

**UNIVERSIDAD COMPLUTENSE DE MADRID**

**FACULTAD DE CIENCIAS BIOLÓGICAS**



**P.E.T.R.I. (PROGRAMMING EVOLUTION THROUGH  
REITERATED INFECTION):  
DISEÑO DE UN ALGORITMO GENÉTICO INSPIRADO  
EN MECANISMOS GENÉTICOS MICROBIANOS**

**MEMORIA PARA OPTAR AL GRADO DE DOCTOR  
PRESENTADA POR**

**Carlos Perales Graván**

**Madrid, 2008**

- **ISBN: 978-84-692-0087-2**

**FACULTAD DE CIENCIAS BIOLÓGICAS  
DE LA  
UNIVERSIDAD COMPLUTENSE DE MADRID**



**P.E.T.R.I.**

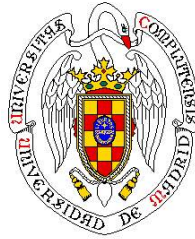
*(Programming Evolution Through Reiterated Infection):*

Diseño de un algoritmo genético inspirado en  
mecanismos genéticos microbianos

**CARLOS PERALES GRAVÁN**

**Marzo de 2008**

**FACULTAD DE CIENCIAS BIOLÓGICAS  
DE LA  
UNIVERSIDAD COMPLUTENSE DE MADRID**



**P.E.T.R.I.**

*(Programming Evolution Through Reiterated Infection):*

Diseño de un algoritmo genético inspirado en  
mecanismos genéticos microbianos

**CARLOS PERALES GRAVÁN**

**Director:** Dr. Rafael Lahoz Beltrá

**Marzo de 2008**



Dedicado a la memoria de mi Padre.

Quiero aprovechar la ocasión para dar las gracias a mi madre, **Esperanza**, sin cuya motivación y comprensión jamás hubiera podido realizar el trabajo presentado en esta Memoria. También a mi hermano, **Jorge**, por el apoyo científico y tecnológico. Sin él todavía estaría simulando. Y a **Marta**, por supuesto, por ayudarme en la redacción de esta Memoria (todos los lectores te lo agradecerán).

Por otro lado, también quiero dar las gracias a **Rafa**, por el apoyo, guía y libertad brindadas y por las discusiones motivadoras en la pizarra del departamento. A **Maria Teresa**, por recibirme siempre con una sonrisa y demostrarme que otro tipo de enseñanza es posible. Y a las **empresas** en las que he trabajado, sin cuya flexibilidad no hubiera podido terminar la investigación. Y también al **Dpto. de Matemática Aplicada** por el soporte concedido y al **Dpto. de Microbiología** por la formación suministrada.

Además quiero dar las gracias a mis **amigos**, sin cuyas bromas y apoyo no hubiera podido resistir el esfuerzo (no quiero olvidarme a nadie así que prefiero no nombraros, de todas formas, vosotros sabéis de sobra quienes sois).

Y por último, quiero hacer una mención especial a mi **despertador**, las seis de la mañana no son nada sin él.

A TODOS, INFINITAS GRACIAS.

# Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos</b>	<b>12</b>
<b>3. Metodología</b>	<b>14</b>
3.1. Algoritmo Genético Bacteriano (SBGA) . . . . .	15
3.2. Experimentos de evaluación de SBGA . . . . .	22
3.3. PETRI: Un Algoritmo Genético Bacteriano con transducción . . .	33
3.4. Experimentos de evaluación de PETRI . . . . .	37
<b>4. Resultados</b>	<b>43</b>
4.1. Experimentos con SBGA . . . . .	44
4.2. Experimentos con PETRI . . . . .	52
<b>5. Discusión</b>	<b>60</b>
<b>6. Conclusiones</b>	<b>68</b>
<b>Bibliografía</b>	<b>70</b>

# **Capítulo 1**

## **Introducción**

La Naturaleza, a través de la evolución, ha sido y es capaz de “resolver” problemas de toda índole consiguiendo estructuras y organismos altamente eficaces y adaptados a ambientes muy dispares, ya sean plantas, animales o microorganismos. El vuelo de los pájaros (Brown, 1963), la ventilación de los nidos de termitas (Korb, 2003) o la obtención de energía a partir de la luz solar de las plantas (Zelitch, 1971) son un claro ejemplo de ello. Desde tiempos remotos la Naturaleza ha representado para el ser humano una fuente de inspiración en la que científicos y tecnólogos han encontrado soluciones bioinspiradas (Lahoz-Beltrá, 2004) altamente optimizadas para la resolución de problemas muy variados. Así por ejemplo, las alas de los aviones se inspiraron en las alas de los pájaros (Kármán, 1954), el velcro se inspiró en los frutos del cardo alpino (*Xanthium spinosum*), los métodos de computación neuronal en el cerebro animal y humano (Hopfield, 1982), o los métodos de la computación evolutiva en evolución de las especies por selección natural Darwiniana (Goldberg, 1989), ...

Otro ejemplo lo encontramos en los mecanismos genéticos y evolutivos de las bacterias, organismos que han sido capaces de adaptarse a todos los medios, incluyendo los más extremos. Por ejemplo, las bacterias son capaces de vivir en fumarolas en el fondo del mar (Beatty *et al.*, 2005) así como en suelos con alta concentración de sal (Tehei y Zaccai, 2005).

Estas consideraciones nos condujeron a la principal motivación del trabajo desarrollado y descrito en esta Memoria *¿será posible desarrollar un método computacional o algoritmo con el que resolver problemas de optimización a partir de un modelo elemental de los mecanismos genéticos microbianos?*

El convencimiento inicial de que efectivamente sería posible simular los mecanismos genéticos y evolutivos de las bacterias, diseñando un algoritmo genético inspirado en ellas, así como la ausencia en la bibliografía de un algoritmo similar fueron los dos principales argumentos que nos motivaron a la realización de esta investigación. Teniendo en cuenta estas consideraciones, el objetivo principal del trabajo descrito en esta Memoria ha sido el *diseño de un algoritmo genético no convencional inspirado en los mecanismos genéticos de la conjugación y la transducción bacterianas.*

Aunque algunos trabajos de investigación previos (Smith, 1996; Furuhashi *et al.*, 1995; Harvey, 1996) abordan la conjugación bacteriana, no lo hacen a través de la simulación del mecanismo de conjugación *per se*, sino que desarrollan algoritmos basados en la transferencia horizontal de genes (una de las particularidades de la conjugación), la mutación en bacterias y la transformación bacteriana con métodos de combinación génica tradicionales (recombinación, mutación, etc). La naturaleza de la transducción bacteriana tampoco se simula como tal en trabajos anteriores (Kubota *et al.*, 1996), ya que, en este caso, los virus simulados son un subconjunto de soluciones que, lejos de ser simples agentes de transferencia de

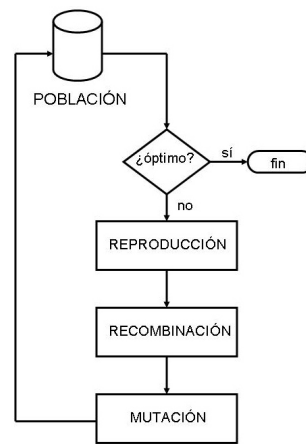
información entre bacterias, son sometidas a evaluación.

La Computación Evolutiva (CE) es un conjunto de técnicas de optimización estocástica inspiradas en el mecanismo evolutivo de la teoría de selección natural de Darwin (Lahoz-Beltrá, 2004). Actualmente, la mayoría de los métodos en Computación Evolutiva se inspiran en los fundamentos principales del neo-Darwinismo, considerando como fuentes principales de variabilidad la recombinación o sobrecruzamiento de cromosomas y la mutación. En la actualidad, esta clase de algoritmos se aplican en problemas muy diversos, como por ejemplo en el problema consistente en encontrar la ruta más corta (Liu y Yang, 2007), en tareas de planificación, ya sea la asignación de tareas u horarios (Aickelin y Dowsland, 2004), el aprendizaje automático (Bies *et al.*, 2006), el diseño de circuitos electrónicos (Lahoz-Beltrá, 2001; Perales-Graván y Lahoz-Beltrá, 2007), así como la simulación de la evolución de patrones en biología, tal es el caso del patrón de manchas de la piel de las cebras (Perales-Graván y Lahoz-Beltrá, 2004), entre otros muchos ejemplos.

Los métodos en CE evolucionan una población de soluciones potenciales (o cromosomas) a partir de la selección de aquellas soluciones que representan la mejor solución o individuo, calculándose un valor que puntúa la bondad o calidad de la solución, denominado *fitness*. Una vez concluida la selección se obtiene la siguiente generación modificando genéticamente los individuos elegidos. El cambio se realiza a través de la simulación de uno o más mecanismos genéticos o fuentes de variabilidad, siendo los más comunes la recombinación cromosómica (esto es el intercambio de fragmentos entre dos soluciones o cromosomas) y la mutación (o cambio aleatorio en una solución). Por consiguiente, la población evolucionará generación tras generación hasta obtener una población de individuos con uno o más individuos o cromosomas representando buenas soluciones.

Los algoritmos genéticos, los algoritmos evolutivos, la programación genética, etc., son diferentes tipos de algoritmos en CE que tienen en común el uso de operadores genéticos, así como los siguientes pasos generales (Fig. 1.1):

1. Obtener aleatoriamente una población inicial  $P(0)$ .
2. Evaluar el valor *fitness* de cada individuo de la población  $P(t)$ .
3. Seleccionar a los individuos mejor evaluados para su reproducción.
4. Obtener una nueva generación a través de la recombinación y/o de la mutación y crear la descendencia.
5. Reemplazar toda o parte de la población con la descendencia.
6. Repetir los pasos 2-5 hasta que se cumpla una condición de terminación.



**Figura 1.1:** Esquema de algoritmo genético

En los algoritmos de CE se usan tres métodos principales para simular la selección natural a través de la reproducción: el primero de ellos se conoce como ruleta sesgada o *roulette wheel parents selection* (Davis, 1991). Se basa en la reproducción de los individuos con mayor *fitness* mediante la simulación de la selección de los individuos girando una rueda de ruleta que asigna a cada cromosoma o individuo un sector cuyo arco es proporcional a su valor *fitness*. El segundo método se denomina *tournament selection* (Miller y Goldberg, 1995). En este método se eligen para pasar a la siguiente generación los cromosomas que en una muestra elegida al azar tengan el mayor *fitness*. Finalmente, una tercera metodología se basa en la regla simple de muerte y supervivencia de los individuos basada en su valor de *fitness*, simulada frecuentemente como un valor de energía. Aquel individuo con un valor *fitness* menor que un umbral no podrá reproducirse.

Como se ha señalado previamente, en la naturaleza existen diversos mecanismos genéticos que aumentan la variabilidad de las poblaciones, tales como la recombinación o la mutación. Además, de estos mecanismos clásicos de variabilidad la Genética Microbiana constituye una rica fuente de mecanismos no suficientemente explorados en CE como por ejemplo la transformación, la conjugación, la transducción, la transposición, u otros mecanismos genéticos tales como la translocación, la eliminación o la duplicación de material genético. Estos mecanismos alternativos no se consideran normalmente en CE. Así por ejemplo, la transferencia horizontal de genes llevada a cabo por los llamados retrovirus, es decir la transferencia de información genética entre especies (Benveniste, 1985), y la simbiogénesis o unión de varios organismos para formar uno solo (Margulis y Sagan, 2002), ilustran dos claros ejemplos de este tipo de fenómenos.

La eficiencia de un algoritmo de CE viene dada por el grado de variación

cromosómica alcanzado con el operador genético responsable de la variabilidad poblacional. Simoes y Costa (2000, 2001) encontraron, a través de experimentos de simulación por ordenador, que un algoritmo genético con los operadores de transformación y transposición consigue mejores soluciones que un algoritmo genético estándar basado en la recombinación.

En esta Memoria se proponen dos nuevos operadores genéticos inspirados en los mecanismos genéticos microbianos, en particular un *operador conjugación*, y un *operador transducción*.

### **Mecanismos genéticos microbianos: Una fuente de inspiración en CE**

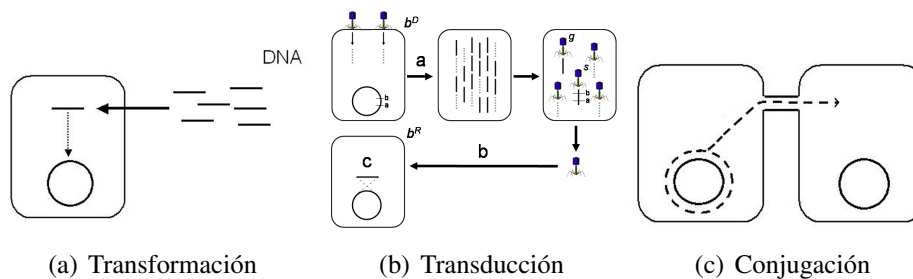
La conjugación bacteriana, esto es la transferencia de material genético entre una bacteria donadora y una receptora, es también una fuente de variabilidad genética para las bacterias (Miller, 1992; Birge, 2000) que podría ser una fuente de inspiración para crear un nuevo operador genético orientado a mantener o aumentar la variabilidad poblacional en los algoritmos de CE. Aunque en la actualidad hay algunos trabajos previos que exploran esta propuesta el potencial de la conjugación como operador genético en CE no ha sido estudiado en detalle ni en sus posibles aplicaciones hasta la fecha. Algunos estudios preliminares son: Pseudo-Bacterial Genetic Algorithm (PBGA), Microbial Genetic Algorithm (Microbial GA), y el trabajo realizado por Smith (1996).

De un modo resumido el algoritmo Pseudo Bacterial Genetic Algorithm (PBGA) introducido por Furuhashi *et al.* (1995) se inspira vagamente en la genética bacteriana y en la evolución microbiana, aplicando en una población un proceso de mutación, evaluación, selección y reemplazo mejorando porciones parciales de cromosomas. El método se basa en la selección de clones de una población de cromosomas y su mutación. Una vez elegidas las  $n$  mejores partes mutadas, éstas son transferidas y recombinadas a la población de cromosomas originales. El Microbial Genetic Algorithm (Microbial GA), presentado por Harvey (1996) está basado en un torneo microbiano, en el cual se enfrentan dos miembros o cromosomas parentales elegidos aleatoriamente de una población inicial. El cromosoma de mayor *fitness* se mantiene intacto mientras que el de menor puntuación es infectado por una copia del cromosoma ganador y posteriormente mutado. Esta forma de recombinación se inspira en la característica de transferencia horizontal de genes de la conjugación bacteriana, donde se transfiere un fragmento de ADN desde un miembro de la población a otro. Por último, Smith (1996) modeló la conjugación bacteriana para determinar si este mecanismo era potencialmente mejor que la recombinación. El algoritmo asume una población de cien especímenes distribuidos al azar en una rejilla de  $15 \times 15$  moviéndose aleatoriamente. La conjugación se produce cuando los sujetos están en posiciones adyacentes de la matriz (ca-

da espécimen contiene información de la capacidad conjugativa y la longitud del fragmento a transferir). En ninguno de los casos anteriores se estudió en detalle y mucho menos se explotó a través de alguna aplicación todo el potencial conjugación bacteriana.

A día de hoy muy pocos mecanismos diferentes a la conjugación han sido estudiados en CE. Es el caso de Simoes y Costa (2000), que presentaron varios algoritmos simulando el mecanismo de transposición, es decir, la presencia de unidades genéticas móviles o transposones capaces de saltar o recolocarse en el mismo cromosoma o en otro diferente. Los autores presentaron también el algoritmo Transformation Genetic Algorithm (TGA) (Simoes y Costa, 2001), un algoritmo genético que incluye la simulación de un mecanismo de transformación bacteriana. La transformación es uno de los mecanismos responsables de la variación genética bacteriana en el que la bacteria tomaría ADN del medio. En cada generación del algoritmo, los cromosomas que van a ser transformados se seleccionan al azar y se modifican con segmentos génicos contenidos en un grupo de segmentos. Posteriormente, cuando los nuevos cromosoma reemplazan a la población antigua, el grupo de segmentos se varía usando la población antigua y el azar.

**Conjugación** El intercambio de información genética en poblaciones de bacterias en la naturaleza es muy dinámico, utilizando mecanismos en los que se transfieren los genes desde una bacteria donante a otra receptora (Fig. 1.2):

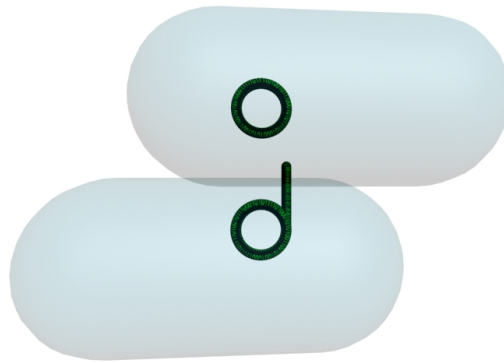


**Figura 1.2:** Mecanismos genéticos bacterianos

La transformación (Fig. 1.2(a)), tal y como se adelantó con anterioridad, es un fenómeno natural en el que una bacteria receptora toma fragmentos de ADN del entorno que pueden ser incorporados al cromosoma. Por otro lado, la transducción (Fig. 1.2(b)) involucra la transferencia de genes (fragmentos de ADN) de una bacteria donante a otra receptora a través de un bacteriófago, esto es de un virus que infecta bacterias y al que también se denomina fago. Los bacteriófagos se reproducen dentro de las bacterias interactuando con la maquinaria celular para

llevar adelante su ciclo vital. Durante la replicación del bacteriófago dentro de la bacteria, un fragmento del genoma bacteriano puede empaquetarse por error en la cápsida del virus. Posteriormente, este fago infectará a otra bacteria (la receptora) y le inyectará el ADN de la bacteria donante, pudiéndose dar la recombinación con las regiones homólogas del cromosoma bacteriano receptor.

En contraste con la transducción, en la conjugación (Fig. 1.2(c)) la ausencia de un mediador requiere un contacto físico directo entre dos bacterias: la donante y la receptora. Por esta razón, y considerando que normalmente la bacteria se reproduce asexualmente, la conjugación es incorrectamente asumida como el equivalente bacteriano de la reproducción sexual. La conjugación no debe entenderse como tal, ya que no involucra la fusión de gametos ni la creación de un cigoto.



**Figura 1.3:** Esquema de conjugación

De una forma más detallada, en la conjugación (Llosa *et al.*, 2002) la habilidad de una bacteria de ser donante viene dada por la presencia de un plásmido (ADN autónomo extracromosómico circular) de alrededor de 100 kilobases de longitud denominado factor  $F$  o factor de fertilidad (Fig. 1.3). El factor  $F$  incluye los genes necesarios para obtener la habilidad de transferir ADN a una bacteria receptora, produciendo un complejo macromolecular (Christie, 2001) que construye un pelo sexual (apéndice proteico que une a las bacterias), así como un poro (Lawley *et al.*, 2003) en las superficies bacterianas. Actualmente se cree que la transferencia de ADN desde la bacteria donadora a la bacteria receptora no se realiza a través del pelo sexual, sin dejar de considerarla una estructura muy importante en el proceso de conjugación. El factor  $F$  puede ser autónomo o estar integrado en el cromosoma bacteriano, recibiendo la bacteria donadora el nombre de  $F^+$  y Hfr (*High Frequency of Recombination*) respectivamente. La bacteria sin el factor  $F$  es la receptora (contener el factor  $F$  evita ser el receptor de la conjugación) y se

denomina  $F^-$ . El mecanismo de la conjugación involucra dos tipos distintos de cruces:  $F^+ \times F^-$  y  $Hfr \times F^-$ . En el cruce  $F^+ \times F^-$ , en el momento en el que el complejo macromolecular crea un puente entre las dos bacterias, una cadena del plásmido bacteriano de la bacteria  $F^+$  comienza a entrar en la bacteria  $F^-$ . Si la bacteria  $F^+$  consigue transferir una copia completa del factor  $F$  a la receptora, ambas bacterias se convertirían en  $F^+$ , ya que ambas tendrían el factor  $F$ . Por otro lado, en los cruces  $Hfr \times F^-$  la bacteria donante empieza a transferir el factor  $F$  y, en el caso de que consiga permanecer unida, continúa transfiriendo los genes cromosómicos posteriores a la zona de inserción. La bacteria receptora raramente se convierte en  $Hfr$ , ya que para ello debería poder transferirse todo el cromosoma de la bacteria donante. De hecho, la transferencia del cromosoma completo se produce con poca frecuencia, ya que, normalmente, el puente conjugativo se rompe antes de la transferencia completa debido a gran diversidad de factores, dependiendo la longitud del fragmento transferido de la temperatura y la agitación del medio (Wood, 1968), la concentración de sales (Novotny y Fives-Taylor, 1974) y de otros muchos factores como los expuestos por Pansegrau y Lanka (1996), etc. Una vez terminada la transferencia, las bacterias se separan. Posteriormente, en la bacteria receptora se produce una recombinación homóloga entre la cadena cromosómica receptora y la proveniente de la donadora. La bacteria donante se mantiene intacta. Es importante señalar que en la naturaleza sólo se llevaría a cabo la recombinación después de la conjugación en el caso de que la bacteria donante sea  $Hfr$ . Después de la recombinación, el nuevo cromosoma de la bacteria receptora puede pasar a generaciones futuras.

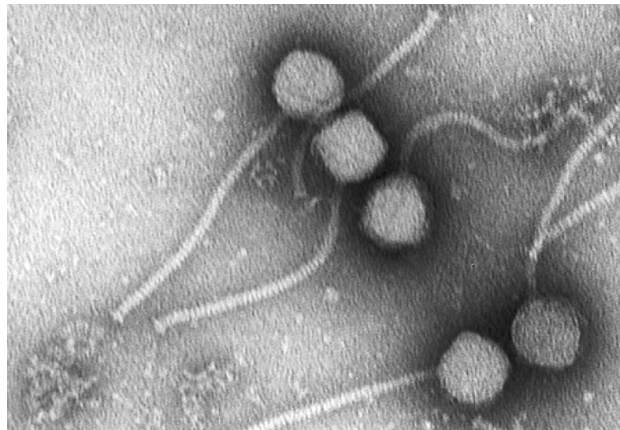
**Transducción** Se denomina transferencia horizontal de genes a cualquier proceso en el que un organismo transfiere un segmento genético a otro que no pertenezca a su descendencia. En la actualidad, la mayoría de los algoritmos genéticos se basan en la transferencia vertical de genes (mecanismo en el que un organismo recibe su material genético desde su antecesor), pudiéndose afirmar que el ámbito para la transferencia vertical es la población. Por otro lado el ámbito de la transferencia horizontal es la biosfera (parte del sistema ecológico planetario donde ocurre la vida, incluyendo la atmósfera, la geosfera y la hidrosfera) lo que lo convierte en un mecanismo evolutivo importante.

