

IMPLEMENTACIÓN PARALELA DE UN ALGORITMO GENÉTICO PARA LA OPTIMIZACIÓN DE UN DISPOSITIVO DE REFRIGERACIÓN

GUILLERMO A. BURGNER^{†*}, MARIO A. STORTI^{†*}

[†] Centro Internacional de Métodos Computacionales en Ingeniería (CIMEC), CONICET / U.N.L
INTEC
Güemes 3450, 3000 Santa Fe, Argentina
e-mail: guiyoteb@yahoo.com.ar

* Facultad de Ingeniería y Ciencias Hídricas (FICH), de la Universidad Nacional del Litoral
Ciudad Universitaria C.C. 217, 3000 Santa Fe, Argentina
e-mail: mstorti@intec.unl.edu.ar

Palabras claves: algoritmo genético, computación paralela, refrigeración, elementos finitos.

Resumen. *Los Algoritmos Genéticos (AGs) tratan de simular la evolución natural darwiniana, de allí que los componentes fundamentales de este mecanismo son los cromosomas, el material genético de un individuo biológico que determina sus características únicas. En general y particularmente en el problema seleccionado, uno de los inconvenientes que tienen los AGs es el elevado costo de procesamiento debido a que se requiere obtener el valor de la función objetivo para cada uno de los miles de individuos necesarios para llegar a la convergencia del método.*

En este artículo se utilizará un algoritmo genético para optimizar la geometría de un dispositivo de enfriamiento que emplea un fluido refrigerante que circula a través de aletas de disipación. Con la utilización de las técnicas de algoritmo genético se buscará aquella configuración óptima de las aletas que disipe la mayor cantidad de calor posible. Para solucionar el problema de alto costo de procesamiento, anteriormente mencionado, se ha optado por la paralelización del AG, la cual simplemente consiste en distribuir en forma dinámica la evaluación de los individuos (distintas configuraciones de aletas) de una población, en cada procesador. Gracias a la paralelización y a la utilización de estrategias de balance de carga se han obtenido eficiencias de cálculos elevadas.

1 INTRODUCCIÓN

Los algoritmos genéticos (AGs) junto con las estrategias evolutivas y la programación evolutiva forman parte de las nuevas técnicas empleadas para dar solución a numerosos problemas de optimización y búsqueda.

Los AGs tratan de simular la evolución natural darwiniana, de allí que los componentes fundamentales de este mecanismo son los cromosomas, el material genético de un individuo biológico que determina sus características únicas.

Para llevar a cabo esta simulación los AGs tratan de emular los fenómenos que se presentan durante la reproducción de las especies, a los cuales se le da el nombre de operadores artificiales de: selección natural, cruce y mutación. La competencia entre los individuos por los recursos naturales limitados y por la posibilidad de reproducción limitada permite que solo los más fuertes o más adaptados sobrevivan¹, lo cual significa que el material genético de los mejores individuos sobrevive y se reproduce, mientras que los genes de los individuos más débiles o menos adaptados, mueren o se extinguen; siendo esta la razón fundamental por la que se utilizan los AGs para la resolución de distintos problemas de optimización.

Los AGs se basan, en general, en la evaluación de la función costo (fitness) para un número muy elevado de individuos. Por lo tanto, un punto crítico de la implementación es reducir al mínimo el tiempo de evaluación de los fitness, y más aun, si estos tiempos son relativamente largos. Un ejemplo de ello es el problema que trataremos de resolver, la búsqueda de una configuración de aletas que disipen el calor en forma óptima. Por este motivo es imprescindible e inevitable la implementación de los AGs en paralelo (al menos para estos tipos de problemas), ya que de esta manera se minimiza el tiempo global de evaluación de fitness.

2 PLANTEO DEL PROBLEMA

Se requiere resolver el diseño de un dispositivo de enfriamiento, que disipe de manera óptima el calor proveniente de una determinada máquina (Figura 1). Una posible solución a este problema, es la utilización de un dispositivo, en el cual, en uno de sus extremos se introduzca un líquido refrigerante a una determinada temperatura y presión, ésta última, mayor a la presión de salida, de manera de obtener la circulación del líquido por el dispositivo, logrando así una disminución en la temperatura de la máquina.

Para obtener una mayor disipación de calor, se hará circular el líquido a través de aletas ubicadas dentro del dispositivo (Figura 2). Según la ubicación y diseño de las mismas, se logrará una mayor o menor disipación de calor.

De esta forma el problema a resolver será el de optimizar la ubicación de las aletas, para lograr así la mayor disipación de calor posible. La búsqueda se realizará a través de la utilización de técnicas de algoritmos genéticos, los cuales explorarán el espacio de soluciones buscando la configuración óptima de aletas.

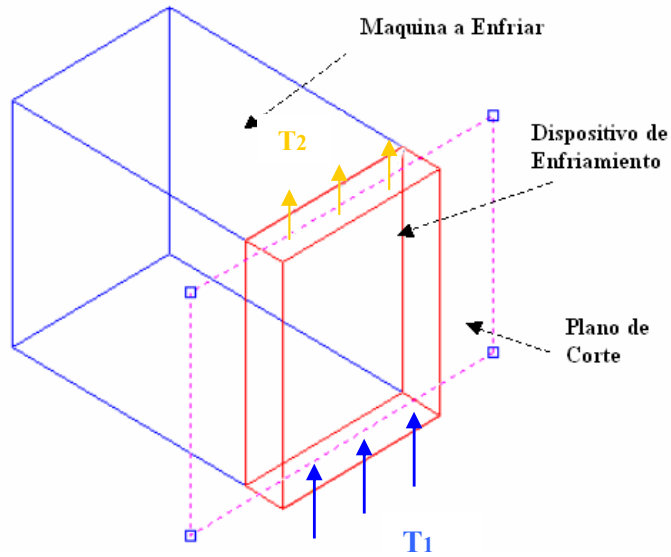


Figura 1: Gráfico simple del dispositivo de enfriamiento

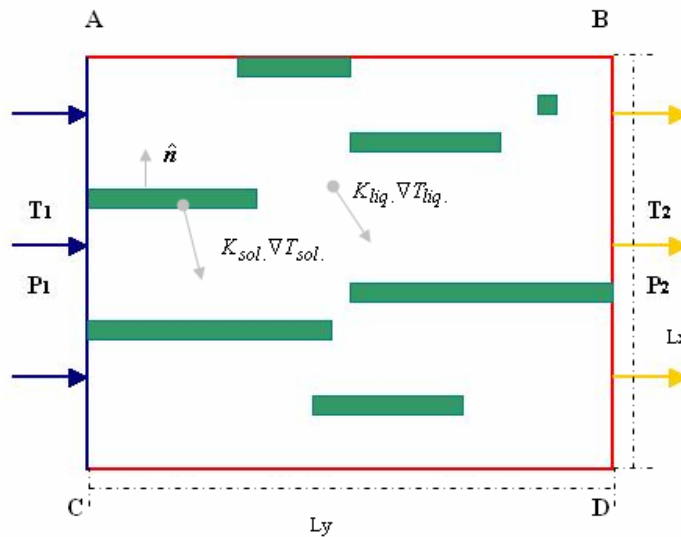


Figura 2: Vista de corte del dispositivo de enfriamiento

Las dimensiones del dispositivo son:

$$\begin{aligned} Lx &= 0,4m \\ Ly &= 4m \end{aligned} \tag{1}$$

Se ha utilizado una malla cartesiana con elementos rectangulares (la misma malla sobre el cual se define los individuos del AG), con las siguientes características:

$$Nx = Ny = 80 \tag{2}$$

$$hx = \frac{Lx}{Nx} = 0,005m \quad \wedge \quad hy = \frac{Ly}{Ny} = 0,05m \quad (3)$$

