



**INAOE**

# **Esquema descentralizado para la calendarización de ciclos de encendido/apagado en un conjunto de micro generadores**

por

**Jorge Palominos Villegas**

Tesis sometida como requisito parcial para obtener el  
grado de

**Maestro en Ciencias Computacionales**

en el

**Instituto Nacional de Astrofísica, Óptica y  
Electrónica**

Febrero, 2016

Tonantzintla, Puebla

Supervisada por:

**José Enrique Muñoz de Cote Flores**

©INAOE 2016

El autor otorga al INAOE el permiso de reproducir y  
distribuir copias en su totalidad o en partes de esta  
tesis





*A mi familia y a mi prima Georgina que se fue dos semanas antes de mi examen.*



# Resumen

En la actualidad la energía eléctrica es importante para el desarrollo de la vida diaria. En algunos lugares pequeños y alejados no se tiene acceso a la red eléctrica, por lo que se hace necesario implementar una solución para satisfacer su demanda de energía. Para resolver el problema de acceso a la red se propone la creación de redes a pequeña escala y autosuficientes llamadas microrredes. Las microrredes son capaces de satisfacer las necesidades energéticas de lugares remotos desconectados de la red eléctrica nacional. Un problema importante en las microrredes es contar con un programa de calendarización adecuada de encendido y apagado de los microgeneradores de energía, para poder minimizar costos de generación así como satisfacer la demanda.

El problema de la calendarización de encendido y apagado de microgeneradores de energía se puede atacar de manera centralizada y descentralizada. Se pretende enfrentar el problema de la calendarización de manera descentralizada utilizando la teoría multiagente para permitir la escalabilidad, dar mayor robustez ante cambios y reducir el tiempo de ejecución computacional.

Se propone una extensión a la formulación clásica de un problema de optimización con restricciones distribuido (DCOP) para que los algoritmos de esta familia de problemas puedan resolver problemas con temporalidad, es decir, problemas que consideren más de un tiempo para su resolución. El objetivo general es optimizar los ciclos de encendido y apagado a un horizonte de 24 horas.

La solución incluirá temporalidad, pero esta representación crece exponencialmente en relación al número de variables en el problema, por lo que la solución consta de los siguientes elementos:

1. Particionar el problema en *ventanas* y repetir el proceso varias veces (por ejemplo: 24, 12, 8, ... veces para ventanas de tamaño 1, 2, 3, ...), donde cada ventana tiene diferentes condiciones iniciales las cuales dependen de la ventana anterior. Para las ventanas de tamaño mayor a 2 ya se incluye temporalidad. Cuando se aumenta el tamaño en las ventanas de optimización se va mejorando la solución. Con esta representación se puede llegar a un punto de equilibrio entre el tiempo de ejecución y la solución obtenida. Este punto de equilibrio puede ser la configuración óptima debido a que no es necesario contar con toda la información del problema para encontrar la configuración óptima.
2. Debido a que el problema de optimización crece de manera exponencial en relación al número de variables, se propone usar una heurística para guiar una búsqueda. Se propone dividir el dominio de cada variable para poder encontrar un mínimo local. Al usar esta heurística se reduce el tiempo de ejecución.

3. Al usar una heurística para reducir el dominio de las variables se corre el riesgo de caer en mínimos locales por lo que se propone realizar una búsqueda guiada para mejorar el resultado. Se propone usar la búsqueda guiada  $A^*$  con modificaciones para que pueda funcionar sin un objetivo. Con esta propuesta se puede mejorar la solución previamente obtenida.

Los experimentos realizados en este trabajo tienen como objetivos validar la solución descentralizada contra la centralizada y evaluar la representación propuesta utilizando diversos algoritmos de la familia *DCOP* con diferentes características. Para los experimentos se utilizaron los siguientes cuatro algoritmos de la familia DCOP: *Synchronous Branch and Bound (SynchBB)*, *Distributed Pseudotree-Optimization Procedure (DPOP)*, *Memory Bounded - Distributed Pseudotree-Optimization Procedure (MB-DPOP)* y *Asynchronous Forward Bounding (AFB)*. Para la validación de la solución descentralizada se utilizan cuatro medidas: error, tiempo de ejecución, número de mensajes y tamaño total de los mensajes intercambiados entre agentes.

