

Optimización del Espacio Aulico Mediante Algoritmos Genéticos

Sebastián Hernandez, Paula A. Millado, M. Laura Ivanissevich

Universidad Nacional de la Patagonia Austral

Lisandro de la Torre 1070 - Te: 02966-442321

{ shernandez; pmillado; mivanissevich}@uarg.unpa.edu.ar

Claudio A. Delrieux

Universidad Nacional del Sur

claudio@acm.org

Resumen En la Unidad Académica Río Gallegos de la Universidad Nacional de la Patagonia Austral (UNPA UARG), se realiza la distribución y asignación de aulas a una cátedra de forma manual, se conforma así la grilla de horarios de la unidad. Son conocidas las frecuentes dificultades con las que se encuentra el personal a cargo al momento de realizar la reasignación de espacios ante algún evento, como resultado de la premura se desaprovecha el recurso edilicio. Nos hemos propuesto diseñar un modelo efectivo y flexible que realice esta tarea para minimizar la diferencia entre capacidad del aula y alumnos anotados por asignatura. Dicho modelo adopta las estrategias de un algoritmo evolutivo, el cual nos abre un gran abanico de opciones al momento de su implementación dada la naturaleza del problema y la potencia de las herramientas utilizadas para la búsqueda de la solución.

Palabras Claves: Distribución de Aulas - Estrategias Evolutivas - Algoritmos Genéticos.

1. Introducción

La asignación de la carga horaria de cada cátedra a un aula (o espacio físico), es una tarea que se realiza manualmente en la UNPA UARG. Las dificultades y las malas soluciones no se hacen esperar al momento de reasignar dichos espacios por diversos motivos. Ni en las asignaciones originales ni en las reasignaciones

se tienen en cuenta el aprovechamiento máximo de un espacio físico con desperdicio de este recurso. Esto sin mencionar los conflictos de asignaciones que sólo son detectados cuando llega el reclamo de los docentes al momento de descubrir la falla. En este trabajo nos concentramos en lograr una distribución factible de aplicación, es decir, una grilla utilizable que evite el problema de superposición de docentes y alumnos según el año de la carrera. En comparación con otros trabajos realizados [4, 5], en nuestro caso la tarea se ve simplificada por el hecho de disponer de todos los espacios físicos en un mismo campus universitario, y así se evita el problema de la movilidad de los alumnos. Además de considerar que todas las aulas disponen de las condiciones ambientales óptimas para su funcionamiento (calefacción, en nuestro caso).

Dentro de la amplia calificación de Algoritmos Evolutivos (AE), se decidió plantear el problema con Algoritmos Genéticos (AG), sin descartar otros modelos para futuros enfoques, y teniendo en cuenta que un modelos similar fue estudiado en [1].

2. Algoritmos Genéticos

Un AG consta de una representación esquemática a la que aplica una técnica de búsqueda (enfocada a problemas de optimización) basada en las teorías evolutivas Neo-

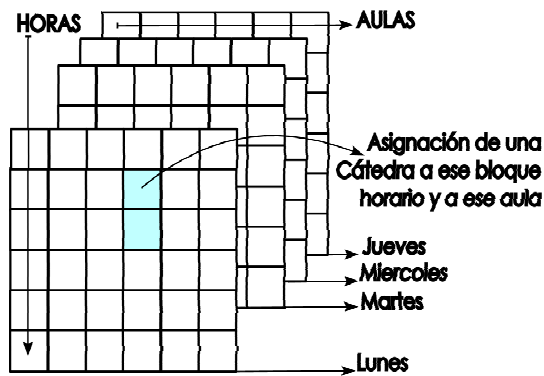


Figura 1: Representación del individuo, mostrando una matriz de horas y aulas por día

darwinianas [3]. Es así que consta de procesos de selección de individuos más aptos de una población, que sobreviven al adaptarse más eficientemente a las exigencias del entorno. Este proceso se controla al actuar sobre los genes de un individuo en los que está codificada cada una de sus características. Un AG difiere de los algoritmos tradicionales en los siguientes aspectos:

- Trabaja con la codificación de un conjunto de parámetros, no con los parámetros entre sí.
- Evalúa un grupo de soluciones en lugar de un solución por vez.
- Usa reglas de transiciones probabilísticas en lugar de reglas determinísticas.
- Evalúa las posibles soluciones sin aplicar ningún proceso de inferencia, salvando obstáculos vinculados a la definición de la función de supervivencia.

Un AG codifica las características del problema, al representar los genes de cada individuo (o cromosoma) de la población de alguna manera. La forma tradicional de un AG (propuesta por John Holland) [6] representa por ceros y unos los genes de cada cromosoma, aunque existen otros tipos de representaciones. La elección adecuada de dicha representación juega un papel importante en la eficacia del AG. La función de *ajuste* o *fitness* representa el ecosistema donde el individuo vivirá; el que

mejor se ajuste sobrevive. Matemáticamente es la función-parámetro que evalúa una solución propuesta por el algoritmo.

La primer etapa de un AG es la generación de la población inicial, proceso conocido como *birthpop*, es realizado generalmente de forma aleatoria.

Para la generar descendencia es necesario aplicar *cruzamientos* o *crossover* [6] y/o *mutaciones* [3] tanto al nivel de gen o a nivel de individuo, y para lograr la supervivencia del mejor adaptado se aplica un proceso de *selección* [3] que permite su elección para su reproducción.

