro Cabral

*Department of Production and Systems, University of Minho, 4800-058 Guimarães, PORTUGAL*

---

## *Abstract*

This paper presents an application developed to visualize and edit Activity-On-Arc (AoA) project networks. It is part of a larger project that has the goal to study the resource allocation problem in stochastic multimodal activity networks. This application was developed in Java programming language and uses the open source libraries JGraph and JGraphT. It enables one to visualize and edit AoA networks with multiple resources, and save it into one of the two types of files available (TXT or XML).

*Key Words:* Project Management and Scheduling, Resource Allocation, Java, Activity-On-Arc Networks

---

## **1 INTRODUCTION**

This application is to facilitate interaction with previously developed project management applications in the field of resource-constrained scheduling problems with stochastic multimodal characteristics.

To further understand the context in which this application was developed, one needs to understand some basic concepts of project management. We define a project as a set of activities which have to be performed in order to fulfil a specified goal. Each activity may use several resources, from material to machinery and human, and it has time and cost dimensions, restrictions and implications.

In researching alternative modes of project representation we encountered several editors such as yEd Graph Editor [17] and CharGer - Conceptual Graph Editor [18] [19]. Despite being powerful applications in designing many types of diagrams and in editing conceptual graphs, these applications did not allow Activity-on-Arc (AoA) networks to be developed. Further, they lacked specific facilities that are required in the more advanced treatment of projects, such as the ability to accommodate several constraints, or allowing multiple resource allocation to each activity and the specification of the quantities in which these resources could be used within the activity, or the ability to store costs associated with the use of the resources and their work content, including the specification of the project due time and the cost incurred in case of delay in delivery; etc.

A project may be represented by an “activity network” that describes the sequence of tasks to be performed until the conclusion of the project. This network includes information about the precedence relations between

---

\* Corresponding author: Anabela Tereso. E-mail: [anabelat@dps.uminho.pt](mailto:anabelat@dps.uminho.pt)

activities. Such representation can be divided into two groups: Activity-on-Node (AoN) and Activity-on-Arc (AoA).

In AoN networks, activities are represented by nodes, enabling the arcs to represent the precedence among activities.

---



---

Fig. 1. Example of an Activity-on-Node Network (AoN)

In AoA networks, activities are represented by arcs and the nodes represent the start and ending epochs of each activity.

---



---

Fig. 2. Example of an Activity on Arc Network (AoA)

Fig.1 and Fig.2 represent the same activity network (with two activities) in AoN and AoA formats, respectively.

This application, for compatibility reasons, uses the AoA representation.

As stated before, several resources may be allocated to an activity and, in our application, each resource has two associated costs: *maintenance cost per unit* (mcu) and *quantity cost per unit* (qcu).

As it is easily perceived, AoA networks may be viewed as Directed Acyclic Graphs (DAG). Each activity can have only one start node and one end node, and there cannot be any cycles [4]. This will be tested before a project is saved.

This work is part of a project that has as the goal the study of the resource allocation problem in stochastic multimodal activity networks (see references [7] to [16]). Each activity is assumed to possess work content that is exponentially distributed with parameter  $\lambda$ . Once the resource(s) allocation to an activity is decided upon, the duration of the activity is immediately known, also exponentially distributed with parameter that is proportional to  $\lambda$ .

## 2 THE APPLICATION DEVELOPMENT

This application was developed in Java [3]. The following sections will describe some aspects of the implementation and some decisions taken.

### File Formats

One of the purposes of this application was to be able to read pre-existing types of files created by other applications developed in the scope of the research program and also to standardize the storage of the data in the same formats. There were two types of files, one in TXT format which was used to store single resource networks, and the other in XML format, which was used to store multiple resources networks. An additional file had to be created in order to store the positioning of the nodes and other graphic characteristics of the network. This file is saved with the extension AoA.

#### TXT

This format contains the following information:

- Line 1: number of nodes; number of arcs;
- Line 2 to n-2: origin node; target node; lambda; Min; Max;
- Line n-1: delay cost per unit;
- Line n: due date of the project.

