Un nuevo enfoque para asignación óptima de múltiples recursos

Facundo E. Cancelo, Pablo N. Cababie, Gabriel Barrera, Daniela López De Luise
Unversidad de Palermo, *ITLab, AIGROUP,*
Ciudad Autónoma de Buenos Aires, Argentina, C1188AAB
aigroup@palermo.edu

Abstract

The problem in the optimal room's assignment has been developed and researched with lot of different techniques, tools and technologies. This job shows the design of a solution based on Artificial Intelligence, specifically using Multi-objective Genetic Algorithms that optimizes the classrooms and teachers assignment simultaneously. This implementation has been built optimizing more than one objective, letting both of them competing to establish a general equilibrium and get to a feasible solution quickly and efficiently.
This development shows the parameters involved in the problem, the structure of the gene set, the architecture chosen for implementation and the definition of the fitness function. This job is based on a functional prototype. Under the terms of the particular problem has been necessary to redesign the operations of traditional genetic algorithms, to define weights for the objectives and restrictions of the case. From this analysis emerge the hypercube and distance concepts to ensure comparability, and close relationship of the solutions.
This system meets the feature known for Genetic Algorithms: solves one problem but it is especially applicable to similar problems with entirely different scenarios.
Keywords: Genetic algorithms, Epsilon MOEA, Resource management, Optimization, Spaces, classrooms, Assignment.

Resumen

El problema de la asignación óptima de aulas ha sido desarrollado e investigado con varias técnicas, herramientas y tecnologías. Este trabajo muestra el diseño de una solución basada en inteligencia artificial, más específicamente utilizando algoritmos genéticos multi-objetivo que realizan la asignación de aulas y docentes en forma conjunta. La implementación consiste en optimizar más de un objetivo, permitiendo que ambos compitan para establecer un equilibrio general y llegar a un conjunto de soluciones factibles en forma ágil, eficaz y eficiente.
A lo largo de este desarrollo, se detallan los parámetros que intervienen en el problema, la estructura definida del gen, la arquitectura elegida para la aplicación y la definición de la función de optimización. Es de destacar que este trabajo se basa en un prototipo funcional. A su vez, cabe aclarar que de acuerdo a las condiciones del problema en particular ha sido necesario rediseñar las operaciones de algoritmos genéticos tradicionales, definir ponderaciones para los objetivos y restricciones del caso. De este análisis surgen conceptos como distancia e híper-cubo para establecer la comparabilidad, relación y cercanía de las soluciones.
Éste sistema cumple con la característica conocida de los Algoritmos Genéticos: resuelve un problema en especial pero es aplicable a problemas del mismo tipo con escenarios totalmente distintos.
Palabras claves: Algoritmos Genéticos, Epsilon MOEA, Administración de Recursos, Optimización, Espacios, Aulas, Asignación.

INTRODUCCION

El presente trabajo se basa en una investigación que se lleva a cabo en el ITLab de la Universidad de Palermo. Dentro de este marco, se propone mejorar el sistema de asignación de aulas y docentes al comienzo de cada ciclo lectivo de esta institución. El propósito es implementar esta tarea en forma automatizada para que se pueda determinar la mejor distribución de las aulas optimizando el uso de los espacios y la asignación de los docentes. Actualmente, esta tarea se realiza en forma manual, estimando y estipulando los recursos como mejor parecen ajustarse según el criterio del responsable. Sin embargo, debido al volumen de datos y variables que intervienen en el proceso, la tarea es costosa, compleja y no siempre culmina en resultados satisfactorios, como por ejemplo, realizar una asignación con aulas que permanecen medianamente vacías o recintos muy pequeños para la cantidad de estudiantes. Por otro lado, puede no contemplarse el mejor uso de los horarios disponibles de cada docente, con lo cual no se convoca a los mismos en todos sus horarios disponibles.

