

Bernabé Dorronsororo

Patricia Ruiz

[bernabe.dorronsororo@uca.es](mailto:bernabe.dorronsororo@uca.es)

[www.bernabe.dorronsororo.es](http://www.bernabe.dorronsororo.es)

[patricia.ruiz@uca.es](mailto:patricia.ruiz@uca.es)

[patriciaruiz.es](http://patriciaruiz.es)



# Lab Assignment

- Trabajo propuesto: Buscar alguna solución de fitness que pueda permitir al algoritmo encontrar soluciones más robustas que la configuración original de AEDB
  - Es posible intentar otras técnicas para encontrar soluciones robustas
  - También se puede proponer (lo antes posible) otro trabajo relacionado con la optimización en MANETs

- Parámetros para el algoritmo de optimización
- Densidad:
  - 100dev/km<sup>2</sup>: 25
  - 200dev/km<sup>2</sup>: 50
  - 300dev/km<sup>2</sup>: 75
- Librería jMetal
- Algoritmo: `coevolutionary.metaheuristics.nsgall.NSGAll_main.java`
- Problema: `coevolutionary.problems.NS3custom.java`
- Resultados, frente de Pareto:
  - Fitness: FUN.0
  - Variables: VAR.0
- El paquete `experiments` puede ayudar con el procesamiento de los resultados y los tests estadísticos!

- Ficheros que se entregan:
  - libns3.so: biblioteca ns3 que implementa AEDB
  - a.out: programa C que ejecuta el protocolo AEDB con los parámetros especificados:
    - ▶ Densidad de la red (25, 50, o 75 para 100, 200 y 300 disp/km<sup>2</sup>, resp.)
    - ▶ Número de simulaciones
  - jmetal/: librería jMetal
  - lanza.sh: ejemplo para ejecutar el algoritmo
  - DecodeVars.java: Fichero necesario para descodificar los valores de las variables del frente de Pareto, del fichero VAR.\*

- Configuración original de AEDB contra la que comparar:
  - Densidad: minDelay maxDelay borderT marginT neighbT
  - 100 dev/km<sup>2</sup>: 0.0 1.0 -90.0 0.5 8
  - 200 dev/km<sup>2</sup>: 0.0 1.0 -90.0 0.5 10
  - 300 dev/km<sup>2</sup>: 0.0 1.0 -90.0 0.5 12
- Ejemplo de ejecución para la red:
  - Dispersa LD\_LIBRARY\_PATH=./a.out 25 10 0.0 1.0 90.0 0.5 8
  - Intermedia LD\_LIBRARY\_PATH=./a.out 50 10 0.0 1.0 90.0 0.5 8
  - Densa LD\_LIBRARY\_PATH=./a.out 75 10 0.0 1.0 90.0 0.5 8