

Genetic Algorithm for Automatic Group Formation Considering Student's Learning Styles

Algoritmo Genético para la Formación Automática de Grupos en Base al Estilo de Aprendizaje

Germán Lescano
CONICET
Univ. Nacional de Santiago del Estero
Avda. Belgrano (S) 1912
(4200) Santiago del Estero, Argentina
+54-0385-4509560
gelescano@unse.edu.ar

Rosanna Costaguta
Univ. Nacional de Santiago del Estero
Avda. Belgrano (S) 1912
(4200) Santiago del Estero, Argentina
+54-0385-4509560
rosanna@unse.edu.ar

Analia Amandi
CONICET
Univ. Nac. del Centro Pcia. de Bs. As.
Campus Paraje Arroyo Seco
(B7001BBO) Tandil, Bs. As., Argentina
+54-02293-439682
amandi@exa.unicen.edu.ar

ABSTRACT

The group formation is an important topic in Computer Supported Collaborative Learning (CSCL) because that has implications in the group performance. In this paper, we propose a genetic algorithm for automatic generation of groups considering learning styles of your members. The group formation with genetic algorithm is a permutative problem, for this reason, genetic operators were designed. We use historical data about performance of groups and we create association rules which are used in the fitness function. The algorithm proposed was analyzed with different size of groups given for the teacher. Through the experimentation we can see what kind of configuration tends to be more appropriate.

RESUMEN

En Aprendizaje Colaborativo Soportado por Computadora (ACSC) la formación de los grupos de estudiantes es un aspecto de particular importancia ya que impacta en el rendimiento académico del mismo. En este artículo se presenta un algoritmo genético que conforma automáticamente los grupos considerando el estilo de aprendizaje de cada uno de sus miembros. La formación de grupos mediante un algoritmo genético es un problema de permutaciones, razón por la cual se diseñan los operadores genéticos acordes para solucionarlo. Para definir la función de evaluación del algoritmo se usaron reglas de asociación descubiertas sobre datos históricos vinculados con el rendimiento de grupos colaborativos reales. El algoritmo creado fue sometido a pruebas experimentales con grupos de diferentes tamaños propuestos por el profesor. La experimentación realizada permitió detectar los parámetros bajo los cuales el algoritmo genético brinda las mejores configuraciones grupales.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EATIS'16, April 27 - 29 2016, Cartagena, Colombia.

Copyright 2016 IEEE 978-1-5090-2435-3 \$31.00

Categories and Subject Descriptors

J.1. [Computer Applications]: Administrative Data Processing - Education

K.3.1 [Computing Milieux]: Computers and Education - Computer Uses in Education - Collaborative Learning, Distance Learning

General Terms

Algorithms, Design, Experimentation

Keywords

Computer Supported Collaborative Learning; Genetic Algorithm; Groups Formation; Learning Styles.

Palabras clave

Aprendizaje colaborativo soportado por computadora; Agente asistente; Dinámica de grupo; Roles de equipo.

1. INTRODUCCIÓN

Los estudiantes manifiestan un determinado modo de actuar cuando encaran las diferentes actividades que llevan a cabo para estudiar. Este comportamiento describe una inclinación hacia un estilo de aprendizaje específico. En un ambiente de aprendizaje colaborativo los estudiantes trabajan en grupos pequeños para ayudarse a aprender unos a otros [1]. Este paradigma es la filosofía de base del Aprendizaje Colaborativo Soportado por Computadora (ACSC), también conocido como CSCL (Computer Supported Collaborative Learning). El ACSC se vincula con situaciones en las que el aprendizaje se organiza mediante actividades colaborativas mediadas por computadora, que involucran a pequeños grupos de estudiantes, permitiendo el aprendizaje a través de una dinámica de colaboración con independencia del tiempo y espacio donde estén localizados los miembros de esos grupos.

Un grupo de aprendizaje se define como una estructura formada por personas que interactúan para lograr objetivos específicos de aprendizaje a través de su participación [2]. En ACSC hay diferentes enfoques para formar grupos, es posible: seleccionar de manera aleatoria a los miembros; permitirles agruparse a sus propios miembros; o escoger los integrantes atendiendo ciertos criterios establecidos por el docente; además, estos enfoques de formación de grupos pueden concretarse de manera manual (por

el docente) o automática (por el sistema) [3]. Una conformación por aleatoriedad puede generar grupos muy desbalanceados con poca probabilidad de resultar efectivos [11] [14] [13], la autoselección puede causar discriminación entre los estudiantes con relaciones sociales pobres [3], y la creación manual es inviable cuando el número de estudiantes es elevado o cuando los criterios de selección son complejos.

En [3] se revisan distintos enfoques y técnicas de aprendizaje de máquina utilizados hasta el momento para la formación automática de grupos, lo cual incluye: algoritmos genéticos, agentes, clustering, optimización de restricciones, etc. También queda claro que los criterios aplicados para guiar el agrupamiento son variados, algunos investigadores consideran estilos psicológicos, otras relaciones sociales, nivel de participación, estilos de aprendizaje, nivel de conocimiento, etc. Según [3] muchos de estos enfoques han sido testeados, pero orientando las pruebas a validar el correcto funcionamiento del algoritmo de agrupamiento más que a evaluar los efectos de tal agrupamiento sobre el rendimiento de los grupos formados.