Las ecuaciones de gobierno son

Las ecuaciones de balance de momento, continuidad y balance de energía para el fluido:

$$\left(\begin{array}{l} v \cdot \nabla v = -\frac{\nabla p}{\rho} + \nu \Delta v \\ \nabla \cdot v = 0 \\ v \cdot \nabla T = \nabla \cdot (K_{liq.} \nabla T) \end{array} \right) \quad (4)$$

La ecuación de balance de energía para el sólido:

$$(\nabla \cdot (K_{sol.} \nabla T) = -Q) \quad (5)$$

Las ecuaciones de balance de energía en la interfase líquido-sólido:

$$\left(\begin{array}{l} v = 0 \\ (K_{sol.} \nabla T_{sol.} - K_{liq.} \nabla T_{liq.}) \cdot \hat{n} = 0 \end{array} \right) \quad (6)$$

Condiciones de contorno:

$$\left(\begin{array}{l} p = p_1 \text{ en } (AC) \\ p = p_2 \text{ en } (BD) \\ (AB) \text{ periódico con } (CD) \\ \frac{\partial T}{\partial n} = 0 \text{ en } (BD) \\ T = T_1 \text{ en } (AC) \end{array} \right) \quad (7)$$

Parámetros:

$$\left(\begin{array}{ll} p_1 = 100 & K_{sol.} = 100 \\ p_2 = 0 & K_{liq.} = 0,1 \\ Q = 10 & T_l = 0 \\ v = 0.1 & \end{array} \right) \quad (8)$$

3 APLICACIÓN DEL AG AL PROBLEMA

3.1 Estructura

El primer aspecto a resolver es la codificación del problema en un alfabeto finito. Tradicionalmente se emplea el sistema de representación binario por conveniencia para la realización de los operadores de cruce y mutación.

Cada individuo de una población tiene dos posibles representaciones (Figura 3), fenotipo (representación real), y genotipo (codificada a binarios), como se puede apreciar el genotipo es una matriz binaria, en la cual los “unos” representan las posiciones que se encuentran ocupadas por parte de una aleta, mientras que los “ceros” indican la falta de aletas, o dicho de otra manera, el lugar por donde puede circular el líquido refrigerante.

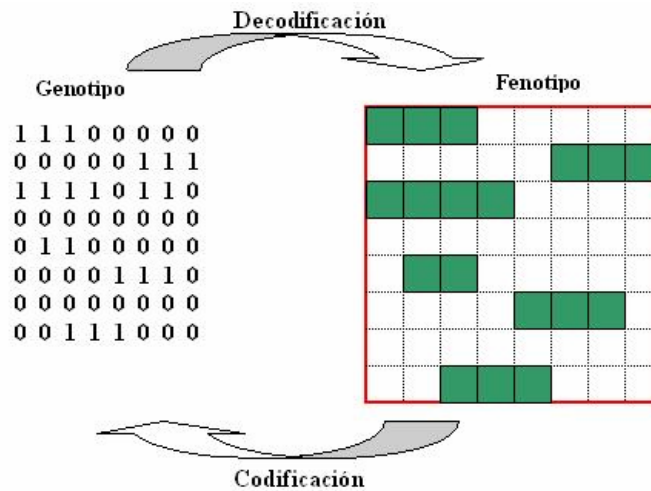


Figura 3: Representación de un individuo (cromosoma)

3.2 Función de Fitness

Debemos obtener una medida de que tan buena es la solución que nos ofrece cada individuo. Para ello utilizaremos la siguiente función de fitness:

$$Fitness(\text{individuo } n) = - \left(\max_{\forall i,j} (T_{i,j}) \right) \quad (9)$$

3137

Donde T_{ij} es la matriz de temperatura, obtenida según la configuración de las aletas que presenta cada individuo (Figura 4). Cabe señalar, que para hallar la matriz de temperatura, se ha utilizado la herramienta PETSC-FEM² provista por el CIMEC.

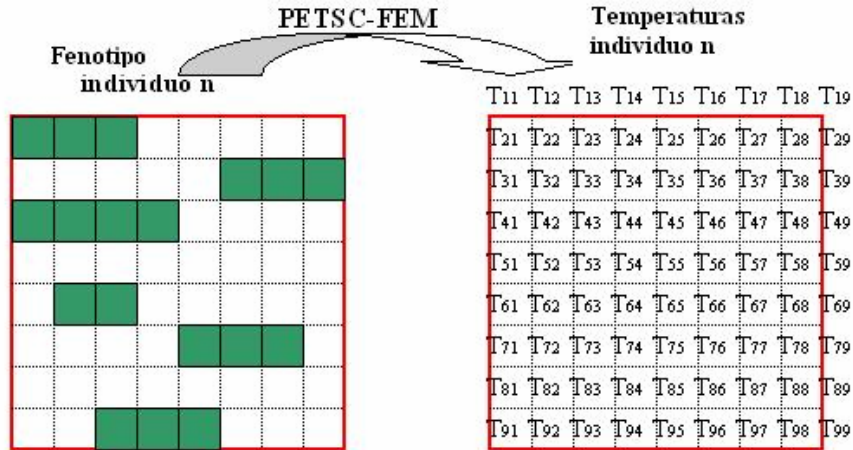


Figura 4: Obtención de matriz de temperaturas

3.3 Selección

Existen varias formas de realizar la selección de los progenitores, pero para cualquiera de ellas se debe tener en cuenta, que al igual que en la naturaleza, no se seleccionan simplemente los mejores, ya que la selección no está relacionada directamente con el fitness de un individuo, sino a través de operadores probabilísticos.

La selección es realizada por el método de competencia, en el que se eligen completamente al azar $k > 1$ individuos, luego se los hacen competir quedando seleccionado aquel que tenga mejor función de fitness. Generalmente se utiliza valores de k entre 2 y 5, para las corridas que hemos realizado se utilizó un $k = 2$, ya que de esta forma se obtiene una convergencia más rápida.

3.4 Reproducción

La reproducción es el proceso mediante el cual se obtiene la nueva población a partir de los individuos seleccionados de la generación actual, a los que luego se aplican los operadores de variación. Existen varias alternativas para realizar la reproducción; nosotros utilizamos la estrategia en el que se obtienen todos los individuos de la nueva generación a partir de las cruces y mutaciones de los progenitores seleccionados, pero con la excepción de que el mejor individuo de la generación anterior, independientemente de la selección y variaciones, se lo copie exactamente en la nueva población. De esta manera nos aseguramos de no perder la mejor solución generación tras generación, esta estrategia es llamada elitismo³, aunque no es un método muy biológico, nos permite elevar el índice de mutación y aumentar la dispersión de las soluciones en el espacio de búsqueda.

