

# Análisis comparativo del método de asignación por clases en GAVaPS

*Lic. Laura Lanzarini<sup>1</sup>, Lic. Cecilia Sanz<sup>2</sup>, Lic. Marcelo Naiouf<sup>3</sup>, Ing. Fernando Romero<sup>4</sup>*

*Laboratorio de Investigación y Desarrollo en Informática<sup>5</sup>  
Facultad de Informática  
Universidad Nacional de La Plata*

## Resumen

Se presentan tres alternativas dentro del método de asignación por clases para el cálculo del tiempo de vida de los individuos en algoritmos genéticos con población de tamaño variable (GAVaPS).

En la estrategia planteada (asignación por clases), se agrupa a los individuos de acuerdo a su fitness. El objetivo es utilizar el rango de valores permitidos para el tiempo de vida de una manera más adecuada para la búsqueda del óptimo, que en las estrategias proporcional, lineal y bilineal.

Se realiza un estudio comparativo de las tres posibilidades de asignación por clases con relación a los métodos tradicionales, y se muestran resultados en la búsqueda del óptimo sobre cinco funciones. Finalmente, se presentan algunas conclusiones y se marcan posibles líneas de trabajo futuro.

*Palabras Claves: Computación Evolutiva. Algoritmos Genéticos. Algoritmos Genéticos de Población Variable*

---

<sup>1</sup> Profesor Adjunto Dedicación Exclusiva. Facultad de Informática. Universidad Nacional de La Plata. E-mail: laural@lidi.info.unlp.edu.ar

<sup>2</sup> Becaria de Perfeccionamiento CIC. Jefe de Trabajos Prácticos Dedicación Exclusiva. Facultad de Informática. Universidad Nacional de La Plata. E-mail: csanz@lidi.info.unlp.edu.ar

<sup>3</sup> Profesor Adjunto Dedicación Exclusiva. Facultad de Informática. Universidad Nacional de La Plata. E-mail: mnaiouf@lidi.info.unlp.edu.ar

<sup>4</sup> Jefe de Trabajos Prácticos Semi Dedicación. Facultad de Informática. Universidad Nacional de La Plata. E-mail: fromero@lidi.info.unlp.edu.ar

<sup>5</sup> Calle 50 y 115 1er Piso, (1900) La Plata, Argentina, TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

## Introducción

En los últimos años ha crecido el interés en los sistemas de resolución de problemas basados en principios de evolución y herencia. Los mismos mantienen una población de soluciones potenciales, poseen algún proceso de selección basándose en el fitness o bondad de los individuos y algunos operadores genéticos. Estos algoritmos imitan los principios de la evolución natural para problemas de optimización de parámetros [Gallard 98] [Goldberg 98][Grefenstette 85][Michalewicz 96]. Los algoritmos genéticos de John Holland son un tipo de tales sistemas basados en evolución. Están inspirados en la Teoría de Evolución de Darwin [Holland 75].

Las implementaciones de AGs no resultan siempre satisfactorias por distintas razones:

- la codificación de un problema no es la adecuada y hace que los AGs operen sobre un espacio de búsqueda que no se corresponde con el del problema.
- existe un límite sobre el número de generaciones y otro sobre el tamaño de la población.

Por ello, bajo ciertas condiciones se produce una convergencia prematura hacia óptimos locales. Este es un problema típico de los algoritmos genéticos.

El tamaño de la población es una de las elecciones más importantes y puede ser crítico en distintas aplicaciones. Si es muy pequeño, el AG puede converger prematuramente y si es muy grande, el tiempo de espera para una mejora puede ser muy largo con el consiguiente gasto de recursos computacionales. Por otro lado, tanto la diversidad poblacional como la presión selectiva se ven afectados por el tamaño de la población.

Los algoritmos genéticos comienzan con un conjunto de soluciones o individuos (representados por cromosomas) llamado población. Las soluciones de una población son utilizadas para generar una nueva población. Esto se realiza con la esperanza de que la nueva sea mejor que su predecesora. En los algoritmos genéticos de tamaño de población fija (SGA) los individuos elegidos para formar nuevos descendientes se seleccionan de acuerdo a su fitness. Aquellos con mayor fitness, tendrán más chance para la reproducción. Esto se repite hasta que se verifique alguna condición tal como número de poblaciones o alcanzar un óptimo esperado.

Los algoritmos genéticos con tamaño de población variable (GAVaPS) tratan de resolver algunos de los problemas que se producen en los AG con tamaño fijo de la población. No utilizan ninguna de las variantes de selección para la eliminación de individuos, sino que incorporan el concepto de **edad** del cromosoma (dependiente de su fitness). Esto es equivalente al número de generaciones que el cromosoma ha permanecido vivo hasta el momento, y va variando con el paso de las generaciones. En consecuencia, la edad del cromosoma reemplaza al concepto de selección. Esto hace que el tamaño de la población no permanezca constante sino que varíe con el correr de las generaciones. Además se introduce otro concepto muy importante que es el de tiempo de vida (lifetime), el cual indica la cantidad de generaciones que un individuo debe permanecer vivo. El tiempo de vida es asignado sólo una vez a cada individuo al momento de su nacimiento y permanece constante durante la evolución.

Existen diferentes métodos de asignación de tiempo de vida. Este trabajo analiza diferentes alternativas del método de asignación por clases [Lanzarini 99] y sus bondades respecto de los métodos tradicionales [Michalewicz 96].

## Estrategias tradicionales de asignación de tiempos de vida en GAVaPS

Los algoritmos genéticos de población variable no utilizan mecanismo de selección sino presión selectiva mediante la asignación del tiempo de vida a cada individuo. La estrategia para la asignación de tiempo de vida debería considerar:

- Respalda a los individuos con fitness superior a la media
- Ajustar el tamaño de la población corriente de acuerdo a la etapa actual en la búsqueda, evitando principalmente el crecimiento exponencial

Hay tres enfoques tradicionales para la asignación del tiempo de vida a los individuos. Estos utilizan algunas medidas del estado de la búsqueda, tales como el fitness promedio, el fitness máximo y el mínimo. Estos enfoques son las asignaciones proporcional, lineal y bilineal.

