University of Minho
Engineering School
Production and Systems Department

ILS 2010
Third International Conference
on
Information Systems, Logistics and Supply Chain
April 13-16, 2010 - Casablanca, Morocco

On The Optimal Resource Allocation in Projects Considering the Time Value of Money

Anabela Teseso, Duarte Barreiro, Madalena Araújo
University of Minho – Portugal

Salah Elmaghraby
North Carolina State University – USA

anabelat@dps.uminho.pt

Topics

- Introduction and Problem Definition
- Present Worth of Resource Cost
- The Dynamic Programming Model
- The Electromagnetism-like Mechanism
- The Evolutionary Algorithm
- Results and Conclusions

Introduction and Problem Definition

- Problem: optimal resource allocation in stochastic activity networks, considering the time value of money
- Reasons for taking the time value of money into consideration:
  - Long term projects that span several years should take account of the changing value of money
  - Discounting future commitments is another way of expressing uncertainty (choice of the discount rate)

Introduction and Problem Definition

- Project in AoA mode of representation: G(N,A)
  - N: set of nodes
  - A: set of arcs
- Goal: minimize total cost (resource and tardiness cost)
- Each activity $i \in A$ has an associated work content:

$$W_i \sim exp(\lambda_i)$$

- x_i is the amount of resource allocated to activity *i*:

$$x_i \in [l_i, u_i] \text{ with } l_i \geq 0 \text{ and } u_i < +\infty$$

Introduction and Problem Definition

- The duration of the activity is: $Y_i = \frac{w_i}{x_i}$.
- The total cost of the project will be: $Total\ Cost = rc + tc$,
- Where *rc* is the resource cost: $rc = \sum_{i \in A} c_R\, x_i^2\, Y_i = \sum_{i \in A} c_R x_i W_i$
- And *tc* is the tardiness cost: $tc = c_L \times max\{0, Y_n - T\}$
- We assume that each activity will start as soon as it is precedence-feasible.

Introduction and Problem Definition

- The resource allocation models used are: Dynamic Programming (DP), Electromagnetism-like Mechanism (EM) and Evolutionary Algorithm (EA).
- For each model we shall present two different approaches: Discrete-Time Discounting and Continuous-Time Discounting.
- In either approach the goal is to determine the resource allocation that optimizes the p.v. of the project.

Present Worth of Resource Cost

- This section is devoted to the introduction of some basic concepts in “interest” and “discounting” which may not be familiar to all.
- Discrete-Time Discounting (*Version 1)*
  - In discrete-time discounting the duration of the activity is divided into discrete time intervals and discounting is applied to the receipts/disbursements in each interval.
  - Suppose the annual interest rate is given as i_a. Then the annual discount rate, denoted by β, is given by

$$\beta = \frac{1}{1+i_a}$$

Present Worth of Resource Cost

- Discrete-Time Discounting (*Version 1)*
  - The period interest rate, denoted by i_p, is given by the solution to the equation: $(1+i_p)^{n_p} = 1 + i_a$, which $\Rightarrow i_p = (1+i_a)^{1/n_p} - 1$
  - The period discount factor, α, is evaluated from: $\alpha = \frac{1}{1+i_p}$ or $\alpha^{n_p} = \beta$
  - Assuming that the work content *W* is expended uniformly over the activity duration *Y*, then the work content in each period is *W/Y*, which, by the definition of *Y*, is equal to *x*.

Present Worth of Resource Cost

- Discrete-Time Discounting (*Version 1)*
  - The p.v. of the work content at the start of the activity at discount rate α is given by $VW = \underbrace{x + \alpha x + \alpha^2 x + \cdots + \alpha^{Y-1} x}_{Y\ terms} = x\frac{1-\alpha^Y}{1-\alpha}.$
  - If the activity starts at time *d* then the p.v. of the activity work content, is given by $PVW = VW.\alpha^d$
  - Finally, the p.v. of the resource cost is

$$rc = c_R x^2.\frac{PVW}{x} = c_R \times x \times PVW$$

Present Worth of Resource Cost

- Discrete-Time Discounting (*Version 2)*
  - In this second version, we assume that the cost of the work content of an activity is incurred at its completion.
  - The p.v. of the work content at the start of the activity, when the cost of the activity is incurred at its completion, will be:

$$VW = W\alpha^{Y}$$

  - Then we use the same formulas as before:

$$PVW = VW.\alpha^{d}$$

$$rc = c_R x^2 . \frac{PVW}{x} = c_R \times x \times PVW$$

Present Worth of Resource Cost

- *Continuous-Time Discounting*
  - An alternate approach is to consider time as a continuum and the effort is continuously applied to the activity.
  - The continuous discounting of $1 spent at time t is given by

$$e^{-i_p t}$$

  - For the whole year we have the sum

$$rc = 1 + e^{-i_p} + e^{-2i_p} + \cdots + e^{-364 i_p}$$
$$= \frac{1-\left(e^{-i_p}\right)^{365}}{1-e^{-i_p}}.$$

