# The Optimal Resource Allocation in Stochastic Activity Networks via the Evolutionary Approach: a platform implementation in Java.


Anabela P. Tereso

anabelat@dps.uminho.pt

Lino A. Costa

lac@dps.uminho.pt

Rui A. Novais

rui.fafe@gmail.com

M. Madalena T. Araújo

mmaraujo@dps.uminho.pt

University of Minho - Portugal

# Topics

1 – Problem Definition

2 – Research Lines

3 – Evolutionary Approach

4 – Application Development

5 – Results

6 – Conclusions

7 – Future Research

# 1. Problem Definition (1/3)

- Given a multimodal activity network under stochastic conditions, we want to optimize the resource allocation in order to minimize the cost

# 1. Problem Definition (1/3)

- Given a multimodal activity network under stochastic conditions, we want to optimize the resource allocation in order to minimize the cost

# 1. Problem Definition (2/3)

- Work Content

  $W_a \sim \exp(\lambda_a)$

- Resource Allocation

  $l_a \leq x_a \leq u_a$

- Duration

  $Y_a = W_a / X_a$

- Resource Cost

  $RC_a = x_a W_a$

# 1. Problem Definition (3/3)

- Due date

  $T$

- Tardiness Cost

  $TC = c_L \max\{0, Y_n - T\}$

- One resource
- Goal: determine the resource allocation vector X, such that the total expected cost is minimized

# 2. Research Lines

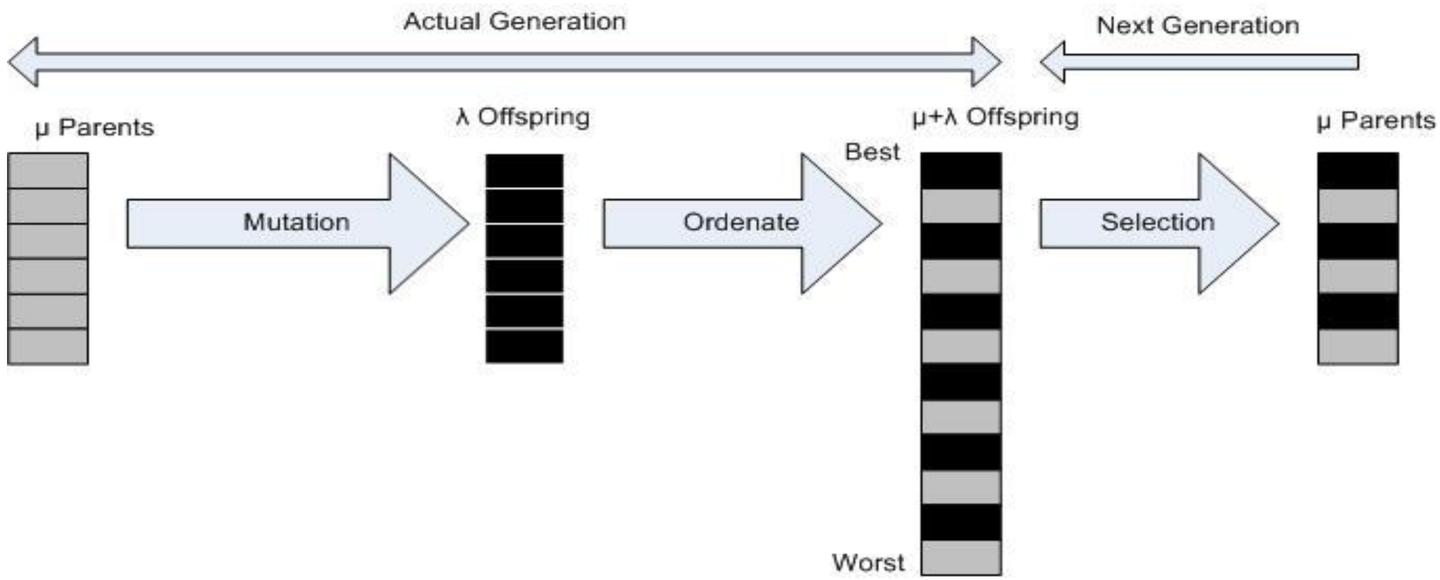
- DP model
- Approximation still using DP, and NLP
- Electromagnetism Approach
- Evolutionary Approach

