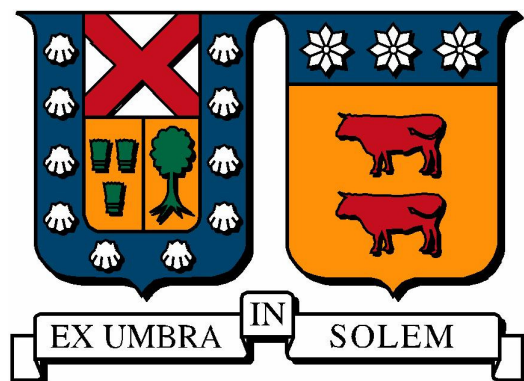


Universidad Técnica Federico Santa María
Departamento de Informática
Valparaíso - Chile



EL PROBLEMA DE PLANIFICACIÓN DE LA GENERACIÓN ELÉCTRICA RESUELTO CON OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

RENATO RIVERA MOHANA

renato@amucan.com

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

PROFESOR GUÍA: CARLOS CASTRO

PROFESOR CORREFERENTE: MARÍA CRISTINA RIFF

Marzo - 2017

Resumen

En el siguiente trabajo se presenta un enfoque para solucionar el problema de planificación eléctrica a corto plazo usando la inteligencia de enjambres, específicamente la técnica de optimización por enjambre de partículas o Particle Swarm Optimization, PSO por sus siglas en inglés, basada en el comportamiento social de las bandadas o cardúmenes, el cual es un algoritmo de optimización estocástica de poblaciones que comparte ciertas características con la computación evolutiva. La planificación eléctrica es un problema de optimización combinatorial con satisfacción de restricciones el cual consta de dos optimizaciones separadas; el Unit Commitment, y el Economic Dispatch, centrándose el trabajo en la primera de estas, describiendo las condiciones de operación, restricciones técnicas y de mercado que el conjunto de generadores eléctricos debe cumplir. Se describe una metodología de trabajo y herramientas para mejorar la performance del algoritmo.

Palabras Claves: Planificación eléctrica, PSO, bio-inspiradas, Unit Commitment, Economic dispatch.

Índice general

1. Introducción	1
2. Optimización por enjambre de partículas	3
2.1. Optimización por enjambres de partículas	3
2.2. Mejorando la optimización por enjambres	5
2.2.1. Sintonización de parámetros	5
2.2.2. Control de parámetros	6
3. Planificación de Generación Eléctrica	8
3.1. Planificación de generación eléctrica	8
3.1.1. Modelo Matemático	9
3.1.2. Estado del arte	11
4. Planteo de la solución	15
4.1. Representación	15
4.2. Movimientos	16
4.2.1. Alargar bloque de funcionamiento o descanso	16
4.2.2. Acortar bloque de funcionamiento o descanso	16
4.2.3. Agregar un bloque	17
4.2.4. Remover un bloque	17
4.2.5. Intercambio de bloques	17

ÍNDICE GENERAL

4.3. Evaluación	20
4.3.1. Costo de encendido	20
4.3.2. Costo combustible	20
4.3.3. Costo de falla	20
4.4. Parámetros	21
4.5. Estructura del algoritmo	21
5. Resultados	22
5.1. Sistema de prueba	22
5.2. Análisis de resultados	23
5.2.1. Convergencia	23
5.2.2. Comparación	25
6. Conclusiones	26
6.1. Resumen y Conclusiones	26
6.2. Trabajo Futuro	27

Índice de figuras

4.1. Representación binaria de las partículas	15
4.2. Ejemplo de alargar bloque	16
4.3. Ejemplo de acortar bloque	16
4.4. Ejemplo de agregar un bloque	17
4.5. Ejemplo de remover un bloque	17
4.6. Ejemplo de intercambio de bloques	18
4.7. Ejemplo reconstrucción de partícula	19
4.8. Ejemplo reconstrucción de partícula, paso 2	19
4.9. Ejemplo reconstrucción de partícula, paso 3	20
5.1. Convergencia PSO	23
5.2. Convergencia PSO con control de parámetros	24

Indice de tablas

4.1. Estado previo al intercambio de bloques	18
5.1. Sistema de prueba Kazarlis96	22
5.2. Carga por hora, sistema Kazarlis96	23
5.3. Comparativa de resultados	25

Capítulo 1

Introducción

Nuevos enfoques a problemas tradicionalmente difíciles de solucionar mediante métodos convencionales debido a su alta complejidad, se dan gracias a la computación bioinspirada, la cual se basa en sistemas biológicos o de comportamientos naturales. Sistemas bioinspirados como redes neuronales, computación evolutiva o la inteligencia de enjambres se han transformado en populares técnicas para la resolución de problemas con satisfacción de restricciones, adaptándose a un sin número de éstos. La inteligencia de enjambres se inspira en el comportamiento social de las colonias de hormigas o cardúmenes, alineamiento de aves en vuelo y crecimiento bacteriano. Estos comportamientos colectivos de simples individuos interactuando el uno con el otro, logran crear modelos de enjambres que son utilizados como optimizadores.

Estas técnicas bioinspiradas ayudan en la resolución de problemas como la planificación eléctrica; un desafío interesante, debido a que en un sistema eléctrico típico existe una gran variedad de unidades disponibles para la generación de electricidad, y cada una tiene diferentes características. En el caso de una unidad de energía nuclear puede proporcionar electricidad a un muy bajo costo adicional por cada megavatio-hora adicional de energía, sin embargo, éste tiene un alto costo de puesta en marcha cada vez que es cerrado; y se necesita un tiempo para que vuelva a su potencia plena. Por otro lado, un generador de turbina de gas puede ser puesto en marcha en pocos minutos, aunque su coste incremental por megavatio-hora sea mucho más costoso. Minimizar estos costos de operación, así como proveer los niveles adecuados de generación para mantener los niveles de disponibilidad, supone que el sistema cuente con herramientas que permitan lograr una planificación eficiente de los recursos de generación.

CAPÍTULO 1. INTRODUCCIÓN

Dentro de la planificación de generación eléctrica existe una división dependiendo del horizonte de tiempo a planificar, largo, mediano y corto plazo. La planificación a largo plazo consiste en períodos de aproximadamente 1 a 5 años, en donde se planifica la cantidad de energía a producir mensual o trimestralmente. Planificación de mediano plazo comprende períodos de meses o semanas, basándose en resultados de la planificación de largo plazo. Los períodos que comprende la planificación de corto plazo son diarios o semanales, dividiendo estos en horas, logrando los planes de funcionamiento de cada generador del sistema. Los principales problemas de ésta se dan debido a la dificultad de determinación en la predicción del consumo energético por parte de la población, con esto se hace más importante el tener soluciones en un tiempo corto y de buena calidad.

Los objetivos de la memoria son los siguientes:

- Objetivo general:
 - Establecer una metodología para resolver el problema de planificación de generación eléctrica a corto plazo.
- Objetivos específicos:
 - Resolver mediante la optimización por enjambres de partículas el problema de planificación de generación eléctrica a corto plazo
 - Aplicar herramientas que permitan mejorar el desempeño de la técnica.
 - Generar y comparar resultados con otros algoritmos y distintas configuraciones de generadores eléctricos.

