

Asignación Dinámica de Aulas Utilizando Algoritmos Genéticos

Marcelo J. Karanik

marcelo@frre.utn.edu.ar

Grupo de Investigación Sobre Inteligencia Artificial
Universidad Tecnológica Nacional
Facultad Regional Resistencia
Chaco - Argentina

Abstract. En el proyecto se propone la utilización de algoritmos genéticos para realizar la asignación dinámica de aulas en la Facultad Regional Resistencia de la Universidad Tecnológica Nacional. Para poder resolver el problema de la asignación de las aulas a los cursos, se pretende desarrollar un modelo donde cada posible solución sea un individuo que tenga un valor representativo de su aptitud dentro del dominio de aplicación. Una vez obtenidos los individuos de la generación 0, se aplicarán los operadores de selección, cruce y mutación, analizando los resultados que se obtengan con cada método en particular. Los objetivos son maximizar el uso de los recursos, minimizar el desperdicio del espacio físico y lograr la asignación completa de las aulas mejor acondicionadas, generando el mínimo movimiento de alumnos en los cambios de módulo y reasignando las aulas en caso de inasistencias de docentes o alguna circunstancia no prevista. En la primera parte se hace una descripción del problema a tratar. En la segunda, se presentan las etapas del proyecto. En la tercera parte se describe la forma de utilización de los algoritmos genéticos. Al final se plantea la posibilidad de extender el dominio de aplicación a otras áreas.

Palabras clave: Algoritmos Genéticos, Asignación de Recursos, Operadores Genéticos, Optimización de Múltiples Objetivos.

1. Introducción

La asignación de recursos limitados, sujetos a restricciones de distinto tipo, ha sido un problema de relevancia en áreas como la investigación operativa, la administración de las organizaciones e incluso la inteligencia artificial. Por lo general, la solución de este tipo de problemas trae aparejada una serie de condiciones de eficiencia, tiempo y oportunidad que deben ser tenidas en cuenta, más allá de la correctitud en la asignación.

Es deseable considerar todas las variables al mismo tiempo en un problema de asignación, ponderándolas según su importancia, y buscando obtener el mayor beneficio en la resolución.

Este problema puede ser abstraído de la siguiente manera: supongamos que tenemos n recursos que son necesarios asignar a $[1, \dots, i, \dots, n]$ demandantes del recurso y que las características de los recursos son parecidas en sustancia pero no en características cuantificables. La cantidad posible de combinaciones será de:

$$\sum_{i=1}^n \left[\frac{n!}{(n-i)!} \right]^2 \cdot \frac{1}{i!}$$

A medida que n aumenta, la explosión combinatoria se vuelve inmanejable, si se quieren revisar todos los casos. Es entonces cuando podemos pensar en esas combinaciones como un espacio de estados donde cada estado es una configuración posible de asignación. De esta manera, y como revisar un espacio de combinaciones muy grande es dificultoso, conviene dirigir o conducir la búsqueda de la solución que permita resolver el problema de forma eficiente.

Podemos pensar en varios métodos de resolución como ser la búsqueda nodo por nodo de forma sistemática [1][6][7][8], que en algún momento debieran encontrar la solución, pero nada puede garantizar que no se deba recorrer todo el espacio de estados para lograrlo. Si bien se cuenta con métodos de ramificación y acotación (que se utilizan en problemas de programación lineal entera –

mixta), los valores en las restricciones varían con el tiempo, haciendo que se deba calcular las configuraciones constantemente.

Otra alternativa es utilizar métodos de búsqueda heurística[1][6][7][8], donde se sacrifica la completitud de la exploración por la eficiencia en el tiempo de respuesta. Estos métodos necesitan contar con información de qué tan lejos estamos de una posible solución para poder conducir la búsqueda y, por lo general, la determinación de una buena función heurística de que contemple varias variables no es simple de conseguir.