# 3. Evolutionary Approach (1/4)

- Global Optimization Technique
- Multimodal and nonconvex problems
- Mimic the natural evolution of species
- Based on Evolution Strategies
- Information required: objective function and constraints
- Evolution Strategies more efficient than Genetic Algorithms in number of objective function evaluations
- Initial population / Deterministic transition rules
- Parent population(μ) / Offspring population(ג)
- Individuals: vectors of real coded decision variables
- Recombination and Mutation
- Best individuals selected for next generation

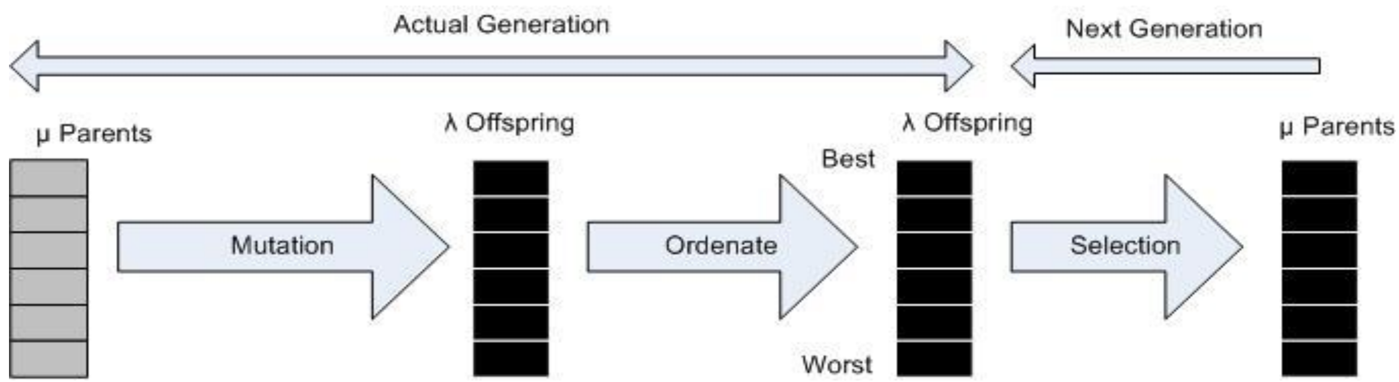
# 3. Evolutionary Approach (2/4)

- (μ+λ) nomenclature



# 3. Evolutionary Approach (3/4)

- (μ,λ) nomenclature



# 3. Evolutionary Approach (4/4)

- Adaptation to the RCPSP

  (Individual: vector of resource allocations)

