

DISEÑO ACOPLADO DE CELDAS Y RECARGAS DE COMBUSTIBLE NUCLEAR MEDIANTE ALGORITMOS GENÉTICOS

Juan José Ortiz-Servin^{1,*}, David A. Pelta², José Luis Montes-Tadeo¹, Alejandro Castillo¹

1: Departamento Sistemas Nucleares; Instituto Nacional de Investigaciones Nucleares de México.
Carr. México Toluca S/N. La Marquesa Ocoyoacac, Edo México. MEXICO.

juanjose.ortiz@inin.gob.mx, <http://www.inin.mx>
joseluis.montes@inin.gob.mx, <http://www.inin.mx>
alejandro.castillo@inin.gob.mx, <http://www.inin.mx>

2: ETS Ingeniería Informática y Telecomunicaciones, Universidad de Granada,
C/Daniel Saucedo Aranda, s/n 18071, Granada, SPAIN
dpelta@decsai.ugr.es, <http://modo.ugr.es/>

Palabras clave: Reactor Nuclear BWR, Algoritmo Genético, Celda de Combustible Nuclear, Recarga de Combustible Nuclear

Resumen *En este trabajo se presenta una metodología para la solución acoplada del diseño de celdas de combustible nuclear y de recargas de combustible en un reactor de agua en ebullición. Para esto se utilizó un algoritmo genético basado en dos poblaciones, una que codifica la celda de combustible nuclear y otra que codifica la recarga de combustible. Ambas poblaciones evolucionan de forma independiente de acuerdo a sus operadores. Pero la función de fitness de cada población depende del comportamiento de la otra población. En esta investigación se describe a detalle el problema de ingeniería nuclear planteado y cómo fue implementado el algoritmo genético para resolverlo. Los resultados obtenidos muestran un buen comportamiento del núcleo del reactor bajo simulación estática por computadora. Se demuestra que los límites térmicos de operación se cumplen y que se alcanzan los requerimientos de energía.*

1. INTRODUCCIÓN

La operación eficiente y segura de un reactor nuclear de agua en ebullición (BWR por sus siglas en inglés), requiere de la optimización de cuatro problemas complejos íntimamente relacionados entre sí. El primero de ellos es el diseño de las celdas de combustible, que de forma simplificada consiste en colocar pastillas de uranio enriquecido y gadolinia con distintas concentraciones en un tablero de ajedrez de NxN casillas de modo que se cumplan algunos parámetros de seguridad y las expectativas de producción de energía. El segundo problema es el diseño axial del ensamble de combustible, que consiste en apilar celdas de combustible en un arreglo vertical de 25 casillas de modo que se cumpla un perfil axial de potencia propuesto. El tercer problema es el del diseño de la recarga de combustible, que se

puede resumir como otro tablero de ajedrez de tamaño $M \times M$ (que representa al núcleo del reactor) donde se deben colocar ensambles de combustible de modo que se cumplan criterios de seguridad y que se extraiga la energía requerida. El último problema es el de diseño de patrones de barras de control, en el que la altura de inserción de las barras de control desde abajo del núcleo, se debe determinar para garantizar la correcta operación del reactor. Lo anterior se esquematiza en la Figura 1.

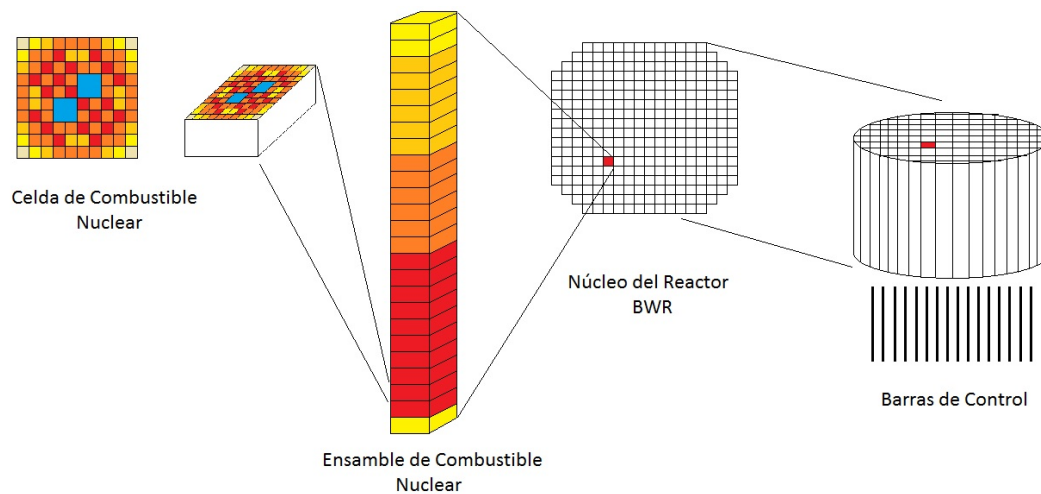


Figura 1. Problemas de Optimización Combinatoria de la Administración de Combustible Nuclear en un BWR

Estas cuatro etapas de diseño nuclear son problemas de optimización combinatoria que están estrechamente ligados entre sí. Esto quiere decir que no se puede lograr una verdadera optimización si ellos se resuelven por separado tomando a los otros como constantes. Esto quiere decir que, por ejemplo, no se puede diseñar la recarga de combustible sin optimizar al mismo tiempo los ensambles de combustible y los patrones de barras de control. Tampoco se pueden optimizar las celdas de forma aislada porque su correcto desempeño depende de una buena optimización de la recarga de combustible y de los patrones de barras de control.

