

# Distribución de Cargas en una Esfera Mediante Estrategias Evolutivas

Cristóbal Luque del Arco-Calderón, Pedro Isasi Viñuela, Julio César Hernández Castro

**Resumen**— En este trabajo se plantea la resolución mediante estrategias evolutivas de un problema clásico de la física, el problema de Thomson, consistente en distribuir  $n$  cargas iguales en la superficie de una esfera o, lo que es lo mismo, hallar la distribución que hace mínimo el potencial electrostático de las cargas. La dificultad de hallar este mínimo radica en el hecho de que el número de las disposiciones estables aumentan exponencialmente con el número de cargas. Hasta la fecha aún no existe una función que relacione de forma exacta el potencial mínimo con el número de cargas, si bien se han propuesto aproximaciones asintóticas para dicha función. El objetivo de este trabajo es encontrar un algoritmo que evite estas dificultades y permita calcular configuraciones para más cargas con menor coste computacional. Como resultado se obtiene un método que mejora los estándares dentro de las estrategias evolutivas.

**Index Terms**— Optimización, Algoritmos Genéticos.

## I. INTRODUCCIÓN

UN problema clásico de la física es el problema de Thomson, consistente en distribuir  $N$  partículas con cargas iguales en la superficie de una esfera, de forma que se hallen en equilibrio [6][4]. Este problema clásico tiene aplicaciones en física, química y biología tales como la cristalización de iones o *bubblons* cerca de la superficie de un aglomerado de helio líquido, la cristalización de electrones (huecos) en el contorno de un punto semiconductor esférico, el sistema de iones enfriados con láser y los electrones en una trampa plana [8], y también podría tenerlas en el cálculo de poliedros regulares, en la búsqueda de moléculas estables y en el análisis del entrelazado de los granos de polen y las cadenas de ADN [16]. Para la resolución de éste problema se han propuesto numerosas aproximaciones, como el descenso del gradiente (*steepest descend*) [16][3], *simulated annealing* [11], algoritmos genéticos [9] y otros métodos [14]. En nuestro caso, se ha tratado de encontrar soluciones usando estrategias evolutivas [5][2][1]. En este artículo se enumeran las diversas aproximaciones exploradas y los resultados

obtenidos con cada una de ellas.

## II. EL PROBLEMA DE LA DISTRIBUCIÓN DE CARGAS

La configuración de cargas buscada debe minimizar el potencial electrostático, es decir, minimizar la función

$$f(N) = \sum_{i < j < N} \frac{1}{d(x_i, x_j)}$$

donde  $x_i$  representa la posición de la carga  $i$ , con  $i < N$ .

Supongamos que discretizamos la superficie de la esfera, esto es, dibujamos un enrejado sobre ella. Cuantos más meridianos y paralelos tracemos, más cerca estaremos de la idea de continuidad. Colocaremos cargas solamente en cada punto de intersección de un meridiano con un paralelo. Así pues, si hay 20 meridianos y 50 paralelos, tendremos 1000 posibles puntos donde situar una carga. Si suponemos que hay  $P$  posibles posiciones donde colocar la carga en la discretización de la esfera, el problema de colocar 2 cargas consiste en buscar el óptimo en un espacio de  $P(P-1)$  posibilidades (puesto que no puede haber 2 cargas en la misma posición). Si en lugar de dos queremos colocar  $N$  cargas estamos ante un problema con un espacio de búsqueda de tamaño:

$$P(P-1)(P-2)\dots(P-N+1) = P! / (P-N)!$$

Es decir, el espacio de búsqueda crece exponencialmente con el número de cargas.

En [15] se pueden encontrar valores del potencial para configuraciones de hasta 282 cargas. Estos valores parecen sugerir una relación entre en número de cargas y los mínimos absolutos del potencial. Para configuraciones superiores de cargas no se han realizado experimentos exhaustivos. Como aún se desconoce la ecuación que determina  $f$  en función de  $N$ , se debe usar como referencia las cotas y límites [7], que pueden encontrarse en la bibliografía, [4][8], siendo la propuesta óptima la [9] ya que es la que mejor aproxima a los resultados existentes.

El método clásico para encontrar la solución se basa en el análisis del gradiente de  $f(N)$  [16][3], para hallar, para cada carga, la dirección en que disminuye más rápidamente el potencial. El hecho de que la función potencial presente numerosos mínimos locales hace de este un problema muy complejo, tal y como ocurre con otros problemas clásicos, como el de Griewangk, Ackley o la *Generalized Rastrigin*

Investigación financiada por el Proyecto TIC2002-04498-C05-4 del Ministerio de Ciencia y Tecnología.

Cristóbal Luque del Arco-Calderón, (e-mail: [cluque@inf.uc3m.es](mailto:cluque@inf.uc3m.es)).