También se pueden considerar los modelos de redes neuronales[1][6][7][8][4] para clasificar los problemas y entrenar a la red con una serie de soluciones bien conocidas que permita que, ante una situación desconocida, la red adapte sus pesos a lo más parecido. Obviamente al encontrar nuevas alternativas deberíamos volver a entrenar a la red.

La propuesta que se hace en este proyecto es utilizar algoritmos genéticos para la resolución de problemas de asignación entera – mixta, tomando como dominio de problema la asignación dinámica de aulas a cursos que se dictan en la Facultad Regional Resistencia de la Universidad Tecnológica Nacional.

2. Descripción del problema

El problema a tratar es la asignación dinámica de aulas a distintos cursos de las carreras dictadas en la Facultad Regional Resistencia, de la Universidad Tecnológica Nacional. Las características de este problema son:

- Las configuraciones posibles de aulas y cursos son variables en cantidad de asignaciones, es decir, puede ser que no se utilicen todas las aulas durante varios intervalos de tiempo durante la jornada, y en otros no haya aulas libres.
- Las aulas tienen distintas capacidades.
- Las aulas están acondicionadas de forma distinta.
- Los cursos no tienen la misma cantidad de alumnos, a esto se le suma que a medida que transcurre el año el número de alumnos por curso tiende a disminuir.
- La inasistencia de los docentes genera liberación de aulas durante ciertos módulos y requieren reprogramación de la asignación.
- Existen eventos durante el año que requieren el uso de aulas, generalmente estos eventos se programan con un mes de anterioridad.
- Se requiere que el movimiento de alumnos con el cambio de hora sea mínimo.

Los objetivos de la asignación son maximizar el uso de los recursos (incluyendo equipos multimediales para el dictado de clases y material didáctico), minimizar el desperdicio del espacio físico y lograr la asignación completa de las aulas mejor acondicionadas, generando el mínimo movimiento de alumnos en los cambios de módulo y reasignando las aulas en caso de inasistencias de docentes o alguna circunstancia no prevista.

La Facultad Regional Resistencia cuenta con 19 aulas que pueden ser asignadas a 195 cursos aproximadamente (esto es contando con materias del primer y segundo cuatrimestre, anuales y optativas) sin tener en cuenta que existe más de una división en algunos cursos. Como el número máximo de cursos simultáneos que pueden funcionar en la facultad es de 19 (no se cuentan los laboratorios, ya que estos tienen asignación fija), existen tres turnos para el dictado de materias, siendo el turno de la noche el más conflictivo. Como máximo se pueden asignar 19 cursos en forma simultánea lo que representa una cantidad total de $1,21 \cdot 10^{17}$ combinaciones (esto es si se cubrieran todas las aulas), pero esta cantidad aumenta si se producen cambios en la asignación y se desocupan aulas. El número estimado de combinaciones, utilizando la expresión indicada en la sección anterior, teniendo en cuenta todas las variaciones en la ocupación pasa a ser de $6,99 \cdot 10^{19}$ aproximadamente.

Este espacio de búsqueda es tan grande que genera la necesidad de asignar a todos los cursos de un mismo nivel en las mismas aulas (así también se evita el excesivo movimiento de alumnos en los

cambios de módulo). Esta tarea se realiza manualmente, obviamente generando que en algunos momentos algunas aulas estén colmadas de alumnos, mientras otras con mayor capacidad estén desocupadas o con cantidad de alumnos inferior a su capacidad. Las etapas del desarrollo del proyecto se muestran en la Figura 1.