El proceso de asignación de aulas que a simple vista parece la optimización de algunos pocos parámetros como cantidad de alumnos por clase, capacidad de las aulas y cantidad de recintos disponibles ha sido motivo de varios trabajos.[10]. El mismo puede ser comparado en forma análoga al problema de definición de rutas de vuelos comerciales para líneas aéreas.[18] En ambos casos, será necesario implementar una solución que acate las limitaciones dinámicas y múltiples considerando el conflicto de objetivos existente para permitir un modelo más cercano a la realidad.

Por último, cabe mencionar, que el modelado de las relaciones complejas entre los distintos parámetros del problema como las aulas, los docentes con sus horarios disponibles, las materias que pueden dictar y la oferta académica y otros más, plantean un tipo de problema no resuelto hasta la actualidad de manera conjunta e integral, en parte por la escasez de investigación en el área.

El resto de este trabajo se organiza como sigue: en la sección II se realiza un análisis de los productos y alternativas vigentes en el mercado; en la sección III se explica la estructura y arquitectura general que tiene Gdarim, tanto a nivel de implementación y código como a nivel de su diseño. La sección IV define las conclusiones y el trabajo propuesto para el futuro.

ANTECEDENTES

Hasta el momento, no se encuentra un producto que pueda determinar la mejor distribución de aulas en función de las características edilicias y de las franjas horarias en que estén disponibles los docentes.

Se han desarrollado diversas versiones de software que ayudan visualmente a la asignación manual de las aulas. No obstante estas versiones no utilizan inteligencia alguna y no optimizan de manera automática la distribución de los recursos [2], [4], [5]. También, se han encontrado desarrollos que recurren a métodos matemáticos, como el simplex [8].

Recientemente, se hallaron registros y documentación del inicio de una línea de investigación que implementaría inteligencia artificial [9].

En su mayoría, las soluciones existentes, ayudan a una visualización temporal y la distribución de los espacios y carecen de inteligencia alguna.

- Visual Classroom Scheduler [2]: Aplicación modelada en lenguaje Java que brinda una forma simple de distribuir y acomodar las clases en los ámbitos disponibles.
- Classroom Scheduler [5]: Permite una visualización bastante clara de la distribución de los salones de clase. El mismo es Open Source, por lo que se dispone el código fuente y puede ser modificado y adaptado según necesidades específicas de cada institución.
- phpScheduleIt [4]: Alternativa Open Source. Es un software que si bien no asigna los recursos eficientemente, sí gestiona los cambios de aula ante eventos especiales o el uso de laboratorios. Es decir, dado un esquema y cronograma, ayuda a gestionar los cambios para que la repercusión sea mínima. El mismo fue desarrollado en lenguaje PHP (PHP Hypertext Pre-processor).

- Softaula [3]: En sus 4 tipos de licencia ofrece desde la administración de la inscripción de alumnos a materias hasta la búsqueda de aulas libres para asignar nuevos alumnos a través de procesos automáticos. Si bien es un producto integral para todas las necesidades administrativas, gestión académica, estadísticas, marketing y contabilidad; no resuelve el problema de la optimización del uso de aulas y de docentes ni mejora la performance en la distribución.

Otras organizaciones han intentado mejorar su proceso de asignación manual incorporando el proceso a sus sistemas informáticos e integrándolo. Tal es el caso de la Universidad nacional del Comahue y el Municipio de Pereira con sus respectivos sistemas.

- SIU Guaraní [6]: Sistema de la Universidad nacional del Comahue que se integra a sus sistemas informáticos. Dicho sistema realiza la gestión académica de alumnos, desde su matriculación hasta su egreso, complementándose con gestión de aulas, mesas de exámenes, jurados, etc. Las inscripciones a exámenes y cursado están disponibles por internet, como así también consultas de la ficha del alumno, inscripción en materias, carreras, cursadas y exámenes, registro de equivalencias y exámenes rendidos.
- Matrícula [1]: Desarrollado por el Municipio de Pereira en Colombia, para que sus instituciones educativas puedan tener la información académica de los alumnos de manera integrada. Pero no optimiza el uso de los espacios ni utiliza inteligencia para determinar la administración de los mismos.

