

Técnicas de Inteligencia Artificial

Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital (DSIC-UPV)

Prácticas de Laboratorio

Teoría. Lunes (15:00-17:30)		Laboratorio. Jueves (15:00 -17:30 , 17:30-20:00) (PL-2: Lab DSIC-3, PL-1: Sem 0S02)
Presentación. Búsqueda de Soluciones en IA Tipología de la búsqueda Heurísticos y Metaheurísticos	14-IX	Presentación. Propuestas de trabajos y entornos
Metaheurísticas Poblacionales (Evolutivas) Algoritmos Genéticos Algoritmos meméticos. Búsqueda dispersa	21-IX	<u>Algoritmos genéticos.</u> Diseño e implementación
Metaheurísticos de Mejora Iterativa Búsqueda Tabú. Enfriamiento Simulado Búsqueda en Haz	28-IX	Implementación y evaluación
Metaheurísticas Constructivas GRASP Variantes	5-X	<u>Enfriamiento simulado.</u> Diseño e implementación
Inteligencia Social y Colectiva de Enjambre Alg. Hormigas. Colonia de Abejas Enjambre de Partículas (PSO) Otras variantes	19-X	Implementación y evaluación
Prueba escrita / evaluación teoría	26-X	Presentación de trabajos y discusión de resultados

1) Diseñar y desarrollar soluciones metaheurísticas para resolver problemas propuestos

- Algoritmo Genético
- Enfriamiento Simulado
- Otras técnicas y/o problemas (a elección del alumnado)

2) Evaluar y contrastar la utilidad de los métodos desarrollados

- Diseño. Contraste y evaluación de parámetros
- Escalabilidad : Evaluación soluciones frente a tamaños del problema.
- Evaluación: nº de soluciones generadas vs. calidad solución, Otras medidas.

Poliformat:

- Presentación.
- Propuestas
- Opt4J: Entorno, Boletín, Docs

Opciones. Diseño y desarrollo soluciones metaheurísticas para resolver problemas

1) Diseño y codificación metaheurística (lenguaje de elección como Java, C#, Python, etc.)

¡La implementación de los algoritmos no es compleja!

Ventaja: mayor conocimiento y control;

Desventaja: Hay que implementar los procesos

2) Entornos Generalistas

Nos centramos en el modelado del problema, no en la implementación del algoritmo

Ventaja: utilización de las facilidades del entorno que no hay que implementar;

Desventaja: ¡hay que conocerlo!, limitaciones del entorno, menor control y conocimiento

3) Librerías. Requiere conocer el diseño realizado de la metaheurística e integrarlo con el fragmento de código que implementamos

Nos centraremos en el modelado del problema, pero hemos de integrarlo y utilizar la API que nos ofrece.

Ventaja: tenemos APIs que nos permiten personalizar nuestro proceso de resolución;

Desventaja: ¡hay que conocerlo!

Opt4J. Entorno libre

A Modular Framework for Meta-heuristic Optimization

Disponible (ejecutable y fuente) en: <https://sdarg.github.io/opt4j/>

Formulación sencilla de problemas utilizando librerías implementadas en Java

Existe un boletín que explica su instalación, uso e integración en ECLIPSE (java).



Otros entornos libres:

HeuristicLab <http://dev.heuristiclab.com/>



HeuristicLab
A Paradigm-Independent and Extensible
Environment for Heuristic Optimization

Entorno generalista en C#

PARADISEO <https://nojhan.github.io/paradiseo/>



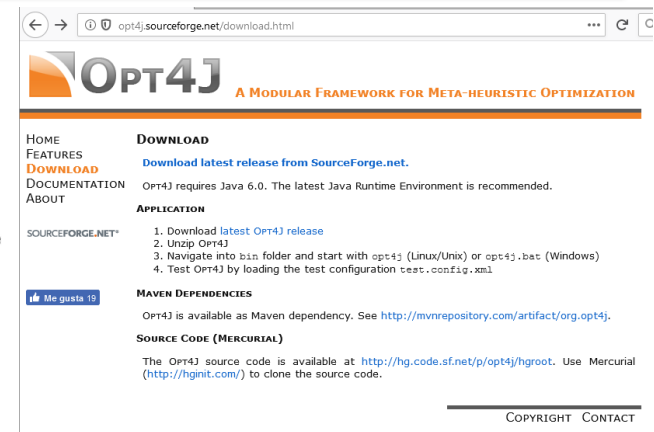
API de programación en C++

jMetal <https://jmetal.sourceforge.net/>



Framework de optimización multi-objetivo en Java

Etc.



