Aplicación de Algoritmos Genéticos en la Planeación y Administración de Recursos en Entidades Académicas

AUTORES: Katherine Viteri[1], Christian Salazar[2], Carlos Paredes[3], Fabricio Echeverria[4]

Resumen

Los algoritmos genéticos son parte de la computación evolutiva, un área creciente de la inteligencia artificial que esta basada en la evolución natural biológica, y pueden adaptarse a la resolución de múltiples problemas. Como es de esperarse, los algoritmos genéticos están basados en la teoría de la evolución de Darwin, y son apropiados para resolver problemas donde el dominio de la solución pueda resultar demasiado extenso. Dada la dificultad de encontrar una solución óptima en la planificación de horarios de entidades académicas que cumpla con los requerimientos iniciales como el acato de la disponibilidad de horarios por profesor, aulas y la asignación de recursos utilizando un procedimiento manual, se plantea un sistema lo suficientemente flexible, capaz de generar un conjunto de posibles soluciones clasificadas por un nivel de aptitud que será determinado por el cumplimiento de los parámetros iniciales para dar paso a la obtención de una solución optima.

Palabras Claves: *fitness, mutación, elitismo, cruzamiento.*

Abstract

Genetic algorithms are part of the evolutionary computing, a growing area of the artificial intelligence based on the biological natural evolution, that can be adapt to resolution of multiple problems. As it is of being expected, genetic algorithms are based on the evolution theory of Darwin, and are appropriate to solve problems where the solution domain is too extensive. Given the difficulty in finding the best solution in the academic schedules planning that fulfills the initial requirements as professor schedules, classrooms readiness and resources assignment in a manual procedure, thinks about a flexible system, able to generate a group of possible solutions classified by an aptitude level or fitness that will be determined by the execution of the initial parameters to open the way to obtaining the optimal solution

Key Words: *fitness, mutation, elitism, cross over*

1. Introducción

En un enfoque minucioso sobre como se lleva de forma manual los procesos de planificación de horarios en las entidades educativas siempre surge mas de un rompecabezas de difícil resolución, convirtiéndose en procesos que una vez planteados difícilmente pueden someterse a modificaciones o actualizaciones debido a que una alteración mínima podría resultar en una nueva planificación, afectar los horarios de profesores, o lo que es aun mas riesgoso caer en una mala distribución de recursos.

Son cada una de estas problemáticas las que nos llevan a buscar mecanismos automatizados y de manera mas enfocada de optimización heurística para solucionar los problemas a los que se enfrentan millares de instituciones académicas a nuestro alrededor.

La utilización de algoritmos genéticos para este tipo de problemas nos permite realizar búsquedas ciegas en diferentes entornos evitando el desgaste de los recursos. El poder de cómputo del que disfrutamos

[1] Ingeniera en Computación 2005, kviteri@yahoo.com

[2] Ingeniero en Computación 2005, csalazar@bankguay.com

[3] Ingeniero en Computación 2005, carlosb_paredeso@hotmail.com

[4] Director de Tesis, Ingeniero en Computación Escuela Superior Politécnica del Litoral, pechever@uniplex.com

en la actualidad nos admite costear un sinnúmero de iteraciones hasta llegar a una solución óptima.

Lo que es aun más interesante es la facultad que nos dan los algoritmos genéticos de detener una búsqueda en proceso, reiniciarlos, agregar recursos, horarios, profesores y nuevas planificaciones sin que esto afecte a lo ya planteado.

El grado de dispersión con el que cuentan los algoritmos genéticos nos dan pauta para iniciar el proceso con una búsqueda dispersa, recombinar y mutar las soluciones, evitar recaer en mínimos locales, usar distintas representaciones para el problema, y recurrir a métodos intrínsecos de la evolución natural explotándolos para encontrar un conjunto de soluciones que satisfagan los requerimientos de el problema planteado.

2. Desarrollo

Se plantea un sistema que sea lo suficientemente flexible, para obtener a partir de un listado de posibles soluciones, la alternativa optima de acuerdo a los requerimientos iniciales. Requerimientos como disponibilidad de horarios por profesor, y disponibilidad de horarios de aulas, material didáctico o condiciones físicas solicitadas por el profesor para dictar una materia.

La solución deberá ser expresada por un espacio de soluciones, hallado a partir de operaciones de exploración y explotación hasta encontrar los individuos más óptimos. Se consideró entonces el uso de algoritmos genéticos debido a que están diseñados para encontrar cada vez mejores cromosomas (soluciones) mediante manipulación "ciega" (cruzamiento y mutación) de sus contenidos.

Esta manipulación de su contenido, se basa en tres aspectos:

- Escoger una población inicial de soluciones de prueba elegidas en forma aleatoria
- Cada solución es replicada hacia un nuevo espacio de salida
- A cada recurso se le calcula su costo dentro del conjunto solución llamado "costo óptimo" o "fitness". (Este valor dependerá, no solo de cumplir las condiciones iniciales, sino además de la optimización de la distribución de los recursos concernientes al problema)

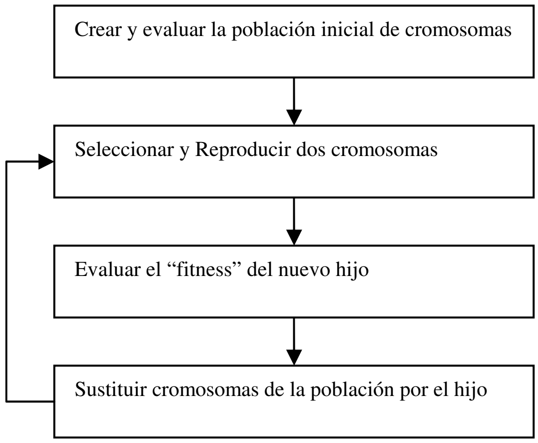

Figura 1. Esquema básico de los algoritmos genéticos

2.1. Componentes de un algoritmo genético

- Una representación de una potencial solución al problema.
- Una forma de crear una población inicial a partir de soluciones potenciales.
- Una función de evolución que juega el rol del ambiente, razón de la solución en términos de su "fitness"
- Operador genético que altera la composición de sus hijos
- Valor de varios parámetros que el algoritmo genético usa (tamaño de la población, generación, etc.)