Sin embargo, la implementación de los sistemas solo ayudan a aligerar los trámites académicos evitando la carga burocrática, sin perder el control de los mismos y brindar a las autoridades encargadas de la conducción, herramientas de control sobre el manejo de la gestión académica.
Todas las alternativas aportan un avance considerable en el proceso de asignación de aulas, aunque, ninguna de estas consigue resolverlo por completo.

ESTRUCTURA DE GDARIM

Ante un profundo relevamiento de datos, la problemática se reduce a una asignación de recursos, con una cantidad extensa y determinada de parámetros dependientes imposibles de ser tenidos en cuenta en su totalidad por la mente humana. Por ello, este trabajo propone utilizar un algoritmo Epsilon MOEA[11][19] con el fin de optimizar ambos objetivos simultáneamente.
Cabe destacar que la tecnología mencionada permite trabajar con una gran cantidad de parámetros y deja la posibilidad de agregar nuevos objetivos a optimizar en el futuro.
El lenguaje seleccionado para desarrollo es Java, dada su portabilidad, eficiencia y su característica de Open Source

3.1 Modulos

El prototipo Fig. 1, consta de los siguientes tres módulos:

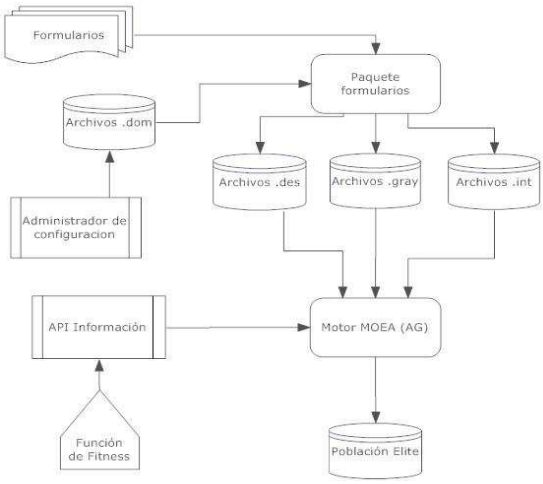

Fig. 1. Arquitectura General de Gdarim.

3.1.1 Un módulo API (Application Programming Interface) de información

Fig. 2: Contiene los valores del dominio del problema. Administra la configuración necesaria del algoritmo MOEA como ser las restricciones, evaluaciones y ponderaciones específicas de cada objetivo del problema

3.1.2 Un módulo motor AG

Implementa el core del algoritmo genético y la inteligencia de la solución. Incluye la administración de las poblaciones, el proceso de los operadores genéticos y la evaluación conjunta de los objetivos a optimizar.

3.1.3 Un módulo API de asignación

Registra las posibles asignaciones y el espacio de solución que genera el algoritmo.

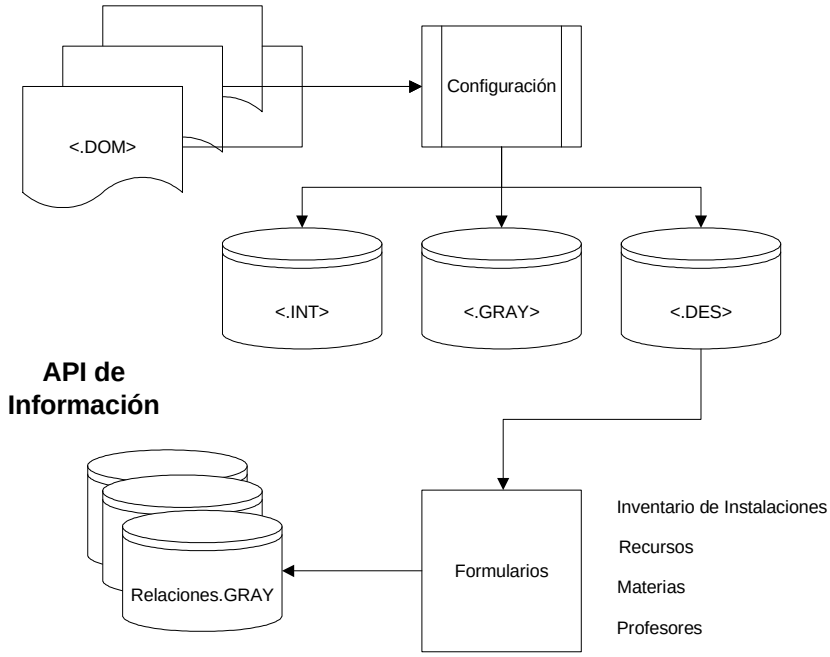

Fig. 2. Modulo de información donde se configuran las características del problema