El presente informe está organizado de la siguiente manera; en la sección 2 se describe la optimización por enjambre de partículas, explicando los principales componentes del algoritmo. Luego, en la sección 3 se introduce el problema de planificación de generación eléctrica, junto con el modelo matemático. Finalmente, en la sección 4 se plantea la solución; representación, movimientos, parámetros y estructura.

Capítulo 2

Optimización por enjambre de partículas

2.1. Optimización por enjambres de partículas

Optimización por enjambre de partículas o conocido como PSO, por sus siglas en inglés (Particle Swarm Optimazation), es una técnica de optimización estocástica basada en poblaciones desarrollada por Eberhart y Kennedy en [1], inspirada en comportamientos sociales de aves y peces. PSO comparte muchas similitudes con los algoritmos evolutivos. El sistema es inicializado con una población con soluciones aleatorias y busca soluciones óptimas a través de generaciones. Sin embargo, a diferencia de los algoritmos genéticos, PSO no contiene operadores de evolución, ni cruzamiento, ni mutaciones. En PSO las potenciales soluciones son llamadas partículas, que se mueven en el espacio de búsqueda siguiendo a las partículas óptimas actuales.

PSO está inspirado en sistemas biológicos, específicamente en los sistemas biológicos sociales, estos comportamientos en donde los individuos interactúan con el ambiente y entre ellos mismos. Algunos le llaman inteligencia de enjambres. Los intentos originales se hicieron para simular gráficamente las coreografías de aves y peces, sin embargo, se encontró que los enjambres se pueden ocupar para la optimización de problemas.

Para entender el funcionamiento se presenta el siguiente ejemplo; suponga el siguiente escena-

CAPÍTULO 2. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

rio: un grupo de aves aleatorias buscando comida en cierta área. Existe sólo un trozo de comida en el área de búsqueda, las aves no saben dónde está la comida, pero saben qué tan lejos está en cada iteración. Entonces la mejor estrategia para encontrar la comida, es seguir al ave que está más cerca de ésta. PSO aprende del escenario y lo utiliza para resolver el problema. Cada solución es un ave en el espacio de búsqueda. Todas las partículas tienen un valor de fitness, que es evaluado en la función a optimizar, también tienen velocidad que repercute en el vuelo de la partícula. En cada iteración la partícula es actualizada por dos valores óptimos. El primero es la mejor solución llevada hasta ahora por la partícula (fitness), este valor es llamado pbest. El otro valor óptimo llevado por el enjambre, es el mejor valor obtenido por todas las partículas en la población, este valor es llamado gbest. Cuando la partícula forma parte de una topología de vecinos, el mejor valor es un óptimo local y es llamado lbest.

Después de encontrar los dos valores, la partícula actualiza su velocidad y sus posiciones según las diferentes ecuaciones

$$\begin{aligned}v_i(t+1) &= w * v_i(t) + c_1 * r_1 * (lb(t) - x_i(t)) + c_2 * r_2 * (gb(t) - x_i(t)) \\x_i(t+1) &= x_i(t) + v(t+1)\end{aligned}$$

donde:

- $v_i(t)$ velocidad generador i en un tiempo t
- $x_i(t)$ posición generador i en un tiempo t
- r_1 y r_2 son números aleatorios entre 0 y 1.
- c_1 y c_2 son constantes de aprendizaje, llamadas constantes sociales y cognitivas. La información cognitiva, c_1 , tiene relación con la mejor posición local de la partícula, en cambio la información social tiene que ver con la mejor posición global del enjambre.

El peso de inercia es un intento por mejorar la estabilidad en la búsqueda ocupando un peso w que hace que la velocidad en el próximo paso dependa de la velocidad actual, este método es introducido por Shi y Heberhart en [2], quedando la ecuación de la siguiente forma:

$$\begin{aligned}x_1(t+1) &= x_i(t) + v_i(t+1) \\v_i(t+1) &= wv_i(t) + R(p_i - x_i(t)) + R(g_i - x_i(t))\end{aligned}$$

CAPÍTULO 2. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

Aquí un w grande hará relativamente más difícil cambiar la dirección del movimiento de la partícula, como resultado el enjambre estará por más tiempo en un dominio D . Esto naturalmente es deseado en una fase inicial de la búsqueda donde una mayor parte del dominio D es desconocido. Existen muchas variaciones sobre PSO, para realizar mejores y más rápidas convergencias, con algunas mejoras al algoritmo o haciendo algoritmos híbridos. Van Den Bergh y Engelbrecht [3] proponen un PSO cooperativo llamado PSO split, donde cada componente de la solución es un vector que es separado en PSO de menores dimensiones. Sin embargo, el poder sintonizar los problemas individualmente con la cantidad de parámetros manejados es poco eficiente. Entonces, los mismos autores propusieron un PSO híbrido, que reduce en gran medida el tamaño del enjambre, y necesita menor tiempo de computación para la mantención de convergencia. Morten Lovbjerg y otros [4] combinaron la velocidad tradicional y las reglas de actualización de posición con ideas de reproducción de subpoblaciones, haciendo que PSO tenga el potencial de alcanzar convergencia más rápido y poder encontrar una mejor solución.

2.2. Mejorando la optimización por enjambres

PSO presenta una cantidad considerable de parámetros los que influyen en la búsqueda, haciendo que el algoritmo sea sensible a estos, además, presenta problemas de estancamiento en óptimos locales o convergencia prematura. Afortunadamente existen herramientas como la sintonización o el control de parámetros, que resuelven en gran medida estos problemas. A continuación, se describen estas herramientas, para poder conformar una metodología en torno a PSO, además de presentar una forma de mejorar el desempeño o *performance* del algoritmo mediante computación de alto rendimiento.

2.2.1. Sintonización de parámetros

En la resolución de problemas mediante técnicas evolutivas o de enjambres, con la meta de obtener las mejores soluciones se diseñan algoritmos con componentes que dependen de distintos parámetros que afectan directamente el desempeño del algoritmo. Estos parámetros pueden ser del tipo numérico, que describe un espacio de valores continuos o del tipo de componentes, que describen un conjunto de valores discretos. Dentro de los componentes parametrizables en un algoritmo de enjambres se encuentran la representación de individuos, función de evaluación, coeficientes sociales y cognitivos, tamaño de población, número de iteraciones. La sintonización de parámetros permite encontrar buenos valores antes de la ejecución del algoritmo. Existen diferentes formas

CAPÍTULO 2. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

de sintonizar parámetros; de forma manual, usando la experiencia del diseñador y el desempeño observado, por analogía, usando información de otros estudios, basados en diseño de experimentos, basados en búsqueda e híbridos. A continuación, se describen los principales algoritmos de sintonización:

- F-RACE, basado en diseño de experimentos, propuesto por Biratti en [5], pertenece a los métodos de fuerza bruta que estiman el desempeño del algoritmo en base al número suficiente de ejecuciones en un *set* suficiente de instancias, esto quiere decir que asigna los recursos o explota las mejores soluciones candidatas, fomentando la competencia y eliminando las de bajo desempeño usando la evidencia estadística.

F-RACE requiere la definición de un set de valores posibles para los parámetros, es decir, las posibles combinaciones y el número de corridas iniciales para evaluación de los candidatos. A través de test de hipótesis se rechazan o mantienen configuraciones hasta que se cumpla el criterio de término.