3. Nuestra Representación

En los AG la elección de la representación del individuo es de suma importancia para el algoritmo debido a que ésta es la clave para una performance exitosa. Para nuestro caso necesitamos representar la grilla semanal de asignaciones de las aulas a las diferentes actividades que se desarrollan, para lo cual optamos por caracterizar a nuestro individuo como una serie de matrices que representan la semana laboral, donde cada matriz conserva la información por día, compuesta por filas que representan a las aulas y por columnas de horas, de este modo cada celda determina la asignación de un aula a una asignatura por una unidad de tiempo (generalmente una hora) como se muestra en la figura 1, debido a que las asignaturas pueden constar de 4, 6, 8, 9 o 10 horas a semanales en módulos de 2hs, 3hs o 4hs, así podemos adaptar nuestra grilla de horas a cualquier configuración de módulos. Este modelo evita la superposición de asignaciones.

El código de una materia es un número decimal:

- a) la parte entera guarda la información de su pertenencia a las carreras en su descomposición en valores primos.
- b) los dígitos de la parte decimal informan 1: sobre el año en el plan de estudios, 2: sobre la cantidad de alumnos inscriptos, 3: sobre

el id del docente y en 4: sobre la modulación horaria.

Por ejemplo habra una materia 6,1 17 22 3.

En cuanto a la función de ajuste se estableció como premisa la minimizar el espacio áulico desperdiciado, el monitoreo de la sumatoria de las diferencias entre el espacio físico del aula y el número de alumnos inscriptos conduce la sobrevivencia de los individuos. En síntesis:

$$Dif_p = CapAula_i - AlCatedra_i$$

$$\Delta_p = \Phi Dif_p$$

$$F_p = \sum_{i=1}^{pob} (\Delta_p),$$

El valor Φ es un coeficiente de penalización [9] a la mala calidad del individuo. Di penaliza al individuo cuando:

- el aula no tiene la capacidad suficiente,
- se excede la disponibilidad horaria diaria,
- el docente tiene superposición horaria,
- el alumnado regular tiene superposición horaria.

4. Implementación

Hemos realizado la implementación en Mat-Lab, puesto que este software posee una gran flexibilidad para trabajar con datos numéricos, siendo óptimo para generar códigos iterativos como algoritmos genéticos. La implementación es de tipo modular, el módulo principal que requiere como datos de entrada la cantidad de aulas chicas, medianas y grandes; la cantidad de bloques horarios utilizables; la cantidad de días de la semana en que utilizará el edificio, un vector que representa las materias a ubicar; el tamaño de la población con la que operará el AG, la cantidad de iteraciones a realizar, la probabilidad de cruzamiento y la probabilidad de mutación (ver figura 2). La penalización se realiza en pasos aditivos al evaluar problemas acumulativos

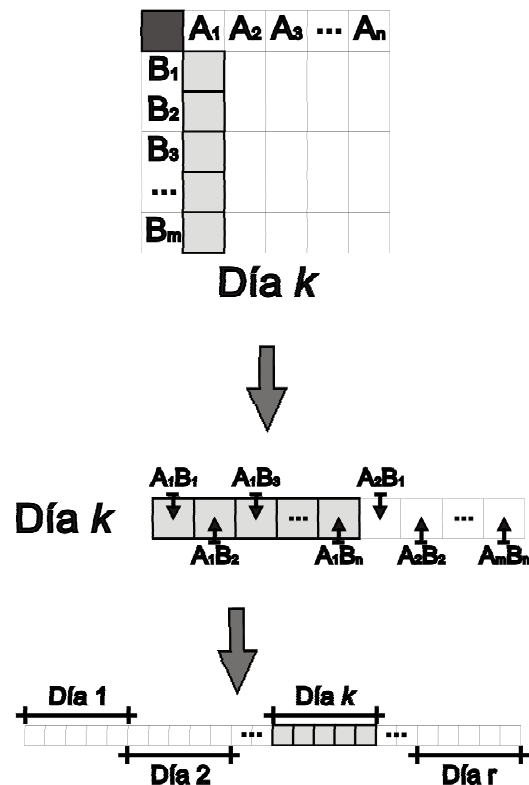


Figura 2:

5. Conclusión

Se ajustará el algoritmo a medida que registremos diferentes resultados y posteriormente se afinará el modelo de la situación real en estudio.

Referencias

- [1] E.K. Burke and J. P. Newall. A multistage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, vol. 3(nro 1):63–74, April 1999.
- [2] Eric Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns. Elements of REusable Object-Oriented Software*. Addison-Wesley, 1995.
- [3] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, The University of Alabama, 1998.

- [4] Marcelo J. Karanik. Asignación dinámica de aulas utilizando algoritmos genéticos. In *Anales del VII Workshop de Investigadores en Ciencias de la Computación: WICC 2005*, pages 56–60. Red de Universidades Nacionales con Carreras en Informática (Red UNCI), Universidad Nacional de Río Cuarto, 2005.
- [5] Marcelo J Karanik and José S. Pérez. Algoritmos genéticos para la optimización de asignación de espacios Áulicos. In *Anales del Octavo Workshop de Investigadores en Ciencias de la Computación (WICC 2006)*. Red de Universidades Nacionales con Carreras en Informática (Red UNCI), junio 2006. Universidad de Morón Provincia de Buenos Aires Argentina.
- [6] Zbigniew Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, third, revised and extended edition, 1999.
- [7] Herbert Schildt. *Borland C++ Manual de Referencia*. Serie McGraw-Hill de Informática. Osborne McGraw-Hill, 1997.
- [8] Graham Seed. *An Introduction to Object-Oriented Programming in C++*. Springer, 1996.
- [9] Alice E. Smith and David M. Tate. Genetic optimization using a penalty function. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 499 – 505. Morgan Kaufmann Publishers Inc., 1993.