En la literatura especializada en este tema, se pueden encontrar diversos trabajos donde se resuelven estos problemas por separado mediante algoritmos genéticos, colonias de hormigas, redes neuronales, búsqueda tabú, por mencionar algunos. No obstante, existen pocos trabajos que resuelven dos o más de estos problemas en forma acoplada. Por ejemplo, en [1-5] se resolvió el problema del diseño de celdas de combustible nuclear de forma independiente a los demás problemas mencionados. En [6] se resuelve el problema del diseño axial del combustible tomando como base, celdas de combustible ya optimizadas de forma independiente. El problema que más ha sido abordado es el de la recarga de combustible [7-15], en el que por lo general se asumen ensambles de combustible ya optimizados previamente y se usan modelos simplificados para estimar el comportamiento del reactor sin necesidad de diseñar los patrones de barras de control. Finalmente, para el diseño de patrones

de barras de control [16-18] se asume que la recarga de combustible y los ensambles de combustible son variables conocidas y previamente diseñadas. Finalmente, como ejemplos de resolución de los problemas en forma acoplada, se tienen el de diseño de celdas y ensambles de combustible mediante redes neuronales [19]. Como ejemplo de solución acoplada de recargas de combustible y de patrones de barras de control se ha utilizado búsqueda tabú [20]. La restricción principal para resolver el problema de forma acoplada es el inmenso tamaño del espacio de soluciones que tiene el problema y lo caro que resulta la evaluación de la función objetivo involucrada. Por ejemplo, el diseño de celdas de combustible con un arreglo de 10×10 , puede tener un espacio de soluciones de 10^{45} . El diseño del ensamble de combustible, dado un diseño de celdas de combustible, puede ser de 10^{17} . Una vez que se tiene un diseño de ensamble de combustible, en un núcleo de 22×22 , el espacio de soluciones puede ser de $444!$ que se puede reducir a 10^{60} si se aplican algunas reglas heurísticas de acomodo de combustible. Finalmente, los patrones de barras de control, para un núcleo con 109 barras de control y 20 posiciones axiales donde éstas se pueden colocar, considerando que el tiempo de operación del reactor se divide en 10 segmentos; el espacio de soluciones es de $\{(5)^{20}\}^{10}$ (si se aplican algunas simplificaciones). El espacio total de soluciones de los 4 problemas se calcula multiplicando los 4 espacios, es decir en torno a 10^{262} . Además, tampoco hay una metodología establecida para resolver los 4 problemas de forma acoplada. Lo que se hace actualmente es optimizar la celda de combustible usando modelos simplificados para cuantificar el desempeño de la celda. Enseguida, el ensamble de combustible se diseña usando las celdas previamente optimizadas y donde no se analizan muchas combinaciones de celdas en el arreglo vertical de 25 posiciones. Para el diseño de la recarga de combustible se utilizan los ensambles previamente diseñados y se aplican modelos simplificados de quemado del núcleo como el Principio Haling [21] para estimar cuanta energía se extrae del núcleo. Esta estimación del quemado del núcleo, evita tener que optimizar los patrones de barras de control y el resultado se obtiene en cuestión de segundos. Sin embargo, la operación real del reactor, requiere que se establezcan los patrones de barras de control y esto se hace considerando a la recarga de combustible como una variable fija durante la optimización.

Como ya se mencionó anteriormente, otra de las restricciones en el acoplamiento de las 4 etapas de diseño es el costo de la evaluación de la función objetivo. Si se aplica una metodología desacoplada, como se menciona en el párrafo anterior, los tiempos típicos de optimización de CPU son los siguientes: para la celda de combustible en torno a 5100 segundos, el ensamble de combustible es de 9560 segundos, la recarga de combustible se hace en 150 000 segundos. Finalmente, la optimización de los patrones de barras de control puede demorar aproximadamente 450 000 segundos. Es decir que, la propuesta de una solución para los 4 problemas aplicando esta metodología, toma alrededor de una semana de CPU. Si el resultado no es convincente, habrá que hacer cambios y repetir todo el proceso de nuevo. Cabe mencionar, que para estos tiempos de CPU se considera una estación de trabajo a 3.1 GHz y 8 Gb de RAM en plataforma LINUX.

La evaluación de una sola celda de combustible, aplicando un modelo simplificado; demora 1 segundo de CPU, la construcción del ensamble de combustible demora 9560 segundos de CPU, la evaluación de una sola recarga de combustible usando patrones de barras de control previamente optimizados, demora 480 segundos de CPU. Esto quiere decir que la evaluación

del comportamiento de una sola celda de combustible en el núcleo del reactor mediante una simulación por computadora, demora alrededor de 10000 segundos (la suma de los 3 tiempos). Por lo tanto, con una metodología inteligente, se pueden evaluar más celdas de combustible y se pueden tener más soluciones en un tiempo menor.

El objetivo del trabajo realizado, y cuyos resultados más importantes se presentan en este foro, fue diseñar un algoritmo genético para optimizar la celda de combustible y la recarga de combustible de forma acoplada. El diseño del ensamble de combustible se resuelve de forma simplificada considerando la extrapolación de celdas a lo largo de la dirección axial. Los patrones de barras de control se consideran fijos a lo largo del proceso de optimización. Sin embargo, al final del trabajo se dan algunos lineamientos para considerar los dos problemas faltantes. Desde luego, el algoritmo genético se hace más complejo y su ejecución es más lenta.

2. CELDAS Y RECARGAS DE COMBUSTIBLE

En este apartado se describen con más detalle los dos problemas que estarán acoplados en el algoritmo genético. Esta explicación ayudara a entender que representa un cromosoma y que representa cada gen.