3.2 Objetivos

El diseño comienza definiendo el tipo de problema que incluye optimización de aulas y docentes. El primer objetivo consiste en utilizar la menor cantidad de aulas sin degradar la oferta académica de materias.

El segundo, en cambio, intenta utilizar al docente dentro de su disponibilidad horaria y en la medida que sea posible, en forma atómica y consecutiva.

La optimización de un objetivo puede contraponerse con la del segundo, lo cual conlleva a la necesidad de configurar la ponderación de cada objetivo para que ambos compitan en función de su importancia y obtener un equilibrio en ambas necesidades.

3.3 Restricciones

Con el fin de acotar el espacio de soluciones, se define una serie de restricciones, entre las cuales, por ejemplo, se establece que no se puede asignar un aula en desuso, ni utilizar un aula para una materia cuando no cuente con los recursos requeridos por la misma. En cambio, para el caso de docentes, no puede haber dos asignaciones de un docente en el mismo horario, ni tampoco ser asignado a una materia para la que no está habilitado.

3.4 Proceso de carga

La información se ingresa a través de varios formularios.

Un caso es el formulario de docentes Fig. 4, en el cual se ingresan los datos básicos del empleado, las materias que puede dictar y los horarios en los que se encuentra disponible para dictar dichas materias.
En el formulario de instalaciones Fig. 5, se ingresan las aulas y otros datos de las mismas como la sede a la que pertenece, su capacidad, de que tipo es y con que recursos cuenta.
Por otro lado se cuenta con un formulario de materias Fig. 6, donde se especifica la capacidad (cupo máximo de inscriptos) de cada una, el cupo probable, su sede de preferencia y una lista de recursos requeridos para el dictado de las clases.
La información de estos formularios es almacenada en archivos que luego son utilizados por la aplicación para conocer todos los posibles valores que pueden recibir cada uno de los parámetros que conforman al gen. El mismo representa una composición que especifica qué parámetros se evalúan para la representación del problema Tabla 1. Estos últimos, son los valores de dominio del problema.
A su vez, los mismos permiten ingresar y interrelacionar la información del problema. Como resultado de los mismos, además de retroalimentarse, quedan definidas las restricciones que tendrán las siguientes poblaciones. Como por ejemplo, el cupo máximo de un aula o la materia que puede dictar un docente en particular.