En [15] se presenta un estudio de técnicas de data mining aplicadas en ámbitos educativos, diferenciando por temática tales como análisis y visualización de datos, generación de feedback para docentes, sistemas de recomendación para estudiantes, predicción de rendimiento de estudiantes, modelado de estudiantes, detección de comportamientos no deseables en estudiantes, análisis de redes sociales, desarrollo de mapas conceptuales de manera automática, construcción automática de cursos y planificación. Se incluye también una sección en la que se expone cómo han sido usadas las técnicas de data mining, tanto supervisadas como no supervisadas, para realizar agrupamientos de estudiantes.

Encontrar la condición óptima para agrupar estudiantes es un problema NP-Hard [14] y se necesitan tiempos exponenciales para resolverlo. Por esta razón suelen utilizarse algoritmos de aproximación que, si bien no necesariamente llegan a la solución óptima, se acercan considerablemente, constituyéndose en una solución apropiada. Uno de los algoritmos más usados para resolver problemas de este tipo son los algoritmos genéticos [8].

En [10] [11] [12] [13] [16] [17] se proponen algoritmos genéticos para la formación de grupos heterogéneos. Un grupo es heterogéneo cuando en sus integrantes se observa diversidad de atributos que pueden ser necesarios para desarrollar una actividad. Como atributos se pueden considerar, por ejemplo: género, etnia, nivel de conocimientos previos, motivaciones, actitudes, intereses, habilidades, estilo de aprendizaje, logros de aprendizajes previos, factores psicológicos como personalidad y estilos de pensamiento, etc. En [18] se propone conformar grupos bien balanceados en base a los nueve roles presentes en los equipos según el modelo de los roles de Belbin. Un grupo no balanceado es aquel cuyos roles no aparecen naturalmente o cuando el mismo rol se manifiesta en diferentes miembros. Los autores sugieren la existencia de tres indicadores y la implementación de un algoritmo genético. El primer indicador muestra cuándo cada uno de los roles aparece naturalmente en el grupo. El segundo indicador se aplica a cada grupo y calcula el nivel de balance de los roles basado en el primer indicador. El tercer indicador maximiza el nivel de balance promedio en todos los grupos, y se toma como indicador de la función de evaluación del algoritmo genético.

En el presente trabajo se presenta un algoritmo genético para la generación automática de grupos colaborativos que, teniendo en cuenta los estilos de aprendizaje de los estudiantes, tiende a

optimizar la cantidad de grupos que logren alcanzar buenos resultados académicos.

El artículo se organiza como sigue. En la próxima sección se establece qué se entiende por estilos de aprendizaje. En la sección 3 se describe específicamente el algoritmo genético creado para conformar de manera automática grupos de estudiantes colaborativos. En la sección 4 se presentan las configuraciones con las que se efectuó la experimentación y se analizan los resultados obtenidos. Finalmente, en la sección 5, se enuncian algunas conclusiones vinculadas con el trabajo realizado.

2. ESTILOS DE APRENDIZAJE

Un modelo de estilo de aprendizaje clasifica a los estudiantes de acuerdo con la forma en cómo reciben y procesan la información. En particular, Felder y Silverman [5] propusieron un modelo aplicable sobre estudiantes de ingeniería. Los autores presentaron en su trabajo original las dimensiones percepción, entrada, organización, procesamiento y comprensión, como variables de análisis para identificar los estilos de aprendizaje manifestados. En una versión posterior suprimieron la dimensión organización. Los tipos de aprendizajes resultantes considerando las dimensiones propuestas se describen brevemente en Tabla 1.

Tabla 1. Dimensiones y estilos de aprendizaje

Dimensión	Estilos	Descripción
Procesamiento	<ul style="list-style-type: none"> • Activo • Reflexivo 	Un estudiante activo se siente más cómodo con la experimentación que con la observación reflexiva, al revés de un estudiante reflexivo. Un estudiante activo no aprende en situaciones de pasividad y trabaja bien en grupo. Un estudiante reflexivo no aprende en situaciones que no proporcionan la oportunidad de pensar sobre la información que se le presenta, trabaja mejor solo y tiende a ser teórico.
Percepción	<ul style="list-style-type: none"> • Sensitivo • Intuitivo 	Un estudiante sensitivo gusta de hechos, datos, y experimentaciones; resuelve problemas comprendiendo los métodos, no le agradan las sorpresas ni las complicaciones; es paciente con los detalles; es bueno memorizando hechos; es cuidadoso pero lento. Un estudiante intuitivo prefiere principios y teorías; gusta de innovaciones y complicaciones; y no de repetición; se aburre con detalles; es bueno para asimilar nuevos conceptos.
Entrada	<ul style="list-style-type: none"> • Visual • Verbal 	Un estudiante visual recuerda mejor lo que ve (figuras, diagramas, cuadros, demostraciones, etc.). Un estudiante verbal recuerda más lo que escucha, aprende a partir de la discusión, y prefiere las explicaciones verbales a las demostraciones visuales.
Comprensión	<ul style="list-style-type: none"> • Secuencial • Global 	Un estudiante secuencial aprende en un orden de progresión lógica, regido por el tiempo y el calendario. Un estudiante global no se rige por el tiempo ni el calendario, puede pasar días ocupado en resolver un simple problema o demostrando poca comprensión hasta que de repente logra una rápida comprensión del todo.