# Abstract

Nowadays, electrical energy is important for the development of daily living. In some small and remote areas, there are not access to the electric network, so it is necessary to implement a solution to meet its energy demand. To solve this problem is proposed to create selfsufficient and smallscale networks, called microgrids. A major problem in microgrids is to have a proper scheduling program for on and off the microgenerators of energy, to minimise generation costs and meet the demand.

The problem of scheduling on and off the energy microgenerators can be attacked with a centralised and decentralised way. It aims to address the scheduling problem by the decentralised way using multiagent theory to allow scalability and provide greater robustness to changes and reduce computational time execution.

We propose an extension to the classic formulation of a distributed constraint optimisation problem DCOP for the algorithms of this family of problems to resolve problems with temporality, that is, problems that consider more than one time for resolution. The overall objective is to optimise on and off cycles to a horizon of 24 hours.

We propose an extension of the classic constraint graph formulation for DCOPs so that classic algorithms can solve long-term objectives. The overall objective is to optimise to a 24-hour horizon.

We propose a solution where temporality can be included, but this representation increases exponentially regarding to number of variables in the problem. For this reason, we propose:

1. Partitioning the problem in optimisation windows and repeat the process several times (e. g. 24, 12, 8, ... times, for window sizes of 1, 2, 3, ...), where each window has different initial conditions which depend on the previous window. For windows larger than 2, temporality is already included. When the window optimisation size is increased, the solution will be improved. This representation can reach a point of balance between runtime and the obtained solution. That balance may be the optimal configuration because it is not necessary to have all the information of the problem to find the optimal configuration.
2. Due to the optimisation problem grows exponentially with the number of variables, we propose to use a heuristic to guide a search. We propose to divide the domain of each variable to find a local minima. Using this heuristic, the runtime is reduced.
3. By using a heuristic to reduce the domain of the variables you run the risk of falling into local minima, so we propose to perform a guided search to improve the outcome. We propose

to use the guided search  $A^*$  with some small modifications so that it can run without a goal. With this proposal can improve the previously obtained solution.

The aim of the experiments is to validate the proposed centralised solution against existing decentralised solution. Besides that, another goal of the experiments is to validate the operation of the proposed solution with various algorithms from DCOP family. For validation of decentralised solution, four measures are used: error, runtime, the number of messages and the total size of the messages exchanged between agents.

For experiments, this four algorithms from DCOP family were used: *Synchronous Branch and Bound (SynchBB)*, *Distributed Pseudotree-Optimization Procedure (DPOP)*, *Memory Bounded - Distributed Pseudotree-Optimization Procedure (MB-DPOP)* y *Asynchronous Forward Bounding (AFB)*.



## Agradecimientos

Primero que nada quiero agradecer a mis padres y a mis hermanos. A mis padres Jorge Palominos Mendoza y María del Carmen Villegas Barrera les agradezco que siempre apoyaron mis decisiones y me dieron impulso cuando creía que no podía. A mis hermanos Alejandro Palominos Villegas y Karla Itzel Palominos Villegas les agradezco que siempre estuvieron para lo que necesite y siempre creyeron en mi.

Mi siguiente agradecimiento es para mi asesor de tesis José Enrique Muñoz de Cote debido a que con él aprendí muchas cosas y me apoyo cuando lo necesite. Además le agradezco todo el tiempo que me dedico para orientarme a la realización de mi investigación. Les agradezco a mis revisores Felipe Orihuela Espina, Enrique Sucar Succar y Leopoldo Altamirano Robles por tomarse el tiempo de revisar mi documento de tesis y darme sus comentarios para mejorarlo.

Quiero agradecer a mi familia que siempre creyeron en mi y me apoyaron en lo que necesite.

Agradezco a mis compañeros y amigos de maestría Ramón Izquierdo Córdoba, Ojilvie Avila Córtes, Argenis Aldair Aroche Villarruel, Jesus Joel Rivas, Mauro Martín Letras Luna, Nahúm Sánchez Taxis y Magdiel Jiménez Guarneros porque con ellos aprendí muchas cosas durante mi estancia en el INAOE.