The first line contains the number of nodes and arcs of the network, and the two last lines contain the delay cost per unit time and the due date of the project. Each one of the remaining lines represents one activity of the project. For each activity the information stored will be: *origin node*, *target node*, *lambda* (the parameter of the exponential distribution that specifies the random work content for the activity); *Min* and *Max* that specify the minimum and maximum amount of resource that can be allocated to the activity. The single resource networks used to test this application were generated through an application called RANG [5]. RANG uses the method of elimination described in [2] to randomly generate this type of AoA networks.

#### XML

This format contains the following information:

- Project name;
- Project due date and delay cost per unit;
- Resources (id, max, min, mcu, qcu);
- Activities (id, origin and target nodes)
  - Resources used by the activity (id, lambda, and additional changes to the default resource parameters).

This format is more complex because it has to allow the storage of multiple resources, the activities that use them and additional changes to the default resource values for each activity it is assigned to (max and min).

#### AoA

- Uses the XML format;
- Stores the coordinates and physical attributes of the node;
- Stores additional points that specify the arcs curves.

As this application has a graphical environment, it was necessary to store an additional file each time a project was saved. This file (.AoA) stores the information needed to open a project with the same appearance it had when it was last edited.

### Data Structures

This section describes the most relevant data structures used by the application. The main data that the application needs to keep are the nodes, the arcs and the resources. All this data would have to be stored efficiently so, we chose to keep it stored in Hashmaps, which use the object identifiers as keys (name for the nodes, id for the resources, and a string with origin and target nodes for the activity (arc)) and the objects themselves as fields.

#### Nodes

For each node it is necessary to store its name (identifying number) and also, because we are working in a graphical environment, it is necessary to store the coordinates of the node and also the physical attributes of the node (height and length).

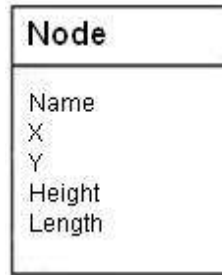


Fig. 3. Representation of the node data structure

#### Arcs

As we can see in the figure 4, arcs are a lot more complex. For each one we need to keep its name, the origin node and the destination node. Because each activity can have more than one resource allocated to it, it was necessary to create a resources field that stores a Hashmap, which keeps all the resources allocated to the activity stored.

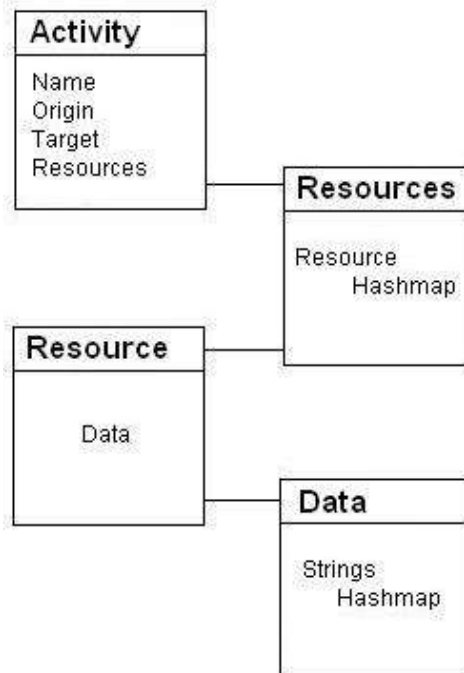


Fig. 4. Representation of the arc data structure

**Resources**

Each activity can change the default data of the project resources. For this purpose an additional Hashmap was used. That Hashmap key is the resource id and for the data we created a data field which is a Hashmap of strings that contains the fields that were altered and their respective values.

**Resource**

For each project resource, it is necessary to store its *id*, the *max* and *min* quantities, the *maintenance cost per unit* (mcu) and the *quantity cost per unit* (qcu).