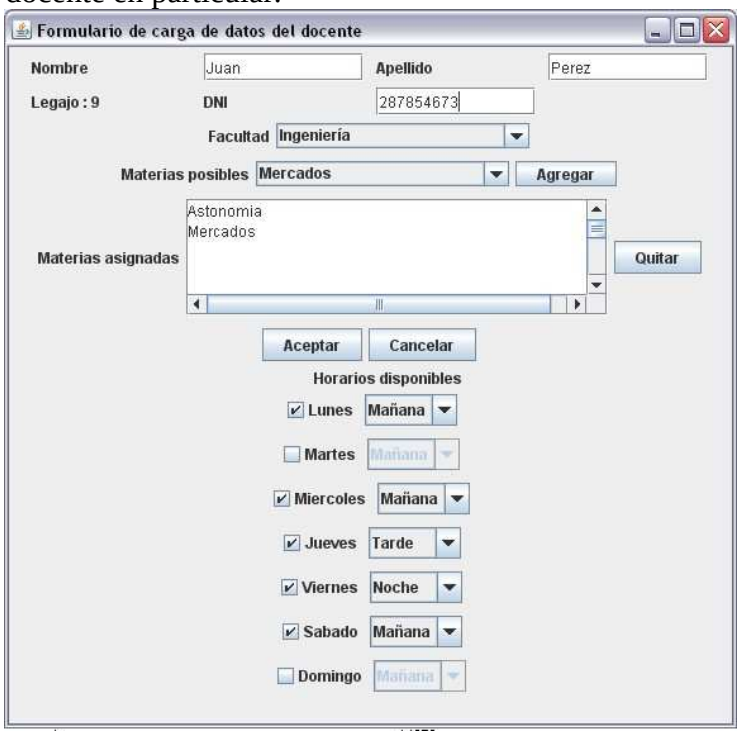

Fig. 4. Formulario de carga de docentes. Se especifican horarios disponibles como las materias que puede dictar.

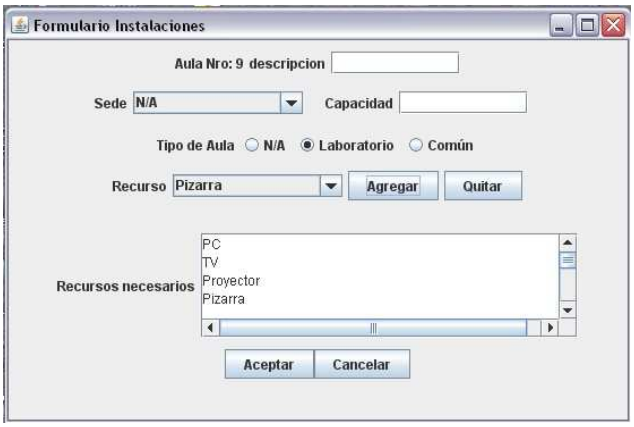

Fig. 5. Formulario ingreso de instalaciones. Aulas, Sedes, Recursos que dispone

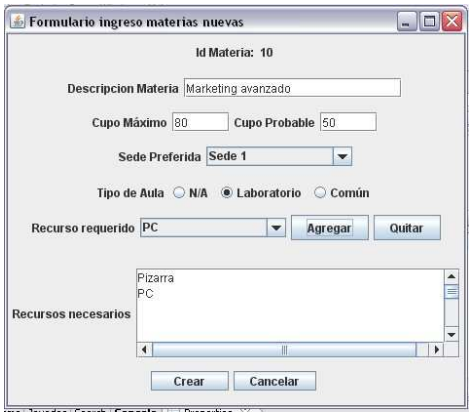

Fig. 6. Alta de nueva materia.

TABLA 1

EJEMPLO DE GEN QUE MUESTRA ALGUNOS DE LOS RECURSOS PARA UN AULA Y UN DOCENTE.

GEN
Nro. de aula
Materia
Docente
Turno
Sede
etc.

3.4 Proceso

Inicialmente, se ingresan los archivos con los valores de dominio del problema, lo cual implica la discretización de todas las posibilidades de cada parámetro incluido en la solución. Luego, se crea una población inicial en forma azarosa desde la cual se da paso a la búsqueda de uno o varios resultados óptimos.

Conceptualmente, la solución del problema será constituida por uno o varios individuos, que constituyen una composición de asignaciones de todas las aulas en todos los horarios disponibles para un día determinado

Sin embargo, el problema de crear soluciones azarosamente reside en la alta probabilidad de que estas cumplan con una o varias restricciones. Consecuentemente, la población queda invalidada por lo que nos encontramos en lo que se denomina “mortalidad prematura”, ya que por definición, una restricción invalida al individuo como solución factible.

3.4.1 Restricción penalizada

Para evitar este fenómeno, se implementa el concepto de restricciones penalizadas, con el cual se parte de la hipótesis de que una restricción inicialmente se comportará como una penalización, pero que no invalide al individuo. (El método evita la invalidación de la población inicial, pero no permite mejorarla)

3.4.2 Mutación ponderada

Por otro lado, directamente proporcional al índice de restricciones cumplidas por individuo, se aumenta el índice de mutación, para explorar en mayor medida dentro del espacio de búsqueda y en forma gradual lograr una baja en las restricciones.

A medida que las restricciones se cumplen en menor medida, y debido a la tarea exploratoria, decrece el nivel de mutación hasta alcanzar un valor estable, el cual se configura como cantidad de restricciones por individuo.