- REVAC, basado en búsqueda, propuesto por Eiben y Nannen [6]. Definido como un algoritmo de estimación de distribución, requiere la definición de un rango de valores posibles para los parámetros. Se inicializa una población usando una distribución uniforme para los valores dados de parámetros y se mide la relevancia de cada parámetro a través del cálculo de entropía. Por cada iteración se crea un sólo hijo, ocupando los operadores genéticos de cruzamiento y mutación. Los parámetros propios de REVAC son el tamaño de población, la cantidad de padres, el intervalo de mutación y la cantidad de iteraciones.

- ParamILS, basado también en búsqueda, fue propuesto por Hutter en [7], está inspirado en la sintonización manual y requiere la definición de un *set* de valores para los parámetros, comenzando con una configuración inicial de estos parámetros. Realiza la búsqueda inicial de forma aleatoria y por cada iteración busca la mejor solución del vecindario. Los parámetros propios del algoritmo son el número de soluciones aleatorias iniciales, la cantidad de soluciones vecinas aleatorias por cada iteración, probabilidad de reinicio y el máximo número de corridas del algoritmo.

2.2.2. Control de parámetros

En casos donde la sintonización de parámetros es demasiado costosa para el tiempo requerido para la resolución del problema, el control de parámetros es una buena opción. Este proceso de búsqueda de buenos valores se desarrolla en conjunto con el proceso de búsqueda del problema y

CAPÍTULO 2. OPTIMIZACIÓN POR ENJAMBRE DE PARTÍCULAS

permite variar los valores de los parámetros durante dicho proceso tomando en consideración el estado actual de la búsqueda. Además, el control de parámetros sirve para diferentes problemas y toma en consideración los cambios en la búsqueda, como los óptimos obtenidos durante la ejecución. El control requiere de un diseño, identificar los parámetros a controlar, interpretar y estudiar el comportamiento para definir gatilladores de cambio y a qué velocidad se realizará éste.

Existen diferentes tipos de control de parámetros:

- **Determinista:** Como el nombre lo dice, los cambios se hacen de forma determinista con respecto a algún tipo de parámetro de *tiempo* del algoritmo. Esto quiere decir que siempre el parámetro variará de la misma forma, un ejemplo simple es disminuir cada n iteraciones el valor del parámetro x en un 10%.
- **Adaptivo:** Los cambios se realizan en base a la información de la búsqueda, se monitorean ciertos elementos del algoritmo de forma de determinar cambios en los valores de los parámetros en base a cambios en los elementos. Los valores de los parámetros son globales, es decir, para todo el algoritmo. Un ejemplo para este tipo de control en PSO, sería cuando el óptimo global sea el mismo por un determinado número de iteraciones, el factor social aumente y así fomente la exploración.
- **Auto-adaptivo:** Este tipo de control funciona de la misma manera que el anterior, basado en la información de la búsqueda para realizar los cambios. La diferencia entre ellos es que en el caso de auto-adaptivo los valores de parámetros son propios de cada solución o individuo. El ejemplo sería parecido al anterior, sólo que los cambios se realizan para cada solución del problema, lo que nos permite controlar con mayor profundidad, sin embargo, perdiendo performance al agrandar el espacio de búsqueda de soluciones.

Capítulo 3

Planificación de Generación Eléctrica

3.1. Planificación de generación eléctrica

La planificación de la generación eléctrica a corto plazo consiste en determinar, para un horizonte de tiempo establecido, las condiciones de operación de cada generador del sistema, es decir, cuáles generadores deberán operar y qué nivel de energía deberán producir para cada hora del período de planificación.

Cada generador involucrado en la producción de energía necesaria posee características particulares. Se caracterizan por el nivel mínimo y máximo de potencia a generar, así como distintas curvas de consumo de combustible. Otro punto importante que los caracteriza son los parámetros de tiempo de marcha y tiempo de descanso; un generador no puede operar durante un tiempo inferior al tiempo de marcha mínimo, así como tampoco puede estar apagado durante un tiempo por debajo al mínimo de descanso. Además del costo del combustible asociado a la generación misma de energía, cada vez que un generador se enciende gasta una cantidad adicional de combustible. Esta cantidad varía dependiendo del tiempo que el generador ha estado apagado. Para tiempos muy largos, es necesario calentarlo antes de ponerlo en funcionamiento, cuanto más tiempo haya estado apagado, mayor será el gasto de combustible para ponerlo en marcha.

La planificación de la generación de corto plazo es un problema de optimización combinatorial mixto no lineal extremadamente complejo. Se divide en dos subproblemas: el Predespacho de Unidades, o Unit Commitment, y el Despacho Económico de Carga, o Economic Dispatch. El Unit

CAPÍTULO 3. PLANIFICACIÓN DE GENERACIÓN ELÉCTRICA

Commitment Problem (UCP) es un problema de optimización combinatorial que consiste en determinar que generadores estarán encendidos (o despachados en jerga eléctrica) para cada hora del período de planificación, de manera de minimizar los costos de combustible, encendido y apagado de las unidades, respetando las restricciones de operación. El Economic Dispatch Problem (EDP) es un problema de optimización que busca asignar los niveles de potencia que suministrará cada unidad en cada hora del período de planificación, de manera de minimizar el costo de satisfacer la demanda energética, respetando las restricciones de operación.

3.1.1. Modelo Matemático

Se presentará el modelo de optimización propuesto por Kasarlis en [8] Parámetros:

- N : cantidad de generadores a planificar.
- T : número de periodos en el horizonte de planificación.
- $Pmin_i$: potencia mínima generada por la unidad i .
- $Pmax_i$: potencia máxima generada por la unidad i .
- A_i, B_i, C_i : constantes de cada generador para calcular su costo de combustible.
- $Tamin_i$: tiempo mínimo consecutivo que debe estar apagado el generador i .
- $Temin_i$: tiempo mínimo consecutivo que debe estar encendido el generador i .
- $Tcold_i$: tiempo que demora el generador i en enfriarse.
- Sc_i : costo de encender el generador i en frio.
- Sh_i : costo de encender el generador i en caliente.
- E_i estado inicial del generador i , número de horas que ha estado encendido (si es positivo) o apagado (si es negativo).
- D_t : demanda energética para el tiempo t .
- R_t : reserva en gira para el tiempo t .

Variables:

$$■ x_{it} = \begin{cases} 1 & \text{si la unidad } i \text{ está encendida en el tiempo } t \\ 0 & \text{si no} \end{cases}$$

CAPÍTULO 3. PLANIFICACIÓN DE GENERACIÓN ELÉCTRICA

- $P_{i,t}$: potencia generada por unidad i en el tiempo t , si está encendida.
- $x_{it} = \begin{cases} 1 & \text{si la unidad } i \text{ ha estado apagada por un tiempo superior a } Tcold_i \\ 0 & \text{si no} \end{cases}$
- $te_{i,t}$: tiempo consecutivo que lleva encendido la unidad i en el tiempo t .
- $ta_{i,t}$: tiempo consecutivo que lleva apagada la unidad i en el tiempo t .

Función Objetivo:

$$\min : \sum_{i=1}^N \sum_{t=1}^T x_{i,t} CC(P_{i,t}) + CE_{i,t}$$

En donde $CC(P_{i,t})$ es el costo del combustible y $CE_{i,t}$ es el costo del encendido.