2.1. Celda de Combustible Nuclear

Una celda de combustible típica se representa en la Figura 2. Se trata de un arreglo de 10x10 posiciones o localidades. En cada posición, una barra formada por pastillas de uranio enriquecido a las cuales se les puede adicionar o no gadolinia, debe ser colocada. Los posibles valores de uranio enriquecido y gadolinia se muestran en la Tabla 1. Los cuadros azules de la Figura 2 representan canales de agua en los que no se colocan barras de uranio. Los cuadros pequeños representan las posiciones donde se colocan las barras de uranio, los colores denotan distintos enriquecimientos de uranio en las posiciones de la celda. Obsérvese en la Tabla 1, que una concentración de 0% de gadolinia implica que esa barra solo contiene uranio enriquecido. Nótese también, que solo se permiten barras de gadolinia con alto enriquecimiento de uranio.

Durante la optimización de la celda de combustible se busca minimizar el Factor de Pico de Potencia Local (FPPL), es decir, se busca uniformizar la generación de potencia en la celda. La restricción es que el factor infinito de multiplicación de neutrones, denotado por k_{inf} , se mantenga en un rango de valores predeterminado. Estos dos parámetros se calculan mediante la ejecución del código de transporte de neutrones llamado CASMO-4 [22]. La ejecución de este código para determinar los valores de FPPL y k_{inf} demora 1 segundo de CPU.

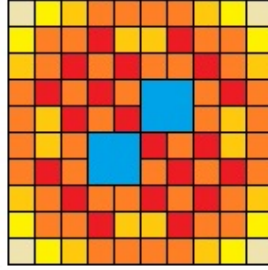


Figura 2. Esquema de una Celda de Combustible Típica

% Gadolinia	% Uranio Enriquecido														
	2.0	2.4	2.8	3.0	3.2	3.4	3.6	3.8	3.95	4.2	4.4	4.6	4.8	4.9	4.95
0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
3							X	X	X	X	X	X	X	X	X
4							X	X	X	X	X	X	X	X	X
5							X	X	X	X	X	X	X	X	X
6							X	X	X	X	X	X	X	X	X

Tabla 1. Uranio enriquecido y concentraciones de gadolinia usadas en celdas típicas de combustible

Una simplificación que suele aplicarse para reducir el espacio de soluciones y el tiempo de cálculo con CASMO-4 es dividir la celda a la mitad. Esto se hace trazando una línea desde la esquina superior izquierda hacia la esquina inferior derecha de la Figura 2.

2.2. Recargas de Combustible Nuclear

El núcleo del reactor típico de 22x22 canales se muestra en la Figura 3. Los cuadros representan a los canales donde se colocan los ensambles de combustible. Las cruces de color rojo representan a las 109 barras de control de este tipo de reactor.

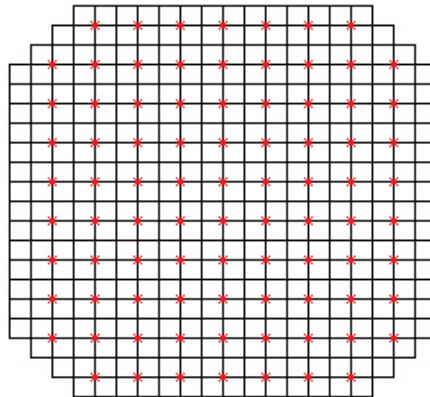


Figura 3. Esquema de un núcleo de reactor típico

Al final de la operación del reactor, después de un periodo de tiempo llamado ciclo de operación, el reactor es apagado. Entonces, los ensambles de combustible más quemados se sacan del núcleo y son reemplazados por los ensambles de combustible previamente optimizados. Combustibles nuevos y quemados son reacomodados en el núcleo para extraer la mayor cantidad de energía. La estimación de la energía que produce el reactor se hace mediante el Principio Haling que asume una distribución axial de potencia lo más uniforme posible de modo que se minimicen los picos de potencia. Como ya se dijo, la alternativa para determinar la energía que se extrae del ciclo de operación, de manera más aproximada, es con el diseño de patrones de barras de control. Ambas alternativas de determinación del final del ciclo requieren la ejecución del simulador del reactor llamado SIMULATE-3 [23]. El parámetro a maximizar es la energía generada a partir del uranio, sujeto al cumplimiento de las fracciones de los límites térmicos, es decir la fracción de Generación Lineal de Calor (FLPD por sus siglas en inglés), la fracción de la generación de calor en un plano (MAPRAT por sus siglas en inglés) y la fracción de la razón de potencia crítica (FLCPR por sus siglas en inglés).

3. IMPLEMENTACIÓN DEL ALGORITMO GENÉTICO

El algoritmo genético que se empleó en este trabajo está formado por dos poblaciones independientes con sus operadores genéticos, pero cuya función de fitness depende de la otra población y es única para las dos poblaciones. Una población codifica celdas de combustible y la otra codifica recargas de combustible. La descripción de la implementación se presenta por separado y al final de la sección se describe como trabajan juntas para resolver el problema de manera global.

3.1. Población de Celdas de Combustible y Operadores Genéticos

Como se dijo antes una celda de combustible puede ser simplificada utilizando la mitad de la misma. De este modo solo se tienen que asignar barras de uranio y gadolinia en 51 posiciones. Por lo tanto, lo primero es visualizar la mitad de la celda como un vector donde su primera posición corresponde con la posición de la esquina superior izquierda de la celda. La segunda y tercer posiciones del vector corresponden con las dos posiciones del segundo renglón de la mitad de la celda. Así sucesivamente, hasta que la última posición del vector corresponde con la posición de la esquina inferior derecha de la celda.

