

Distribución Óptima de Horarios de Clases utilizando la técnica de Algoritmos Genéticos

Tesis Profesional
Que para obtener el Título de

Ingeniero en Computación

Presenta

CARLA LENINCA PACHECO AGÜERO

Acatlima, Huajuapán de León, Oaxaca

Agosto, 2000

Tesis Profesional presentada
el 11 de Agosto del 2000
ante los sinodales:

Dr. Juan Carlos Mendoza
Dra. Virginia Berrón Lara
Ing. Francisco Espinoza Maceda

Asesores:
Dr. Carlos A. Coello Coello
Laboratorio Nacional de Inteligencia Artificial
Xalapa, Veracruz

Dr. Manrique Mata Montero
University Memorial of Newfoundland
St. John's, NF, Canadá

M.C. Agustín Santiago Alvarado
Universidad Tecnológica de la Mixteca

A mis queridos padres, Julio y Antonia,
por su apoyo constante, su confianza y
su inmenso cariño.

A Iván, por su amor incondicional y por
darme ánimos durante la realización de
este trabajo.

A mis hermanos: César, Julio y Mayra,
para que este trabajo les sirva como
estímulo para que alcancen su meta.

A mis sobrinas Tani y Pame por ser los
seres mas puros y maravillosos sobre la
tierra.

Agradecimientos

A mis padres y a Iván por ser parte fundamental de mi vida y porque gracias a su apoyo incondicional he logrado mis metas.

Al Dr. Carlos Coello Coello, por su amistad, por compartir su conocimiento conmigo y brindarme su ayuda durante el desarrollo de este trabajo.

Al Dr. Manrique Mata Montero, por tener la fortuna de contar con su amistad y apoyo durante el desarrollo de este trabajo.

2.6 Cuándo aplicar un Algoritmo Genético	29
CAPÍTULO 3 DISEÑO DEL ALGORITMO GENÉTICO PARA LA ASIGNACIÓN DE HORARIOS	
3.1 Introducción	30
3.2 Planteamiento de la solución a la asignación de horarios	31
3.3 Implementación del Algoritmo Genético	33
3.3.1 Entorno de Desarrollo	34
3.3.2 Interfaz del Sistema	35
CONCLUSIONES	40
REFERENCIAS	42
APÉNDICE A MANUAL PARA EL USUARIO DEL SISTEMA PARA LA ASIGNACIÓN DE HORARIOS	
Componentes del Sistema	44
Software del AG para la asignación de horarios	46
APÉNDICE B CÓDIGO DOCUMENTADO DE LOS PROCEDIMIENTOS Y FUNCIONES MAS IMPORTANTES DEL SISTEMA	
Programa para la asignación de horarios	61

ABSTRACT

Elaborar una asignación de horarios óptimos, es un problema por el que se atraviesa constantemente en cualquier institución educativa, sobre todo si su población de estudiantes crece desproporcionadamente con relación a su infraestructura. En el presente trabajo se describe cómo este problema puede ser resuelto utilizando un Algoritmo Genético. Esta técnica ha sido muy utilizada en problemas similares. En este caso se decidió utilizarla ya que combina rapidez y eficiencia, produciendo así buenos resultados. Actualmente, esta técnica de búsqueda ha alcanzado un impresionante éxito debido a que su aplicación en problemas de optimización no requiere de conocimiento auxiliar sobre el problema.

INTRODUCCION

Bajo el término de Técnicas Heurísticas Modernas (Reeves, 1993) se engloba a un conjunto de técnicas de búsqueda de soluciones, cuyo funcionamiento se inspira en procesos naturales. Entre estas técnicas se encuentran los Algoritmos Genéticos (Holland, 1975), el recocido simulado (Kirkpatrick, 1982) y la búsqueda tabú (Glover, 1989).

Estas técnicas se caracterizan por ser métodos que generan soluciones que, a partir de valores iniciales, emplean un conjunto de mecanismos de búsqueda para mejorar éstas. En los Algoritmos Genéticos, estos mecanismos se basan en los modelos biológicos de la evolución propuestos por Charles Darwin, en su libro "Del origen de las especies por medio de la selección natural" de 1859, tales como el mantenimiento de una solución (población) o la naturaleza probabilística de ciertos operadores (selección, reproducción, cruce,...), aunque ciertamente no son una réplica exacta del modelo de la evolución.

El objetivo de este trabajo es el de contribuir a optimizar los recursos, tanto materiales como humanos con que cuenta la Universidad Tecnológica de la Mixteca (UTM), misma que fue creada en el año de 1991 en la región del mismo nombre del estado de Oaxaca.

Para el ciclo escolar 2000-2001 la UTM cuenta en su infraestructura con 39 aulas, 5 salas de cómputo, 2 laboratorios de electrónica, 3 laboratorio de alimentos, 1 laboratorio de telecomunicaciones, etc., para un total aproximado de 1200 alumnos, que se encuentran divididos en grupos (de cuarenta estudiantes como máximo) que requieren como mínimo 5 horas de clase al día. Al hacer una asignación de cursos de forma manual para tal cantidad de alumnos, puede ocasionarse que los horarios sean en muchas ocasiones incompatibles con las actividades de investigación de los profesores de la planta docente de la UTM. Por lo que se pretende que mediante la elaboración de un Algoritmo Genético, trabajo de esta Tesis, se reduzcan los errores que se pueden presentar actualmente en la asignación de horarios. Para la realización de este algoritmo se deben tomar en cuenta las siguientes condicionantes:

- ❖ En las salas de cómputo no deben encontrarse más de 20 alumnos, por lo que los grupos que cuenten con más alumnos deberán ser reasignados.
- ❖ Las clases comienzan a las 08:00 hrs y terminan a las 18:00 hrs, teniendo un receso de 13:00 hrs a 16:00 hrs.
- ❖ Las clases de las asignaturas de Matemáticas y Física deberán impartirse en las primeras horas de clase.

El siguiente trabajo de Tesis, comprende un total de cuatro capítulos referentes a diversas áreas temáticas, dos apéndices y un anexo, estructurados de la siguiente manera:

En el primer capítulo se abordan algunos de los métodos de optimización que existen, así como sus características. Posteriormente se muestra una comparación con los Algoritmos Genéticos.

En el segundo capítulo se expone la investigación que se hizo respecto a los fundamentos y bases biológicas de los Algoritmos Genéticos, así como sus conceptos básicos y sus principales componentes.

En el tercer capítulo se da una introducción al problema a resolver así como la solución propuesta. Posteriormente se detalla como trabaja el sistema para la asignación de horarios mostrando también algunos resultados obtenidos

En el cuarto y último capítulo se muestran las conclusiones a las que se llegó después de realizar este trabajo de Tesis.

Al final de este documento se agregan los apéndices A y B, los cuáles contienen el manual del usuario y el código de las funciones y procedimientos más importantes del Algoritmo Genético para la asignación de horarios, respectivamente. También por último se anexa un ejemplo de asignaciones realizadas por el Algoritmo Genético, implementado en este trabajo, para los semestres de Octubre-Febrero y Marzo-Julio.

CAPITULO 1

MÉTODOS DE OPTIMIZACIÓN.

1.1 Introducción.

El término “optimización” se refiere a un problema complejo de decisión, que incluye la selección de valores para cierto número de variables interrelacionadas, centrando la atención en un solo objetivo diseñado para cuantificar el rendimiento y medir la calidad de la decisión. Este único objetivo se maximiza según las restricciones que pueden limitar la selección de valores de las variables de selección. [Luenberger, 1989]

A continuación se hace una clasificación de los métodos de optimización, mencionando brevemente cómo funcionan algunos de ellos.

1.2 Método del Gradiente.

Un número de diferentes métodos de optimización se comportan como funciones continuas que han sido desarrolladas para utilizar información del gradiente de la función para guiar la búsqueda del óptimo global. Estos métodos

como el del gradiente conjugado, que usa la información de la función objetivo así como de sus derivadas parciales, tienen algunas desventajas, ya que si la derivada de la función no puede ser computada por el hecho de ser discontinua, el método tiende a fallar. [Beasley, 1993]

1.3 Método del *hill climbing*.

Otro método es el denominado comúnmente como *hill climbing* (escalado de colinas), que no requiere de la información de las derivadas de la función y que funciona adecuadamente sólo si la función contiene un solo máximo (funciones unimodales). Pero en funciones con muchos máximos (funciones multimodales) tienden a cometer errores ya que el primer máximo local que encuentren en la función lo toman como si fuera el máximo global de la función y por tanto no continúan la búsqueda. Un ejemplo en una sola dimensión se presenta en la figura 1. El escalado comienza en un punto aleatorio **X** y se mueve en forma ascendente donde el máximo **B** es encontrado, por lo tanto los máximos mejores, **A** y **C** no son encontrados. [Beasley, 1993]

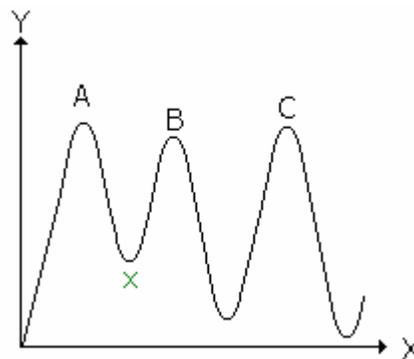


Figura 1. Método del escalado de colinas.

1.4 Búsqueda Iterativa.

La búsqueda aleatoria combinada con el método del gradiente da como resultado la búsqueda iterativa. Cada vez que se encuentra un máximo, la búsqueda comienza de nuevo pero con otro punto escogido al azar. Esta técnica es de enorme simplicidad y su desempeño es bueno en funciones que no contengan muchos máximos locales.

Cada vez que se busca otro punto aleatoriamente, éste es aislado y no se vuelve a tomar como parte del espacio de búsqueda. Esto significa que al estar buscando puntos éstos pueden encontrarse en regiones donde existan máximos locales y no globales. [Beasley, 1993]

1.5 Algoritmo Metrópolis.

Este método está basado en la analogía con el proceso de recocido de sólidos:

- Se incrementa la temperatura a una temperatura alta.
- Posteriormente se decrementa la temperatura muy lentamente hasta alcanzar un estado base.

La idea principal de este algoritmo es la siguiente:

Dado un estado inicial i , con energía E_i ,

Genera un nuevo estado j mediante un mecanismo de perturbación (pequeña distorsión del estado i).

Calcula la energía del estado E_j .

Si $(E_j - E_i) \leq 0$ entonces acepta el estado j como nuevo.

Si no, acepta el estado con probabilidad $\exp\left(\frac{E_i - E_j}{K_B \cdot T}\right)$ donde T denota la temperatura y K_B la constante de Boltzmann.

Si se baja la temperatura lentamente se puede alcanzar el equilibrio térmico en cada temperatura. Esto se hace mediante la generación de varias transiciones en cada temperatura.

El equilibrio térmico está caracterizado por la distribución de Boltzmann. La distribución da la probabilidad de que el sólido esté en el estado i con energía E_i a temperatura T y está dada por:

$$P_T\{X = i\} = \frac{1}{Z(T)} \exp\left(\frac{-E_i}{K_B T}\right)$$

Donde X es la variable estocástica que denota el estado actual del sólido y $Z(T)$ es la función de partición definida como:

$$Z(T) = \sum_j \exp\left(\frac{-E_j}{K_B T}\right). [\text{Morales, 1999}]$$

1.6 Recocido Simulado.

Esta técnica fue inventada por Kirkpatrick en 1982, está basada en el algoritmo Metrópolis propuesto en 1953 y es esencialmente una variación del *hill climbing*. Kirkpatrick introdujo los conceptos de recocido dentro del mundo de la optimización combinatoria. Comienza con la búsqueda de un punto escogido aleatoriamente y posteriormente con un movimiento aleatorio. Si el punto encontrado es uno de los máximos, entonces es aceptado. Si el punto es un mínimo entonces es aceptado con probabilidad $p(t)$, donde t es el tiempo. El valor inicial de la función $p(t)$ es muy cercano a uno y gradualmente se va acercando a cero; esto es análogo al enfriamiento de un sólido.

Inicialmente cualquier movimiento es aceptado, pero conforme la temperatura se va reduciendo, la probabilidad de aceptar un movimiento negativo es mínima. Algunas veces los movimientos negativos son esenciales sobre todo si no se ha encontrado el máximo global, aunque obviamente muchos movimientos negativos pueden desviarnos a la búsqueda del óptimo global.

Al igual que la búsqueda aleatoria, el recocido simulado trata con una solución el mismo tiempo, esto ocasiona que no se vaya construyendo la figura a partir del espacio de búsqueda. La información de movimientos previos no es guardada para guiar los nuevos movimientos. Esta técnica está en constante actividad de investigación (recocido rápido, recocido paralelo, etc.). [Beasley, 1993]

1.7 Búsqueda Tabú (BT).

La BT es un procedimiento heurístico de "alto nivel" introducido y desarrollado en su forma actual por Fred Glover (1989) y (1990).

En términos generales el método de BT puede resumirse de la siguiente manera: Se desea moverse paso a paso desde una solución factible inicial de un problema de optimización combinatoria hacia una solución que proporcione el valor mínimo de la función objetivo C . Para esto se puede representar a cada solución por medio de un punto s (en algún espacio) y se define una vecindad $N(s)$ para cada punto s como un conjunto de soluciones adyacentes a la solución s .

El paso básico del procedimiento consiste en empezar desde un punto factible s y generar un conjunto de soluciones en $N(s)$, de estas se elige la mejor s^* y se posiciona en este nuevo punto ya sea que $C(s^*)$ tenga o no mejor valor que $C(s)$.

Hasta este punto se está cercano a las técnicas de mejoramiento local a excepción del hecho de que se puede mover a una solución peor s^* desde s .

La característica importante de la BT es precisamente la construcción de una lista tabú T de movimientos: aquellos movimientos que no son permitidos (movimientos tabú) en la presente iteración. La razón de esta lista es la de excluir los movimientos que nos pueden regresar a algún punto de una iteración anterior. Ahora bien, un movimiento permanece como tabú sólo durante un cierto número

de iteraciones, de forma que se tiene que T es una lista cíclica donde para cada movimiento s a s^* el movimiento opuesto s^* a s se adiciona al final de T donde el movimiento más viejo en T se elimina. Las condiciones tabú tienen la meta de prevenir ciclos e inducir la exploración de nuevas regiones. Pero esto no representa que las restricciones tabú no sean inviolables bajo toda circunstancia. Cuando un movimiento tabú proporciona una solución mejor que cualquier otra previamente encontrada, su clasificación tabú puede eliminarse. La condición que permite dicha eliminación se llama *criterio de aspiración*.