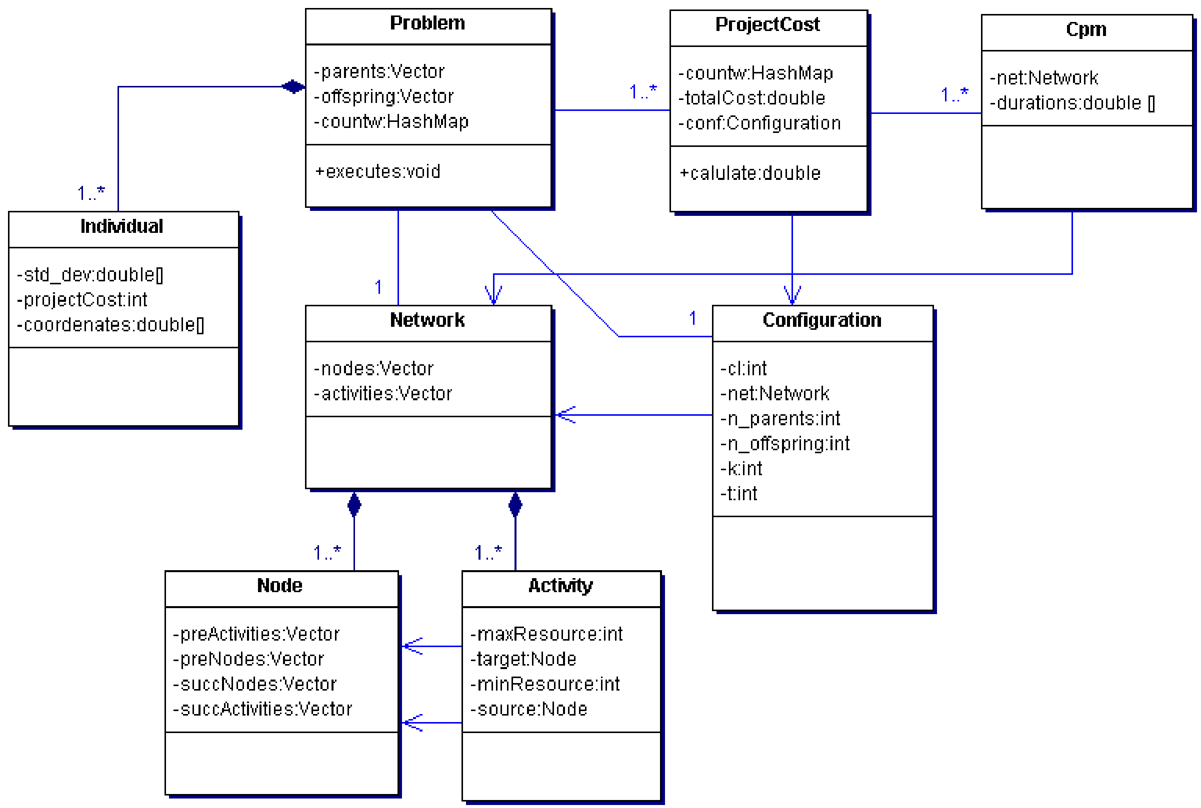
1. Generate k vectors of $W=(w_1..w_n)$ randomly
2. Generate initial population: μ vectors of $X=(x_1..x_n)$ to start with
3. For each vector X
4. For each vector W
5. $rc=\sum x_a W_a$; $tc = c_L \max\{0, Y_n-T\}$; $c=rc+tc$
6. $f = \sum c / k$;
7. Generate offspring population by mutation/recombination: λ vectors of $X=(x_1..x_n)$
8. Next generation: best μ vectors
9. Go to step 3 until nº of iterations specified is reached

# 4. Application Development (1/3)

- Programming language: Java
- Main classes created:
  - Node
  - Activity
  - Network
  - Individual
  - ProjectCost
  - Problem
  - Configuration

