



Grado en Ingeniería Informática en Sistemas de Información  
Inteligencia Artificial - Curso 2017/18  
EXAMEN NOVIEMBRE-28/11/2017

APELLIDOS: \_\_\_\_\_ NOMBRE: \_\_\_\_\_ D.N.I.: \_\_\_\_\_

**IMPORTANTE:** Descarga del Aula Virtual el fichero MaterialExamen\_Busquedas.zip necesario para el desarrollo del examen. Debe entregar a través de la actividad habilitada para el examen un fichero llamado TusApellidos.zip que contenga los ficheros fuentes .java que hayáis creado o modificado para el Ejercicio.

**EJ1.- (5 puntos) Asignación de tareas.** Supongamos que  $n$  trabajadores tienen que realizar  $n$  tareas, y que conocemos el tiempo  $q_{ij}$  de realización por parte del trabajador  $i$ -ésimo ( $t_i$ ) de la tarea  $j$ -ésima ( $T_j$ ). El problema consiste en cómo asignar a cada trabajador una y solo una tarea, de manera que se realicen todas las tareas en un tiempo total mínimo. Una tarea no puede ser realizada por dos trabajadores, provocando que otra tarea quede sin hacer. A continuación, presentamos como ejemplo para  $n = 4$  una tabla  $Q = (q_{ij})$  con los tiempos que cada trabajador necesita para realizar cada una de las tareas:

		Tareas			
Trabajador	Q	$T_1$	$T_2$	$T_3$	$T_4$
	$t_1$	12	43	15	7
	$t_2$	9	10	6	4
	$t_3$	5	13	29	2
	$t_4$	4	11	17	9

Resolver el problema mediante un algoritmo genético utilizando la librería AIMA. Se facilitan las clases java a implementar, conteniendo parte del código ya desarrollado:

- **GeneticTareasFitnessFunction:** implementar la función de evaluación de los individuos *getValue*; y la generación aleatoria de individuos, *generateRandomIndividual*, siendo de gran ayuda la función *Collections.shuffle* para asignar aleatoriamente cada tarea a cada trabajador.
- **GeneticTareasGoalTest:** clase que indica si el estado que se pasa como parámetro al método es un estado objetivo de la búsqueda.
- Modifique la clase de AIMA *GeneticAlgorithm*, o crear en el mismo paquete una copia con el nombre *GeneticAlgorithmTareas*, de manera que:
  - El mejor individuo de la población pase a la siguiente generación.
  - La reproducción se hará a partir de un solo individuo obtenido previamente con *randomSelection*. Fijando aleatoriamente un punto de corte en el individuo padre, el nuevo individuo estará formado por una copia literal de una de las dos partes del individuo padre, y por la otra parte del individuo padre reordenada de manera aleatoria. Igualmente, la parte fija se echará a suerte.
  - Por último, en el proceso de mutación, se intercambiará el contenido de dos posiciones del individuo elegidas aleatoriamente.
- **TareasDemo:** finalmente, incluir en esta clase lo necesario para probar la resolución del problema, teniendo en cuenta que la población inicial debe contener 10 individuos.