2.2. Condiciones Iniciales

1. Cada profesor propone un horario de disponibilidad.
2. Dos materias del mismo nivel no pueden cruzarse.
3. Ciertas materias se dictan en aulas predeterminadas.
4. Cargas horarias distribuidas.

2.3. Generación Inicial

Se crean conjuntos materia-profesor ordenados por la categoría del profesor. Asociamos un horario de asignación al gen en base a la disponibilidad del conjunto materia-profesor.

Se agrega al gen el aula en base al número de aulas y se valida su disponibilidad. Aceptado el gen se detalla en la matriz del aula que el horario ha sido ocupado.

Ubicado un conjunto materia profesor en un aula, restar este horario de los demás conjuntos materia-profesor del mismo nivel y especialización para cumplir criterio 2.

Encontrar generación inicial variando horario asignado y/o aulas para conjuntos materia profesor.

2.4. Definición del String

El string que representa el problema contendrá los siguientes índices de matrices:

Cromosoma = (materia- profesor(1..k), horario asignado (1..90), aulas(1..n))

Que se cumpla hasta ubicarse todos los valores de k

Horario asignación es el horario de disponibilidad del profesor evaluado en el horario de disponibilidad del aula.

2.5. Operaciones de Cruzamiento

Con el cruzamiento horizontal (two point crossover), los dos puntos de cruzamiento se seleccionan aleatoriamente.

Otras consideraciones son las de no cruzar un hijo que tenga un fitness superior mientras no se encuentre un cromosoma con fitness superior al de este (Elitismo genètico).

Crossover point 1

Crossover point 2

Materia-Profesor	Horario-Asignado	Aula
1	2	2
2	9	1
3	14	3
4	15	4
5	12	2
6	20	3
7	66	1
8	65	5
9	8	6
…	..	
k	90	5

Figura 2. Representación de un Cromosoma

2.6. Proceso de mutación

Cambiar el valor de horario de asignación para un conjunto profesor-materia

Cambiar el valor de aula para un gen determinado con énfasis en los horarios repetidos para un aula

2.7. Calculo del fitness

Tendrá mejor fitness el conjunto solución que tiene menor cantidad de penalidades, que son valores que dependen del grado de violación para cada criterio.

3. Cada profesor propone un horario de disponibilidad
4. Dos materias del mismo nivel no pueden cruzarse
5. Ciertas materias se dictan en aulas predeterminadas
6. Cargas horarias distribuidas

2.8. Resultados Obtenidos

- Horarios por aula
- Horarios por profesor
- Listados de aulas, profesores y materias
- Penalizaciones (violación de condiciones iniciales)
- Reporte de Conflictos

3. Conclusiones

El hecho de pensar que los algoritmos computacionales puedan tener una base en la evolución de los organismos puede ser sorprendente, sin embargo el hecho de que los AG puedan ser aplicados a muchas áreas donde los conocimientos computacionales actuales como las búsquedas randómicas o graduales no han tenido resultados óptimos nos conduce a concluir que los AG`s constituyen realmente una poderosa herramienta de búsqueda heurística con la habilidad de explotar y aprender de sus dominios.

Lo AG'S son relativamente fáciles de entender e implementar, y su principal ventaja y desventaja a la vez es su robustez, si se posee una mala implementación, el AG seguirá corriendo y tarde o temprano encontrará la solución del problema o encontrará un optimo local.

El algoritmo genético obtenido puede ganar velocidad con más procesadores, puesto que todo el trabajo está equitativamente repartido en sus operaciones de cruzamiento, mutación y evaluación de fitness o bien sea en una implementación por medio de hilos.

En las pruebas realizadas se practico un aumento del ritmo de mutación, los resultados obtenidos muestran que es preferible evitar en el problema la pérdida de diversidad genética en una población de cromosomas, o bien introducir mecanismos en los cuales el fitness de un individuo se divide por el número de individuos similares a él.

Por ultimo dejemos que la naturaleza actúe de forma distribuida, por tanto, se debe de minimizar la necesidad de operadores que "vean" a la población.

Ello permite, además, una fácil paralelización del algoritmo genético. Por ejemplo, en vez de comparar el fitness de un individuo con todos los demás, se puede comparar sólo con los vecinos, es decir, aquellos que estén, de alguna forma, situados cerca de él.

4. Referencias

[1] Fogel David B., *Evolutionary Computation*. IEEE Press. Estados Unidos. 1995

[2] Goldberg David E., *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc. Estados Unidos. 1989.

[3] G.Winter, J.Periaux & M.Galan *Genetic Algorithms in Engineering and Computer Science*, published JOHN WILEY & SON Ltd., 1995.

[4] Ww. Fecha de la última actualización. Disponible en http://www.orcero.org/irbis/disertacion/node209.html

[5] Ww. Fecha de la última actualización. Disponible en http://www.redcientifica.com/doc/doc199904260011.html

[6] Ww. Fecha de la última actualización. Disponible en http://the-geek.org/docs/algen/

[7] Adenso Diaz, Fred Glover, JL Gonzales, Manuel Laguna, Pablo Moscazo y Fan Tseng. *Optimización Heurística y Redes Neuronales*. 1996. Capitulo 3