La cruce actúa sobre dos individuos para obtener otros dos. Existen dos tipos de cruza, cruce simple y cruza múltiple. Nosotros utilizamos la cruce simple, en la cual solamente se elige un punto de cruce totalmente al azar y luego se intercambian el material genético correspondiente.

La mutación se trata solamente de invertir los alelos (elementos del genotipo) con una probabilidad muy baja y uniforme en toda la población. El papel fundamental de la mutación es no dejar que todos los miembros de una población se conviertan en mediocres (caigan en mínimos locales).

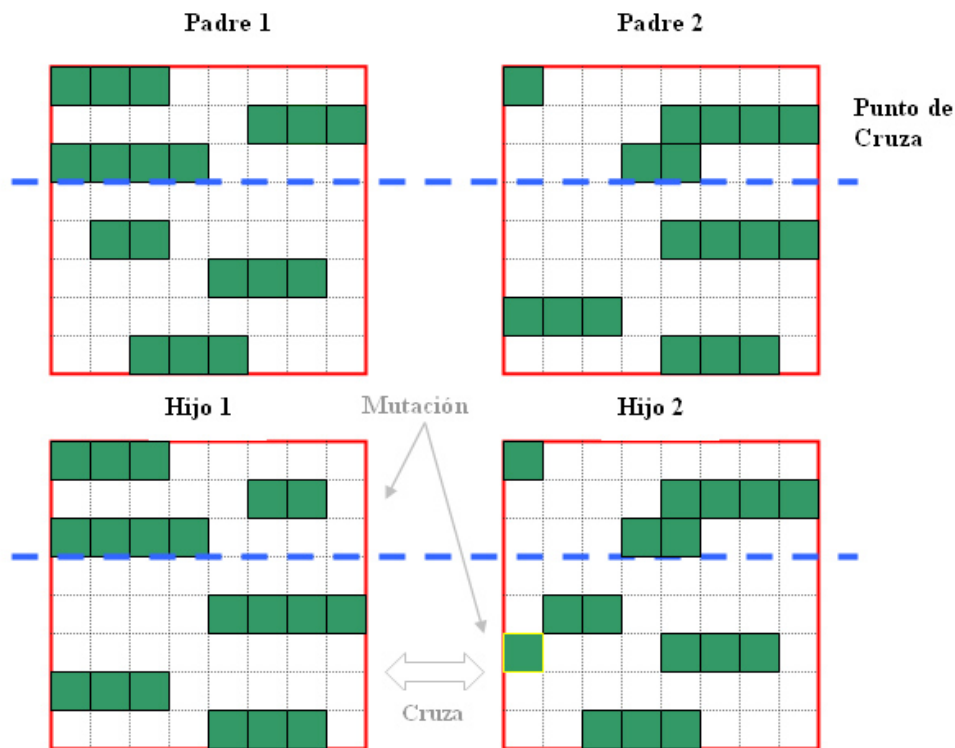


Figura 5: Cruza y mutación

4 PARALELIZACIÓN DEL AG

4.1 AGs Secuenciales y en Paralelo

Generalmente la implementación de los AGs se topa con la dificultad de un elevado tiempo de evaluación, que en ciertos casos imposibilita su implementación en forma secuencial. Un ejemplo de esta dificultad es la búsqueda de la solución del problema anteriormente planteado, ya que el tiempo requerido para hallar el fitness de cada individuo es de aproximadamente 1,25 minutos (malla de 80x80, Pentium IV con 1Gb. de memoria), si contamos con una población mínima de 100 individuos, la misma tardaría en ser evaluada unos 125 minutos, si además consideramos que aproximadamente para que el AG llegue a converger se

requieren 300 generaciones, su tiempo total de procesamiento sería extremadamente elevado, aproximadamente 26 días.

Una de las herramientas (a nuestro criterio, la única) posible para reducir eficientemente el tiempo de evaluación es la paralelización de los Ags. Para ello, se optó por la utilización de la librería MPI. ⁴, con la cual se realizó un algoritmo en paralelo con las siguientes características:

- Paralelismo Homógeno*: las mismas instrucciones en todos los procesadores, pero con distintos datos.
- Paralelismo Explícito*: paralelismo realizado por el programador, ya que existen compiladores que realizan la paralelización en forma semi-automática, como es el caso de HPF ⁵.
- Modelo de Transferencia de Mensajes*: utilización de memoria de acceso local, coordinación a través de transmisión de mensajes.
- Balance de Carga Dinámico*: tareas se distribuyen a medida que los procesadores lo soliciten.

4.3 Detalles de implementación en paralelo

El pseudo-código más común para los AGs se puede esquematizarse de la siguiente manera:

```
Inicio_Población( P );  
Evaluar_Población( P , MejorFitness);  
While ( MejorFitness < FitnessRequerido) do  
    Seleccionar( P, S);  
    Cruza&Mutación( S, P);  
    Evaluar_Población( P , MejorFitness);  
end;
```

Figura 6: Pseudo-código del AG

Una de las tareas más crítica, como hemos dicho anteriormente, es la evaluación de la población “*Evaluar_Población(P, MejoresFitness)*”, por ello solamente a esta parte del AG se le aplicó una simple paralelización.

Con dicho objetivo se ha realizado un algoritmo común para todos los nodos, pero a través de una sentencia “if - else”, una parte del algoritmo es solamente ejecutada por el “nodo-master” o por los “nodos-slaves”.

Como se puede observar (Figura 7), el nodo-master es el encargado de “generar” cada individuo a través de los operadores de selección, cruza y mutación, para luego repartirlos equitativamente (en este caso 3 individuos por nodo) a cada nodo-slave. Estos procesan los individuos recibidos, como se pueda apreciar en la figura 7; cuando algunos de los nodos terminan de evaluar todos los individuos ^① (envío del último individuo evaluado en el nodo-2), el nodo-master envía nuevamente un nuevo grupo de individuos ^②, de esta forma se garantiza un balance de carga dinámico en todo el cluster.

En el nodo-master también se evalúan algunos individuos con el objetivo de reducir su tiempo ocioso producidos por la espera del envío de los fitness evaluados en los nodos-slaves.