Este vector representa a un cromosoma (a un individuo) de la población de celdas de combustible. A cada gen se le puede asignar una etiqueta de la forma "%U/%Gd" donde %U es el enriquecimiento de uranio y %Gd es la concentración de gadolinia. Los valores de %U y %Gd deben ser escogidos de la Tabla 1. La población de N_C cromosomas de celdas de combustible se debe inicializar con valores válidos.

El operador de cruce que se usó, es el clásico, en el que dados dos padres; se escoge un punto de corte y se mezclan los genes para producir dos hijos. El operador de mutación funciona escogiendo un gen aleatoriamente y alterando su contenido de uranio y gadolinia.

3.2. Población de Recargas de Combustible y Operadores Genéticos

Al igual que la celda de combustible, en el diseño de la recarga de combustible se aplican simplificaciones para reducir el tamaño del problema. Trazando una línea vertical y una horizontal que pasen por el centro del núcleo del reactor, esquematizado en la Figura 3 se obtienen 4 sectores del núcleo con 111 canales cada uno. El siguiente paso es convertir esta representación bidimensional en un vector de tamaño 111. Para esto, el canal que está en el centro del núcleo ocupa la primera posición del vector. Los canales se van posicionando en el vector por fila y columna hasta llegar al 111. Esto representa un cromosoma de la población de recargas de combustible con tamaño N_R . Los genes del cromosoma son etiquetas de la forma CCGdXSec que representan los ensambles de combustible. “CC” es el número de ciclo de operación en que el ensamble entró al reactor, “Gd” es la concentración de gadolinia que tiene el ensamble. “X” representa el sector del núcleo del reactor, X=A es el sector superior izquierdo, X=B es el sector superior derecho, X=C es el sector inferior izquierdo y X=D es el sector inferior derecho. En este trabajo se empleó el sector D. “Sec” es un número secuencial entero.

Para el operador de cruce se aplicó el operador de correspondencia parcial (PMX) propuesto por Goldberg y Lingle [24]. El operador de mutación se aplica escogiendo dos genes del cromosoma e intercambiando sus valores.

3.3. Algoritmo Genético para optimización de celdas y recargas de combustible

Como ya se mencionó, el algoritmo genético trabaja con dos poblaciones de cromosomas. Una para representar la recarga de combustible y la otra para representar la celda de combustible. La población de celdas tiene un tamaño N_C y la población de recargas tiene un tamaño N_R . Dado que se quiere medir el desempeño de cada una de las N_C celdas al ser utilizadas en N_R recargas de combustible, $N_C \times N_R$ simulaciones con SIMULATE-3 son necesarias para evaluar el desempeño de las N_C celdas y las N_R recargas. Una simulación con SIMULATE-3 requiere una celda de combustible, la construcción de su ensamble combustible y una recarga de combustible. En la discusión que sigue se hablará de un par de celda y recarga de combustible para referirse a una simulación de SIMULATE-3.

Al inicio del proceso iterativo, se evalúan las N_C celdas de acuerdo a su desempeño en las N_R recargas de combustible. El valor de fitness de cada celda es el promedio de su desempeño en las N_R recargas. Por otro lado, las N_R recargas son evaluadas con las N_C celdas y su valor de fitness se determina como el promedio de desempeño de cada recarga con cada una de las N_C celdas. De este modo, las mejores celdas y mejores recargas tienen mayor probabilidad de ser escogidas como padres.

Cabe mencionar que para la evaluación inicial de ambas poblaciones K procesadores de una estación de trabajo se utilizan para ejecutar primero las celdas de combustible y armar los ensambles de combustible mediante CASMO-4. Posteriormente las recargas de combustible se simulan con SIMULATE-3. La evaluación completa de ambas poblaciones demora alrededor de 6 horas de CPU.

La construcción de una celda nueva y su correspondiente ensamble combustible es muy costosa en tiempo. Por lo tanto, en cada generación solo N_C' celdas son sustituidas aplicando

los operadores de cruce y mutación. También, solo N_R' recargas de combustible son sustituidas aplicando los operadores de cruce y mutación. Para hacer eficiente el proceso iterativo, se debe cumplir que $N_C' \ll N_C$. De este modo, solo en la primera generación ambas poblaciones completas son simuladas y en las siguientes, solo N_C' son cambiadas y simuladas; mientras que la totalidad de N_R son simuladas. Si $N_C' = 2$ y $N_R' = 4$, se logra reducir el tiempo de ejecución de cada generación a 2 horas de CPU.

En forma coloquial, un símil de este proceso de optimización se puede visualizar como el querer evolucionar al zorro que mejor cace a cualquier conejo. Pero por otro lado, se busca evolucionar un conejo que sea capaz de escapar de cualquier zorro. Cuando se optimiza un problema y se deja constante el otro es equivalente a evolucionar un zorro para que aprenda a cazar solo un tipo de conejo. Cuando se le presenta otro tipo de conejo, su desempeño podría no ser el mejor.