Es así como las restricciones tabú y el criterio de aspiración de la BT, juegan un papel dual en la restricción y guía del proceso de búsqueda. Las restricciones tabú, permiten que un movimiento sea admisible si no está clasificado como tabú, mientras que si el criterio de aspiración se satisface, permite que un movimiento sea admisible aunque este clasificado como tabú.

En la BT tres aspectos merecen énfasis:

1. El uso de T proporciona la "búsqueda restringida" de elementos de la aproximación y por lo tanto las soluciones generadas dependen críticamente de la composición de T y de la manera como se actualiza.
2. El método no hace referencia a la condición de optimalidad local, excepto implícitamente cuando un óptimo local mejora sobre la mejor solución encontrada previamente.
3. En cada paso se elige al "mejor" movimiento.

Para problemas grandes, donde las vecindades pueden tener muchos elementos, o para problemas donde esos elementos son muy costosos de examinar, es de importancia aislar a un subconjunto de la vecindad, y examinar este conjunto en vez de la vecindad completa. Esto puede realizarse en etapas, permitiendo al subconjunto de candidatos expanderse si los niveles de aspiración no se encuentran. [Gutiérrez, 1999]

1.8 Búsqueda ciega o sin información.

El orden en que la búsqueda se realiza no depende de la naturaleza de la solución buscada. La localización de la meta(s) no altera el orden de expansión de los nodos. [Morales, 1999].

1.8.1 Búsqueda Primero en Anchura.

Explora progresivamente en capas del grafo de una misma profundidad. La búsqueda primero en anchura es una búsqueda completa (encuentra una solución si existe) y óptima (encuentra la solución más corta) si el costo del camino es una función que no decrece con la profundidad del nodo. Pero tiene dos desventajas: requiere de mucha memoria, ya que básicamente tiene que guardar la parte completa de la red que está explorando y la segunda es que en búsquedas de complejidad exponencial no se puede utilizar.

La búsqueda en amplitud trabaja de la siguiente forma, si se tiene un factor de arborescencia de b y la meta está a profundidad d , entonces el máximo número de nodos expandidos antes de encontrar una solución es: $1 + b + b^2 + b^3 + \dots + b^d$. [Morales, 1999]

1.8.2 Búsqueda primero en profundidad.

Esta búsqueda siempre expande uno de los nodos a su nivel más profundo y sólo cuando llega a un camino sin salida se regresa a niveles menos profundos.

Aquí cada nodo que es explorado genera todos sus sucesores antes de que otro nodo sea explorado. Después de cada expansión el nuevo hijo es de nuevo seleccionado para expansión.

Si no se puede continuar, se regresa al punto más cercano de decisión con alternativas no exploradas.

Esta búsqueda necesita almacenar un solo camino de la raíz a una hoja junto con los "hermanos" no expandidos de cada nodo en el camino.

Por lo que con un factor de arborescencia de b y profundidad máxima de m , su necesidad de almacenamiento es a los más bm .

Su complejidad en tiempo (cuánto se tarda) es $O(b^m)$ en el peor de los casos.

Para problemas con muchas soluciones la búsqueda en profundidad puede ser más rápida que de amplitud porque existe una buena posibilidad de encontrar una solución después de explorar una pequeña parte del espacio de búsqueda

La desventaja de la búsqueda en profundidad es que en situaciones donde los árboles de búsqueda tienen una profundidad muy grande puede caerse en ciclos infinitos. [Morales, 1999]

1.9 Búsqueda aleatoria.

La búsqueda aleatoria basa su trabajo en el uso de números aleatorios para encontrar puntos mínimos. Desde que las librerías de los programas de las computadoras tienen generadores de números aleatorios, esta búsqueda puede ser utilizada convenientemente. Algunas de las ventajas de esta búsqueda aleatoria son:

- ❖ Esta búsqueda puede trabajar adecuadamente aún si la función objetivo es discontinua y no diferenciable en alguno de sus puntos.
- ❖ La búsqueda aleatoria puede ser usada para encontrar el mínimo global aún cuando la función posea varios mínimos relativos.
- ❖ Esta búsqueda se puede aplicar cuándo otros métodos tienden a fallar al presentarse dificultades como: funciones con variaciones muy marcadas y funciones con regiones planas. [Rao, 1996]

1.10 Algoritmos Genéticos y los Métodos de optimización tradicionales.

La siguiente tabla muestra una breve comparación a modo de resumen de las características que poseen los Algoritmos Genéticos, con respecto a otros métodos de optimización. [Goldberg, 1989]

Algoritmos Genéticos	Métodos de optimización tradicionales.
<ul style="list-style-type: none"> ❖ Trabajan con parámetros codificados, es decir que deben codificarse como cadenas de longitud finita sobre algún alfabeto infinito. ❖ Utilizan poblaciones de puntos, es decir que se usa una base de datos de puntos simultáneamente, de tal forma que la probabilidad de quedar atrapados en óptimos locales se reduce. ❖ No necesitan conocimientos auxiliares sobre el problema ya que usan información de la función de evaluación con respecto a los cromosomas. ❖ Utilizan reglas de transición probabilísticas. 	<ul style="list-style-type: none"> ❖ Trabajan con los parámetros mismos. ❖ Operan sobre puntos individuales, ya que sus movimientos en el espacio de búsqueda se hacen de un punto a otro usando reglas de transición determinística. Esto puede ocasionar que se encuentren óptimos locales en lugar de óptimos globales. ❖ Requieren de mucha información auxiliar para trabajar adecuadamente. ❖ Usan reglas determinística.

CAPITULO 2

ALGORITMOS GENÉTICOS.

2.1 Introducción

Una de las características más notables de la Naturaleza es la existencia de organismos adaptados para la supervivencia en prácticamente cualquier ecosistema, incluso en los más inhóspitos. La importancia de este hecho se manifiesta cuando se considera que el medio ambiente se encuentra sometido a continuos cambios, lo cual motiva que ciertas especies se extingan y otras evolucionen y adquieran preponderancia gracias a su adaptación a la nueva situación.

Basándose en estos principios, muchos investigadores han intentado trasladar dichos principios al campo de la algoritmia, con el objetivo de obtener sistemas con características análogas. De aquí, que en los últimos años, la comunidad científica internacional haya mostrado un creciente interés en una nueva técnica de búsqueda de soluciones basada en la Teoría de la Evolución y que se conoce como Algoritmo Genético (AG).

Esta técnica se basa en los mecanismos de selección que utiliza la naturaleza, de acuerdo a los cuales los individuos más aptos de una población son los que sobreviven al adaptarse más fácilmente a los cambios que se producen en su entorno. Estos cambios se efectúan en los genes¹ de un individuo, y sus atributos más deseables, es decir los que le permiten adaptarse mejor a su entorno, se transmiten a sus descendientes cuando éste se reproduce sexualmente. [Coello, 1995]

Los AGs son procedimientos de búsqueda basados en el mecanismo de selección natural, lo que permite que la exploración de posibles soluciones en el espacio de búsqueda sea mayor que si se utilizaran métodos tradicionales. [Goldberg, 1999]. Estos algoritmos fueron desarrollados por John H. Holland² en 1960 para resolver problemas de aprendizaje de máquina (*machine learning*).

En un AG se define una estructura de datos que admite todas las posibles soluciones a un problema. Cada uno de los posibles conjuntos de datos admitidos por esa estructura será una solución al problema. Unas soluciones serán mejores, otras peores.

Solucionar el problema consiste en encontrar la solución óptima, y por tanto, los AGs son en realidad un método de búsqueda. Pero un método de búsqueda muy especial, en el que las soluciones al problema son capaces de reproducirse entre sí, combinando sus características y generando nuevas soluciones.

En cada ciclo se seleccionan las soluciones que más se acercan al objetivo buscado, eliminando el resto de las soluciones. Las soluciones seleccionadas se reproducirán entre sí, permitiendo de vez en cuando alguna mutación o modificación al azar durante la reproducción. [Gascón De la Herrán, 1999a]

¹ Se denomina gene o gen a cualquier posición a lo largo de una cadena que representa a un individuo.

² John Holland. Investigador de Universidad de Michigan, propuso los Algoritmos Genéticos como una técnica de búsqueda. [Holland, 1975]

Los AGs son programas de computadora que desarrollan diferentes patrones de la selección natural para poder resolver problemas muy complejos, en los cuales se desconoce la solución óptima global. Estos algoritmos hacen posible que se puedan explorar grandes espacios de posibles soluciones para un problema.

Su funcionamiento está regido por dos procesos básicos: la selección natural y la reproducción sexual. El primero es la conexión entre los cromosomas³ y la capacidad de sobrevivencia de sus estructuras decodificadas. Esto contribuye a que los cromosomas que codifican estructuras aptas se reproduzcan con más frecuencia que los que codifican estructuras poco aptas, así se determina que miembros de la población sobreviven para poder reproducirse, y el segundo, asegura la recombinación de genes de los progenitores. [Holland, 1992]

2.2 Bases Biológicas.

La evolución de los seres vivos es un proceso extremadamente complejo y fascinante. A lo largo de la historia se ha tratado de explicar el funcionamiento del mismo mediante diversas teorías. Una de ellas es el Maltusianismo que menciona que dada la elevada tasa geométrica de reproducción de todos los seres orgánicos, su número tiende a crecer de manera exponencial. Dado que los alimentos, espacio físico, etc. no lo hacen en la misma proporción, nacerán muchos más individuos de los que es posible que sobrevivan, y en consecuencia, generalmente se recurre a la lucha por la sobrevivencia, ya sea con individuos de la misma especie o de especies distintas, o simplemente con el entorno, intentando modificar sus características adversas. De aquí que un individuo, si actúa de un modo provechoso para él, tendrá una mayor probabilidad de sobrevivir, y será seleccionado naturalmente. [Malthus, 1860]

³ En biología son elementos orgánicos para codificar la estructura de los seres vivos, en el contexto de los AGs es una estructura de datos que contiene una cadena de genes.

La teoría de la selección de las especies sostiene que aquellos individuos de una población que posean las características más ventajosas dejarán proporcionalmente más descendencia en la siguiente generación; y si tales características se deben a diferencias genéticas, que pueden transmitirse a los descendientes, tenderá a cambiar la composición genética de la población, aumentando el número de individuos con dichas características. De esta forma, la población completa de seres vivos se adapta a las circunstancias variables de su entorno. El resultado final es que los seres vivos tienden a perfeccionarse en relación con las circunstancias que los envuelven. [Darwin, 1859]

Las bases de un AG son: la genética, la reproducción sexual y las mutaciones. Aunque los AGs están inspirados en las ideas biológicas de la teoría de la evolución, esto no quiere decir que sean una réplica del modelo exacto de la evolución.

En un algoritmo genético se establece una relación entre cada una de las posibles soluciones a un problema, y un código genético, patrón o cadena de información, que de alguna forma las representa. Este código genético podrá sufrir mutaciones, transmitirse a la descendencia y combinarse con otros mediante reproducción sexual. El código es seleccionado naturalmente decidiendo si se mantiene o no en el sistema. También puede ser evaluado cuantitativamente, asignándole un peso.

2.2.1 Reproducción Sexual.

En la reproducción sexual, la información genética de ambos progenitores se combina generándose el código genético de un nuevo individuo. Además, existe una cierta probabilidad de que en este proceso se produzca alguna mutación (variación al azar) del material genético resultante.

La reproducción sexual tiene la ventaja de que combina dos (o más) bloques de genes que ya han probado su efectividad, de manera que es posible que el patrón resultante herede las características deseables en ambos progenitores, en cuyo caso será seleccionado, aumentando la existencia de dichas características en la población. Lo más sencillo es que el patrón represente directamente una solución. Sin embargo, muchas veces esto no es posible.

En los seres vivos, los genes, más que determinar directamente las características que definen un nuevo individuo, definen un conjunto de reglas de desarrollo que, aplicadas y combinadas, provocarán dichas características. Para ello, la dotación genética, que existe en todas las células de cada ser vivo, controla la realización de ciertas reacciones químicas, de manera que los genes son los que determinan, pero no directamente, la constitución y comportamiento de dicho individuo. [LBS, 1999]

El operador de reproducción sexual puede ser implementado en un algoritmo de diferentes formas. Sin embargo la forma más fácil es la técnica conocida como la *rueda de la ruleta*. En esta técnica cada cadena de la población está representada como una porción de la rueda, de acuerdo con su evaluación. Posteriormente se gira la rueda y se escogen números al azar. La cadena seleccionada será aquella cuya evaluación sea mayor o igual al número aleatorio. Posteriormente cada cadena seleccionada para ser reproducida, es replicada exactamente. Esto se puede ver claramente con el siguiente ejemplo:

Consideremos una población inicial de cuatro cadenas:

```
0 1 1 0 1
1 1 0 0 0
0 1 0 0 0
1 0 0 0 1
```

Al aplicarles la función de aptitud (que puede verse como una caja negra), arroja los siguientes valores:

No.	Cadena	Aptitud	% del Total
1	0 1 1 0 1	169	14.4
2	1 1 0 0 0	576	49.2
3	0 1 0 0 0	64	5.5
4	1 0 0 1 1	361	30.9
		1170	100.0

Sumando las aptitudes de todas las cadenas obtenemos un total de 1170 puntos. Los porcentajes correspondientes de las cadenas pueden verse de la siguiente forma en la rueda de la ruleta:

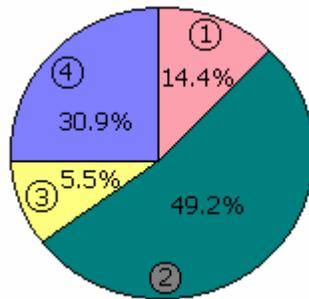


Figura 1. Reproducción simple utilizando la rueda de la ruleta, donde cada cadena es representada según el valor del % de la aptitud.