El costo de combustible $CC(P_{i,t})$ corresponde a la cantidad de combustible que consume la unidad i para producir la potencia $P_{i,t}$. Este costo es cuadrático y está dado por la expresión:

$$CC(P_{i,t}) = A_i P_{i,t}^2 + B_i P_{i,t} + C_i$$

$CE_{i,t}$ costo de encendido:

$$CE_{i,t} = x_{i,t}(1 - x_{i,t-1})(y_{i,t}Sc_i + (1 - y_{i,t})Sh_i)$$

Restricciones:

- Condición de cobertura de demanda eléctrica:

$$\sum_{i=1}^N x_{i,t} P_{i,t} = D_t$$

- Las unidades encendidas deben tener una holgura de generación energética:

$$\sum_{i=1}^N x_{i,t} (Pmax_i - P_{i,t}) \geq R_t$$

- La siguiente restricción limita las potencias generadas por las unidades:

$$Pmin_i < P_{i,t} < Pmax_i$$

CAPÍTULO 3. PLANIFICACIÓN DE GENERACIÓN ELÉCTRICA

3.1.2. Estado del arte

En esta sección se presenta una recopilación de los principales trabajos realizados en la planificación eléctrica. En primer lugar, se presentarán técnicas de resolución como heurísticas, sistemas expertos y relajación lagrangiana, para luego centrarse en computación evolutiva y de enjambres, particularmente en PSO y las distintas representaciones de soluciones y movimientos utilizados en la actualización de posición y velocidad del algoritmo.

3.1.2.1. Heurísticas, sistemas expertos y relajación lagrangiana

En la búsqueda de buenas asignaciones de generadores se han implementado diferentes heurística, una de las más comunes corresponde a las listas de prioridad, donde las unidades se rankean dependiendo de su rendimiento. Esta técnica se basa en la observación que en las buenas planificaciones los generadores de mayor rendimiento están más tiempo encendidos.

En [8] se utilizan listas de prioridad para el problema de planificación de generadores termales, aplicándose al sistema Británico CEGB (Central Electric Generating Board) con 74 unidades y un horizonte de 24 horas. Para mejorar el rendimiento de estas heurísticas se mezclan con sistemas expertos que basan la resolución en casos anteriores similares. Esto es presentado en [9], dividiendo el proceso en tres etapas; la primera busca un caso anterior semejante dentro de la base de datos, luego en la etapa dos se ajustan los niveles de generación para satisfacer las restricciones de demanda, y, finalmente, en la tercera etapa se realizan los reemplazos de las unidades con peor rendimiento para un sistema de 26 generadores y un horizonte de 24 horas.

Otra técnica bastante común en los primeros trabajos es la Relajación Lagrangiana, este consiste en relajar algunas restricciones de un problema de minimización, incluyéndola en la función objetivo y multiplicándolas por términos denominados multiplicadores de Lagrange. Esto produce un segundo problema, cuya solución entrega una cota inferior para el problema original, obteniendo la solución final modificando la solución arrojada por el segundo problema mediante las denominadas heurísticas lagrangianas. Al relajar las restricciones que involucran varios generadores, se obtienen varios sub-problemas, uno por cada generador.

Trabajos como en [10] analizan métodos de solución tanto para el primer problema como para los sub-problemas, agrupando generadores y reduciendo en un 30 % el espacio de búsqueda. Para la inicialización de variables de decisión se usa un método basado en listas de prioridad para sistemas

CAPÍTULO 3. PLANIFICACIÓN DE GENERACIÓN ELÉCTRICA

entre 20 y 25 generadores.

Las técnicas presentadas requieren costos de cómputo muy bajos para obtener soluciones, sin embargo, los resultados generalmente están muy por debajo de los óptimos, y para la capacidad de cómputo de los sistemas actuales, tales como la computación de alto rendimiento, dejan obsoletas las ventajas que ofrecen estos métodos. En el caso particular de la relajación lagrangiana, si bien simplifica el problema, al haber un mayor número de generadores la efectividad decrece perdiendo calidad en las soluciones.

3.1.2.2. Computación Evolutiva

Dentro de la computación evolutiva, los algoritmos genéticos han sido ampliamente utilizados, con variadas representaciones continuas y binarias para diferentes horizontes de tiempo, además se hibridizan las técnicas genéticas con Tabú Search o Simulated Annealing para poder mejorar la calidad de las soluciones.

En [11] se utiliza una representación binaria y se realiza una planificación con un horizonte de una hora, repitiéndose para lograr las 24 horas de planificación requeridas. Para los individuos que violen restricciones se les penaliza en la función objetivo. Este trabajo pone el énfasis en los operadores genéticos, además de incluir un cruzamiento multipunto y una mutación aleatoria se proponen cuatro operadores adicionales, dos de ellos exclusivos de los mejores individuos de cada generación, implementados con el método Hill Climbing. El problema resultó con una población de 100 individuos que evolucionan durante 100 generaciones para sistemas de 13 generadores y un horizonte de 24 horas.

En [12] los autores proponen una representación de matriz de números reales para representar el nivel de carga para cada generador, el sistema consta de 11, 19 y 40 unidades con un horizonte de 24 horas. Se resuelve el sistema con 200 individuos en 400.000 generaciones.

En [13] se ocupa un algoritmo genético híbrido mediante el uso de Tabú Search, que se utiliza para la optimización local de los individuos y Simulated Annealing implementado como una prueba para que los individuos sean aceptados para la generación siguiente. La representación consiste en un string binario, que, en vez de penalizar las soluciones, ocupa un sistema de reparación. La población es de 50 con 1.000 generaciones para sistemas de 10 y 24 unidades con un horizonte de

24 horas.

Las técnicas evolutivas permiten adaptarse a diferentes tipos de problemas, ya sean estos continuos o discretos, tambien permiten una gran escalabilidad de instancias de prueba. A diferencia de las primeras técnicas, los algoritmos evolutivos requieren un mayor poder de cómputo, por esto las representaciones u operadores que puedan reducir el espacio de búsqueda mejoraran considerablemente el desempeño del algoritmo. Además, la habilidad de poder crear técnicas híbridas para mejorar aún más el rendimiento es una ventaja a considerar.

3.1.2.3. Particle Swarm Optimization

Se han seleccionado los mejores trabajos en PSO para el problema de planificación eléctrica, centrándose en la representación de las soluciones como partículas y las mejoras al algoritmo en su rendimiento. En [14] se presenta un PSO híbrido, que mezcla representaciones continuas y binarias. El PSO basado en valores reales optimiza la energía generada, mientras que el PSO binario optimiza el estado del generador, si debe estar prendido o apagado. Esto da la ventaja de poder optimizar en paralelo, mejorando el rendimiento en cuanto a tiempo de búsqueda. Otro punto a considerar en este trabajo es la satisfacción de restricciones, en la cual se penalizan algunas y otras se reparan para evitar soluciones no factibles.

En [15] se introduce el Improved Particle Swarm Optimization que es una extensión del PSO normal, que ocupa más informacion de las partículas para el control de la mutación de estas, haciendo un símil con el comportamiento social en donde algunos individuos lideres toman mejores decisiones. Además, se presenta una estrategia adaptiva para la selección de parámetros que asegura la convergencia del algoritmo.