MATLAB & Global Optimization Toolbox. Sistema Comercial MatLaB

Enlaces:

<http://es.mathworks.com/products/global-optimization/>

<http://es.mathworks.com/help/gads/index.html>

<https://es.mathworks.com/help/gads/genetic-algorithm.html>



MATLAB (toolBox disponibles). Sistema Comercial. Lenguaje MatLaB

Toolbox libres sobre Genetic Algorithms:

GPLAB: <http://gplab.sourceforge.net/>

GA_framework: <https://sourceforge.net/projects/gatoolbox/>

Toolbox libres sobre Simulated Annealing:

<http://es.mathworks.com/matlabcentral/fileexchange/10548-general-simulated-annealing-algorithm>

Toolbox libres sobre Particle Swarm Optimization:

<http://psotoolbox.sourceforge.net/>

In: *JCLEC-MO: A Java suite for solving many-objective optimization engineering problems*,

A. Ramírez, J. Romero ,
C. García, S. Ventura.
Engineering App. of
Artificial Intelligence 81
(2019).

<https://doi.org/10.1016/j.engappai.2019.02.003>

Table 1

Summary of the characteristics of general-purpose MOFs.

Characteristic	ECJ v25 (2017)	HeuristicLab v3.3.15 (2018)	EvA v2.2 (2015)
Metaheuristics	DE, EDA, ES, GA, GE, GP, PSO	ES, GA, GE, GP, LS, PSO, SS, TS, SA, VNS	DE, EP, ES, GA, GP, HC, PSO, SS, SA
Encodings	Binary, integer, real, tree	Binary, integer, real, tree	Binary, integer, real, tree
Optimization problems	c/u, min/max	u, min/max	c/u, min
MOO algorithms	NSGA-II, SPEA2	MO-CMAES, NSGA-II	MO-CMAES, MOGA, NSGA, NSGA-II, PESA, PESA-II, Random Weight GA, SPEA, SPEA2, VEGA
MOO benchmarks	Fors.&Flem., Kursawe, Poloni, Quagli.&Vicini, Schaffer, Sphere, ZDT	Fonseca, Kursawe, Schaffer, DTLZ, ZDT	TF
Quality indicators		GD, HV, Spacing	ER, GD, HV, Max. PF error, ONVG
Characteristic	Opt4J v3.1.4 (2015)	PaGMO v2.6 (2017)	JCLEC v4 (2014)
Metaheuristics	DE, GA, PSO, SA	ABC, DE, ES, GA, PSO, SA	GA, GP
Encodings	Binary, integer, real	Integer, real, mixed	Binary, integer, real, tree
Optimization problems	c/u, min/max	c/u, min	min/max
MOO algorithms	NSGA-II, SPEA2, SMS-EMOA, OMOPSO	MOEA/D, NSGA-II	NSGA-II, SPEA2
MOO benchmarks	DTLZ, Knapsack, LOTZ, Queens, WFG, ZDT	DTLZ, ZDT	
Quality indicators	HV	HV	

ABC: Artificial bee colony, DE: Differential evolution, EDA: Estimation of distribution algorithms, EP: Evolutionary programming, ES: Evolution strategy, GA: Genetic algorithm, GE: Grammatical evolution, GP: Genetic programming, HC: Hill climbing, LS: Local search, SS: Scatter search, SA: Simulated annealing, TS: Tabu search, VNS: Variable neighborhood search, c: Constrained, u: Unconstrained, min: Minimization, max: Maximization.