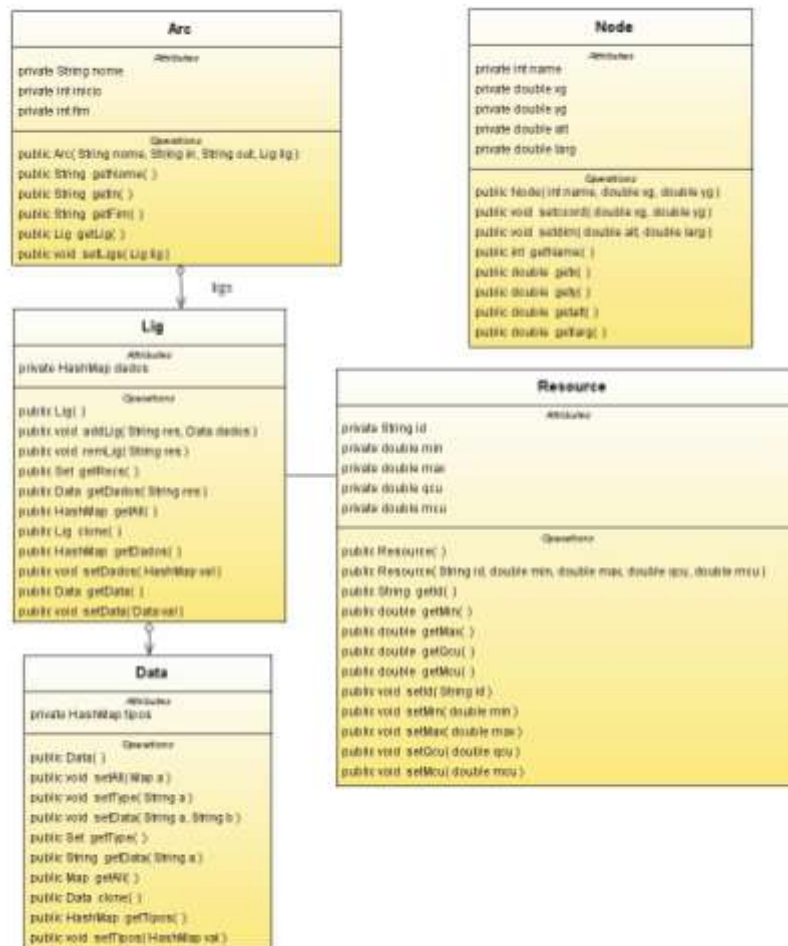


Fig. 5. Class Diagram of the Application Data

**Libraries**

In this section we will present the libraries used in the development of the application.

**JGraph**

JGraph is an open source library, developed in Java, which is used to visualize graphs. JGraph is compatible with Java Swing and can be used since J2SE 4.0. This library is quite extensive and complex and enables us to modify the presentation of graphs, by changing shapes, sizes or colours. It also contains two listeners that were used: one that detects when a graph component is selected and another that detects the changes made to that component [1].

**JGraphT**

JGraphT is an open source library, developed in Java, which is used to manipulate graphs. It allows defining several types of graphs and enables us to perform various mathematical operations based on the graph theory.

JGraphT interacts easily with JGraph, enabling this way the visualization of the graphs created [1].

**Application Interface**

This application can run in English or in Portuguese. This feature was implemented by creating a copy of the interface for each language, to optimize the application efficiency. Figure 6 shows the Interface Class Diagram, with connections between the interface classes and the data classes.

As we can see, there are several interfaces. The Main interface is the first interface of the application and lets one choose the language. If we choose Portuguese the interface MainPT will be called. In MainPT the initial state of the application will be stored and we can choose to start a new AoA network (with single or multiple resources), to load an existing AoA network or to import a single or multiple resource network.

By choosing any of the previous options, the VisualPT class will be called. This is the longest interface of the application and in it we can perform every operation possible on AoA networks, like removing or adding nodes and arcs, save the network in single or multiple resources formats, and export the graph to an image file (PNG or EPS). We can also edit the position of the nodes, add curves to the arcs and change the zoom. This is also the class in which the data structures are stored.

There are also two minor interfaces, ResVisPT and InfoVisPT, which can be called through VisualPT. They enable us to see and edit the project resources and the activity arcs of the network.