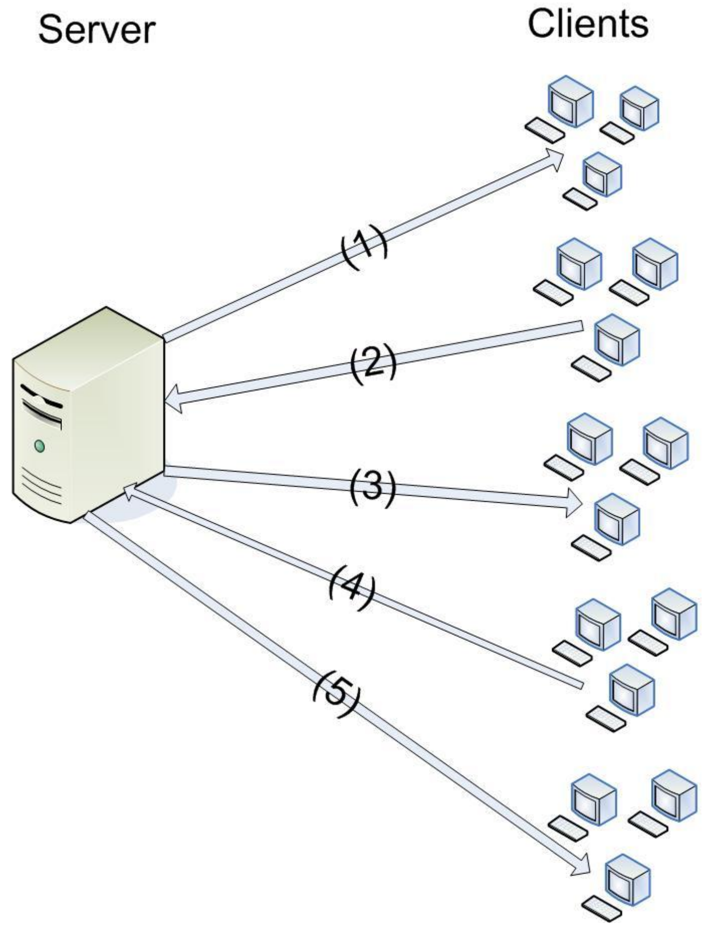
# 4. Application Development (2/3)

## Class Diagram



# 4. Application Development (3/3)

**Distributed implementation of EVA**



# 5. Results (1/4)

**Experiment Layout**

| Network | Number of activities | T | $c_L$ |
|---|---|---|---|
| 1 | 3 | 16 | 2 |
| 2 | 5 | 120 | 8 |
| 3 | 7 | 66 | 5 |
| 4 | 9 | 105 | 4 |
| 5 | 11 | 28 | 8 |
| 6 | 11 | 65 | 5 |
| 7 | 12 | 47 | 4 |
| 8 | 14 | 37 | 3 |
| 9 | 14 | 188 | 6 |
| 10 | 17 | 49 | 7 |
| 11 | 18 | 110 | 10 |
| 12 | 24 | 223 | 12 |
| 13 | 38 | 151 | 5 |
| 14 | 49 | 155 | 5 |

# 5. Results (2/4)

**Experiment Layout**

| PARAMETER | EMA | EVA |
|---|---|---|
| m | 15 | ----- |
| popsizepar | ------ | 15 |
| popsizeoff | ------ | 15 |
| poprecomb | ------ | 15 |
| delta | 0.05 | ----- |
| pertpar | 0.25 | ----- |
| localiterations | 1 | ----- |

# 5. Results (3/4)

**Single Mode Results**
**k=10; 50,000 evaluations**

| Network | Total Cost (EMA) | Run Time (EMA) | Total Cost (EVA) | Run Time (EVA) |
|---|---|---|---|---|
| 1 | 24.39 | 4.64s | 24.24 | 3.41s |
| 2 | 492.00 | 8.25s | 491.53 | 7.00s |
| 3 | 200.10 | 14.52s | 206.35 | 13.42s |
| 4 | 395.62 | 24.75s | 392.26 | 23.68s |
| 5 | 132.76 | 33.94s | 132.80 | 32.89s |
| 6 | 473.90 | 40.14s | 458.68 | 39.28s |
| 7 | 327.18 | 53.36s | 326.09 | 52.21s |
| 8 | 122.32 | 58.95s | 119.03 | 58.20s |
| 9 | 242.18 | 1m 26s | 242.91 | 1m 25s |
| 10 | 134.47 | 2m 02s | 128.38 | 2m 01s |
| 11 | 238.87 | 3m 14s | 232.42 | 3m 14s |
| 12 | 110.89 | 6m 42s | 71.66 | 6m 46s |
| 13 | 996.74 | 18m 45s | 1028.89 | 18m 57s |
| 14 | 526.53 | 2h 11m 24s | 518.86 | 2h 13m 19s |