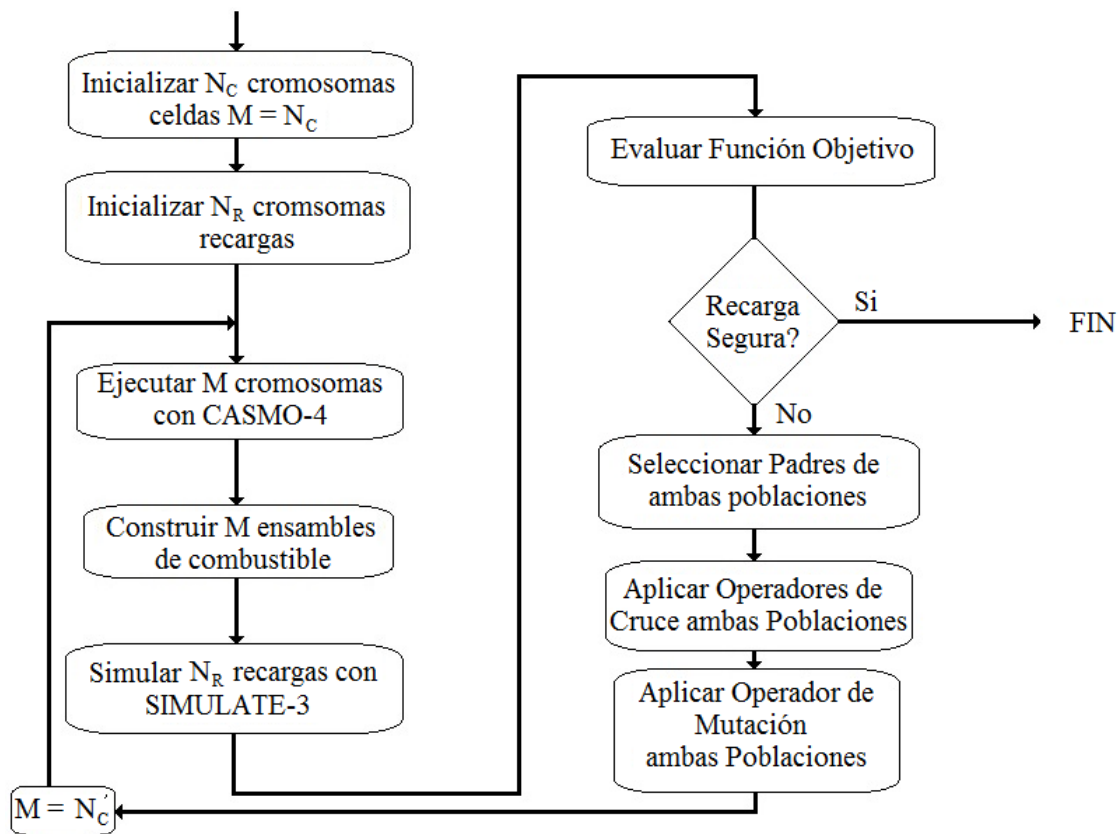


Figura 4. Diagrama de Flujo del Algoritmo Genético

La Figura 4 muestra el diagrama de flujo del algoritmo genético. Al principio se generan las poblaciones de celdas y recargas de combustible. En la primera generación se ejecutan las N_C celdas de combustible con CASMO-4. Se reconstruyen los ensambles de combustible y se

simulan las N_R recargas de combustible con SIMULATE-3. Entonces se evalúa la función objetivo y si se cumplen los tres límites térmicos y se alcanza la energía requerida del ciclo, entonces se detiene el proceso iterativo. De lo contrario se aplican operadores genéticos a ambas poblaciones y se crean N_C' celdas hijo y N_R' recargas hijo. Solo las N_C' celdas se ejecutan con CASMO-4 y todas las N_C recargas se simulan con SIMULATE-3. Este proceso se repite hasta que se cumple la condición de parada mencionada antes.

Dado que las celdas de combustible son las que requieren más tiempo para ejecutarse, se decidió paralelizar las ejecuciones en K procesadores de una estación de trabajo. De este modo, las N_C celdas se pueden ejecutar en bloques de K celdas y reducir el tiempo de ejecución por un factor de N_C/K . En seguida los K procesadores ejecutan las simulaciones de las recargas de combustible de modo que un procesador simula el comportamiento de las N_C celdas en una sola recarga de combustible. Para hacer eficiente el uso de los procesadores N_C/K debería de ser un número entero. De lo contrario podrían estar sin utilizarse varios procesadores y unos pocos en funcionamiento.

3.4. Función Objetivo

Lo que determina si una celda de combustible y si una recarga de combustible son adecuadas para usarse en un reactor nuclear de potencia, es si los tres límites térmicos se cumplen y si se alcanza la energía del ciclo. Como se dijo antes, las posiciones de las barras de control se deben determinar a medida que se va consumiendo el uranio. Al inicio del ciclo, la cantidad de barras de control dentro del reactor es grande, pero conforme avanza el ciclo, las barras se van sacando. En este ciclo de operación, se tienen 35 intervalos de tiempo y en cada uno de ellos se diseñó un patrón de barras de control. Por lo tanto, se tienen valores de límites térmicos de los 35 intervalos de tiempo.

El requerimiento de seguridad implica que las fracciones a los límites térmicos en cada uno de los 35 pasos de tiempo, sea menor a 0.95 para garantizar la adecuada operación del reactor. La función objetivo debe castigar a aquellos pares de celdas y recargas de combustible que no cumplen con los límites térmicos. Por otro lado, también debe premiar a los pares de celdas y recargas de combustible que cumplan con los límites térmicos. Por lo tanto, la función objetivo es la siguiente:

$$Max F = w_1 keff - w_2 \sum_i^{35} \{ (FLPD_i - 0.95) + (MAPRAT_i - 0.95) + (FLCPR_i - 0.95) \} \quad (1)$$

Donde w_1 y w_2 son factores de peso determinados con un análisis estadístico. El subíndice i denota cada uno de los 35 intervalos de tiempo en que se divide el ciclo. La variable $keff$ cuantifica la energía que se extrae del núcleo. Mientras más alto es mayor la energía que se puede obtener del combustible.

4. RESULTADOS

Este algoritmo genético se aplicó a un ciclo de 18 meses con 35 intervalos de tiempo. La celda de combustible debe ser de 4.057% de uranio enriquecido y 40% de gadolinia. El valor de $keff$ requerido es de 0.99511. Las fracciones a los límites térmicos deben ser inferiores a

0.95.