Los estilos de aprendizajes de Tabla 1 pueden combinarse. Así, es posible encontrar estudiantes que responden a un estilo de aprendizaje sensitivo, visual, activo, secuencial; mientras que otros lo hacen a un estilo intuitivo, visual, reflexivo, global; u a otra combinación posible. Respondiendo a este modelo, Felder y Soloman [6] crearon el Test de estilos de aprendizaje. Este es un instrumento que permite determinar el estilo de aprendizaje de un estudiante, mediante cuarenta y cuatro preguntas con dos opciones de respuesta cada una (a y b), en las que el estudiante debe elegir necesariamente sólo una de ellas.

3. ALGORITMO GENÉTICO PROPUESTO

Para resolver el problema planteado en este artículo proponemos un algoritmo genético de agrupamiento. Los algoritmos genéticos fueron inventados por John Holland en el año 1975. Holland propuso los algoritmos genéticos como un método heurístico basado en la supervivencia de los mejores. Estos tipos de algoritmos son adecuados para problemas de búsqueda y optimización [7]. Estos algoritmos crecieron en popularidad debido a su capacidad para desenvolverse bien en muchos tipos diferentes de problemas [8]. Si bien no son algoritmos veloces, ofrecen buenas heurísticas para problemas combinatorios. Estos algoritmos parten de una población inicial de soluciones candidatas para un problema y evolucionan hacia una solución óptima basados en los principios de selección natural y supervivencia del más apto, aplicando operaciones de cruzamiento y mutación.

El comportamiento general del algoritmo que se propone en este trabajo es el siguiente. Considere una clase de n estudiantes que será dividida en m grupos con k integrantes por grupo, siendo k un valor establecido por el docente. El algoritmo inicia la evolución desde una población inicial de soluciones candidatas. Cada solución codifica un conjunto de m grupos los cuales pueden ser definidos cuando los n estudiantes son distribuidos. Luego, cada solución de la población se decodifica y evalúa por una función de *fitness*. La función *fitness* evalúa cada solución en relación al objetivo de optimización del problema. En nuestro caso se desea maximizar la cantidad de grupos asociados con un buen rendimiento. Por lo tanto, tomando en consideración una solución dada, la función evalúa los m grupos representados por la solución. Algunas soluciones de la población son seleccionadas y luego emparejadas. En general, las soluciones con los valores de *fitness* más elevados tienen más chances de ser seleccionadas. Luego, se aplica la operación de cruce a cada par solución para generar nuevas alternativas factibles. La operación de mutación se realiza posteriormente sobre las soluciones generadas por la operación de cruce. El objetivo de la mutación es introducir diversidad genética en las soluciones. El cruce es exploratorio en tanto que la mutación es explotadora. La exploración permite descubrir áreas prometedoras en el espacio de búsqueda (ganando información sobre el problema); la explotación permite optimizar un área prometedora (usando información) [7].

Finalmente, una estrategia de selección de sobrevivientes se usa para crear una nueva población a partir de la población actual y las nuevas soluciones generadas. Este proceso se repite hasta alcanzar un criterio de detenimiento. Para nuestro trabajo, el criterio definido es alcanzar un número predeterminado de repeticiones o iteraciones. En las secciones siguientes se analizará con mayor detalle el algoritmo genético desarrollado.

3.1 Representación de la solución

Cada solución en la población del algoritmo evolutivo representa o codifica un conjunto G de m grupos los cuales son construidos

cuando los n estudiantes de la clase son divididos. La solución debe ser codificada de modo tal que los diferentes operadores de cruzamiento y mutación generen nuevas soluciones factibles. Las soluciones codificadas se conocen como cromosomas.

El problema en cuestión puede tratarse como un caso de permutación. Esto lleva a considerar que una forma adecuada de representar los grupos es como una lista de números enteros en donde cada elemento de la lista representa un alumno. La lista a su vez es subdividida en sublistas de longitud k , siendo k la cantidad de integrantes que conformarán cada grupo (valor especificado por el profesor). En la Figura 1 se ilustra la estructura de un cromosoma.

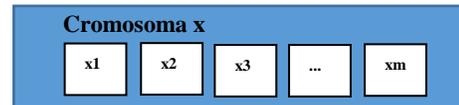


Fig. 1. Representación codificada de una solución

En la Figura 1, $x_1, x_2, x_3, \dots, x_m$ son los m grupos generados. Cada x_m es una lista de números enteros de tamaño k , siendo k el total de integrantes por grupo. Cada número entero en este bloque se correlaciona con el número de orden que le corresponde a un alumno en una lista numerada.

3.2 Decodificación de la solución

Dado que la solución se representa como una lista de números enteros, se necesita un proceso que permita interpretar los grupos formados y su constitución. Se sabe que los grupos tienen que ser de tamaño k , donde k es especificado por el profesor, y que los grupos deben estar balanceados en cuanto a la cantidad de integrantes, es decir, la diferencia máxima entre miembros entre los distintos grupos debe ser de 1.

Para determinar la cantidad de grupos m a formar, se emplea el cálculo que muestra la ecuación (1).

$$m = \text{redondeo} \left(\frac{n}{k} \right) \quad (1)$$

Si la cantidad de alumnos n es múltiplo de k , cada grupo tendrá k integrantes. Cuando n no es un múltiplo de k , se pueden tener grupos con tamaño k y $k - 1$ integrantes, o bien, grupos con tamaño $k - 1$ y $k - 2$ integrantes. En los casos que a) $(k - n) \bmod k \leq m + 1$, se configurarán g_{11} grupos con k integrantes y g_{12} grupos con $k - 1$ integrantes, en tanto que, para b) $(k - n) \bmod k > m - 1$, se dispondrán de g_{11} grupos con $k - 1$ integrantes y g_{12} grupos con $k - 2$ integrantes. Para calcular g_{11} y g_{12} se aplica la ecuación (2).