# 5. Results (4/4)

**Distributed Mode Results**

**EMA, k=600**

| Network | Run Time (SM) | Run Time (DM, Cli=4) | Run Time (DM, Cli=6) |
|---|---|---|---|
| 10 | 2h 12m 03s | 7h 59m 06s | 6h 10m 50s |
| 11 | 3h 28m 32s | 8h 01m 48s | 6h 13m 10s |
| 12 | 4h 00m 29s | 8h 09m 40s | 6h 18m 55s |
| 13 | 20h 36m 02s | 10h 20m 07s | 7h 19m 10s |
| 14 | 27h 56m 40s | 23h 30m 48s | 18h 12m 02s |

**EVA, k=600**

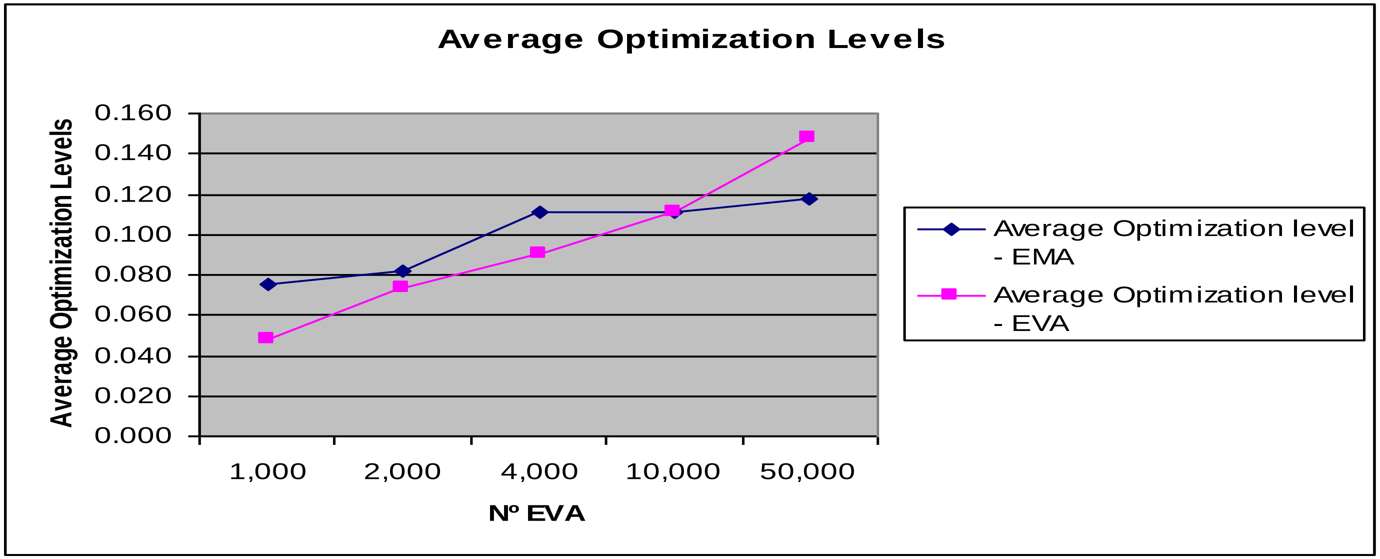
| Network | Run Time (SM) | Run Time (DM, Cli=4) | Run Time (DM, Cli=6) |
|---|---|---|---|
| 10 | 2h 19m 12s | 8h 14m 24s | 6h 22m 48s |
| 11 | 3h 32m 24s | 8h 15m 00s | 6h 23m 28s |
| 12 | 4h 08m 13s | 8h 17m 51s | 6h 25m 20s |
| 13 | 21h 18m 04s | 10h 25m 32s | 7h 24m 36s |
| 14 | 28 h 10m 02s | 24h 31m 38s | 18h 59m 48s |