Pedro Isasi Viñuela, (e-mail: [isasi@ia.uc3m.es](mailto:isasi@ia.uc3m.es)).

Julio César Hernández Castro (e-mail: [jcesar@inf.uc3m.es](mailto:jcesar@inf.uc3m.es)).

Departamento de Informática, Universidad Carlos III de Madrid  
C/Butarque 15, E-28911 Leganés, España. <http://www.inf.uc3m.es>

*Function* [10]. De hecho, el número de mínimos locales aumenta exponencialmente con el número de cargas: p.e. para  $N \sim 200$  cargas, hay alrededor de 8000 mínimos locales [9] (en la bibliografía se suele denominar a estos mínimos locales estados metaestables del potencial). La ventaja que ofrecen las estrategias evolutivas, algoritmos genéticos y *simulated annealing* sobre un algoritmo como el *steepest descent* es que disponen de mecanismos para evitar el estancamiento en mínimos locales.

Para ilustrar estas dificultades, mostraremos un claro ejemplo de mínimo local para 8 cargas [11]: una solución local sería inscribir un cubo en la esfera, y los 8 puntos donde se tocan sería una disposición de las cargas que, sin embargo, no es el mínimo global para el potencial. Este mínimo global puede observarse en la Fig. 1 (8), junto con otras configuraciones para otros casos. Para obtener la disposición de las cargas que produce el potencial mínimo bastaría girar 45 grados la tapa superior del cubo inscrito en la esfera.

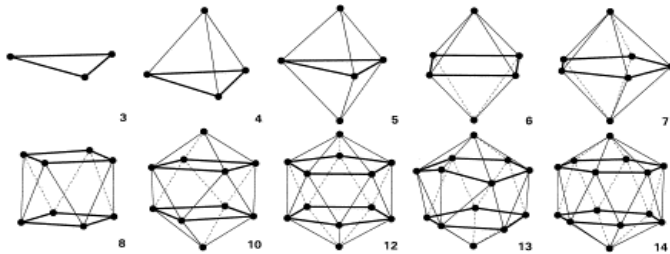


Fig. 1: Diversas configuraciones de cargas.

Aparte de la dificultad de evitar caer en, y de salir de los, mínimos locales, para resolver este problema hay que tener en cuenta otro hecho que contribuye a aumentar su complejidad: Alterar la posición de una carga hace que tengamos que modificar la posición de las restantes para encontrar la configuración mínima, esto es, todas las cargas dependen de todas y por tanto la posición de mínimo potencial para una carga depende de cómo estén situadas las  $N-1$  restantes. Si se modifica la posición de cualquiera de ellas, se deben cambiar también las posiciones de todas las demás.

En este artículo se pretende construir un método que permita obtener buenas aproximaciones a las soluciones en tiempos aceptables, esto es, que dichas aproximaciones no crezcan exponencialmente con el tamaño del problema.

### III. ESTRATEGIAS EVOLUTIVAS

Dado un problema de optimización de una función  $f(x_1, \dots, x_n)$ , la solución vendrá dada por cierto valor de las variables  $x_1, \dots, x_n$ . Los algoritmos evolutivos trabajan con una población de individuos, y cada individuo codifica la información (es decir, ciertos valores para las variables  $x_1, \dots, x_n$ ) de una posible solución al problema, análogamente a cómo codifica la información el ADN en cada individuo en la naturaleza. La idea de los algoritmos genéticos es producir generaciones de individuos en las cuales los hijos toman información de sus padres a partir de operadores de

sobrecruzamiento y mutación, y posteriormente seleccionar los mejores individuos, es decir, los que maximizan la función  $f$ , para que sean padres en la siguiente generación. La función  $f$  se suele llamar función *fitness*, dado que es la que mide la *aptitud* de cada individuo como solución del problema.

Las estrategias evolutivas se enmarcan dentro del paradigma de los algoritmos evolutivos [12][13]. La principal diferencia de las estrategias evolutivas con respecto a los otros algoritmos evolutivos es que en las estrategias evolutivas únicamente se usa el operador de mutación [5][2], es decir, no hay intercambio de información entre padres para tener un descendiente, lo cual efectivamente ocurre en otros paradigmas como los algoritmos genéticos o la programación genética [5][12][13]. En las estrategias evolutivas el proceso de engendrar produce los  $\lambda$  hijos a partir de los  $\mu$  padres mediante mutación. Hay varias formas de producir estos descendientes por mutación de los datos del padre, de las cuales detallamos las dos que usaremos en nuestros experimentos. La primera consiste en alterar aleatoriamente la posición de las cargas según una función aleatoria gaussiana. Matemáticamente escribimos este operador de mutación del elemento  $x$  como  $N(x, ex)$ , es decir, la mutación del valor de  $x$  es una variable aleatoria normal de media  $x$  y varianza  $ex$ . A su vez esta varianza también suele ser un parámetro que el descendiente hereda y muta según la fórmula  $ex * e^{N(0,A)}$ , donde  $A$  es una constante que se suele ajustar para cada problema. Esta varianza viene a representar cuán cerca está ese elemento en concreto de la solución: una varianza pequeña producirá, la mayoría de las veces, alteraciones pequeñas, y una varianza grande, alteraciones grandes, con lo cual es lógico que este parámetro se herede.