Agradezco a Antonio de Jesús Olivares Vargas, Hiram Enrique Martinez Mateo y Berni Manolo Pérez Ramos por ser mis compañeros y amigos fuera del INAOE.

Agradezco a CONACYT por apoyarme con la beca número 367051.

Por último, quiero agradecer a mi prima Georgina que siempre estuvo apoyándome cuando lo necesite. Siempre te recordaré por tu gran carácter y tu buen humor.



# Contenido

<b>Resumen</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Agradecimientos</b>	<b>ix</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Problemática . . . . .	3
1.2. Pregunta de investigación . . . . .	4
1.3. Hipótesis . . . . .	4
1.4. Objetivo . . . . .	4
1.4.1. Objetivos específicos . . . . .	4
1.5. Metodología . . . . .	4
1.6. Contribuciones . . . . .	5
1.7. Estructura de la tesis . . . . .	6
<b>2. Preliminares</b>	<b>7</b>
2.1. Microredes . . . . .	7
2.1.1. Productor de Energía Virtual . . . . .	10
2.2. Optimización descentralizada . . . . .	11
2.2.1. Problema de satisfacción de restricciones . . . . .	11
2.2.2. Grafos factorizados . . . . .	12
2.2.3. Pseudoárbol . . . . .	13
2.2.4. Problema de optimización con restricciones distribuido (DCOP) . . . . .	14
2.3. FRODO . . . . .	21
2.4. Búsqueda A* . . . . .	21
2.5. Resumen . . . . .	22
<b>3. Formalización</b>	<b>23</b>
3.1. Formulación matemática del problema . . . . .	23
3.1.1. Función objetivo . . . . .	24
3.1.2. Restricciones . . . . .	25
3.2. Escenarios . . . . .	27

3.3.	Solución centralizada . . . . .	28
3.3.1.	Solución de Morais et al. 2010 . . . . .	28
3.3.2.	Solución replicada . . . . .	29
3.4.	Resumen . . . . .	30
<b>4.</b>	<b>Trabajo relacionado</b>	<b>31</b>
4.1.	Optimización de despacho centralizado en el área de energía . . . . .	32
4.2.	Optimización de despacho descentralizado en el área de energía . . . . .	34
4.3.	Otras aplicaciones . . . . .	36
4.4.	Resumen . . . . .	37
<b>5.</b>	<b>Método propuesto</b>	<b>41</b>
5.1.	Optimización por ventanas . . . . .	42
5.2.	Optimización utilizando búsqueda primero el mejor . . . . .	43
5.3.	Optimización utilizando $A^*$ . . . . .	44
5.4.	Resumen . . . . .	48
<b>6.</b>	<b>Pruebas experimentales</b>	<b>49</b>
6.1.	Solución centralizada . . . . .	50
6.2.	Solución descentralizada . . . . .	55
6.2.1.	Ventanas de optimización . . . . .	55
6.2.2.	Solución descentralizada usando búsqueda primero el mejor . . . . .	62
6.2.3.	Solución descentralizada usando una heurística para guiar la búsqueda . . . . .	68
6.3.	Resumen . . . . .	73
<b>7.</b>	<b>Conclusiones y trabajo futuro</b>	<b>75</b>
7.1.	Conclusiones . . . . .	75
7.2.	Contribuciones . . . . .	77
7.3.	Trabajo futuro . . . . .	77
	<b>Referencias</b>	<b>79</b>