Por ejemplo, si se tienen 11 alumnos ($n = 11$) y el profesor desea crear grupos con 4 integrantes ($k = 4$), por (1) m es igual a 3. Como $(k - n) \bmod k \leq m + 1$ se tendrán grupos con cuatro integrantes y grupos con tres integrantes. Para saber cuántos grupos de cada uno se deben formar, se resuelve el sistema de ecuaciones planteado para el caso a). En la Figura 2 se resuelve esta situación.

Según los cálculos mostrados en la Figura 2, se deben crear dos grupos de cuatro integrantes y un grupo de tres integrantes. Los valores de las variables g_{11} y g_{12} son fundamentales al momento de recorrer la lista para determinar los grupos formados.

Caso a)

$$\begin{cases} g_{t1} + g_{t2} = m \\ kg_{t1} + (k-1)g_{t2} = n \end{cases} \quad (2)$$

Caso b)

$$\begin{cases} g_{t1} + g_{t2} = m \\ (k-1)g_{t1} + (k-2)g_{t2} = n \end{cases}$$

$$\begin{cases} gt1 + gt2 = 3 \\ 4gt1 + 3gt2 = 11 \end{cases}$$

$$gt1 = \frac{\begin{vmatrix} 3 & 1 \\ 11 & 3 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 4 & 3 \end{vmatrix}} = \frac{-2}{-1} = 2$$

$$gt2 = \frac{\begin{vmatrix} 1 & 3 \\ 4 & 11 \end{vmatrix}}{\begin{vmatrix} 1 & 1 \\ 4 & 3 \end{vmatrix}} = \frac{-1}{-1} = 1$$

Fig. 2. Determinación de la cantidad de grupos a formar siendo $n = 11$ y $k = 4$

3.3 Población inicial

La población inicial contiene un número específico de soluciones codificadas factibles. Cada solución codificada consiste de una permutación de los n estudiantes de la clase. Se diseñó un método aleatorio para generar cada solución de esta población. Este tipo de método garantiza un buen nivel de diversidad genética en la población inicial, y por lo tanto, ayuda a prevenir la convergencia prematura del algoritmo.

El método funciona de la siguiente manera. Empieza con una lista vacía a la cual se le agrega un nuevo elemento en cada iteración hasta lograr la longitud n que es la cantidad de alumnos disponibles. Cada elemento que se agrega a la lista es un alumno elegido al azar de una lista numerada. La lista de alumnos consiste de un número de orden, el nombre completo del alumno y una cadena que representa el estilo de aprendizaje del alumno, proporcionado al realizar el test de Felder y Soloman [6]. La cadena está constituida en cuatro partes, cada una de las cuales indica un valor correspondiente a cada una de las dimensiones visualizadas en la Tabla 2. De esta manera se genera una solución aleatoria para la población.

En cada dimensión de los estilos de aprendizaje se admite el valor neutro. Por ejemplo, en la dimensión de procesamiento, cuando el valor activo o reflexivo no supera 3, se considera que se tiene neutro en tal dimensión. De manera similar ocurre con las restantes dimensiones. En la Figura 3 se muestra una situación hipotética del resultado conseguido por una persona al realizar el test de estilos de aprendizaje de Felder y Soloman. En este caso, la persona tiene estilo de aprendizaje neutral en la dimensión de procesamiento; neutral, en percepción; neutral, en entrada; y global en la dimensión de comprensión.

3.4 Función fitness

El *fitness* en un algoritmo genético es la función que define la bondad de una solución en el dominio real. Para calcular el *fitness*, el cromosoma tiene que ser primero decodificado y luego la función tiene que ser evaluada. Podría afirmarse que el valor de *fitness* no sólo indica cuán buena una solución es, sino que también qué tan próximo el cromosoma se corresponde con la solución óptima [7].

Tabla 2. Valores que puede tomar cada parte que compone la cadena que indica el estilo de aprendizaje de un alumno

Dimensión del Estilo de Aprendizaje	Valores	Condición
Procesamiento	Activo	Valor activo > 3
	Reflexivo	Valor reflexivo > 3
	Neutral	Valor activo <= 3 ó
	Activo Reflexivo	Valor reflexivo <= 3
Percepción	Sensitivo	Valor sensitivo > 3
	Intuitivo	Valor intuitivo > 3
	Neutral	Valor sensitivo <= 3 ó
	Intuitivo Sensitivo	Valor intuitivo <= 3
Entrada	Visual	Valor visual > 3
	Verbal	Valor verbal > 3
	Neutral	Valor visual <= 3 ó
	Visual Verbal	Valor verbal <= 3
Comprensión	Secuencial	Valor secuencial > 3
	Global	Valor global > 3
	Neutral	Valor secuencial <= 3 ó
	Secuencial Global	Valor global <= 3

ACT	11	9	7	5	3	1	1	3	5	7	9	11	REF
						X							
						<-- -->							
SEN	11	9	7	5	3	1	1	3	5	7	9	11	INT
						X							
						<-- -->							
VIS	11	9	7	5	3	1	1	3	5	7	9	11	VRB
						X							
						<-- -->							
SEQ	11	9	7	5	3	1	1	3	5	7	9	11	GLO
						X							
						<-- -->							

Fig. 3. Ejemplo de estilos de aprendizaje de una persona

El algoritmo genético propuesto permite la formación automática de grupos en base a la cantidad de integrantes especificado por el profesor, y con la condición de optimizar la cantidad de grupos que obtienen buen rendimiento, constituyéndose este último requisito en la función de *fitness*. En la ecuación (3) se muestra la función a optimizar.

$$fitness = \max \left(\frac{total\ grupos\ con\ buen\ rendimiento}{total\ grupos} \right) \quad (3)$$

Para determinar si un grupo conseguirá un buen rendimiento se aplica una función que implementa las reglas de asociación obtenidas al procesar datos históricos sobre rendimientos de grupos en base a los estilos de aprendizaje de sus integrantes. Las reglas y el proceso seguido para su obtención se documentan en [4].