La segunda forma es la llamada *regla 1/5*, basada en los postulados de Rechenberg [2], que asegura que en una estrategia (1+1) la proporción de descendientes que sustituyen al padre debe ser 1/5. Si es mayor que 1/5 debemos incrementar la varianza, y si es menor, decrementarla. En este método la varianza se mantiene fija, y si la función a optimizar tiene  $n$  variables, cada  $n$  generaciones observamos la proporción de sustituciones del padre que se han producido en las últimas  $10n$  generaciones. Si esta proporción es mayor que 1/5, la varianza se multiplica por una constante de incremento  $c_i$ ; si es menor que 1/5, la varianza se multiplica por una constante de decremento  $c_d$ . Si la proporción es exactamente 1/5, entonces se mantiene la varianza. Como constantes se suelen usar  $c_d = 0,82$  y  $c_i = 1/0,82$ .

A la hora de seleccionar la población hay dos alternativas:  $(\mu+\lambda)$  y  $(\mu, \lambda)$ . En la  $(\mu+\lambda)$   $\mu$  padres producen  $\lambda \geq \mu$  hijos, y entre la población total de  $\mu+\lambda$  individuos se seleccionan los  $\mu$  mejores, que pasan a ser los padres de la siguiente generación. En la  $(\mu, \lambda)$   $\mu$  padres producen  $\lambda \geq \mu$  hijos, y de entre estos  $\lambda$  hijos se seleccionan los  $\mu$  mejores, que pasan a ser los padres de la siguiente generación.

Se puede encontrar una introducción a los algoritmos evolutivos en [5][2][12][13], y más concretamente a las estrategias evolutivas, en [1].

Para nuestra aproximación, los padres codificarán la

información de la posición de las cargas. La estrategia usada en un primer momento será una  $(\mu+\lambda)$ , siendo a veces una  $(1+1)$ . Finalmente la función de *fitness* será el potencial electrostático.

IV. EXPERIMENTOS

En este trabajo se plantea la solución al problema de Thomson mediante estrategias evolutivas. La condición de parada empleada es, o bien alcanzar un número determinado de generaciones, o bien que el mejor individuo de la población no haya variado en las últimas 5000 generaciones. Se han realizado varias aproximaciones a la solución, de las cuales sólo se comentan las que mejores resultados produjeron. En todos los casos se partía de una misma población inicial: todos los individuos eran idénticos al de partida, el cual codificaba una distribución uniforme de las cargas en el ecuador de la esfera. Se probó con otros tipos de poblaciones iniciales, pero los resultados fueron idénticos.

A. Primer método: Coordenadas polares

El primer método utilizado consiste en que los padres codifiquen las coordenadas de las cargas en coordenadas polares, tal como se ve en la Fig. 2. Las coordenadas cartesianas  $(x, y, z)$  se calculan a partir de las polares  $(\alpha, \beta)$  según la transformación:

$$\begin{aligned} x &= \cos \alpha \cos \beta \\ y &= \sin \alpha \cos \beta \\ z &= \sin \beta \end{aligned}$$

Cada individuo codifica, además de la posición en coordenadas polares  $(\alpha, \beta)$  de cada carga, las varianzas de la mutación  $e\alpha$  para la coordenada  $\alpha$  y  $e\beta$  para la coordenada  $\beta$  en cada carga. El proceso puede verse en la Fig. 3. Así pues la mutación (no sólo en las coordenadas, también en la varianza) seguirá las distribuciones normales de las gaussianas:

$$\begin{aligned} \alpha' &= N(\alpha, e\alpha) \\ \beta' &= N(\beta, e\beta) \\ e\alpha' &= e\alpha * e^{N(0,1)} \\ e\beta' &= e\beta * e^{N(0,1)} \end{aligned}$$