Para la reproducción, en este ejemplo la cadena número 1 tiene una aptitud de 169, la cual representa el 14.4% del total del mismo. Como resultado, la cadena 1 está representada en la rueda como el 14.4% de ella, es decir la probabilidad de que esta cadena sea seleccionada para la reproducción es de 0.144. Durante esta fase a las cadenas seleccionadas se les hace una copia idéntica.

Estas nuevas cadenas serán una población de padres a la cuál se le aplicará el operador de cruza.

2.2.2 Operador de Cruza.

Este operador es fundamental en la evolución, ya que efectúa el intercambio de material genético entre los individuos de una población. Este operador es lo que diferencia a los AGs de otros algoritmos. En un AG la cruza simple recombina el material genético de dos cromosomas para crear descendientes. Esta operación involucra dos pasos: 1) las cadenas de la nueva población son apareadas aleatoriamente, 2) cada par de cadenas experimentan un corte aleatorio e intercambian ese material genético entre ellas. La figura 2 muestra un ejemplo del operador de cruza. [Goldberg, 1989]

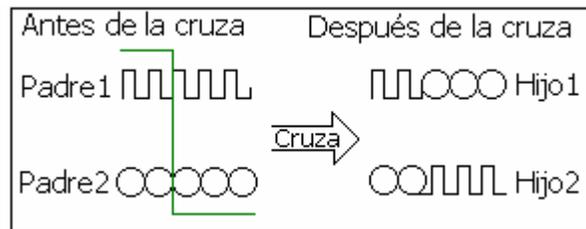


Figura 2. Operador de cruza, usando un punto de cruza escogido al azar.

2.2.3 Mutación.

Permite que las poblaciones se vuelvan heterogéneas, conservando así una diversidad que mantiene constante el proceso de evolución. Una mutación consiste en seleccionar aleatoriamente un cromosoma, posteriormente se escoge un gen al azar y se cambia el valor de este gen.

2.3 Representación.

En un AG tradicional una solución al problema puede considerarse como un cromosoma. La representación tradicional usada es el esquema binario en el cual un cromosoma es una cadena de la forma (a_1, a_2, \dots, a_k) , donde a_1, a_2, \dots, a_k se denominan *alelos* y éstos pueden tomar el valor de cero o uno.

Existen muchas razones por las cuales la codificación binaria es la más utilizada. La mayoría de ellas se refieren al trabajo desarrollado por Holland, ya que comparó dos representaciones diferentes que tuvieran aproximadamente la misma capacidad de acarreo de información, pero de ellas, una tenía pocos alelos y cadenas largas y la otra tenía un número grande de alelos y cadenas cortas. Holland argumentó que la primera opción permitía un grado más elevado de *paralelismo implícito* porque permite la creación de más *esquemas*. Un *esquema* es una plantilla que describe un subconjunto de cadenas que comparten ciertas similitudes en algunas posiciones a lo largo de su longitud. [Holland, 1992]

El hecho de tener más esquemas favorece la diversidad e incrementa la posibilidad de que se formen buenos bloques constructores (porción de un cromosoma que le produce una aptitud elevada a la cadena en la cual está presente) en cada generación, lo que en consecuencia mejora el desempeño del AG.

El *paralelismo implícito*, se refiere al hecho de que mientras el AG calcula las aptitudes de los individuos en una población, calcula de forma implícita las aptitudes promedio de un número mucho más alto de cadenas cromosómicas a través del cálculo de las aptitudes promedio observadas en los *bloques constructores* que se detectan en la población.

Por lo tanto, es preferible contar con muchos genes que tengan pocos alelos posibles que contar con pocos genes con muchos alelos posibles. Esto tiene una justificación biológica, ya que en genética es más usual tener cromosomas con muchas posiciones y pocos alelos por posición que pocas posiciones y muchos alelos por posición. [Coello, 1999]

2.4 Componentes Básicos.

Un AG, como se ha dicho antes, es un proceso probabilístico e iterativo que opera sobre un conjunto P de *individuos* (población), cada uno de los cuales contiene uno o más cromosomas. Dichos cromosomas permiten que cada individuo represente una posible solución al problema que se está considerando. Un proceso de codificación-decodificación (ρ) permite obtener la solución que los cromosomas de cada individuo contienen. Inicialmente, esta población es generada aleatoriamente.

Cada uno de los individuos de la población recibe, a través de una función de evaluación de la aptitud (*fitness* (F)), una medida de su comportamiento con respecto al problema a resolver. Este valor es utilizado por el algoritmo para guiar la búsqueda.

El proceso completo de un AG se muestra en la figura 3.

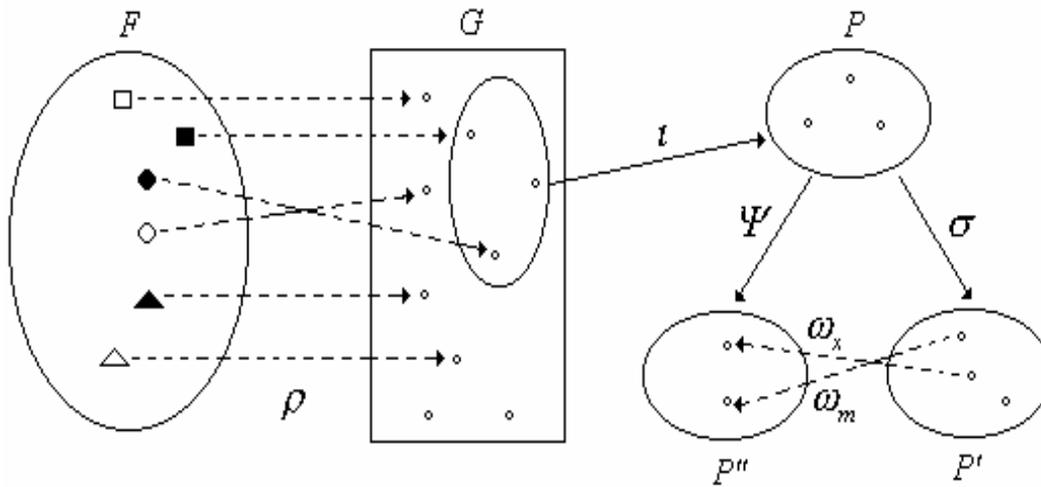


Figura 3. Proceso que efectúa un Algoritmo Genético

Como puede apreciarse, el algoritmo está estructurado en tres fases principales que se ejecutan de manera circular: selección (σ), reproducción (ω_x , ω_m) y el reemplazo (ψ), las cuales se llevan a cabo de manera repetitiva. Cada una de las iteraciones del algoritmo se denomina ciclo reproductivo básico o generación. Este proceso se realiza hasta que se alcanza un determinado criterio de terminación.

Durante la fase de selección se crea una población temporal P' en la que aquellos individuos más aptos (los correspondientes a las mejores soluciones contenidas en la población) estarán representados un mayor número de veces que los pocos aptos (*principio de selección natural*). A los individuos contenidos en esta población temporal se les aplican los llamados operadores reproductivos. El objetivo de esta fase es producir individuos con nuevas características, idealmente mejores (*principio de adaptación*).

Finalmente, durante la fase de reemplazo, se substituyen individuos de la población inicial por los nuevos individuos creados. Este reemplazo afecta a los peores individuos y tiende a conservar a los mejores (*supervivencia de los mejor adaptados*). Todo este proceso se sigue hasta alcanzar el objetivo deseado.

2.4.1 Elementos Fundamentales.

Un AG puede ser definido como una tupla **AG** $(\Sigma, \lambda, f, I, S, R, X, K_x, M, K_m, \tau)$, donde cada componente tiene el significado que a continuación se indica:

- ❖ Σ y λ son los elementos del alfabeto empleado para construir cromosomas y la longitud de los mismos. Dichos cromosomas forman el espacio de genotipos⁴. En su forma más convencional, el alfabeto es binario ($\Sigma = \{0, 1\}$).
- ❖ f es la función de evaluación que relaciona los cromosomas con el comportamiento de las soluciones que representan.
- ❖ I es el operador de inicialización, responsable de generar los individuos que formarán parte de la población inicial. Usualmente estos individuos se generan en forma aleatoria.
- ❖ S es el denominado operador de selección. Su misión es determinar cuáles de los cromosomas presentes en la población pasan a la fase reproductiva. Dicha selección está guiada por los valores de la evaluación de estos cromosomas, de manera que aquellos que tengan una evaluación mayor tendrán una mayor probabilidad de ser seleccionados.

⁴ El genotipo es la cadena cromosómica utilizada para almacenar la información contenida en un individuo.

Un mecanismo utilizado para realizar esta selección es el que está en función de la proporción de la contribución de cada individuo a la aptitud total de la población. Para esto se realiza una selección aleatoria en la que el individuo i -ésimo cuya evaluación es f_i y tiene una probabilidad p_i de ser seleccionado calculada como

$$p_i = \frac{f_i}{\sum_i f_i}$$

para toda $i = 1, \dots, N$

❖ R es el operador de reemplazo. Su misión es formar una nueva población a partir de la ya existente y de los individuos creados durante la fase reproductiva.

❖ X es el operador de cruce y K_x sus parámetros. La misión de este operador es producir soluciones a partir de las ya existentes. Este operador es una función de la forma

$$X : G^{k1} \times K_x \longrightarrow G^{k2}$$

Dónde G es el espacio de cromosomas y G^{k1} , G^{k2} son los cromosomas que se cruzaran entre sí.

❖ M es el operador de mutación y K_M su parámetro. La misión de este operador es producir nuevas soluciones a partir de la modificación de un cierto número de posiciones de una solución ya existente. La mutación se representa como

$$M : G \times K_M \longrightarrow G$$

En las versiones clásicas del operador, K_M indica la proporción de posiciones de la cadena que sufrirán modificación. Esta modificación es necesaria para introducir nuevo material genético a la población.

- ❖ τ es el criterio de terminación que determina cuándo se debe concluir la ejecución del algoritmo. Este puede ser:
 - ◆ Finalizar la ejecución tras realizar un número máximo de evaluaciones de la función objetivo.
 - ◆ Finalizar la ejecución cuando tras de un determinado número de iteraciones no se ha obtenido ninguna mejora global de la solución.
 - ◆ Finalizar la ejecución cuando se obtiene una solución cuya calidad se estima suficiente.

2.5 Generación.

Los mecanismos que ligán a un AG con el problema a resolver son: *la codificación del problema en un cromosoma y la función de evaluación*⁵. Esta última es esencial en procesos de optimización. La operación básica de un AG es muy simple: (a) Mantener una población de soluciones para un problema, (b) Seleccionar las mejores soluciones y recombinarlas entre sí, y (c) Usar a los progenitores para reemplazar aquellas soluciones menos convenientes.

El proceso de selección y combinación se lleva a cabo a través de reproducción sexual y mutación. Generalmente esto nos conduce a soluciones mejores que las originales, y frecuentemente a soluciones mejores que las propuestas por otros métodos. [Goldberg, 1999]

⁵ En general una función de evaluación tiene como entrada un cromosoma y como salida un número o una lista de números que representan el funcionamiento del cromosoma del problema a ser resuelto. [Goldberg, 1989]

Los pasos que realiza un algoritmo genético son:

1. Se genera un conjunto de soluciones válidas al problema. Cada una de estas entidades representa una solución distinta a un mismo problema. Estas entidades se pueden generar al azar. También se pueden generar a partir de soluciones ya conocidas del problema, que se pretendan mejorar, o mediante posibles "pedazos de soluciones" (más conocidos como bloques constructores). Es decir, con lo que creemos que pueden ser elementos componentes de la solución final aunque no sepamos cómo combinarlos.
2. Se evalúan las soluciones existentes, y se decide, en función de esta evaluación, dos cosas. Por una parte, cuáles soluciones van a sobrevivir y cuáles no; y por otra, cuáles se van a reproducir y cuáles no. En el caso de reproducirse, se especifica la capacidad reproductora de la solución, de forma que es posible decidir que unas soluciones se reproduzcan más que otras.
3. Tal como se ha establecido en el paso anterior, se eliminan ciertas soluciones y se mantienen otras, y se efectúa la reproducción o recombinación de genes (normalmente por parejas) de las entidades existentes. Por ejemplo, se realizan cruzas de patrones a partir de cierto punto elegido al azar, de forma que los nuevos patrones posean un segmento de cada uno de los progenitores.
4. Se efectúan mutaciones (cambios al azar en los genes) de los nuevos patrones, según una tasa determinada.

5. Regresa al paso 2 hasta que se cumpla el criterio de parada, que puede ser por ejemplo, que el peso o propiedad de la mejor entidad supere cierto valor. [Gascón, 1999]

Esto puede ejemplificarse mejor con el pseudocódigo fundamental de un AG:

```
BEGIN /*Comienzo del algoritmo genético */
    Generación de la población inicial
    Cómputo de la aptitud de cada individuo
WHILE NOT finalice DO
    BEGIN /*Produce una nueva generación */
    FOR (Tamaño de la población)/2 DO
        BEGIN /*Ciclo reproductivo */
        Selecciona dos individuos de la población anterior para realizar la
        cruza.
        /* Preferencia para los individuos de mayor aptitud */
        Se recombinan los dos individuos para generar dos descendientes.
        Cómputo de la aptitud de los descendientes.
        Se insertan los descendientes en la nueva población.
        END
        IF población converge THEN
        Finalice := TRUE
        END
    END
END
```

2.6 Cuándo Aplicar Un Algoritmo Genético.

La aplicación más común de los AGs ha sido la solución de problemas de optimización, en donde han mostrado ser muy eficientes y confiables. Sin embargo no todos los problemas pueden ser apropiados para esta técnica, por lo que se deben tomar en cuenta las siguientes características antes de usar este método de solución:

- ❖ El espacio de búsqueda (posibles soluciones) del problema a resolver debe estar delimitado dentro de un cierto rango.
- ❖ Debe poder definirse una *función de aptitud* que indique que tan buena o mala es una respuesta dada.
- ❖ Las soluciones deben codificarse de una forma que resulte relativamente fácil de implementar en una computadora.
- ❖ No debe existir un algoritmo determinístico o heurístico eficiente y confiable para resolver el problema.