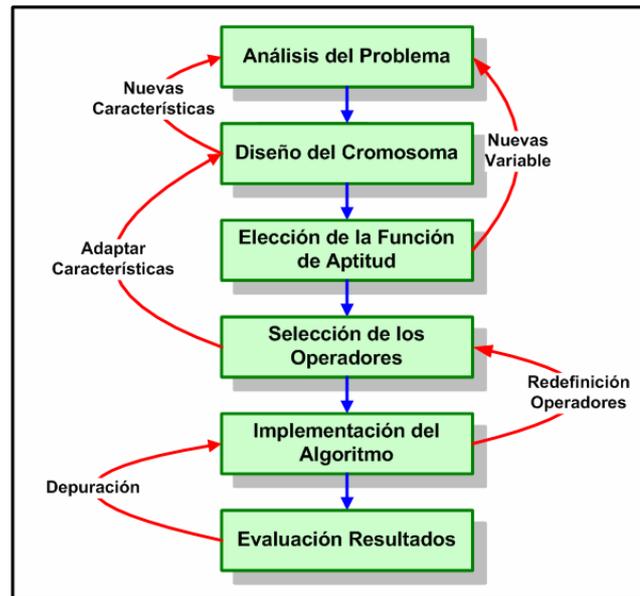


Figura 1. Etapas del desarrollo del proyecto.

3. Descripción de los AG

Los algoritmos genéticos son algoritmos probabilísticos caracterizados por el hecho de que un número N de posibles soluciones (llamadas individuos) registran el espacio de soluciones en un problema de optimización[2][5][9]. Los algoritmos genéticos crean poblaciones iniciales de soluciones que tienen distinto grado de eficiencia que, una vez medida, sirve para determinar si algún individuo es solución, o si no lo es, qué transformación sufrirá para pasar a la siguiente generación[2][5][9]. Los algoritmos genéticos simulan la evolución natural de la especie, donde los individuos más aptos sobreviven para ser utilizados en una población siguiente, o son cruzados con otros individuos o alguna de sus características sufre una mutación. La poblaciones se generan en base a los operadores de selección, cruza y mutación, y cada uno de ellos puede ser implementado con diversas técnicas. Una vez generada la siguiente población de individuos se mide la eficiencia de cada uno, si la solución es adecuada, el proceso termina, sino los operadores generan una nueva población y así hasta obtener una solución adecuada.

Los métodos de selección[2][3][5][9] tienen como objetivo elegir los individuos más aptos de una población para pasarlos directamente a la siguiente. Estos métodos pueden ser proporcionales, o bien, basados en algún tipo de ordenamiento según su medida de aptitud. Entre los métodos más usados se encuentran:

- Selección por Torneo.
- Selección por Ruleta.
- Selección con Control Sobre el Número Esperado.
- Selección Elitista.
- Selección por Ranking.

Una vez que se han elegido los individuos más aptos para la siguiente generación, se aplica el operador de cruza[3][5][9], que tiene como objetivo combinar a los individuos. Esto permite lograr una diversidad en el espacio de estados, evitando que los más aptos dominen rápidamente las generaciones siguientes.

Entre los métodos de cruce, más utilizados se encuentran:

- Cruza Simple.
- Cruza Multipunto.
- Cruza Binomial.

El último operador que se aplica a la población generada es el de mutación[3][5][9]. Este operador tiene el objetivo de alterar el valor de alguna característica de algún individuo, también para contribuir a la diversificación de la población. Este operador debe aplicarse con una probabilidad muy baja. Los métodos de mutación más difundidos son:

- Mutación Simple.
- Mutación Adaptativa por Convergencia.
- Mutación Adaptativa por Temperatura (Ascendente y Descendente).

El proceso completo de aplicación de los operadores se muestra en la Figura 2.

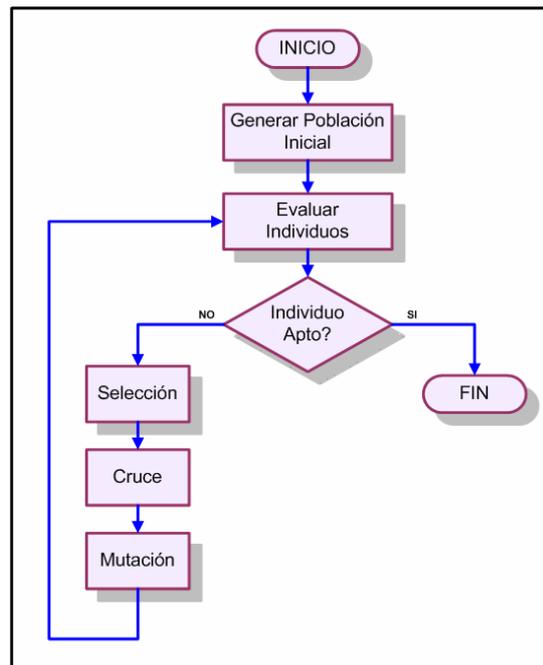


Figura 2. Funcionamiento de un AG.