#### *Asignación proporcional.*

El tiempo de vida del  $i$ -ésimo individuo se calcula de la siguiente manera:

$$\text{TiempoVida}(i) := \min(\text{MIN\_LT} + (\eta * \text{fitness}(i) / \text{avgfit}), \text{MAX\_LT});$$

donde  $\eta = \frac{1}{2} (\text{MAX\_LT} - \text{MIN\_LT})$

Esto proviene de la idea de selección por ruleta, donde el valor de tiempo de vida de un individuo es proporcional a su fitness, dentro de los límites MIN\_LT y MAX\_LT.

Este enfoque presenta fallas dado que solamente utiliza el fitness promedio para caracterizar a la población actual. De esta forma, para tener el máximo tiempo de vida, un individuo debe ser muy bueno (el fitness debe ser dos veces el promedio), con lo cual no se utiliza todo el rango de valores disponibles de tiempo de vida en forma adecuada.

#### *Asignación lineal.*

$$\text{TiempoVida}(i) := \text{MIN\_LT} + 2 * \eta * (\text{fitness}(i) - \text{absminfit}) / (\text{absmaxfit} - \text{absminfit})$$

donde  $\eta = \frac{1}{2} (\text{MAX\_LT} - \text{MIN\_LT})$ .

Esta estrategia asigna un tiempo de vida proporcional al máximo y mínimo fitness encontrados en la población anterior. Como consecuencia de la mejora que se produce de generación en generación en los fitness de los individuos, se asignan tiempos de vida cada vez mayores. Esto provoca un gran incremento en el tamaño de la población.

#### *Asignación bilineal.*

Si  $\text{avgfit} \geq \text{fitness}(i)$

entonces  $\text{TiempoVida}(i) := \text{MIN\_LT} + \eta * (\text{fitness}(i) - \text{minfit}) / (\text{avgfit} - \text{minfit})$

Sino  $\text{TiempoVida}(i) := \frac{1}{2} (\text{MIN\_LT} + \text{MAX\_LT}) + \eta * (\text{fitness}(i) - \text{avgfit}) / (\text{maxfit} - \text{avgfit})$

donde  $\eta = \frac{1}{2} (\text{MAX\_LT} - \text{MIN\_LT})$ .

Este método trabaja utilizando el fitness promedio y divide de esta forma a los individuos en dos clases. El rango posible de tiempos de vida se separa en dos partes iguales y de acuerdo a la distancia del fitness del individuo respecto del promedio, se establece el offset dentro del segmento correspondiente.

### **Asignación por Clases**

La asignación por clases surgió con el objetivo de asignar tiempos de vida con las siguientes características:

- obtener una distribución más adecuada dentro del rango de valores permitidos, en el sentido de premiar más a aquellos individuos con mejor fitness
- limitar el crecimiento de la cantidad de individuos para encontrar el óptimo
- evitar la convergencia hacia un óptimo local
- limitar la dispersión de los individuos una vez que se localizó la “zona” del óptimo

En la estrategia planteada, se agrupa a los individuos en diferentes clases de acuerdo a su fitness. Para ello se trabaja ingresando un número de clases, que da la cantidad de agrupamientos distintos que se utilizarán. Las clases se forman utilizando el algoritmo de clusterización de k-medias

[Maravall 94][Kaufman 90][Jain 86], que es un método típico de clasificación no supervisada, donde para cada clase se tiene un "centro" o "individuo medio" de la clase.

Inicialmente, el algoritmo de k-medias toma los k primeros fitness distintos como centros y arma las clases con el resto de los individuos de la primera generación distribuyéndolos de acuerdo a su menor distancia con un centro. Luego de distribuir todos los individuos los centros son recalculados. El algoritmo continúa iterando hasta que no se produzcan cambios significativos en los centros de las clases.

Para cada nueva generación, el tiempo de vida de sus individuos se determina sobre la agrupación de los individuos de la generación anterior.

En función de la clase a la que pertenecen (clase con la que tiene distancia mínima al centro) y del valor de su propio fitness recibirá su tiempo de vida, dado que cada clase tendrá un subrango asignado dentro del total de tiempos de vida.

Las clases varían desde las que contienen individuos muy malos (con bajo fitness) a aquellas con individuos muy buenos (alto fitness).

Se implementaron dos estrategias para la elección del rango de tiempos de vida de cada clase.

#### Estrategias para la elección del rango de tiempos de vida

##### a) Con tiempo de vida fijo por clase

Se divide el rango total de tiempos de vida entre las k clases existentes. El valor de tiempo de vida de un individuo depende de su fitness y de la clase a la que pertenece, sin tener en cuenta al resto de las clases.

$$\{TiempoVida := Tiempo Vida Clases Anteriores + OffsetTiempoVida Clase más cercana\}$$

$$AnchoClase := MAX\_LT / k$$

$$TVClasesAnt := (ClaseMasCercana - 1) * AnchoClase$$

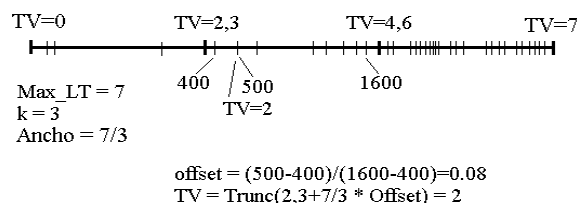
$$TVClaseAct := AnchoClase$$

$$Offset = (fitness[i] - Clases[ClaseMasCercana].MinFit) / abs(Clases[ClaseMasCercana].MaxFit - Clases[ClaseMasCercana].MinFit)$$

$$TiempoVida[i] := trunc(TVClasesAnt + AnchoClase * Offset)$$

donde

- AnchoClase es el rango de tiempos de vida asignados para cada clase (dado por MAX\_LT / Cantidad de Clases)
- ClaseMasCercana es el número de clase a la que pertenece el individuo
- TVClaseAct es el ancho de tiempos de vida de la clase del individuo
- TVClasesAnt es el ancho de tiempos de vida de las clases anteriores a la que pertenece el individuo
- Clases[ClaseMasCercana].MinFit y Clases[ClaseMasCercana].MaxFit son los fitness mínimo y máximo de la clase del individuo en cuestión
- Fitness[i] es el fitness del individuo i.
- Offset es un número entre 0 y 1 que indica el desplazamiento del individuo dentro de la clase