Java

ECJ (Java-based Evolutionary Computation Research System)

<http://cs.gmu.edu/~eclab/projects/ecj/>

MOEA (Free and Open Source Java Framework for Multiobjective Optimization)

<http://moeaframework.org/>

C++

ECF - Evolutionary Computation Framework

<http://ecf.zemris.fer.hr/>

GAlib (C++ Library of Genetic Algorithms)

<http://lancet.mit.edu/ga/>

Open BEAGLE: A Generic Evolutionary Computation Framework in C++

<http://chgagne.github.io/beagle/>

METSlib - metaheuristics optimization framework

<https://github.com/coin-or/metslib> y <https://www.coin-or.org/Doxygen/metslib/index.html>

Python

Distributed Evolutionary Algorithms (DEAP)

<https://github.com/deap>

Pyevolve

<http://pyevolve.sourceforge.net/>

1. Algoritmos Genéticos

Diseño Algoritmo Genético

- Diseño del individuo. Codificación y decodificación de soluciones.
- Función de evaluación (fitness)
- Generación población inicial.
- Procesos: Selección. Cruce (individuos inválidos). Mutación. Reemplazo.

Evaluación algoritmo genético

- Criterios de evaluación: Fitness versus Soluciones generadas, Tiempo cómputo.
- Tamaños del problema. Convergencia.
- Parámetros de evaluación: Población, Selección, Cruce, Mutación, etc.

Conclusiones Algoritmo Genético

2. Enfriamiento simulado

Diseño algoritmo enfriamiento simulado

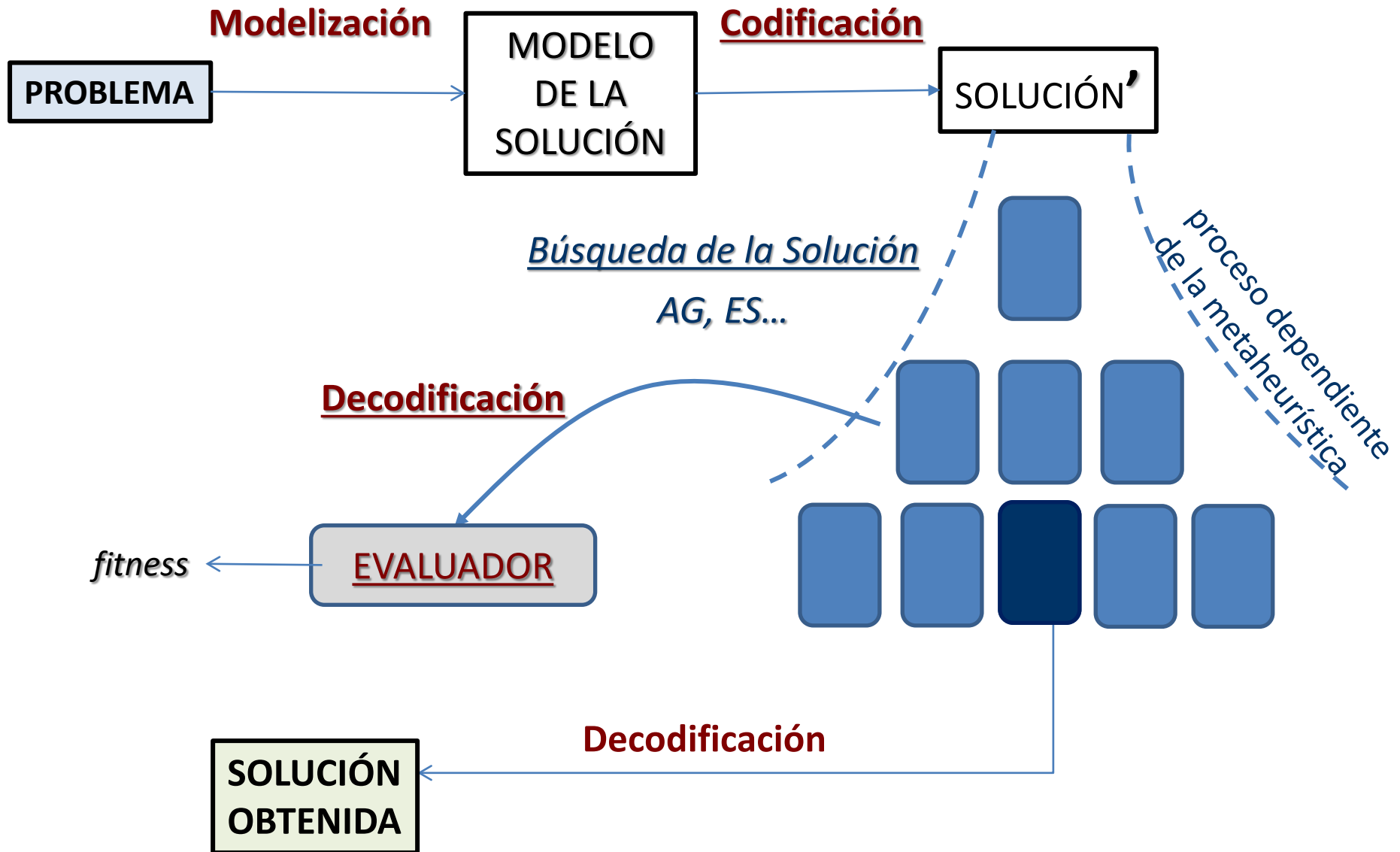
- Generación de soluciones vecinas. Solución inicial.

