

RESUMEN DEL CURSO: SOFT COMPUTING APLICADO A LA GESTIÓN DEL COMBUSTIBLE DE UNA CENTRAL NUCLEAR

AUTOR

José Ángel Díaz García

Recuperación de las sesiones 1, 2 y 3 del curso impartido por el Dr Ignacio Requena en el curso de Aplicaciones de Ciencias de Datos en el máster en Ciencia de Datos e Ingeniería de Computadores de la Universidad de Granada.

El ciclo de recargas y actividad del combustible en una central nuclear es probablemente una de las labores más críticas a las que una central se enfrenta y debe modelarse de manera que se permita maximizar beneficios y minimizar tiempo y recursos de la central. Para conseguir estos beneficios, se plantean dos posibles problemas por un lado optimizar los ciclos de carga de combustible y por otro lado optimizar los patrones de barras de control. Ambos problemas pueden modelarse como problema de optimización que pueden ser resueltos por medio de técnicas de soft computing. Sobre el análisis y comparación de estas técnicas en el problema real de la Central Nucleoeléctrica de Laguna Verde (CNLV) es lo que versa este resumen, pero antes de entrar a detallar las mismas es menester comprender el problema de optimización que tenemos entre manos.

1. Problema de la recarga de combustible

El problema de la recarga de combustible puede enunciarse como la posibilidad de cargar N elementos de combustible en N número de canales. En el caso de nuestro problema, tenemos 444 elementos y 444 canales, pero solo se considera $1/4$ del núcleo por simetría rotacional, es decir 111 elementos y canales. Debemos colocar el combustible de manera que se maximice la energía extraída a la vez que se cumplen reglas de seguridad como límites térmicos o colocar el combustible más reciente en la periferia. Para la determinación de la energía producida entra en juego nuestro segundo objetivo de determinación de patrones de barras de control. Este problema es afrontado inicialmente con el criterio de un ingeniero nuclear, pero en algunos casos es necesaria más velocidad y precisión por lo que se modela el problema utilizando técnicas de soft computing. Para ello, tendremos dos criterios el inicio de ciclo **BOC** y el final de ciclo **EOC**, en ambos tendremos que optimizar el K teniendo en cuenta el problema de **RPPF**, o Radial Peaking Factor en el inicio y el problema de los límites de temperatura o **MLHGR** y **MCPR** en el final de ciclo. Estos datos, serán ofrecidos en un dataset de prueba de la central nuclear anteriormente descrita (CNLV), por lo que llegados a este punto estamos en posición de comenzar a describir las soluciones estudiadas.

2. Predicción basada en Redes Neuronales Artificiales

La red neuronal es un perceptron multicapa con 111 entradas con 2 salidas para EOC y 3 para BOC. Se utiliza una sola capa oculta con 10, 20, 30, 40 y 80 neuronas. Sobre esta red neuronal se realizan varios entrenamientos para conseguir una solución aceptable. Estos entrenamientos se realizan utilizando datos de 220 y 300 recargas, sobre los que se dividió 75% para entrenar y 25% para validar, para estimar los valores de salida se usa el software PRESTO (simulador para calcular las ecuaciones diferenciales).

Los resultados alcanzados por esta solución son buenos para predecir MCPR, MLHGR y K , por otro lado no son tan buenos para predecir FPPR, debido a que las recargas malas este valor es muy sensible y las redes neuronales no generalizan correctamente. Los experimentos se comportan de manera similar para una red conjunta que para redes separadas para el ciclo inicial y el final. Por otro lado, las mejores redes han sido las obtenidas por experimentos con 10 y 30

neuronas en la capa oculta. En cuanto a los resultados de predicción de los valores en comparación con el beneficio obtenido podrían fijarse por tanto en 200000\$ por día, tomando de 30 a 40 segundos para predecir las variables interesantes para una recarga en equipos no muy avanzados, otorgando por tanto una fase de ayuda al decisor (ingeniero nuclear) decisiva que se traducirá como hemos visto en ahorro, eficiencia y claro está, seguridad.

3.Optimización basada en algoritmos evolutivos

Los algoritmos genéticos han sido ampliamente utilizados en problemas de optimización combinatoria. Permiten analizar gran cantidad de soluciones de manera efectiva y tienen ciertas habilidades para no caer en mínimos locales. Es por ello que se prestan al problema que tenemos entre manos de manera apropiada, habiendo sido estudiados anteriormente en la literatura, donde encontramos referencias como el sistema **FUELGEN, CIGARO, ALGER o SOPRAG**. En el estudio del que trata este resumen se propone por tanto un sistema **RECOPIA**, en el que la función objetivo es penalizada si no cumple las reglas básicas que vimos en la sección anterior de redes neuronales y que busca maximizar el K en BOC y EOC. Este es un algoritmo genético que divide cromosomas de orden en 3 trozos o reglas básicas sobre los cuales se aplican operadores de cruce y mutación. Para el proceso de selección es un básico sistema de ruleta. La solución busca por tanto ubicar en la periferia las unidades más quemadas y en el centro las más frescas y que por tanto tienen más radioactividad, los operadores de cruce y mutación serían por tanto:

-**Cruce**: Se seleccionan dos puntos de corte por ruleta y se emparejan por probabilidades. Los puntos duplicados se reinsertan por intercambio entre los hijos.