CAPÍTULO 3. PLANIFICACIÓN DE GENERACIÓN ELÉCTRICA

En [16] se presenta una representación binaria para la partícula de NxT en donde 1 representa si el generador está prendido y 0 si está apagado. Se utilizan mejoras a las restricciones de demanda y giros de reservas mediante listas de prioridad. También se utiliza peso de inercia para manejar la convergencia prematura de PSO, para luego comparar los resultados con los entregados por la relajación lagrangiana, mostrando un rendimiento superior.

La literatura respecto a PSO, es diversa, pero se centra en los puntos mencionados anteriormente, representaciones binarias y continuas, diferentes herramientas para mejorar el rendimiento de PSO y las modificaciones a restricciones mediante listas de prioridad o relajación lagrangiana, buscando siempre mejorar el desempeño del algoritmo y soluciones de mejor calidad.

Capítulo 4

Planteo de la solución

En este capítulo se presenta el enfoque utilizado para abordar el problema de planificación eléctrica a corto plazo. Para esto se detalla la representación, los movimientos de las partículas, método de evaluación, parámetros y estructura general del algoritmo.

4.1. Representación

La representación típica utilizada para resolver problemas de planificación es binaria, 1 para indicar que el generador i está prendido en una hora t y 0 para indicar que está apagado, como se ve en la figura 4.1. Debido a las restricciones de tiempos mínimos de encendido de los generadores, la representación tendrá muchas combinaciones infactibles. En [18] se propone una codificación en bloques de funcionamiento, lo que disminuye considerablemente el espacio de búsqueda. Para reparar este problema de codificación, los movimientos del algoritmo se encargaran de actualizar las partículas en bloques de funcionamiento, identificando el principio y el final de un bloque.

	0	1	2	3	4	5	6	7	8	9	10	11
Unidad 1	0	1	1	1	0	0	0	0	0	0	1	0
Unidad 2	1	1	1	0	0	0	0	0	1	1	1	1

Figura 4.1: Planificación de dos generadores en un período de 12 horas.

4.2. Movimientos

Los movimientos están encargados de transportar a las partículas e imitar la posición de los mejores individuos del enjambre. Para cambiar las posiciones de las partículas se consideran bloques de funcionamiento y de descanso que viene dado por las restricciones técnicas de cada generador. Las horas entre bloques, que al cambiar harían infactibles las partículas, no se consideran en el movimiento, esto disminuye considerablemente el espacio de búsqueda de la representación binaria. Cada movimiento posee un modo inteligente que revisa que el cambio mejore el fitness de la partícula o ayude a satisfacer la demanda eléctrica.

4.2.1. Alargar bloque de funcionamiento o descanso

Se agregan horas de funcionamiento o descanso dependiendo del bloque en que se encuentre. En el modo normal se elige aleatoriamente entre los largos posibles, en cambio en el modo inteligente el largo se modifica hasta que satisfaga la demanda. En la figura 4.2 tenemos una unidad que funciona de la hora (1,3) y se puede alargar para las horas (4, 5). Para alargar un bloque de descanso tenemos la unidad 2 se puede alargar en la hora 8.

	0	1	2	3	4	5	6	7	8	9	10	11
Unidad 1	0	1	1	1	0	0	0	0	0	0	1	0
Unidad 2	1	1	1	0	0	0	0	0	1	1	1	1

Figura 4.2: Alargar bloque de funcionamiento para la unidad 1 y el de descanso para la unidad 2.

4.2.2. Acortar bloque de funcionamiento o descanso

Se eliminan horas de funcionamiento o descanso al bloque dependiendo del bloque en que se encuentre. En el modo inteligente se acortan horas mientras estas no incurran en demanda.

	0	1	2	3	4	5	6	7	8	9	10	11
Unidad 1	0	1	1	1	1	1	0	0	0	0	1	0
Unidad 2	1	1	1	0	0	0	0	0	1	1	1	1

Figura 4.3: Se acortan los bloques de funcionamiento para la unidad 1 y descanso para la 2.

CAPÍTULO 4. PLANTEO DE LA SOLUCIÓN

4.2.3. Agregar un bloque

Se agrega un bloque si existen las horas necesarias para los mínimos de funcionamiento y descanso. El modo inteligente alarga el bloque mientras se necesite cubrir demanda.

	0	1	2	3	4	5	6	7	8	9	10	11
Unidad 1	0	1	1	1	1	1	0	0	0	0	1	0
Unidad 2	1	1	1	0	0	0	0	0	0	0	0	0

Figura 4.4: Se agrega un bloque si se cumplen las horas mínimas de funcionamiento y descanso.

4.2.4. Remover un bloque

Se remueve un bloque de la planificación. En el modo inteligente sólo se remueve el bloque si mejora el fitness de la partícula.

	0	1	2	3	4	5	6	7	8	9	10	11
Unidad 1	0	1	1	1	1	1	0	0	0	0	1	0
Unidad 2	1	1	1	0	0	0	0	0	0	0	0	0

Figura 4.5: En el modo inteligente sólo se remueve el bloque si el fitness mejora.

4.2.5. Intercambio de bloques

Los movimientos anteriores hacen a las partículas incurrir en altos niveles de insatisfacción de demanda. El intercambio de bloques es el movimiento más importante, ya que toma en cuenta la eficiencia de cada generador. Si las planificaciones son construidas sin considerar la eficiencia, en relación al costo versus potencia generada, estas tendrán fitness elevados debido a que, o presentan altas horas de insatisfacción de demanda, por lo tanto penalizada, o bien cumple, pero con los generadores más ineficientes.

En [18] se plantea una medida de utilización y rendimiento denominado UR, la cual está definida por:

$$UR_t = \text{horas_utilizadas} * \text{Costo}_{max} / \text{Potencia}_{max}$$

El valor de UR para la unidad i será mayor a medida que su razón costo/potencia sea mayor y que esté encendida una cantidad de horas mayor. Si la unidad tiene una medida de UR alto,

CAPÍTULO 4. PLANTEO DE LA SOLUCIÓN

Tabla 4.1: Estado previo al intercambio de bloques

Unidad	UR	cost/potencia	Temin	Tamin
0	45.0	9.0	3	3
1	63	10.5	1	1
2	75.0	15.0	5	5
3	108.0	18.0	6	6
4	180.0	22.5	8	8

indicará que es una unidad ineficiente que está encendida muchas horas; y si el UR es bajo, indicará que se trata de una unidad de buen rendimiento que está encendida pocas horas.

Debido que el set de generadores utilizados en las pruebas las unidades más eficientes son las que generan la mayor cantidad de energía, se elimina el doble de unidades con UR altos que en unidades con UR bajo.

Luego de eliminar los generadores con el UR más bajo y dos con los más altos, se reconstruye la partícula con los generadores más eficientes. Se agregan los bloques con las horas mínimas y si existe demanda insatisfecha se agrega otra hora de funcionamiento.

Para ejemplificar este movimiento, supóngase la configuración de generadores de la figura 4.6. En la tabla 4.1 está la relación costo/potencia y los valores mínimos de funcionamiento y descanso.