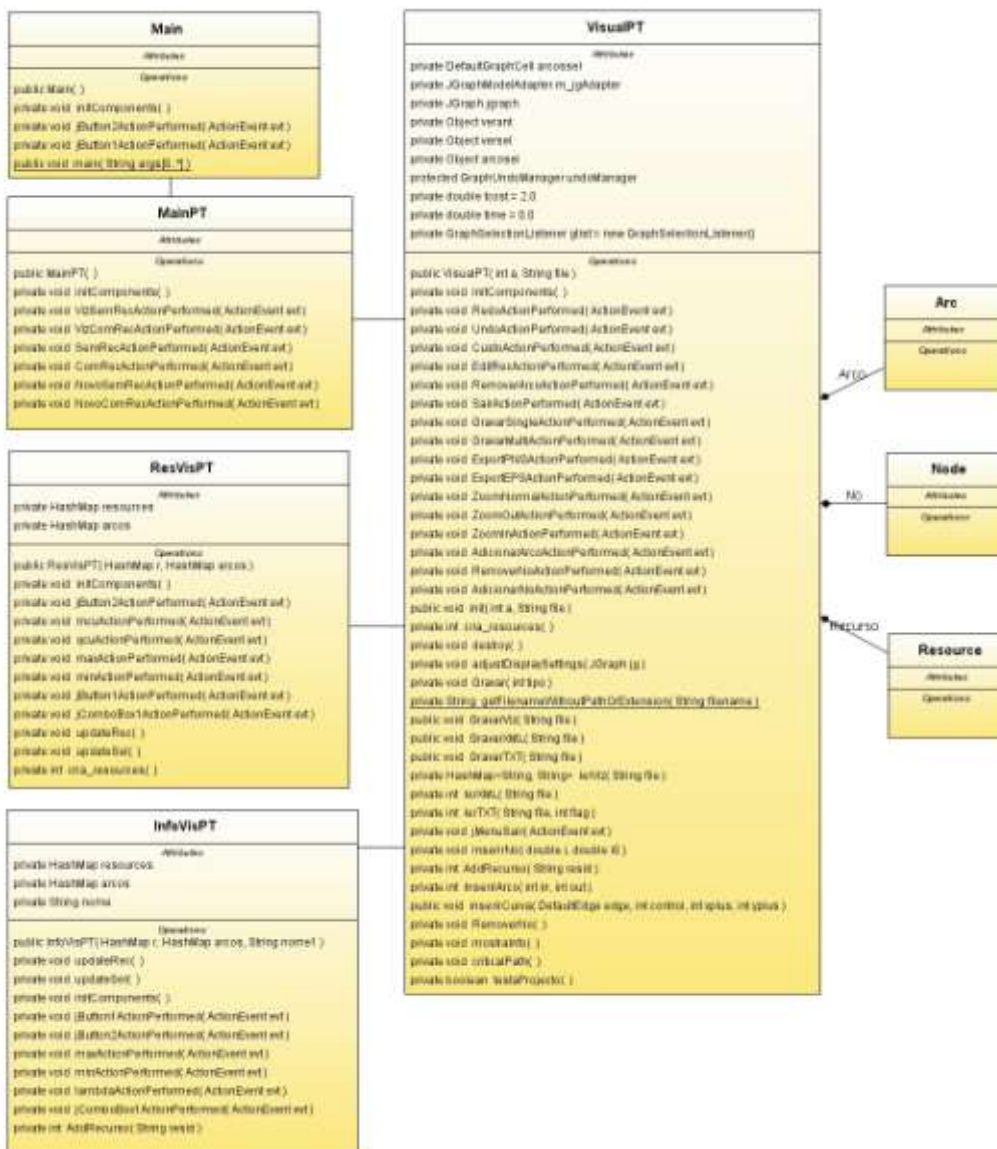


Fig. 6. Interface Class Diagram

### Critical Path Method

The critical path method constitutes an important tool in project management. It is applied to projects that can be decomposed into a set of activities which are considered to have deterministic work content, costs and durations, and have relations of precedence among them. An activity can only be performed when its predecessors have finished.

The precedence restrictions imply, as stated above, the order in which activities must be performed.

The critical path is the longest path between the initial node (beginning of the project) and the final node (end of the project). The activities that belong to the critical path are called critical activities [6].