3.5 Selección de padres

La selección es el operador genético por el que se escogen dos padres de la población para cruzarlos. El propósito de la selección es enfatizar los individuos más aptos en la población para reproducirlos con la esperanza de que sus hijos tengan mejor *fitness*. Así, los cromosomas de la población son seleccionados para ser padres para reproducir.

Para desarrollar el proceso de selección de padres se decidió diseñar la selección por torneo y también la selección basada en ranking. En la selección por torneo se eligieron 3 cromosomas al azar de los cuales se selecciona al que tiene el mejor *fitness*. En la selección por ranking se ordenan los cromosomas en forma creciente en función a su *fitness* y se selecciona a los que se ubican en las primeras posiciones de la lista. La probabilidad de selección por ranking se establece en base al mapeo exponencial mostrado en la ecuación (4), en el cual, el valor de c se elige de forma que la suma de las probabilidades sea igual a 1.

$$P_{exp-rank}(i) = \frac{1 - e^{-i}}{c} \quad (4)$$

3.6 Mecanismo de cruce

El operador de selección determina un número de pares de solución que deberían ser recombinados, y cada par experimenta el operador de cruce con una probabilidad P_c [9]. En este trabajo, se utilizó una probabilidad de cruce de 50%. El cruce aplicado en un par de soluciones genera dos nuevas soluciones (soluciones hijas). El operador de cruce es uno de los operadores genéticos más importantes porque preserva y combina las mejores características de las soluciones padres de modo que, mejores soluciones puedan ser definidas [7].

En el presente trabajo se experimentó con el cruce de orden (Figura 4) y el cruce PMX (Figura 5), los cuales son adecuados para problemas de permutación.

1. Elegir dos puntos de cruce al azar
2. Copiar los valores del padre 1 que se encuentran entre los dos puntos de cruce en el hijo 1.
3. Copiar en el hijo 1 los valores que aún no se han incluido en dicho hijo:
 - a. Comenzar a partir del segundo punto de cruce del padre 2.
 - b. Copiar los valores no incluidos en el hijo 1 respetando el orden en el cual dichos valores aparecen en el padre 2.
 - c. Al terminar la lista del padre 2, continuar con los primeros valores de la misma.
4. El segundo hijo es creado de manera análoga (pasos 2 y 3) invirtiendo el rol de los padres.

Fig. 4. Procedimiento del cruce de orden

3.7 Mecanismo de mutación

El operador de mutación ayuda a introducir diversidad genética en la población. Para hacerlo, el operador aleatoriamente altera una o más características de alguna solución obtenida después de aplicar el operador de cruce. Al hacer esto, se previene que el algoritmo sea atrapado por algún mínimo local. La mutación juega el rol de recuperador de material genético perdido, así como de productor de aleatorización del material genético [7].

El operador de mutación se aplica a cada solución obtenida por el proceso de cruce en base a una probabilidad P_m [9]. Se trabajó con una probabilidad de mutación de 3%. En particular, se implementaron dos alternativas de mutación adecuadas para problemas de permutación: por intercambio y por mezcla.

En la mutación por intercambio se seleccionan dentro del cromosoma dos miembros al azar y se intercambian las posiciones que ocupan en la lista. En la mutación por mezcla se eligen dentro del cromosoma dos posiciones al azar y los miembros delimitados por ambas posiciones son aleatorizados, es decir, se los reubica en lugares aleatorios entre las posiciones seleccionadas.

1. Elegir dos puntos de cruce al azar, y copiar los valores del padre 1 que se encuentran entre los dos puntos de cruce en el hijo 1.
2. Posicionarse en el primer punto de cruce sobre el padre 2, y revisar qué elementos del padre 2 (existentes entre los dos puntos de cruce) no han sido incluidos en el hijo 1.
3. Por cada elemento i de los mencionados, revisar qué elemento j ha sido copiado en su posición (sobre el padre 2) en el hijo 1.
4. Ubicar i , en el hijo 1, en la posición ocupada por j sobre el padre 2 (esto es posible de hacer porque j ya se ha ubicado en el hijo 1).
5. Si la posición ocupada por j en el padre 2 ya ha sido llenada en el hijo 1 por un elemento k , ubicar i en la posición ocupada por k en el padre 2.
6. Una vez revisados los elementos entre los puntos de cruce, las posiciones restantes del hijo 1 deben ser llenadas a partir del padre 2.
7. El segundo hijo es creado de manera análoga al hijo 1 invirtiendo los roles de los padres.

Fig. 5. Procedimiento del cruce PMX

3.8 Selección de sobrevivientes

La selección de sobrevivientes es la última etapa del proceso. En este paso se decide qué individuos seguirán formando parte de la población.