Present Worth of Resource Cost

- *Continuous-Time Discounting*
  - If the work content is continuously discounted each day, during *n* days, then the p.v. of the work content would be:

$$VW = x + xe^{-i_p} + xe^{-2i_p} + \cdots + xe^{-(Y-1)i_p}$$
$$= x \times \frac{1-\left(e^{-i_p}\right)^n}{1-e^{-i_p}}.$$

  - If the activity starts approximately *d* days from present time:

$$PVW = VW \times e^{-d \times i_p}$$

  - The p.v. of the resource cost of this activity would be given by:

$$rc = c_R x^2 . \frac{PVW}{x} = c_R \times x \times PVW$$

The Dynamic Programming Model

- The Dynamic Programming Model (DP) divides the activities into two groups: those with fixed resource allocations, denoted by the set *F*, and those with yet-to-be-decided resource allocations, the *decision variables*, denoted as the set *D*, with $F \cup D = A$, the set of all activities.
- The set *D* is the set of activities on the longest path in the network (the path containing the largest number of activities).

The Dynamic Programming Model

- A stage is defined as an epoch of decision making. We define stage (k) as the decision epoch of the allocation for each activity $a \in D$.
- In each stage only one decision variable is optimized since each uniformly directed cutset (u.d.c.) in the network contains exactly one activity in *D*.
- There is also the concept of state, which is defined as a vector of times of realization of the set of nodes that allows us to decide on x_a and evaluate the contribution of the stage, for $a \in D$.

The Dynamic Programming Model

- In DP, the numbering of stages is done backwards. The decision variable of stage *k* is identified as $x_{[k]}$, where *k* means the number of stages that are still missing for the conclusion of the project.
- Without considering the time value of money we have:

$$f_1(s_1|F) = rcf + \min_{x_{[1]}\in D} \varepsilon\left\{c_r x_{[1]} W_{[1]} + c_L \times U\right\}.$$

$$U = max\{0, \Upsilon_n - T\}$$

$$f_k(s_k|F) = \min_{x_{[k]}\in D} \varepsilon\left\{c_{[k]}\left(\left[x_{[k]}\right], s_k\right) + \varepsilon f_{k-1}(s_{k-1}|F)\right\}$$

The Dynamic Programming Model

- Using the discrete time approach, we get:

$$f_1(s_1|F) = PV_{rcf} + \min_{x_{[1]} \in D} \varepsilon\{c_r x_{[1]} PVW_{[1]} + PV(c_L \times U)\}$$

$$PV_{rcf} = \varepsilon \sum_{k \in F} c_r x_k PVW_k = \sum_{k \in F} c_r x_k \varepsilon(PVW_k)$$

  - In version 1 we will have $PVW_k = x_k \dfrac{1-\alpha^Y}{1-\alpha} \times \alpha^d$
  - And in version 2 $PVW_k = W\alpha^Y \alpha^d$

$$PV(c_L \times U) = c_L \times U \times \alpha^{Y_k}$$

$$f_k(s_k|F) = \min_{x_{[k]} \in D} \varepsilon\{PVW_{[k]}([x_{[k]}], s_k) + \varepsilon f_{k-1}(S_{k-1}|F)\}$$

The Dynamic Programming Model

- Using the continuous time approach, we need to use the following equations:

$$PVW_k = x_k \times \frac{1-\left(e^{-i_p}\right)^Y}{1-\left(e^{-i_p}\right)} \times e^{-d \times i_p}$$

$$PV(c_L \times \mathrm{U}) = c_L \times \mathrm{U} \times e^{-\Upsilon_k \times i_p}$$

The Electromagnetism-like Mechanism

- The Electromagnetism-like Mechanism (EM) is based on the principles of electromagnetism and it was developed by Birbil and Fang (2003).
- Those principles say that two particles experience forces of mutual attraction or repulsion depending on their charges.

