

Ejercicio 8. Formación de equipos de trabajo para los TFG en la Facultad de Informática.

Para los proyectos TFG en la Facultad de Informática hemos decidido modificar el método de creación de grupos de trabajo que a partir de ahora será automático. Se quiere diseñar un algoritmo genético para crear estos equipos de trabajo teniendo en cuenta que los equipos mejores han sido los formados por estudiantes que incluyen una combinación de roles equilibrada.

Roles/Personas de tipo: {líder, investigador, programador, escritor, creativo}.

Queremos un programa para formar automáticamente equipos equilibrados entre 5 y 10 personas. Teniendo en cuenta que, para generar 10 equipos de 3 personas, existen 1.208.883.745.669.600.000 configuraciones distintas¹ y que queremos crear equipos para 2000 alumnos nos resulta muy costoso poder resolver el problema de forma óptima por lo que vamos a intentar resolverlo con un algoritmo genético.

Los datos de entrada del algoritmo serán el número total de alumnos (2000), número de roles posibles (5 en nuestro ejemplo), qué rol posee cada estudiante y el número de integrantes por equipo (todos los equipos serán del mismo tamaño entre 5 y 10). Podemos suponer que a cada rol se le asocia un identificador (r0, r1, r2...) y a cada alumno se le asocia un identificador (a0, a1, a2...) y el rol que le corresponde (vamos a suponer que cada alumno tiene un único rol principal).

Se pide:

- Proponer una representación adecuada para resolver el problema utilizando un algoritmo genético.
- Definir una función de fitness que valore la calidad de la solución, en este caso debe valorar las siguientes condiciones:
 - o Relación entre el número de grupos formados y número de grupos pedidos en la entrada. Si tenemos 60 estudiantes y formamos equipos de 5 deberíamos formar 15 grupos. Ejemplo de función de valoración (entre 0 y 1) que da 1 si los grupos formados son los esperados:
$$Score = 1 - \frac{|gruposFormados - gruposEsperados|}{numALumnos - gruposEsperados}$$
 - o Relación entre los estudiantes por grupo y los estudiantes esperados en cada grupo. Se puede usar una fórmula similar a la anterior para cada equipo.
 - o Relación entre los roles que contiene un equipo y los esperados. Si nos piden equipos de 5 miembros y 5 roles, buscaremos que sean todos de un rol distinto. Si nos piden 6 miembros por equipo y 5 roles, buscaremos que estén al menos los 5 roles representados. Se puede suponer que el número de miembros por equipo \geq número de roles.
- Determinar mediante pruebas cual es la mejor configuración del algoritmo para lograr una solución suficientemente buena con un coste computacional adecuado. Se propone realizar pruebas para determinar la mejor combinación de parámetros (hacer pruebas con unas cuantas configuraciones y elegir la mejor, aunque se podría optimizar la combinación de parámetros con otro algoritmo genético)
 - o Tamaño de la población. Indicar cuantas generaciones son necesarias para que converja
 - o Algoritmo de selección y sus parámetros: probabilidades, tamaño K de un torneo, etc...
 - o Algoritmo de cruce utilizado y sus parámetros: número de puntos de cruce, probabilidades.

¹ $(n \cdot m)! / (n!)^m \cdot m!$, n = número de alumnos por equipo y m = número de equipos.