This application uses this method to determine the Project due date when a project is saved, considering the mean work content of each activity as surrogate for the random variable. It has also the capability to show the critical path (see figure 7).

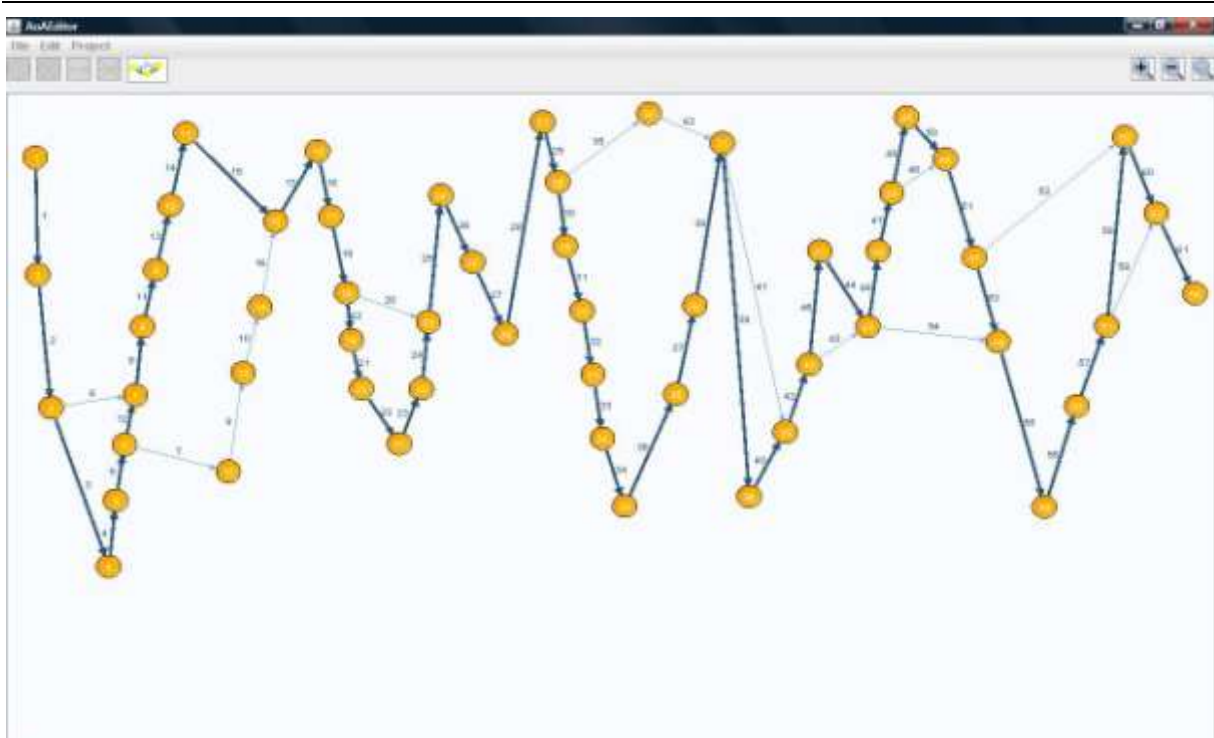


Fig. 7. The Critical Path

### 3 CONCLUSIONS

Next we present some screenshots of the application running, as the results of our work (figures 8 to 13).