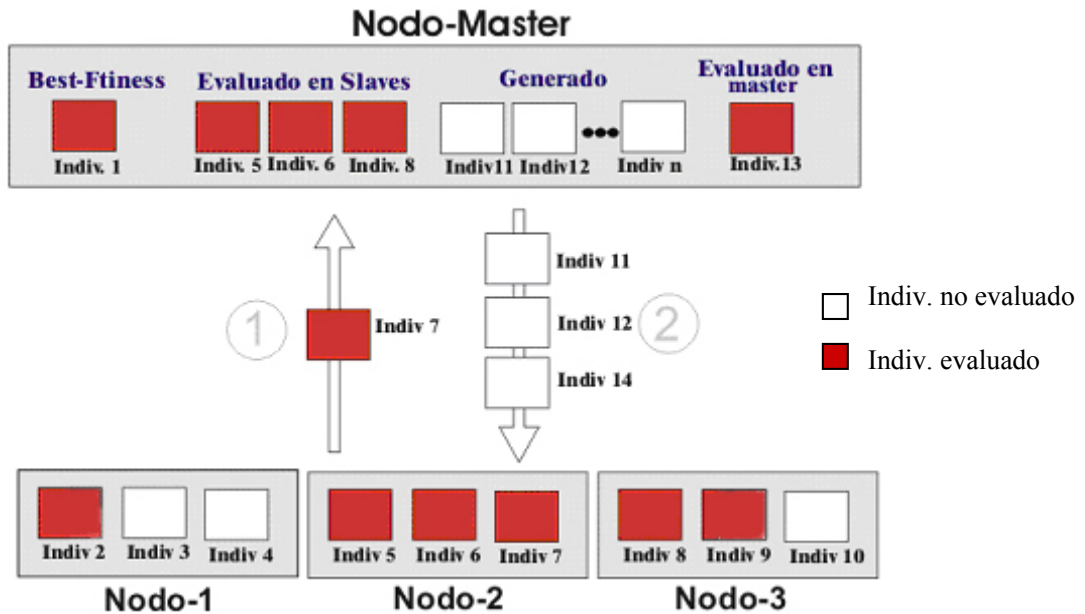


Figura 7: Esquema de intercambio de información entre los procesos

4.4 Problemas en la Convergencia.

Como hemos mencionado anteriormente, cada individuo es evaluado a través del programa PETS-FEM, el cual utiliza técnicas de elementos finitos para obtener la temperatura de cada nodo de la malla (Figura 4).

Inicialmente, para cierta diferencia de presiones p_2-p_1 y una dada viscosidad del fluido, la evolución del campo de velocidades puede ser estacionaria o entrar en un régimen no estacionario por emisión de vórtices tipo von Karman. En principio, para evaluar la eficiencia de una dada disposición de las aletas se debería realizar una simulación en estado transiente hasta llegar a un estado periódico y tomar un promedio temporal de las temperaturas. Como esto es muy costoso, nos hemos restringido a valores de los parámetros (fundamentalmente " p_2-p_1 " y " ν ") de tal forma que se llegue a un estado estacionario. De esta manera se puede resolver el problema en una serie de iteraciones del esquema de Newton Raphson sin necesidad de iterar a su vez en el tiempo. Sin embargo, para un dado juego de parámetros (" p_2-p_1 " y " ν ") no esta garantizado que el algoritmo de Newton-Raphson para capturar el estado estacionario converja para todos los individuos, ya que el problema sea estacionario o no depende también de la "densidad" de aletas. Si la densidad de aletas es grande, el obstáculo que se le opone al flujo es grande y se obtiene un estado estacionario. Por el contrario, si la densidad de aletas es baja, entonces el flujo se hace más intenso y se puede

llegar a un estado no-estacionario. Para un dado juego de parámetros (“ p_2-p_1 ” y “ v ”) y un individuo (es decir una distribución de aletas) si las condiciones son tales que el régimen de flujo resulta no-estacionario, entonces el algoritmo de Newton-Raphson no convergerá y declararemos al individuo como “penalizado”. En parte debido a esta solución, surgió un segundo problema; a medida que el AG convergía, la cantidad de individuos penalizados se incrementaba, llegando al punto en que casi toda la población era conformada por individuos penalizados (Figura 8), de esta forma la búsqueda se bloquea. Este problema es causado por la configuración óptima de aletas, que como veremos mas adelante, son aletas pequeñas distribuidas por todo el área en forma uniforme, este tipo de aletas nos lleva a estar muy cerca de la zona de no-convergencia, generalmente producido por vórtices formados por aletas pequeñas. Para evitar estos casos anómalos, donde se produce falta de convergencia, se restringe la longitud mínima de las aletas a un valor “ l_{min} ”. Para lograr esto, el algoritmo trabaja con genotipos que pueden tener longitud de aleta variable, pero al momento de evaluar el individuo se hace una convolución con una aleta de longitud “ l_{min} ”, lo cual equivale a extender todas las aletas una longitud “ l_{min} ” en la dirección del flujo. Desde el punto de vista del AG, implica una reducción del espacio de búsqueda y un remapeo de los genotipos.

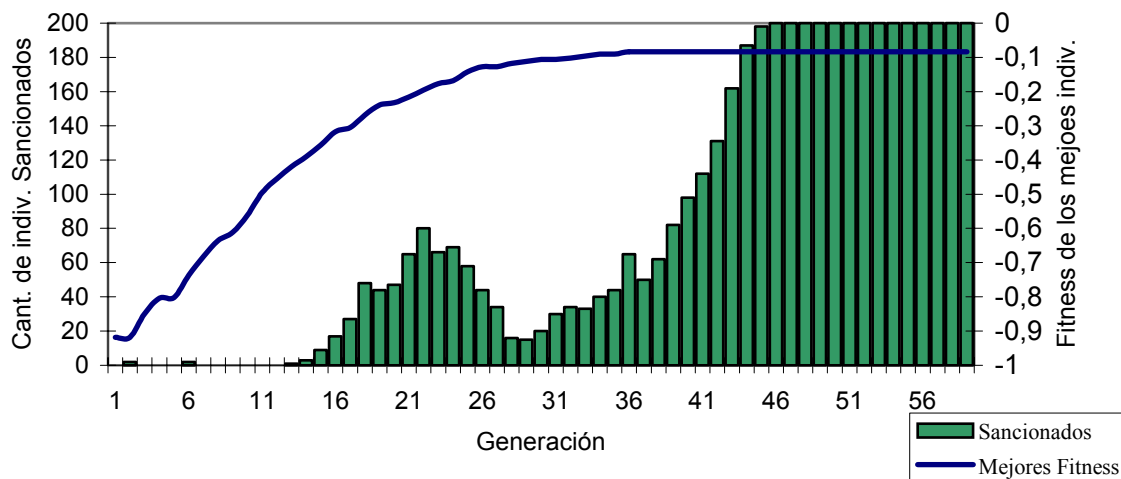


Figura 8: Individuos sancionados vs. fitness de una corrida bloqueada

5 RESULTADOS

5.1 Mediciones de eficiencia del Algoritmo

Para saber que tan eficiente es la paralelización del AG, se obtuvieron las medidas de speed-up y eficiencia. Para tal objetivo se realizaron dos corridas en paralelo con 7 nodos, con la única diferencia que una de ella contaba solamente con una población de 50 individuos, mientras que en la otra su población era de 500 individuos (cada individuo formado por una matriz de 80 x 80). El tiempo secuencial de cada corrida fue obtenido a través del tiempo

promedio en que un individuo demoraba en ser evaluado (1,25 min.) multiplicado por la cantidad de individuos de la población menos uno; ello es debido a la utilización del método elitista, en el cual el mejor individuo de cada generación se lo traslada directamente a la siguiente generación, por lo tanto no necesita ser evaluado. De esta manera se obtuvieron las siguientes estimaciones del tiempo de evaluación de una generación en forma secuencial:

Tiempo Secuencial para población de 50 individuos: 49 indiv. x 1.25 min. = 61.25 min.

Tiempo Secuencial para población de 500 individuos: 499 indiv. x 1.25 min. = 623.75 min.

Los tiempos promedios de las corridas en paralelo para una generación fueron los siguientes:

Tiempo en Paralelo, para población de 50 individuo = 9.33 min.

Tiempo en Paralelo, para población de 500 individuo = 83.70 min.