## b) Con tiempo de vida proporcional a la cantidad de individuos en cada clase

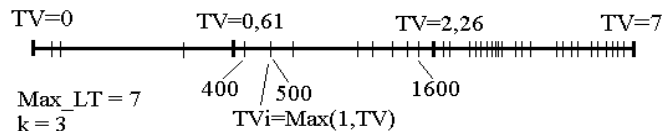
Se toma como ancho para cada clase una función de la cantidad de individuos que posee la misma, de forma tal que las que tienen mayor cantidad de individuos tengan un subrango mayor de tiempos de vida para asignar. Esto provoca una mayor presión selectiva en la evolución.

$$\{TiempoVida := Tiempo Vida Clases Anteriores + OffsetTiempoVida Clase más cercana\}$$

```
CantAnt := 0
for i:=1 to ClaseMasCercana-1 do CantAnt := CantAnt + Clases[i].cantIndiv
TVClasesAnt := .MAX_LT * CantAnt / CantTotalIndiv;
TVClaseAct := MAX_LT * Clases[ClaseMasCercana].CantIndiv / CantTotalIndiv
Offset = ( fitness[i] - Clases[ClaseMasCercana].MinFit ) /
          abs(Clases[ClaseMasCercana].MaxFit - Clases[ClaseMasCercana].MinFit)
TiempoVida[i] := trunc(TVClasesAnt + AnchoClase * Offset)
```

donde

- Clases[ClaseMasCercana].CantIndiv es la cantidad de individuos en la clase más cercana
- CantTotalIndiv es la cantidad total de individuos en la población



$$\text{offset} = (500-400)/(1600-400)=0.08$$
$$TV = \text{Trunc}(0,61+1,64*0.08)=\text{Trunc}(0,74)=0$$

En base a las dos estrategias se desarrollaron tres alternativas:

1. Asignación por clases iguales: utiliza la estrategia planteada en a).
2. Asignación por clases con mayor presión selectiva: utiliza la estrategia planteada en b).
3. Asignación por clases mixta: tiene una primera etapa donde ejerce presión selectiva según el concepto b) y una segunda donde trabaja según la estrategia a). La división entre una etapa y la siguiente se basa en un porcentaje (25%) de la cantidad de generaciones.

*Condición de terminación de la asignación por clases*

La condición de terminación del algoritmo se basa en los siguientes criterios:

- ⇒ En la cantidad máxima de generaciones indicada en los parámetros de entrada al sistema
- ⇒ En no registrar cambios significativos en el óptimo durante el 20% de la cantidad máxima de generaciones
- ⇒ En que no exista una diferencia significativa en el centro de la mejor clase durante las últimas 7 generaciones.
- ⇒ En la distancia entre los centros de las dos mejores clases. De generación en generación, los centros de las clases se van acercando al óptimo, reduciéndose la distancia entre ellos. Luego, cuando los centros de las mejores clases de generaciones consecutivas difieren en un cierto error ( $\epsilon$ , parámetro del sistema) se considera que se alcanzó el óptimo.

## Resultados Obtenidos

El GAVaPS con las distintas estrategias de asignación fue aplicado a las siguientes funciones:

$$G1: f(x) = x^2 \quad x \in [-50,50]$$

$$G2: f(x) = -x \sin(10\pi x) + 1 \quad x \in [-2,1]$$

$$G3: f(x) = \text{int}(8x) / 8 \quad x \in [0,1]$$

$$G4: f(x) = x * \text{sgn}(x) \quad x \in [-1,2]$$

$$G5: f(x) = 0.5 + \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{(1 + 0.001 * (x^2 + y^2))^2} \quad x, y \in [-100,100]$$

Estas funciones fueron elegidas porque cubren un amplio espectro de tipos de funciones posibles a ser optimizadas. Las funciones G2 y G5 son multimodales, con varios máximos locales. La función G3, no puede ser optimizada por medio de la técnica de gradiente (no existe el gradiente en una dirección). La función G4 representa un problema reconocido como problema engañoso [Goldberg 89]

Para las distintas funciones, se consideraron los siguientes parámetros para comparar las distintas estrategias de asignación:

- fitness máximo alcanzado, para evaluar el error con el que se aproxima el óptimo.
- cantidad de individuos en la última generación, lo que da una idea del costo computacional
- la varianza dentro del espacio de soluciones

Por otra parte, para obtener los resultados se promediaron 30 corridas de cada asignación sobre cada una de las 5 funciones.

En las Figuras 1 a 5, se comparan las estrategias de asignación, correspondiendo la **0** a la **proporcional**, **1** a la **lineal**, **2** a la **bilineal**, **3** a la **asignación por clases iguales**, **4** a la **asignación por clases con mayor presión selectiva**, y **5** a la **asignación por clases mixta**.

Como se puede apreciar en las Figuras correspondientes a las funciones G1, G2, G3, G4, G5 la estrategia de asignación por clases (4) (5) y (6) obtiene un bajo error en la aproximación al óptimo acorde a la cota fijada en los parámetros de entrada, trabajando generalmente con la menor cantidad de individuos y con baja varianza.

Asimismo, en las distintas experiencias pudo observarse que la asignación por clases fue el único método que reguló la cantidad de individuos durante la búsqueda de la solución. Las restantes estrategias encuentran el óptimo "inundando" de individuos el espacio de búsqueda, mientras la asignación por clases encuentra la solución de manera más dirigida, tratando de no superpoblar con soluciones malas.