The Electromagnetism-like Mechanism

- This algorithm is divided in four phases:
  - Initialization of the algorithm
  - Calculation of the vector of total force exerted on each particle
  - Movement along the direction of the force
  - Application of neighborhood search to exploit the local minima
- The initialization disperses randomly the *m* particles in the n-dimensional space (hyper-cube).

The Electromagnetism-like Mechanism

- Each particle is a vector of dimension |A| with a fixed allocation of the resources to the activities.
- For each particle the value of the objective function is calculated and the best point is saved in x^{best}.
- The charge of each particle is evaluated as:

$$q^c = \exp\left[-\mathrm{n} \times \frac{f(x^c) - f(x^{best})}{\sum_{k=1}^{\mathrm{m}}[f(x^k) - f(x^{best})]}\right]$$

$$c = 1,2,\ldots,m\,.$$

The Electromagnetism-like Mechanism

- The total force exerted on a particle, is determined by:

$$F^c = \sum_{b \neq c}^{m} (x^b - x^c) \frac{q^c q^b}{\|x^b - x^c\|^2},$$

$$c = 1,2, \dots, m$$

- After determining the total force, it is just necessary to move the particle according to:

$$x^{m'} = x^{\mathrm{m}} + \beta \frac{F^c}{\|F^c\|} (RNG).$$

The Electromagnetism-like Mechanism

- The total cost of the project is given by the sum of the p.v. of the resource cost (*RC*) and the tardiness cost (*TC*)

$$PVC = \sum_{a=1}^{n} PVRC_a + PVTC$$

$$\mathrm{PVRC} = \sum_{a=1}^{n} c_R \times x_a \times \mathrm{PVW}_a$$

- For the discrete time approach:
  - In version 1 we will have

$$\mathrm{PVW}_a = x_a \frac{1-\alpha^{\Upsilon}}{1-\alpha} \times \alpha^{d}$$

$$\mathrm{PVTC} = c_L \times \max(0, \Upsilon_n - T) \times \alpha^{\Upsilon_n}$$

  - And in version 2

$$\mathrm{PVW}_a = W\alpha^{\Upsilon}\alpha^{d}$$

The Electromagnetism-like Mechanism

- For the continuous time approach, we need to use the following equations:

$$PVW_a = x_a \times \frac{1-\left(e^{-i_p}\right)^{\Upsilon}}{1-\left(e^{-i_p}\right)} \times e^{-d \times i_p}$$

$$PVTC = c_L \times \max(0, \Upsilon_n - T) \times e^{-\Upsilon_n \times i_p}$$

The Electromagnetetism-like Mechanism

1. Generate K vectors of $W = (w_1 .. w_n)$ randomly
2. Generate m vectors of $X = (x_1 .. x_n)$ to start with
3. For each vector X
4. For each vector W
5. $rc = c_R \times x_a \times W_a$
6. $tc = c_L \times max\{0, \Upsilon_n - \mathrm{T}\}$
7. $c = rc + tc$
8. End for
9. $f = \sum \frac{c}{K}$
10. Evaluate charges
11. Evaluate forces
12. End for
13. Move the points
14. Go to step 3 until nº of iterations specified is reached

Remark: Without considering the time value of money, for simplification

The Evolutionary Algorithm

- The Evolutionary Algorithm (EA) is based on the natural evolution of the species, and it was developed by Costa and Oliveira (2001).
- It is usually used in optimization problems and it is based on the population evolution.
- There are two important approaches to EA: Evolutionary Strategies (EA-ES) and Genetic Algorithms (EA-GA). In our study we adopted the EA-ES.