Hace más de treinta años Anderson (1970) sugirió que la “transducción vírica” fuese considerada como uno de los mecanismos clave de la propagación horizontal, ya que existía la posibilidad de transportar segmentos de ADN incluso desde individuos pertenecientes a un *phylum* hasta individuos pertenecientes a otro *phylum* (una de las divisiones de primer orden usadas para clasificar especies, por ejemplo, *Arthropoda*, *Chordata*, *Actinobacteria*, *Briophyta*, *Coniferophyta*, etc.).

Además, la dinámica evolutiva de las poblaciones puede depender de la trans-

ferencia de ADN de una población a otra. Por ejemplo, Syvanen (1985) sugirió que la transferencia de genes entre especies puede ayudar a explicar muchas de las observaciones constatadas en la naturaleza, tales como la rápida explosión evolutiva (véase (Gould y Eldredge, 1977), la hipótesis del equilibrio puntuado) y la gran ocurrencia de paralelismo en el registro fósil (o de un modo equivalente la evolución convergente de rasgos similares).

De acuerdo con Margulis (1981) la adquisición y acumulación de mutaciones aleatorias no es suficiente para explicar el modo en el que se suceden las variaciones heredadas. Mientras el Darwinismo, principal teoría evolutiva utilizada en CE, enfatiza el papel de la selección natural como fuerza evolutiva, Lynn Margulis y otros autores hacen lo propio con el papel de la cooperación. En el ámbito de la genética microbiana, las bacterias y los virus comparten una larga relación evolutiva. Esta relación está principalmente promovida por bacteriófagos (Davis *et al.*, 1990), un tipo de parásito intracelular que se multiplica dentro de la bacteria haciendo uso de su maquinaria biosintética (Fig. 1.4).



**Figura 1.4:** Bacteriófago  $\lambda^1$ .

Este tipo de virus tiene unas dimensiones de entre 20 y 200 nanómetros, y consta de dos componentes esenciales: un ácido nucleico (el material genético, ADN o ARN) y una cápsida proteica (cabeza protectora que conserva y transporta el ácido nucleico). Cuando los bacteriófagos infectan una bacteria, su modo normal de reproducción usa la maquinaria de replicación bacteriana, generando numerosas copias de su ácido nucleico que después son empaquetadas en las cápsidas recién sintetizadas.

<sup>1</sup>**Fuente:** Powledge TM: *New Antibiotics Resistance Is Futile*. PLoS Biol 2/2/2004: e53. <http://dx.doi.org/10.1371/journal.pbio.0020053>. **Autor:** Vincent Fischetti y Raymond Schuch, The Rockefeller University. Creative Commons 2.5

Algunos bacteriófagos, debido a un error, son capaces de mover ADN bacteriano de una bacteria a otra, proceso conocido como transducción. En consecuencia, la transducción puede ser considerada como un tipo particular de fenómeno de “emigración génica” a nivel celular. Considerando el ciclo vital de un bacteriófago en particular, podemos definir dos grandes tipos de transducción (Fig. 1.2(b)):

- La *transducción generalizada* tiene lugar si por error cualquier parte del cromosoma bacteriano (en vez del ADN vírico) es secuestrado en la cápsida viral (por ejemplo fagos T4 en *Escherichia coli*).
- La *transducción especializada* se produce únicamente en el caso de que genes específicos o ciertos segmentos característicos del cromosoma bacteriano sean empaquetados por error (por ejemplo fagos  $\lambda$  en *Escherichia coli*).

Aunque el papel en problemas de optimización de la transferencia horizontal, lateral o entre poblaciones, inspirándose en la genética microbiana, no ha sido explorada suficientemente en CE, no así la transferencia de información entre poblaciones, siendo ésta una de las metodologías clásicas en CE.

La incorporación de un procedimiento de transferencia génica entre poblaciones se conoce como Algoritmo Genético Distribuido o DGA (*Distributed Genetic Algorithm*). En los DGA, la población se divide en subpoblaciones o islas. Posteriormente, y de acuerdo con una tasa de migración o intervalo generacional establecido, algunos genes o cromosomas son transferidos de una isla a otra en un fenómeno llamado migración. Cuando los emigrantes seleccionados llegan a la nueva isla, los individuos nativos son reemplazados por los inmigrantes. En la actualidad existen dos grandes modelos de migración en DGA. El primero de ellos se denomina *Stepping Stones Population* (Nang y Matsuo, 1994) y está bioinspirado en un modelo bastante conocido de anillo de especies, en el que se asume una topología anular secuencial (Liebers *et al.*, 2004; Chen y Wineberg, 2004) en la que los individuos emigran de un nodo a otro de su vecindad. Por ejemplo, uno de los modelos clásicos en la literatura científica es ASPARAGOS (Gorges-Schleuter, 1989, 1991). El segundo modelo, llamado *Random Migration Model* (Hiroyasu *et al.*, 1999) asume que la isla objetivo se elige al azar o a través de algún criterio estocástico. Independientemente del modelo de migración del DGA, cada isla o subpoblación ejecuta su propio algoritmo genético, normalmente implementado como un *Simple Genetic Algorithm* (SGA) como el descrito por Goldberg (1989).

La mayoría de los métodos desarrollados para implementar eficazmente un DGA se inspiran en la Teoría de la Estructura Poblacional de Wright (1951). De acuerdo con Wright, la estructura de la población más favorable es aquella que subdivide las especies en poblaciones locales pequeñas y aisladas, debiendo existir suficiente difusión para que una subpoblación que adquiriera una característica

---

favorable pueda transformar a sus vecinas, sin olvidar otros factores que también intervienen, como los relacionados con la migración y la topología de la comunicación (las conexiones entre las subpoblaciones). El autor del artículo (Cantú-Paz, 1998) encontró que la política de selección de emigrantes y el reemplazo de individuos existentes con los inmigrantes afecta a la velocidad de convergencia. En el estudio, los individuos que emigran de una subpoblación pueden ser seleccionados bien al azar, bien entre los mejores, o bien a través de políticas elitistas. Igualmente, los inmigrantes pueden reemplazar a los individuos al azar o eligiendo los peores cromosomas de la subpoblación receptora.

En esta Memoria hemos estudiado el efecto beneficioso del operador transducción sobre la conjugación. Es importante indicar que pese a realizar los experimentos con subpoblaciones o colonias bacterianas, nuestros protocolos u algoritmos no son o se encuadran dentro de los algoritmos DGA.

## **Capítulo 2**

### **Objetivos**

El objetivo principal del trabajo descrito en esta Memoria ha sido el diseño de un algoritmo genético no convencional inspirado en los mecanismos genéticos de la conjugación y la transducción bacterianas.

Con el fin de lograr este objetivo se han abordado diversas aproximaciones experimentales compendiadas en los siguientes objetivos parciales:

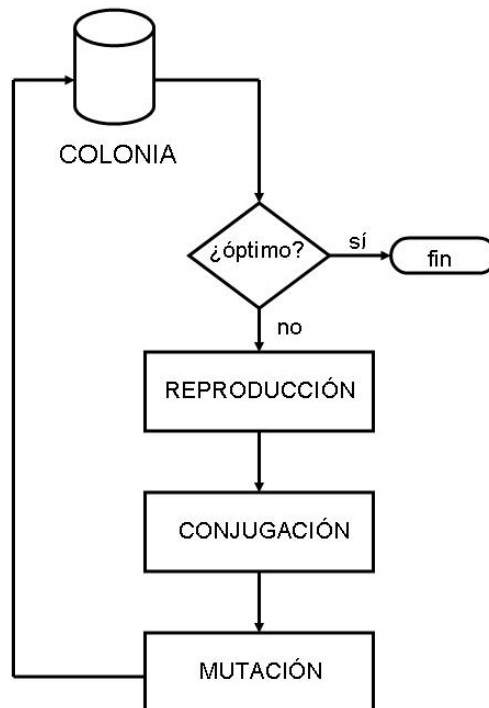
1. Diseño e implementación de un operador de conjugación bacteriana.
2. Integrar dicho operador junto con un operador mutación en un algoritmo genético bacteriano SBGA (*Simple Bacterial Genetic Algorithm*), estudiando la eficacia de distintos protocolos.
3. Evaluar la eficacia y utilidad del nuevo algoritmo SBGA en problemas de optimización, por ejemplo el diseño de un receptor de radio AM.
4. Diseño e implementación de un operador de transducción.
5. Integrar el nuevo operador de transducción en el algoritmo genético SBGA, obteniendo el algoritmo PETRI (*Programming Evolution Through Reiterated Infection*).
6. Evaluar la eficacia y utilidad del nuevo algoritmo PETRI en problemas de optimización, por ejemplo el diseño de un receptor de radio AM.

## **Capítulo 3**

### **Metodología**

En esta Sección se describe en una primera parte (3.1) el algoritmo genético bacteriano (SBGA) y los experimentos realizados para evaluar la eficacia de la conjugación en un problema de optimización. En la segunda parte (3.3), se introduce el algoritmo PETRI, esto es una extensión o ampliación del algoritmo SBGA a un modelo con subpoblaciones o colonias entre las que tiene lugar la transducción. Asimismo, se describen los experimentos realizados para evaluar el efecto positivo de la transducción durante la optimización. En ambos casos, la utilidad práctica de los algoritmos fue ilustrada con el diseño de un receptor de radio AM.

### 3.1. Algoritmo Genético Bacteriano (SBGA)



**Figura 3.1:** Diagrama de flujo del Algoritmo Genético Bacteriano (SBGA)

El algoritmo SBGA (*Simple Bacterial Genetic Algorithm*) o Algoritmo Genético Bacteriano es un procedimiento de optimización basado en el mecanismo de selección natural Darwiniana en una colonia de bacterias en las que se aplican los operadores genéticos conjugación y mutación.

El algoritmo propuesto (Fig. 3.1) consta de las siguientes etapas:

1. Obtención de manera aleatoria de una población de tamaño  $N$ , a la que se denomina generación 0.
2. Evaluación de los cromosomas (o bacterias) por medio de una función de evaluación o función *fitness*.
3. Aplicación del operador reproducción simulando la selección natural.
4. Aplicación del operador conjugación en parejas de bacterias.
5. Aplicación del operador mutación.
6. Repetir, una y otra vez, los pasos anteriores (2-5) hasta que se cumpla la condición de terminación, ya sea obteniendo uno o más cromosomas óptimos o llegando a un número de generaciones determinado.

A continuación, describiremos los operadores del algoritmo SBGA:

### 3.1.1. Operador reproducción

El operador reproducción tiene por finalidad establecer los cromosomas que serán seleccionados para formar parte de la siguiente generación. Uno de los algoritmos más populares es el método de la ruleta sesgada denominado originalmente como *Roulette Wheel Parents Selection* (Davis, 1991). Este algoritmo otorga mayor probabilidad de reproducción, y por tanto de pasar a la siguiente generación, a aquellos cromosomas que tras su evaluación con la función *fitness* hayan obtenido mejor puntuación con respecto al resto de la población.

La implementación del operador reproducción requiere los siguientes supuestos (Lahoz-Beltrá, 2004):

- La población tiene un tamaño  $N$ , manteniéndose constante de generación en generación.
- Se dispone de una función con la que evaluar la idoneidad o *fitness* de cada cromosoma.
- Los cromosomas dentro de la población son ordenados de acuerdo a un número de identificación representado de acuerdo con la siguiente notación #1,#2 ,..., #N.

El algoritmo para implementar la ruleta sesgada se describe a continuación:

1. Obtener para cada cromosoma  $x_i$  de la población su puntuación  $f(x_i)$  o *fitness*.

2. Calcular para cada cromosoma la puntuación o *fitness acumulado*, es decir, la suma de su puntuación y las puntuaciones de los cromosomas que le preceden, según el orden establecido:

$$fitness\ acumulado_j = \sum_{i=1}^j f(x_i) \quad (3.1)$$

3. Repetir los pasos 1 y 2 para todos los cromosomas (los  $N$  cromosomas de la población).
4. Obtener un número aleatorio  $U_i$  comprendido entre 0 y el *fitness total*, siendo este último valor la suma de las puntuaciones de todos los cromosomas de la población:

$$fitness\ total = \sum_{i=1}^N f(x_i) \quad (3.2)$$

5. El primer cromosoma, según el orden establecido, cuyo *fitness acumulado* supere o iguale al número aleatorio  $U_i$ , se reproducirá una vez:

$$R = \begin{cases} 1, & \text{si } \sum_{i=1}^j f(x_i) \geq U_i \\ 0, & \text{si } \sum_{i=1}^j f(x_i) < U_i \end{cases} \quad (3.3)$$

siendo  $R$  el número de copias o réplicas del cromosoma. Se repite el procedimiento descrito, una y otra vez, hasta obtener una población de tamaño  $N$ .

### 3.1.2. Operador conjugación

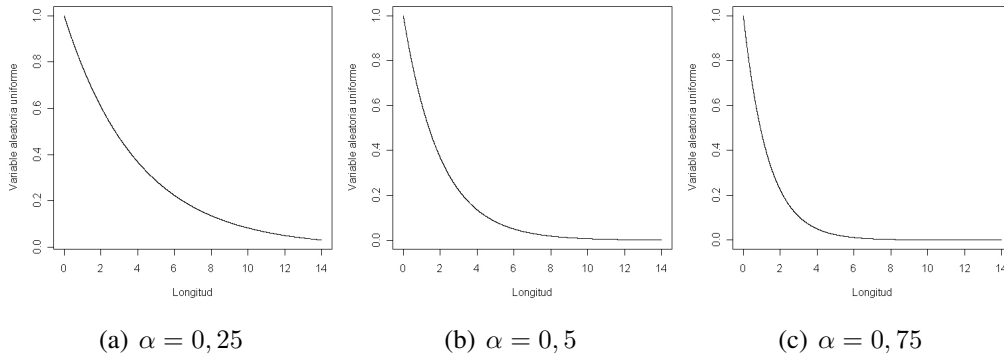
El operador representa una de las principales aportaciones originales introducida en esta Memoria. Se trata del procedimiento computacional con el que se simula el fenómeno de intercambio genético de conjugación bacteriana (Stanier *et al.*, 1986). Con dicho operador se seleccionan una bacteria donadora, una bacteria receptora y un fragmento cromosómico cuya longitud es simulada por el método de Monte Carlo. La conjugación puede ser simulada por medio de dos operadores, cuya diferencia no es otra que el criterio de selección del punto de inicio de la conjugación. En las dos versiones se asumen los siguientes supuestos:

1. Se selecciona aleatoriamente una pareja de bacterias, una donadora y otra receptora. En la bacteria donadora no se distingue si se trata de una bacteria  $F^+$  o Hfr.

- La longitud del fragmento cromosómico  $l$  se simula con el método de Monte Carlo asumiéndose que dicha longitud es una variable aleatoria con distribución exponencial negativa (Thai Dam, 1997; Lahoz-Beltrá, 2004), tal y como sugiere la función descrita por Wood (1968):

$$l = -\frac{1}{\alpha} \ln(U) \quad (3.4)$$

en la que  $U$  es un número aleatorio ( $0 < U \leq 1$ ) y  $\alpha$  el parámetro de la distribución o parámetro de la conjugación con el que simulamos aquellos factores que influyen en la conjugación (por ejemplo, agitación del medio, concentración de sales, temperatura, etc.).



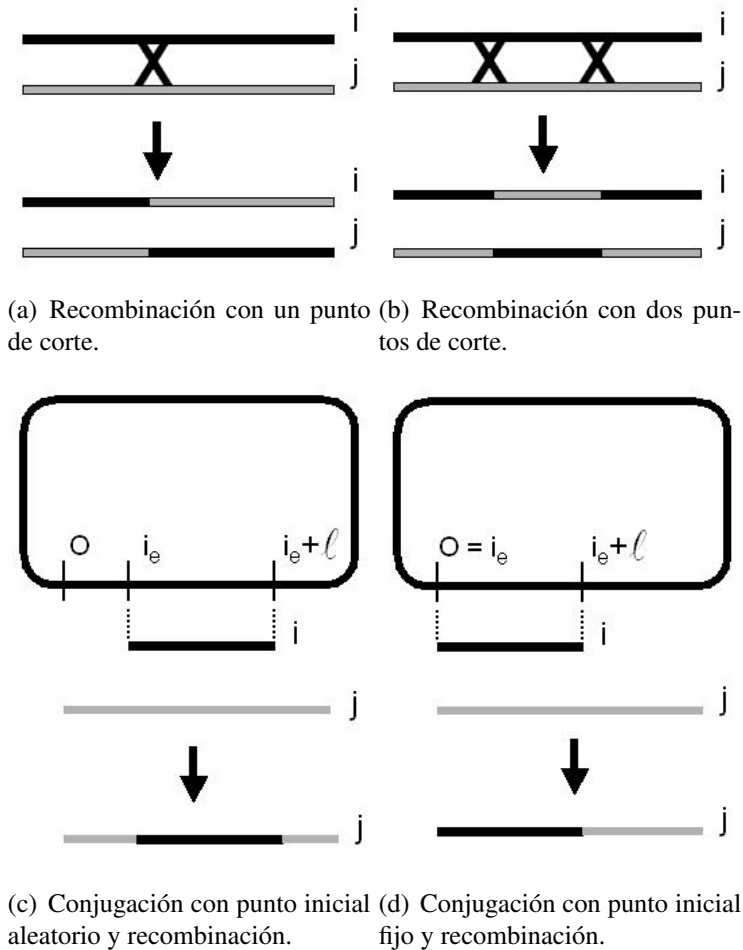
**Figura 3.2:** Simulaciones de  $l$  con el método de Monte Carlo

Obsérvese como (Fig. 3.2) a un mayor valor del parámetro  $\alpha$  de la distribución, menor será la longitud del fragmento cromosómico transferido.

La simulación de la conjugación bacteriana en los algoritmos genéticos SBGA se basa en dos modelos de conjugación de acuerdo con la elección del punto de inicio  $i_e$  de la transferencia: el operador **COFP** (*Conjugation Operator with a Fixed Point*), en el que el punto de inicio se fija arbitrariamente (Fig. 3.3(d)), y el operador denominado **CORP** (*Conjugation Operator with a Random Point*), en el que el punto de inicio es fijado al azar en cada una de las recombinaciones (Fig. 3.3(c)). Estas dos definiciones del operador se describen a continuación:

**CORP (Conjugation Operator with a Random Point)** El operador conjugación con punto de inicio aleatorio CORP se implementa con el siguiente algoritmo (Fig. 3.3(c)):

- Obtener aleatoriamente de la colonia de bacterias de tamaño  $N$  una pareja de bacterias (o cromosomas) donante y receptora, siendo  $i, j$  las bacterias donante y receptora, respectivamente.



**Figura 3.3:** Operadores de recombinación y conjugación:

- En 3.3(a) y 3.3(b):  $i$  y  $j$  son los cromosomas parentales. Los cromosomas superiores son los anteriores a la recombinación y los inferiores son los posteriores.
- En 3.3(c) y 3.3(d):  $O$  es el ORI del cromosoma,  $i_e$  el punto de inicio de la conjugación y  $i_e + l$  el punto final.  $i$  es el fragmento transferido del cromosoma donador y  $j$  el cromosoma receptor antes (superior) y después (inferior) de la conjugación

2. Elegir una probabilidad de conjugación  $p_c$ , y seleccionar un número aleatorio  $U_c$  en el intervalo  $[0, 1]$ . Se asume que la ocurrencia ( $C = 1$ ) o no ( $C = 0$ ) de conjugación es un suceso de Bernoulli, tal que:

$$C = \begin{cases} 1, & \text{si } U_c < p_c \\ 0, & \text{si } U_c \geq p_c \end{cases} \quad (3.5)$$

3. Si la conjugación tiene lugar ( $C = 1$ ) el mecanismo genético es simulado aplicándose el siguiente procedimiento:
  - a) Seleccionar aleatoriamente el punto de inicio  $i_c$  de la conjugación.
  - b) La longitud  $l$  del fragmento de cromosoma que es transferido desde la bacteria donadora  $i$  hasta la receptora  $j$  se obtiene aplicando la expresión (3.4).
  - c) La conjugación se simula sustituyendo el segmento transferido en el cromosoma de la bacteria receptora, entre los puntos cromosómicos  $i_c$  e  $i_c + l$ .
  - d) Repetir los pasos anteriores  $N$  veces, siendo  $N$  el tamaño poblacional o número de bacterias de la colonia.

**COFP (Conjugation Operator with a Fixed Point)** El algoritmo del operador conjugación con punto de inicio prefijado (Fig. 3.3(d)) es similar al algoritmo CORP, excepto en que el punto de inicio  $i_c$  de la conjugación es un punto fijo. Desde un punto de vista biológico se correspondería con aquel locus en el que se haya ubicado el primer gen del cromosoma bacteriano.

### 3.1.3. Operador mutación

El operador mutación es el operador más característico de los algoritmos evolutivos, ya que al carecer de operador recombinación, es el operador responsable de introducir variabilidad en la población de cromosomas. En el caso particular de los algoritmos genéticos su inclusión en el algoritmo es opcional, aplicándose por regla general a continuación de la recombinación, o la conjugación como será en nuestro caso. En el algoritmo SBGA el operador mutación tiene la misma finalidad que en otros procedimientos CE, asegurar que la probabilidad de encontrar una determinada región en el espacio problema cuyos puntos representan cromosomas con un alto grado de semejanza entre sí, nunca sea igual a cero, compensándose además la pérdida de información en la población que resulta de la eliminación de aquellos cromosomas con peores puntuaciones o *fitness* (Lahoz-Beltrá, 2004).

El operador, en su definición más sencilla, efectúa la inversión del valor de un dígito binario en un cromosoma cuyos genes son subcadenas en binario. Si los valores o alelos de los genes son números enteros o reales la mutación consistirá en el cambio de su valor por otro elegido aleatoriamente dentro del rango de valores establecidos para ese gen.