De las tres alternativas de asignación por clases, la de mayor presión selectiva es la que utiliza la menor cantidad de individuos (tal como era previsible), aunque en ciertos casos pierde algo de precisión en la aproximación al óptimo. Este "error" está influenciado por la cota  $\epsilon$ . Los resultados para la asignación por clases mixta se encuentran entre las otras dos alternativas de asignación por clases en cuanto a cantidad de individuos y varianza, registrando buenos valores para el óptimo. Un caso particular para la asignación por clases se observa en la función G5, donde la varianza es mayor que para las estrategias tradicionales. Esto se debe a la presencia de menor cantidad de individuos y la necesidad de diversidad genética por tratarse de una función con gran cantidad de óptimos locales.

## **Conclusiones y trabajo futuro**

Se presentan tres variantes para la asignación por clases de tiempos de vida en GAVaPS, en donde se logran mejoras en algunos aspectos con relación a las estrategias tradicionales. Estas últimas concentran la asignación de valores para tiempos de vida sólo en una parte de todo el rango posible. La asignación por clases hace uso de todo el rango disponible, distinguiendo más adecuadamente entre individuos “buenos” y “malos” de acuerdo al fitness.

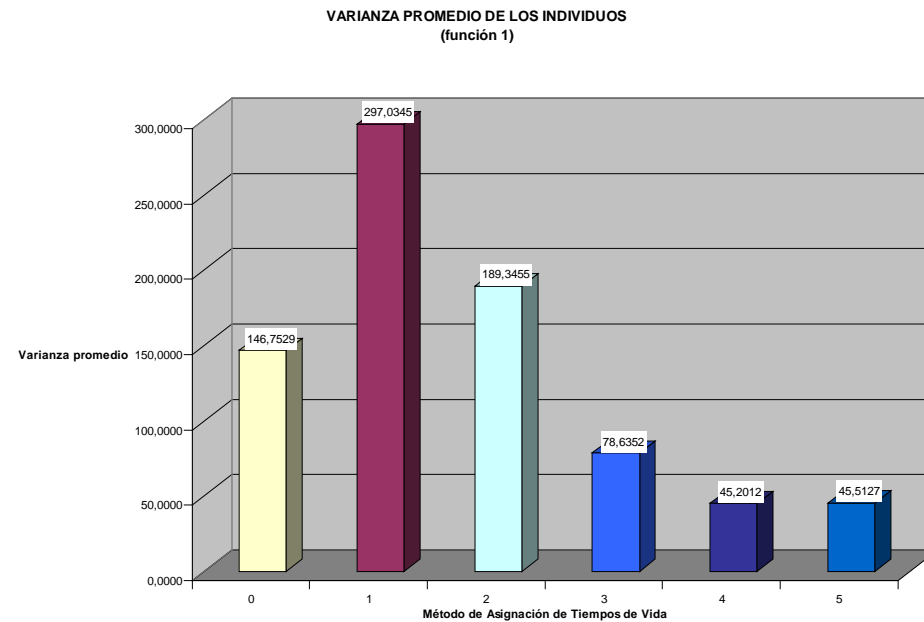
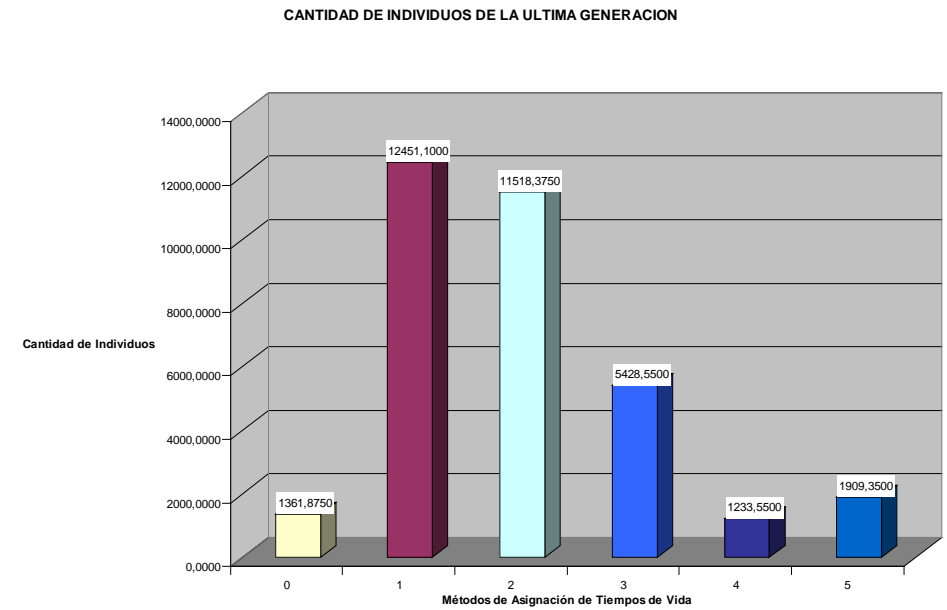
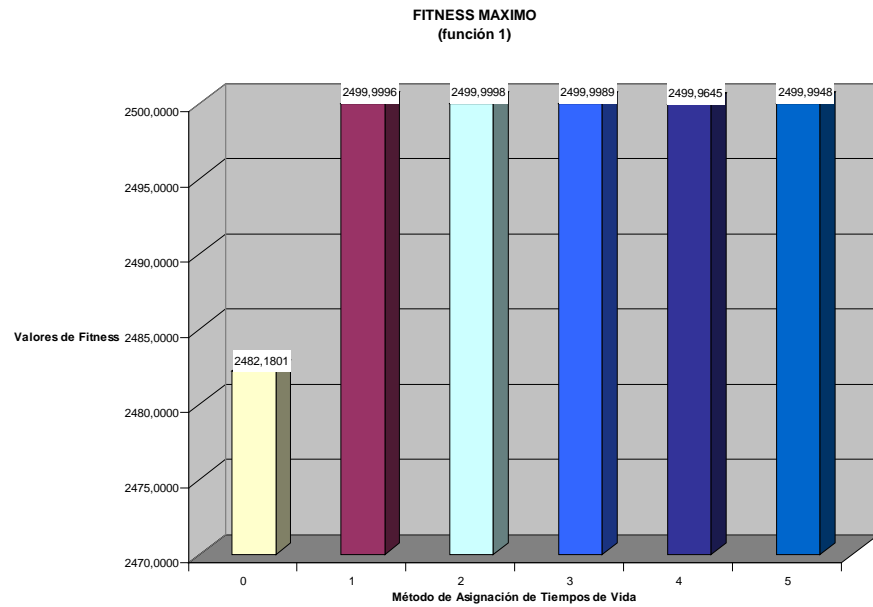
Por otro lado, se observa que la convergencia de la asignación por clases se produjo por las condiciones relacionadas con la obtención del óptimo, mientras que en los resultados mostrados para los otros métodos, generalmente, se provoca la terminación al inundarse de individuos el espacio de búsqueda.