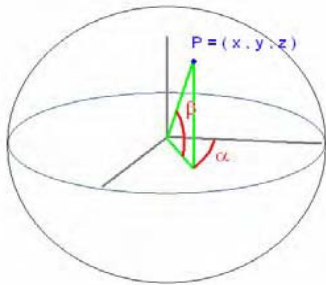


Fig. 2: Posición sobre una esfera en coordenadas polares

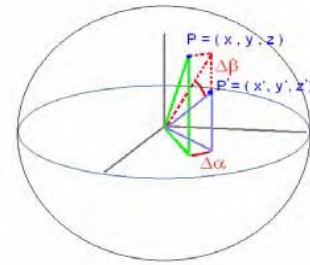


Fig. 3: Mutación de la posición de una carga en coordenadas polares

Siguiendo el procedimiento comentado anteriormente se han realizado una serie de experimentos para comprobar la eficacia de esta aproximación. En las figuras 4, 5 y 6 podemos observar 5 experimentos: para 8 cargas, 5 padres y 5 hijos; otros 5 experimentos para 47 cargas con 1 padre y 1 hijo; y por último 5 experimentos para 100 cargas con 10 padres y 10 hijos. En todos ellos se observó el *fitness* tras 10000 generaciones. Los experimentos están ordenados de mejor a peor aproximación, y se comparan con el mínimo absoluto.

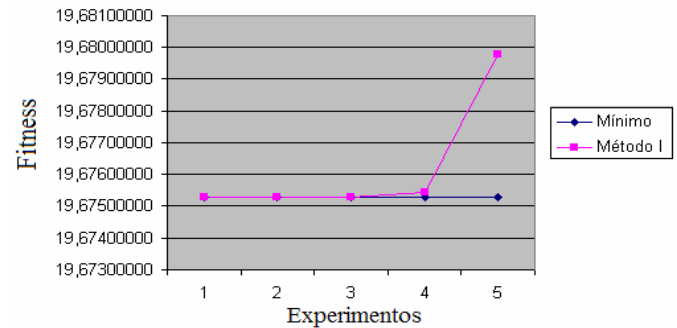


Fig. 4: Experimentos del método I para 8 cargas.

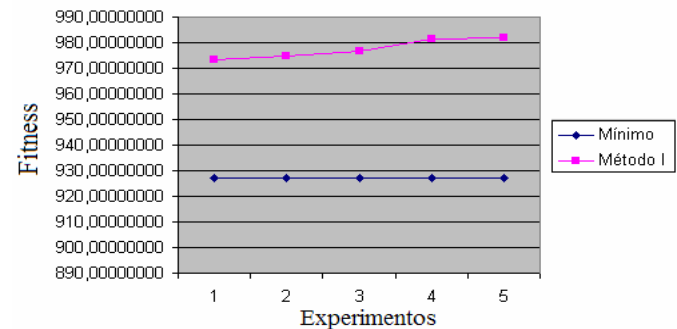


Fig. 5: Experimentos del método I para 47 cargas.

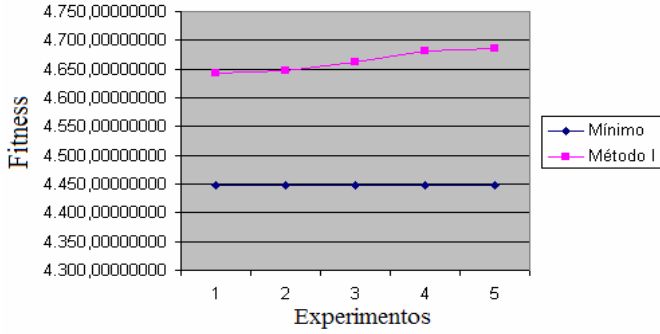


Fig. 6: Experimentos del método I para 100 cargas.

En esta aproximación se aprecia que las soluciones obtenidas no se acercan al valor mínimo, como sería deseable.

*B. Segundo método: Coordenadas cartesianas*

En esta nueva implementación se decidió trabajar directamente sobre las coordenadas cartesianas y mutarlas, para que el algoritmo se ahorrara los procesos de conversión. También se pensó en mantener una sola varianza común a todos las cargas del mismo individuo, ya que en caso de mantener una para cada coordenada de cada carga, el espacio en memoria requerido por cada individuo se incrementaría un 50%, ralentizando así los procesos de copia. Con el cambio indicado, no sólo no se aumenta el espacio en memoria del individuo, sino que se reduce un 50%. En este nuevo método, la mutación podría sacar la carga de la esfera unidad  $S^2$ , por lo cual se necesitó realizar un proceso de normalización para colocarlo de nuevo en la superficie de la esfera, tal como puede verse en la Fig. 7. El proceso de mutación seguiría las ecuaciones:

$$\begin{aligned} x' &= N(x, e) \\ y' &= N(y, e) \\ z' &= N(z, e) \\ e' &= e * e^{N(0,1)} \end{aligned}$$

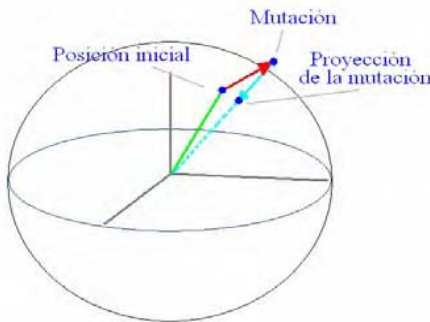


Fig. 7: Mutación de la posición de la carga, y posterior proyección sobre la esfera.