El algoritmo general del operador mutación consta de los pasos siguientes:

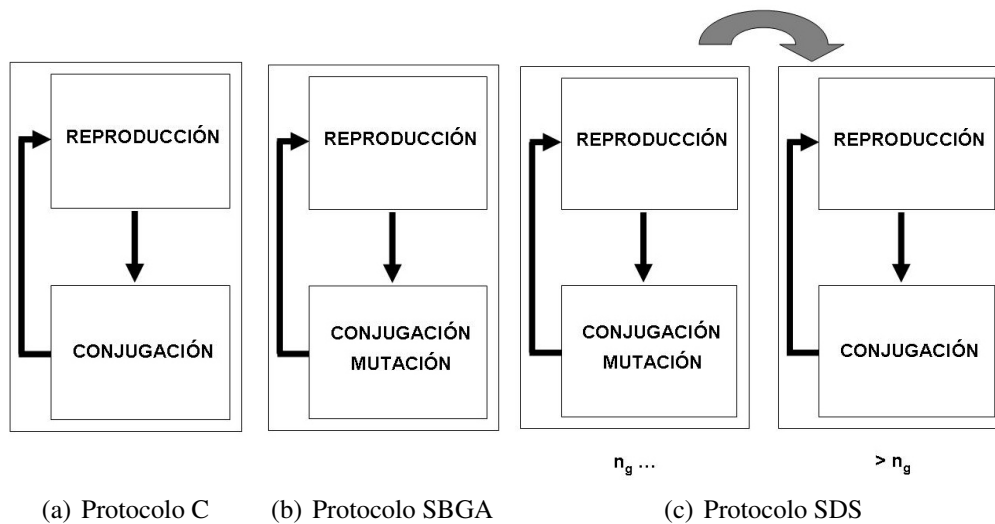
1. Elegir un valor de probabilidad de mutación  $p_m$ . Seleccionar a continuación un número aleatorio  $U_m$  en el intervalo  $[0, 1]$ . La mutación es simulada asumiendo que se trata de un suceso  $M$  de Bernoulli, y por tanto:

$$M = \begin{cases} 1, & \text{si } U_m < p_m' \\ 0, & \text{si } U_m \geq p_m' \end{cases} \quad (3.6)$$

2. Elegir un cromosoma aleatoriamente.
3. Elegir al azar un gen de ese cromosoma.
4. Cambiar el valor del gen elegido por otro valor obtenido aleatoriamente dentro del rango de valores posibles del gen.
5. Repetir los pasos anteriores  $N$  veces, siendo  $N$  el tamaño poblacional o de la colonia.

### 3.1.4. Protocolos experimentales

Los operadores conjugación y mutación pueden ser aplicados en el algoritmo SBGA de acuerdo con distintos protocolos en los experimentos de simulación.



**Figura 3.4:** Protocolos experimentales

1. El protocolo C, mostrado en la Fig. 3.4(a), es un algoritmo genético que no aplica el operador mutación en ninguna generación.

2. El protocolo SBGA, mostrado en la Fig. 3.4(b), es un algoritmo genético estándar (Goldberg, 1989), excepto en que se utiliza el operador conjugación en lugar del operador recombinación, incluyéndose el operador mutación en cada generación o ciclo de la simulación.
3. El protocolo SDS (*Simulated DNA Shuffling*), mostrado en la Fig. 3.4(c), fue introducido por Lahoz-Beltrá (2000, 2001) y Lahoz-Beltrá *et al.* (2002) inspirándose en los protocolos experimentales usados en laboratorio en ensayos de evolución *in vitro* de enzimas y péptidos con interés industrial en biotecnología (Harayama, 1998). El protocolo comprende un ciclo de mutación durante  $n_g$  generaciones simulando la inserción aleatoria de nucleótidos o *error-prone* PCR, y el ensamblaje de ADN por recombinación homóloga, seguido de la repetición de ciclos en ausencia de mutación.

## 3.2. Experimentos de evaluación de SBGA

Para evaluar la utilidad del algoritmo SBGA se realizaron una serie de experimentos de simulación en los que SBGA fue utilizado en un problema de optimización consistente en el diseño de un circuito analógico clásico, más concretamente el diseño de un receptor de radio AM.

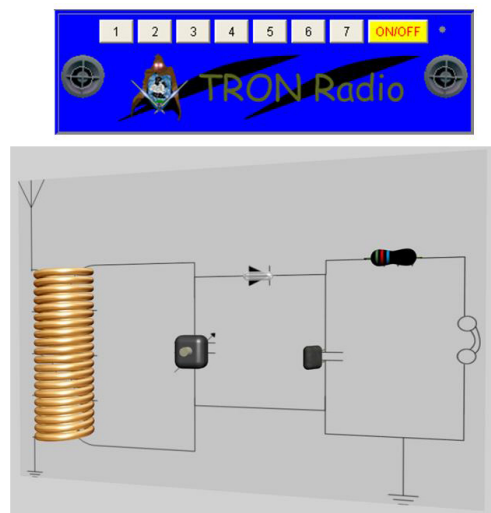
### 3.2.1. Fundamento del funcionamiento de un receptor de radio AM.

Un receptor de radio AM es un instrumento que recibe ondas de radio de Amplitud Modulada (A.M.) y emite a través de un altavoz o de auriculares la señal recibida (Fig. 3.5). Pero ¿cómo funciona una radio?

En primer lugar, la antena recibe las ondas electromagnéticas o señales de radio presentes en el aire que son conducidas a través de una bobina que induce una corriente eléctrica a otra bobina, denominada osciladora. La bobina osciladora se encuentra conectada a un condensador variable llamado dial. El circuito oscilador formado por la bobina osciladora o de sintonía y el dial entra en resonancia con la frecuencia característica, elegida por el radioyente, en la que emite una emisora (Tipler, 1993).

En el momento en el que una señal entra en resonancia con el oscilador, todas las demás frecuencias que se reciben por la antena se eliminan. A continuación, y dado que los altavoces requieren corriente continua, y como las ondas transmitidas por una emisora generan corriente alterna, será necesario transformar la señal a través de un diodo, dispositivo que pone a cero todo voltaje negativo dejando intactos los voltajes positivos (los signos de los voltajes dependen de la orientación

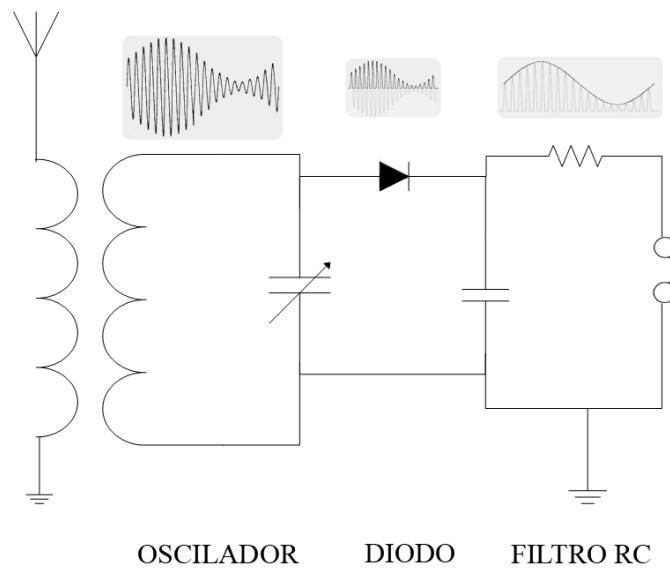
del diodo). A continuación, y dado que la onda seleccionada tiene una frecuencia muy superior a la que el oído humano es capaz de percibir, el circuito de radio extrae la señal audible codificada en esa onda de alta frecuencia mediante un filtro que crea una onda uniendo los picos de voltaje de la onda de alta frecuencia que recibe. El volumen de la señal puede modificarse utilizando un amplificador, no incluido en las simulaciones realizadas. De esta forma se consigue aumentar la amplitud de la onda sin modificar su frecuencia. Finalmente, y en el diseño de radio llevado a cabo en esta Memoria, la onda llega a unos auriculares en los que la onda eléctrica se convierte en onda sonora.



**Figura 3.5:** Carátula y circuito de radio AM

A continuación, se describen las fases existentes entre el momento en el que la antena capta la onda hasta su audición mediante unos auriculares (Fig. 3.6).

En primer lugar, y puesto que la antena no es selectiva, se recogen todas las frecuencias de radio que le llegan, incluyendo ruido eléctrico. Para discriminar entre unas estaciones u otras se utiliza un oscilador que consta de una bobina, también llamada bobina de sintonía, y un condensador variable conectado en paralelo, conocido popularmente como dial. El oscilador posee una frecuencia de resonancia que depende de las características de la bobina y el condensador. Por lo tanto, lo que se consigue al buscar una emisora de radio con el dial es variar la capacidad del condensador, y por lo tanto, la frecuencia de resonancia, sintonizando una emisora u otra. Si la frecuencia de la señal de radio que llega a través de la antena es igual a la frecuencia de resonancia o frecuencia del oscilador, la señal de radio se vuelve máxima, eliminándose o ignorándose otras señales de radio cuyas frecuencias estén por encima o debajo de aquella que ha sido “capturada”. En una radio ideal la frecuencia de resonancia del oscilador se calcula a partir de



**Figura 3.6:** Circuito y onda de radio AM

los valores de inductancia ( $L$ ) de la bobina y la capacidad ( $C$ ) del condensador variable de acuerdo con la expresión:

$$\nu_{resonancia} = \frac{1}{2\pi\sqrt{LC}} \quad (3.7)$$

Una bobina está constituida por un cable enrollado sobre un cilindro relleno por un núcleo material, por ejemplo ferrita o aire (utilizado en nuestros experimentos de simulación). La inductancia  $L$  de la bobina depende del material del núcleo, del número  $N$  de espiras o vueltas que tenga la bobina, la longitud  $l$  de la bobina y del área  $A$  que tenga la sección de la bobina, o en su defecto la de una de las espiras. En una radio ideal las características de la bobina mencionadas anteriormente se relacionan con la inductancia  $L$  a través de la expresión:

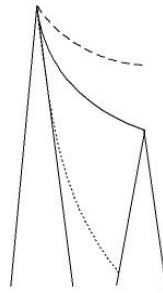
$$L = \mu_0 \frac{N^2}{l} A \quad (3.8)$$

donde  $\mu_0$  es la permeabilidad magnética del medio que para el aire es prácticamente igual a la del vacío, esto es  $\mu_0 \cong 4\pi \cdot 10^{-7} \text{ Wb/Am}$

A continuación, y en una segunda etapa, la señal de radio se filtra o rectifica por medio de un diodo. Este paso es básico, ya que los auriculares de la radio requieren corriente continua y la señal de radio captada con el oscilador es una corriente alterna. Una vez que la señal de radio pasa a través del diodo, se consigue eliminar la semionda negativa de la señal de radio, modificándose la semionda

positiva. Esta modificación dependerá en última instancia del material del que se compone el diodo.

En un caso ideal el diodo actuaría como un interruptor que permitiría el paso de la corriente únicamente en el caso de que el voltaje de entrada fuera positivo. En el caso de que el voltaje de entrada fuera negativo o igual a cero, el diodo no conduciría corriente alguna dejando el circuito abierto y el voltaje a cero. Sin embargo, en un circuito real, a pesar de que el diodo se comporte del modo indicado anteriormente para una corriente de entrada cuyo voltaje fuese cero o negativo, si el voltaje de entrada fuese positivo permitiría igualmente el paso de corriente siempre que el voltaje en cuestión igualara o superara un valor de voltaje umbral. En tal caso, el diodo actuaría como un circuito cerrado, dependiendo el voltaje umbral del material con el que esté construido el diodo (los más habituales son el germanio (umbral  $0,3 V$ ) y el silicio (umbral  $0,7 V$ )). En resumen, cuando el diodo se somete a una diferencia de potencial menor o igual al voltaje umbral, el voltaje de salida es cero. Por el contrario, si el diodo se somete a una diferencia de potencial superior al voltaje umbral, el voltaje de salida es igual al voltaje de entrada menos el voltaje umbral.



**Figura 3.7:** Descargas del filtro RC: Descarga adecuada (Línea sólida). Descarga lenta (Línea discontinua). Descarga rápida (Línea de puntos).

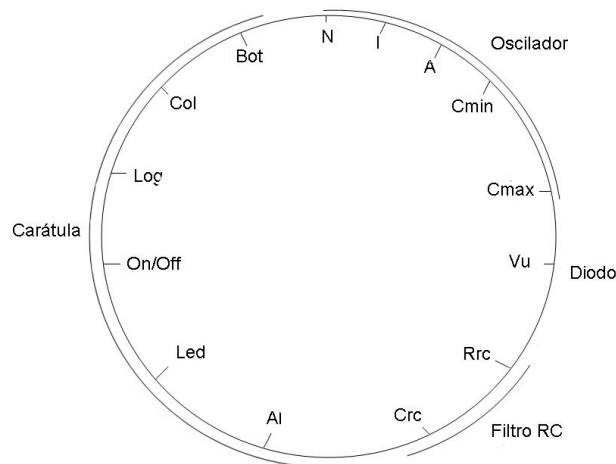
Finalmente, en la tercera y última etapa, y aunque esta etapa es opcional, la calidad de la señal de salida del diodo se mejora a través de un filtro RC. La salida del diodo es una señal de alta frecuencia que sería audible con baja calidad, pese a que los auriculares no pueden oscilar a esa frecuencia, ya que éstos responderían a los picos máximos de voltaje. El filtro RC, también llamado de paso bajo, se compone de una resistencia y de un condensador conectados en paralelo. El condensador se carga con un pico de voltaje hasta que el voltaje de la onda de alta frecuencia empieza a descender, descargándose de una forma característica en función de la capacidad ( $C$ ) y de la resistencia ( $R$ ), por lo que el voltaje también descenderá. En un radio ideal, el resultado audible debería ser la unión de los picos de alta frecuencia. La velocidad de carga y descarga del condensador es la propiedad que determina la calidad del filtro. Si la velocidad es alta, obtendremos

picos muy parecidos a los de la onda de alta frecuencia, mientras que si es lenta, el filtro no reaccionará a los cambios de amplitud de la señal de entrada. Ambas opciones son inadecuadas (Fig. 3.7). Por consiguiente, los valores de  $R$  y  $C$  se obtienen con una expresión empírica (ver 3.9) que relaciona estos valores con una frecuencia máxima o umbral por encima de la cual el filtro surte efecto:

$$\nu_{m\acute{a}xima} = \frac{1}{2\pi\sqrt{RC}} \quad (3.9)$$

Concluidas las tres fases o etapas descritas la señal de radio capturada por la antena podrá ser escuchada con los auriculares.

### 3.2.2. Codificación de las características del receptor de radio AM.



**Figura 3.8:** Cromosoma bacteriano usado para optimizar la radio AM.

Las características de los componentes electrónicos del receptor de radio AM se codificaron en un cromosoma circular con 14 genes (Fig. 3.8). En el cromosoma también se incluyeron aquellos genes que codifican las características de la carátula de la caja en cuyo interior se encontrará el circuito electrónico. Los valores de los genes o alelos son los que se recogen a continuación:

**Gen #1** Número de vueltas  $N$  del cable que forma la bobina de sintonía alrededor del núcleo (en nuestro caso aire). Rango de  $N$  igual a  $[1, 200] \in \mathbb{R}$  vueltas

- Gen #2** Longitud  $l$  de la bobina. Rango de  $l$  igual a  $[0,000001, 100] \in \mathbb{R}$  cm
- Gen #3** Área  $A$  de una espira de la bobina. Rango de  $A$  igual a  $[0,000001, 10000] \in \mathbb{R}$  cm<sup>2</sup>
- Gen #4** Capacidad mínima  $C_{min}$  del condensador variable. Rango de  $C_{min}$  igual a  $[0,000001, 20] \in \mathbb{R}$  pF
- Gen #5** Capacidad máxima  $C_{max}$  del condensador variable. Rango de  $C_{max}$  igual a  $[21, 200] \in \mathbb{R}$  pF
- Gen #6** Voltaje umbral  $V_u$  del diodo. Rango de  $V_u$  igual a  $[0, 10] \in \mathbb{R}$  V
- Gen #7** Resistencia  $R_{RC}$  del filtro RC. Rango de  $R_{RC}$  igual a  $[0,1, 10] \in \mathbb{R}$  k $\Omega$
- Gen #8** Capacidad  $C_{RC}$  del condensador del filtro RC. Rango de  $C_{RC}$  igual a  $[100, 10000] \in \mathbb{R}$  pF
- Gen #9** Número  $Al$  de altavoces. Rango de  $Al$  igual a  $[0, 3] \in \mathbb{N}$
- Gen #10** Presencia de  $LED$  indicativo de conexión. Es una variable Booleana 0/1.
- Gen #11** Presencia de botón de encendido  $On/Off$ . Es una variable Booleana 0/1.
- Gen #12** Índice  $Log$  de logotipo. Rango de  $Log$  igual a  $[0, 3] \in \mathbb{N}$
- Gen #13** Índice  $Col$  de color de la carcasa. Rango de  $Col$  igual a  $[0, 7] \in \mathbb{N}$
- Gen #14** Número  $Bot$  de botones de presintonía. Rango de  $Bot$  igual a  $[0, 7] \in \mathbb{N}$

### 3.2.3. Evaluación de los diseños de receptores de radio AM.

La evaluación de los cromosomas o diseños de receptores de radio AM en los experimentos de optimización fue realizada utilizándose la siguiente función *fitness*:

$$f = f_{oscilador} \cdot e^{P_D + P_{RC}} + f_{carátula} \quad (3.10)$$

La puntuación  $f_{oscilador}$  o *fitness* obtenido para el oscilador, elemento principal de la radio, se penaliza a través del término exponencial con los valores de  $P_D$  y  $P_{RC}$  obtenidos como resultado de la evaluación del diodo y del filtro RC del circuito electrónico. A esta puntuación se le suma la puntuación o *fitness*  $f_{carátula}$  que se obtiene de evaluar la carátula.

El cálculo del *fitness* del oscilador es efectuado teniendo en cuenta la función del oscilador: sintonizar el máximo número de emisoras que emiten en AM. Así pues, cuantas más frecuencias sea capaz de sintonizar, mayor será la puntuación, *fitness* o valor  $f_{oscilador}$ . Por ejemplo, en la actualidad el rango de frecuencias en las que se emite en Madrid capital se encuentra comprendido entre 500 y 1600 kHz. Por consiguiente, y para el ejemplo propuesto, el oscilador o radio óptimo será aquel que sea capaz de resonar en toda esa ventana de frecuencias. Si su rango fuera mayor no se traduciría en una puntuación mayor. El valor *fitness* del oscilador se calcula con la expresión:

$$f_{oscilador} = \nu_{max} - \nu_{min} \quad (3.11)$$

siendo  $\nu_{max}$  y  $\nu_{min}$  las frecuencias máximas y mínimas, respectivamente. Sus valores para el diseño codificado en un cromosoma se obtuvieron a partir de las expresiones que se muestran a continuación:

$$\nu_{max} = \min \left[ 1600, \frac{1}{2\pi \sqrt{\mu_0 \frac{N^2}{lA} C_{min}}} \right] = \quad (3.12)$$

$$= \min \left[ 1600, \frac{1}{2\pi \sqrt{4\pi \cdot 10^{-7} \frac{Gen_1^2}{Gen_2 10^{-2} Gen_3 10^{-4} Gen_4 10^{-12}}}} \right] \text{ Hz} \quad (3.13)$$

$$\nu_{min} = \max \left[ 500, \frac{1}{2\pi \sqrt{\mu_0 \frac{N^2}{lA} C_{max}}} \right] = \quad (3.14)$$

$$= \max \left[ 500, \frac{1}{2\pi \sqrt{4\pi \cdot 10^{-7} \frac{Gen_1^2}{Gen_2 10^{-2} Gen_3 10^{-4} Gen_5 10^{-12}}}} \right] \text{ Hz} \quad (3.15)$$

En el caso de que  $\nu_{min}$  estuviera por debajo de la frecuencia mínima en la que se emite en un cierto lugar, por ejemplo 500 kHz en el ejemplo anterior, se debería tomar este último valor, ya que al no existir para ese lugar emisiones de radio AM en esa frecuencia carece de utilidad que la radio sea capaz de resonar con frecuencias inferiores. Asimismo, se aplica el mismo criterio con  $\nu_{max}$  y la frecuencia máxima a la que se emite en un cierto lugar, por ejemplo 1600 kHz en el ejemplo de Madrid capital.

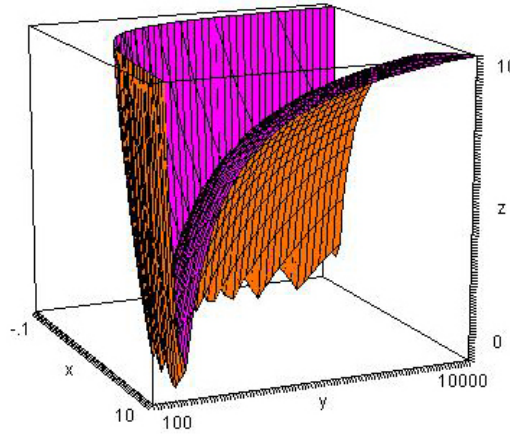
El término exponencial o de penalización que multiplica al valor  $f_{oscilador}$  se ha introducido al considerarse que un oscilador óptimo debe ir acompañado de una calidad de audio también óptima, calidad que depende del diodo y del filtro RC. Es decir, una radio cuyo oscilador sea óptimo pero no así la calidad de audición no es un buen diseño. De la misma manera, tampoco resulta útil cuando la

calidad de audio es óptima y el oscilador apenas sintoniza unas pocas emisoras. Por consiguiente, el término de penalización incluye dos exponentes: el primero de ellos relativo a la función desempeñada por el diodo y el otro relativo a la función llevada a cabo por el filtro RC.

El exponente  $P_D$  se define como la diferencia entre el voltaje ideal  $V_i$  y el voltaje umbral  $V_\theta$ , valor este último codificado en el  $Gen_6$  del cromosoma:

$$P_D = V_i - V_\theta = 0 - Gen_6 \quad (3.16)$$

De acuerdo con lo expuesto en el apartado 3.2.1, la situación óptima para el diodo será aquella en la que el voltaje umbral iguale al voltaje ideal, evitando en tal caso la distorsión de la señal recibida del oscilador. En consecuencia, cuanto mayor sea la diferencia entre ambos voltajes mayor será la penalización  $P_D$  relativa al diodo.