Finalmente, los resultados obtenidos son satisfactorios respecto de los objetivos iniciales.

Actualmente, se está trabajando en un nuevo método de asignación por clases, donde la cantidad de clases se estima en ejecución de acuerdo a la población actual.

Como es conocido, el método de k-medias requiere que el valor de k sea indicado a priori. Si bien los resultados aquí expuestos corresponden a una distribución en cinco clases, el valor de k es fuertemente dependiente del problema y del rango total de tiempos de vida. Nótese que la cantidad de clases es directamente proporcional a la presión selectiva que se ejerce. Por esto, se está desarrollando un nuevo método que analiza la distribución de los fitness de los individuos y forma las clases necesarias de acuerdo a ello.

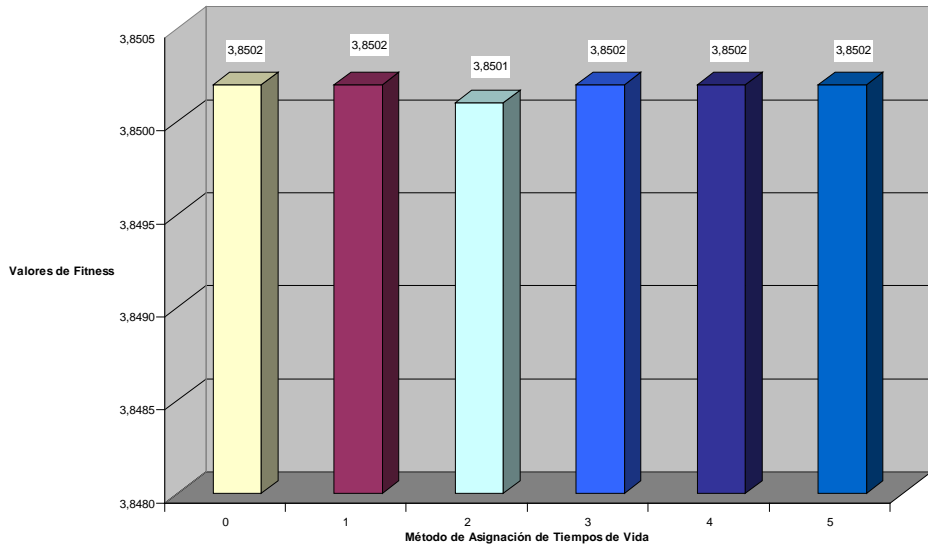
**Figura 1** - Fitness Máximo, cantidad de individuos de la última generación, y varianza encontrados para cada uno de los métodos de asignación de tiempos de vida en la función G1



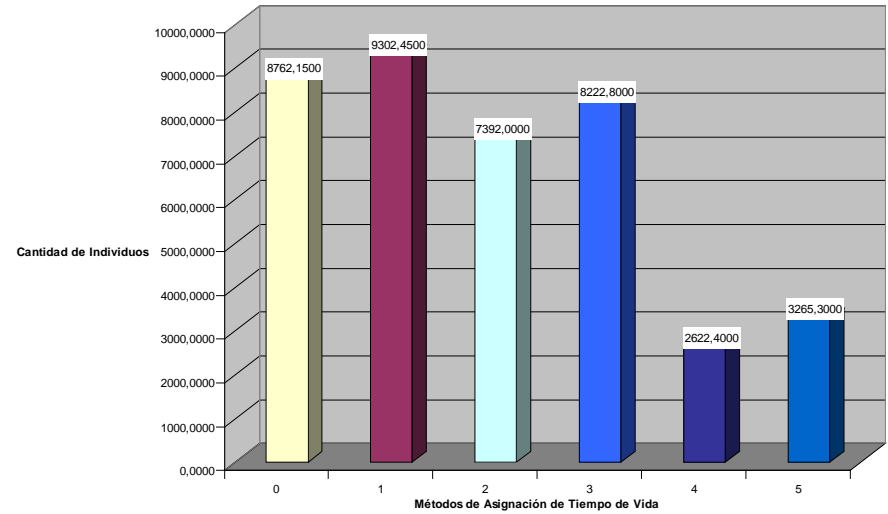


**Figura 2** - Fitness Máximo, cantidad de individuos de la última generación, y varianza encontrados para cada uno de los métodos de asignación de tiempos de vida en la función G2

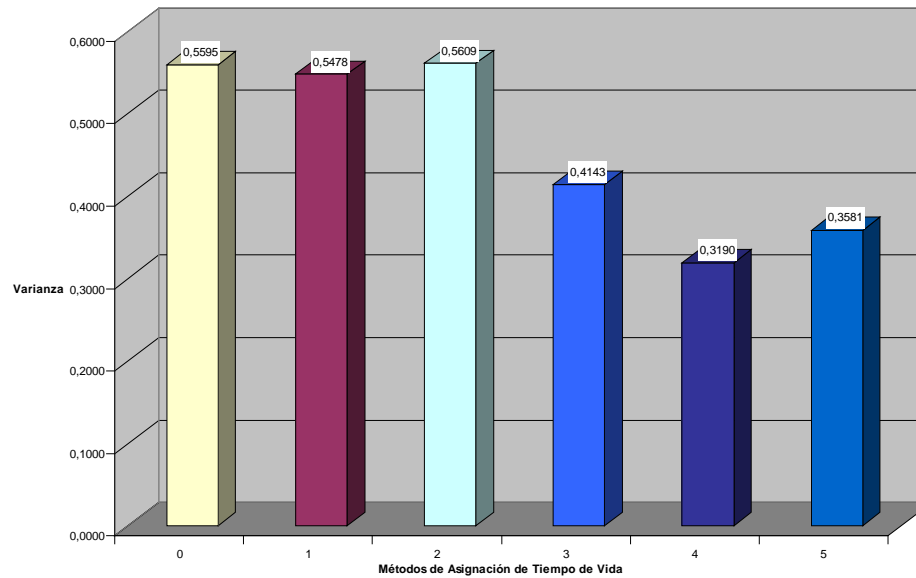
**FITNESS MAXIMO**  
(Función 2)