El AG únicamente maximiza, pero la minimización puede realizarse utilizando el recíproco de la función de maximización. [Goldberg, 1999]

CAPITULO III

DISEÑO DEL ALGORITMO GENÉTICO PARA LA ASIGNACIÓN DE HORARIOS.

3.1 Introducción.

La elaboración de horarios consiste en asignar óptimamente tiempos y lugares a los diferentes cursos que son impartidos en una Institución Educativa, con el objetivo de satisfacer algunas restricciones importantes que se presenten, como son el número limitado de aulas, de laboratorios, de salas de cómputo, de espacios entre un curso y otro, etc. La principal restricción (central en toda asignación de horarios), es que no existan choques, es decir que dos o más cursos que esperan recibir los estudiantes o impartir los profesores no sean asignados simultáneamente.

3.2 Planteamiento de la solución a la asignación de horarios.

El elemento básico de cualquier asignación de horarios es un conjunto de cursos o materias $C = \{C_1, C_2, C_3, C_4, \dots, C_m\}$, donde cada elemento de C requiere que se le asigne un lugar y un tiempo.

Las aulas con que cuenta la Universidad pueden ser vistas como un conjunto de casillas que están disponibles las 7 horas al día (de las 08:00 a las 13:00 hrs y de las 16:00 a las 18:00 hrs) de Lunes a Viernes, esto da como resultado un total de $7n_{aulas}$ (donde n_{aulas} representa el número total de aulas) espacios disponibles para asignarles los cursos correspondientes. Suponiendo que cada curso tiene una lección diaria por cada día de la semana, tendríamos $1 \times 7n_{aulas}$ espacios disponibles, en el caso particular de la Universidad Tecnológica de la Mixteca esto se debe a que la lección del curso que se imparte el día Lunes a una determinada hora será el mismo que en los demás días de la semana.

De esta manera la calendarización de los m cursos es una función que mapea a cada curso con un espacio disponible.

Inicialmente se crea al azar cierto número de funciones. Cada una de estas funciones asigna al conjunto $\{C_1, C_2, C_3, \dots, C_m\}$ al codominio de espacios disponibles denominado $\{E_{0,1}, \dots, E_{7n_{aulas}}\}$. A estas funciones las llamaremos F_1, \dots, F_k , donde k es un número entero positivo.

Esto puede verse mejor en la siguiente figura:

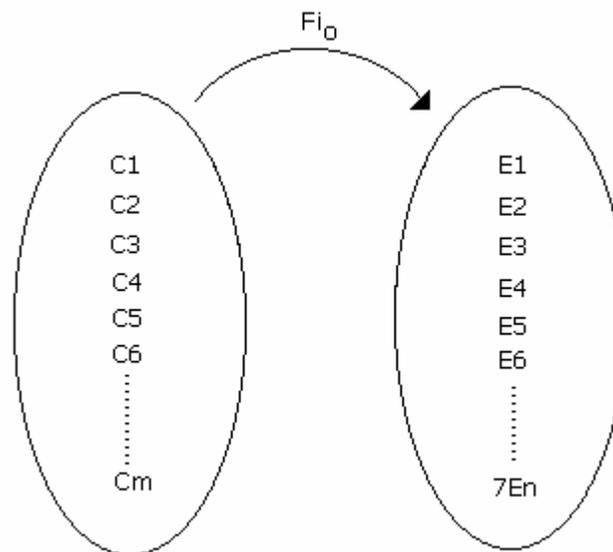


Figura 4. Correspondencia entre los cursos a impartir y los espacios disponibles, determinada por la función F_{i_0} para algún $i_0 \in \{1, \dots, k\}$.

Después de que se han creado las funciones se determina la aptitud de cada función, es decir, qué tan buena es la calendarización que cada función representa. Esto se hace de la siguiente forma: para cada función debemos de determinar cuantos choques les causa la calendarización a los estudiantes. Puesto que sabemos cuáles son los cursos que cada estudiante debe recibir, entonces tomamos los cursos que debe llevar cada estudiante y (digamos que estamos buscando la aptitud de la función F_1) determinamos si ese estudiante tiene algún conflicto de horario en relación con la calendarización especificada por F_1 . Por cada choque le restamos cierta cantidad a la aptitud de F_1 .

El cómputo de la aptitud de una función además incorpora restricciones como: que los cursos del área de Matemáticas y Física deben asignarse en las primeras horas, que dos o más cursos asignados a un estudiante no sean calendarizados a la misma hora, etc.

Así, que cada vez de que se encuentra que algún estudiante tiene asignadas materias de Matemáticas o Física en su horario en las primeras horas o que tiene muy pocos huecos entre cursos, entonces se da un bono a la función, por lo tanto la función que tenga la mayor aptitud será la que represente la mejor calendarización.

Cuando se ha calculado la aptitud de todas las funciones que se crearon al azar, entonces tenemos la primera población. (Los elementos de esta población, o entidades, son las funciones, y "estarán mas adaptadas al ambiente" cuanto más alta sea su aptitud). De esta población se hace el *mating pool*⁶ y posteriormente se realiza la reproducción sexual, esto se lleva a cabo tomando pares de funciones del *mating pool* y recombinándolas. Esto con el objetivo de crear dos nuevas funciones que se agregarán "en la próxima generación" de la población. Cuando ya se han creado suficientes funciones para la nueva generación, se les calcula sus aptitudes y se desecha la generación anterior. La nueva generación se convierte en la generación actual la cuál se usa para crear un *mating pool* y la historia se repite. El proceso termina hasta que por medio de reproducción sexual se obtiene alguna función con una aptitud suficientemente alta para satisfacer nuestras necesidades.

3.3 Implementación del Algoritmo Genético.

Se decidió utilizar un Algoritmo Genético para solucionar los problemas de asignación de horarios en la UTM, debido a que este método de optimización conjuga rapidez y eficiencia en su trabajo produciendo resultados aceptables, por eso es que esta técnica ha sido muy utilizada en problemas similares.

⁶ Se refiere a la selección de las funciones que serán los padres para la primer generación. [Holland,1992]

Actualmente, esta técnica de búsqueda ha alcanzado un impresionante éxito debido a que su aplicación en problemas de optimización no requiere de mucho conocimiento sobre el problema⁷.

3.3.1 Entorno de desarrollo.

Con los sistemas operativos actuales de ambiente visual, un programa debe tener la capacidad de interactuar gráficamente con la pantalla, el teclado, el ratón y la impresora, es por eso que el Sistema para la asignación de horarios, fue desarrollado en Visual Basic 6.0 ya que en la actualidad es uno de los lenguajes más accesibles a los usuarios por su entorno visual. Otra ventaja de Visual Basic es que sus aplicaciones corren sobre plataforma Windows, por lo que ofrece un alto grado de interacción con el usuario ya que utiliza elementos gráficos que forman los objetos que el usuario ve en su ventana de aplicación.

Otro punto importante a tomar en cuenta en la elección del entorno es que Visual Basic permite que los programas con él desarrollados sean de fácil mantenimiento.

⁷ Necesitamos conocer el problema para definir la función de evaluación, pero después, durante la búsqueda, sólo nos concentramos en los cromosomas.

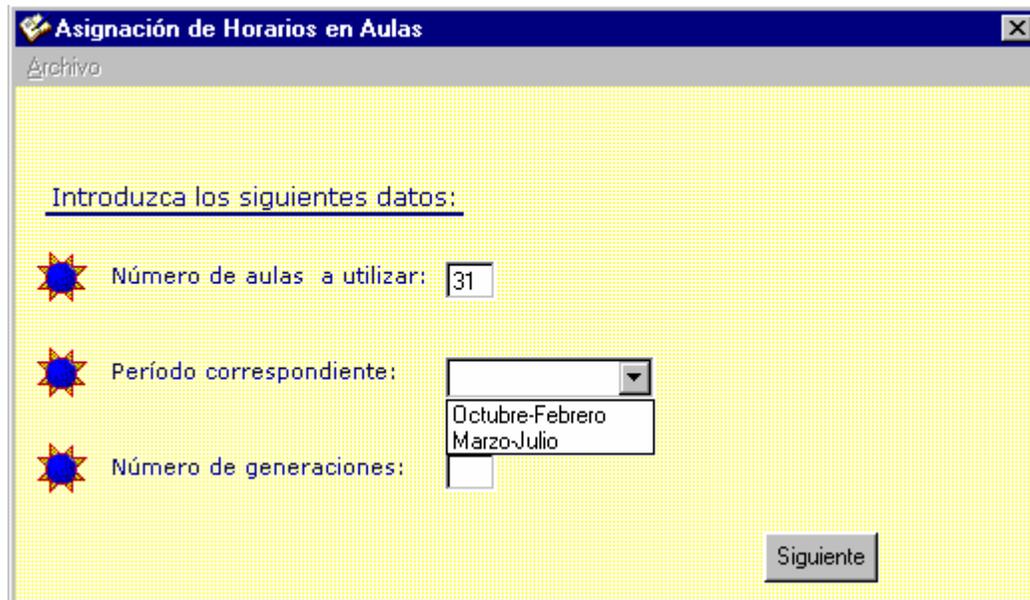
3.2.2 Interfaz del Sistema.

El Sistema para la Asignación de Horarios fue construido utilizando el lenguaje Visual Basic 6.0. El menú principal del sistema para la asignación de horarios consta de tres opciones: entrar al sistema, información sobre el mismo y salir a Windows. A continuación se muestra esta pantalla:



Figura 5. Interfaz del sistema para la asignación de horarios en la UTM.

Al hacer doble click sobre el icono de Entrar al Sistema aparece la siguiente pantalla donde se da opción al usuario de introducir el número de aulas a utilizar, el período que se va a abrir (ya que en el caso particular de la UTM, en el período de Octubre-Febrero se abren los semestres nones y en el período de Marzo-Julio los semestres pares), así como el número de generaciones a producir.



Asignación de Horarios en Aulas

Archivo

Introduzca los siguientes datos:

★ Número de aulas a utilizar:

★ Período correspondiente:

★ Número de generaciones:

Siguiete

Figura 6. Ventana para introducir datos referentes al número de aulas, periodo correspondiente y número de generaciones.

A continuación se muestra la pantalla en dónde se capturan los números de grupos por semestre según la carrera, esto con el fin de escribir las materias correspondientes en el archivo para su posterior lectura y asignación. (Sólo se muestra para una carrera, pero esta captura se hace para las siete carreras que se imparten en la UTM).

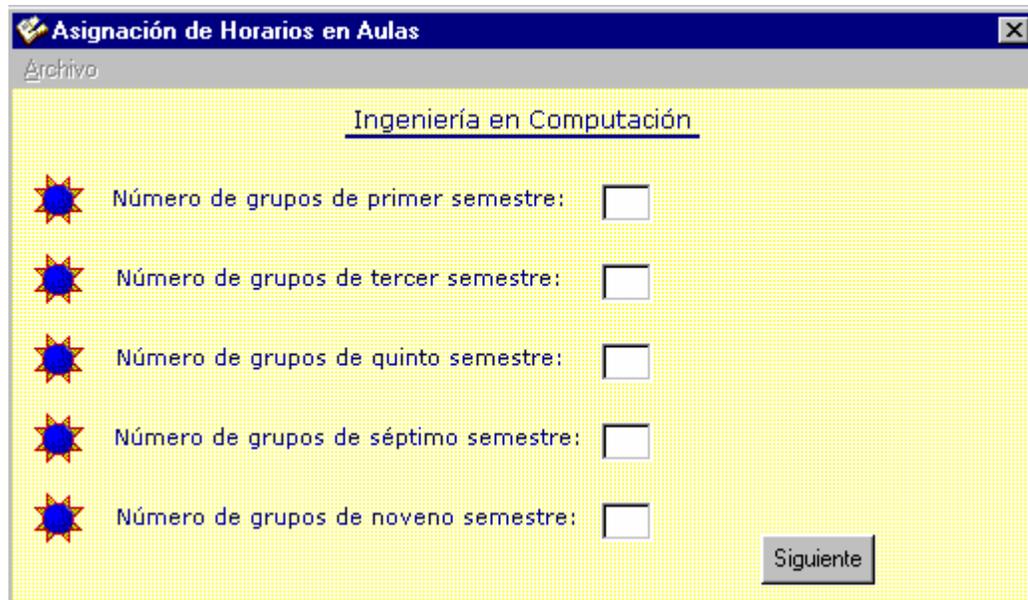
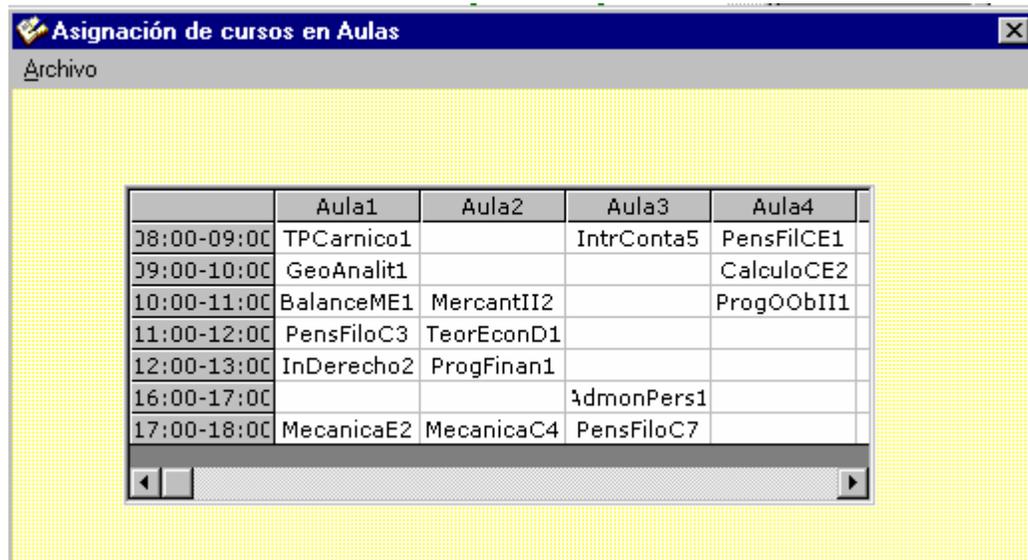


Figura 7. Captura del número de grupos por cada carrera.