Se utilizaron los siguientes tamaños de población: $N_C = 16$, $N_R = 16$, $N_C' = 2$ y $N_R' = 4$. Estos tamaños de población se escogieron de este modo, porque la estación de trabajo empleada tiene $K=16$ procesadores. De este modo la ejecución de todas las celdas se hace en un ciclo de los 16 procesadores. También la simulación de las recargas de combustible se hace en un solo ciclo de uso de todos los procesadores. En las generaciones donde solo se ejecutan N_C' celdas, el uso de los procesadores se distribuye reduciendo el tiempo de construcción de los ensambles de combustible. Con $N_C' = 2$ se usan 12 procesadores y quedan en espera 4 y todo se hace en un ciclo de los procesadores. Si $N_C' = 3$, se usarían dos ciclos de los procesadores; usando los 16 en el primer ciclo y en el segundo ciclo, solo se usarían dos procesadores, dejando el resto en espera. Esto quiere decir que se usan 6 procesadores para construir un ensamble de combustible.

A continuación se presentan algunos resultados obtenidos con este algoritmo genético. Primero se muestra un resultado típico de una ejecución del algoritmo genético. La Figura 5 muestra el comportamiento de los límites térmicos a lo largo del ciclo de operación que tiene una longitud de 12 GWD/T. Nótese que en todo momento del ciclo, las fracciones a los límites térmicos es inferior a 0.95.

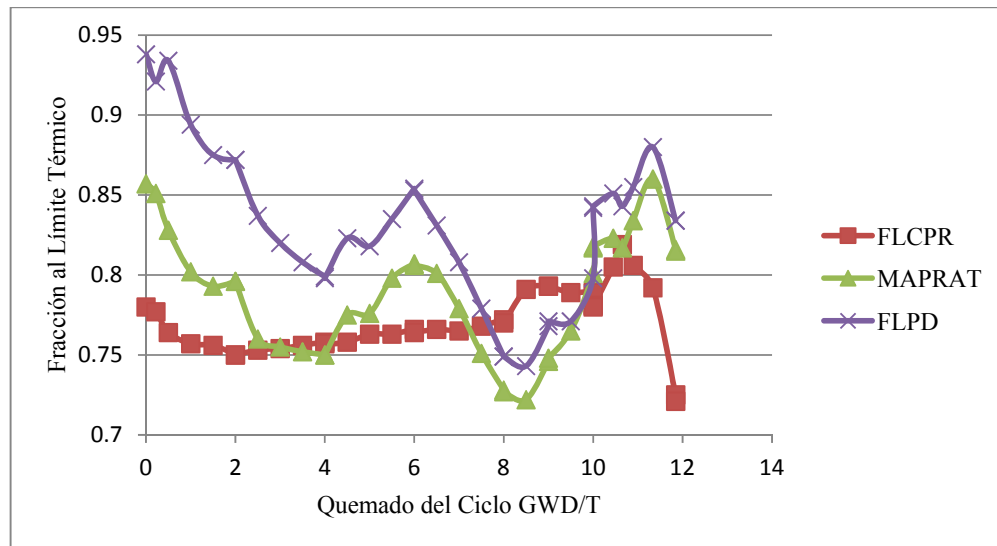


Figura 5. Comportamiento de los Límites Térmicos

Las Figuras 6 y 7 muestran los cromosomas de la celda de combustible y recarga de combustible, respectivamente; obtenidos en esta ejecución. En la Figura 6, las posiciones de la celda indican el enriquecimiento de la barra y las que están marcadas en color tienen gadolinia en los porcentajes indicados. En la Figura 7, el color del canal indica el tipo de ensamble de combustible colocado. En rojo, se muestran los ensambles que tienen 4 ciclos en el reactor; en naranja, se muestran los que tienen 3 ciclos en el reactor; en amarillo, se

muestran los que tienen 1 ciclo en el reactor; y en blanco, los ensambles que son nuevos.