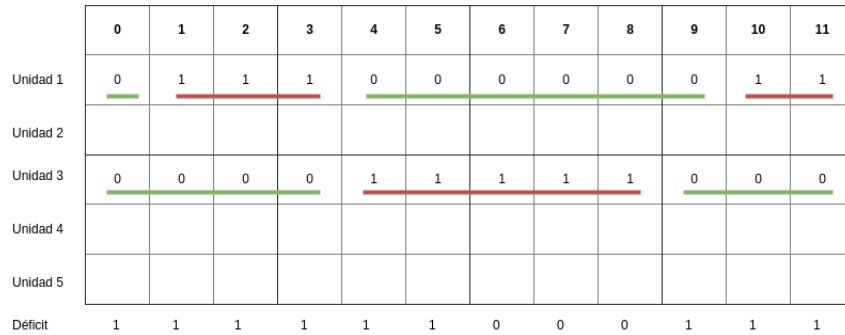


Figura 4.6: Se remueven las planificaciones de las unidades a intercambiar.

CAPÍTULO 4. PLANTEO DE LA SOLUCIÓN

Se empieza a completar la planificación de la unidad 2, que es la que menor UR posee. Se crean los bloques 0,5 y 9, 11 como muestra la figura 4.7. Luego se sigue con la unidad 4 desde la hora 2 que presenta insatisfacción. Se crea el bloque 2,7. Para terminar el intercambio, se sigue con la unidad 5 creando el bloque 9,11, ya que la hora 9 presenta insatisfacción de demanda.

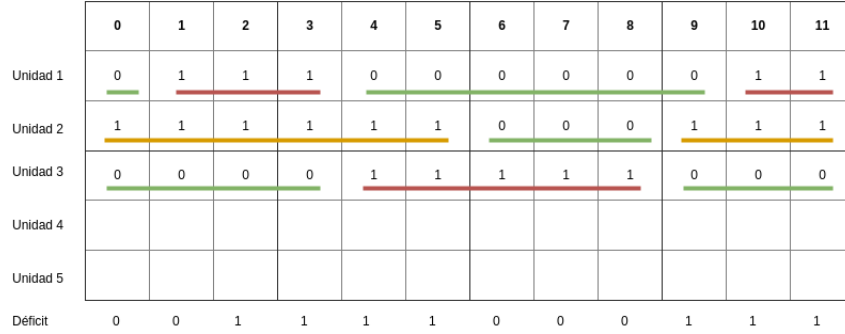


Figura 4.7: Se empieza por el generador de mejor UR.

Finalmente, la partícula queda como en la figura 4.8, en este caso, sin incurrir en insatisfacción de demanda eléctrica.

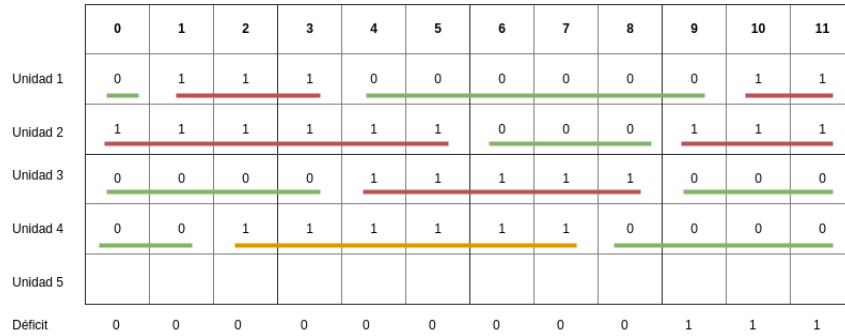


Figura 4.8: Luego con el primer generador de alto UR.

CAPÍTULO 4. PLANTEO DE LA SOLUCIÓN

Al último se agrega el generador con más alto UR, como se ve en la figura 4.9.

	0	1	2	3	4	5	6	7	8	9	10	11
Unidad 1	0	1	1	1	0	0	0	0	0	0	1	1
Unidad 2	1	1	1	1	1	1	0	0	0	1	1	1
Unidad 3	0	0	0	0	1	1	1	1	1	0	0	0
Unidad 4	0	0	1	1	1	1	1	1	0	0	0	0
Unidad 5	0	0	0	0	0	0	0	0	0	1	1	1
Déficit	0	0	0	0	0	0	0	0	0	0	0	0

Figura 4.9: Finalmente el generador de más alto UR.

4.3. Evaluación

La función objetivo es minimizar el costo de la planificación. La evaluación se asocia a una medida de fitness que representa el éxito del enjambre en conseguir su objetivo. Esta medida debe ser positiva, mayor cuando se esté cerca y menor cuando se esté lejos. Para determinar el costo de una planificación, se calculan tres costos asociados, el costo de encendido, el costo de combustible y el costo de falla o penalización.

4.3.1. Costo de encendido

Es el costo de una unidad al encender y depende de la cantidad de horas que ha estado apagado antes de ser encendida. El costo de encender en caliente, es decir, con pocas horas de apagado es menor al costo de encendido en frío.

4.3.2. Costo combustible

El costo de combustible se genera mientras la unidad está encendida y depende de la potencia de cada generador.

4.3.3. Costo de falla

Éste es el costo de fallar en la demanda, corresponde a una penalización por MW no suministrado. La penalización es más alta que el costo de producir un MW en el generador más caro.

4.4. Parámetros

- Número de generadores: Cantidad de unidades a planificar.
- Ventana de planificación: Cantidad de horas a planificar.
- Factor de inercia Factor de inercia del algoritmo.
- Número de iteraciones: Cantidad de iteraciones que correrá el algoritmo.
- Tamaño del enjambre: Cantidad de partículas del enjambre.
- Factor social: Factor que determina qué tanto imitará la partícula al enjambre.
- Factor cognitivo: Factor que determina qué tanto influyen los propios resultados de la partícula.
- Tolerancia EDP: El parámetro *lambda* indica qué tan cerca del valor de demanda debe estar el de oferta para considerarse satisfecha.
- Umbral EDP: Parámetro que determina a cuántas partículas se le aplicará evaluación de EDP.

4.5. Estructura del algoritmo

```
inicializar partículas
evaluación inicial de partículas
Para un número de iteraciones hacer:
  Para tamaño del enjambre hacer:
    actualizar velocidad
    mover partícula
    evaluar partícula
```

Capítulo 5

Resultados

En este capítulo se presenta el sistema de pruebas y los resultados obtenidos para validar la implementación. Para esto se utiliza un sistema escalable de unidades termoelectricas ocupado por varios autores para comparar sus implementaciones.

5.1. Sistema de prueba

Para poder testear algoritmos existen pocos sistemas de prueba estándares en los cuales se pueden hacer comparaciones confiables. Uno de los más utilizados es el planteado por Kazarlis en [11]. Es un sistema de 10 unidades termoelectricas escalable a 100. La reserva de giro se fija en 10% de la demanda de cada hora. La planificación es de 24 horas. Las características y demandas aparecen en las tablas 5.1 y 5.2, respectivamente.