Después de capturar los grupos de cada carrera y de la posterior escritura de las respectivas materias el número de veces igual al número de grupos, se genera aleatoriamente cierto número de funciones a las cuales se les asignan los cursos a impartir, esto también de manera aleatoria. Ya que las materias son asignadas aleatoriamente a las funciones, se realiza su evaluación, tomando en cuenta algunas restricciones como que las materias del área de Matemáticas y Física deben impartirse en las primeras horas, y la más importante: que dos o más cursos que va a cursar algún alumno no se encuentren asignados a la misma hora.

Después de hacer esta evaluación, las funciones cuya aptitud o comportamiento sea mejor son seleccionadas usando el criterio de la rueda de la ruleta (descrito en el capítulo 2). Las nuevas funciones generadas mediante la cruce pasan a formar la nueva población y la población anterior se desecha. Todo este procedimiento se realiza el número de veces que indique el usuario al capturar el número de generaciones o hasta que se encuentre una asignación cuya aptitud satisfaga nuestras condiciones.

Después de realizar este procedimiento se tendrá una calendarización que satisface nuestros requerimientos. A continuación se muestra una pantalla con el resultado de la asignación de cursos en aulas para un número de 5 generaciones:



The screenshot shows a window titled "Asignación de cursos en Aulas" with a menu bar containing "Archivo". The main area displays a table with the following data:

	Aula1	Aula2	Aula3	Aula4
08:00-09:00	TPCarnico1		IntrConta5	PensFilCE1
09:00-10:00	GeoAnalit1			CalculoCE2
10:00-11:00	BalanceME1	MercantII2		ProgOObII1
11:00-12:00	PensFiloC3	TeorEconD1		
12:00-13:00	InDerecho2	ProgFinan1		
16:00-17:00			AdmonPers1	
17:00-18:00	MecanicaE2	MecanicaC4	PensFiloC7	

Figura 8. Asignación resultante de cursos en aulas.

Cabe mencionar que para la asignación de cursos en los Laboratorios de Electrónica, se tuvo que implementar una función determinística⁸ debido a que, en la información proporcionada por el Departamento de Servicios Escolares de la UTM, son sólo dos materias (de séptimo semestre) las que se imparten en dichos laboratorios, por lo que la realización de un AG para asignar solo dos materias representaría un desperdicio de recursos. El resultado de esta función puede apreciarse en la siguiente pantalla.



The screenshot shows a window titled "Asignación de Horarios en Laboratorios de Electrónica" with a menu bar containing "Archivo". The main area displays a table with the following data:

	Lab1	Lab2
08:00-09:00		
09:00-10:00	Modulación	
10:00-11:00		
11:00-12:00		TeoControl
12:00-13:00		
16:00-17:00		
17:00-18:00		

Figura 8. Asignación resultante de cursos en laboratorios de electrónica.

⁸ Una función determinística indica en que lugar deben ser asignadas las materias, mediante el uso de condicionales.

CONCLUSIONES

La elaboración de horarios es un problema complejo, que aún no se ha podido resolver en los departamentos de control escolar de las Universidades. La asignación de los cursos por lo general es hecha a mano, lo cuál puede tardar desde días hasta semanas en hacer la repartición de los cursos, repartición que en muchos casos no es la mejor.

Existen otros métodos para resolver problemas de asignación de espacios, por ejemplo el recocido simulado, la búsqueda tabú, etc., algunos de los cuáles pueden ser rápidos en términos de encontrar buenas soluciones, sin embargo un AG se aproxima más a la asignación que es realizada manualmente, no obstante, cuando esta asignación es hecha a mano se puede tratar de balancear diferentes condiciones u objetivos, algunas de estas condiciones no pueden ser expresadas fácilmente en la función de evaluación. A pesar de esta desventaja un AG produce más de un resultado para una sola ejecución del mismo (sin incrementar el número de generaciones) a diferencia de los métodos arriba mencionados que producen un único resultado.

Por todo lo arriba mencionado es que muchos investigadores continúan la investigación de los Algoritmos Genéticos aplicados a esta área ya que estos son rápidos, eficientes y producen resultados aceptables en problemas de administración de espacios.

Al inicio de este trabajo se planteó el objetivo de crear una herramienta que fuera útil en la asignación de horarios en la Universidad Tecnológica de la Mixteca, utilizando para ello los Algoritmos Genéticos. Con la construcción del Sistema para la Asignación de Horarios, se considera que este objetivo fue alcanzado satisfactoriamente, aunque se presentó un problema de sobre-especificación en el caso de las salas de cómputo ya que las materias a impartir rebasan en mucho el número de espacios disponibles y esto es solucionado utilizando condiciones que no pudieron ser representadas en la función de evaluación, por lo que se decidió incluir estas materias en la asignación de cursos en aulas, dejando al criterio de la persona que realiza la asignación de horarios manualmente, la adaptación de los resultados de este sistema a sus necesidades.

Este trabajo de Tesis pretende ser una alternativa eficaz para realizar esta asignación, utilizando apropiadamente los Algoritmos Genéticos con punto de cruce. Cabe mencionar que en este trabajo se usaron parámetros reales, propios de una asignación (número de aulas, número de grupos de cada semestre por carrera, etc.), lo que hace que la asignación de cursos propuesta en este trabajo sea una mejor aproximación a una calendarización de cursos óptima que la que hasta ahora es hecha a mano en la Universidad Tecnológica de la Mixteca. Aunque posteriormente puede ser modificado agregando límites que reduzcan aún más el espacio de búsqueda de las posibles soluciones.

También se debe mencionar que para el caso de la asignación de cursos en los dos Laboratorios de Electrónica con que cuenta la UTM, se implemento una función determinística que evitó el desperdicio de recursos que hubiera causado la elaboración de un AG para asignar sólo dos materias a un espacio de búsqueda de 14 casillas disponibles.

El Algoritmo Genético implementado en este trabajo de Tesis, reduce eficazmente el tiempo que toma elaborar una correcta asignación de horarios, ya que una asignación hecha manualmente puede tomar desde días hasta semanas, lo cuál no es conveniente, sin embargo el AG propuesto solo tarda uno minutos en elaborar una asignación de cursos mejor que la que es realizada actualmente en la Universidad Tecnológica de la Mixteca.

Otra de las ventajas de utilizar el AG para la asignación de horarios, es que no se limita a un determinado número de aulas si no que proporciona una opción abierta para que dicho algoritmo se adapte al crecimiento de la infraestructura de dicha institución. La única condición para que el AG trabaje de manera eficiente es que exista una concordancia entre el número de aulas y el número de grupos que deben recibir los cursos correspondientes.

REFERENCIAS

Beasley, David; Bull, David R. & Martin, Ralph R. [1993], *An Overview of Genetic Algorithms: Part 1, Fundamentals*, University Computing, Pp. 58-69

Coello, Coello Carlos A. [Enero 1995], *Introducción a los Algoritmos Genéticos*, Soluciones Avanzadas, Tecnologías de Información y Estrategias de Negocios, Año 3, Número 17, Pp. 5-11.

Coello, Coello Carlos A. [Mayo 1999], *Representación en los Algoritmos Genéticos (Parte 1)*, Soluciones Avanzadas, Tecnologías de Información y Estrategias de Negocios, Año 7, Número 69, Pp. 50-56.

Gascón de la Herrán, Manuel. [1999], *Los largos ritmos frenéticos de la Computación Evolutiva*, GAIA,
http://www.geocities.com/SiliconValley/Vista/7491/alggen_c.htm#quees

Gascón de la Herrán, Manuel. [1999a], *Algoritmos Genéticos*, GAIA,
http://www.geocities.com/SiliconValley/Vista/7491/ag_c.htm

Goldberg, David E. [1989], *Genetic Algorithms in Search Optimization and Machine Learning*, Reading Massachusetts, Addison-Wesley Publishing Company.

Goldberg, David E. [Noviembre 1999], *Genetic Algorithms*, Illinois Genetic Algorithms Laboratory, <http://gal4.ge.uiuc.edu/>

Gutiérrez Miguel Angel; De los Cobos Sergio & Pérez Blanca. [1999], *Búsqueda Tabú: Un Procedimiento Heurístico para Solucionar Problemas de Optimización Combinatoria*, División CBI UAM Azcapotzalco e Iztapalapa, <http://www-azc.uam.mx/enlinea2/1-3.htm>

Holland, John H. [1975], *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, Massachusetts, Second Edition.

Holland, John. [Julio 1992], *Genetic Algorithms*, Scientific American, Volumen 267, Número 1, Pp. 66-72.

Kearfott, Baker R. [Junio 1998], Departamento de Matemáticas University of Southwestern Louisiana, <http://interval.louisiana.edu/GLOBSOL/Dian-approximate-optimizer/node2.html>

LBS Capital Management, Inc. [1999], *Genetic Algorithms*, http://www.lbs.com/about_lb/technolo/ga.html

Luenberger, David. [1989], *Programación lineal y no lineal*, Addison-Wesley Publishing, Pp. 3-6.

Morales, Eduardo. [Mayo 1999], *Búsqueda, optimización y aprendizaje*, Instituto Tecnológico de Estudios Superiores de Monterrey, <http://w3.mor.itesm.mx/~emorales/Cursos/Busqueda/principal.html>

Rao, Singiresu S. [1996], *Engineering Optimization, Theory and Practice*, John Wiley & Sons, Inc., Third Edition, Pp. 343,348.

APÉNDICE A

MANUAL PARA EL USUARIO DEL SISTEMA PARA LA ASIGNACIÓN DE HORARIOS.

Componentes del Algoritmo Genético para la asignación de horarios.

Las partes que conforman un AG, principalmente son:

- Creación de la población de cromosomas.
- Evaluación de la aptitud de cada cromosoma.
- Selección de los mejores elementos de la población.
- Cruza de los mejores cromosomas.

Requerimientos mínimos de operación.

- Procesador Pentium III.
- Windows 98 o superiores.
- 120 MB en RAM.
- Disco duro 4 GB.

Para el funcionamiento del sistema deben estar presentes los siguientes archivos en las ubicaciones especificadas:

C:\Algoritmo_Genetico\Algoritmo_Genetico.exe

C:\Algoritmo_Genetico\Forma1.frm

C:\Algoritmo_Genetico\Forma2.frm

C:\Algoritmo_Genetico\Forma3.frm

C:\Algoritmo_Genetico\Forma4.frm

C:\Algoritmo_Genetico\Forma5.frm

C:\Algoritmo_Genetico\Forma6.frm

C:\Algoritmo_Genetico\Forma7.frm

C:\Algoritmo_Genetico\Modulo1.bas

C:\Archivos de Programa\Microsoft Visual Studio\Vb98\Evaluacion.bas

C:\Archivos de Programa\Microsoft Visual Studio\Vb98\Modulo_archivo.bas

C:\Archivos de Programa\Microsoft Visual Studio\Vb98\Reproduccion.bas

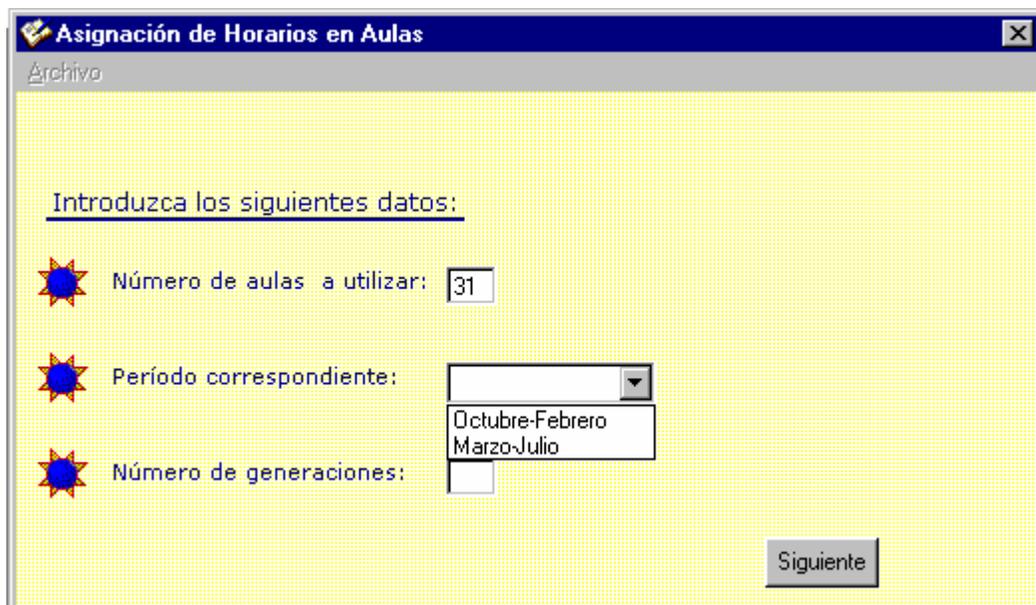
Todos estos archivos se encuentran contenidos en el disco flexible que acompaña a este documento.

Software del Algoritmo Genético para la asignación de horarios.

El programa principal es C:\Algoritmo_Genetico\Algoritmo_Genetico.exe, el cual puede ser ejecutado desde la línea de comandos o desde su icono en el escritorio de Windows.

Después de que el programa ha sido cargado, aparece la interfaz de sistema que se muestra en la figura 5 (página 35). Esta ventana contiene tres opciones: **entrar al sistema** donde al hacer un doble click muestra la ventana de la figura 6 (página 36), **acerca de** que proporciona una ventana donde se muestra información referente a la fecha de elaboración del AG así como su propósito y por último la **opción de salir a Windows**, para abandonar el sistema.

Al aparecer la siguiente pantalla (véase la figura al final de la página) se debe capturar el número de aulas a utilizar (por lo menos debe ser un aula), el período correspondiente (Octubre-Febrero ó Marzo-Julio) y el número de generaciones con las que trabajará el AG (mínimo deberá ser una generación para que el AG pueda trabajar).