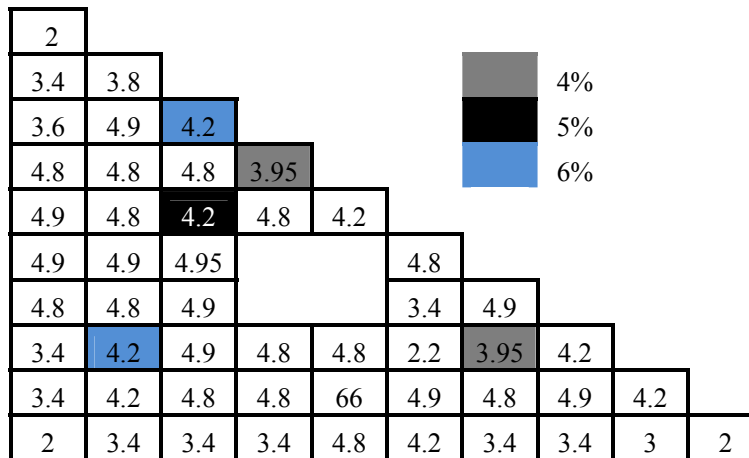


Figura 6. Ejemplo de Cromosoma de Celda de Combustible obtenido

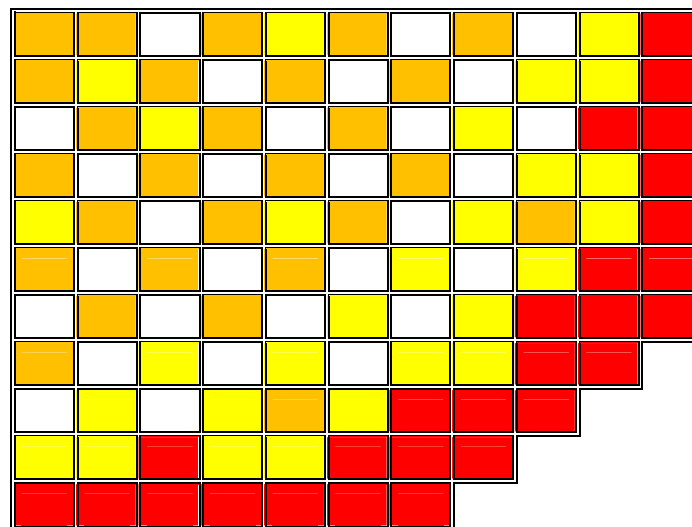


Figura 7. Ejemplo de Cromosoma de Recarga de Combustible obtenido

En particular el ciclo de operación mostrado alcanza una $keff$ de 0.9941. Este valor es inferior al de referencia de 0.99511. Sin embargo, como los patrones de barras de control no fueron diseñados para esta celda y recarga de combustible, aún es posible aumentar el valor de 0.99413 que se obtuvo, diseñando unos patrones de barras de control adecuados.

De la Figura 5, se puede observar que las fracciones de los límites térmicos tienen un valor máximo que no supera el límite de 0.95. Se definen las variables: MFLPD como el máximo FLPD, MMAPRAT como el máximo MAPRAT y MFLCPR como el máximo FLCPR del ciclo. Basta con que estas variables sean menores a 0.95 para garantizar la seguridad del

núcleo durante operación.

En la Tabla 2 se presentan 5 ejecuciones del algoritmo genético indicando los valores de MFLPD, MMAPRAT, MFLCPR y la $keff$ obtenida.

Caso	MFLPD	MMAPRAT	MFLCPR	$keff$
1	0.942	0.868	0.837	0.9954
2	0.938	0.86	0.819	0.9941
3	0.92	0.868	0.851	0.9944
4	0.911	0.86	0.849	0.9944
5	0.946	0.831	0.802	0.9933

Tabla 2. Comportamiento del núcleo del reactor obtenido de los resultados del algoritmo genético

En esta tabla se puede ver que los límites térmicos se cumplen en todos los casos. Sin embargo, la energía de referencia no se alcanza en 4 de los casos presentados. Como ya se dijo, esto se puede resolver diseñando los patrones de barras de control de dichos casos. Nótese también que el caso 1 supera el valor de referencia de $keff$. Esto quiere decir que ese caso, en principio puede generar más energía que el caso de referencia.

5. CONCLUSIONES

- En este trabajo se presentó un algoritmo genético basado en dos poblaciones de cromosomas que representan soluciones a dos problemas acoplados. La primera población son celdas de combustible y la segunda población son recargas de combustible para un reactor nuclear de agua en ebullición. La función de fitness de ambas poblaciones es la misma e involucra parámetros de comportamiento del núcleo del reactor simulado en 3 dimensiones.
- El algoritmo genético se adaptó para trabajar en 16 procesadores en paralelo. De este modo se evalúa primero la población de cromosomas de celdas y posteriormente la población de recargas. El tamaño de ambas poblaciones se escogió de forma tal que la mayoría de los procesadores se utilizan al mismo tiempo.
- Los resultados muestran que ambas poblaciones fueron capaces de evolucionar para tener cromosomas celdas y cromosomas recargas que proporcionan buenas soluciones al problema que se planteó. Los parámetros de seguridad del núcleo se cumplen a lo largo de todo el ciclo y se alcanza la energía requerida del combustible nuclear.
- Los resultados muestran que la metodología propuesta, es adecuada para resolver el problema que se presentó. Es necesario afinar un poco los operadores genéticos, pues en algunos casos obtenidos se violan algunas restricciones de simetría en la recarga de combustible y de colocación de uranio en la celda de combustible.
- Como ya se mencionó, el problema completo requiere resolver 4 problemas de optimización combinatoria. En este trabajo, solo se presentó una metodología para

resolver dos de éstos. Con el propósito de agregar los otros dos problemas, se propone usar 4 poblaciones de cromosomas, una para cada problema. La población de cromosomas de ensambles combustibles puede ser una representación axial del ensamble. Mientras que la población de patrones de barras de control es un poco más compleja pues tiene que representar las posiciones axiales de las barras en cada intervalo en que se divide el ciclo. La función de fitness no cambia pues, lo que define si una solución es buena para todo el problema, es el comportamiento del reactor medido por los límites térmicos y por la *keff*. El tiempo de ejecución tampoco es problema pues no hay que ejecutar simulaciones extra del reactor.

6. AGRADECIMIENTOS

Los autores agradecen al Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Instituto Nacional de Investigaciones Nucleares (ININ), por el apoyo brindado para la realización de la presente investigación a través de los proyectos CB-2011-01-168722 y CA-215, respectivamente. D. Pelta agradece el apoyo del Ministerio de Economía y Competitividad, España, a través del proyecto TIN2011-27696-C02-01; y a la Consejería de Innovación, Ciencia y Empresa, Junta de Andalucía, a través del proyecto P11-TIC-8001 (incluyendo fondos FEDER).