3.5 Etapa de estabilización de la población inicial

En esta fase, se busca una población estable, en función del no cumplimiento de las restricciones del problema. No es necesaria la evaluación del fitness ni tampoco la cruza de individuos, ya que se desea encontrar soluciones lo mas heterogéneas posible.

En esta etapa, el objetivo a cumplir es generar una población que no cumpla restricciones, lo que define la necesidad de hacer decrecer paulatinamente el índice de restricciones por individuo. Para esto, de selecciona solo una característica de la solución a ser mutada para los casos que una solución sea invalidada por la restricción. Es decir, se define el alelo a mutar según cada restricción. De esta manera se acelera la búsqueda de soluciones estables.

En la Fig. 7, se puede apreciar como se mantiene el índice de restricciones durante 2 generaciones para luego comenzar a caer aceleradamente hacia la convergencia de una población estable al cabo de no más de 15 generaciones.

La cantidad de restricciones por un individuo que se observa por generación esta proporcionalmente relacionada con la cantidad de restricciones definidas. Es decir que a mayor cantidad de restricciones, mayor cantidad de restricciones cumplidas en la generación inicial. Lo destacable del grafico es la trayectoria que muestra la curva que representa la incidencia de la mutación restringida (implementada) para la mejora de la población.

Así es como el sistema, se focaliza en la búsqueda del punto de partida para determinar el óptimo, con selección y mutación de individuos para bajar el índice de restricciones. La ponderación de la mutación decrece mientras lo hacen las restricciones hasta alcanzar un valor esperado.

Es preciso aclarar, que esta etapa puede demorar varias iteraciones o generaciones. Sin embargo, el costo de procesamiento no es crítico en este tipo de problema, y resulta indispensable contar con una población diversa y válida para dar inicio a la búsqueda del óptimo.

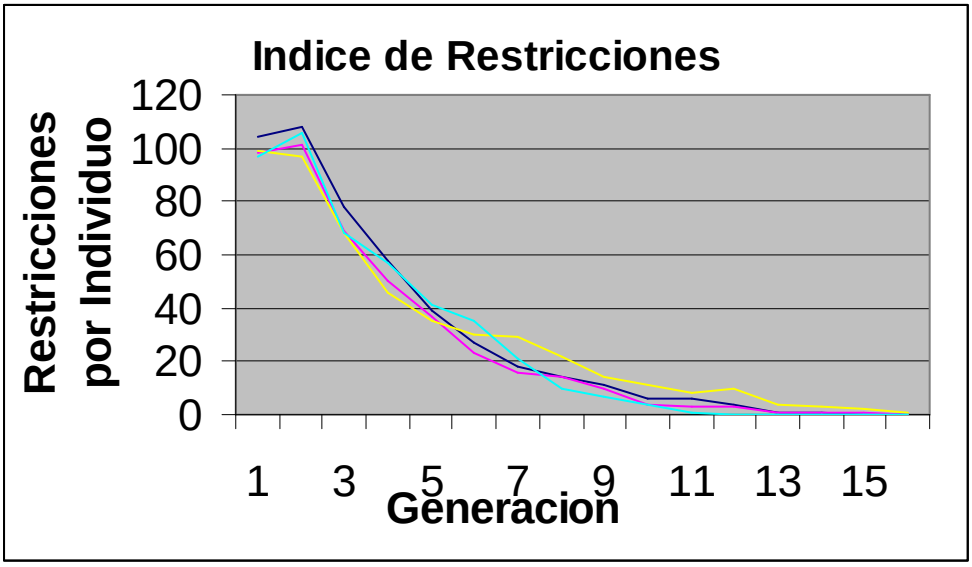

Fig. 7. Índice de restricciones por Individuo en Etapa de Estabilización

3.6 Etapa de procesamiento

La población inicial estabilizada, producto de la etapa inicial, se codifica en genomas que son reproducidos, mutados (operador genético), cruzados (operador genético), reemplazados y evaluados de acuerdo a las operaciones de AG definidas [7], y a una serie de funciones de aptitud que son las encargadas de determinar que alternativa resuelve mejor el problema.