Por lo tanto, las mediciones de eficiencia de la paralelización para la corrida de 50 individuos son:

$$\begin{aligned} SpeedUp &= \frac{TiempoSecuencial}{TiempoParalelo} = \frac{61,25 \text{ min}}{9,33 \text{ min}} = 6,56 \\ Eficiencia &= \frac{SpeedUp}{Cant.Nodos} = \frac{6,56}{7} = 0,94 \end{aligned} \tag{10}$$

Corrida de 500 individuos:

$$\begin{aligned} SpeedUp &= \frac{TiempoSecuencial}{TiempoParalelo} = \frac{623,75 \text{ min}}{83,70 \text{ min}} = 7,45 \\ Eficiencia &= \frac{SpeedUp}{CantNodos} = \frac{7,45}{7} = 1,06 \end{aligned} \tag{11}$$

Como se puede apreciar en la corrida de 500 individuos hay un speed-up sospechoso, ya que el mayor speed-up posible es igual a la cantidad de nodos; este error es inducido por la aproximación en el calculo del tiempo secuencial; debido a que el tiempo de evaluación de cada individuo no es constante, si no que depende del tiempo de convergencia del algoritmo implementado en el programa PETS-FEM. El tiempo de evaluación por individuo que hemos utilizados (1,25 min.) fue obtenido de promediar los tiempos de evaluación de un grupo aleatorio de individuos; por lo tanto las medidas de efectividad pueden tener un pequeño error de estimación. Pero pese a ello, se puede vislumbrar que la implementación del algoritmo en paralelo es totalmente efectiva (aproximadamente un 94% de eficiencia), con lo cual se logra totalmente su objetivo, la reducción del tiempo de búsqueda. Esta reducción se puede apreciar

mucho mejor si pensamos que las corridas que realizamos para hallar la configuración óptimas son típicamente de 500 individuos, y que convergen en aproximadamente en 90 generaciones, por lo tanto las corridas en paralelo demoran alrededor de 5 días, en cambio si lo realizáramos en forma secuencial la misma llevaría un mínimo de 38 días.

5.2 Configuración óptima.

5.2.1 Características de las Corridas

Se efectuaron tres corridas, con el objetivo de encontrar una configuración óptima de aletas para el problema planteado. Las características y parámetros de cada corrida solamente difieren en longitud mínima tolerable de cada aleta, 7, 5 y 3 hy, ya que los parámetros típicos de los AGs son los mismos para cada una de ellas, 500 individuos por población y una probabilidad de 0,1 de mutación; también todos los individuos son representados a través de una matriz binaria de 80 filas y columnas.

Los resultados obtenidos fueron los siguientes:

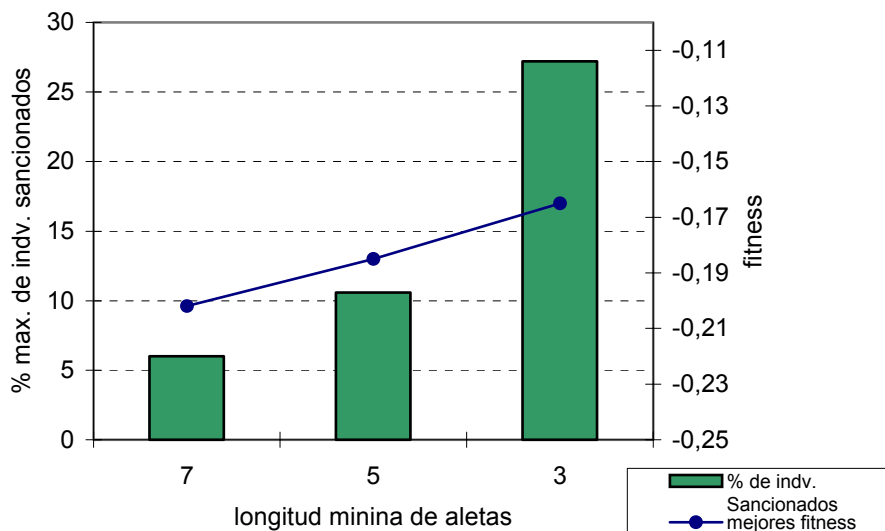


Figura 9: Porcentaje de individuos sancionados y fitness de los mejores individuos

Como se puede observar, a medida que se reduce la longitud mínima de cada aleta, se obtiene un mejor fitness, pero también se produce un incremento de la cantidad de individuos sancionados por generación (por lo mencionado en 4.4), lo cual provoca un empobrecimiento en la búsqueda del AG

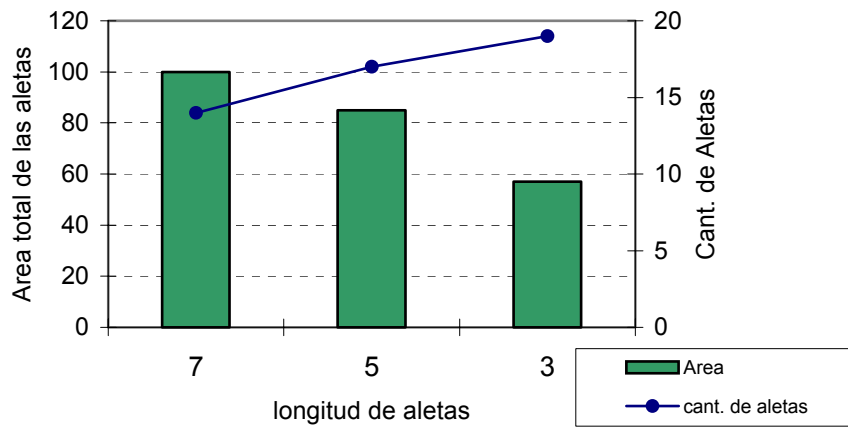


Figura 10: Area total vs. cantidad de aletas, de las mejores individuos

Como se visualiza en la Figura 10, el área total de las aletas de la mejor configuración obtenida en cada corrida, como era de esperar, disminuye al reducir la longitud mínima permitida para cada aleta, pero inversamente, la cantidad de aletas de los mejores individuos se incrementan, con lo cual podemos deducir que una cantidad considerable de aletas con pequeñas longitudes será la característica deseada para la configuración óptima que estamos buscando.