Pmin	Pmax	A	B	C	Temin	Tamin	Sh	Sc	Tcold	E
150	455	0.00048	16.19	1000	8	8	4500	9000	5	8
150	455	0.00031	17.26	970	8	8	5000	10000	5	8
20	130	0.000200	16.60	700	5	5	550	1100	4	5
20	130	0.000211	16.50	680	5	5	560	1120	4	5
25	162	0.000398	19.70	450	6	6	900	1800	4	6
20	80	0.000712	22.26	370	3	3	170	340	2	3
25	85	0.00079	27.74	480	3	3	260	520	2	3
10	55	0.000413	25.92	660	1	1	30	60	0	1
10	55	0.000222	27.27	665	1	1	30	60	0	1
10	55	0.000173	27.79	670	1	1	30	60	0	1

Tabla 5.1: Sistema de prueba Kazarlis96

CAPÍTULO 5. RESULTADOS

El programa fue escrito en C++ y compilado con gcc(c++11) en Ubuntu 14.04. El computador de prueba posee un procesador Intel core i7 y 6 GB en RAM.

hora	1	2	3	4	5	6	7	8
carga	700	750	850	950	1000	1100	1150	1200
hora	9	10	11	12	13	14	15	16
carga	1300	1400	1450	1500	1400	1300	1200	1050
hora	17	18	19	20	21	22	23	24
carga	1000	1100	1200	1400	1300	1100	900	800

Tabla 5.2: Carga por hora, sistema Kazarlis96

5.2. Análisis de resultados

5.2.1. Convergencia

Se puede observar en la figura 5.1 que la búsqueda converge a un óptimo local prematuramente, quedando la mayor parte del tiempo atrapado en este mínimo. Esto se debe a que el control de parámetros implementado no logra la diversidad requerida para mover al enjambre a otra zona de exploración.

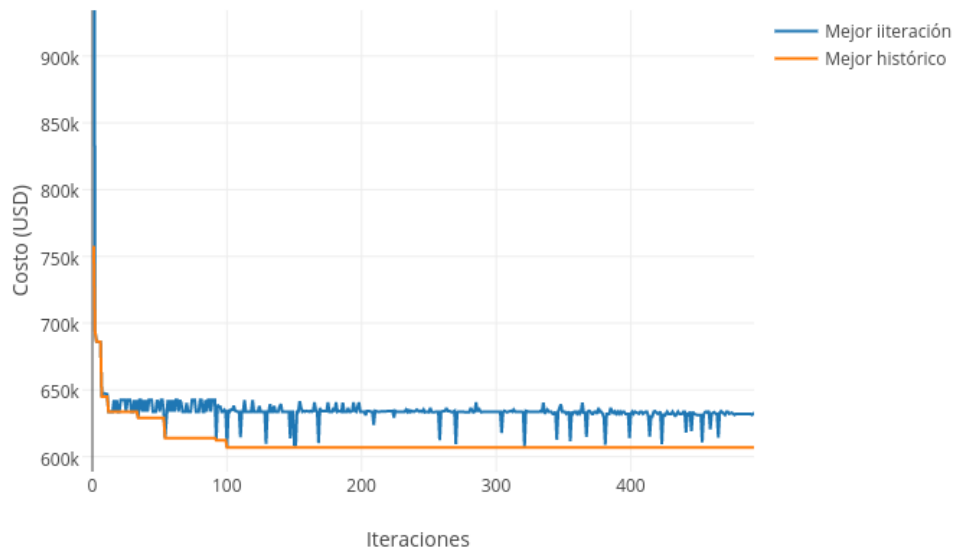


Figura 5.1: Convergencia PSO.

CAPÍTULO 5. RESULTADOS

En la figura 5.2 se aplicó un control que reemplaza partículas por nuevas cada 20 iteraciones. Se puede observar que las mejores partículas por iteración bajan su calidad, pero los movimientos se encargan de converger nuevamente al óptimo local.

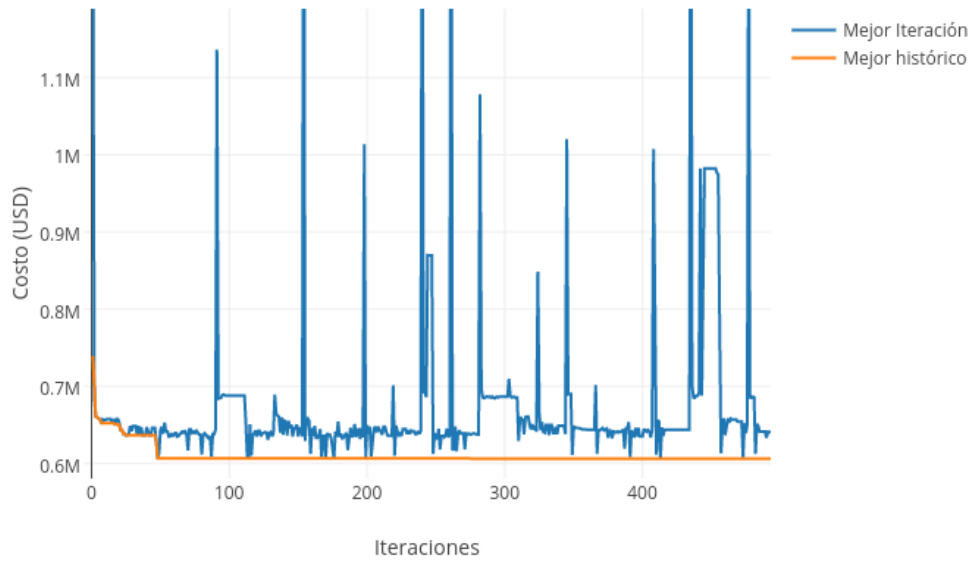


Figura 5.2: Convergencia PSO con control de parámetros.

La calidad de la población inicial de partículas influye directamente con áreas de búsqueda y la obtención de mínimos. Al construir partículas con insatisfacción de demanda, pero que cumplen las restricciones técnicas, permite agregar mayor variedad a la búsqueda y encontrar partículas de mayor calidad.

CAPÍTULO 5. RESULTADOS

5.2.2. Comparación

A continuación, en la tabla 5.3, se presenta una comparativa con los mejores costos de diferentes implementaciones. Se compara una relajación lagrangiana y un algoritmo genético presentado en [17], LRV y GAV respectivamente. Además, un algoritmo genético presentado por el autor del sistema de pruebas en [11], GAK y, finalmente, con el algoritmo evolutivo presentado en [18], EAM, que obtiene los mejores resultados en costos en todas las pruebas. PSOR es el algoritmo propuesto en esta memoria. .

Algoritmo	tipo	Número de generadores					
		10	20	40	60	80	100
LRV	Mejor	\$566107	\$1128362	\$2250223	\$3374994	\$4496729	\$5620305
	Promedio	\$566493	\$1128395	\$2250223	\$3374994	\$4496729	\$5620305
GAV	Mejor	\$565866	\$1128876	\$2252909	\$3377393	\$4507692	\$5626362
	Promedio	\$567329	\$1130160	\$2262585	\$3394044	\$4525204	\$5669362
EAM	Mejor	\$557110	\$1116240	\$2228070	\$3363220	\$4473210	\$5597470
	Promedio	\$562018	\$1120010	\$2246736	\$3367000	\$4484988	\$5605562
GAK	Mejor	\$565825	\$1126243	\$2251911	\$3376625	\$4504933	\$5627437
	Promedio	\$565825	\$1126243	\$2251911	\$3376625	\$4504933	\$5627437
PSOR	Mejor	\$603544	\$1207256	\$2415622	\$3835071	\$4958048	\$6120874
	Promedio	\$604588	\$1211265	\$2419578	\$3838001	\$4960355	\$6122532

Tabla 5.3: Comparativa de resultados

Capítulo 6

Conclusiones

En este capítulo se presenta un resumen de la metodología realizada y las principales conclusiones obtenidas de la implementación y pruebas, para luego, entregar las guías de trabajo futuro para mejorar el desempeño y resultados del algoritmo.