Asignación de Horarios en Aulas

Archivo

Introduzca los siguientes datos:

★ Número de aulas a utilizar:

★ Período correspondiente:
Marzo-Julio

★ Número de generaciones:

Siguiete

Después aparece la pantalla de la figura 7 (página 37), que permite introducir el número de grupos por semestre ya previamente especificados para cada carrera que se imparte en la Universidad. Estos datos capturados permiten que las materias a cursar sean escritas - el número de veces igual al número de grupos introducidos - en el archivo **C:\materias.txt** esto para el caso del semestre Octubre-Febrero, y para el semestre Marzo-Julio las materias son escritas en el archivo **C:\materias1.txt**. Los datos introducidos para el número de grupos por carrera deberán ser únicamente enteros positivos.

La información contenida en estos archivos permite que se lleve a cabo la asignación de cursos en los cromosomas, parte esencial para que trabaje correctamente el AG. Así mismo los resultados del Algoritmo Genético variarán de acuerdo al número de generaciones introducido. Para salir del sistema solo se deberá elegir la opción de Archivo y posteriormente la opción Salir. La opción Archivo también cuenta con la opción de Imprimir.

Por último es importante explicar las abreviaturas, de los nombres de las materias, que se utilizaron para la realización del Algoritmo Genético, mismas que aparecen en la asignación de horarios resultante. A continuación se proporciona la lista de dichas abreviaturas, dónde la última letra representa la carrera a la que corresponde y el último número indica el grupo al que se asignó dicha materia, por ejemplo:

“ProgOOBII3” nos indica que Programación Orientada a Objetos II se debe impartir para el grupo 3 de séptimo semestre de computación, aquí no se especifica para que carrera deberá impartirse la materia ya que esta es propia de computación, a diferencia de “SistDigiC” o “SistDigiE” dónde se especifica para que carrera impartir dicha materia ya que la toman tanto computación como electrónica.

Lista de las Materias de Ingeniería en Computación:

- "MecanicaC" = Mecánica Clásica
- "Calculo C" = Cálculo
- "PensFiloC" = Historia del Pensamiento Filosófico
- "ProgEstrC" = Programación Estructurada
- "IntIngCom" = Introducción a la Ingeniería en Computación
- "ElectrosC" = Electroestática
- "AlgLineaC" = Análisis Vectorial y Álgebra Lineal
- "TeorGenSC" = Teoría General de Sistemas
- "EstDatosC" = Estructura de Datos
- "SistCompC" = Construcción de Sistemas de Computación
- "ElectDinC" = Electrodinámica
- "EcuacDifC" = Ecuaciones Diferenciales
- "TeoMatCom" = Teoría Matemática de la Computación
- "CircuitEC" = Circuitos Eléctricos
- "FisOndulC" = Física Ondulatoria
- "MetodNumC" = Métodos Numéricos
- "ProgSistC" = Programación de Sistemas
- "ProbyEstC" = Probabilidad y Estadística
- "TeorEconC" = Teoría Económica
- "SistOperC" = Sistemas Operativos
- "TeorComuC" = Teoría de Comunicaciones
- "DisCircuC" = Diseño de Circuitos Electrónicos
- "CircuitoC" = Circuitos Lógicos
- "AdministC" = Administración
- "CompiladC" = Compiladores
- "ProgOObjI" = Programación Orientada a Objetos I
- "SistComuC" = Sistemas de Comunicaciones

"ContabilC" = Contabilidad
"IngSoftwC" = Ingeniería de Software
"BasDatosI" = Bases de Datos I
"ProgOObII" = Programación Orientada a Objetos II
"SistDigiC" = Sistemas Digitales
"EstAdmvsC" = Análisis y Diseño de Estructuras Administrativas
"TeoAlgorT" = Teoría de Algoritmos
"BsDatosII" = Bases de Datos II
"MicroComC" = Arquitectura de Microcomputadoras
"IntArtifiC" = Inteligencia Artificial
"DisProgAd" = Diseño de Programas Administrativos
"ArqCompuC" = Arquitectura de Computadoras
"MercadotC" = Mercadotecnia
"DiProgFin" = Diseño de Programas Financieros
"SExpertol" = Sistemas Expertos I
"OCTInforma" = Organización de Centros de Teleproceso e Informática
"ProgTecno" = Diseño de Programas Tecnológicos
"InvOperaC" = Investigación de Operaciones
"DireccioC" = Dirección
"SExpertII" = Sistemas Expertos II

Lista de las Materias de Ingeniería en Electrónica:

"MecanicaE" = Mecánica Clásica
"Calculo E" = Cálculo
"PensFiloE" = Historia del Pensamiento Filosófico
"ProgEstrE" = Programación Estructurada
"IntIngEle" = Introducción a la Ingeniería en Electrónica
"ElectrosE" = Electroestática

"AlgLineaE" = Análisis Vectorial y Álgebra Lineal
"TeorGenSE" = Teoría General de Sistemas
"EstDatosE" = Estructura de Datos
"EquipElec" = Construcción de Equipo Electrónico
"ElectDinE" = Electrodinámica
"EcuacDifE" = Ecuaciones Diferenciales
"MetyTrans" = Metrología y Transductores
"CircuitEE" = Circuitos Eléctricos
"FisOnduleE" = Física Ondulatoria
"MetodNumE" = Métodos Numéricos
"ProgSistE" = Programación de Sistemas
"ProbyEstE" = Probabilidad y Estadística
"TeorEconE" = Teoría Económica
"SistOperE" = Sistemas Operativos
"TeorComuE" = Teoría de Comunicaciones
"DisCircuE" = Diseño de Circuitos Electrónicos
"CircuitoE" = Circuitos Lógicos
"AdministE" = Administración
"CompiladE" = Compiladores
"CPotencia" = Diseño de Circuitos Electrónicos de Potencia
"SistComuE" = Sistemas de Comunicaciones
"ContabileE" = Contabilidad
"SistDigiE" = Sistemas Digitales
"IngSoftwE" = Ingeniería de Software
"Modulacion" = Modulación y Control
"TeoControl" = Teoría del Control
"EstAdmvsE" = Análisis y Diseño de Estructuras Administrativas
"PDSeñales" = Procesamiento Digital de Señales
"Microonda" = Sistemas de Comunicaciones vía Microondas

"MicroComE" = Arquitectura de Microcomputadoras
"ControlDg" = Control Digital
"CSatelite " = Sistemas de Comunicaciones vía Satélite
"MercadotE" = Mercadotecnia
"SCFOptica" = Sistemas de Comunicaciones vía Fibra Óptica
"ArqCompuE" = Arquitectura de Computadoras
"IntArtifE" = Inteligencia Artificial
"Robotica " = Robótica
"InvOperaE" = Investigación de Operaciones
"DireccioE" = Dirección
"RedesComp" = Redes de Computadoras
"SExpertos" = Sistemas Expertos

Lista de las Materias de Ingeniería en Diseño:

"MecanicaD" = Mecánica Clásica
"Calculo D" = Cálculo
"PensFiloD" = Historia del Pensamiento Filosófico
"ProgEstrD" = Programación Estructurada
"IntIngDis" = Introducción a la Ingeniería en Diseño
"ElectrosD" = Electroestática
"AlgLineaD" = Análisis Vectorial y Álgebra Lineal
"TeorGenSD" = Teoría General de Sistemas
"EstDatosD" = Estructura de Datos
"DibTecnil" = Dibujo Técnico I
"ElectDinD" = Electrodinámica
"EcuacDifD" = Ecuaciones Diferenciales
"MetDiseño" = Métodos de Diseño

"GeDescrip" = Geometría Descriptiva
"EstaticaD" = Estática
"FisOndulD" = Física Ondulatoria
"MetodNumD" = Métodos Numéricos
"RMaterial" = Resistencia de Materiales
"GeCuerpos" = Geometría de los Cuerpos
"ProbyEstD" = Probabilidad y Estadística
"TeorEconD" = Teoría Económica
"Ergonomia" = Ergonomía
"TecRepres" = Técnica de Representación
"MatyProcl" = Materiales y Procesos I
"Composici" = Composición
"AdministD" = Administración
"EstudForm" = Estudio de la Forma
"Modelos" = Modelos
"ProyEspacial" = Proyecto Espacial
"ProcesoII" = Materiales y Procesos II
"ContabilD" = Contabilidad
"TecIllust" = Técnicas de Ilustración
"MatyProce" = Materiales y Procesos
"DisInteri" = Diseño Interior
"DisEditor" = Diseño Editorial
"EstAdmvsD" = Análisis y Diseño de Estructuras Administrativas
"MediosAud" = Medios Audiovisuales
"Plasticos" = Materiales y Procesos. Plásticos
"Estetica" = Estética
"AplicGraf" = Aplicaciones Gráficas por Computadora
"EcoUrbana" = Ecología Urbana
"InvyProto" = Investigación y Prototipos

"MercadotD" = Mercadotecnia

"DesDeProd" = Desarrollo de Producto

"ProyCompt" = Proyecto por Computadora

"Vidrios " = Materiales y Procesos. Vidrios

"DesProducto" = Desarrollo de Producto

"Comercial" = Producción y Comercialización

"DireccioD" = Dirección

"DiseñoUyA" = Diseño Urbano y Ambiental

"TProdyDis" = Técnicas Avanzadas de Producción y Diseño

Lista de las Materias de Ingeniería en Alimentos:

"MecanicaA" = Mecánica Clásica

"Calculo A" = Cálculo

"PensFiloA" = Historia del Pensamiento Filosófico

"ProgEstrA" = Programación Estructurada

"QuimiGral" = Química General

"AlgLineaA" = Análisis Vectorial y Álgebra Lineal

"IntIngAli" = Introducción a la Ingeniería en Alimentos

"EstDatosA" = Estructura de Datos

"TeoGenSA" = Teoría General de Sistemas

"Organical" = Química Orgánica I

"EcuacDifA" = Ecuaciones Diferenciales

"BalanceME" = Balance de Materia y Energía

"QuimOrgII" = Química Orgánica II

"BioQuimil" = Bioquímica I

"MicroGral" = Microbiología General

"MetodNumA" = Métodos Numéricos

"FisicoQuI" = Físicoquímica I
"QuimAlimt" = Química de Alimentos
"BioQuimiII" = Bioquímica II
"MicroAlim" = Microbiología de Alimentos
"ProbyEstA" = Probabilidad y Estadística
"TeorEconA" = Teoría Económica
"FisicoQII" = Físicoquímica II
"FenTransI" = Fenómenos de Transporte I
"QuimAnali" = Química Analítica
"Nutricion" = Nutrición
"AdministA" = Administración
"AnalisExp" = Diseño y Análisis de Experimentos
"FOAliment" = Físicoquímica de Alimentos
"FenTranII" = Fenómenos de Transporte II
"TecFrutas" = Tecnología de Frutas y Hortalizas
"AnalisisI" = Análisis Instrumental
"ContabilA" = Contabilidad
"IngAlimel" = Ingeniería en Alimentos I
"TCereales" = Tecnología de Cereales
"TPCarnico" = Tecnología de Productos Cárnicos
"DibIngAli" = Dibujo de Ingeniería en Alimentos
"AnalisisA" = Análisis de Alimentos
"EstAdmvsA" = Análisis y Diseño de Estructuras Administrativas
"IngAlimII" = Ingeniería en Alimentos II
"TPLacteos" = Tecnología de Productos Lácteos
"IngBioQuim" = Ingeniería Bioquímica
"DesProduc" = Desarrollo de nuevos Productos
"Sensorial" = Evaluación Sensorial
"MercadotA" = Mercadotecnia

"IngAliIII" = Ingeniería en Alimentos III
"FEvalProy" = Formulación y Evaluación de Proyectos
"EmpaqueyE " = Empaque y Embalaje
"ContDeCal" = Control de Calidad
"DireccioA" = Dirección
"InvOperaA" = Investigación de Operaciones
"TecBebida" = Tecnología de Bebidas
"PlantasAl" = Desarrollo y Diseño de Plantas Alimentarias
"SemiTesis" = Seminario de Tesis

Lista de las Materias de Licenciatura en Matemáticas:

"CalculoMI" = Cálculo I
"PensFiloM" = Historia del Pensamiento Filosófico
"AlgSuperi" = Álgebra Superior
"GeoAnalit" = Geometría Analítica
"AlgLineal" = Álgebra Lineal II
"AnálisisI" = Análisis Vectorial I
"CalculoII" = Cálculo II
"TeoGraISM" = Teoría General de Sistemas
"ProgEstrM" = Programación Estructurada
"AlgLineII" = Álgebra Lineal II
"AnálisisII" = Análisis Vectorial II
"EcuadifMI" = Ecuaciones Diferenciales I
"EstDatosM" = Estructura de Datos
"AMatematI" = Análisis Matemático
"ComplejaI" = Variable Compleja I
"AlgModerI" = Álgebra Moderna I

"EcuDifMII" = Ecuaciones Diferenciales II
"AMatemaII" = Análisis Matemático II
"ComplejII" = Variable Compleja II
"AlgModeII" = Álgebra Moderna II
"MFClasica" = Modelos de la Física Clásica
"AnalisiNI" = Análisis Numérico
"ProbabilI" = Probabilidad I
"AnalisiNII" = Análisis Numérico II
"MDiscreta" = Matemáticas Discretas
"MetOptimi" = Modelos y Métodos de Optimización I
"MFModerna" = Modelos de la Física Moderna
"ProbabilII" = Probabilidad II
"MOptimiII" = Modelos y Métodos de Optimización II
"ProgLogic" = Programación Lógica
"CalcVaria" = Cálculo de Variaciones y Teoría de Control
"Algoritmo" = Análisis de Algoritmos
"EstadistI" = Estadística I
"IntArtifi" = Inteligencia Artificial
"TopologiI" = Topología I
"Geneticos" = Algoritmos Genéticos
"EstadisII" = Estadística II
"TopologII" = Topología II
"SeTesisMI" = Seminario de Tesis I
"SExpertoM" = Sistemas Expertos
"RedesNeur" = Redes Neuronales
"STesisMII" = Seminario de Tesis II
"EconomíaM" = Economía Matemática

Lista de las Materias de Licenciatura en Ciencias Empresariales:

