

Práctica 4

Algoritmos genéticos

Fecha de entrega: 12 de diciembre del 2017

El objetivo de esta práctica es implementar la resolución de un problema mediante algoritmos genéticos y realizar pequeñas modificaciones en el algoritmo genético básico para estudiar su impacto.

1. Algoritmos genéticos en la librería Java de AIMA

La librería de AIMA implementa la búsqueda local mediante algoritmos genéticos en la clase `GeneticAlgorithm.java` (`aima.core.search.local`).

Revisad como está implementado el algoritmo genético:

- ¿Qué tipo de representaciones del problema admite el algoritmo?
- ¿Qué método de selección utiliza?
- ¿Cómo realiza el cruce?
- ¿Cómo se realiza la mutación?
- ¿Qué valores se pueden configurar en el algoritmo?

Como ejemplo de uso del algoritmo genético la librería implementa la resolución del problema de las n reinas en la clase `NQueensDemo` (`aima.gui.demo.search`), en el método `nQueensGeneticAlgorithmSearch()`.

Revisad como está implementada la demo de las n reinas con algoritmos genéticos:

- ¿En qué clase está implementada la función de fitness? ¿Cómo está implementada?
- ¿En qué clase está implementada la función objetivo? ¿Cómo está implementada?
- ¿Cómo se genera la población inicial?

2. Problema de asignación de turnos a los exámenes CFI

Se desea diseñar una aplicación para la asignación de turnos de examen CFI a profesores. Los CFI¹ son cursos online que se imparten a través del Campus Virtual de la UCM. Se ofrecen una gran variedad de cursos en distintas áreas temáticas: Ofimática, Internet o Tecnologías Informáticas. Al finalizar el curso se obtiene un Título oficial de Aprovechamiento del Curso sin necesidad de examen. Si se quiere obtener un diploma de aptitud y/o un certificado para el reconocimiento de créditos, entonces se debe realizar un examen final presencial. A lo largo del curso hay un total de 5 convocatorias

¹ <http://cursosinformatica.ucm.es/>

de examen. En cada convocatoria de examen deben de estar presentes algunos de los profesores de los cursos.

La asignación de los profesores a cada turno la realiza actualmente la directora académica de forma manual. Antes de cada convocatoria se estima el número de turnos que serán necesarios y se pide a los profesores sus restricciones y preferencias. Con estos datos se asignan de forma manual los profesores a los turnos necesarios.

La directora académica piensa que el uso de algoritmos genéticos podría ayudarle en su labor, sobre todo cuando el número de turnos y restricciones crece y la asignación se vuelve muy complicada, y nos ha pedido ayuda. Para intentar ayudarle vamos a crear una aplicación que ayudándose de la librería de AIMA haga de forma automática la asignación de profesores a turnos.

2.1. Turnos de examen

Los turnos podrán ser de lunes a jueves y cada día puede haber 4 turnos: 2 de mañana (10:30 y 12:00) y 2 de tarde (15:00 y 16:30). Para facilitar la gestión de los turnos y las restricciones cada turno será representado mediante un número tal y como se muestra en la Tabla 1.

| | Lunes | Martes | Miércoles | Jueves |
|-------|-------|--------|-----------|--------|
| 10:30 | 1 | 5 | 9 | 13 |
| 12:00 | 2 | 6 | 10 | 14 |
| 15:00 | 3 | 7 | 11 | 15 |
| 16:30 | 4 | 8 | 12 | 16 |

Tabla 1. Turnos disponibles y número asignado a cada turno

2.2. Datos de entrada

- Número de turnos necesarios. Será un entero que indique a cuantos turnos hay que asignar profesor.
- Listado de profesores.
- Restricciones de los profesores. Lista con todos los profesores y sus restricciones. Las restricciones de cada profesor será el conjunto de turnos a los que **no** podrán asistir. Por ejemplo, si un profesor no puede hacer ningún turno el martes mandará la siguiente lista de turnos como restricciones: 5,6,7,8.
- Preferencias de los profesores. Listado con todos los profesores y sus preferencias. Por ejemplo, si un profesor puede asistir a todos los turnos (es decir, su lista de restricciones está vacía) pero prefiere hacer los turnos de

mañana, mandará la siguiente lista de turnos como preferencias: 1, 2, 5, 6, 9, 10, 16, 14.