Para la selección de sobrevivientes se diseñaron dos alternativas: reemplazo generacional y elitismo. En el reemplazo generacional la población actual es totalmente reemplazada por los hijos de dicha población. Como esquema de elitismo se escogió preservar el mejor cromosoma de la población actual y reemplazar todos los demás cromosomas por los hijos de la dicha población.

4. RESULTADOS EXPERIMENTALES

En la sección 3 se detallaron las distintas alternativas de operadores genéticos que se diseñaron para el algoritmo genético creado. En la Tabla 3 se resumen estas alternativas.

El algoritmo genético desarrollado se ejecutó 400 generaciones. Las distintas alternativas mostradas en la Tabla 3 pueden combinarse de 16 maneras diferentes (Tabla 4). Cada una de estas combinaciones fue probada a fin de determinar cuál proporciona los mejores resultados.

Dado que las soluciones generadas por un algoritmo genético son estocásticas, cada posible combinación se ejecutó 10 veces determinándose el mejor *fitness*, el peor *fitness*, el *fitness* promedio, y el desvío estándar.

Para la experimentación se trabajaron con cuatro datasets, sobre los cuales se formaron 4 tipos de grupos que varían en cuanto a la cantidad de miembros. Los datasets se generaron de manera aleatoria, distribuyendo los estilos de aprendizaje de manera

uniforme. La Tabla 5 muestra las diferentes configuraciones de los grupos para los cuatro datasets.

Tabla 3. Alternativas de configuración de diseño implementadas

Representación	String de enteros
Cruce	Alternativa 1: De orden Alternativa 2: PMX
Mutación	Alternativa 1: Por intercambio Alternativa 2: Por mezcla
Selección de Padres	Alternativa 1: Por torneo Alternativa 2: Basada en Ranking (Mapeo Exponencial)
Selección de Sobrevivientes	Alternativa 1: Generacional Alternativa 2: Elitismo

Tabla 4. Combinaciones de las alternativas de diseño

#	Cruce	Mutación	Selección Padres	Selección Sobrevivientes
1	De Orden	Intercambio	Torneo	Generacional
2	De Orden	Intercambio	Torneo	Elitismo
3	De Orden	Intercambio	Ranking	Generacional
4	De Orden	Intercambio	Ranking	Elitismo
5	De Orden	Mezcla	Torneo	Generacional
6	De Orden	Mezcla	Torneo	Elitismo
7	De Orden	Mezcla	Ranking	Generacional
8	De Orden	Mezcla	Ranking	Elitismo
9	PMX	Intercambio	Torneo	Generacional
10	PMX	Intercambio	Torneo	Elitismo
11	PMX	Intercambio	Ranking	Generacional
12	PMX	Intercambio	Ranking	Elitismo
13	PMX	Mezcla	Torneo	Generacional
14	PMX	Mezcla	Torneo	Elitismo
15	PMX	Mezcla	Ranking	Generacional
16	PMX	Mezcla	Ranking	Elitismo

El algoritmo se ejecutó con tres tamaños de población, a saber, con tamaño de cinco, diez y veinte. En la Figura 6 se observa el comportamiento del algoritmo ante los distintos tamaños de población al trabajar con grupos de 4 integrantes. El eje X representa las distintas alternativas de diseño del algoritmo genético descritas en Tabla 4. Como puede observarse trabajando con tamaño de población 20 se tiende a obtener mejores valores de *fitness* y un menor desvío estándar. En general este comportamiento se observa en todos los casos, independientemente del dataset usado y del tipo de grupo a formar (#Grupos) de Tabla 5

Tabla 5. Configuraciones de grupos a experimentar sobre cada dataset

DS	Total alumnos	#Grupos (4 int.)	#Grupos (5 int.)	#Grupos (6 int.)	#Grupos (7 int.)
1	15	3 de 4 int 1 de 3 int	3 de 5 int	3 de 5 int	3 de 5 int
2	30	6 de 4 int 2 de 3 int	6 de 5 int	5 de 6 int	5 de 6 int
3	45	9 de 4 int 3 de 3 int	9 de 5 int	5 de 6 int 3 de 5 int	3 de 7 int 4 de 6 int
4	60	15 de 4 int	12 de 5 int	10 de 6 int	6 de 7 int 3 de 6 int

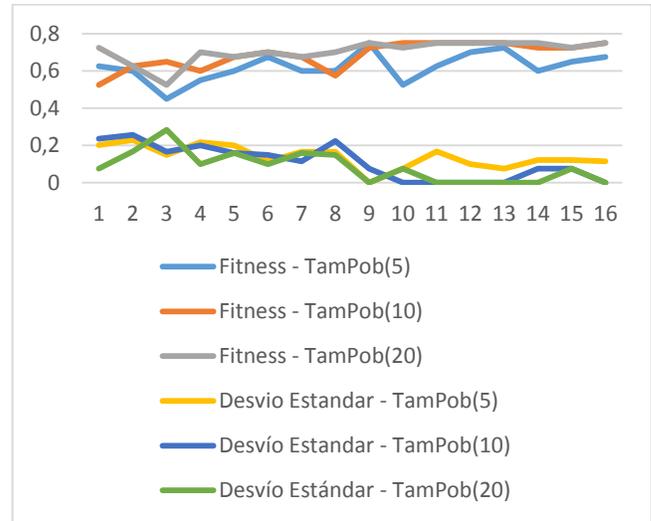


Fig. 6. Valores de *fitness* y desvío estándar al formar grupos con 4 integrantes

En las Figuras 7, 8, 9 y 10 se visualiza la función de *fitness* para cada uno de los datasets, respectivamente, considerando tamaño de población veinte. El eje X representa las distintas alternativas de diseño del algoritmo genético. Asimismo, en cada figura se muestra el valor asumido por la función de *fitness* al formar los distintos tipos de grupos considerados.