The Evolutionary Algorithm

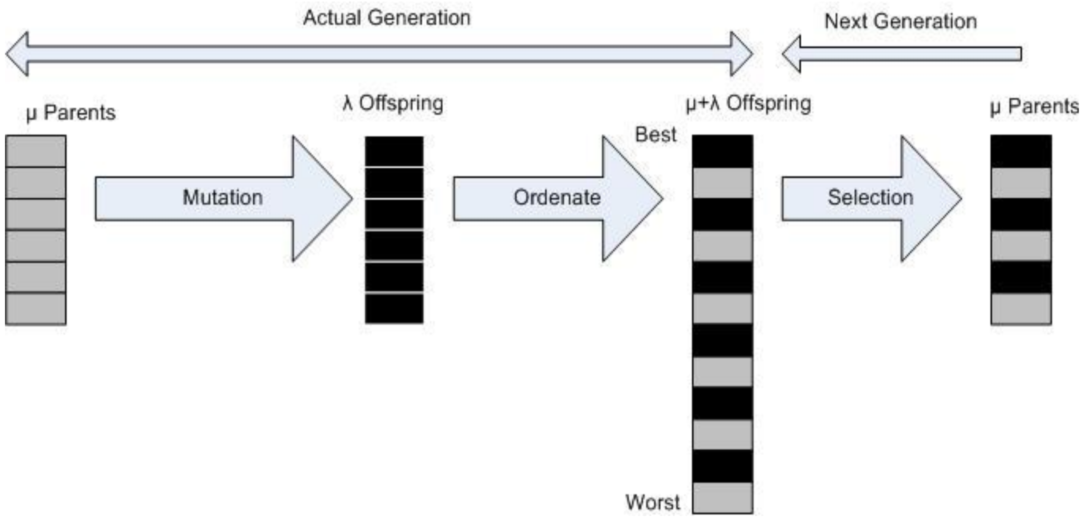
- We start to generate an initial population (ancestors) of size μ that will create a new population (descendents) of size λ after applying mutation and recombination operations.
- The best individuals are chosen to go to the next generation.
- All of these individuals are represented by vectors of real decision variables.

The Evolutionary Algorithm

- The mutation and recombination processes are used to preserve the genetic diversity between ancestors and descendents so that the algorithm will not be trapped in a local minimum.
- The nomenclature often used for representing ES is based on the number of the ancestors μ, on the number of the descendents λ and on the type of selection chosen.

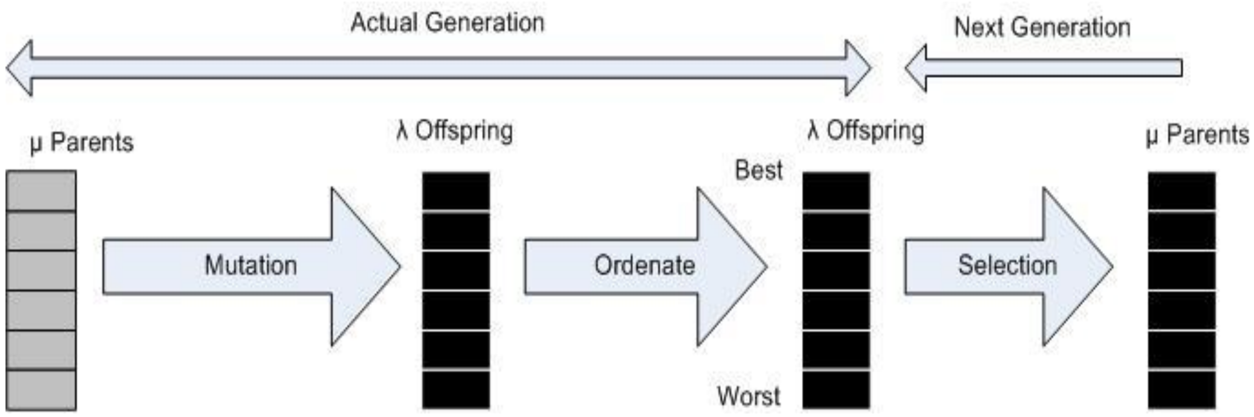
The Evolutionary Algorithm

- If we adopt the nomenclature (μ+λ), after the descendent population has been generated they are added to the ancestors population and then the μ best individuals are selected to go to the next generation.



The Evolutionary Algorithm

- If we adopt the nomenclature (μ,λ), after the descendent population has been generated the μ best individuals are selected to go to the next generation.



- The total cost of the project when using this method is given by the same formulas used in EM.

The Evolutionary Algorithm

1. Generate K vectors of $W = (w_1 .. w_n)$ randomly
2. Generate m vectors of $X = (x_1 .. x_n)$ to start with
3. For each vector X
4. For each vector W
5. $rc = c_R \times x_a \times W_a$
6. $tc = c_L \times max\{0, \Upsilon_n - \mathrm{T}\}$
7. $c = rc + tc$
8. End for
9. $f = \sum \frac{c}{K}$
12. End for
10. Apply mutation
11. Apply recombination
13. Generate the next population
14. Go to step 3 until stop criteria is reached.