2.3. Datos de salida

El programa devolverá una asignación válida de profesores a turnos. Para que la asignación sea válida se debe de cumplir lo siguiente:

- El número de turnos asignados coincide con el número de turnos necesarios.
- Se cumplen todas las restricciones dadas por los profesores. Es decir, no se ha asignado a ningún profesor uno de los turnos de su lista de restricciones.

De todas las soluciones válidas la mejor solución será aquella que:

- Equilibre el número de turnos asignado a cada profesor. Por ejemplo, si hay que cubrir 10 turnos y tenemos 10 profesores, será mejor solución aquella que asigna 1 turno por profesor que aquella que asigna 10 turnos a un solo profesor.
- Cumpla el mayor número de preferencias de los profesores, lo ideal sería que se cumplieran todas.

2.4. Ejemplo de ejecución

Supongamos que nuestro programa recibe los siguientes datos:

- Número de turnos necesarios = 4.
- Listado de profesores: Antonio, Belén, Fernando, Guillermo, José Luis, Juanan, Marlon, Raquel, Virginia.
- Restricciones de los profesores:
 - Antonio:
 - Belén: 3,4,7,8,11,12,15,16
 - Fernando: 1,2
 - Guillermo: 7,10,16
 - Juanan: 1,2,3,4,5,6,7,8,9,10,11,12,13,14
 - Marlon:
 - Raquel: 1,2,5,6,9,10,13,14
 - Virginia: 1,5,9,13
- Preferencias de los profesores:
 - Antonio: 1,2
 - Belén: 1,6
 - Fernando:
 - Guillermo: 3,4
 - Juanan:
 - Marlon:
 - Raquel: 3,4

- Virginia: 2,4

Una solución posible sería la presentada en la Tabla 2.

| | Lunes | Martes | Miércoles | Jueves |
|-------|-----------|--------|-----------|--------|
| 10:30 | | | | |
| 12:00 | Virginia | Belén | | |
| 15:00 | Guillermo | | | |
| 16:30 | Raquel | | | |

Tabla 2. Asignación de turnos a profesores del ejemplo de ejecución

Esta solución es válida ya que:

- Número de turnos = 4
- Se cumplen todas las restricciones de los profesores.

Además, se han cumplido las preferencias de todos los profesores y el número de turnos por profesor está equilibrado (todos los profesores, que tienen turnos asignados, tienen sólo 1 turno asignado).

2.5. Pruebas a realizar

En el campus tenéis 3 archivos con distintas configuraciones del algoritmo, probad con ellas vuestra aplicación y estudiad los resultados obtenidos.

3. Modificación del algoritmo genético de AIMA

Una vez que tengáis el programa de asignación de profesores a turnos con el algoritmo genético de AIMA debéis introducir las siguientes modificaciones en el algoritmo:

1. Introducid una probabilidad de cruce en el algoritmo. Una vez hecha esta modificación probad con los siguientes valores: 0.7, 0.8 y 0.9 y comparad los resultados obtenidos con los resultados del algoritmo original.
2. Modificad el cruce para que devuelva dos hijos en lugar de uno². Una vez hecha esta modificación comparad los resultados obtenidos con los resultados del algoritmo original.
3. Cambiad la estrategia destructiva del algoritmo original de AIMA (los descendientes se insertan en la siguiente generación, aunque sus padres tengan un mejor ajuste) por una estrategia no destructiva (la descendencia pasará a la siguiente generación únicamente si supera la bondad de ajuste de los padres).

² Modificad todo lo necesario para que el algoritmo continúe funcionando correctamente.

Una vez hecha esta modificación comparad los resultados obtenidos mediante la estrategia no destructiva con los resultados obtenidos con el original.