El proyecto cuenta con archivos de configuración en formato texto que se pueden acceder en forma manual para ver la información allí ingresada. Los mismos se encuentran localizados en una carpeta de configuración denominada “cfgs”. Ahí se localizan los archivos de dominio de valores con extensión “.dom” (punto de partida del proyecto). Con el uso de los formularios se

actualizan estos archivos, al mismo tiempo que se establecen las interrelaciones entre las variables. Luego, un proceso conversor se encarga de codificar estos archivos y generarlos en codificación Gray[16] en la carpeta “gray/AG” y en codificación decimal en la carpeta “ints”.
El problema Gdarim consta de 20 variables binarias (Gen) y una cantidad de restricciones que en gran parte se extraen de la carga de los formularios y otras implícitas como ser que las materias no se superpongan que son definidas según el tipo de problema especifico.
El motor del algoritmo posee un administrador de datos que se encarga de ejecutar la lógica de acceso a los mismos. Dado que este administrador fue implementado con un Singleton Pattern [15], no existen accesos alternativos a dicha información.
El MOEA desarrollado optimiza el uso de aulas y docentes, como ambos objetivos pueden ser contrapuestos [11], [12] , [13], se aplica la función de fitness (1) para aulas y (2) para docentes con cada individuo.

$$f(x)=\sum(AV-AN) \quad (1)$$

Donde AV son las aulas no asignadas a un docente, y AN las aulas en desuso ya sean por tareas de mantenimiento u otros motivos similares.

$$f(y)=\sum(SHD-SMax)+\sum(HsD_t-HsD_u) \quad (2)$$

Donde SHD es la separación horaria del docente. SMax es la separación máxima entre horarios contiguos. HsD(t) es la disponibilidad horaria total de un docente. HsD(u) es la carga horaria utilizada para un docente.
Se evalúa el beneficio y desventaja determinado en ambas funciones [10]. Para ésta investigación, la optimalidad es minimizar la función de fitness de ambos objetivos, bajo los siguientes criterios:.
A mayor tasa de desuso de aulas, se obtiene un peor fitness de aulas.
Cuantas más horas disponibles tenga un docente sin ser convocado, se obtiene un peor fitness de docente.
Con el fin de acotar el espacio de búsqueda, el algoritmo trabaja con una o varias funciones que modelan las restricciones del problema.
En cada población, se calcula el fitness de sus individuos con respecto a cada uno de sus objetivos como se ha descripto anteriormente. Se extrae una muestra de sus individuos en función de una probabilidad obtenida por un archivo de configuración. La muestra obtenida es procesada por los operadores genéticos de cruza y mutación, de la misma manera, en función de otra probabilidad por configuración.
Para la cruza se toman dos individuos y se establece un punto de corte que determina que parámetros se toman del primer individuo y cuales del segundo, para formar dos descendientes. La cruza permite la explotación de las soluciones permitiendo la búsqueda en profundidad.
En cambio, en la mutación se parte de un único individuo del cual se muta sólo alguno de sus parámetros, transformando su valor por algún otro válido dentro del dominio del problema para el parámetro seleccionado. Paralelamente, la mutación colabora con la diversidad de soluciones, explorando el espacio de búsqueda para evitar finalizar en un mínimo local.
Finalmente, resulta de este proceso, una segunda población como descendencia de la inicial, que contendrá individuos de la cruza de dos individuos, y otros, productos de una mutación.
De ese modo, la población inicial y la descendiente deben evaluarse para establecer la dominancia entre ellas. Aquellos que por fitness dominan a otros y no son dominados podrán pasar a formar parte de la población elite. Esta población elite representa a individuos con el mejor fitness obtenido y son soluciones no dominadas, o sea, que forman parte del Pareto-Óptimo [13].
De los Individuos restantes se toman los n mejores candidatos, donde n es el tamaño de la población y con ellos se forma la nueva generación con la que iniciará un nuevo ciclo, en cada uno de los cuales, se procesan los mejores candidatos que intentan ingresar a la población elite, ya sea como nuevo integrante o reemplazando a uno o varios existentes.