Evaluación enfriamiento simulado

- Criterios de evaluación: Fitness versus Soluciones generadas, Tiempo cómputo.
- Tamaños del problema. Convergencia.
- Parámetros de evaluación: Temperatura inicial, Función decremento.

Conclusiones Enfriamiento Simulado

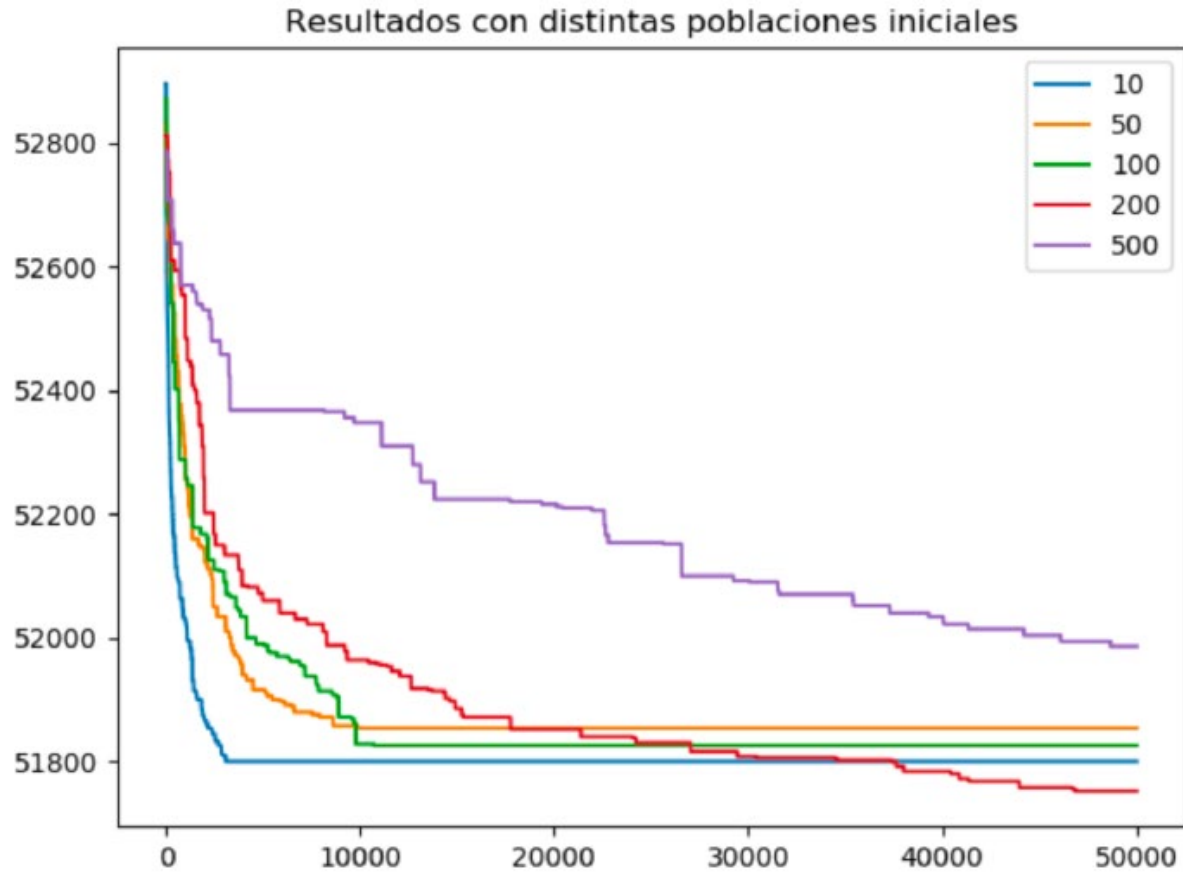
➤ **Comparativa de ambos métodos de optimización. Presentación y discusión de resultados.**