Tras llevar a cabo algunas pruebas, se comprobó que,

efectivamente, esta nueva implementación era más rápida. Estos resultados pueden verse en las figuras 8, 9 y 10, en los cuales se comparan 15 experimentos con el método II a sus análogos realizados para el método I. También los tiempos mejoraron notablemente.

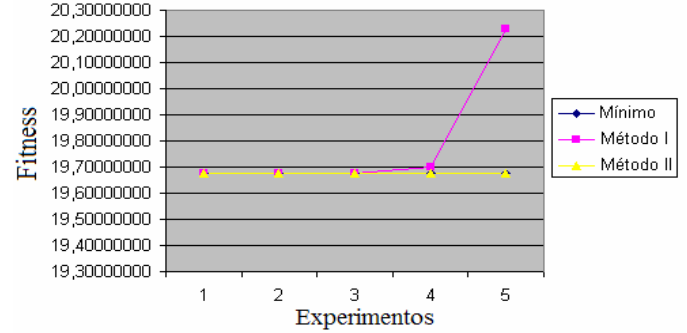


Fig. 8: Comparación de experimentos para 8 cargas, métodos I y II.

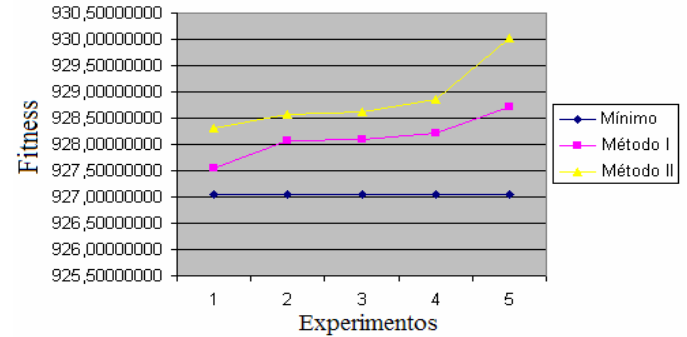


Fig. 9: Comparación de experimentos para 47 cargas, métodos I y II.

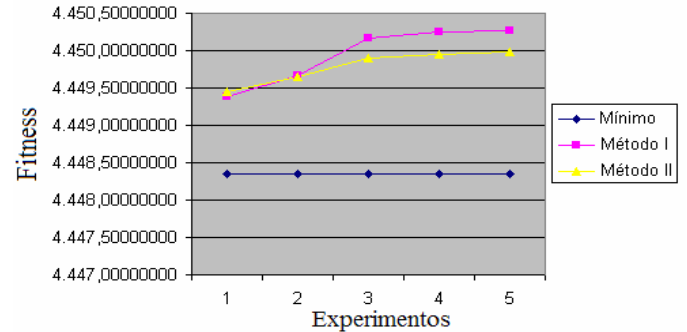


Fig. 10: Comparación de experimentos para 100 cargas, métodos I y II.

*C. Tercer método: Estrategia (1+1), regla 1/5*

Dado que la varianza era común a todos las cargas del mismo individuo, se planteó recurrir a una estrategia (1+1), es decir, un único padre engendra un único hijo, y se implementó el método recomendado por [2], usando la regla de 1/5 y las constantes  $c_d = 0,82$  y  $c_i = 1/0,82$ .

*D. Cuarto método: Estrategia (1+1), sin regla 1/5.*

Este último método es mucho más sencillo. Simplemente, cada generación decrementa la varianza del padre multiplicándola por 0,95, y la del hijo la aumenta

multiplicándola por 1,25. Este proceso, al igual que sucede en el tercer método, produce un descenso de la varianza más gradual, y en caso de encontrarse con una mejora del *fitness*, vuelve a aumentarla para explorar mejor las soluciones cercanas. Todos los métodos anteriores al cuarto producen unas mutaciones de las varianzas bastante caóticas y para nuestro caso concreto se comprobó en los resultados que no es lo más apropiado. Muy posiblemente, que un método u otro sea más apropiado para un problema dependa de la función de *fitness*. Para este caso concreto no hay ninguna duda de que de todos los métodos presentados hasta ahora, el cuarto es el mejor.