De esta forma, se reitera el ciclo hasta una cantidad de veces definida en la configuración o alcanzado el grado de solución aceptable. Puede haber más de una solución resultante al problema, donde quedan a disposición del usuario para su elección.

4 CONCLUSIONES Y TRABAJO A FUTURO

La siguiente etapa de la investigación se focalizará primordialmente en la configuración general del sistema para obtener resultados con mayor eficacia.
Para ello, se requerirá definir valores ponderados de penalización para cada restricción en particular, de este modo, se logrará relativizar estas últimas entre sí.
También, se requerirá modificar la ponderación de los objetivos por configuración estática a un proceso dinámico de pesos en tiempo de ejecución, es decir el peso o relevancia de cada objetivo dentro del problema definido
Asimismo, la distancia, será un parámetro del sistema que permitirá evaluar y descartar las posibles soluciones propuestas. El mismo podrá ser determinado y configurado según se crea necesario. Por esto, se requerirá ajustar la configuración de las distancias entre las diferentes soluciones con el fin de establecer su nivel de comparación y su pertenencia al hipercubo o entorno de homogeneidad.
Acorde al crecimiento del volumen de información con el opera el sistema, y a su vez, con el fin de agilizar y simplificar el proceso, analizaremos la implementación y acceso a la información mediante una base de datos.
En última instancia, se realizarán pruebas con datos reales a fin de evaluar los resultados, refinar y mejorar las ponderaciones y configuraciones definidas, para mejorar la respuesta del sistema y su eficacia.

Referencias

[2] Sistema Matricula, Pereira Educa, Colombia, http://www.pereiraeduca.gov.co/modules.php?name=Matricula&file=matricula_valoraciones

[3] Visual Classroom Scheduler, http://www.vss.com.au/index.asp

[4] Softaula, http://www.softaula.com/es/prod/prod_comparativa.asp

[5] PHPScheduleIt, software opensource, http://sourceforge.net/projects/phpscheduleit/

[6] Classroom Scheduler, software Open Source, http://sourceforge.net/projects/cr-scheduler/

[7] "SIU Guaraní", Universidad Nacional de Comahue, Tecnologías de la Información, http://www.uncoma.edu.ar/

[8] "Introducción a la computación evolutiva", Anselmo Perez Serrada, 1996

[9] "Modelos de despacho eléctrico económico – ambiental", Pablo Pizarro. Universidad de Mendoza, 2006.

[10] "Modelado de la Distribución de Espacios Físicos mediante Algoritmos Evolutivos", C.A. Delrieux, IX WICC, 2007.

[11] "Optimización Multiobjetivos del Proceso de Torneado", Ing. R. Quiza Sardiñas, Matanzas, 2004.

[12] "Twenty Years of Evolutionary Multi-Objetive Optimization: A Historical View of the Field", Carlos A. Coello Coello. Mexico, D.F., Nov. 11, 2005.

[13] "Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map", S. Obayashi et al., 980-8577. Japan.

[14] "Introducción a la Optimización Evolutiva Multiobjetivo", Carlos A. Coello Coello, Sept. 2002, http://neo.lcc.uma.es/

[15] Anselmo Perez Serrada, "Una Introducción a la Computación Evolutiva", 1996, p. 17.

[16] "Analysis Patterns: Reusable Object Models", Martin Fowler, Addison-Wesley Professional, 1996.

[17] http://en.wikipedia.org/wiki/Gray_code

[18] "Method of Inequality-Based Multiobjective Genetic Algorithm for Domestic Daily Aircraft Routing", Ta-Yuan Chou; Tung-Kuan Liu; Chung-Nan Lee; Chi-Ruey Jeng, Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on Volume 38, Issue 2, March 2008

[19] “Multiobjective genetic algorithms applied to solve optimization problems”, Dias, A.H.F.; de Vasconcelos, J.A., Magnetics, IEEE Transactions on Volume 38, Issue 2, Mar 2002 Page(s):1133 - 1136