**Figura 3.9:** Representación gráfica del filtro RC escalado

De un modo similar al diodo, un filtro RC de baja calidad empeorará la señal que recorre el circuito electrónico de la radio. En los experimentos de simulación, el filtro óptimo es aquel que es ideal para un valor de frecuencia de 16 kHz, que es el valor máximo de frecuencia audible por todo oído humano (frecuencias superiores pueden ser oídas por algunas personas). Teniendo en cuenta este hecho, podremos definir el exponente  $P_{RC}$  a partir de la siguiente expresión:

$$P_{RC} = \max \left[ P_{D_{min}}, -w \left| v_{ideal} - \frac{1}{2\pi RC} \right| \right] = \quad (3.17)$$

$$= \max \left[ -10, -\frac{10}{14409} \left| 1,6 \cdot 10^4 - \frac{1}{2\pi Gen_7 10^3 Gen_8 10^{-12}} \right| \right] \quad (3.18)$$

siendo  $w$  el factor de escala. El factor de escala se incluye en la expresión para considerar la forma de la función del filtro RC (Fig. 3.9) con el fin de asemejar la penalización del filtro a la causada por el alejamiento durante el proceso de diseño y optimización de la radio AM con respecto al diodo óptimo. En el caso de que la penalización del filtro supere el valor máximo de la penalización relativa al diodo, la penalización del filtro se igualaría a la de aquel, evitándose así que valores de penalización muy elevados corrompiesen la simulación.

La puntuación  $f_{carátula}$  o *fitness* obtenido para la carátula, elemento accesorio de la radio e independiente del circuito electrónico, se calcula mediante la siguiente expresión:

$$f_{carátula} = w' \sum_{i=1}^6 p_i \quad (3.19)$$

siendo  $w'$  el factor de escala. En las simulaciones  $w'$  se fijó en 1100000/146, introduciéndose este factor en la expresión con el fin de equiparar el peso de la puntuación, o *fitness* de la carátula, a la puntuación o *fitness* aportado por el circuito electrónico.



**Figura 3.10:** Etiquetas decorativas de la carátula

De acuerdo con la expresión (3.20) los valores  $p_1, p_2, p_3, p_4, p_5$  y  $p_6$  expresan la puntuación arbitraria recibida en función de las siguientes características de la carátula:  $i = 1$ , número de altavoces (0, ..., 3);  $i = 2$ , ausencia (0) o presencia (1) de un diodo LED en la carátula que indique el estado encendido o apagado de la radio;  $i = 3$ , la presencia (1) o ausencia (0) de un interruptor de encendido/apagado de la radio;  $i = 4$ , el diseño de una etiqueta decorativa o distintivo para el frontal de la carátula de la radio entre cuatro posibles (Fig. 3.10);  $i = 5$ , el color de la carátula de la radio (recibiendo la mayor puntuación los colores primarios, una puntuación media los colores secundarios, correspondiendo la menor puntuación al blanco y al negro. Finalmente se puntuaría con un cero al color rosa); finalmente para  $i = 6$ , número de botones de presintonía (0, ..., 7) aumentando la puntuación con el número de botones y por tanto de emisoras de AM cuya frecuencia de señal pueda ser memorizada.

$$p_i = \left\{ \begin{array}{l} i = 1, Gen_9 \left\{ \begin{array}{l} 0, 0 \\ 1, 20 \\ 2, 30 \\ 3, 10 \end{array} \right. \\ \\ i = 2, Gen_{10} \left\{ \begin{array}{l} 0, 0 \\ 1, 20 \end{array} \right. \\ \\ i = 3, Gen_{11} \left\{ \begin{array}{l} 0, 0 \\ 1, 20 \end{array} \right. \\ \\ i = 4, Gen_{12} \left\{ \begin{array}{l} 0, 5 \\ 1, 15 \\ 2, 20 \\ 3, 35 \end{array} \right. \\ \\ i = 5, Gen_{13} \left\{ \begin{array}{l} 0, 19 \\ 1, 20 \\ 2, 18 \\ 3, 0 \\ 4, 15 \\ 5, 14 \\ 6, 5 \\ 7, 6 \end{array} \right. \\ \\ i = 6, Gen_{14} \left\{ \begin{array}{l} 0, 0 \\ 1, 3 \\ 2, 6 \\ 3, 9 \\ 4, 12 \\ 5, 15 \\ 6, 18 \\ 7, 21 \end{array} \right. \end{array} \right. \quad (3.20)$$

### 3.2.3.1. Experimentos de simulación

En los experimentos realizados con el algoritmo SBGA se definió una población de cromosomas (o bacterias) en la que cada cromosoma representa un posible diseño de receptor de radio AM.

Los experimentos se realizaron en tres lotes (Tabla 3.1). En el primer lote se realizaron experimentos de optimización del circuito oscilador; en el segundo lote del circuito oscilador, los filtros y la carátula de la radio AM, es decir de la radio AM completa, y en el tercer lote se estudió el efecto del tamaño poblacional o del número de bacterias en la optimización de la radio AM completa.

En cada uno de los lotes se realizaron a su vez ensayos con las dos modalidades de conjugación (COFP y CORP) que fueron comparados con ensayos efectuados con las dos modalidades de recombinación clásica en algoritmos genéticos, esto es recombinación con un punto de corte (Fig. 3.3(a)) y recombinación con dos puntos de corte (Fig. 3.3(b)) descritas en (Lahoz-Beltrá, 2004).

En los tres lotes de experimentos la probabilidad de conjugación  $p_c$  del algoritmo genético bacteriano, y el de recombinación en el algoritmo genético estándar, fue igual al 75 % en ambos casos, eligiéndose para la conjugación un parámetro de conjugación  $\alpha$  en el rango  $[0,001, 1]$ . Además, las simulaciones se realizaron durante un tiempo total de simulación de  $t = 200$  iteraciones o ciclos, siendo el número de réplicas de cada experimento igual a 50.

En los dos primeros lotes también se simularon los distintos protocolos experimentales del apartado 3.1.4 con una probabilidad de mutación  $p_m$  de 5 %, no simulándose la mutación en el tercer lote. Por el contrario, en el tercer lote se llevaron a cabo experimentos de simulación bajo diferentes tamaños poblacionales o de colonia. Es este caso los experimentos de simulación fueron realizados con 200, 300, 400, 500, 600, 700 y 800 bacterias (o cromosomas), mientras que en los dos primeros lotes el tamaño poblacional fue de 500 cromosomas o bacterias.

	Oscilador	Radio
<b>C</b>	CORP, COFP, 1Pto, 2Pto	CORP, COFP, 1Pto, 2Pto
		Tamaño poblacional
<b>SBGA</b>	CORP, COFP, 1Pto, 2Pto	CORP, COFP, 1Pto, 2Pto
<b>SDS</b>	CORP, COFP, 1Pto, 2Pto	CORP, COFP, 1Pto, 2Pto

**Tabla 3.1:** Resumen de experimentos realizados con SBGA

### 3.2.3.2. Análisis estadístico

Para comparar la eficiencia de los operadores de conjugación y recombinación se usó la medida de eficacia *off-line* de De Jong (De Jong, 1975), representada por

la ecuación:

$$x_e^*(s) = \frac{1}{T} \sum_{t=1}^T f_e^*(t) \quad (3.21)$$

donde  $e$  y  $s$  son el ambiente y el protocolo,  $T$  el número total de generaciones,  $t$  la generación actual y  $f_e^*(t)$  es el mejor *fitness* en cada generación. Al realizarse 50 repeticiones, se obtuvieron (con  $T = 200$ ) 50 medidas de evaluación por cada protocolo de algoritmo genético.

A continuación, y en cada uno de los protocolos (C, SBGA, SDS), se realizó un test Kruskal-Wallis (Kruskal y Wallis, 1952) para comprobar el supuesto de que las medianas de las medidas *off-line* de De Jong obtenidas bajo diferentes operadores genéticos (CORP, COFP, recombinación con un punto de corte y con dos puntos de corte) eran iguales. Las pruebas estadísticas se completaron con el método gráfico exploratorio de Múltiple Box-and-Whisker (Tukey, 1977), determinando qué medianas eran diferentes significativamente.

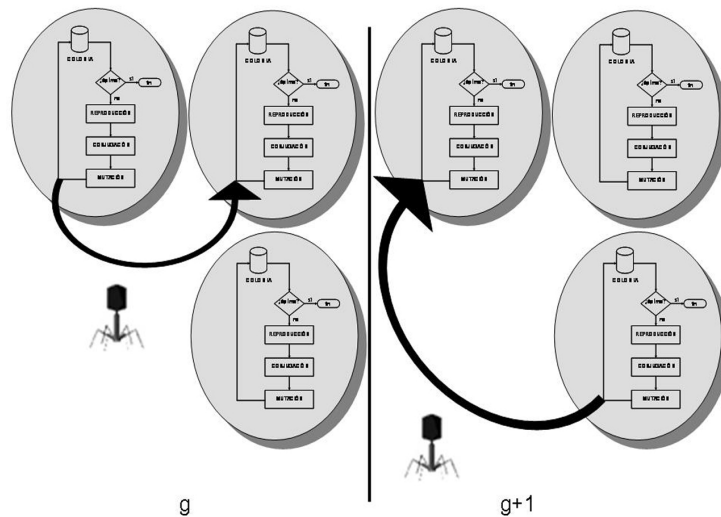
La capacidad de conducir a la población a soluciones óptimas de un operador genético se evaluó como se describe a continuación: en primer lugar, los valores de la media de los *fitness* medios en la última generación (en particular la generación 200) se calculó para las 50 repeticiones de cada uno de los operadores genéticos y protocolos de algoritmos genéticos. En segundo lugar, se crearon gráficos Multiple Box-and-Whisker (Tukey, 1977) de esos valores considerando los diferentes operadores genéticos en cada uno de los protocolos. Nótese que la eficiencia de los operadores aumenta cuando aumenta el valor medio MAF, es decir, cuando el valor medio MAF se desplaza hacia la derecha dentro de la caja. La caja representa el rango intercuartílico de los valores MAF en el gráfico Box-and-Whisker.

Además, como la longitud de la caja que representa el rango intercuartílico es una medida de variabilidad, si se tienen dos operadores genéticos, aquél cuya caja tenga la mayor longitud será el operador que consiga llevar a la población a un mayor número de soluciones óptimas. Este hecho puede estar relacionado con el papel del operador genético en el mantenimiento o promoción de la diversidad poblacional, uno de los principales requerimientos de la selección natural Darwiniana.

### 3.3. PETRI: Un Algoritmo Genético Bacteriano con transducción

El algoritmo que introducimos a continuación ha sido denominado PETRI (abreviatura del inglés *Programming Evolution Through Reiterated Infection*). Es la ampliación del algoritmo SBGA (ver apartado 3.1) incluyéndose junto con la conjugación bacteriana un segundo mecanismo de transferencia horizontal de

genes: la transducción. En la transducción se transferirán cromosomas o fragmentos cromosómicos desde una colonia, subpoblación o placa Petri a otra. Por consiguiente, obsérvese como mientras que en la conjugación la transferencia de información es *dentro* de una colonia, en la transducción la transferencia tendrá lugar *entre* colonias.



**Figura 3.11:** Esquema del algoritmo PETRI

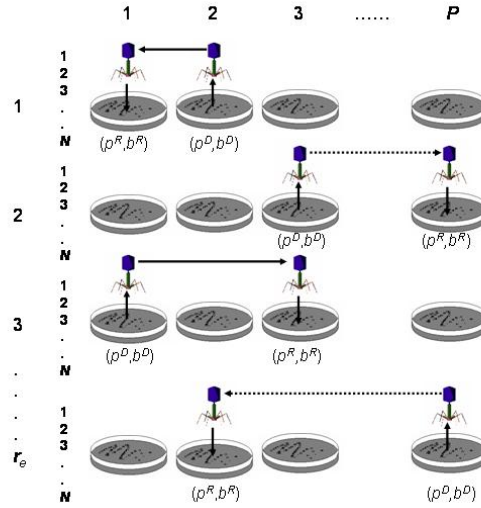
El algoritmo propuesto (Fig. 3.11) consta de las siguientes fases:

1. Obtener  $P$  subpoblaciones o colonias, estando cada colonia gobernada por el algoritmo SBGA.
2. Asumiendo que el tiempo de generación se haya sincronizado en las  $P$  colonias, aplicar en cada generación el operador transducción al conjunto de colonias. El operador será aplicado concluida la aplicación de los mecanismos de la conjugación y/o mutación en las colonias.
3. Ejecutar el algoritmo SBGA en cada población.
4. Repetir, una y otra vez, los pasos anteriores (2-3) hasta que se cumpla la condición de terminación, ya sea obteniendo uno o más cromosomas óptimos o llegando a un número de generaciones determinado.

### 3.3.1. Operador transducción

El operador transducción representa otra de las aportaciones descritas en la presente Memoria. El mecanismo de la transducción es simulado considerando

tres etapas consecutivas (Fig. 3.12): la elección de la placa y la bacteria donadoras, la elección de la placa y la bacteria receptoras, y la elección del cromosoma o fragmento transferido.



**Figura 3.12:** Esquema de experimentos con el algoritmo PETRI:  $p^D$  y  $b^D$  son la placa y la bacteria donadoras y  $p^R$  y  $b^R$  la placa y la bacteria receptoras.  $N$  es el número total de generaciones y  $r_e$  el número de réplicas

Sea  $b$  una bacteria  $(1, \dots, j, \dots, N)$  y  $p$  una placa Petri  $(1, \dots, i, \dots, P)$ , la elección de la bacteria y placa donadoras tiene lugar por el siguiente procedimiento:

**Elección de donadores** La placa Petri donadora  $p^D$  es aquella de la que se selecciona la bacteria donadora  $b^D$ . La placa  $p^D$  se elige siguiendo uno de los siguientes criterios:

1. Selección al azar (método r): En este caso, la placa Petri  $p^D$  se elige al azar de acuerdo con una distribución uniforme con rango  $[1, P]$ .
2. *Fitness* máximo (método max): Se elige la placa que tenga el mayor *fitness* de  $P$  placas tal que  $\max_{1, \dots, P} \{ \bar{f}_1, \bar{f}_2, \dots, \bar{f}_i \}$  siendo  $\bar{f}_i$  el valor *fitness* de la  $i$ -ésima placa Petri.
3. *Fitness* medio (método ave): Se elige la placa más representativa, cuyo valor  $\bar{f}_i$  sea  $\min_{1, \dots, P} | \bar{f}_i - \bar{f}_P | \geq 0$ , siendo  $\bar{f}_P$  el *fitness* medio de las placas.

4. Ruleta (método roul): En este caso, se elige  $p^D$  girando una ruleta que asigna a cada placa un sector cuyo arco es proporcional a su valor *fitness*. Este método es similar al usado en el operador reproducción (ver 3.1.1).

Una vez elegida la colonia o placa Petri donadora, la elección de la bacteria donadora  $b^D$  de esa colonia se basa en uno de los siguientes criterios:

1. Selección al azar (método r): En este caso, la bacteria  $b^D$  se elige al azar en una distribución uniforme con rango  $[1, N]$ .
2. *Fitness* máximo (método max): Se elige la bacteria que tenga el mayor *fitness* de  $N$  bacterias tal que  $\max_{1, \dots, N} \{\bar{f}_1, \bar{f}_2, \dots, \bar{f}_j\}$  siendo  $\bar{f}_j$  el valor *fitness* de la  $j$ -ésima bacteria.
3. *Fitness* medio (método ave): Se elige la bacteria más representativa, cuyo valor  $\bar{f}_j$  sea  $\min_{1, \dots, N} |\bar{f}_j - \bar{f}_B| \geq 0$ , siendo  $\bar{f}_B$  el *fitness* medio de las bacterias de la placa.
4. Ruleta (método roul): En este caso, se elige  $b^D$  girando una ruleta que asigna a cada bacteria un sector cuyo arco es proporcional a su valor *fitness*.

**Elección de receptores** La placa Petri receptora  $p^R$  es aquella de la que se selecciona la bacteria receptora  $b^R$ . La placa receptora  $p^R$  se elige siguiendo uno de los siguientes criterios:

1. Selección al azar (método r): En este caso, la placa Petri  $p^R$  se elige al azar de acuerdo a una distribución uniforme con rango  $[1, P]$ .
2. *Fitness* mínimo (método min): Se elige la placa que tenga el menor *fitness* de  $P$  placas tal que  $\min_{1, \dots, P} \{\bar{f}_1, \bar{f}_2, \dots, \bar{f}_i\}$  siendo  $\bar{f}_i$  el valor *fitness* de la  $i$ -ésima placa Petri.
3. *Fitness* medio (método ave): Se elige la placa más representativa, cuyo valor  $\bar{f}_i$  sea  $\min_{1, \dots, P} |\bar{f}_i - \bar{f}_P| \geq 0$ , siendo  $\bar{f}_P$  el *fitness* medio de las placas.
4. Ruleta inversa (método inv-roul): En este caso, se elige  $p^R$  girando una ruleta que asigna a cada placa un sector cuyo arco es proporcional a la inversa de su valor *fitness*, es decir  $1/\bar{f}_i$ .

Una vez elegida la colonia o placa Petri receptora, la elección de la bacteria receptora  $b^R$  de esa colonia se basa en uno de los siguientes criterios:

1. Selección al azar (método r): En este caso, se elige la bacteria  $b^R$  al azar de acuerdo a una distribución uniforme con rango  $[1, N]$ .
2. *Fitness* mínimo (método min): Se elige la bacteria que tenga el menor *fitness* de  $N$  bacterias tal que  $\min_{1, \dots, N} \{\bar{f}_1, \bar{f}_2, \dots, \bar{f}_j\}$  siendo  $\bar{f}_j$  el valor *fitness* de la  $j$ -ésima bacteria.
3. *Fitness* medio (método ave): Se elige la bacteria más representativa, cuyo valor  $\bar{f}_j$  sea  $\min_{1, \dots, N} |\bar{f}_j - \bar{f}_B| \geq 0$ , siendo  $\bar{f}_B$  el *fitness* medio de las bacterias de la placa.
4. Ruleta inversa (método inv-roul): En este caso, se elige  $b^R$  girando una ruleta que asigna a cada bacteria un sector cuyo arco es proporcional a la inversa de su valor *fitness*, es decir  $1/\bar{f}_j$ .

**Elección del material genético a transferir** Una vez elegidas las placas y bacterias participantes en la transducción, se debe seleccionar el cromosoma o el segmento cromosómico que va a ser transferido desde  $b^D$  hasta  $b^R$  a través de un virus o bacteriófago simulado. Si la transferencia fuera de un segmento cromosómico entonces la selección del segmento se efectuaría eligiendo un valor aleatorio  $s$  de una distribución uniforme con rango  $[1, F]$ , siendo  $F$  el número de fragmentos o segmentos en los que se divide el cromosoma.

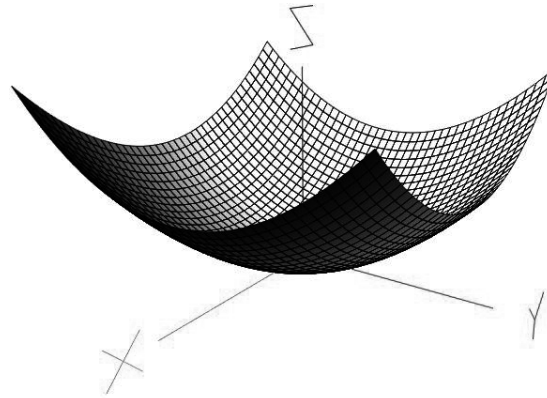
### 3.4. Experimentos de evaluación de PETRI

Con el fin de evaluar la calidad del algoritmo PETRI se realizaron una serie de experimentos de simulación en los que se utilizó el algoritmo en diversos problemas de optimización. En primer lugar, se realizaron dos experimentos de optimización con dos funciones de evaluación estándar: la función esférica y la función Rastrigin. En segundo lugar, PETRI fue evaluado con el problema que ya resolviéramos con SBGA, consistente en el diseño de un receptor de radio AM.

#### 3.4.1. Función esférica

La función esférica es una función de evaluación para problemas de optimización definiéndose como la suma de los valores de una variable (o genes en los experimentos) al cuadrado:

$$f_x = \sum_{i=1}^n x_i^2 \quad (3.22)$$



**Figura 3.13:** Representación gráfica de función esférica.

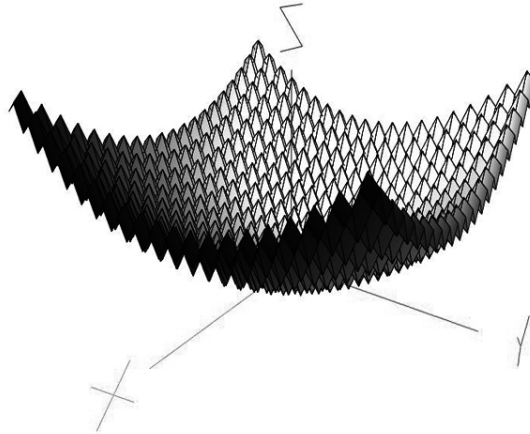
En los experimentos de simulación con esta función, que sólo tiene un óptimo, se utilizó un valor de  $n = 12$  y un rango de los valores génicos en el intervalo  $[-100, 100]$ .

Los experimentos de simulación se llevaron a cabo con 12 genes por cromosoma bacteriano, 200 bacterias en cada placa, 4 placas Petri y 15 repeticiones por cada tipo de experimento. Las simulaciones se realizaron durante 200 generaciones con el algoritmo SBGA y protocolo SDS, siendo la probabilidad de conjugación  $p_c$  igual a 75 % y la probabilidad de mutación  $p_m$  igual a 5 %. Se realizaron dos clases de experimentos en función de que la transducción implicase la transferencia del cromosoma completo o la transferencia de segmentos cromosómicos. Los segmentos fueron definidos como tres bloques funcionales con el mismo número de genes:

- Bloque 1: Genes del 1 al 4 incluidos
- Bloque 2: Genes del 5 al 8 incluidos
- Bloque 3: Genes del 9 al 12 incluidos

Dado que experimentos preliminares indicaron que la mejor eficacia en la transducción se conseguía con los criterios max-max para la elección de placa y bacteria donadoras ( $p^D, b^D$ ) y con los criterios roul-r para la elección de la placa y bacteria receptoras ( $p^R, b^R$ ), se utilizaron estos criterios de selección con distintas probabilidades de transducción  $p_t$ : 0 % (experimento control sin transducción), 25 %, 50 %, 75 % y 100 %.