Se han hecho una serie de experimentos para los métodos II, III y IV: 5 con 47 cargas, otros 5 para 100 y otros 5 con 212 cargas. Todos ellos con estrategia 1+1 y 100.000 generaciones. En la Fig. 11 vemos una comparación de los 5 experimentos para 47 cargas ordenados por *fitness*. En la Fig. 12 tenemos la Comparación para 100 cargas, y en la Fig. 13 los 4 mejores resultados. En la Fig. 14 tenemos la comparación para 212 cargas, y en la 15 los 3 mejores resultados. En las figuras 16, 17 y 18 se muestra la evolución del *fitness* para los métodos II, III y IV con 3 nuevos experimentos para 47, 100 y 212 cargas respectivamente, estrategia 1+1, a lo largo de las generaciones. Se observa cómo presenta una convergencia mucho mayor que los otros métodos, y este hecho se hace más patente cuando aumenta el número de cargas.

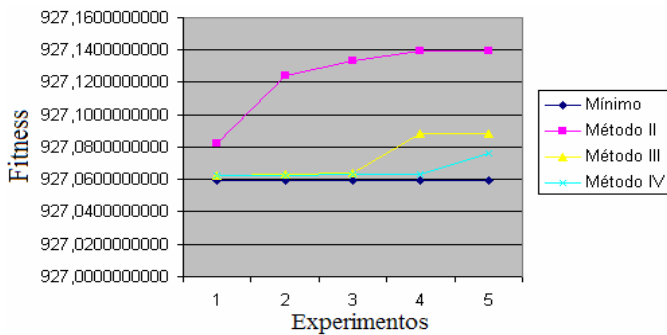


Fig. 11: Comparación de experimentos para 47 cargas, métodos II, III y IV.

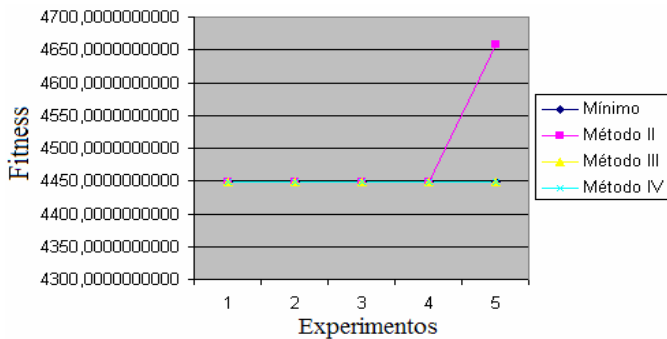


Fig. 12: Comparación de experimentos para 100 cargas, métodos II, III y IV.

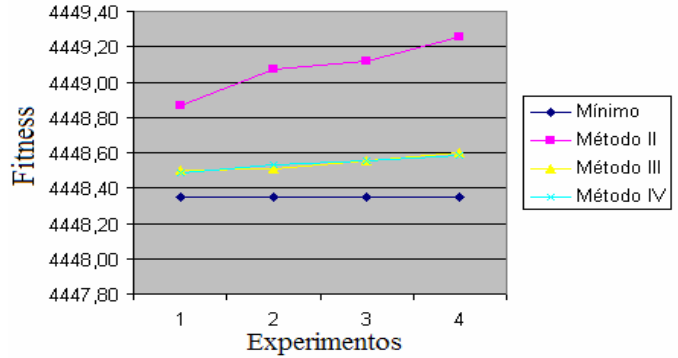


Fig. 13: Comparación de los 4 mejores resultados de la Fig. 12.

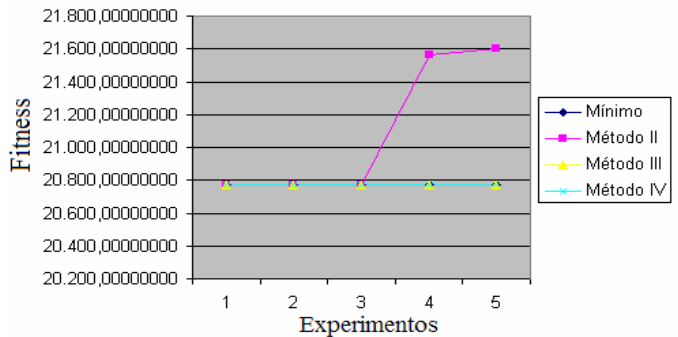


Fig. 14: Comparación de experimentos para 212 cargas, métodos II, III y IV.