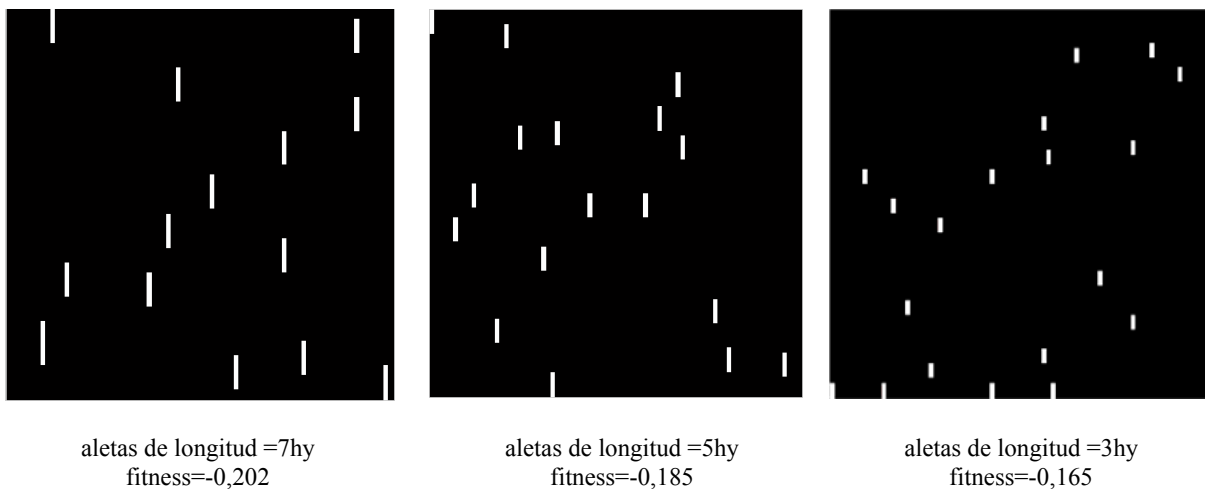


Figura 11: Mejores configuraciones de aletas para cada corrida

Otro hecho destacable, es que las aletas de las configuraciones óptimas que se han hallado en cada una de las tres corridas, tiene una longitud igual a la longitud mínima permitida (7, 5, y 3 hy), esto no hace más que afirmar la suposición realizada anteriormente, que aletas de pequeñas longitudes realizarán una mayor disipación de calor.

5.2.2 La mejor configuración.

Como expresamos anteriormente, al reducir la longitud mínima de las aletas se pueden hallar mejores resultados, pero también se incrementará la cantidad de individuos sancionados por generación, con lo cual se corre el riesgo de que la búsqueda no sea efectiva, y que por lo tanto, no se encuentre la configuración óptima para esa determinada mínima longitud de aleta.

Como podemos observar (Figura 12) con una longitud mínima de aleta igual a 3 hy, se llega a obtener un 27% de individuos sancionados por generación, por lo tanto la población de búsqueda se reduce a unos 365 individuos (73% de la población). A nuestro entender, tal reducción no afecta significativamente a la búsqueda, además como podemos apreciar, la misma se produce en generaciones elevadas.

Otro efecto de reducir la longitud mínima de las aletas a tres, se puede apreciar en la Figura 13, como vemos, el área total de la mejor configuración encontrada está muy cerca de las áreas totales de los individuos que no convergen, es por ello que se produce un incremento en la cantidad de individuos sancionados.

En la Figura 14 podemos apreciar la forma en que convergen los individuos de todas las generaciones; la misma se realiza relativamente rápida debido a la utilización de un método elitista. También podemos observar, en forma global, como el área total de las aletas de cada individuo disminuye a medida que la búsqueda converge.