### 3.4.2. Función Rastrigin



**Figura 3.14:** Representación gráfica de función Rastrigin.

La función Rastrigin es un ejemplo típico de función multimodal no lineal con expresión:

$$f_x = \sum_{i=1}^n x_i^2 - 10 \cdot \cos(2\pi x_i) + 10 \quad (3.23)$$

Fue propuesta por Törn y Zilinskas (1989) como una función bidimensional y fue generalizada por Mühlenbein *et al.* (1991). Esta función tiene un amplio espacio de búsqueda y un gran número de mínimos locales.

En los experimentos de simulación se utilizó un valor de  $n = 12$ , fijándose los rangos de los valores genéticos en el intervalo  $[-10, 10]$ .

Se realizaron los mismos ensayos que con la función esférica, utilizándose los mismos criterios, probabilidades y procedimientos.

### 3.4.3. Diseño de un receptor de radio AM

Encontrar el diseño óptimo de una radio AM como problema de optimización fue descrito en el apartado 3.2.1. La función *fitness* utilizada es la expresión (3.10), tal y como se recuerda a continuación:

$$f = f_{oscilador} \cdot e^{P_D + P_{RC}} + f_{carátula}$$

Los condiciones experimentales del algoritmo SBGA fueron similares a los elegidos en el apartado 3.2.3.1 para el protocolo SDS. Se fijó un valor del parámetro de conjugación  $\alpha$  igual a 0,5 y se utilizaron 9 placas por experimento.

Se realizaron distintos ensayos donde se estudió cuál de los criterios de selección de bacterias y placas, tanto donantes como receptoras, descritos en el apartado 3.3.1, mejoraba la eficacia de la conjugación. Para simplificar el número de posibles experimentos de simulación se restringió la selección de la bacteria a ser igual a la selección de la placa que la contenía. Para las placas y bacterias donadoras se utilizaron los siguientes criterios de selección: la selección al azar, la selección por *fitness* máximo, por *fitness* medio y por ruleta nombrándose como r-r, max-max, ave-ave y roul-roul, respectivamente. De manera similar, para las placas y bacterias receptoras los criterios fueron la selección al azar, por *fitness* mínimo, por *fitness* medio y por ruleta inversa nombrándose como r-r, min-min, ave-ave y roul.inv.-roul.inv., respectivamente. Además se ensayaron algunas combinaciones de selección que no cumplían la restricción anterior.

Se realizaron varios experimentos transduciendo fragmentos con distintas políticas de selección y posteriormente, cuando se encontró la mejor política, se realizó un ensayo donde se transducía el cromosoma completo. Los segmentos cromosómicos que podían ser transducidos se definieron como tres bloques funcionales:

- Bloque 1: Genes del oscilador
- Bloque 2: Genes del filtro RC y del diodo
- Bloque 3: Genes de la carátula

En la tabla (Tabla 3.2) se resumen los experimentos realizados en función de la política de selección empleada.

$p^R-b^R$ \ $p^D-b^D$	max-max	rou-rou	r-r	min-min	ave-ave	r.inv-r.inv
rou-rou	X	X	X	X	X	X
rou-r	X					
rou-min	X					
rou-ave	X					
rou-max	X					
rou-r.inv	X					
max-rou	X					
max-r	X					
max-min	X					
max-ave	X					
max-r.inv	X					
max-max	X	X				

*continúa en la página siguiente*

continuación de la página anterior							
$p^R-b^R$	$p^D-b^D$	max-max	rou-rou	r-r	min-min	ave-ave	r.inv-r.inv
	<b>r-r</b>		X	X			
	<b>min-min</b>		X		X		
	<b>ave-ave</b>		X			X	
	<b>r.inv-r.inv</b>		X				X

**Tabla 3.2:** Resumen de experimentos realizados con PETRI. Las políticas de selección aplicadas están marcadas con x.

### 3.4.4. Análisis estadístico

Una vez llevados a cabo los experimentos de simulación, se evaluó su eficacia tal y como se comenta a continuación. En primer lugar, se calculó la media de los valores *fitness* medios ( $MAF$ ) en la última generación (en particular la generación 200, ver (Perales-Graván y Lahoz-Beltrá, 2007)) para el total de placas Petri ( $MAF_T$ ) como se muestra a continuación:

$$MAF_T = \frac{\sum_{k=1}^{P \cdot r_e} \bar{f}_k}{P \cdot r_e} \quad (3.24)$$

siendo  $P$  el número total de placas Petri,  $r_e$  el número de réplicas del experimento y  $T$  el número total de placas en los experimentos, por lo que  $T = P \cdot r_e$ . Este valor fue el único calculado para los problemas de optimización con funciones objetivo (esférica y Rastrigin). Además, en el problema de optimización del receptor de radio AM también se calculó el valor  $MAF$  en la última generación (en particular la 200) para las placas Petri donantes ( $MAF_D$ ), las placas Petri receptoras ( $MAF_R$ ) y para aquellas que no son ni donantes ni receptoras ( $MAF_{RD}$ ):

$$MAF_D = \frac{\sum \bar{f}_D}{D} \quad (3.25)$$

$$MAF_R = \frac{\sum \bar{f}_R}{R} \quad (3.26)$$

$$MAF_{RD} = \frac{\sum_{k=T-D-R} \bar{f}_k}{T - D - R} \quad (3.27)$$

siendo  $D$  el número de placas Petri donadoras y  $R$  el número de placas Petri receptoras. Como se mencionó con anterioridad en los experimentos con la función

esférica y con la función Rastrigin:  $P = 4$ ,  $r_e = 15$  y  $T = 60$ . Sin embargo, en las simulaciones consistentes en encontrar el diseño óptimo de un receptor de radio AM:  $P = 9$ ,  $r_e = 50$  y  $T = 450$ . Nótese que en todos los experimentos de optimización  $D = R$  siendo  $D$  y  $R$  iguales a 1, es decir, se selecciona sólo una placa Petri donadora y una receptora por generación.

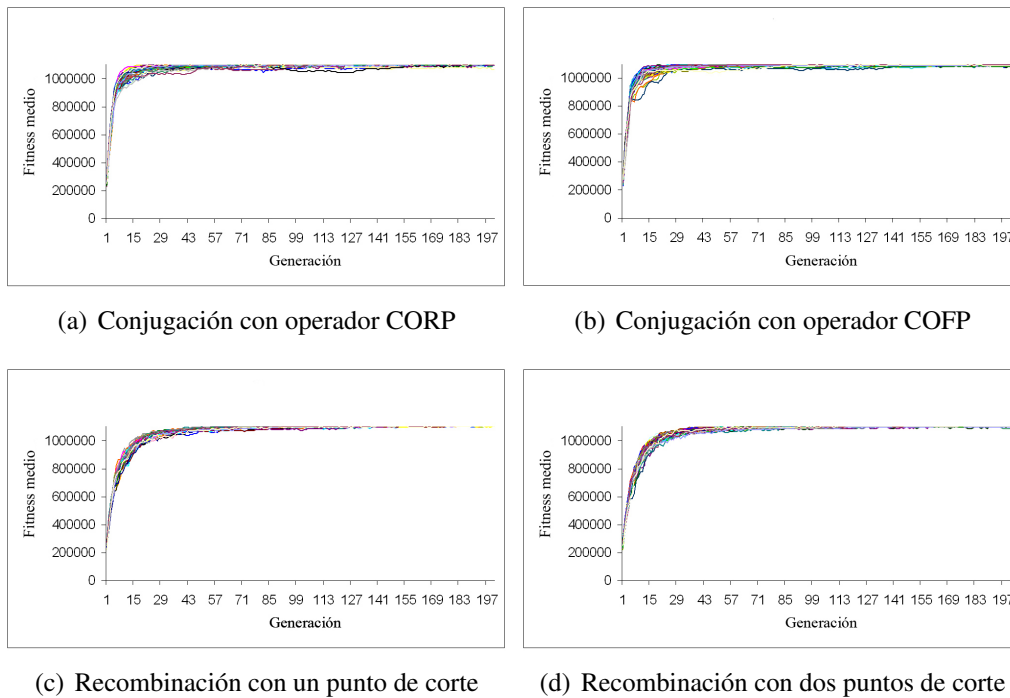
En segundo lugar, se crearon gráficos Multiple Box-and-Whisker de los valores obtenidos (Tukey, 1977) considerando que cada uno de los gráficos representa un protocolo experimental. Nótese que la eficiencia del protocolo aumenta cuando el valor  $MAF$  medio aumenta, es decir, cuando el valor medio  $MAF$  se desplaza hacia arriba en el interior de la caja. La caja representa el rango intercuartílico de los valores  $MAF$  en el gráfico Box-and-Whisker. Además, como la longitud de la caja que representa el rango intercuartílico es una medida de variabilidad, cuando se comparan dos políticas de transducción o dos protocolos, aquel cuya caja tenga la mayor longitud será el que lleve a la población a alcanzar un mayor número de soluciones óptimas.

# **Capítulo 4**

## **Resultados**

## 4.1. Experimentos con SBGA

Los experimentos de simulación por ordenador se llevaron a cabo usando las características descritas en el apartado 3.2.3.1, esto es el tamaño de colonia bacteriana, los valores de los componentes electrónicos y los atributos de la carcasa de la radio AM, así como las probabilidades de mutación y conjugación (o recombinación) indicadas en dicha sección.



**Figura 4.1:** Gráficos de eficacia correspondientes a la búsqueda del diseño óptimo del circuito oscilador de un receptor de radio AM

### 4.1.1. Optimización del oscilador

En la Fig. 4.1 se muestra un gráfico representativo de la eficacia medida a través del *fitness* medio por generación en los experimentos de optimización en los que una colonia bacteriana evoluciona bajo diferentes protocolos (C, SBGA y SDS) y operadores genéticos (CORP, COFP, recombinación con un punto de corte, recombinación con dos puntos de corte) buscando un circuito oscilador óptimo (en ausencia de filtros y carátula). En todos los protocolos la población convergía en una población compuesta por cromosomas con una distribución uniforme de genotipos similares con un *fitness* medio máximo próximo a  $10^6$ . Además, los

resultados indicaron que la evolución del circuito oscilador no parece depender del protocolo de algoritmo genético (C, SBGA y SDS), obteniéndose resultados similares en presencia o ausencia de mutación.

Para una correcta interpretación de los resultados es importante indicar que el máximo *fitness* que se puede obtener con el circuito oscilador es  $1,1 \times 10^6$ . No obstante, para este valor de *fitness* varios son los osciladores óptimos que pueden encontrarse para el rango de frecuencias entre 500 kHz y 1600 kHz. Asimismo, el *fitness* óptimo para el oscilador tiene el mismo valor que el *fitness* óptimo para el oscilador incluyendo el diodo y el filtro RC. Esto es así si tenemos en cuenta que en dicho caso el valor óptimo será igual a  $1,1 \times 10^6$  si efectivamente el valor  $e^{P_D+P_{RC}}$  es igual a la unidad, lo que ocurre cuando los filtros, es decir el diodo y el filtro RC, tienen valores óptimos. Sin embargo, el valor *fitness* máximo de la radio AM completa, es decir del circuito de la radio y la carátula, es  $2,2 \times 10^6$ . En este caso dicho valor representa a varios diseños de radio óptimos, y por tanto a toda una familia de “diseños correctos”. Es importante indicar que la existencia de varios “diseños correctos” de radio AM es responsabilidad del circuito oscilador que es el rasgo a optimizar que introduce varios óptimos, y no la carátula, ya que ésta sólo tiene un “único diseño” que le permita alcanzar el valor máximo de *fitness*.

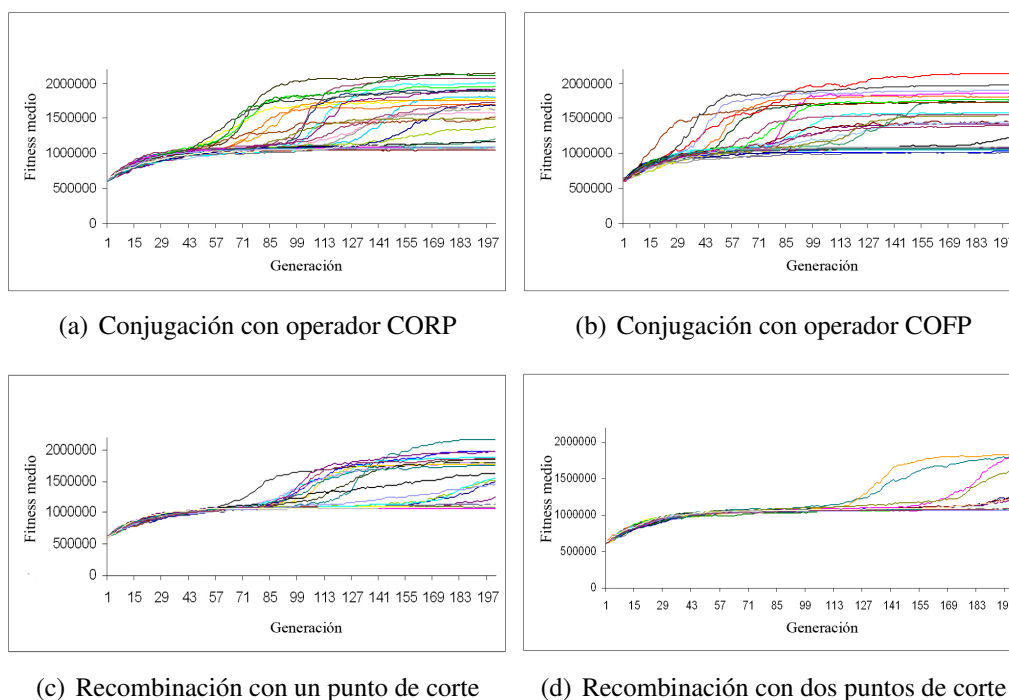
#### 4.1.2. Optimización de la radio AM

En los experimentos donde se optimizó la radio AM completa (oscilador, diodo, filtro RC y la carátula), se encontraron diferencias significativas entre la conjugación y los operadores clásicos de recombinación con un punto de corte y dos puntos de corte.

**Protocolo C** En el protocolo C (Fig. 4.2) la población de cromosomas escala por la superficie evolutiva alcanzando el diseño óptimo de la radio AM en dos pasos, sin importar que la población bacteriana tenga un tipo u otro de conjugación (CORP o COFP) o alternativamente de recombinación (recombinación con uno o dos puntos de corte). Durante el primer paso se lleva a cabo la optimización de la carátula de la radio AM hasta llegar a un *fitness* medio óptimo cercano a  $10^6$ . Posteriormente, cuando la colonia ha llegado al *fitness* óptimo de la carátula, las bacterias encaran un nuevo reto evolutivo. Es en este segundo paso cuando la colonia busca el óptimo del circuito electrónico (oscilador, diodo y filtro RC).

Los datos expuestos demuestran que las poblaciones bacterianas con conjugación llegan al segundo *fitness* óptimo en generaciones anteriores con mayor frecuencia que aquellas poblaciones basadas en el mecanismo clásico de la recombinación, ya sea con uno o dos puntos de corte. El menor grado de convergencia se

da en poblaciones bacterianas con recombinación en dos puntos de corte, logrando sólo unas pocas colonias bacterianas llegar al diseño óptimo de la radio AM. Una comparación de las medianas a través del test Krusal-Wallis (Tabla 4.1) nos permite concluir con un  $p$ -valor igual a 0,0017 que existen diferencias significativas entre las medianas al 95,0 % de confianza.



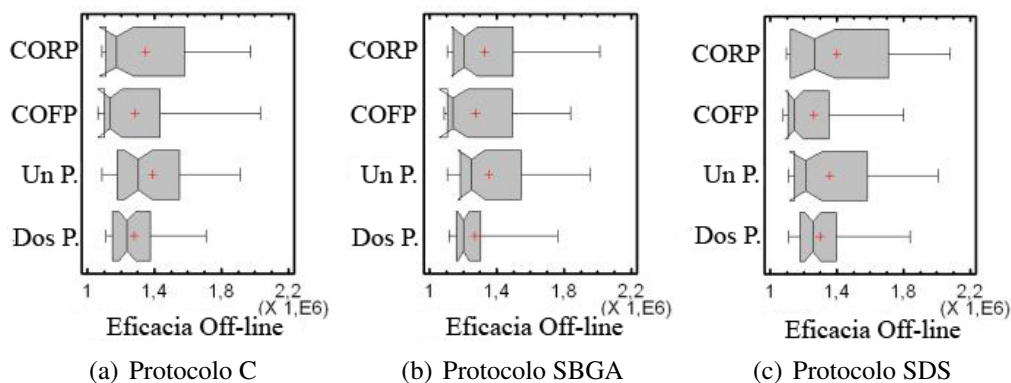
**Figura 4.2:** Gráficos de eficacia correspondientes a la búsqueda del diseño óptimo de un receptor de radio AM bajo protocolo C

Operador	n	Rango Medio
CORP	50	98,08
COFP	50	72,82
1PTO	50	122,46
2PTO	50	103,64
<b>valor del estadístico = 15,1096</b>		<b>p-valor = 0,0017</b>

**Tabla 4.1:** Test de Krusal-Wallis para el algoritmo con el protocolo C

El gráfico Multiple Box-and-Whisker (Fig. 4.3(a)) muestra la eficacia de cada algoritmo a partir de los valores *offline* de De Jong. Se puede observar que para los operadores de conjugación existe un solapamiento entre las muescas de las me-

dianas. De la misma forma, en los operadores clásicos de recombinación, también existe un solapamiento similar al anterior.



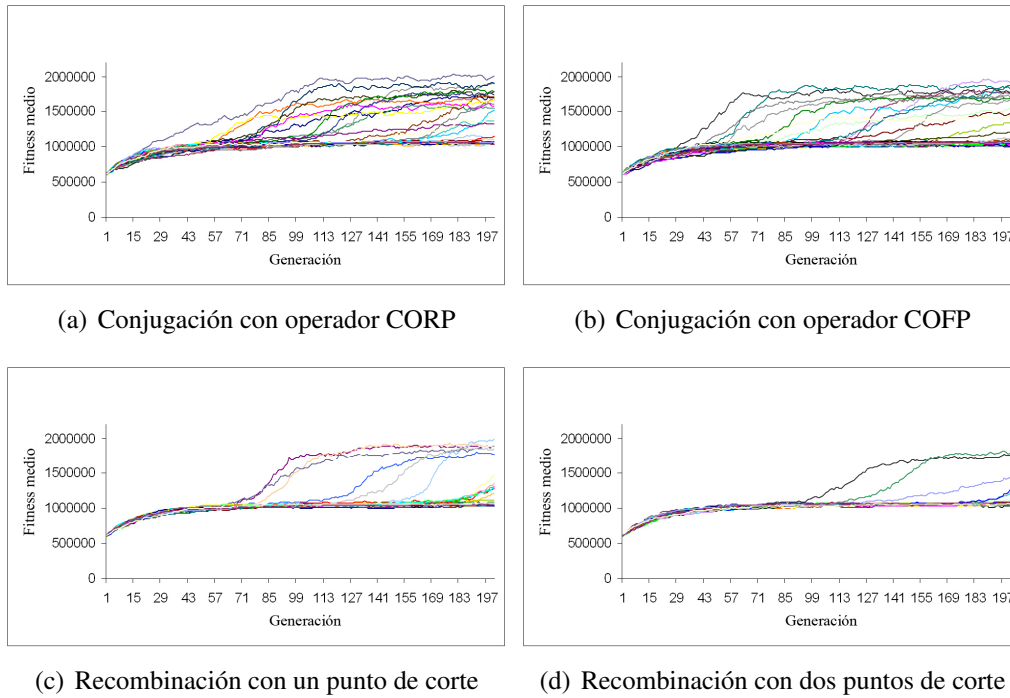
**Figura 4.3:** Gráfico Multiple Box-and-Whisker de la medida *offline* de De Jong obtenida bajo distintos operadores genéticos y protocolos.

**Protocolo SBGA** En el protocolo SBGA la convergencia evolutiva es muy pobre bajo todos los tipos de operadores de conjugación y recombinación (Fig. 4.4). En consecuencia, se puede concluir que la optimización de los componentes electrónicos es muy sensible a las mutaciones que se dan en cada ciclo o generación, mostrando la poca utilidad tanto de los algoritmos genéticos convencionales como del algoritmo genético bacteriano propuesto en esta Memoria, cuando uno u otro se usan en el diseño de circuitos electrónicos en presencia de mutación.

Operador	n	Rango Medio
CORP	50	105,38
COFP	50	76,10
1PTO	50	118,82
2PTO	50	101,70
valor del estadístico = 14,2722		p-valor = 0,0025

**Tabla 4.2:** Test Krusal-Wallis para el algoritmo con el protocolo SBGA

El test Krusal-Wallis (Tabla 4.2) muestra con un  $p$ -valor igual a 0,0025 que existen diferencias significativas entre las medianas con un nivel de confianza del 95,0%. El gráfico Multiple Box-and-Whisker de eficacia (Fig. 4.3(b)) muestra de nuevo un solapamiento entre las muescas, distinguiéndose un solapamiento entre la recombinación con un punto de corte y la recombinación con dos puntos de corte y la conjugación con el operador CORP. El operador genético que difiere significativamente de los anteriores es el de conjugación COFP.