6.1. Resumen y Conclusiones

Se presentó en esta memoria un algoritmo basado en la inteligencia enjambres para solucionar el problema de generación eléctrica a corto plazo. Se explicaron los distintos aspectos del algoritmo, enfocándose en el diseño de los movimientos, representación, evaluación y parámetros.

- Se explicó el origen de la técnica y las principales características de su funcionamiento. Se presentaron las diferentes mejoras al algoritmo, incluyendo los diferentes algoritmos de sintonización y tipos de control de parámetros.
- Se presentó el problema de planificación eléctrica a corto plazo, junto con su modelo matemático y un resumen de los distintos algoritmos aplicados para su resolución, incluyendo heurísticas, sistemas expertos, relajaciones lagrangianas, algoritmos evolutivos y por enjambres.
- Se ha diseñado e implementado un algoritmo PSO para solucionar este problema detallando los movimientos, representaciones y parámetros. Se ha puesto a prueba el algoritmo con diferentes instancias para poder compararlo con otras soluciones y poder concluir los caminos a seguir para mejorar el desempeño.

CAPÍTULO 6. CONCLUSIONES

A continuación, los elementos que a tener en cuenta para obtener buenos resultados en la búsqueda y evitar problemas típicos,

- El considerar en los movimientos los bloques de funcionamiento ha permitido reducir el espacio de búsqueda de la representación binaria a N que generalmente son de un tamaño N^2 .
- La diversidad de la población inicial impacta directamente en la convergencia del algoritmo y su capacidad para explorar el espacio de búsqueda. El construir partículas que respeten sólo las restricciones técnicas, genera mayor variedad de soluciones.
- El movimiento swap fue el más exitoso al momento de reparar partículas que no cumplieran con la demanda eléctrica. Al considerar un factor de utilización y rendimiento aporta información relevante para poder obtener soluciones factibles respecto a la demanda eléctrica.
- La evaluación liviana, que evalúa los generadores con su potencia máxima sin resolver su EDP, permite reducir los tiempos.
- El control de parámetros para salir de óptimos locales es fundamental, luego de obtener una población inicial variada. El cambiar al 10 % de las partículas peor evaluadas por partículas en las fronteras del espacio sirvió para sacar el algoritmo de óptimos locales pequeños. Para la representación binaria las partículas fronterizas no aportan tanta variedad al enjambre y no logró ser tan efectivo. Además, se aplicó el movimiento swap a todas las partículas del enjambre si no hay cambios en 20 generaciones, pero este movimiento no es tan efectivo con partículas de calidad.

6.2. Trabajo Futuro

Existen diferentes aspectos en los cuales mejorar el algoritmo. Se listan a continuación los más relevantes para conseguir mejores resultados.

- Dar mayor información del dominio del problema, de forma que el espacio de búsqueda pueda ser reducido. Dar prioridad a las horas más demandadas o agrupar los generadores con características técnicas similares.
- Mejorar el aporte de los movimientos (crear, eliminar, alargar y acortar bloque) en la búsqueda con un control adaptivo, esto hará que actúen en conjunto y realicen mejores cambios a las partículas.

- Diseñar nuevas formas de explorar y escapar de óptimos locales. Introduciendo partículas con mayor diversidad al enjambre para guiarlo a nuevas zonas de búsqueda. Controlar los parámetros sociales y de inercia de forma adaptiva
- Implementar nuevos métodos en la creación de las partículas iniciales para mejorar la diversidad de la población y exploración del algoritmo.
- Considerar diferentes topologías para compartir información entre las partículas.
- La evaluación y actualización de las partículas es un proceso altamente paralelizable, considerar adaptar estas funciones ocupando múltiples CPUs o GPUs reduciría considerablemente los tiempos de ejecución.

Bibliografía

- [1] R. Kennedy, J.; Eberhart. *Particle Swarm Optimization*. Morgan Kaufmann Publishers, 1995.
- [2] Eberhart R. C. Shi, Y. H. *Experimental Study of Particle Swarm Optimization*. The 4th World. Multiconference on Systemics, 2000.
- [3] F. van den Bergh; A. P. Engelbrecht. *A Cooperative approach to particle swarm optimization*. IEEE Transactions on Evolutionary Computation, 2004.
- [4] Lovbjerg M.; Krink T. *Extending particle swarm with self-organized criticality*. IEEE Transactions on Evolutionary Computation, 2002.
- [5] M. Birattari; T. Statzle; L. Paquete; K. Varrentrap. *A Racing Algorithm for Configuring Metaheuristics*. Morgan Kaufmann, 2002.
- [6] A. Eiben V. Nannen. *Relevance Estimation and Value Calibration of Evolutionary Algorithm Parameters*. Joint International Conference for Artificial Intelligence (IJCAI), 2007.
- [7] H. H. Hoos F. Hutter. *Automatic Algorithm Configuration based on Local Search*. Proceedings of the Twenty-Second Conference on Artificial Intelligence, 2007.
- [8] A. Bramellar E. Khodaverdia and R Duenet. *Semi-rigorous thermal unit commitment for large scale electrical power system*. IEEE Proceedings in generation, transmission and distribution, 1996.
- [9] S. Shahidepour S. Tong and Z. Ouyang. *A heuristic short-term unit commitment*. IEEE Transactions on Power System, 1991.
- [10] A. Moller W. Romisch D. Dentcheva, R. Gollmer and R. Shultz. *Solving the unit commitment problem in power generation by primal and dual methods*. Progress in Industrial Mathematics at ECMI 96, pages 332-339, 1997.

BIBLIOGRAFÍA

- [11] V. Petridis S. Kazarlis, A. Bakirtzis. *A genetic algorithm solution to the unit commitment problem*. IEEE Transactions on Power Systems, 1996.
- [12] D. Srinivasan and A. Tettamanzi. *A genetic algorithm solution to the unit commitment problem*. In Proceedings of the 1995 IEEE International Conference on Evolutionary Computing, 1995.
- [13] Y. Abdel-Magid A. Mantawy and S. Selim. *Integrating genetic algorithms, tabu search, and simulated annealing for the unit commitment problem*. IEEE Transactions on Power Sys- tem, 1999.
- [14] Jalan Ayer keroh Lama. *Solving Unit Commitment Problem Using Hybrid Particle Swarm Optimization*. Multimedia University, 75450 Melaka, Malaysia, 2003.
- [15] B.R. Bai Y.J. Cao B. Zhao, C.X. Guo. *An improved particle swarm optimization algo- rithm for unit commitment*. College of Electrical Engineering, Zhejiang University, Hangzhou, Zhejiang 310027, China., 2006.
- [16] Lixin Tang Jin Lang and Zhongwei Zhang. *An improved binary particle swarm optimiza- tion for unit commitment problem*. The Logistics Institute, Northeastern University, Shenyang, China., 2008.
- [17] J. Valenzuela, A. Smith. *A seeded memetic algorithm for large unit commitment problems*. Journal of Heuristics.
- [18] Jorge Maturana; María-Cristina Riff. *Solving the short-term electrical generation scheduling problem by an adaptive evolutionary approach*. European Journal of Operational Research, 2005.