# Índice de Figuras

1.1. Conjunto de microgeneradores conectados a una microrred. . . . .	3
2.1. Estructura básica de una micro red. . . . .	8
2.2. Dos ejemplos del problema de coloreo de grafos. . . . .	12
2.3. Ejemplo de grafo factorizado. . . . .	13
2.4. Ejemplo de un pseudoárbol. . . . .	14
2.5. Ejemplo del funcionamiento de SynchBB. . . . .	16
2.6. Construcción del pseudoárbol. . . . .	17
2.7. Ejemplo de ejecución de DPOP. . . . .	18
2.8. Ejemplo de un pseudoárbol agrupado por MB-DPOP. . . . .	19
2.9. Grafo con restricciones. . . . .	20
2.10. Arquitectura general de FRODO. . . . .	21
4.1. Marco de trabajo Su et al., 2014. . . . .	33
4.2. Concepto de microrred. . . . .	34
4.3. Niveles de control de un entorno microrred presentado en Dimeas y Hatziargyriou, 2005. . . . .	38
5.1. Optimización por medio de ventanas. . . . .	43
5.2. Diagrama de flujo de la realización del particionado del dominio. . . . .	44
5.3. Ejemplo de particionado. . . . .	45
5.4. Ejemplo de funcionamiento de la búsqueda $A^*$ . . . . .	48
6.1. Resultados de dos diferentes configuraciones de producción óptimos para el escenario 1. . . . .	53
6.2. Resultados de dos diferentes configuraciones de consumo óptimos para el escenario 1. . . . .	53
6.3. Resultados de dos diferentes configuraciones de producción óptimos para el escenario 2. . . . .	54
6.4. Resultados de dos diferentes configuraciones de consumo óptimos para el escenario 2. . . . .	54
6.5. Ventanas de optimización, escenario 1, ventana de tamaño 1, algoritmo AFB. . . . .	58
6.6. Ventanas de optimización, escenario 1, ventana de tamaño 1, algoritmos DPOP, MB-DPOP y SynchBB. . . . .	58
6.7. Ventanas de optimización, escenario 1, ventana de tamaño 2, algoritmo AFB. . . . .	59

6.8. Ventanas de optimización, escenario 1, ventana de tamaño 2, algoritmos DPOP, MB-DPOP y SynchBB. . . . .	59
6.9. Ventanas de optimización, escenario 2, ventana de tamaño 1, algoritmo AFB. . . . .	60
6.10. Ventanas de optimización, escenario 2, ventana de tamaño 1, algoritmos DPOP, MB-DPOP y SynchBB. . . . .	61
6.11. Ventanas de optimización, escenario 2, ventana de tamaño 1, algoritmos AFB, DPOP, MB-DPOP y SynchBB. . . . .	61
6.12. Primero el mejor, escenario 1, ventana de tamaño 1, algoritmos DPOP, MB-DPOP y SynchBB. . . . .	63
6.13. Primero el mejor, escenario 1, ventana de tamaño 2, algoritmo DPOP. . . . .	64
6.14. Primero el mejor, escenario 1, ventana de tamaño 3, algoritmos DPOP, MB-DPOP y SynchBB. . . . .	64
6.15. Primero el mejor, escenario 2, ventana de tamaño 1, algoritmos DPOP, MB-DPOP y SynchBB. . . . .	66
6.16. Primero el mejor, escenario 2, ventana de tamaño 2, algoritmos DPOP, MB-DPOP y SynchBB. . . . .	66
6.17. Primero el mejor, escenario 2, ventana de tamaño 3, algoritmos DPOP y MB-DPOP. . . . .	67
6.18. $A^*$ , escenario 1, ventana de tamaño 1, 2, 4 y 6, algoritmos AFB, DPOP, MB-DPOP y SynchBB. . . . .	70
6.19. $A^*$ , escenario 1, ventana de tamaño 3, algoritmos AFB, DPOP, MB-DPOP y SynchBB. . . . .	70
6.20. $A^*$ , escenario 2, ventana de tamaño 1 y 2, algoritmos AFB, DPOP, MB-DPOP y SynchBB. . . . .	71
6.21. $A^*$ , escenario 2, ventana de tamaño 3, 4 y 6, algoritmos AFB, DPOP, MB-DPOP y SynchBB. . . . .	71