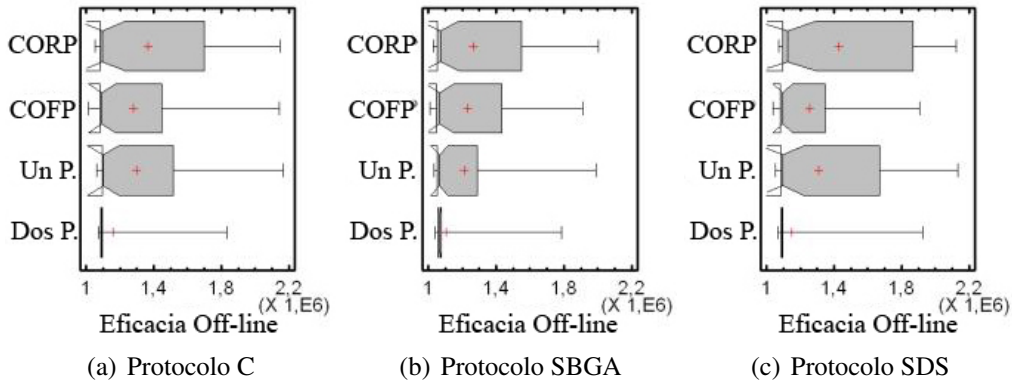
**Figura 4.4:** Gráficos de eficacia correspondientes a la búsqueda del diseño óptimo de un receptor de radio AM bajo protocolo SBGA

**Protocolo SDS** En el protocolo SDS los resultados fueron similares a los obtenidos en el protocolo C, demostrando ser más eficientes las colonias con conjugación frente a las colonias con recombinación en un punto y dos puntos de corte. Una vez más, las colonias con recombinación en dos puntos de corte exhibieron una convergencia evolutiva más pobre hacia el diseño óptimo de radio AM. El test Krusal-Wallis (Tabla 4.3) muestra con un  $p$ -valor igual a 0,0132 que existen diferencias significativas entre las medianas con un nivel de confianza del 95,0 %.

En el gráfico Multiple Box-and-Whisker (Fig. 4.3(c)) el solapamiento entre muescas fue muy similar al obtenido bajo el protocolo SBGA.

Operador	n	Rango Medio
CORP	50	107,46
COFP	50	77,32
1PTO	50	107,74
2PTO	50	109,48
valor del estadístico = 10,7285		$p$ -valor = 0,0132

**Tabla 4.3:** Test Krusal-Wallis para el algoritmo con el protocolo SDS



**Figura 4.5:** Gráfico Multiple Box-and-Whisker mostrando la mediana (línea vertical en muesca) y media aritmética (cruz) de los valores *fitness* medios obtenidos con el algoritmo bajo distintos operadores genéticos y protocolos

**Comparación entre los operadores de conjugación y recombinación** De acuerdo con los resultados anteriores concluiremos que la conjugación con el operador CORP rinde la máxima eficacia en el problema de optimización consistente en encontrar un diseño óptimo de radio AM. Este resultado es congruente con el análisis de la capacidad del operador genético de llevar a la población a una solución óptima (Fig. 4.5). Obsérvese como en los tres protocolos y si consideramos el bigote derecho (la línea horizontal extendida desde el extremo de la caja que representa el tercer cuartil, y por tanto el valor de los *fitness* medios mayores dentro del rango intercuartílico 1,5) aquellos que son más largos corresponden a los de los operadores CORP y recombinación con un punto de corte. Incluso cuando la mediana (línea vertical dentro de la muesca) es similar para todas las cajas, la longitud de la caja (que representa el rango intercuartílico de los valores de *fitness* medios) es mayor para el operador CORP, concluyendo que es este operador el que mejor busca los valores de la radio AM.

Operador	Generación $g_s$			
	50	100	150	200
<b>CORP</b>	0	6	11	14
<b>COFP</b>	2	8	9	9
<b>1PTO</b>	0	1	10	10
<b>2PTO</b>	0	0	1	3

**Tabla 4.4:** Número acumulado de poblaciones bacterianas que bajo el protocolo C hallan el diseño óptimo después de  $g_s$  generaciones

Además, considerando para cada operador genético la media de los valores *fitness* medios (representada como una cruz dentro de la caja, (Fig. 4.5)) se puede

concluir que bajo los protocolos C y SDS, ordenados por su eficacia, el mejor operador es el de conjugación CORP, seguido de la recombinación con un punto de corte, y la conjugación con el operador COFP. Finalmente la recombinación con dos puntos de corte demostró ser el peor de los operadores.

Los resultados anteriores se confirman cuando se considera el número de colonias con éxito que hallan bajo cada uno de los protocolos el diseño óptimo de radio AM (Tablas 4.4, 4.5 y 4.6).

En los experimentos se ha considerado que un diseño de radio AM es óptimo si tiene un valor *fitness* igual o superior a 1.646.244, valor que tendría un diseño “perfecto” con un diodo de silicio. Los resultados indican que para los protocolos C y SDS la conjugación CORP exhibe una eficacia mayor que la recombinación clásica. Los pobres resultados obtenidos con el protocolo SBGA se explican por el efecto disruptivo de la mutación cuando se aplica generación tras generación.

Operador	Generación $g_s$			
	50	100	150	200
<b>CORP</b>	0	2	7	10
<b>COFP</b>	0	3	6	10
<b>1PTO</b>	0	1	4	7
<b>2PTO</b>	0	0	1	2

**Tabla 4.5:** Número acumulado de poblaciones bacterianas que bajo el protocolo SBGA hallan el diseño óptimo después de  $g_s$  generaciones

Considerando los resultados anteriores y después de 200 generaciones, se puede afirmar que en el protocolo C un 63,89 % de las colonias con conjugación hallaron un diseño óptimo, siendo el 36,11 % el porcentaje de colonias que evolucionaron hasta un diseño óptimo a través de la recombinación clásica ya fuera con uno o dos puntos de corte.

Operador	Generación $g_s$			
	50	100	150	200
<b>CORP</b>	1	10	15	19
<b>COFP</b>	0	3	6	7
<b>1PTO</b>	0	3	8	13
<b>2PTO</b>	0	0	0	3

**Tabla 4.6:** Número acumulado de poblaciones bacterianas que bajo el protocolo SDS hallan el diseño óptimo después de  $g_s$  generaciones

Asimismo, en el protocolo SDS el rango de colonias con conjugación que llegó a un diseño óptimo de la radio AM después de 200 generaciones fue igual

al 61,91 %, correspondiendo el 30,09 % restante a aquellas colonias con recombinación clásica. Por último, y aunque en el protocolo C no existen diferencias significativas entre ambos tipos de conjugación (CORP y COFP), las diferencias que se obtuvieron en el protocolo SDS sí fueron significativas. Con este protocolo el número de colonias que alcanzaron un diseño óptimo después de 200 generaciones con la conjugación CORP fue igual al 73,08 %, mientras que sólo el 26,92 % restante de colonias alcanzaron un diseño óptimo después de 200 generaciones a través de la conjugación COFP. Por consiguiente en el protocolo SDS las poblaciones bacterianas con conjugación COFP tienen una eficacia muy pobre comparada con las poblaciones con conjugación CORP. En relación con la recombinación con uno y dos puntos de corte, e independientemente del protocolo experimental, la recombinación con un punto de corte tiene mayor eficacia que la recombinación con dos puntos de corte.

$N$	$\lambda^2$	$p$ -valor
200	0,0333	0,8874
300	0,0465	0,8751
400	<sup>3</sup>	<sup>3</sup>
500	0,0161	0,5951
600	0,0161	0,6375
700	0,0110	0,2672
800	0,0110	0,3664

**Tabla 4.7:** Test de  $\chi^2$  para la tabla de contingencia entre el tipo de operador genético y el número de generaciones  $g_s$ , variando el tamaño poblacional ( $N$ ).

<sup>2</sup> Coeficiente de predictibilidad de Guttman  $\lambda = (F - M)/N - M$ , donde  $F$  es la suma de las frecuencias de las celdas mayores dentro de cada categoría de variable,  $M$  el marginal total mayor en las categorías de la otra variable y  $N$  el número total de casos.

<sup>3</sup> Tablas de contingencia con cero filas o columnas.

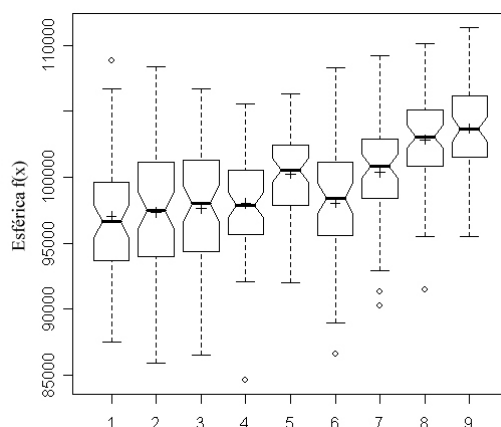
**Tamaño poblacional: Número de colonias** Los resultados de los experimentos de simulación bajo diferentes tamaños poblacionales fueron como sigue. La simulación se realizó con distinto número de bacterias obteniendo  $g_s$  (número acumulado de colonias que alcanzaron el diseño óptimo en varias generaciones). Para cada uno de los tamaños poblacionales se realizó un test  $\chi^2$  para la tabla de contingencia entre el tipo de operador genético y la generación de salto  $g_s$ . Los resultados obtenidos que se muestran en la (Tabla 4.7), muestran la independencia entre el tipo de operador genético y  $g_s$ , sin que en ello influya el tamaño poblacional. Además, los valores  $\lambda$  (medida de la utilidad de la fila o columna para predecir el otro factor) son próximos a cero para todos los tamaños poblacionales estudiados. En otras palabras, sean  $N$  y el tipo de operador genético conocidos

de antemano, dado un cierto valor igual o inferior a  $g_s$  no es posible establecer *a priori* el número de colonias que alcanzarán el diseño óptimo de una radio AM.

## 4.2. Experimentos con PETRI

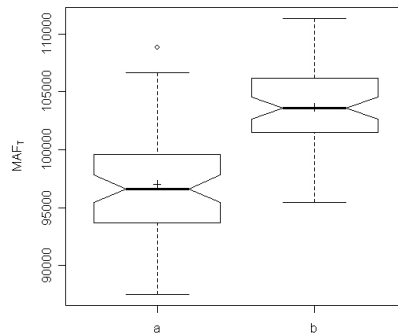
Los experimentos de simulación se llevaron a cabo con los protocolos de transducción y probabilidades de conjugación y mutación mencionadas en la sección 3.4.

### 4.2.1. Optimización de la función esférica



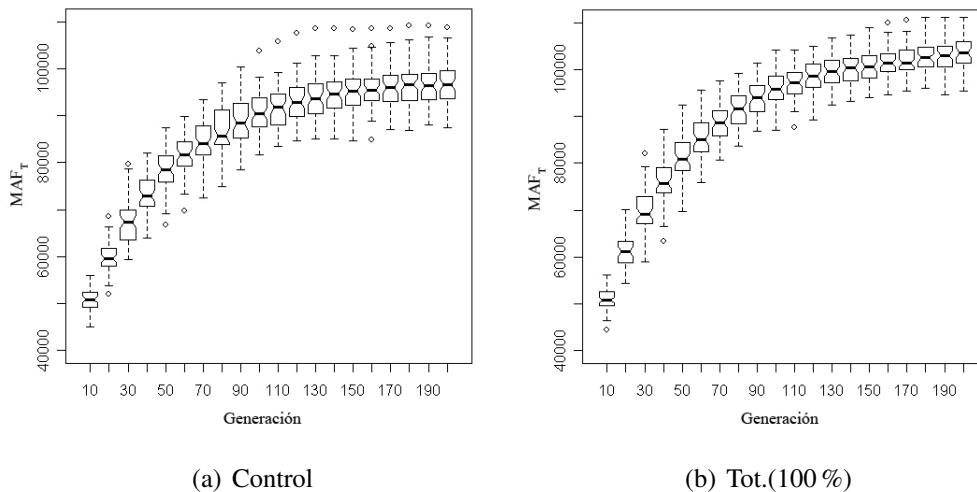
**Figura 4.6:** Gráfico Multiple Box-and-Whisker de *fitness* medios en la generación final de los experimentos con transducción optimizando la función esférica. Los experimentos se llevaron a cabo con los porcentajes de transducción indicados entre paréntesis y con la política de elección del fragmento indicada (segmento cromosómico=Frag., cromosoma completo=Tot.): (1) Experimento control (sin transducción), (2) Frag.(25 %), (3) Frag.(50 %), (4) Frag.(75 %), (5) Frag.(100 %), (6) Tot.(25 %), (7) Tot.(50 %), (8) Tot.(75 %) y (9) Tot.(100 %).

Considerando en conjunto todos los experimentos y comparándolos con el experimento control (Fig. 4.6) se observó que el mejor resultado para la función esférica se obtuvo en el experimento en el que el bacteriófago transfiere el cromosoma completo, siendo la probabilidad de transducción igual al 100 % (Fig. 4.7). Los resultados se corroboran con un test Kruskal-Wallis siendo el  $p$ -valor menor a  $2,2 \times 10^{-16}$ . Puesto que el  $p$ -valor es menor que 0,05 existen diferencias significativas entre las medianas con un nivel de confianza del 95 %.



**Figura 4.7:** Gráfico Box-and-Whisker de los valores de *fitness* medio en la generación final de los experimentos con transducción optimizando la función esférica: (a) experimento control y (b) mejor experimento (Tot.(100 %))

Como se puede observar en la Fig. 4.6 la probabilidad de transducción, y por tanto el número de veces que la información genética migra de una placa a otra, tiene un efecto beneficioso sobre la conjugación cuanto mayor es dicha probabilidad. Si se transfieren segmentos cromosómicos este resultado se cumple independientemente de cuál sea la política de selección del fragmento, verificándose también cuando se transfiere el cromosoma completo.



**Figura 4.8:** Gráfico Box-and-Whisker de los valores de *fitness* medio por cada diez generaciones de los experimentos con transducción optimizando la función esférica

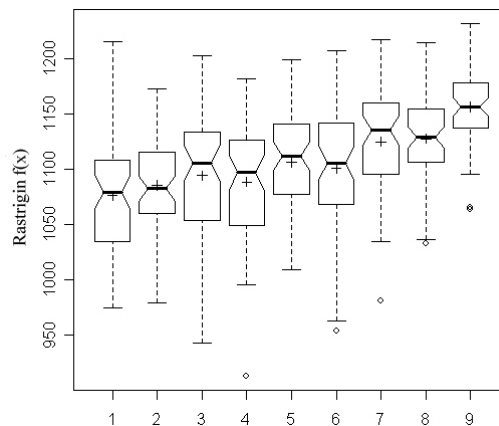
En la Fig. 4.8 se puede observar cómo el mejor experimento obtiene los resultados más altos, no sólo en la última generación, sino que también los consigue alcanzar en generaciones más tempranas.

Prueba	$p$ -valor
Cochran	0,0228
Bartlett	0,0004
Levene	0,0005

**Tabla 4.8:** Test de homogeneidad de la varianza en la función esférica

El análisis de la homogeneidad de la varianza de la población fue realizado en estos ensayos con los test de Bartlett, Cochran y Levene siendo los  $p$ -valores en todos ellos inferiores a 0,05 (Tabla 4.8). La variabilidad de la población depende de la probabilidad de la transducción así como de la política de selección del cromosoma o segmentos cromosómicos. En particular, en los experimentos con probabilidad de transducción igual al 100 % la varianza fue mínima. Sin embargo, la mayor variabilidad poblacional se alcanza cuando la probabilidad de transducción es el 50 %.

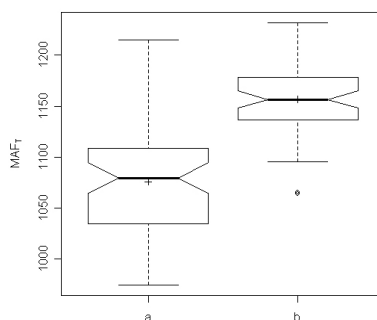
#### 4.2.2. Optimización de la función Rastrigin



**Figura 4.9:** Gráfico Multiple Box-and-Whisker de *fitness* medios en la generación final de los experimentos con transducción optimizando la función Rastrigin. Los experimentos se llevaron a cabo con los porcentajes de transducción indicados entre paréntesis y con la política de elección del fragmento indicada (segmento cromosómico=Frag., cromosoma completo=Tot.): (1) Experimento control (sin transducción), (2) Frag.(25 %), (3) Frag.(50 %), (4) Frag.(75 %), (5) Frag.(100 %), (6) Tot.(25 %), (7) Tot.(50 %), (8) Tot.(75 %) y (9) Tot.(100 %).

Los resultados obtenidos con la función Rastrigin son similares a los obtenidos con la función esférica. Por ello, al igual que sucede en la función esférica,

si se consideran conjuntamente todos los experimentos y se comparan con el experimento control (Fig. 4.9) se puede observar como el mejor resultado para esta función es aquel en el que el bacteriófago transfiere el cromosoma completo siendo la probabilidad de transducción el 100 % (Fig. 4.10). Este resultado es similar al obtenido en el problema de optimización con la función esférica, corroborándose los resultados con un test de Kruskal-Wallis en el que el  $p$ -valor fue inferior a  $2,2 \times 10^{-16}$ .



**Figura 4.10:** Gráfico Box-and-Whisker de los valores de *fitness* medio en la generación final de los experimentos con transducción optimizando la función Rastrigin: (a) experimento control y (b) mejor experimento (Tot.(100 %))

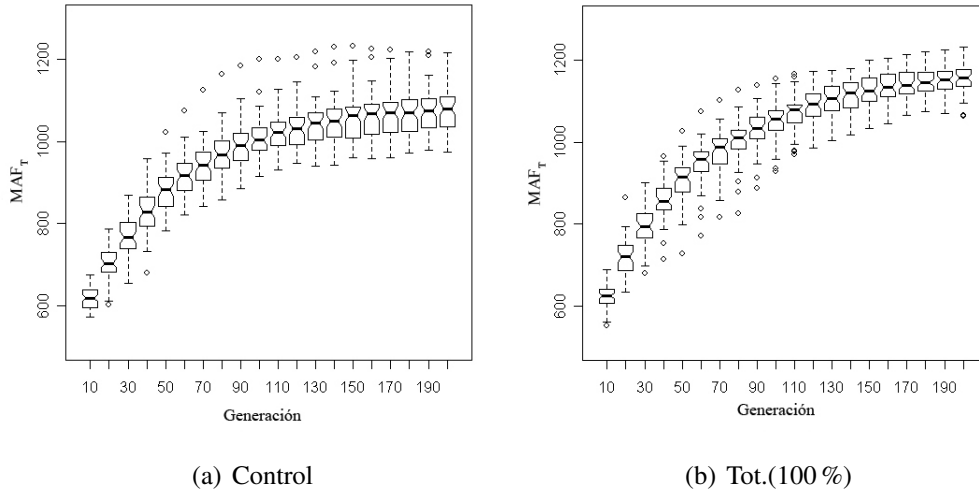
Una vez más, como se puede observar en la Fig. 4.9 la probabilidad de transducción tiene un efecto mayor cuanto mayor sea su valor independientemente de la política de selección del cromosoma o segmento cromosómico.

De nuevo en la Fig. 4.11 puede observarse como el mejor experimento no sólo obtiene los resultados mejores en la última generación, sino que también los consigue alcanzar en generaciones más tempranas.

Prueba	$p$ -valor
Cochran	0,0144
Bartlett	0,0003
Levene	0,0008

**Tabla 4.9:** Test de homogeneidad de la varianza en la función Rastrigin

Finalmente, y de un modo similar a los experimentos realizados con la función esférica, los test de Bartlett, Cochran y Levene rindieron  $p$ -valores por debajo de 0,05 (Tabla 4.9).



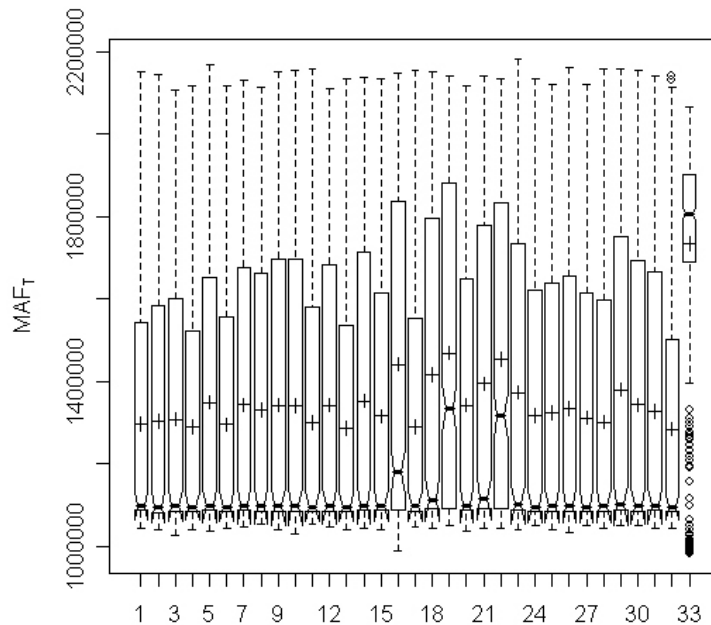
**Figura 4.11:** Gráfico Box-and-Whisker de los valores de *fitness* medio por cada diez generaciones de los experimentos con transducción optimizando la función Rastrigin

### 4.2.3. Optimización de la radio AM

En la Fig. 4.12 se muestran en un gráfico Multiple Box-and-Whisker los valores de *fitness* medio en la última generación obtenidos para cada uno de los experimentos de simulación realizados con la transducción. El test Kruskal-Wallis con un  $p$ -valor menor a  $2,2 \times 10^{-16}$  muestra que hay diferencias significativas entre las medianas con un nivel de confianza del 95 %.