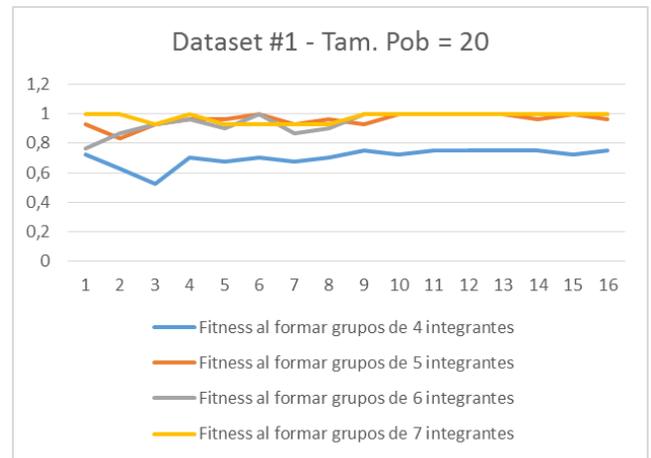


Fig. 7. Valores de *fitness* obtenidos en la experimentación con el dataset I y tamaño de población 20

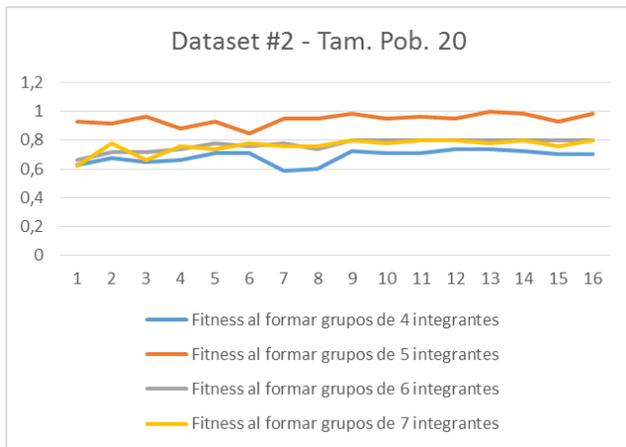


Fig. 8. Valores de fitness conseguidos en la experimentación con el dataset II y tamaño de población 20

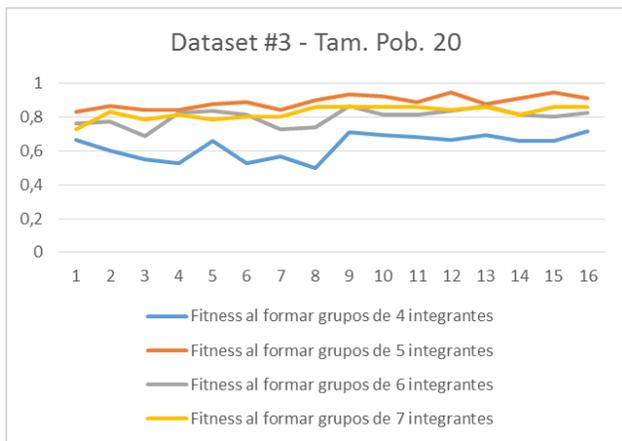


Fig. 9. Valores de fitness conseguidos en la experimentación con el dataset III y tamaño de población 20

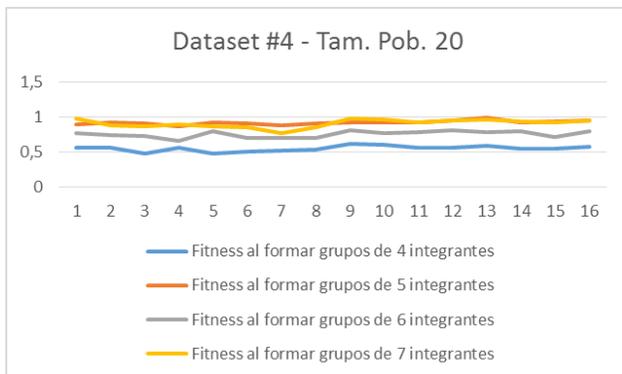


Fig. 10. Valores de fitness conseguidos en la experimentación con el dataset IV y tamaño de población 20

Según los resultados obtenidos, se observa que la función de *fitness* tiende a obtener mejores valores a partir de la alternativa 9. Quizás esto esté vinculado con las alternativas de cruce y mutación involucradas, ya que las alternativas comprendidas entre 9 y 16 comparten la característica de usar el operador de cruce PMX y el operador de mutación por intercambio o mezcla.

5. CONCLUSIONES

En el presente trabajo se presentó un enfoque para la formación automática de grupos en base al estilo de aprendizaje de sus miembros con el objetivo de maximizar la cantidad de grupos que logran obtener un buen rendimiento.