## Índice de Tablas

3.1. Predicciones de generadores y escenarios de consumo obtenidos de Morais et al. 2010.	27
4.1. Resumen de trabajos en la literatura centralizados que enmarcan el desarrollo de la presente tesis. . . . .	39
4.2. Resumen de trabajos en la literatura descentralizados que enmarcan el desarrollo de la presente tesis. . . . .	40
6.1. Predicciones de generadores y escenarios de consumo obtenidos de Morais et al. 2010.	50
6.2. Resultados obtenidos mediante Java-ILOG CPLEX en escenario 1, dos configuraciones óptimas. . . . .	52
6.3. Resultados obtenidos mediante Java-ILOG CPLEX en escenario 2, dos configuraciones óptimas. . . . .	52
6.4. Medidas de comparación de optimización a largo plazo utilizando ventanas de optimización para el escenario 1. El tiempo de ejecución mostrado es en segundos, mientras el tamaño de mensajes es presentado en megabytes. . . . .	57
6.5. Medidas de comparación de optimización a largo plazo utilizando ventanas de optimización para el escenario 2. El tiempo mostrado es en segundos, mientras el tamaño de mensajes es presentado en megabytes. . . . .	60
6.6. Medidas de comparación de optimización a largo plazo con búsqueda primero el mejor para el escenario 1. El tiempo de ejecución mostrado es en segundos, mientras el tamaño de mensajes es presentado en megabytes. . . . .	63
6.7. Medidas de comparación de optimización a largo plazo para el escenario 2 con búsqueda primero el mejor. El tiempo de ejecución mostrado es en segundos, mientras el tamaño de mensajes es presentado en megabytes. . . . .	65
6.8. Error en predicciones y tiempo de ejecución usando heurística para guiar una búsqueda para el escenario 1. El tiempo de ejecución mostrado es en segundos. . . . .	69
6.9. Error en predicciones y tiempo de ejecución usando heurística para guiar una búsqueda para el escenario 2. El tiempo ejecución mostrado es en segundos. . . . .	69
6.10. Promedio del número y tamaño de mensajes intercambiados entre agentes usando heurística para guiar una búsqueda para el escenario 1. El tamaño de mensajes se muestra en megabytes. . . . .	72

6.11. Promedio del número y tamaño de mensajes intercambiados entre agentes usando heurística para guiar una búsqueda para el escenario 2. El tamaño de mensajes se muestra en megabytes. . . . .	72
---	----



# Índice de Algoritmos

5.1. Particionado $A^*$ . . . . .	46
5.2. DCOP- $A^*(\mathcal{C})$ . . . . .	47



# 1

## Introducción

En la actualidad el uso de la energía eléctrica es importante para el desarrollo de la vida diaria. En los hogares se cuenta con dispositivos eléctricos de uso doméstico así como de uso personal que facilitan la vida diaria. En la industria la mayoría de las máquinas utilizadas funcionan con energía eléctrica para la fabricación de alimentos, prendas de vestir y cualquier objeto de uso cotidiano, así como de uso industrial. Así como las aplicaciones mencionadas anteriormente, se pueden encontrar muchas más donde la energía eléctrica es importante para su funcionamiento, por todo lo mencionado se puede decir que la energía eléctrica es indispensable para la vida diaria.

Actualmente existe una tendencia de transición de la producción de energía centralizada hacia sistemas de producción de energía descentralizados, esto sucede tanto para áreas rurales y remotas como para las grandes redes centralizadas.

El uso de generadores de energía por combustible, o térmicos, entre otros, generan una gran cantidad de contaminantes para el medio ambiente. En consecuencia, las grandes plantas generadoras de energía son también grandes generadoras de contaminantes que dañan al medio ambiente. Es por eso que existe la tendencia a utilizar fuentes de generación de energía alternativas (renovables). Tales como la energía solar, eólica, biomasas, entre otras. Al utilizar las energías renovables además de poder satisfacer la demanda, se ayuda al medio ambiente no emitiendo demasiados contaminantes (Bertani et al., 2004).

En algunos lugares pequeños y alejados no se tiene acceso a la red de distribución de energía eléctrica, por lo que se hace necesario implementar una solución para satisfacer su demanda de energía. Una de las opciones comúnmente seleccionadas es la creación de una microrred autosuficiente, es decir, crear una pequeña red de distribución local que sea alimentada por generadores locales. En este tipo de redes la tendencia es usar microgeneradores de energía renovables (Awad et al., 2008; Hatziargyriou y Meliopoulos, 2002; Praça et al., 2003).