REFERENCIAS

- [1] A. Castillo, J.J Ortiz-Servin, R. Perusquía, S.Y. Campos, "Fuel lattice design with Path Relinking in BWR's", *Progress in Nuclear Energy*. Vol. **53**, pp. 368–374, (2011).
- [2] L. Chaung, L. Tung-Hsien, "Automatic fuel lattice design in a boiling water reactor using a particle swarm optimization algorithm and local search", *Annals of Nuclear Energy* Vol. **47**, pp 98–103, (2012).
- [3] G. Cuevas, T. Parish, G. Curry, "Optimization of MOX enrichment distributions in typical LWR assemblies using a simplex method-based algorithm", *Annals of Nuclear Energy*, Vol. **29**, pp. 2001–2017, (2002).
- [4] J.J. Ortiz, A. Castillo, J.L. Montes, R. Perusquía, J.L. Hernández, "Nuclear fuel lattices optimization using neural networks and a fuzzy logic system", *Nuclear Science and Engineering*, Vol. **162**, pp. 148–157, (2009).
- [5] J.L. Montes, J.J. Ortiz, J.L. François, C. Martín-del-Campo, R. Perusquía, "Fuel lattice design in a boiling water reactor using an ant colony based system". *Annals of Nuclear Energy*, Vol. **38**, pp. 1327–1338, (2011).
- [6] J. J. Ortiz, A. Castillo, J. L. Montes, R. Perusquía. *Diseño axial de combustible para BWRs. SNM-LAS/ANS-AMEE, SMSR*. Cancún, México, (2007).
- [7] J. A. Castillo, G. Alonso, L. B. Morales, E. del Valle, "BWR fuel reloads design using a Tabu search technique", *Annals of Nuclear Energy*, Vol. **31**, pp. 151-161, (2004).
- [8] J.J. Ortiz & I. Requena, "Using a Multi-State Recurrent Neural Network to Optimize Fuel Reloads in BWRs". *Annals of Nuclear Energy*. Vol. **31**, pp. 789-803, (2004).
- [9] J. Esquivel-Estrada, J. J. Ortiz-Servin, J. A. Castillo, R. Perusquía, "Azcaalli: A system based on Ant Colony Optimization algorithms, applied to fuel reloads design in

- a Boiling Water Reactor", *Annals of Nuclear Energy*, Vol. **38**, pp. 103–111, (2011).
- [10] M.A. Alvarenga de Moura, M.M. Dornellas, R. Schirru, "Particle swarm optimization applied to the nuclear reload problem of a pressurized water reactor". *Progress in Nuclear Energy*, Vol. **51**, pp. 319–326, (2009).
 - [11] M.H. Da Silva, R. Schirru, "Optimization of nuclear reactor core fuel reload using the new Quantum PBIL", *Annals of Nuclear Energy*, Vol. **38**, pp. 610–614, (2011).
 - [12] S. Jiang, A.K. Ziver, J.N. Carter, C.C. Pain, A.J.H. Goddard, S. Franklin, H.J. Phillips, "Estimation of distribution algorithms for nuclear reactor fuel management optimisation", *Annals of Nuclear Energy*, Vol. **33**, pp. 1039–1057, (2006).
 - [13] Y.P. Mahlers, "Core loading pattern optimization for pressurized water reactors", *Annals of Nuclear Energy*, Vol. **21**, pp. 223–227, (1994).
 - [14] Y. Tahara, K. Hamamoto, M. Takase, K. Suzuki, "Computer aided system for generating fuel shuffling configurations based on knowledge engineering", *Journal of Nuclear Science and Technology*, Vol. **28**, 399–408, (1991).
 - [15] A. Yamamoto, "A quantitative comparison of loading pattern optimization methods for in-core fuel management of PWR", *Journal of Nuclear Science and Technology*, Vol. **34**, 339–347, (1997).
 - [16] A. Castillo, J. J. Ortiz, G. Alonso, L. B. Morales, E. del Valle, "BWR Control Rod Design Using Tabu Search", *Annals of Nuclear Energy*, Vol. **32**, pp. 741–754, (2005).
 - [17] J. J. Ortiz & I. Requena, "Azcatl-CRP: An Ant Colony-based System for Searching Full Power Control Rod Patterns in BWRs", *Annals of Nuclear Energy*, Vol. **33**, pp. 30–36, (2006).
 - [18] J.L. Montes, J.J. Ortiz, I. Requena, R. Perusquía, "Searching for full power control rod patterns in a boiling water reactor using genetic algorithms", *Annals of Nuclear Energy*, Vol. 31, pp. 1939–1954, (2004).
 - [19] J.J. Ortiz, A. Castillo, D. A. Pelta, "GreeNN: A hybrid method for the coupled optimization of the axial and radial design of BWR fuel assemblies", *Progress in Nuclear Energy*, Vol. **52**, pp. 249–256, (2010).
 - [20] A. Castillo, J. J. Ortiz, J. L. Montes, R. Perusquía, "Fuel loading and control rod patterns optimization in a BWR using tabu search". *Annals of Nuclear Energy*, Vol. 34, pp. 207–212, (2007)
 - [21] R. K. Haling, *Operational Strategy for Maintaining an Optimum Power Distribution through Core Life. ANS Topl. Mtg. Nuclear Performance of Core Power Reactors*, TID-7672. US Atomic Energy Commission, (1964).
 - [22] J. Rhodes, M. Edenius, *CASMO-4. A Fuel Assembly Burnup Program. User's Manual*. SSP-01/400 Rev 4. Studsvik Scandpower, (2004)
 - [23] D. W. Dean, *SIMULATE-3. Advanced Three-Dimensional Two-Group Reactor Analysis Code. User's Manual*. Studsvik Scandpower INC. SSP-95/15 - Rev 3, (2005).
 - [24] D. E. Goldberg, Jr. R. Lingle, *Alleles, loci and the traveling salesman problem. First International Conference on Genetic Algorithms and Their Applications*, Pittsburg, USA, pp. 154–159, (1985).