# 6. Conclusions (1/4)

$$Optim.Level = 1 - (\frac{BestOF}{BestIP}),$$

**Optimization Levels**

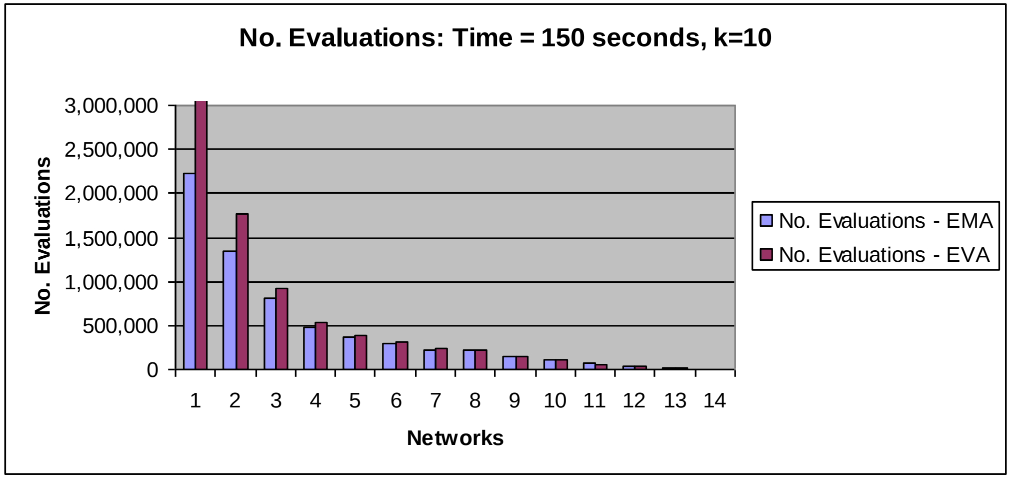
| No. Evaluations | Average EMA Optimization Level | Average EVA Optimization Level |
|---|---|---|
| 1,000 | 0.075 | 0.048 |
| 2,000 | 0.082 | 0.070 |
| 4,000 | 0.111 | 0.090 |
| 10,000 | 0.111 | 0.111 |
| 50,000 | 0.117 | 0.148 |



# 6. Conclusions (2/4)

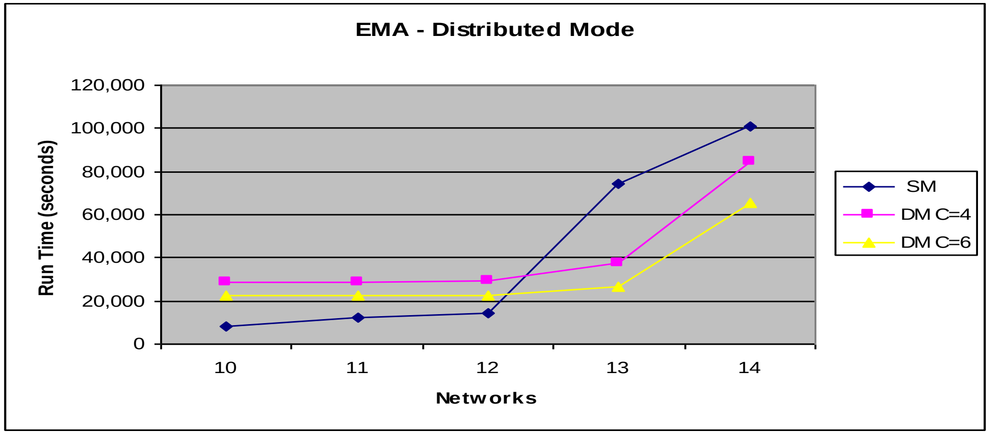
**Algorithms Speed Engine: No. Evaluations (Time = 150 seconds, k=10)**