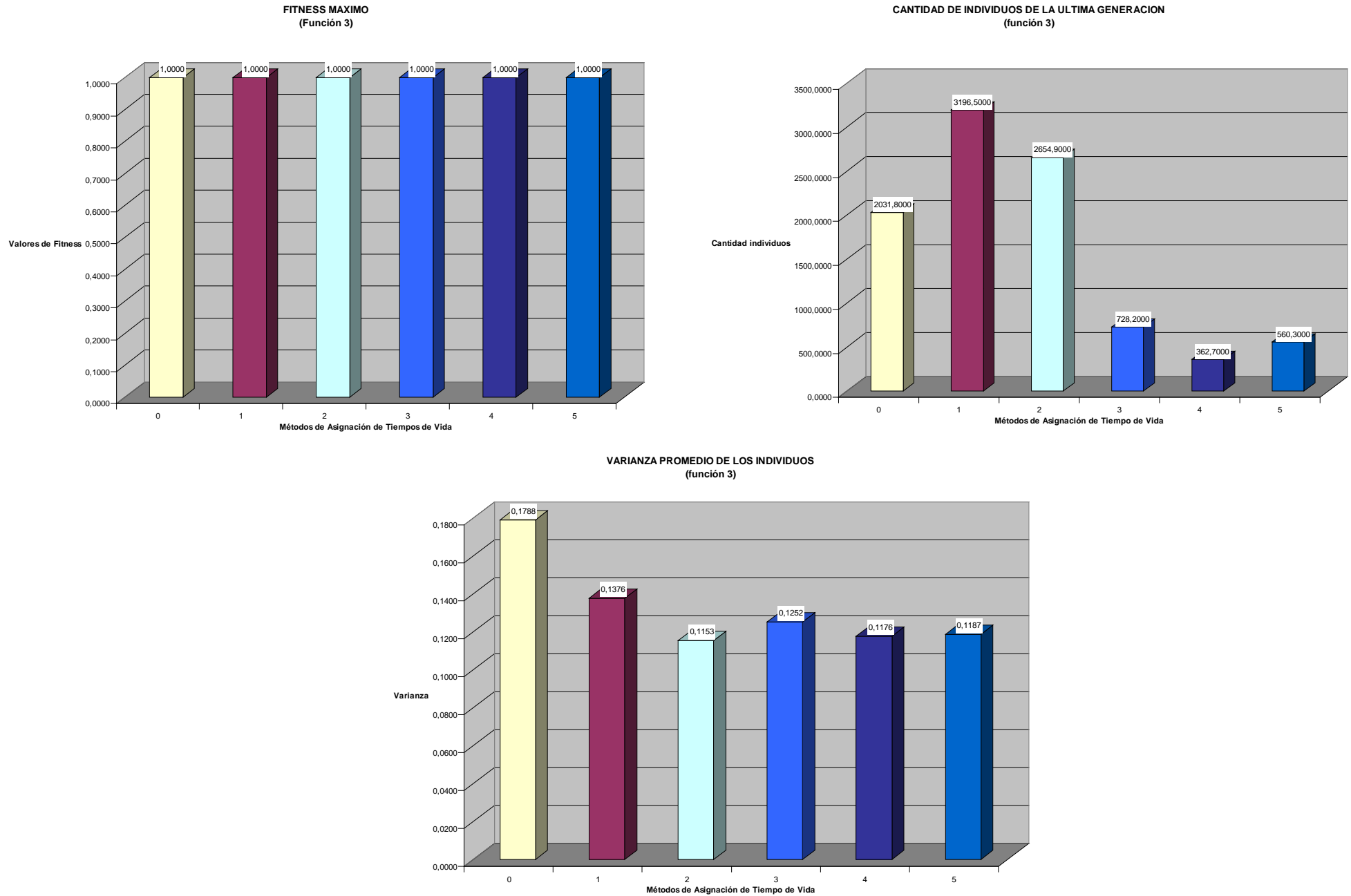
**CANTIDAD DE INDIVIDUOS DE LA ULTIMA GENERACION**  
(función 2)