-**Mutación**: Se aplica una probabilidad de mutación muy pequeña, se intercambia un gen por otro y se conserva la simetría mutando al principio los asimétricos y posteriormente los simétricos.

Los experimentos realizados se basan en 50 recargas con EC frescos, se evalúan con la red neuronal entrenada y se van generando nuevas generaciones de "individuos" con los operadores de cruce, mutación y transición vistos si en 15 nuevas generaciones la población se mantiene, tendremos un óptimo por lo que es una solución a tener en cuenta. Los resultados nos ofrecen que en todas las ejecuciones los resultados están en zonas seguras y además se suele mejorar la longitud del ciclo de recarga. En comparación con las otras soluciones de la literatura RECOPIA cumple las reglas básicas, además, el uso de la red neuronal para predecir variables que vimos anteriormente favorece el proceso lo que se traduce en ahorro.

4.Optimización basada en Red Neuronal Recurrente Multiestado

Dado que el problema se asemeja en cierta medida al viajante (TSP) se han buscado otras técnicas de optimización adecuadas con éxito a este problema. Concretamente se ha estudiado una red neuronal recurrente multiestado. Este problema fue abordado por Sadighi et al. usando una red Hopfield aunque se quedan en los albores lo que se podría haber conseguido, garantizando solamente que no se viola el FPPR, por lo que se ha propuesto una red multiestado frente a la biestado del autor. En esta nueva propuesta, cada estado de una neurona toma valores del conjunto $M=\{1...N\}$ y tiene asociada una función umbral que indica su posición a presentar la salida, además cada estado de la red en cada instante tiene una energía asociada por lo que la función mirará en nuncio de ciertas condiciones de similitud (coincidencia o simetría) la analogía entre las salidas. La red se basa en una dinámica de aumento de potencial en el que las neuronas evolucionan a espacios de estados que más disminuyen la FE es decir incrementan el potencial de la misma.

Esta red se usa para el proceso de optimización asignando a las 111 neuronas 111 posibles estados. Se definen las FE (función de similitud ya que la teoría de perturbaciones se desecha) y los estados, para ello se definen la red periférica la subred central y una matriz de simetría sin su diagonal. Tras esto se eligen neuronas y se va intercalando sus estados y los de sus simétricas si se incrementa el potencial es factible sino se rechaza.

5.Experimentación final de optimización de cargas de fuel

Los experimentos finales se basaron en 5 ejecuciones independientes usando los mismos parámetros para entrenar la red neuronal y posteriormente para el proceso de optimización los siguientes ejemplos:

Algoritmo Evolutivo: 50 cromosomas en la población con probabilidad de cruce de 0.3, probabilidad de mutación de 0.05 y ratio de transición de 0.05.

Red neuronal Recurrente Multiestado: Se entrenó con dos funciones de energía, teoría de perturbación y K.FE.

Los resultados se compraran con el sistema SOPRAG y se puede constatar que el basado en la red multiestado es bastante mejor en casi todos los casos excepto en el ciclo de recarga 5 donde SOPRAG tiene claras mejoras debido a que usan datos de la piscina de fuel usado o quemada. Por tanto también se dispusieron estos algoritmos para este supuesto obteniendo mejores resultados que los algoritmos de a literatura en todos los casos destacando RECOPIA-M para los ciclos 2, 3 y 4 y RENOR para los 5 y 6.

6.Optimización de la colocación de las barras de control

Tras la predicción de los valores y posteriormente basándonos en estos valores optimizar el proceso de recarga de combustible nuclear basado en algoritmos genéticos y redes neuronales multiestado, se basa en la optimización de la colocación de las barras de control. Como vimos anteriormente estas barras se usan para determinar la energía producida y suelen estar en 45 posiciones diferentes aunque para simplificar el estudio se ha realizado con 5 debido a la simetría de 1/8 del reactor. Cada una de estas barras puede estar en 25 posiciones diferentes (someras y profundas) y hay 10 cambios por ciclo. Se ha modelado la solución en base a algoritmos genéticos y algoritmos basados en colonia de hormigas.

-Genéticos: Población inicial de 25 cromosomas de longitud 5, se seleccionan los padres mediante ruleta elitista y un operador de cruce del 70% con un solo punto de corte. Como probabilidad de mutación un 10% y dos mecanismos uno basado en aleatoriedad evitando la zona de posición y otro entre intercambio de posición de barras someras y profundas. Por ultimo solo el 20% de los cromosomas reciben copia del PBC para favorecer la competencia, lo que se traducirá en soluciones mejores.

-Colonia de Hormigas: En esta solución se modela cada barra como un nodo y las hormigas saltan de uno a otro siempre en el mismo orden secuencial en cada nodo la hormiga decide el estado en función de un parámetro feromona t y la conveniencia asociada cada uno de los 19 posibles estados, esta conveniencia vendrá dada en que una posición viole un limite termino o no y se basa, al igual que la función de la feromona en complejas formulas matemáticas.

Los experimentos finales se llevan a cabo con 25 cromosomas en el algoritmos genético y 25 hormigas en el algoritmo de colonias, obteniendo resultados bastante similares. Como conclusiones finales, podríamos por tanto concluir como los métodos de son computing han aumentado los ciclos mejorando en producción y ahorros de unos 400000 millones de € (2M por día). Estas técnicas están actualmente en uso el ININ(México).

En cuanto a vías futuras para cada unos los problemas se abren nuevas ideas como nuevas formas de función objetivo u otras formas de las funciones de energía.