Remark: Without considering the time value of money, for simplification

Results and Conclusions

- Experiment layout
  - The program was tested on a set of fourteen projects

Net	1	2	3	4	5	6	7	8	9	10	11	12	13	14
|A|	3	5	7	9	11	11	12	14	14	17	18	24	38	49
T	16	120	66	105	28	65	47	37	188	49	110	223	151	221
c_L	2	8	5	4	8	5	4	3	6	7	10	12	5	5

  - Each activity *i* has stochastic work content W_i, assumed to be exponentially distributed.
  - Both in the EM and EA tests, we generated a set of work contents randomly (100) to represent the possible values for each activity and then we kept these values for all the experiments, for each network.

Results and Conclusions

- Dynamic Programming Results

Total Cost

Net	No discounting	Discrete Version 1	Discrete Version 2	Continuous
1	$43.326	$43.240	$43.188	$43.240
2	$297.513	$294.629	$293.330	$294.629
3	$197.979	$197.070	$196.623	$197.070
4	$385.321	$382.813	$381.082	$382.813
5	$135.340	$134.974	$134.886	$134.974
6	$293.851	$292.599	$291.886	$292.599
7	$161.825	$161.352	$161.125	$161.352
8	$123.931	$123.671	$123.533	$123.671
9	(*)	(*)	(*)	(*)
10	(*)	(*)	(*)	(*)
11	(*)	(*)	(*)	(*)
12	(*)	(*)	(*)	(*)
13	(*)	(*)	(*)	(*)
14	(*)	(*)	(*)	(*)

(*) – Program aborted after 8 hours running.

Execution Time

Net	No discounting	Discrete Version 1	Discrete Version 2	Continuous
1	0.000s	0.001s	0.001s	0.001s
2	0.032s	0.063s	0.063s	0.047s
3	0.062s	0.093s	0.094s	0.078s
4	2.546s	3.359s	3.312s	3.031s
5	8.266s	11.000s	11.187s	10.719s
6	1m 31.094s	1m 53.359s	1m 54.296s	1m 49.594s
7	10m 36.156s	11m 58.671s	11m 44.734s	11m 42.546s
8	52m 18.594s	1h 01m 25.859s	1h 00m 40.860s	56m 47.453s
9	(*)	(*)	(*)	(*)
10	(*)	(*)	(*)	(*)
11	(*)	(*)	(*)	(*)
12	(*)	(*)	(*)	(*)
13	(*)	(*)	(*)	(*)
14	(*)	(*)	(*)	(*)

(*) – Program aborted after 8 hours running.

Results and Conclusions

- Electromagnetism-like Mechanism Results

Total Cost

Net	No discounting	Discrete Version 1	Discrete Version 2	Continuous
1	$43.945	$43.537	$43.314	$43.728
2	$337.025	$339.608	$324.178	$326.310
3	$225.952	$218.901	$221.431	$220.266
4	$406.242	$387.221	$388.199	$389.353
5	$138.008	$133,808	$134.877	$135.987
6	$263.557	$253.173	$248.337	$251.678
7	$158.929	$156.139	$155.176	$156.772
8	$94.510	$94.442	$93.175	$93.681
9	$801.433	$750.093	$743.219	$746.081
10	$106.720	$105.945	$105.298	$105.508
11	$453.402	$443.930	$443.159	$444.894
12	$1,381.696	$1,167.805	$1,157.423	1.175,338
13	$811.971	$795.434	$776.087	$774.685
14	$532.055	$546.510	$511.184	$518.341

Execution Time

Net	No discounting	Discrete Version 1	Discrete Version 2	Continuous
1	0.235s	0.485s	0.468s	0.453s
2	1.109s	1.781s	1.750s	1.718s
3	2.953s	4.906s	4.890s	4.719s
4	7.844s	10.984s	10.937s	10.782s
5	13.891s	19.422s	19.297s	18.922s
6	16.484s	22.140s	21.922s	21.500s
7	25.344s	31.188s	31.078s	30.328s
8	35.531s	43.172s	43.625s	42.781s
9	53.609s	1m 00.047s	59.328s	59.109s
10	1m 39.125s	1m 52.422s	1m 52.250s	1m 49.985s
11	2m 54.750s	3m 02.235s	3m 02.609s	2m 59.797s
12	27m 43.625s	9m 55.781s	10m 02.172s	9m 50.984s
13	55m 18.406s	56m 47.266s	55m 43.859s	54m 46.610s
14	5h 26m 15.860s	5h 28m 26.969s	5h 26m 15.860s	5h 25m 36.891s