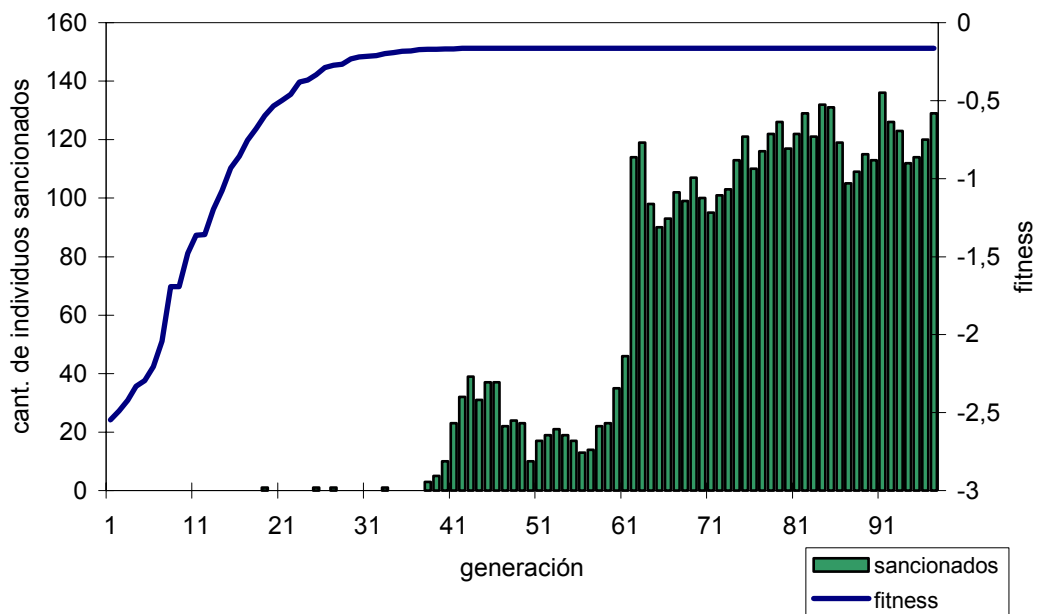


Figura 12: Cant. de individuos sancionados vs. fitness, de la mejor corrida

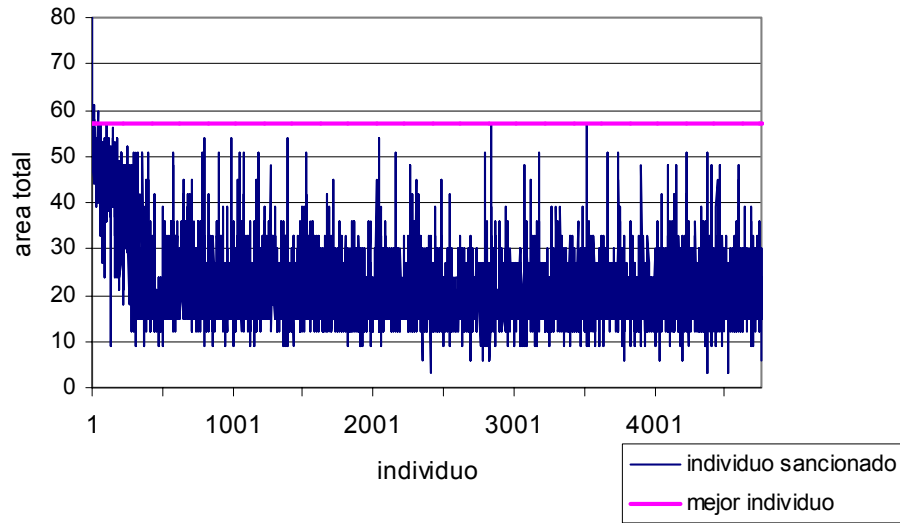


Figura 13: Area de los individuos sancionados y del mejor individuo

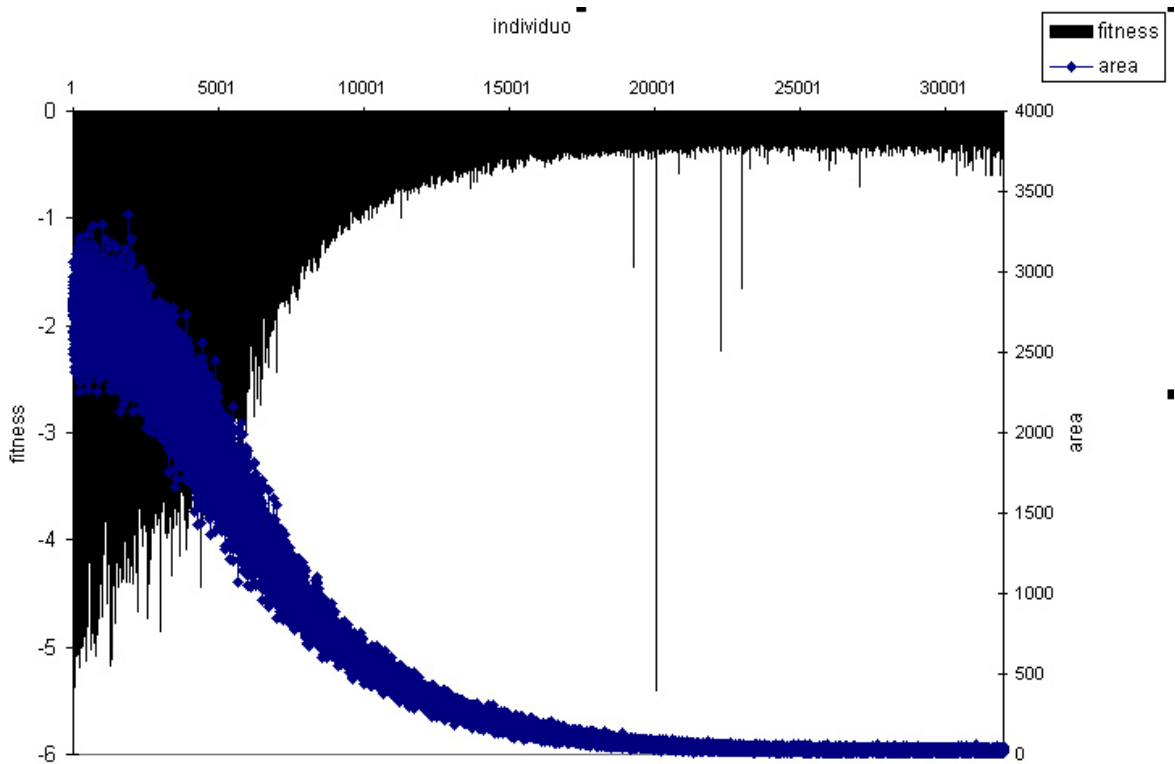


Figura 14: Area vs. fitness, de todos los individuos de la mejor corrida

5.2.3 Comparación

Para tener una idea de que tan buena es la mejor configuración de aletas encontrada por nuestro AG, realizamos una búsqueda en forma manual. Para ello distribuimos en toda el área de búsqueda y en forma equiespaciadas 64, 36, 16, 9, 4, 1 aleta/s de igual longitud. En la Figura 15 se puede observar los resultados de dicha búsqueda.

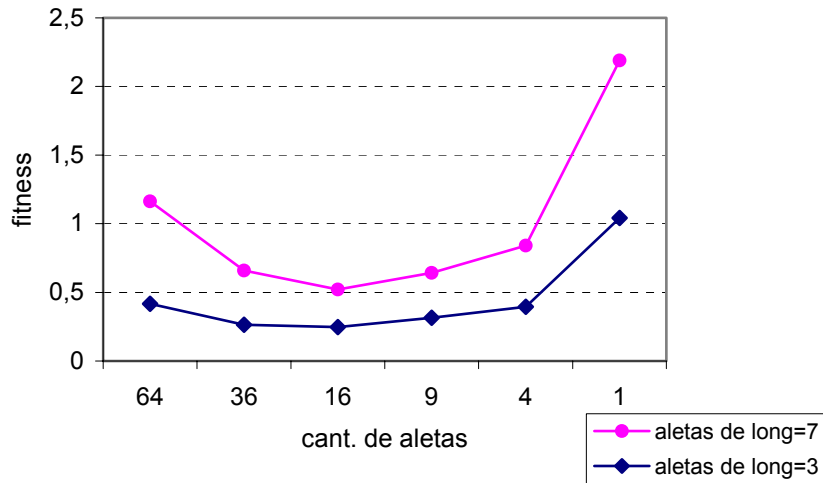


Figura 15: Búsqueda manual de la mejor configuración

Según lo evaluado, la mejor configuración se obtiene con unas 16 aletas equiespaciadas de longitud igual a 3 hy, con lo cual obtenemos un fitness de $-0,246$. Para mejorar este resultado intercalamos las posiciones de las aletas de esta configuración (Figura 16), con lo cual pudimos aumentar el fitness a unos $-0,197$.

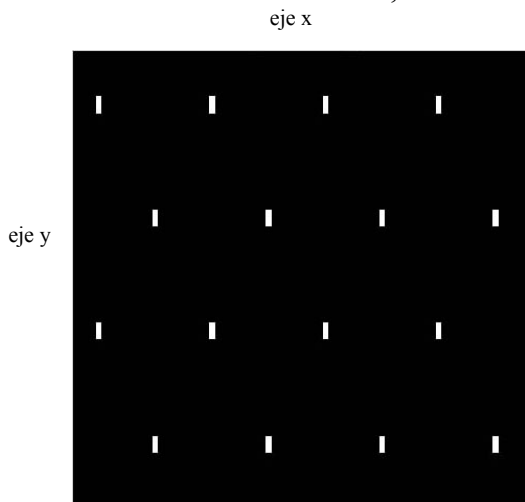


Figura 16: Mejor configuración de la búsqueda manual

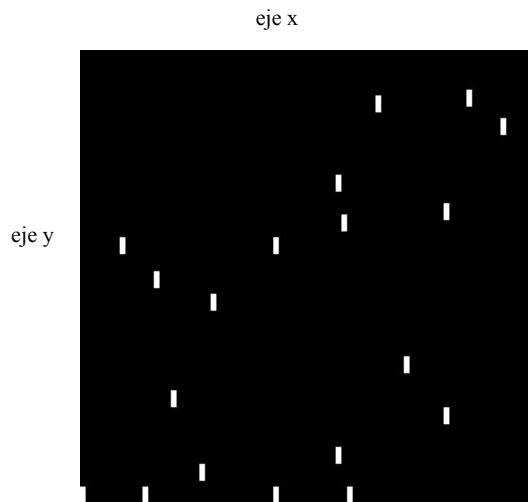


Figura 17: Mejor configuración de la búsqueda por AG

Como podemos apreciar en las Figura 18 y 19, la búsqueda manual no pudo superar al mejor fitness encontrado mediante el AG (corrida de longitud mínima de aleta = 3 hy).