Para lograr el objetivo se propuso un algoritmo genético. Teniendo en cuenta que los estudiantes de la clase deben ser divididos en grupos, el algoritmo diseña diferentes alternativas de agrupamiento teniendo en cuenta los estilos de aprendizaje de los alumnos y bajo la restricción de producir grupos de tamaño balanceado, es decir, que la diferencia máxima de integrantes entre dos grupos cualesquiera no sea superior a 1. A cada alternativa de agrupamiento el algoritmo la evalúa en función a la cantidad de grupos que predice que obtendrán buen rendimiento. Para hacer estas predicciones el algoritmo creado se vale del conocimiento obtenido en una investigación previa, en la cual se aplicaron reglas de asociación, sobre datos históricos de experiencias de ACSC con grupos de estudiantes [4].

Se presentaron distintas alternativas de diseño del algoritmo genético y se experimentó con cada una de ellas. Se determinó que al operar con tamaño de población 20, mecanismo de cruce PMX, y operador de mutación por intercambio o mezcla, el algoritmo tiende a producir los mejores resultados en cuanto a *fitness* promedio y bajo valor para el desvío estándar.

Como trabajo futuro se planifica seguir recolectando datos sobre rendimiento de grupos a fin de refinar las reglas de asociación con las que se genera el conocimiento que utiliza el algoritmo genético creado, puesto que ese conocimiento influye directamente en la función de *fitness*, y por ende, en los resultados obtenidos para la formación de grupos. A futuro, también se pretende verificar en situaciones reales el rendimiento obtenido por los grupos formados automáticamente por el algoritmo genético creado.

6. AGRADECIMIENTOS

Este trabajo se realizó financiado parcialmente por los proyectos PICTO UNSE 2012-0016 y SECYT UNSE 23-C089.

7. REFERENCIAS

- [1] Slavin, R. 1995. *Cooperative Learning: Theory, Research and Practice*. Allyn and Bacon, Boston, Massachusetts.
- [2] Souto, M. 1993. *Hacia una didáctica de lo grupal*. Miño y Dávila, Buenos Aires.
- [3] Costaguta, R. 2015. Algorithms and Machine Learning Techniques in Collaborative Group Formation. *LNAI 9414: Advances in Artificial Intelligence and Its Applications*. Pichardo Lagunas, Obdulia, Herrera Alcántara, Oscar, Arroyo Figueroa, Gustavo (Eds.). Springer International Publishing, Berlin. 249-258.
- [4] Costaguta, R., Menini, M., and Lescano, G. 2016. Applying Data Mining to Discover Successful Collaborative Groups. In *Proceedings of 8th. Euro American Conference on Telematics and Information Systems* (Cartagena, Colombia, April 27 - 29, 2016). EATIS 2016 (en prensa).
- [5] Felder, R. and Silverman, L. 1988. Learning and Teaching Styles in Engineering Education Application. *Journal of Engineering Education*. 78, 7, 674-681.
- [6] Felder, R. and Soloman, V. 1984. *Index of Learning Styles*. Disp. en: <http://www.ncsu.edu/felder-public/ILSpage.html>. Último acceso: 03/03/2016.

- [7] Sivanandam, S. and Deepa, S. 2008. *Introduction to Genetic Algorithms*. Springer International Publishing, Berlin.
- [8] Akman, O. 2014. Introducing Evolutionary Computing in Regression Analysis. *Journal of Computational Science Education*. 5, 1, 23-27.
- [9] Sastry, K., Goldberg, D. and Kendall, G. 2005. Genetic Algorithms. In: *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Edmund K. Burke, Graham Kendall (Eds). Springer International Publishing, Berlin. 97-125.
- [10] Gogoulou, A., Gouli, E., Boas, G., Liakou, E. and Grigoriadou, M. 2007. Forming Homogeneous, Heterogeneous and Mixed Groups of Learners. In *Proceedings of the 11th International Conference on Personalisation in E-Learning Environments at Individual and Group Level* (Corfu, Greece, June 25-29, 2007). P. Brusilovsky, M. Grigoriadou, K. Papanikolaou (Eds.). 33-40.
- [11] Wang, D., Lin, S. and Sun, C. 2010. DIANA: A computer-supported heterogeneous grouping system for teachers to conduct successful small learning groups. *Computers in Human Behavior*. 23, 4 (July, 2007), 1997–2010.
- [12] Ani, Z., Yasin, A., Husin, M. and Hamid, Z. 2010. A Method for Group Formation Using Genetic Algorithm. *International Journal on Computer Science and Engineering*. 2, 9, 3060-3064.
- [13] Moreno, J., Ovalle, D. and Vicari, R. 2012. A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics. *Computers & Education*. 58, 1 (January, 2012), 560– 569.
- [14] Lin, Y., Huang, Y. and Cheng, S. 2010. An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization. *Computers & Education*. 55, 4 (December, 2010), 1483– 1493.
- [15] Romero, C. and Ventura, S. 2010. Educational Data Mining: A Review of the State of the Art. *IEEE Transactions on systems, man, and cybernetics, Part C (Applications and Reviews)*. 40, 6, 601-618.
- [16] Sukstrienwong, A. 2014. A Genetic Algorithm Approach for Forming Heterogeneous Groups of Students. *International Journal of Applied Engineering Research (IJAER)*. 9, 3, 297-311.
- [17] Barati Jozán, M., Taghiyareh, F. and Faili, H. 2012. An Inversion-based genetic Algorithm for Grouping of Students. In *Proceedings of 7th International Conference on Virtual Learning* (Bucharest, Romania, November 2 - 3, 2012). ICVL 2012. 152-161.
- [18] Yannibelli, V. and Amandi, A. 2012. A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. *Expert Systems with Applications*. 39, 10, 8584-8592.