Results and Conclusions

- Evolutionary Algorithm Results

Total Cost

Net	No discounting	Discrete Version 1	Discrete Version 2	Continuous
1	$44.499	$44.065	$44.097	$44.469
2	$343.077	$336.700	$338.077	$348.476
3	$238.832	$233.379	$229.840	$227.811
4	$413.791	$406.344	$402.645	$407.611
5	$148.573	$156.557	$143.630	$142.791
6	$266.330	$257.522	$255.039	$262.052
7	$166.982	$164.889	$160.329	$166.183
8	$106.403	$102.360	$102.608	$97.008
9	$814.795	$785.968	$787.030	$788.552
10	$116.157	$111.512	$113.428	$112.399
11	$489.945	$470.083	$475.137	$475.716
12	$1,518.377	$1,430.934	$1,437.272	$1,434.103
13	$903.669	$829.685	$830,003	$824.082
14	$569.911	$549.359	$551.289	$525.174

Execution Time

Net	No discounting	Discrete Version 1	Discrete Version 2	Continuous
1	0.172s	0.390s	0.406s	0.375s
2	0.594s	1.046s	1.031s	1.016s
3	1.469s	2.265s	2.250s	2.157s
4	3.187s	4.406s	4.328s	4.297s
5	4.984s	6.516s	6.735s	6.500s
6	5.875s	7.500s	7.531s	7.437s
7	8.593s	9.906s	10.156s	9.844s
8	10.985s	12.828s	12.891s	12.875s
9	15.391s	17.735s	18.062s	17.328s
10	25.812s	28.578s	28.797s	27.984s
11	43.344s	45.266s	46.422s	44.766s
12	5m 29.094s	1m 58.485s	2m 00.656s	1m 55.766s
13	7m 29.469s	7m 27.453s	7m 59.016s	7m 05.031s
14	35m 59.500s	37m 08.063s	38m 41.781s	36m 34.500s

Results and Conclusions

- The costs without considering the time value of the money are higher than the costs obtained when using discounting; and the execution times are smaller.
- The Discrete Time Approach model with a daily time interval (discrete version 1) is similar to the Continuous Time Approach model.
- When we analyze the total cost results for EM and EA we see some difference in the results, due to the random component presented in these two algorithms.

Results and Conclusions

- For the smaller networks, DP achieved better results than EM and EA, but when networks increase their number of activities, DP results are worst than EM and EA, in terms of cost and also in terms of execution times.
- Comparing the EM and the EA algorithm, we can conclude that EM reached better results in terms of cost, but EA was faster.

Results and Conclusions

- We presented the results for the resource allocation problem in stochastic activity networks as in previous work, but introduced a new component on the models: the time value of money.
- This model may be better suited for representing real life situations, when this factor is important to be considered.

References

- Birbil, I., and Fang, S.-C. (2003). An electromagnetism-like mechanism for global optimization. *Journal of Global Optimization* 25, 263-282.
- Birbil, I., Fang, S.-C., and Sheu, R.-L. (2004). On the Convergence of a Population-Based Global Optimization Algorithm. *Journal of Global Optimization* 30, 301-318.
- Costa, L., and Oliveira, P. (2001). Evolutionary algorithms approach to the solution of mixed integer non-linear programming problems. *Computers & Chemical Engineering* 25, 257-266.
- Tereso, A., Araújo, M., and Elmaghraby, S. (2004a). Adaptive resource allocation in multimodal activity networks. *International Journal of Production Economics* 92, 1-10.
- Tereso, A., Mota, J., and Lameiro, R. (2006b). Adaptive resource allocation to stochastic multimodal projects: a distributed platform implementation in Java. *Control and Cybernetics* 35, 661-686.
- Tereso, A., Costa, L., Novais, R., Araújo, and Madalena (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*, Beijing – China.
- Tereso, A., Novais, R., Araújo, M., and Elmaghraby, S. (2009). Optimal resource allocation in stochastic activity networks via the electromagnetic approach: a platform implementation in Java. *Control & Cybernetics* 38, 745-782.