"CalculoCE" = Cálculo
"PensFilCE" = Historia del Pensamiento Filosófico
"IntrAdmon" = Introducción a la Administración
"IntrConta" = Introducción a la Contabilidad
"ProgEstCE" = Programación Estructurada
"AlgLineCE" = Álgebra Lineal
"Psicologi" = Psicología Industrial
"TeoGralCE" = Teoría General de Sistemas
"BsDatosCE" = Bases de Datos
"ContabiliI" = Contabilidad I
"ProbabiCE" = Probabilidad y Estadística
"AdmonPers" = Administración de Personal
"ProgConta" = Programas de Contabilidad
"ContabiliII" = Contabilidad II
"InDerecho" = Introducción a Derecho
"AdSueldos" = Administración de Sueldos
"ProgAdmvo" = Programas Administrativos
"ContCosto" = Contabilidad de Costos
"Mercadote" = Fundamentos de Mercadotecnia
"MercantilI" = Derecho Mercantil I
"RecursosM" = Recursos Materiales
"ProgFinan" = Programas Financieros
"FinanzaEI" = Finanzas Empresariales I
"MicroEcon" = Microeconomía
"MercantII" = Derecho Mercantil II
"Auditoria" = Introducción a la Auditoría
"FinanzasP" = Finanzas Públicas

"FinanzEII" = Finanzas Empresariales II
"MacroEcon" = Macroeconomía
"AuditoriA" = Auditoría Administrativa
"MercadosI" = Administración de Mercados I
"DerFiscal" = Derecho Fiscal
"SistFinan" = Sistemas Financieros
"EconomíaI" = Economía Internacional
"InvOperCE" = Investigación de Operaciones
"MercadoII" = Administración de Mercados II
"Presupues" = Presupuestos
"EPublicit" = Estrategia Publicitaria y Promocional
"EconomiaM" = Economía de México
"MetInvest" = Metodología de la Investigación
"MercadosF" = Modelación Matemática de Mercados Financieros
"AdministF" = Administración Financiera
"SEmpresal" = Seminario de Simulación Empresarial I
"EProyecto" = Evaluación de Proyectos
"DesOrgani" = Desarrollo Organizacional
"AnalisisD" = Análisis de Decisiones
"DecisionF" = Toma de Decisiones Financieras
"SEmpresII" = Seminario de Simulación Empresarial II
"PEstrateg" = Planeación Estratégica

Lista de las Materias de Ingeniería Industrial:

"Mecanical" = Mecánica Clásica
"Calculo I" = Cálculo
"PensFilol" = Historia del Pensamiento Filosófico
"ProgEstrI" = Programación Estructurada

"IntIngInd" = Introducción a la Ingeniería Industrial
"DibTecnil" = Dibujo Técnico Industrial
"ElectMagn" = Electromagnetismo
"EstDatosI" = Estructura de Datos
"TeoGenSI" = Teoría General de Sistemas
"AlgLinInd" = Análisis Vectorial y Álgebra Lineal
"DibIndCom" = Dibujo Industrial Asistido por Computadora
"AdmonGral" = Administración General
"EcuacDifI" = Ecuaciones Diferenciales
"QuimicaGr" = Química General
"SistInfor" = Sistemas de Información
"HerraComp" = Herramientas de Computación para Ingenieros
"MetodNumI" = Métodos Numéricos
"AdmHumano" = Administración del Factor Humano
"TermoDinI" = Termodinámica
"MercadotI" = Mercadotecnia
"DerLabora" = Derecho Laboral y Propiedad Industrial
"ProbyEstI" = Probabilidad y Estadística
"TeorEconI" = Teoría Económica
"ContaGral" = Contabilidad General
"CiencMate" = Ciencia de los Materiales
"Estatical" = Estática
"ContCostI" = Contabilidad de Costos
"Productivi" = Estudio del Trabajo y Productividad
"ResistMat" = Resistencia de Materiales
"MecFluido" = Mecánica de Fluidos
"Electroni" = Electrónica
"AdmonRecs" = Administración de Recursos Materiales
"Ergonomil" = Ergonomía

"SistyCont" = Sistemas y Control de la Calidad
"EquipoInd" = Maquinaria y Equipo Industrial
"InstyMetr" = Metrología e Instrumentación
"SegIndust" = Medio Ambiente y Seguridad Industrial
"Presupuel" = Presupuestos
"IngEconom" = Ingeniería Económica
"InvOperal" = Investigación de Operaciones I
"ManufactI" = Procesos de Manufactura
"EnvyEmbal" = Envases y Embalajes
"ImpoyExpo" = Procesos de Importación y Exportación
"Proyectos" = Formulación y Evaluación Económica de Proyectos
"Planeacio" = Planeación y Control de la Producción
"InvOperII" = Investigación de Operaciones II
"SistManuf" = Sistemas Integrados de Manufactura
"PlantaInd" = Diseño y Construcción de Plantas Industriales
"TemasSell " = Temas Selectos de Ingeniería Industrial I
"DiryTomaD" = Dirección y Toma de Decisiones
"MInvestig" = Metodología de la Investigación
"SimIndust" = Análisis y Simulación Industrial
"AutomaInd" = Control y Automatización Industrial
"IngPlanta" = Ingeniería de Planta
"TemasSellI" = Temas Selectos de Ingeniería Industrial II

APÉNDICE B

CÓDIGO DOCUMENTADO DE LOS PROCEDIMIENTOS Y FUNCIONES MÁS IMPORTANTES DEL SISTEMA.

Programa para la asignación de horarios.

Mediante este programa fue implementado el AG que realiza la asignación de cursos en la Universidad, este programa fue codificado en Visual Basic 6.0 y el código de las funciones más importantes se muestra a continuación.

```
/* Procedimiento que escribe las materias a asignar en los  
archivos */
```

```
Sub matcomp1()  
If valorC1 <> 0 Then  
For NúmeroRegistro = 1 To valorC1
```

```

MiRegistro.Materia1 = "MecanicaC" & NúmeroRegistro ' Crea una cadena
MiRegistro.Materia2 = "Calculo C" & NúmeroRegistro ' Crea una cadena
MiRegistro.Materia3 = "PensFiloC" & NúmeroRegistro ' Crea una cadena
MiRegistro.Materia4 = "IntIngCom" & NúmeroRegistro ' Crea una cadena
Put #1, NúmeroRegistro, MiRegistro ' Escribe el registro en el archivo
Next NúmeroRegistro
End If
End Sub

```

Este procedimiento se repite para los diez semestres correspondientes a cada carrera (ya que conocemos las materias en las que se matricularán los estudiantes).

Después de que las materias son escritas en el archivo correspondiente, se realiza su lectura, el código de esta función se describe a continuación:

/*Procedimiento que realiza la lectura del archivo que contiene las materias a asignar*/

```

Public Sub lectura()
NúmeroRegistro1 = NúmeroRegistro - 1 'se ubica al final del archivo
For NúmeroRegistro = 1 To NúmeroRegistro1 ' se posiciona al inicio
del archivo
    Call num_azar(val) ' rutina que busca una posición al azar en
las funciones
    Get #1, NúmeroRegistro, MiRegistro ' Lee la materia del
archivo
    valmat1 = MiRegistro.Materia1
    Call asignacionmateria1 'Asigna la materia a la función en la
posición indicada
    Call num_azar(val)

```

```
    Get #1, NúmeroRegistro, MiRegistro
    valmat2 = MiRegistro.Materia2
    Call asignacionmateria2
    Call num_azar(val)
    Get #1, NúmeroRegistro, MiRegistro
    valmat3 = MiRegistro.Materia3
    Call asignacionmateria3
    Call num_azar(val)
    Get #1, NúmeroRegistro, MiRegistro
    valmat4 = MiRegistro.Materia4
    Call asignacionmateria4
    Call num_azar(val)
    Get #1, NúmeroRegistro, MiRegistro
    valmat5 = MiRegistro.Materia5
    Call asignacionmateria5
    Call num_azar(val)
    Get #1, NúmeroRegistro, MiRegistro
    valmat6 = MiRegistro.Materia6
    Call asignacionmateria6
  Next NúmeroRegistro
End Sub
```

Posteriormente se realiza la evaluación de las funciones, el código de esta subrutina se muestra a continuación:

/*Evalúa el contenido de cada elemento de la población*/

Public Sub evaluacion_poblacion()

 'Analiza el contenido del cromosoma_n

 Call analisis_contenido(cursos_n, calif_cursosn) *'Analiza que no estén asignadas dos o más materias para un mismo grupo a la misma hora*

 Call mat_mate1(cursos_n) *'Procedimiento que asigna bonos a los cromosomas que tengan asignadas materias de matemáticas de primer semestre en las dos primeras horas de clases*

 Call mat_mate3(cursos_n) *'Procedimiento que asigna bonos a los cromosomas que tengan asignadas materias de matemáticas de tercer semestre en las dos primeras horas de clases*

 Call mat_mate5(cursos_n) *'Procedimiento que asigna bonos a los cromosomas que tengan asignadas materias de matemáticas de quinto semestre en las dos primeras horas de clases*

 calif_cursosn = calificacion *'Se asigna la calificación obtenida después de evaluar el cromosoma*

End Sub

/*Procedimiento que evalúa que las materias asignadas en los cromosomas no estén repetidas ni que para el mismo grupo estén asignadas dos o más materias a la misma hora*/

```
Public Sub analisis_contenido(num_curso, calif_curso As Integer)
```

```
    calificacion = 0
```

```
    bandera = 1
```

```
    'Realiza una copia del contenido del cromosoma
```

```
    For numindice = 1 To val
```

```
        temporal(numindice) = num_curso(numindice)
```

```
    Next numindice
```

```
    'Revisa que materia esta asignada hora por hora en cada aula
```

```
    For i = 1 To 7
```

```
        For j = 8 To val
```

```
            If bandera = 1 Then
```

```
                j = 8
```

```
                bandera = bandera + 1
```

```
            ElseIf (bandera = 3) Then
```

```
                j = j + 1
```

```
                bandera = bandera + 1
```

```
            ElseIf (bandera = 5) Then
```

```
                j = j + 2
```

```
                bandera = bandera + 1
```

```
            ElseIf (bandera = 7) Then
```

```
                j = j + 3
```

```
                bandera = bandera + 1
```

```
            ElseIf (bandera = 9) Then
```

```
                j = j + 4
```

```
                bandera = bandera + 1
```

```
Elseif (bandera = 11) Then
    j = j + 5
    bandera = bandera + 1
Elseif (bandera = 13) Then
    j = j + 6
    bandera = bandera + 1
End If
a = calif_primersem(calif_curso) 'Analiza las materias del primer semestre'
c = calif_tercersem(a) 'Analiza las materias del tercer semestre'
e = calif_quintosem(c) 'Analiza las materias del quinto semestre'
g = calif_septimosem(e) 'Analiza las materias del séptimo semestre'
q = calif_novenosem(g) 'Analiza las materias del noveno semestre'
calificacion = q
j = j + 6
Next j
bandera = bandera + 1
Next i
/* analiza que no se repitan las materias, si esto ocurre se le resta cierta cantidad a la aptitud del cromosoma */
For i = 1 To (val - 1)
    If (temporal(i) <> "" And temporal(i + 1) <> "") Then
        If (temporal(i) = temporal(i + 1)) Then
            calificacion = calificacion - 5
        End If
    End If
End For
Next i
End Sub
```

/*Función que revisa materia por materia de primer semestre asignada a cada aula hora por hora*/

Public Function calif_primersem(calif_arreglo As Integer)

a8 = 50 *'Asigna una calificación base a cada cromosoma a evaluar*

'Llamada al procedimiento que evalúa que no se produzcan

'choques entre las materias de Ingeniería en Computación del primer semestre

a1 = evaluacion_computacion1(a8)

'Llamada al procedimiento que evalúa que no se produzcan

'choques entre las materias de Ingeniería en Electrónica del primer semestre

a2 = evaluacion_electronica1(a1)

'Llamada al procedimiento que evalúa que no se produzcan

'choques entre las materias de Ingeniería en Diseño del primer semestre

a3 = evaluacion_diseño1(a2)

'Llamada al procedimiento que evalúa que no se produzcan

'choques entre las materias de Ingeniería en Alimentos del primer semestre

a4 = evaluacion_alimentos1(a3)

'Llamada al procedimiento que evalúa que no se produzcan

'choques entre las materias de Licenciatura en Matemáticas del primer semestre

a5 = evaluacion_matematicas1(a4)

```

'Llamada al procedimiento que evalúa que no se produzcan
'choques entre las materias de Licenciatura en Ciencias Empresariales del
'primer semestre
a6 = evaluacion_ciencias1(a5)

```

```

'Llamada al procedimiento que evalúa que no se produzcan
'choques entre las materias de Ingeniería Industrial del
'primer semestre
a7 = evaluacion_industrial1(a6)

```

```

calif_primersem = a7
End Function

```

Esta función se repite para los diez semestres de cada carrera. Después de hacer esta evaluación, las funciones cuya aptitud o comportamiento sea mejor son seleccionadas usando el criterio de la rueda de la ruleta (descrito en el capítulo 2).