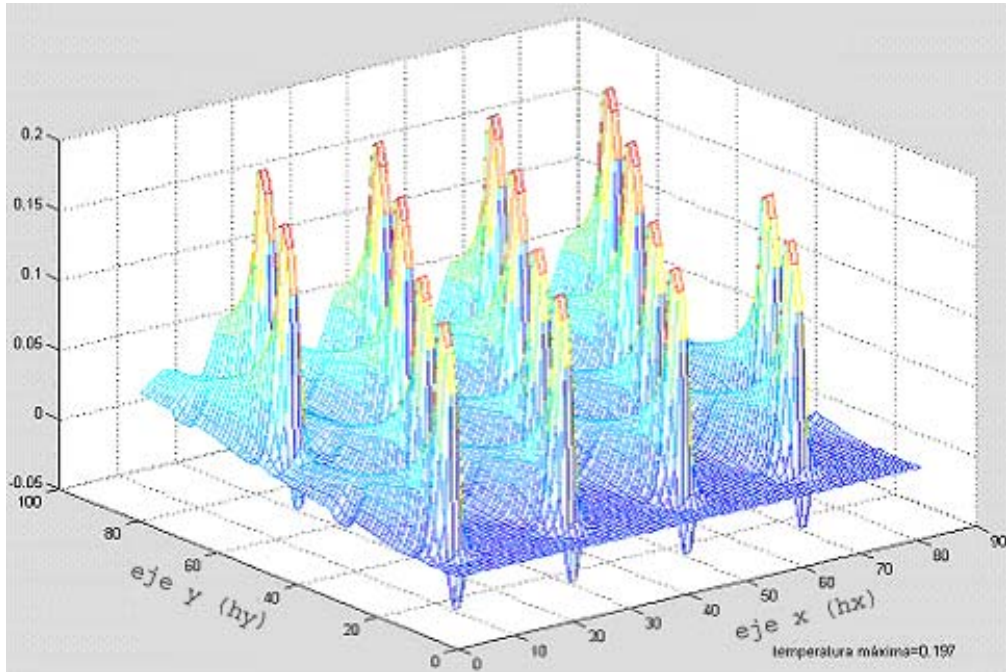


Figura 18: Vista 3D de las temperaturas de la mejor configuración obtenida manualmente

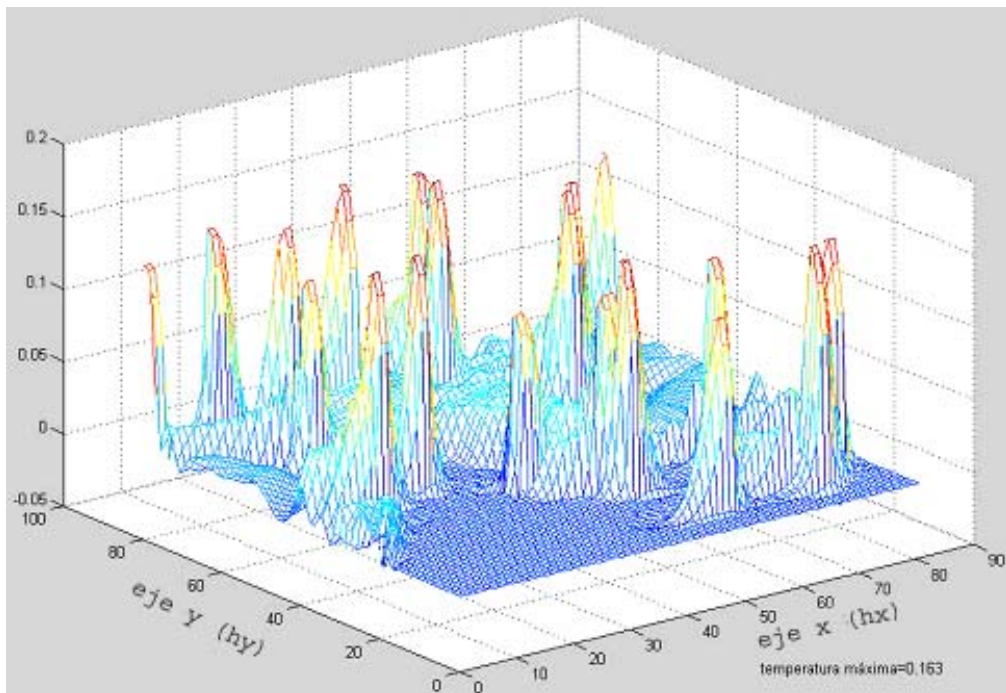


Figura 19: Vista 3D de las temperaturas de la mejor configuración obtenida con el AG

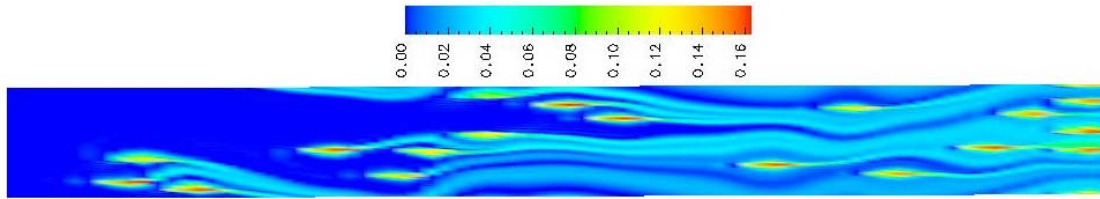


Figura 20: Temperaturas de la mejor configuración obtenida con el AG a escala

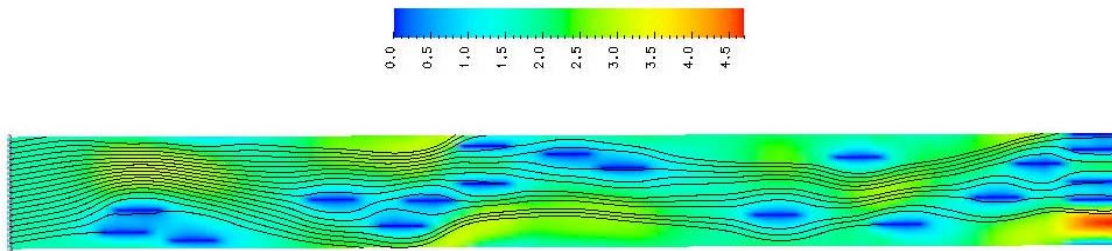


Figura 21: Modulo e isocurvas de velocidad

6 CONCLUSIONES.

El objetivo principal de este trabajo fue el de encontrar una herramienta efectiva para reducir el tiempo de búsqueda de los AGs, especialmente para aquellos que son prácticamente imposibles de implementar en forma secuencial. Tal propósito, a nuestro entender, fue cumplido satisfactoriamente con la paralelización del AG realizada, ya que con el mismo se a logrado una disminución eficiente del tiempo de búsqueda, aproximadamente un 94 % de eficiencia.

También se planteó como objetivo, pero secundario, el de encontrar una solución óptima al problema presentado. Con las distintas corridas realizadas pudimos concluir que las aletas, de la configuración óptima buscada, serán de longitudes pequeñas y que aproximadamente se necesitará 16 aletas de este tipo. Debemos decir que para cumplir íntegramente este objetivo todavía nos falta realizar más corridas, donde no solamente se cambien la longitud mínima de las aletas, sino también, los parámetros típicos de los AGs; también debemos realizar una búsqueda manual más profunda, para saber con más certeza, que tan próxima se encuentra la configuración de aletas encontradas por el AG de la solución óptima.

7 REFERENCIAS

- [1] C. Darwin, “*El Origen de las especies*”, Alianza Editorial S.A., primera edición, (2003).
- [2] *PETSc-FEM*: A general purpose, parallel, multi-physics FEM program. GNU General Public License(GPL), <http://venus.arcrude.edu.ar/petscfem>.
- [3] D.E. Goldberg, “*Genetic Algorithms in Search, Optimization & Machine Learning*”, Addison-Wesley, second edition, (1993).

- [4] W. Gropp, E. Lusk and A. Skjellum, “*Using MPI: Portable Parallel Programming with the Message-Passing Interface*”, MIT Press, second edition, (1999).
- [5] *HPF*: High Performance Fortran, <http://www.vcpc.univie.ac.at/information/mirror/HPFF>