La representación de las soluciones es un problema no trivial y es clave en el desarrollo del proyecto. La idea fundamental es representar en una misma estructura todas las características del problema y asignar, mediante una función de evaluación, un solo valor que refleje la aptitud de cada instancia de dicha estructura. A lo mencionado anteriormente se suman que la determinación de la función de aptitud tampoco es una tarea sencilla, como a priori no se conoce una función “buena” puede ser que existan problemas de múltiples óptimos locales y de aislamiento del óptimo global.

4. Extensión del dominio de problema

Con el desarrollo de este proyecto se pretende obtener el conocimiento necesario para poder extender el dominio de aplicación hacia otras áreas, como ser la industria (asignación de recursos en líneas de producción), comercio (logística de reposición de productos), salud (asignación de recursos hospitalarios), etc. Básicamente esto involucra conseguir una buena representación del conocimiento del dominio de aplicación, asignar la importancia correcta a cada variable implicada dentro del problema a resolver y establecer las relaciones entre esas variables. Por ello, uno de los principales objetivos es obtener un modelo abstracto que se adapte a distintos escenarios y que maneje valores que no sean solamente binarios. Luego, y mediante la prueba de las variantes

existentes para operadores genéticos, poder elaborar un esquema eficiente de manejo de las estructuras, estableciendo ventajas y desventajas de la aplicación de cada operador.

Existen además varias características que hacen a los algoritmos genéticos aptos para poder realizar la extensión de la aplicación a otros dominios, algunas de ellas son:

- Se puede trabajar con varios objetivos simultáneos.
- Pueden ser aplicados en dominios dinámicos.
- Se puede trabajar con variables heterogéneas.
- Las soluciones pueden ser buenas y terminar el proceso con márgenes de error preestablecidos.
- Son un buen método de resolución de problemas no lineales y que involucran muchas variables.
- Utilizan múltiples restricciones.
- Proveen un alto grado de posibilidad para encontrar óptimos globales.

A todas estas características se suma que la aplicación de los operadores genéticos es independiente del dominio de problema, ya que una vez obtenido el esquema (o la plantilla) que describe la estructura de las soluciones, el proceso de recombinación para la obtención de nuevas poblaciones es repetitivo.

5. Bibliografía

[1]BIGUS, J. P. BIGUS, J. *Constructing Intelligent Agents with JAVA*. Wiley.1997.

[2]BLICKLE TOBIAS AND THIELE LOTHAR. *A comparison of selection schemes used in genetic algorithms*. Technical Report 11, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), 1995.

[3]GOLDBERG, DAVID E. *Genetic Algorithms in Search, Optimization and Machine Learning* Addison-Wesley Pub. Co. 1989.

[4]HILERA, J. R. MARTINEZ, V. J. *Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones*. Alfaomega. 2000.

[5]KOZA, JOHN R. *Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems*. Stanford University Computer Science Department technical report. 1990.

[6]NILSSON, N. *Inteligencia Artificial, Una nueva síntesis*. McGraw Hill. 2000.

[7]RICH, E. KNIGHT K. *Inteligencia Artificial*. Mc Graw Hill. 1996.

[8]RUSSELL, S. NORVIG, P. *Inteligencia Artificial, Un Enfoque Moderno*. Prentice Hall.1996.

[9]SPEARS, W.M., DE JONG, K.A., BACK, T., FOGEL, D.B., DE GARIS, H. *An Overview of Evolutionary Computation*. Proc. European Conference on Machine Learning. 1993.