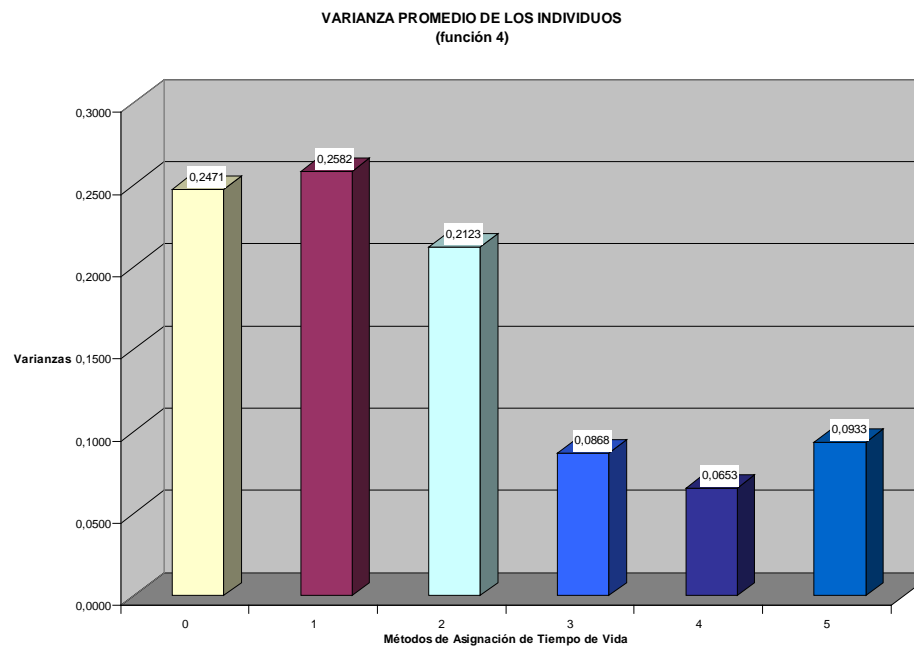
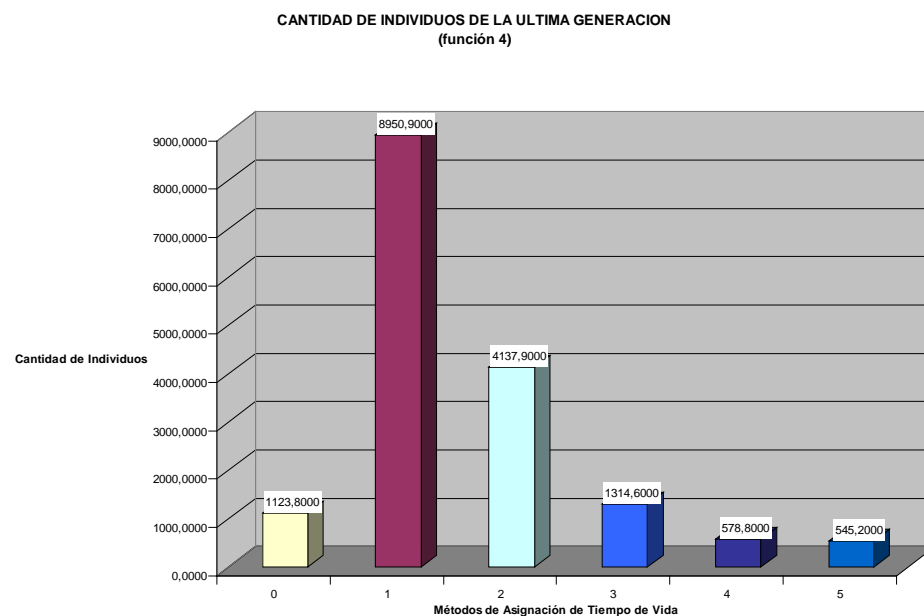
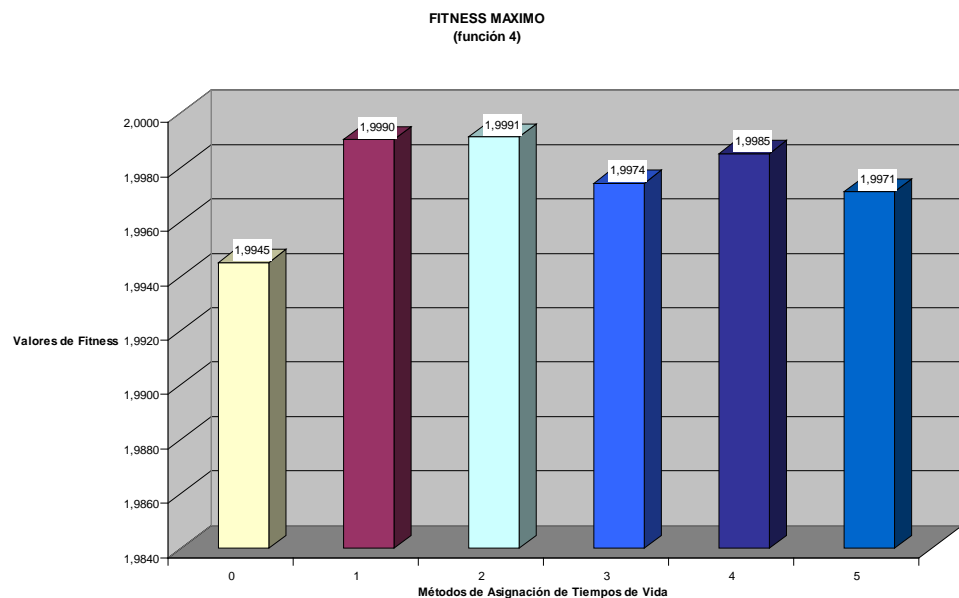
**VARIANZA PROMEDIO DE LOS INDIVIDUOS**  
(función 2)



**Figura 3** - Fitness Máximo, cantidad de individuos de la última generación, y varianza encontrados para cada uno de los métodos de asignación de tiempos de vida en la función G3

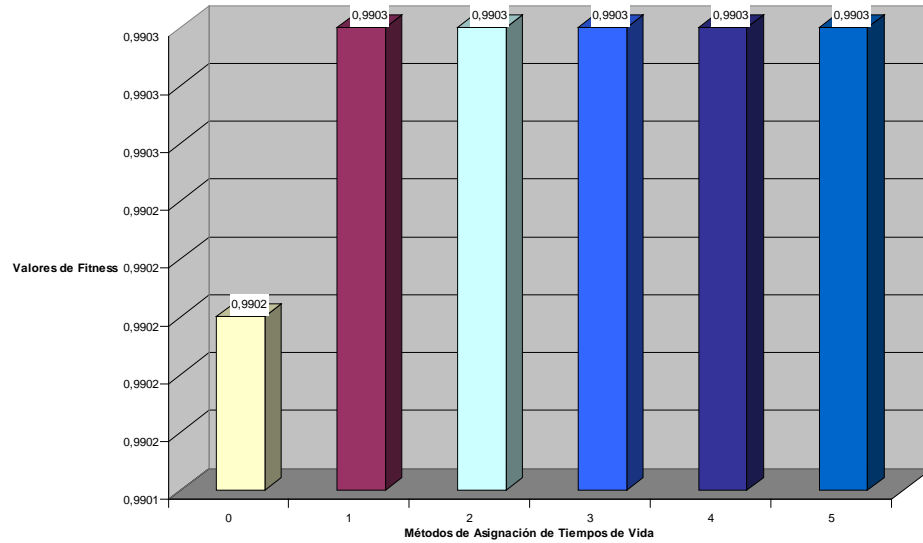


**Figura 4 - Fitness Máximo, cantidad de individuos de la última generación, y varianza encontrados para cada uno de los métodos de asignación de tiempos de vida en la función G4**