Considerando conjuntamente todos los experimentos y comparándolos con el experimento control (experimento de simulación sin transducción (1), Fig. 4.12) podemos concluir lo que sigue. En primer lugar, en los casos en los que la transducción se lleva a cabo sólo con segmentos del cromosoma el mejor resultado (Fig. 4.13) se obtiene cuando el bacteriófago selecciona la placa Petri y la bacteria donadoras ( $p^D$ ,  $b^D$ ) con el criterio de mejor placa y mejor bacteria respectivamente (criterio *max-max*), seleccionando la placa Petri receptora ( $p^R$ ) por el criterio de la ruleta y la bacteria receptora ( $b^R$ ) al azar (criterio *roul-r*) (experimento (19), Fig. 4.12). Estos resultados se corroboran con un test de Kruskal-Wallis siendo el  $p$ -valor igual a  $1,603 \times 10^{-8}$  por lo que existen diferencias significativas entre las medianas con un nivel de confianza del 95 %. En segundo lugar, cuando el bacteriófago transfiere el cromosoma completo (experimento (33), Fig. 4.12) el mejor resultado se obtiene con una probabilidad de transducción del 100 %. En este caso, la comparación de las medianas frente al experimento control (Fig. 4.13) es tal que en el test de Kruskal-Wallis se obtuvo un  $p$ -valor menor que  $2,2 \times 10^{-16}$  por lo que con el nivel de confianza del 95 %, y comparándolo con el caso anterior,

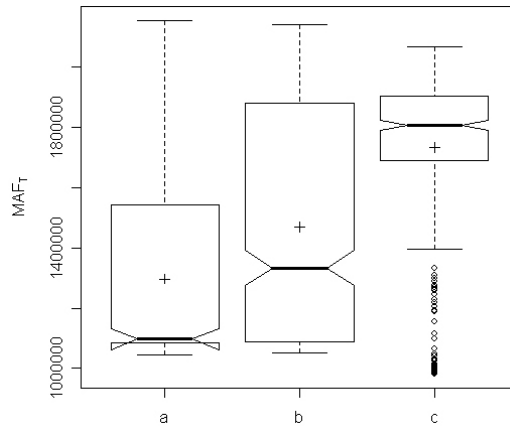
existen mayores diferencias significativas con respecto al experimento control.



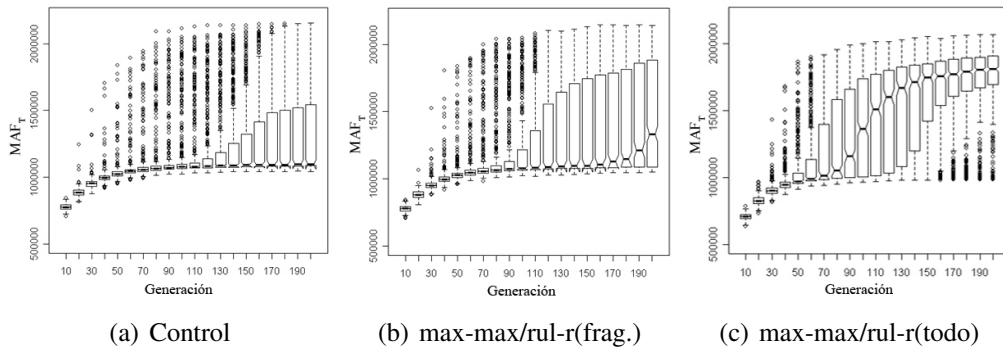
**Figura 4.12:** Gráfico Multiple Box-and-Whisker de *fitness* medios en la generación final de los experimentos con transducción optimizando la radio AM. Los experimentos con probabilidad de transducción del 100 % fueron: (1) Experimento control (sin transducción), (2) r-r, r-r, (3) max-max, max-max, (4) ave-ave, ave-ave, (5) roul-roul, roul-roul, (6) min-min, min-min, (7) inv roul-inv roul, inv-roul-inv roul, (8) roul-roul , min-min, (9) roul-roul, ave-ave, (10) roul-roul, inv-roul-inv roul, (11) roul-roul, max-max, (12) roul-roul, r-r, (13) min-min, roul-roul, (14) ave-ave, roul-roul, (15) inv roul-inv roul, roul-roul, (16) max-max, roul-roul, (17) r-r, roul-roul, (18) max-max, roul-roul, (19) max-max, roul-r, (20) max-max, roul-min, (21) max-max, roul-ave, (22) max-max, roul-max, (23) max-max , roul-inv roul, (24) max-max, max-roul, (25) max-max, max-r, (26) max-max, max-min, (27) max-max, max-ave, (28) max-max, max-inv roul. En los experimentos de transducción con el cromosoma completo se utilizó la política de selección max-max, roul-r con las siguientes probabilidades de transducción: (29) 75 % (30) 50 % (31) 25 % (32) 0 % (33) 100 %.

En la Fig. 4.14, se muestra un gráfico de eficiencia representativo (un Box-and-Whisker plot por cada diez generaciones) de los valores *fitness* medios de los experimentos control en los que no se ha llevado a cabo la transducción (Fig. 4.14(a)). Los resultados de este gráfico se pueden comparar con los obtenidos para el mejor experimento de transducción transfiriendo sólo segmentos cromosómicos

(Fig. 4.14(b)) y con aquellos que transfieren el cromosoma completo (Fig. 4.14(c)). El mejor experimento no solamente es capaz de conseguir mejores resultados en las generaciones finales, sino que incluso en las generaciones intermedias consigue optimizar con eficacia el problema propuesto.



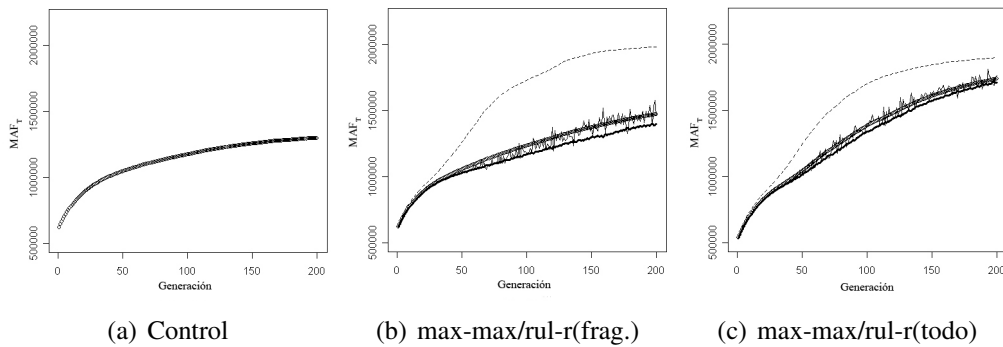
**Figura 4.13:** Gráfico Box-and-Whisker de los valores de *fitness* medio en la generación final de los experimentos con transducción optimizando la radio AM: (a) experimento control y (b) mejor experimento con fragmentos (max-max/rul-r(frag.)) (c) mejor experimento con todo el cromosoma (max-max/rul-r(todo))



**Figura 4.14:** Box-and-Whisker de los valores de *fitness* medio por cada diez generaciones de los experimentos optimizando la radio AM con transducción

Además, en la Fig. 4.15 se muestran las medias de los *fitness* medios por generación o *MAF*. En la Fig. 4.15(a) se muestran las  $MAF_D$ ,  $MAF_R$ ,  $MAF_T$ ,  $MAF_{RD}$  del experimento control. En la Fig. 4.15(b) se muestra el mejor experimento transduciendo sólo segmentos, donde la curva con mayor pendiente corresponde al *MAF* de las placas Petri donantes ( $MAF_D$ ), mientras que las dos curvas

superpuestas muestran el  $MAF$  de las placas receptoras ( $MAF_R$ ) y el  $MAF$  de todas las placas ( $MAF_T$ ). El comportamiento oscilante de  $MAF_R$  se debe a que la placa y la bacteria receptoras se seleccionan con métodos estocásticos. La peor eficacia, representada por la curva con menor pendiente, es la de aquellas placas que no participan en la transducción como donadoras o receptoras ( $MAF_{RD}$ ). En la Fig. 4.15(c) se muestra el mejor experimento transduciendo el cromosoma completo.



**Figura 4.15:** MAF por generación de los experimentos con transducción optimizando la radio AM con transducción

Comparando las tres figuras se puede observar la mejora que supone en la conjugación la participación del bacteriófago, y por tanto de la transducción, y la mayor eficacia de traspasar el cromosoma completo frente al traspaso únicamente de segmentos cromosómicos.

# **Capítulo 5**

## **Discusión**

Los resultados obtenidos y descritos en esta Memoria sugieren que los operadores conjugación y transducción se ajustan razonablemente a los esquemas generales de sus homólogos bacterianos, demostrándose su utilidad práctica en Computación Evolutiva.

El operador genético de la conjugación bacteriana consigue que el algoritmo genético SBGA que lo implementa obtenga mejores resultados en el problema de optimización propuesto, el diseño de un receptor de radio AM, que el algoritmo genético basado en la recombinación clásica. Uno de los logros en los experimentos de simulación descritos en esta Memoria es la aplicación de un algoritmo genético a la “realización física” de un *hardware* en términos de su descripción funcional, en nuestro caso de una radio AM. Dicha realización viene dada por la expresión (3.10) utilizada para evaluar el *fitness* o bondad de un diseño, en particular del diseño del circuito electrónico, así como de las principales características estéticas y funcionales de la carcasa que lo contiene. La función *fitness* usada para evaluar la carcasa de la radio (véase expresión (3.19)) es lineal, pudiendo mejorarse adaptándola la expresión general de la función *fitness* presentada por Bentley (2002) que evalúa las características estéticas en la formación de patrones.

El operador conjugación puede resultar útil en experimentos de *hardware* evolutivo reemplazando la función *fitness ad hoc* para evaluar la radio AM por otra función de evaluación propia del circuito electrónico a optimizar. Además el cromosoma circular podría ser sustituido por funciones de tipo Lisp. Por ejemplo, Funes y Pollack (1999) fueron capaces de evolucionar estructuras de LEGO con el operador recombinación usando como representación genética funciones de tipo Lisp. La adecuación de distintos materiales (Harding y Miller, 2005) para la realización de experimentos de esta clase puede resultar muy beneficiosa. Se trataría de materiales que representan un entorno físico más rico que el simulado en ordenador, debido principalmente a su riqueza de interacciones físicas internas, como por ejemplo ocurre con el cristal líquido, polímeros, coloides, o las propias colonias de bacterias. Por consiguiente, estos materiales, incluyendo las colonias bacterianas, podrían ser utilizadas para experimentos de evolución in materio, que se llevarían a cabo en los llamados FPMA (*Field Programmable Matter Array* (Harding y Miller, 2005)). En un futuro las colonias de bacterias podrán ser utilizadas tecnológicamente con dos funciones completamente diferentes aunque complementarias: por un lado actuar como *materia programable*, y de otro actuar como *agente de búsqueda* para la configuración específica de un circuito electrónico, o cualquier otro dispositivo, a través del mecanismo de la conjugación. Además, la presencia de un citoesqueleto bacteriano (Jones *et al.*, 2001) se puede ver como un posible *hardware* para la computación molecular (Hameroff *et al.*, 1992), apoyando la posibilidad de uso de las bacterias como materia programable

(Di Paola *et al.*, 2004).

En el operador de conjugación propuesto la recombinación es simulada de forma simplificada, ya que en la Naturaleza el mecanismo genético subyacente es más complejo. Por ejemplo, el desarrollo y la formación de un diploide parcial o merocigoto bacteriano ha sido omitido del operador. Sin embargo, pese a tales simplificaciones con el parámetro de conjugación  $\alpha$  se pueden simular aquellos factores que intervienen en la conjugación y que afectan a la ruptura espontánea del puente conjugativo, y en consecuencia, al grado de transferencia o el número de genes transferidos. En consecuencia la probabilidad de que la bacteria receptora reciba los últimos genes del fragmento cromosómico de la bacteria donante es menor. En el caso de la conjugación COFP habrá mayor probabilidad de que un gen sea transferido cuanto menor sea la distancia con respecto al punto de origen.

Sin tener en cuenta el tipo de conjugación que se use, CORP o COFP, podría pensarse erróneamente que la conjugación es un operador de macromutación. Sin embargo, es una interpretación errónea del papel de dicho operador, ya que la recombinación no recíproca de los cromosomas circulares y los segmentos transferidos involucra *schemata* (Holland, 1975) muy mejorados que se mezclan para crear cadenas con un mayor *fitness*.

En el algoritmo SBGA se eligió una probabilidad de conjugación del 75 %, una frecuencia muy alta si se compara con la conjugación en el mundo real. En las bacterias los cambios genéticos a través de este mecanismo ocurren con una frecuencia muy baja, siendo del orden de menos de  $10^{-4}$  veces por gen y generación (Tenaillon *et al.*, 2000). Bjorklof *et al.* (1995) descubrieron que en bacterias que se encuentran en un ambiente nutritivamente rico, y en un área colonizable limitada, existe una frecuencia alta de conjugación. En contraste con estos nutrientes, otras sustancias como los ácidos grasos insaturados son inhibidores de la conjugación bacteriana (Fernandez-Lopez *et al.*, 2005), cuya frecuencia depende de mutaciones genéticas específicas (Segal *et al.*, 1998).

En relación con otros algoritmos, el caso del algoritmo "Microbial GA" (Harvey, 1996), podría convertirse en un caso especial del SBGA propuesto en esta Memoria, ya que la transferencia génica se realiza desde el cromosoma ganador al cromosoma perdedor. Por otro lado, a pesar de que el operador conjugación que hemos propuesto resulta útil en casos prácticos, a día de hoy no disponemos de un método definitivo que pueda emplearse para calcular el valor del parámetro  $\alpha$  que resulte más adecuado a cada nueva aplicación del algoritmo genético bacteriano. Un ejemplo parecido es el descrito por Simoes y Costa (2000) con un operador de transposición, el cual permite al algoritmo genético llegar a mejores soluciones que el operador estándar de recombinación. Sin embargo, la eficacia del algoritmo dependía del tamaño de las secuencias flanqueantes o terminales del transposón, planteándose un problema parecido al del operador conjugación con

el parámetro  $\alpha$ . Por consiguiente, nuestro algoritmo exhibe un problema que es común a otros algoritmos relacionados. Otro ejemplo que ilustra esta situación se da en la versión original de PBGA (*Pseudo-Bacterial Genetic Algorithm*) presentado por Furuhashi *et al.* (1995). En este algoritmo se carecía de un criterio para decidir los puntos de mutación de los cromosomas, así como los puntos de corte para la recombinación, por lo que los autores tuvieron que introducir un segundo operador que decidiera de forma adaptativa los sitios mencionados. Otro ejemplo similar es el de las redes neuronales artificiales u otros modelos sub-simbólicos o no conexionistas en Inteligencia Artificial, en los que los valores de los parámetros son elegidos empíricamente. Más aún, en algunos casos los algoritmos genéticos exhiben otros problemas que son comunes a las redes neuronales artificiales. Por ejemplo, Smith (1996) observó la forma en la que las poblaciones bacterianas simuladas por ordenador exhiben un fenómeno de saturación debido a los individuos donadores. En un primer momento, este fenómeno puede asemejarse a la saturación de los pesos sinápticos observados en redes neuronales artificiales con la regla de aprendizaje hebbiano. Sin embargo, esta saturación se debe al éxito de la conjugación creando mejores especímenes de bacterias con la característica de ser individuos  $F^-$ . La solución al “problema de saturación del factor  $F$ ” consiste en mutar dicho factor con una alta probabilidad. Otra posibilidad, tal y como se propone en las simulaciones tratadas en esta Memoria, es la de elegir a la bacteria donante o  $F^+$  y a la receptora o  $F^-$  sin incluir el factor  $F$  en el cromosoma, convirtiéndolo en una capacidad no heredable, y evitando así el problema de saturación de dicho factor.

En el ámbito de las simulaciones por ordenador, la selección de las bacterias para la conjugación puede simularse a través métodos distintos al propuesto en esta Memoria. Así por ejemplo, Harvey (1996) elegía los genes parentales por medio del método del torneo. En particular, una vez que los genes parentales son elegidos al azar, la bacteria de mayor *fitness* o ganadora se convertía en donante y la bacteria de menor *fitness* o perdedora en receptora del material genético. Por otro lado, Smith (1996) usó un esquema de selección que se asemeja al “mar de organismos” usado en experimentos de Vida Artificial, también llamados *Turing Gas* (Fontana, 1991a,b). En esta aproximación, la selección se lleva a cabo con aquellos individuos localizados en posiciones adyacentes o vecinas de la matriz.

Si tenemos en cuenta la calidad de las soluciones, el hecho de que la población de bacterias llegue a mejores soluciones con la conjugación CORP con respecto a la recombinación con uno y dos puntos de corte estaría de acuerdo con los resultados de otros operadores genéticos, tal es el caso del operador transposición (Simoes y Costa, 2000) y el operador de transformación (Simoes y Costa, 2001).

En la actualidad, la mayoría de los Algoritmos Genéticos Distribuidos con-

sideran como único mecanismo de transferencia horizontal de genes el proceso de migración de cromosomas de una población a otra. Sin embargo, a lo largo de esta Memoria se muestra como, basándose en la genética microbiana, es posible desarrollar nuevas políticas de selección que podrían ser aplicadas tras su correspondiente adaptación en el área de la Computación Evolutiva Distribuida. Por ejemplo, en el operador de transducción propuesto se combinan la infección de bacterias con bacteriófagos y la recombinación homóloga o sobrecruzamiento entre el ADN de la bacteria donante y el cromosoma bacteriano receptor. Además, también se tiene en cuenta la política de selección de los donantes por la que un virus infectaría a las poblaciones mejor adaptadas. En otras palabras, con la transducción se crea un modelo donde la migración es algo más que el simple movimiento o transferencia de información genética.

No obstante, y pese a estas consideraciones, PETRI no es un Algoritmo Genético Distribuido. La optimización por el algoritmo PETRI del receptor de radio AM y de las funciones esférica y Rastrigin se llevaron a cabo con un ordenador monoprocesador, usando eso sí una estrategia multitarea. Asumiendo que la conjugación ocurre en cada generación, cada población (o placa Petri) se actualiza de una generación  $t$  a una  $t + 1$ . Cuando todas las poblaciones se han actualizado al mismo número de generación se lleva a cabo la transducción entre dos de las poblaciones. Posteriormente, las poblaciones vuelven a ser actualizadas y transducidas. Como opción de mejora del uso del algoritmo, PETRI podría ser implementado, obviamente tras el correspondiente trabajo de adaptación, para ejecutarse en un sistema computacional paralelo pudiéndose desarrollar una librería de rutinas para ser usada en el campo de la Computación Evolutiva Distribuida.

Aunque nuestro algoritmo no está paralelizado, ni es en un sentido estricto tal y como hemos mencionado un Algoritmo Genético Distribuido, en el algoritmo PETRI se combinan dos mecanismos genéticos diferentes exhibidos por las bacterias reales que nos son habituales en los algoritmos distribuidos. El primero de ellos es un mecanismo local (*dentro de la colonia*) que simula la transferencia horizontal de información dentro de una población a través de la conjugación; el segundo es un mecanismo global (*entre colonias*) que simula la transferencia horizontal de información entre poblaciones a través de la transducción. Además, en la mayoría de los algoritmos genéticos multipoblacionales la migración es ocasional y poco frecuente. Sin embargo, en el algoritmo propuesto la transducción se aplica una vez por generación obteniendo mejores resultados cuanto mayor es la probabilidad de transducción (o lo que es equivalente de migración). Como futuro trabajo se deja indicada la necesidad de estudiar la posible mejora del algoritmo PETRI aumentando el número de bacteriófagos implicados en la transducción o definiendo en éstos un modelo *ad hoc* que los simule como entidades o vectores

de información. Por otro lado, y de acuerdo con Cohoon *et al.* (1987), hay que señalar que siempre que se tenga gran conectividad y diámetros pequeños (cuatro y nueve placas Petri en los ensayos de esta Memoria) la topología de un algoritmo genético multipoblacional (hipercubos, anillos, etc.) carecerá de gran importancia en la eficacia del algoritmo. Para la mayoría de los modelos, la complejidad es algo próximo a  $O(\sqrt{n})$  siendo  $n$  el número de poblaciones. Sin embargo, se ha demostrado que el factor principal más influyente en los Algoritmos Genéticos Distribuidos o multipoblacionales es la política de migración, es decir la selección y destino de los cromosomas emigrantes. En este sentido, hay que señalar que los experimentos se basan en una topología dinámica: la población o colonia donante (o placa Petri), el emigrante (la bacteria o su cromosoma) y los receptores (placa Petri y bacteria) se basan en el algoritmo PETRI en criterios dinámicos. De acuerdo con Grefenstette (1981) se piensa que la migración previene la convergencia prematura de las poblaciones. Así, en el modelo propuesto en esta Memoria se demuestra que la mejor política de migración es aquella donde la placa Petri y la bacteria donante se seleccionan basándose en el criterio del máximo *fitness*, y las receptoras se seleccionan por métodos estocásticos, tales como el criterio de la ruleta y selección al azar respectivamente. En lo que respecta a la transducción (que incluye una recombinación homóloga posterior) en PETRI promueve el salto súbito de la población o colonia de bacterias receptoras hacia mejores soluciones. Una vez más, y aunque PETRI no sea un algoritmo distribuido en un sentido estricto, creemos que de un modo similar a los Algoritmos Genéticos Distribuidos (Cohoon *et al.*, 1991) la llegada y recombinación de “nueva y mejor” información genética (incluyendo buenos *schemata* de Holland) puede romper el equilibrio poblacional en la población o colonia receptora (o placa Petri) provocando un cambio evolutivo. Los resultados descritos a lo largo de esta Memoria apoyan observaciones similares realizadas en la Naturaleza. Por ejemplo, Forde *et al.* (2004) descubrieron que el flujo de genes vía bacteriófago T7 entre poblaciones genéticamente diferenciadas puede alterar su evolución, llegando a adaptarse a su nicho ecológico.

En el artículo (Kubota *et al.*, 1996) sus autores introdujeron la posibilidad de simular la transducción proponiendo VEGA, un Algoritmo Genético Evolutivo Viral. Sus autores presentaron un operador de infección vírica asociando a cada virus un valor *fitness* vírico. La diferencia principal entre este modelo y el propuesto en esta Memoria, PETRI, es el hecho de que en VEGA la población huésped representa las soluciones candidatas, y la población vírica que inicialmente se genera desde la población huésped, representa un grupo de subcadenas de soluciones. En consecuencia, el modelo subyacente en VEGA es la coevolución de dos poblaciones, una principal o huésped y otra secundaria o vírica, compartiendo soluciones a través de la transducción. Además, en VEGA los virus propagan

sus propias subcadenas o fragmentos cromosómicos entre los individuos huésped mientras que los bacteriófagos de PETRI propagan la subcadenas o fragmentos cromosómicos de los huéspedes entre los propios individuos huésped. Otra diferencia interesante es que en VEGA los virus representan una población real donde cada uno tiene un valor *fitness* que viene dado por el *fitness* del huésped antes y después de la infección. Sin embargo, en el modelo propuesto en la Memoria, los virus se simulan como simples agentes responsables del mecanismo de la transducción entre las bacterias huésped. Por último, en VEGA la población huésped no se compone de bacterias, ya que se simula la recombinación como es habitual en los algoritmos genéticos: por medio o a través de la transferencia vertical de genes. Por el contrario, en PETRI se simula la recombinación a través de un operador conjugación o si se prefiere por medio de una transferencia horizontal de genes, tal y como tiene lugar en la Naturaleza.