4. Parte opcional

Se podrá obtener puntuación extra realizando las siguientes partes opcionales:

1. Según van avanzando las convocatorias el número de turnos realizados por cada profesor va siendo distinto y este aspecto debería de ser tenido en cuenta a la hora de asignar los turnos. Por ejemplo, si Juanan lleva 0 turnos realizados y Antonio lleva 2, lo lógico sería priorizar la asignación de turnos a Juanan. Para realizar esta parte opcional, habrá que modificar la aplicación para que reciba un parámetro extra (el número de turnos realizados hasta ahora por cada profesor) y para que esta información sea tenida en cuenta en la función de fitness.
2. Puede haber preferencias de los profesores que no puedan ser representadas por el modelo actual. Por ejemplo, es bastante habitual que los profesores, en caso de tener más de un turno asignado, prefieran realizar los turnos de forma consecutiva. Otro ejemplo, sería que los profesores prefieran hacer los turnos en varios días en lugar de en uno solo. Otra cuestión que no permite el sistema descrito en la Sección 2 es dar una prioridad a las preferencias. Para realizar esta parte opcional habría que cambiar la forma en que se dan las preferencias y cambiar la función de fitness para que tenga en cuenta estos nuevos tipos de preferencias.
3. En ocasiones el número de alumnos en cada turno es muy alto y se necesitan dos profesores por turno. Modifica el algoritmo para que asigne dos profesores a cada turno.
4. Modificad el cruce por un punto que realiza el algoritmo de AIMA por un cruce por dos puntos. Una vez hecha esta modificación comparad los resultados obtenidos con los resultados del algoritmo original.
5. Implementación de algún otro mecanismo de selección (selección elitista, por torneo...) y comparación de los resultados obtenidos por el nuevo mecanismo con los obtenidos con el mecanismo original de AIMA.
6. Implementación de algún otro mecanismo de mutación y comparación de los resultados obtenidos por el nuevo mecanismo con los obtenidos con el mecanismo original de AIMA.

5. Memoria

La memoria que debéis de entregar debe de contener el **nombre de los dos integrantes del grupo** y, al menos, lo siguiente:

1. Descripción de como habéis representado los individuos.

2. Descripción de la generación de población inicial.
3. Descripción de la implementación del operador de cruce que habéis realizado.
4. Descripción de la implementación del operador de mutación que habéis realizado.
5. Descripción de la función de fitness empleada en vuestra implementación.
6. Descripción de la función objetivo empleada en vuestra implementación.
7. Resultados obtenidos por el algoritmo desarrollado para las configuraciones proporcionadas.

Para poder analizar la eficiencia del algoritmo no basta con ejecutar una única vez el algoritmo, habrá que ejecutarlo varias veces (100, por ejemplo) y de cada parámetro (fitness mejor individuo, tiempo de ejecución...) obtener media, máximo...

8. Influencia de la probabilidad de cruce en la aplicación.
9. Influencia de obtener dos individuos en el cruce en lugar de uno en la aplicación.
10. Influencia de la estrategia no destructiva frente a la estrategia destructiva en la aplicación.
11. Parte opcional. Se describirá en detalle si se ha realizado alguna de las partes opcionales, como se han realizado, resultados obtenidos...

6. Entrega de la práctica

La práctica se entregará a través del Campus Virtual. Se habilitará una nueva tarea: **Entrega de la Práctica 4** que permitirá subir un zip que contendrá la memoria de la práctica y las fuentes de la aplicación Java creada.

El fichero subido deberá tener el siguiente nombre: **Practica4GXX.zip**, siendo XX el nombre del grupo. Por ejemplo, *Practica4G03.zip*.

Uno sólo de los miembros del grupo será el encargado de subir la práctica.

La fecha límite para entregar la Práctica 3 será el martes **12 de diciembre a las 23:55**. Se recomienda no dejar la entrega para el último momento para evitar problemas de última hora.

No se corregirá ninguna práctica que no cumpla estos requisitos.