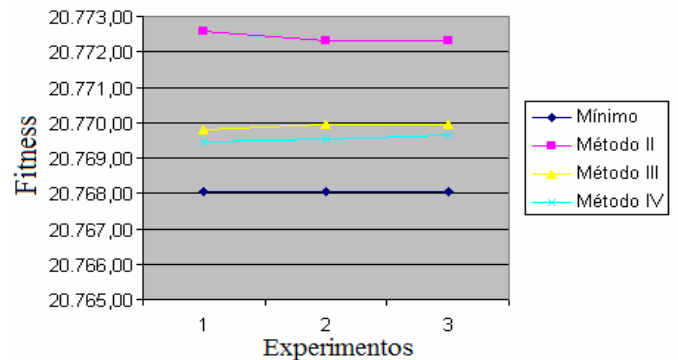


Fig. 15: Comparación de los 3 mejores resultados de la Fig. 14.

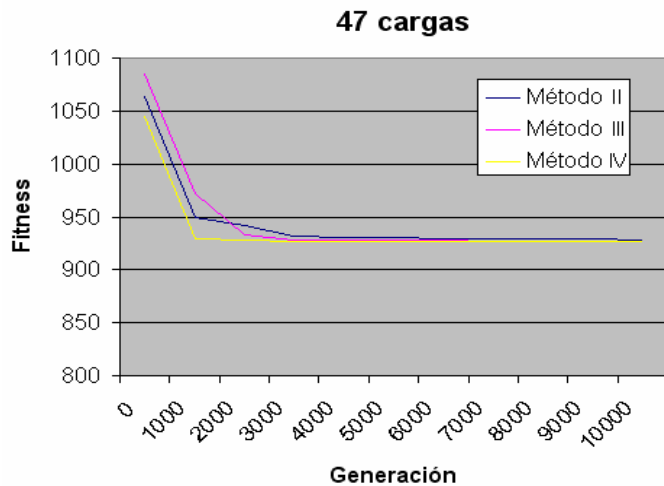


Fig. 16: Evolución del *fitness* para 47 cargas en los distintos métodos.

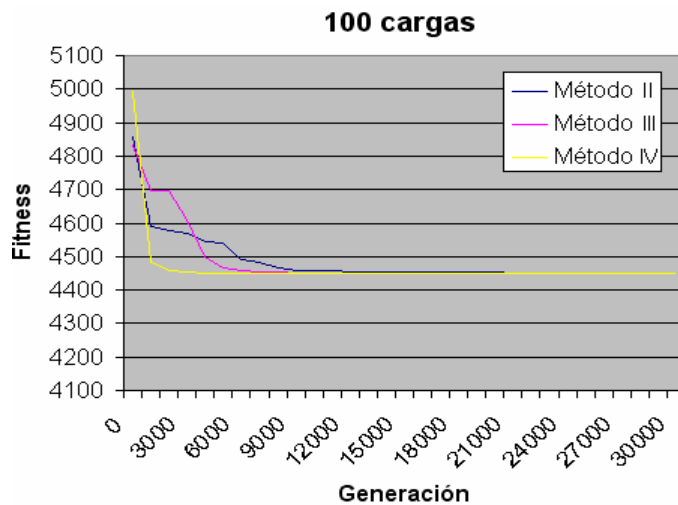


Fig. 17: Evolución del *fitness* para 100 cargas en los distintos métodos.

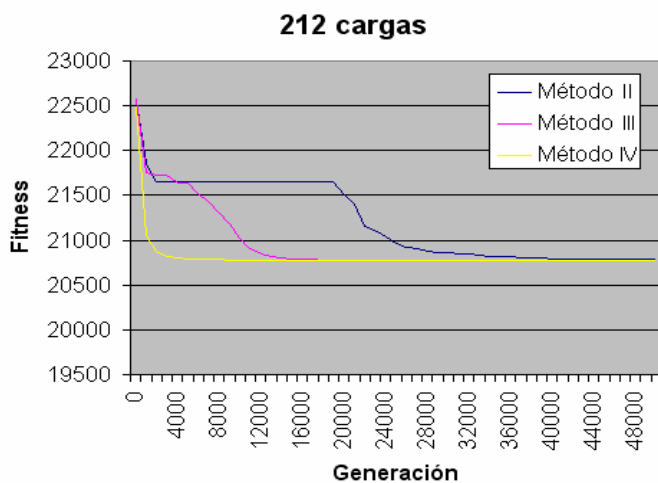


Fig. 18: Evolución del *fitness* para 212 cargas en los distintos métodos.