El estudio realizado con la política de transducción, en el que además se probaron distintas probabilidades de transducción, permite contar con diversas opciones a la hora de utilizar el algoritmo PETRI. La naturaleza del problema a resolver permitiría elegir entre dos opciones diferentes: obtener el “mejor resultado” en unas poblaciones homogéneas transfiriendo la totalidad del cromosoma con una probabilidad de transducción del 100 %, o alternativamente obtener “resultados buenos” en poblaciones heterogéneas, y por tanto más preparadas para un cambio súbito de ambiente transfiriendo fragmentos cromosómicos con una probabilidad de transducción del 50 %. Esta última opción podría aplicarse en Algoritmos Genéticos en Tiempo Real, como el utilizado en (Lin *et al.*, 2003) donde el problema a resolver no es estático.

En los experimentos realizados no se han incluido otros factores que deberían ser estudiados en detalle. Por ejemplo, la calidad de la solución y la velocidad de mejora del *fitness* y su posible relación con el número de colonias en la simulación, el número de bacteriófagos a usar (en los experimentos sólo se utilizaba uno), etc. Otra posibilidad interesante sería la que contempla estudiar la aplicación de sistemas expertos (SE) (Stevens, 1984) en hibridación con el algoritmo PETRI. En tal caso los bacteriófagos de PETRI serían un medio idóneo de transmisión de la información entre el SE y PETRI, provocando así una mejora en los resultados.

El diseño del receptor de radio AM con el Algoritmo Genético Bacteriano o SBGA debe considerarse sólo como un ejemplo ilustrativo de la eficiencia del operador de conjugación bacteriano, por lo que no se comenta nada acerca de la complejidad o grado de dificultad de este problema. Tampoco se considera el estudio del coste computacional requerido para el diseño de un circuito electrónico, ya que dependería del tipo específico de circuito a solucionar. Sin embargo, en la bibliografía hay gran variedad de referencias sobre estudios de complejidad en circuitos, *hardware*, etc. Por ejemplo, Chaudhuri y Radhakrishnan (1996) estudiaron la complejidad de un circuito compuesto por puertas AND, OR y NOT. Widiger *et*

*al.* (2006) desarrollaron una función Hash para *hardware* evolutivo considerando la complejidad de dicha arquitectura.

El problema práctico elegido en esta Memoria se asemeja a otros problemas de optimización (Johnson y Rahmat-Samii, 1997) que generalmente conllevan una búsqueda de valores discretos, continuos, o ambos tipos incluyendo normalmente límites en los valores permitidos. Tal y como se muestra en Johnson y Rahmat-Samii (1997) los algoritmos genéticos son particularmente útiles cuando se aplican a este tipo de problemas.

Finalmente, proponemos que la Genética Microbiana (Streips y Yasbin, 2002), esto es la forma en la que la información genética es heredada en microorganismos tales como por ejemplo las bacterias, sus características (ADN microbiano, expresión de genes en operones, etc.), los mecanismos de transferencia genética bacteriana (transformación, transducción, conjugación, transposones, etc.) así como la genética de virus como los bacteriófagos, y un largo etcétera abren la posibilidad de desarrollar nuevos algoritmos, operadores genéticos y protocolos de simulación, siendo una fuente de inspiración en el campo de la Computación Evolutiva.

## **Capítulo 6**

### **Conclusiones**

En esta Memoria hemos llegado a las siguientes conclusiones:

1. Los algoritmos propuestos, SBGA y PETRI, recogen de forma suficiente la lógica y los elementos básicos presentes en la evolución de una colonia de bacterias a través de la selección natural, a través de dos mecanismos de transferencia horizontal de información genética, la conjugación y la transducción, respectivamente.
2. El operador conjugación tiene la misma eficacia que la recombinación o sobrecruzamiento clásicos cuando el problema de optimización implica un único rasgo o característica a optimizar.
3. El operador conjugación se muestra especialmente eficaz con respecto a la recombinación clásica cuando el problema de optimización implica dos rasgos distintos o características a optimizar.
4. Cuando el problema de optimización implica dos rasgos distintos o características a optimizar, el orden de eficiencia de los operadores estudiados es de mayor a menor: conjugación CORP, recombinación con un punto de corte, conjugación COFP, y recombinación con dos puntos de corte.
5. El operador transducción, una vez incluido en el algoritmo SBGA (Algoritmo PETRI), aumenta la velocidad y eficacia en el problema de optimización.
6. La política más eficiente de selección de bacteria y placa donadoras es, en la transducción, aquella que elige como donadoras a la bacteria y placa con *fitness* máximos. Mientras que la política de selección de bacteria y placa receptoras más eficiente es aquella que las elige por métodos estocásticos.
7. En la transducción el resultado es óptimo cuando se transfiere el cromosoma completo con una probabilidad de transducción del 100 %.
8. En la transducción se obtienen poblaciones o colonias con mayor variabilidad, y por tanto más adecuadas para un ambiente no estacionario o cambiante, cuando se transfieren fragmentos de cromosomas con una probabilidad de transducción del 50 %.
9. Los resultados obtenidos en el problema consistente en el diseño de un receptor de radio AM, sugieren la utilidad práctica de los algoritmos propuestos en problemas de optimización del mundo real.

# Bibliografía

- AICKELIN, U. y DOWSLAND, K.A. (2004). «An indirect Genetic Algorithm for a nurse-scheduling problem». *Computers & Operations Research*, **31(5)**, pp. 761–778.
- ANDERSON, N.G. (1970). «Evolutionary significance of virus infection». *Nature*, **227**, pp. 1346–1347.
- BEATTY, J.T.; OVERMANN, J.; LINCE, M.T.; MANSKE, A.K.; LANG, A.S.; BLANKENSHIP, R.E.; VAN DOVER, C.L.; MARTINSON, T.A. y PLUMLEY, F.G. (2005). «An obligately photosynthetic bacterial anaerobe from a deep-sea hydrothermal vent». *Proc Natl Acad Sci U.S.A.*, **102(26)**, pp. 9306–9310.
- BENTLEY, K. (2002). «Evolving asynchronous adaptive systems for an exploration of aesthetic pattern formation». *EASy MSc Thesis, Computer Science Department, University College London*.
- BENVENISTE, R.E. (1985). «The contribution of retroviruses to the study of mammalian evolution». *Molecular Evolutionary Genetics*, R.J. MacIntyre, Ed. New York, **Plenum**, pp. 359–417.
- BIES, R.R.; MULDOON, M.F.; POLLOCK, B.G.; MANUCK, S.; SMITH, G. y SALE, M.E. (2006). «A Genetic Algorithm-Based, Hybrid Machine Learning Approach to Model Selection». *Journal of Pharmacokinetics and Pharmacodynamics*, **33(2)**, pp. 195–221.
- BIRGE, E.A. (2000). *Bacterial and Bacteriophage Genetics*. Springer-Verlag, New York, EE.UU..
- BJORKLOF, K.; SUONIEMI, A.; HAAHTELA, K. y ROMANTSCHUCK, M. (1995). «High frequency of conjugation versus plasmid segregation of RP1 in epiphytic *Pseudomonas syringae* populations». *Microbiology*, **141**, pp. 2719–2727.
- BROWN, R.H.J. (1963). «The flight of birds». *BioSystems*, **38**, pp. 460–489.

- CANTÚ-PAZ, E. (1998). «A survey of parallel genetic algorithms». *Calculateurs Paralleles, Reseaux et Systems Repartis*, **10(2)**, pp. 141–171.
- CHAUDHURI, S. y RADHAKRISHNAN, J. (1996). «Deterministic restrictions in circuit complexity». *Proceedings of the Twenty-Eight Annual ACM Symposium on the Theory of Computing*, **STOC 96**, pp. 30–36.
- CHEN, J. y WINEBERG, M. (2004). «Enhancement of the shifting balance genetic algorithm for highly multimodal problems». En: *Congress of Evolutionary Computation*, volumen 1, pp. 744–751. CEC2004, Portland.
- CHRISTIE, P.J. (2001). «Type IV secretion: intercellular transfer of macromolecules by systems ancestrally related to conjugation machines». *Microreview. Molecular Biology*, **40**, pp. 249–305.
- COHOON, J.P.; MARTIN, W.N. y RICHARDS, D.S. (1987). «A multi-population genetic algorithm for solving the k-partition problem on hyper-cubes». En: *Proceedings of the Fourth International Conference on Genetic Algorithms*, R.K. Belew and L.B. Booker, volumen 1, pp. 244–248. Morgan Kaufman, San Mateo, CA.
- COHOON, J.P.; MARTIN, W.N. y RICHARDS, D.S. (1991). «Punctuated equilibrium: A parallel genetic algorithm». En: *Proceedings of the Fifth International Conference on Genetic Algorithms*, J.J. Grefenstette, Ed., Hillsdale, NJ, volumen 1, pp. 148–154. Lawrence Erlbaum Associates, San Mateo, CA.
- DAVIS, B.D.; DULBECCO, R.; EISEN, H.N. y GINSBERG, H.S. (1990). *Microbiology*. Lippincot Company, Philadelphia.
- DAVIS, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
- DE JONG, K.A. (1975). *Analysis of the behavior of a class of genetic adaptive systems*. Tesis doctoral, Dept. Comput. Sci., Univ. Michigan, Ann Arbor, MI.
- DI PAOLA, V.; MARIJUAN, P.C. y LAHOZ-BELTRÁ, R. (2004). «Learning and evolution in bacterial taxis: an operational amplifier circuit modeling the computational dynamics of the prokaryotic 'two component system' protein network». *BioSystems*, **74**, pp. 29–49.
- FERNANDEZ-LOPEZ, R.; MACHÓN, C.; LONGSHAW, C.M.; MARTIN, S.; MOLIN, S.; ZECHNER, E.L.; ESPINOSA, M.; LANKA, E. y DE LA CRUZ, F. (2005). «Unsaturated fatty acids are inhibitors of bacterial conjugation». *Microbiology*, **151**, pp. 3517–3526.

- FONTANA, W. (1991a). «Algorithmic chemistry». *Artificial Life II*, C.G. Langton, C. Taylor, J.D. Farmer, S. Rasmussen, Eds. Addison-Wesley, pp. 159–209.
- FONTANA, W. (1991b). «Functional self-organization in complex systems». *L. Nadel, D.L. Stein, Eds. 1990 Lectures in Complex Systems*, Addison-Wesley, pp. 407–426.
- FORDE, S.E.; THOMPSON, J.N. y BOHANNAN, B.J.M. (2004). «Adaption varies through space and time in a coevolving host-parasitoid interaction». *Nature*, **431**, pp. 841–844.
- FUNES, P. y POLLACK, J.B. (1999). «Computer evolution of buildable objects». *Evolutionary Design by Computers*, P. Bentley, Ed. San Francisco, CA: Morgan-Kaufmann, pp. 387–403.
- FURUHASHI, T.; MIYATA, Y.; NAKAOKA, K. y UCHIKAWA, Y. (1995). «A new approach to genetic based machine learning for efficient finding of fuzzy rules». *Lecture Notes in Artificial Intelligence*, **1011**, pp. 173–189.
- GOLDBERG, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.
- GORGES-SCHLEUTER, M. (1989). «ASPARAGOS An Asynchronous Parallel Genetic Optimization Strategy». En: J.D. Schaffer (Ed.), *ICGA*, pp. 422–427. Morgan Kaufmann.
- GORGES-SCHLEUTER, M. (1991). «ASPARAGOS: A parallel genetic algorithm and population genetics». *Lecture Notes in Artificial Intelligence*, **565**, pp. 407–418.
- GOULD, S.J. y ELDREDGE, N. (1977). «Punctuated equilibria: the tempo and mode of evolution reconsidered». *Paleobiology*, **3**, pp. 115–151.
- GREFENSTETTE, J.J. (1981). «Parallel adaptive algorithms for function optimization». *Vanderbilt University, Computer Science Department, Nashville, TN, EE.UU.*, **Tech.Rep.No. CS-81-19**.
- HAMEROFF, S.R.; DAYHOFF, J.E.; LAHOZ-BELTRÁ, R.; SAMSONOVICH, A.V. y RASMUSSEN, S. (1992). «Models for molecular computation: conformational automata in the cytoskeleton». *IEEE Computer (Special Issue Molecular Computing)*, **25**, pp. 30–39.
- HARAYAMA, S. (1998). «Artificial evolution by DNA shuffling». *Trends Biotechnol*, **16**, pp. 76–82.

- HARDING, S. y MILLER, J.F. (2005). «Evolution in materio: evolving logic gates in liquid crystal». C. Teuscher, A. Adamatzky, Eds. *European Conference on Artificial Life (ECAL 2005), Workshop on Unconventional Computing: From cellular automata to wetware*, Beckington, UK: Luniver Press, pp. 133–149.
- HARVEY, I. (1996). «The microbial genetic algorithm». No publicado.
- HIROYASU, T.; MIKI, M. y NEGAMI, M. (1999). «Distributed genetic algorithms with randomized migration rate». *IEEE Proc. Of Systems, Man and Cybernetics Conference (SMC'99)*, **565**, pp. 695–700.
- HOLLAND, J.H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor.
- HOPFIELD, J.J. (1982). «Neuronal networks and physical systems with emergent collective computational abilities». *Proc. Nat. Acad. Sci.*, **79**, pp. 2554–2558.
- JOHNSON, J.M. y RAHMAT-SAMII, Y. (1997). «Genetic algorithms in engineering electromagnetics». *IEEE Antennas and Propagation Magazine*, **39**, pp. 7–21.
- JONES, L.J.F.; CARBALLIDO-LOPEZ, R. y ERRINGTON, J. (2001). «Control of cell shape in bacteria: Helical, actin-like filaments in *Bacillus subtilis*». *Cell*, **104(6)**, pp. 913–922.
- KORB, J. (2003). «Thermoregulation and ventilation of termite mounds». *Naturwissenschaften*, **90(5)**, pp. 212–219.
- KÁRMÁN, T. VON (1954). *Aerodynamics - Selected Topics in the Light of their Historical Development*. Cornell University Press, Ithaca, NY, EE.UU..
- KRUSKAL, W.H. y WALLIS, W.A. (1952). «Use of ranks in one-criterion variance analysis». *American Statistical Association*, **47**, pp. 583–621.
- KUBOTA, N.; SHIMOJMA, K. y FUKUDA, T. (1996). «The role of virus infection in virus-evolutionary genetic algorithm». En: *Proceeding of the IEEE International Conference on Evolutionary Computation*, pp. 182–187. Proceeding of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan.
- LAHOZ-BELTRÁ, R. (2000). *Enzima electrónico con sitio activo reconfigurable para el reconocimiento de una señal digital de entrada y su procesamiento en otra señal digital de salida previamente establecidas*. Patente española nº 200000174 (véase Catálogo de Oferta Científico-Tecnológica, 2001, Editorial Complutense: 422a-422b).

- LAHOZ-BELTRÁ, R. (2001). «Evolving hardware as model of enzyme evolution». *BioSystems*, **61**, pp. 15–25.
- LAHOZ-BELTRÁ, R. (2004). *Bioinformática: Simulación, Vida Artificial e Inteligencia Artificial*. Ediciones Díaz de Santos.
- LAHOZ-BELTRÁ, R.; RINCÓN, C. RECIO y DI PAOLA, V. (2002). «Autómatas moleculares evolutivos: algoritmo SDS y sus aplicaciones». En: *Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados*, pp. 333–340. Eds. Alba, España.
- LAWLEY, T.D.; KLIMKE, W.A.; GUBBINS, M.J. y FROST, L.S. (2003). «F factor conjugation is a true type IV secretion system». *Minireview. FEMS Microbiology Letters*, **224**, pp. 1–15.
- LIEBERS, D.; DE KNIJFF, P. y HELBIG, A.J. (2004). «The herring gull complex is not a ring species». *Proc. R. Soc. Lond. B*, **271**, pp. 893–901.
- LIN, F.J.; CHOU, W.D. y HUANG, P.K. (2003). «Adaptive sliding-mode controller based on real-time genetic algorithm for induction motor servo drive». *IEE Proceedings - Electric Power Applications*, **150(1)**, pp. 1–13.
- LIU, H. y YANG, X. (2007). «Bus Transit Route Network Design Using Genetic Algorithm». En: *International Conference on Transportation Engineering 2007*, pp. 1135–1141. International Conference on Transportation Engineering 2007, Chengdu, China.
- LLOSA, M.; GOMIS-RÜTH, F.X.; COLL, M. y CRUZ, F. DE LA (2002). «Bacterial conjugation: a two-step mechanism for DNA transport». *Molecular Microbiology*, **45(1)**, pp. 1–8.
- MARGULIS, L. (1981). *Symbiosis in Cell Evolution*. W.H.Freeman & Company.
- MARGULIS, L. y SAGAN, D. (2002). *Acquiring Genomes: A Theory of the Origins of Species*. Perseus Books Group, Jackson, TN.
- MILLER, B.L. y GOLDBERG, D.E. (1995). «Genetic Algorithms, Tournament Selection, and the Effects of Noise». *Complex Systems*, **9**, pp. 193–212.
- MILLER, J.H. (1992). *A Short Course in Bacterial Genetics*. Cold Spring Harbor, Laboratory Press, New York, EE.UU..
- MÜHLENBEIN, H.; SCHOMISCH, D. y BORN, J. (1991). «The Parallel Genetic Algorithm as Function Optimizer». *Parallel Computing*, **17(6-7)**, pp. 619–632.

- NANG, J. y MATSUO, K. (1994). «A survey on the parallel genetic algorithms». *Journal of the Society of Instrument and Control Engineering*, **33(6)**, pp. 500–509.
- NOVOTNY, C.P. y FIVES-TAYLOR, P. (1974). «Retraction of F Pili». *Journal Of Bacteriology*, **117(3)**, pp. 1306–1311.
- PANSEGRAU, W. y LANKA, E. (1996). «Mechanisms of Initiation and Termination Reactions in Conjugative DNA Processing». *The Journal Of Biological Chemistry*, **271(22)**, pp. 13068–13076.
- PERALES-GRAVÁN, C. y LAHOZ-BELTRÁ, R. (2004). «Evolving morphogenetic fields in the zebra skin pattern based on Turing's morphogen hypothesis». *Int. J. Appl. Math. Comput. Sci.*, **14**, pp. 351–361.
- PERALES-GRAVÁN, C. y LAHOZ-BELTRÁ, R. (2007). «An AM radio receiver designed with a genetic algorithm based on a bacterial conjugation genetic operator». *IEEE Transactions on Evolutionary Computation*, **11(4)**.
- SEGAL, G.; PURCELL, M. y SHUMAN, H.A. (1998). «Host cell killing and bacterial conjugation require overlapping sets of genes within a 22-kb region of the *Legionella pneumophila* genome». *Proc. Natl. Acad. Sci. USA*, **95**, pp. 1669–1674.
- SIMOES, A. y COSTA, E. (2000). «Using genetic algorithms with asexual transposition». En: *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pp. 323–330. Morgan Kaufmann Publishers, San Francisco, CA. D. Whitley and D. Goldberg and E. Cantú-Paz and L. Spector and I. Parmee and H. Beyer.
- SIMOES, A. y COSTA, E. (2001). «On biologically inspired genetic operators: transformation in the standard genetic algorithm». En: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 584–591. Morgan Kaufmann Publishers, San Francisco, CA. L. Spector and E. Goodman and A. Wu and W.B. Langdon and H.M. Voigt and M. Gen and S. Sen and M. Dorigo and S. Pezeshk and M. Garzon and E. Burke.
- SMITH, P. (1996). *Conjugation - A Bacterially Inspired Form of Genetic Recombination*. pp. 167–176. J. R. Koza, Proc. Late Breaking Papers at the Genetic Programming Conf..
- STANIER, R.Y.; INGRAHAM, J.L.; WHEELIS, M.L. y PAINTER, P.R. (1986). *The Microbial World*. Prentice Hall.

- STEVENS, L. (1984). *Artificial Intelligence. The Search for the Perfect Machine*. p. 40. Hayden Book Company, Hasbrouck Heights, N.J..
- STREIPS, U.N. y YASBIN, R.E. (2002). *Modern Microbial Genetics*. Wiley-Liss, New York, EE.UU..
- SYVANEN, M. (1985). «Cross-species transfer. Implications for a new theory of evolution». *theor. Biol*, **112**, pp. 333–343.
- TEHEI, M. y ZACCAI, G. (2005). «Adaptation to extreme environments: macromolecular dynamics in complex systems». *Biochim Biophys Acta*, **1724(3)**, pp. 404–410.
- TENAILLON, O.; LE NAGARD, H.; GODELLE, B. y TADDEI, F. (2000). «Mutators and sex in bacteria: conflict between adaptive strategies». *PNAS*, **97**, pp. 10465–10470.
- THAI DAM, D. (1997). *MICRORAM: Un modelo orientado a la simulación de la evolución de una población en bacterias artificiales*. Tesina. Facultad de Biología, Universidad Complutense de Madrid.
- TIPLER, P.A. (1993). *Física*. Reverté, Barcelona.
- TÖRN, A. y ZILINSKAS, A. (1989). «Global Optimization». *Lecture Notes in Computer Science*, **350**.
- TUKEY, J.W. (1977). *Exploratory Data Analysis*. Addison-Wesley.
- WIDIGER, H.; SALOMON, R. y TIMMERMANN, D. (2006). «Packet classification with evolvable hardware hash functions - An intrinsic approach». A.J Ijspeert, T. Masuzawa, S. Kusumoto, Eds. *Lecture Notes in Computer Science, Biologically Inspired Approaches to Advanced Information Technology (BioADIT 2006)*, Berlin: Springer, **3853**, pp. 64–79.
- WOOD, T.H. (1968). «Effects of Temperature, Agitation, and Donor Strain on Chromosome Transfer in *Escherichia coli* K-12». *Journal of Bacteriology*, **96(6)**, pp. 2077–2084.
- WRIGHT, S. (1951). «The genetical structure of populations». *Ann. Eugen.*, **15**, pp. 323–354.
- ZELITCH, I. (1971). *Photosynthesis, photorespiration, and plant productivity*. Academic Press, New York.