Fig. 8. Language Selection Menu



Fig. 9. English Main Menu

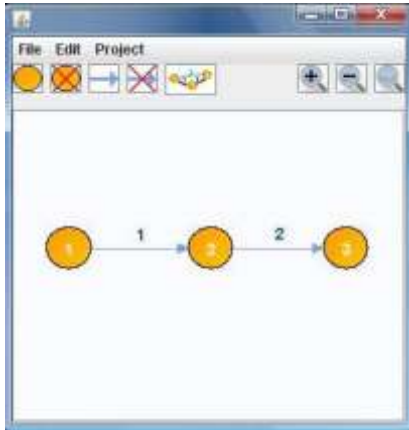


Fig. 10. New Network with zoom in

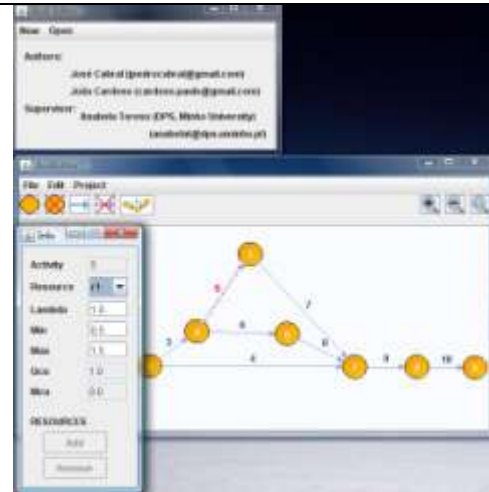


Fig. 11. Editing Activity Data

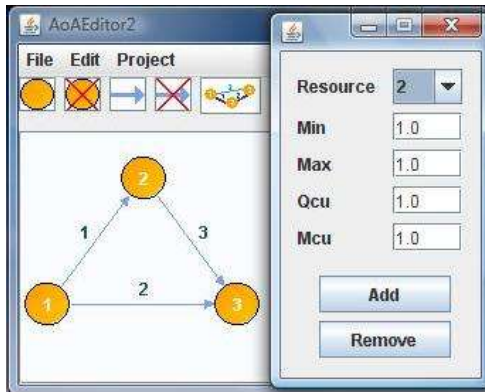


Figure 12 - Editing Project Resources

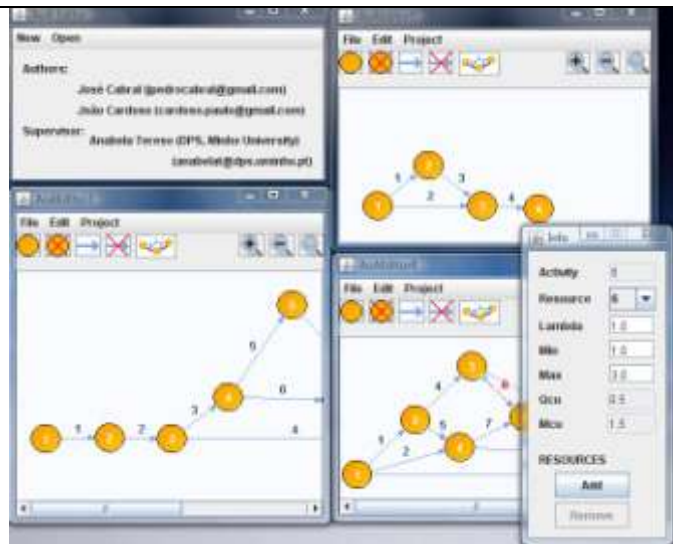


Figure 13 - Editing Several Projects

This application is ready to be used to visualize and edit AoA networks. It provides a practical solution for modelling and defining the planning phase of a project. It has a simple and practical interface and can be displayed in two languages (Portuguese and English). It is capable of exporting its projects into an image file (PNG or EPS) and it can edit single and multiple resources projects. It allows also for single resource networks to be saved in multiple resource format and vice versa.

In our future work we shall integrate the previous studies developed in this project (see references [7] to [16]) with this application, to be able to test, in more depth, the models developed with new networks generated.