Resultados

Calidad Solución vs {Tiempo de Cómputo, Soluciones probadas (fitness) }

Parametrizado con los diferentes parámetros de la metaheurística



1. Algoritmos Genéticos

Diseño algoritmo genético

- Diseño del individuo. Codificación y decodificación.
- Función de evaluación (fitness)
- Generación población inicial.
- Selección. Cruce (individuos inválidos). Mutación. Reemplazo.

Evaluación algoritmo genético

- Criterios de evaluación: Fitness versus Soluciones generadas, Tiempo cómputo.
- Tamaños del problema. Convergencia.
- Parámetros de evaluación: Población, Selección, Cruce, Mutación, etc.

Conclusiones Algoritmo Genético

Trabajo:

- Cada persona elegirá una de las propuestas del boletín o propondrá una en particular.
- Se podrá implementar el algoritmo mediante cualquier **lenguaje de programación**, utilizar un entorno generalista o librerías existentes. **Se deja a elección del alumno.**

Calendario:

21/09	Algoritmos genéticos
28/09	Algoritmos genéticos

***Recomendación:**
borrador provisional de la memoria
(1ª parte) para el 5/10*

2. Enfriamiento simulado

Diseño algoritmo enfriamiento simulado

- Soluciones vecinas. Solución inicial.

Evaluación enfriamiento simulado

- Criterios de evaluación: Fitness versus Soluciones generadas, Tiempo cómputo.
- Tamaños del problema. Convergencia.
- Parámetros de evaluación: Temperatura inicial, Función decremento.

Conclusiones enfriamiento simulado

Trabajo:

- Se seguirá con el problema elegido para el AG.
 - *Parte del trabajo realizado en el diseño del AG es reutilizable en el diseño del ES.*
 - *Evaluación Comparativa*
- Se podrá implementar el algoritmo mediante cualquier lenguaje de programación, utilizar un entorno generalista o librerías existentes. **Se deja a elección del alumno**

Calendario:

5/10	Enfriamiento simulado
19/10	Enfriamiento simulado

Comparativa. Presentación y discusión de resultados.

Fecha	Laboratorio	Fecha	Laboratorio
21/09	Algoritmos genéticos	5/10	Enfriamiento simulado
28/09	Algoritmos genéticos	19/10	Enfriamiento simulado

Trabajo:

- Entrega Memoria:
 - Algoritmo Genético, Enfriamiento Simulado
 - Diseño, Evaluación, Escalabilidad y Comparativa
- Presentación Individual (*se detallarán recomendaciones*)
- Alternativas de Presentación: Presencial / Vídeo
- Evaluación (por pares) comparativa de otros trabajos

Calendario:

Fecha	Laboratorio
26/10	Memoria + Presentación oral (presencial/vídeo)
Semana siguiente	Finalización de las evaluaciones comparativas

Ejemplos de Aplicaciones