La selección fue implementada de la siguiente manera:

```

/*Procedimiento que realiza la reproducción de los cromosomas
seleccionados*/

```

```

Public Sub selección()
w = 0
z = 0
y = 0
Do While (w < numpob-1) 'ciclo que recorre todos los elementos de la
población
v = selec() ' llamada a la función que determina los cromosomas que se
reproducirán

```

```

v1 = selec()
Call copia_padre1 'procedimiento que realiza una copia de los cromosomas
seleccionados como padres de la nueva generación
Call copia_padre2
Call cruza(progenitor1, progenitor2) 'procedimiento que realiza la cruza
entre dos cromosomas
Call nueva_poblacion 'procedimiento que elimina la población anterior y
guarda la nueva
w = w + 2
Loop
End Sub

```

En donde **select** es la función que selecciona las funciones según su aptitud:

```

/*Función que realiza la selección de los mejores cromosomas
según el valor de su aptitud, siguiendo el criterio de la rueda de la
ruleta*/

```

```

Public Function selec()
Call calificaciones 'procedimiento que realiza la suma de las evaluaciones de
los cromosomas
suma = 0
i = 0
aleatorio = Int((evaluación * Rnd) + 1) 'genera un número al azar
dependiendo de la longitud del cromosoma
Do
    i = i + 1

```

```

suma = suma + evaluaciones(i)
Loop Until suma >= aleatorio Or i = 40 'ciclo que se efectúa hasta que la
suma de las calificaciones sea igual o mayor al punto escogido al azar

selec = i 'devuelve el cromosoma cuya aptitud permitió que la suma de las
calificaciones sea igual o mayor al punto escogido al azar

End Function

```

Las cadenas seleccionadas son utilizadas por el operador de cruza, que se implementó de la siguiente forma, siguiendo el criterio del punto de cruza (descrito en el capítulo 2):

/*Procedimiento que realiza la cruza de parejas de cromosomas con mejor aptitud*/

```

Public Sub cruza(padre1, padre2)
Randomize
punto = Int((val * Rnd) + 1) 'determina al azar el punto de cruza
If punto < val Then
    For i = 1 To punto 'mientras el punto escogido sea menor que la longitud
del cromosoma se hace una copia de los padres en los hijos desde la
posición inicial hasta el punto de cruza
        hijo1(i) = padre1(i)
        hijo2(i) = padre2(i)
    Next i
    For i = punto + 1 To val ' se hace una copia de los padres en los hijos
desde el punto de cruza hasta la ultima posición en el cromosoma

```

hijo1(i) = padre2(i)

hijo2(i) = padre1(i)

Next i

End If

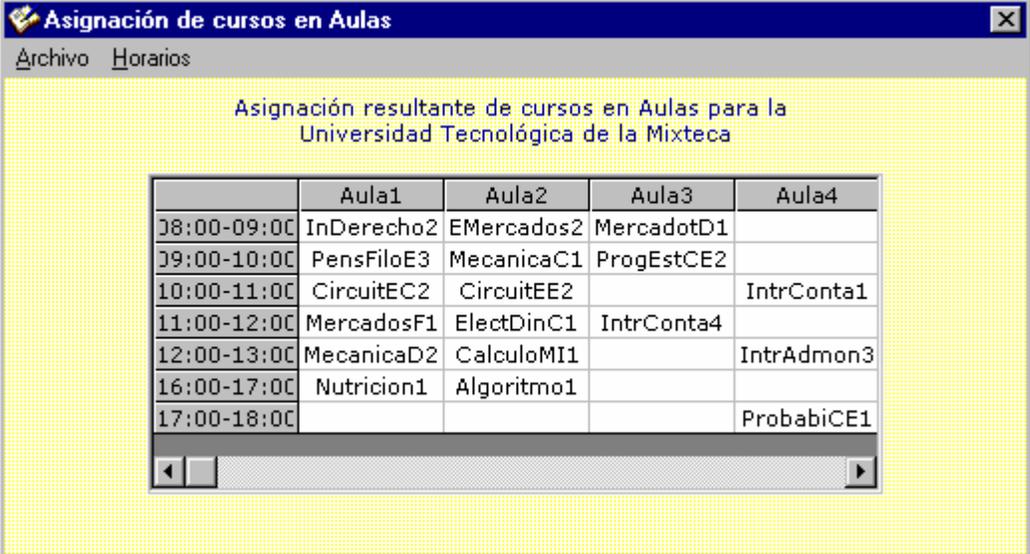
End Sub

ANEXO

EJEMPLO DE UNA ASIGNACIÓN DE CURSOS.

Asignación de cursos para el semestre Octubre-Febrero.

Estas asignaciones fueron realizadas para un total de 39 aulas y con 20 generaciones.



Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula1	Aula2	Aula3	Aula4
08:00-09:00	InDerecho2	EMercados2	MercadotD1	
09:00-10:00	PensFiloE3	MecanicaC1	ProgEstCE2	
10:00-11:00	CircuitEC2	CircuitEE2		IntrConta1
11:00-12:00	MercadosF1	ElectDinC1	IntrConta4	
12:00-13:00	MecanicaD2	CalculoMI1		IntrAdmon3
16:00-17:00	Nutricion1	Algoritmo1		
17:00-18:00				ProbabiCE1

Asignación de cursos en Aulas [X]

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula5	Aula6	Aula7	Aula8
08:00-09:00	CSatelite1	SistOperE1	Calculo C5	Calculo C2
09:00-10:00		AuditoriA1		TeorEconE2
10:00-11:00	RecursosM1		InvyProto1	
11:00-12:00		CalculoCE6	ProbyEstA1	
12:00-13:00	SEmpresaI1	CircuitEC1	MecanicaE3	DisProgAd1
16:00-17:00	ProgEstrE5	MicroEcon1		
17:00-18:00	DisEditor1	PensFiloI1		ProgEstCE3

Asignación de cursos en Aulas [X]

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula9	Aula10	Aula11	Aula12
08:00-09:00	ProgConta1	ElectDinE1	MercantIII1	ElectDinE2
09:00-10:00		DisCircuE1	DibTecniI1	ImpaqueyE1
10:00-11:00	SCOptica1			MOptimiII1
11:00-12:00	FinanzaEI1	DerFiscal1	ConsCivil1	PensFiloC4
12:00-13:00	IngAlimeI1	MecanicaC7	FisicoQII1	
16:00-17:00		TecRepres1	ProbyEstC2	PensFilCE2
17:00-18:00	SExpertoI1	Calculo A2	PensFilCE5	BalanceME1

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula13	Aula14	Aula15	Aula16
08:00-09:00	CalculoCE4	ProgEstrC1	QuimiGral2	IntrConta5
09:00-10:00	MecanicaE4	AdmonPers2	ProgFinan2	MetyTrans2
10:00-11:00	ProbyEstD1		ProgEstrI1	
11:00-12:00		ProbyEstE2	Calculo E5	
12:00-13:00	ProgEstrD2	EconomiaI1	MecanicaA1	EProyecto1
16:00-17:00	PensFilCE6	IntIngDis1		InDerecho1
17:00-18:00	ComplejIII1	Composici1	SistOperC2	

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula17	Aula18	Aula19	Aula20
08:00-09:00		IntrConta6	ProgEstrC7	
09:00-10:00	GeDescrip1	MFClasica1	QuimAnali1	CalcVaria1
10:00-11:00	PensFiloE4	PensFiloE1	EcuadifMI1	PensFiloC8
11:00-12:00	SistOperC1	ContabiII2	MicroEcon2	
12:00-13:00	IngSoftwC1		IntIngEle5	IntrConta2
16:00-17:00	MecanicaD3	EcuadifC2		IntIngEle2
17:00-18:00		MetInvest1	PensFiloA1	IntrAdmon4

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula21	Aula22	Aula23	Aula24
08:00-09:00		PensFilCE3	AdministF1	ProgEstrA1
09:00-10:00	Calculo D1	IngAliIII1	Calculo D2	AlgSuperi1
10:00-11:00			DisInteri1	Calculo I1
11:00-12:00		ArqCompuE:	ContabilA1	MecanicaC5
12:00-13:00	ProgEstrE4	ProgEstrD3	MercantII2	EcoUrbana1
16:00-17:00	IntrAdmon1	IntIngEle3		
17:00-18:00		TeoMatCom:		TeorComuC:

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula25	Aula26	Aula27	Aula28
08:00-09:00	ProgEstCE4	ProgEstCE1	CalculoCE2	EcuacDifC1
09:00-10:00		Calculo C4		
10:00-11:00		IntrConta3	PensFiloD1	PensFiloC7
11:00-12:00		AdmonPers1		PensFiloE3
12:00-13:00	AnalisiNI1	MercadotE1		TeorEconA1
16:00-17:00			EcuacDifE1	
17:00-18:00			FEvalProy1	ProgEstCE2

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula29	Aula30	Aula31	Aula32
08:00-09:00			IngAliIII1	
09:00-10:00	AdmonPers1	DesDeProd1	ProgConta2	
10:00-11:00		MecanicaE4	EProyecto1	TCereales1
11:00-12:00	MecanicaC6	IntrAdmon6		
12:00-13:00	ContabilD1	Calculo C5	SistDigiC1	PensFilCE4
16:00-17:00	CalculoMI1	IntrConta3	IntrConta5	SistOperC2
17:00-18:00	ProgFinan1	TeorEconD1		

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula33	Aula34	Aula35	Aula36
08:00-09:00	ProgEstCE4	MecanicaD2	MecanicaC8	QuimiGral2
09:00-10:00	MecanicaI1	DisCircuE1	Calculo D1	CalcVaria1
10:00-11:00				Calculo E3
11:00-12:00	MicroEcon2	MecanicaC3		CircuitEE2
12:00-13:00	MercadotC1		ProgEstrC1	ProbyEstA1
16:00-17:00	MecanicaC1	QuimOrgII1	IntIngCom9	
17:00-18:00		EMercados1	ProgEstrD2	IntrConta2

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula37	Aula38	Aula39
08:00-09:00	ProgEstrE4		InDerecho1
09:00-10:00	ProbabiCE1		SEmpresaI1
10:00-11:00	ProgEstrA1	InvyProto1	AdmonPers2
11:00-12:00			IntIngCom6
12:00-13:00	Composici1	PensFilCE6	BasDatosI1
16:00-17:00	ContabilE1	GeDescrip2	IntrAdmon5
17:00-18:00	RecursosM1	IntrAdmon3	

En esta asignación puede observarse que no existe ningún choque para los alumnos, ya que las materias no se repiten, ni están asignadas dos materias del mismo grupo a la misma hora.

Asignación de cursos para el semestre Marzo-Julio.

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula1	Aula2	Aula3	Aula4
08:00-09:00		MetodNumD:		
09:00-10:00	ProgOObjI1		ProgEstrM1	CompiladE1
10:00-11:00	Vidrios 1	ProgAdmvo1	ElectrosC1	AlgLineCE1
11:00-12:00	MetodNumC:			QuimAlimt1
12:00-13:00				
16:00-17:00		IngAlimIII1		
17:00-18:00	TeoGralCE3		FisOndulC3	

Asignación de cursos en Aulas				
Archivo Horarios				
Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca				
	Aula5	Aula6	Aula7	Aula8
08:00-09:00	ContCosto2			FisOndulC2
09:00-10:00	EstDatosC5		InvOperaC1	
10:00-11:00	CircuitoE1	CalculoII1	Psicologi4	
11:00-12:00	EcuDifMII1			SistComuC1
12:00-13:00	MetodNumC:	AlgLineaE2		PEspacial1
16:00-17:00	AlgLineaD2	AdministA1		SistCompC1
17:00-18:00	FenTranIII1	TeoGralCE2	PEstrateg1	

Asignación de cursos en Aulas				
Archivo Horarios				
Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca				
	Aula9	Aula10	Aula11	Aula12
08:00-09:00		EstAdmvsA1	DesOrgani1	
09:00-10:00		InvOperaE1		
10:00-11:00	ContabilI3	IntArtifC1		ProgAdmvo2
11:00-12:00		TeoGralSM1		SistCompCE
12:00-13:00				
16:00-17:00	AdSueldos1	ControlDg1	FisicoQu11	
17:00-18:00	EstDatosD2	BsDatosIII1		AlgModerI1

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula13	Aula14	Aula15	Aula16
08:00-09:00	ElectrosD2	ElectrosE2		
09:00-10:00		AnalisisExp1	SEmpresII1	
10:00-11:00				
11:00-12:00	AlgLineCE4	ProgTecno1	Mercadote1	
12:00-13:00		IntIngAli1		3sDatosCE3
16:00-17:00		Auditoria1	MetodNumE1	
17:00-18:00	EstAdmvsE1		CompiladC1	

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula17	Aula18	Aula19	Aula20
08:00-09:00		Microonda1	MetodNumA:	
09:00-10:00	TeorGenSE1	TeoAlgor1	MediosAud1	ComplejaI1
10:00-11:00	AnalisisI1	FinanzEII1		
11:00-12:00	ElectrosC3			
12:00-13:00	MicroComC1		AdSueldos2	
16:00-17:00			DesProduct1	
17:00-18:00		MFModerna1	SExpertos1	CircuitEC3

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula21	Aula22	Aula23	Aula24
08:00-09:00		AlgLineaA1		AlgLineaD1
09:00-10:00		ProbabilI1	FisOndule1	
10:00-11:00				TeoGralCE4
11:00-12:00	DireccioD1			AlgLineaE1
12:00-13:00		DecisionF1		
16:00-17:00				
17:00-18:00	OCTInform1		Psicologi3	

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula25	Aula26	Aula27	Aula28
08:00-09:00			AnalisisI1	
09:00-10:00		DesProduc1	ProgSistE1	
10:00-11:00	PDSeñales1	PEspacial1	ProgAdmvo1	
11:00-12:00			RMaterial1	
12:00-13:00	3sDatosCE1	CompiladE1		
16:00-17:00	EconomiaM1			ContabilI3
17:00-18:00	EstDatosE2		CircuitEC2	

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula29	Aula30	Aula31	Aula32
08:00-09:00				MercadoIII1
09:00-10:00	ProgOObjI1	TeorGenSD2	AlgLineaE1	BioQuimIII1
10:00-11:00	AnalisisNII1		AlgLineCE3	Psicologi4
11:00-12:00	DesOrgani1		FisOndulD1	EstDatosA1
12:00-13:00	AnalisisExp1			
16:00-17:00		PEstrateg1	AdministC1	DireccioE1
17:00-18:00	TeoGralCE4			

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula33	Aula34	Aula35	Aula36
08:00-09:00	MetodNumC:		IntIngAli1	AdministE1
09:00-10:00	AnalisisD1			
10:00-11:00	Mercadote1	AlgModerI1	DireccioC1	EstudForm1
11:00-12:00		SistCompC5		TeoGralCE1
12:00-13:00		ProgAdmvo2	AlgLineaI1	
16:00-17:00			DesProduc1	ProgSistC3
17:00-18:00		AplicGraf1		AlgLineaC5

Asignación de cursos en Aulas

Archivo Horarios

Asignación resultante de cursos en Aulas para la Universidad Tecnológica de la Mixteca

	Aula37	Aula38	Aula39
08:00-09:00	MercantiI1		
09:00-10:00			PDSeñales1
10:00-11:00	FisOndule1	IngAlimII1	
11:00-12:00	DiseñoUyA1	SistCompCE	
12:00-13:00			
16:00-17:00	3sDatosCE2		
17:00-18:00	CPotencia1		

En esta asignación puede observarse que no existe ningún choque para los alumnos, ya que las materias no se repiten, ni están asignadas dos materias del mismo grupo a la misma hora.