#### 4 REFERENCES

- [1] Benson, D. (2007), "JGraph and JGraph Layout Pro User Manual", (<http://www.jgraph.com/support.html>).
- [2] Demeulemeester, E., Dodin, B., Herroelen, W. (1993) "A Random Activity Network Generator", *Operations Research*, Vol. 41, n° 5, pp 972-980.
- [3] Horstmann, C. S. and Cornell, G. (2007) "Core Java Volume I: Fundamentals", Prentice Hall.
- [4] Jungnickel, D. (1999) "Graphs, Networks and Algorithms", Springer.
- [5] Marques, H., Gonçalves, L. (2007) "Random Activity Network Generator (RANG)", Application Report, University of Minho.
- [6] Silva, S. C. (2006), "Gestão/Programação de Projectos", Production Management course support notebook, University of Minho, Portugal.
- [7] Tereso, A.P., Araújo, M.M., Elmaghraby, S.E. (2001) "Adaptive Resource Allocation in Multimodal Activity Networks", *Optimization 2001*, Aveiro, Portugal, July 22-25, 2001. Abstract published in the proceedings.
- [8] Tereso, A.P., Araújo, M.M., Elmaghraby, S.E. (2003) "Experimental Results of an Adaptive Resource Allocation Technique to Stochastic Multimodal Projects", *International Conference on Industrial Engineering and Production Management (IEPM'03)*, Porto, Portugal, May 26-28, 2003. Full paper published in the proceedings (ISBN 2-930294-13-2).
- [9] Tereso, A.P., Araújo, M.M., Elmaghraby, S.E. (2003a) "Basic Approximations to an Adaptive Resource Allocation Technique to Stochastic Multimodal Projects", *Euro/Informs Joint International Meeting*, Istanbul, Turkey, July 6-10, 2003. Abstract published in the proceedings.
- [10] Tereso, A.P., Araújo, M.M., Elmaghraby, S.E. (2004) "Adaptive Resource Allocation in Multimodal Activity Networks", *International Journal of Production Economics*, Elsevier, Vol 92, Issue 1, pp. 1-10, November 2004.
- [11] Tereso, A.P., Araújo, M.M., Elmaghraby, S.E. (2004a) "The Optimal Resource Allocation in Stochastic Activity Networks via the Electromagnetism Approach", *Ninth International Workshop on Project Management and Scheduling (PMS'04)*, Nancy, France, April 26-28, 2004. Extended abstract published in the proceedings.
- [12] Tereso, A. P., Mota, J. R., Lameiro, R. J. (2005) "Adaptive Resource Allocation Technique to Stochastic Multimodal Projects: a distributed platform implementation in JAVA", *IFORS Triennial Conference (IFORS'05)*, Honolulu–Hawaii–USA, July 11-15, 2005. Abstract published in the proceedings.
- [13] Tereso, A. P., Mota, J. R. M., Lameiro, R. J. T. (2006) "Adaptive Resource Allocation to Stochastic Multimodal Projects: A distributed platform implementation in Java", *Control and Cybernetics Journal*, Vol. 35, Issue 3, pp. 661-686, 2006.
- [14] Tereso, A.P., Novais R., Araújo, M.M. (2006a) "The Optimal Resource Allocation in Stochastic Activity Networks via the Electromagnetism Approach: a platform implementation in Java", *21st European Conference on Operational Research (EURO XXI)*, Reykjavík–Iceland, July 2-5, 2006. Abstract published in the proceedings. This paper was submitted to the "Control and Cybernetics Journal" and it is under revision.
- [15] Tereso, A.P., Costa, L. A., Novais, R., Araújo, M.M. (2007) "The Optimal Resource Allocation in Stochastic Activity Networks via the Evolutionary Approach: a platform implementation in Java", *International Conference on Industrial Engineering and Systems Management (IESM' 2007)*, Beijing – China, May 30 to June 2, 2007. Full paper published in the proceedings (ISBN 978-7-89486-439-0). This paper was submitted to the "Computers and Industrial Engineering".

- [16] Tereso, A.P., Araújo, M.M., Moutinho, R., Elmaghraby, S.E. (2008) “Project Management: multiple resources allocation”, to be presented in the International Conference on Engineering Optimization (EngOpt2008), Rio de Janeiro - Brazil, June 1-5, 2008.
- [17] yED Graph Editor, Powered by yFiles Graph Visualization Library, Copyright 2000-2008 yWorks GmbH - All Rights Reserved, <http://www.yWorks.com>.
- [18] Delugach, H.S., CharGer: A Graphical Conceptual Graph Editor, <http://www.cs.nmsu.edu/~hdp/CGTools/proceedings/papers/CharGer.pdf>.
- [19] CharGer - Conceptual Graph Editor, Copyright 1999-2009 - SourceForge, Inc., All Rights Reserved, <http://sourceforge.net/projects/charger/>.