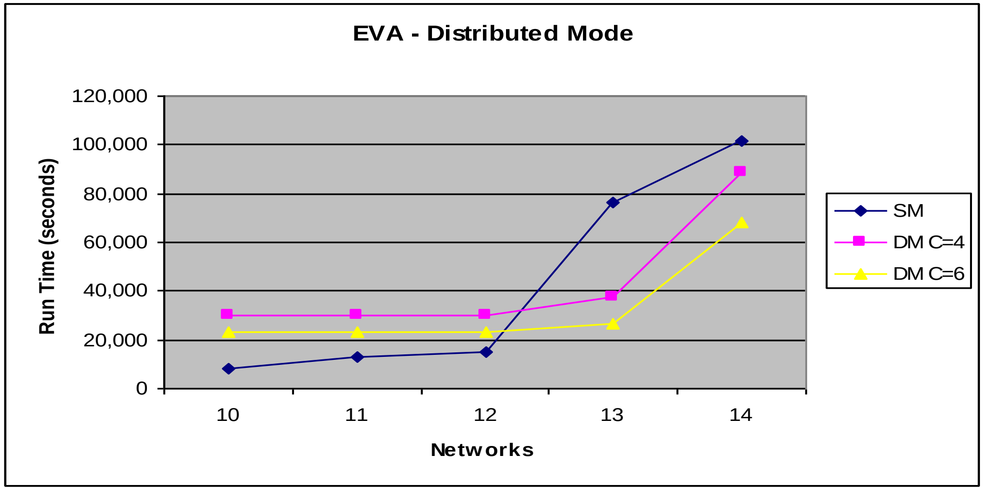
| Network | EMA | EVA | Network | EMA | EVA |
|---|---|---|---|---|---|
| 1 | 2,222,356 | 3,612,175 | 8 | 214,256 | 221,340 |
| 2 | 1,334,885 | 1,764,640 | 9 | 142,514 | 145,860 |
| 3 | 811,970 | 923,315 | 10 | 102,543 | 106,260 |
| 4 | 487,511 | 531,325 | 11 | 64,758 | 62,800 |
| 5 | 370,623 | 391,170 | 12 | 31,590 | 32,340 |
| 6 | 293,973 | 313,290 | 13 | 11,476 | 11,900 |
| 7 | 228,148 | 244,900 | 14 | 1,786 | 2,540 |



# 6. Conclusions (3/4)

## EMA/EVA Run Time



# 6. Conclusions (4/4)

- EMA and EVA are very similar in solving this particular problem
- EMA better for a low number of evaluations
- EVA better for a high number of evaluations
- Since the EVA does a higher number of evaluations than the EMA, for the same running time, we may conclude that the EVA engine is more efficient
- Running in the distributed mode, the EMA produces better results
- When solving this particular problem, these algorithms performed better the Dynamic Programming Model

# 7. Future research

- Use other probability distributions
- Extend the problem to have more than one resource
- Inject the concept of “Intentional Delays” into the problem
  - Use discount factors
  - What is the optimal delay on each activity
  - Minimize the present value of the project

# Major References


1. Costa, L. A. and Oliveira P. (2001). “Evolutionary Algorithms Approach to the Solution of Mixed Integer Non-Linear Programming Problems”, Computers Chem. Engng., 25, 257-266.
2. Birbil, S.I. and Fang, S.-C, (2003), An Electromagnetism-like Mechanism for Global Optimization, Journal of Global Optimization, 25, 263-282
3. Tereso, A. P., Araújo, M. M. and Elmaghraby, S.E. (2004) “Adaptive Resource Allocation in Multimodal Activity Networks”, International Journal of Production Economics, 92:1-10, 2004.
4. Tereso, A. P., Araújo M. M. (2004) “The Optimal Allocation in Stochastic Activity Networks via the Electromagnetism Approach”, Preceding of the Project Management and Scheduling ’04, Nancy - France.
5. Tereso, A. P., Mota, J. R. and Lameiro, R. J. (2005) “Adaptive Resource Allocation Technique to Stochastic Multimodal Projects: a distributed platform implementation in Java”, accepted for publication in the Dynamic Programming Special Issue of the Journal of Control and Cybernetics, June 2005.