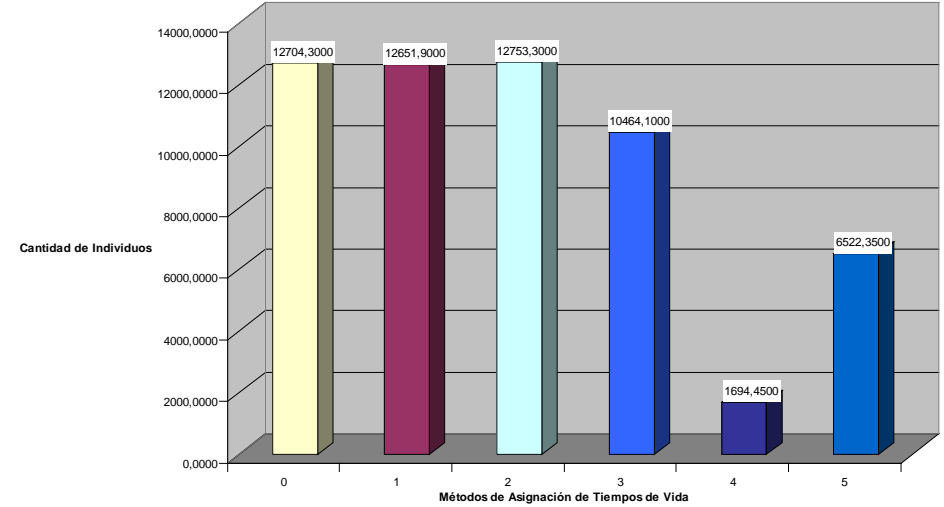


**Figura 5** - Fitness Máximo, cantidad de individuos de la última generación, y varianza encontrados para cada uno de los métodos de asignación de tiempos de vida en la función G5

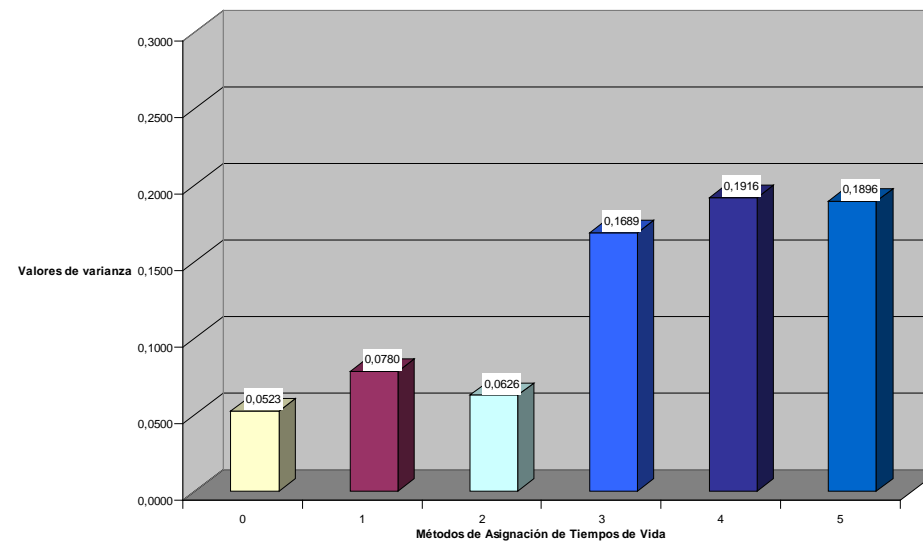
**FITNESS MAXIMO**  
(Función 5)



**CANTIDAD DE INDIVIDUOS DE LA ULTIMA GENERACION**  
(Función 5)



**VARIANZA DE LOS INDIVIDUOS**  
(Función 5)



## Referencias

- [**Gallard 98**] “Técnicas de Computación evolutivas”. Raúl Gallard . Curso de Postgrado dictado en la Dpto. de Informática, Fac. Cs. Exactas. UNLP. 1998.
- [**Goldberg 89**] “Genetic Algorithms. In search, Optimization & Machine Learning”, David E. Goldberg. Addison Wesley. 1989
- [**Goldberg 98**] “Genetic Algorithms. In search, Optimization & Machine Learning”, David E. Goldberg. Addison Wesley. 1998
- [**Grefenstette 85**] “Optimization of control parameters for genetic algorithms”, J.J. Grefenstette. IEEE Transactions of Systems, Man, and Cybernetics, SMC-16, pp. 122-128. 1985
- [**Holland 75**] "Adaptation in Natural and Artificial Systems", Holland. 1975
- [**Jain 86**] "Cluster Analysis". Jain. Handbook of Pattern Recognition and Image Processing. Academic Press. 1986. Pp. 33-57.
- [**Kaufman 90**] "Finding Groups in data". L. Kaufman, P.J. Rousseuw. John Wiley. 1990
- [**Lanzarini 99**] "Una nueva alternativa en los métodos de asignación de tiempo de vida en algoritmos genéticos con población de tamaño variable", L. Lanzarini, C. Sanz, M. Naiouf, F. Romero. Aceptado para International Congress of Information Engineering 99. También disponible versión como Reporte Técnico LIDI.
- [**Maravall 94**] “Reconocimiento de formas y Visión artificial”, Darío Maravall-Gómez Allende. Addison Wesley. 1994
- [**Michalewicz 96**] “Genetic Algorithms+Data Structures = Evolution Programs”. 3er Edition, Zbigniew Michalewicz. Springer. 1996