## V. CONCLUSIONES

El problema de Thomson presenta dos graves inconvenientes que aumentan significativamente su dificultad: El primero es que el potencial es una función de demasiadas variables,  $3n$  para  $n$  cargas. El segundo es que dicha función plantea un gran número de mínimos locales que crece exponencialmente con el tamaño del problema. Las estrategias evolutivas son una buena forma de evitar ambos inconvenientes, y más concretamente el método cuarto, que obtiene unas aproximaciones muy buenas para problemas de gran tamaño, en relativamente poco tiempo. Si bien el método cuarto mejora los métodos estándar (I, II y III) dentro de las estrategias evolutivas, sigue quedando atrapado en mínimos locales para configuraciones con gran cantidad de cargas. Para futuras investigaciones se buscan métodos que afronten problemas cuya función *fitness* tenga muchos mínimos locales, mediante otras perspectivas o simplemente cambiando la codificación. Una vez resuelto este problema se podrían encontrar las configuraciones mínimas para más de 282 cargas. También se busca, no sólo obtener resultados para mayor número de cargas, sino hacerlos más exactos.

## VI. REFERENCIAS

- [1] Thomas Bäck and Hans-Paul Schwefel, "Evolutionary Algorithms: Some Very Old Strategies for Optimization and Adaptation". In Perret-Gallix, pp. 247-254, 1992.
- [2] Thomas Bäck, Frank Hoffmeister, and Hans-Paul Schwefel, "A survey of evolution strategies". In Lashon B. Belew, Richard K.; Booker, editor, Proceedings of the 4th International Conference on Genetic Algorithms, pp. 2--9, San Diego, CA, July 1991.
- [3] U. Depeczynski and J. Stockler, "A differential geometric approach to equidistribution on compact manifolds", Approximation theory IX, Volume 1: Theoretical aspects 1998.
- [4] T. Erber, and G. M. Hockney, "Equilibrium configurations of  $n$  equal charges on a sphere", J Phys A: Math, 1991.
- [5] D. B. Fogel, "An introduction to simulated evolutionary optimization", IEEE transactions on neural networks, vol 5, n 1, Jan 1994.
- [6] A. B. J. Kuijlaars, E. B. Saff, "Distributing many points on a sphere", Mathematical Intelligencer, v19 n1 (1997), pp. 5-11.
- [7] A. B. J. Kuijlaars, E. B. Saff, "Asymptotics for minimal discrete energy on the sphere", Trans. Amer. Math. Soc., 350 (2), pp 523-538, 1998.
- [8] A. M. Livshits and Yu E. Lozovik, "Coulomb clusters on a sphere: topological classification", Chemical Physics Letters 314 (1999), pp. 577-583.
- [9] J. R. Morris, D. M. Deaven and K. M. Ho, "Genetic-algorithm energy minimization for point charges on a sphere", Physical Review B, 53(4): pp. 1740--1743, 1996.
- [10] A. Törn and A. Zilinskas, "Global Optimization", Lecture Notes in Computer Science, N° 350, Springer-Verlag, Berlin, 1989.
- [11] L. T. Wille, "Searching potential energy surfaces by simulated annealing", Nature v 324 n 6 (1984), pp. 46-48.
- [12] J.H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975.
- [13] M. Mitchell, "An introduction to Genetic Algorithms", Cambridge, MA: MIT Press, 1996, pp. 55-65.
- [14] A. Pérez-Garrido, M. Ortuño, E. Cuevas and J. Ruiz "Many-particle jumps algorithm and Thomson's problem", J. Phys. A: Math. Gen. 29 (1996), pp. 1973-1978.

### Webs de interés:

- [15] R. H. Hardin, N. J. A. Sloane and W. D. Smith <http://www.research.att.com/~njas/electrons/>

- [16] Neubauer, Schilling, Watkins & Zeitlin, 1998  
<http://www.csun.edu/~hcmth007/algorithm.html>
- [17] K. Brown  
<http://www.mathpages.com/home/kmath005.htm>

## VII. BIOGRAFÍAS



**Cristóbal Luque:** Nacido en Montilla, Córdoba. Licenciado en Matemáticas por la Universidad de Sevilla, actualmente es profesor del Departamento de Ingeniería Informática de la Universidad Carlos III de Madrid. Entre sus líneas de investigación destacan: los algoritmos genéticos, las estrategias evolutivas, la predicción de fenómenos caóticos y el go.



evolutivos y redes de neuronas artificiales.

**Pedro Isasi:** Es Licenciado y Doctor en Informática por la Universidad Politécnica de Madrid, desde 1994. Actualmente es Catedrático del área de Ciencias de la Computación e Inteligencia Artificial en el departamento de Informática de la Universidad Carlos III de Madrid. Sus investigaciones se centran en el campo de la Inteligencia Artificial, donde ha abordado problemas de Optimización y aprendizaje automático, fundamentalmente en sistemas



**Julio C. Hernández:** Es Licenciado en Matemáticas y doctor en Ingeniería Informática. Sus investigaciones están fundamentalmente orientadas a encontrar aplicaciones de técnicas de inteligencia artificial a la criptografía y la seguridad en redes. Entre sus hobbies confesables destaca el ajedrez y el go. Actualmente es investigador invitado en el INRIA.