El uso de las microrredes no es exclusivo de áreas pequeñas, esto también se implementa en las

grandes redes de energía. En estas grandes redes de distribución se crean microrredes para poder tener zonas autosuficientes que puedan operar en caso de que la red de distribución sufra una falla. Además estas microrredes pueden tener un control separado de la red de distribución. Las microrredes por lo general cuentan con baterías para el almacenamiento de energía. El uso de baterías se debe a que contienen gran capacidad de almacenamiento y retención (pérdida de energía en el proceso de carga y descarga) de energía.

En todas las redes de distribución de energía existe el problema de control de los recursos que esta contiene. Una manera sencilla de administrar los elementos de la red es tener encendidos los generadores y la energía sobrante almacenarla. El problema que esto conlleva es el costo que esto provoca debido a la generación excesiva de energía así como el desgaste de los generadores y de las baterías. Otra manera de administrar este problema es calendarizando los ciclos de encendido y apagado de los microgeneradores. El problema de calendarización se refiere a fijar anticipadamente las fechas de ciertas actividades a lo largo de un período. En las microrredes existen dos tipos de microgeneradores, los controlables y los no controlables. Los microgeneradores controlables son aquellos a los que se les puede encender, apagar y controlar la energía que producen. Los microgeneradores no controlables son aquellos en los que no se puede controlar la cantidad de energía que producen, como ejemplo de estos tenemos las turbinas de viento y los generadores fotovoltaicos. Por lo general las microrredes cuentan con unidades de almacenamiento (baterías), estas son controlables y se utilizan con el fin de satisfacer la demanda cuando las unidades de producción no son suficientes.

Es de importancia crear una calendarización para los ciclos de encendido y apagado de los generadores para poder satisfacer la demanda así como mantener el balance en la red. Esto es debido a que si no se tiene un correcto uso de los microgeneradores pueden ocurrir dos cosas, la primera es que no se produzca suficiente energía para satisfacer la demanda y la segunda es que se produzca energía de sobra y con eso saturar la microrred y provocar apagones. Otro objetivo común en el problema de calendarización de ciclos de encendido y apagado es minimizar el costo de producción a largo plazo.

Para la calendarización de ciclos de encendido y apagado de los generadores en la red existen diversas técnicas centralizadas así como descentralizadas. Cada una de estas técnicas tienen ventajas que dependen de las condiciones de los generadores así como del tamaño de la microrred. Como las redes están creciendo, el utilizar técnicas centralizadas trae consigo muchos problemas. Las técnicas centralizadas necesitan conocer toda la red y por el tamaño que estas tienen es cada vez más complicado resolver la calendarización de toda la red. Otro problema es que si la unidad central falla toda la red falla. Es por estas razones que existe la tendencia hacia técnicas descentralizadas.

Aunque existe mucho trabajo en el desarrollo de algoritmos de optimización en el área de investigación de operaciones esta tesis se centra en el análisis de la complejidad de la representación en diferentes algoritmos DCOP y por lo tanto no se contemplan los problemas referentes a la electrónica.

## 1.1 Problemática

Se cuenta con un conjunto de microgeneradores de energía no controlables (turbinas de viento, generador solar) y un conjunto de microgeneradores de energía controlables (celdas de combustible) conectados a una microrred. Además de los generadores se cuenta con dispositivos con capacidad de almacenamiento (baterías), caracterizadas por valores que indican una capacidad máxima de almacenamiento así como una velocidad máxima de carga y descarga (Morais et al., 2010). La representación gráfica del problema se muestra en la figura 1.1.

Se desea obtener una calendarización de ciclos de encendido y apagado a 24 horas en intervalos regulares de una hora de los microgeneradores controlables en una microrred que optimice el costo de producción mientras que se satisface la demanda de energía.

Se cuenta con las predicciones de energía generada en las siguientes 24 horas para los microgeneradores de energía solar, de viento así como para el consumo (cargas) de la red. Estas predicciones son el estimado de la producción de energía que se generará y consumirá en las siguientes 24 horas. Las predicciones que se tienen están dividida por horas.

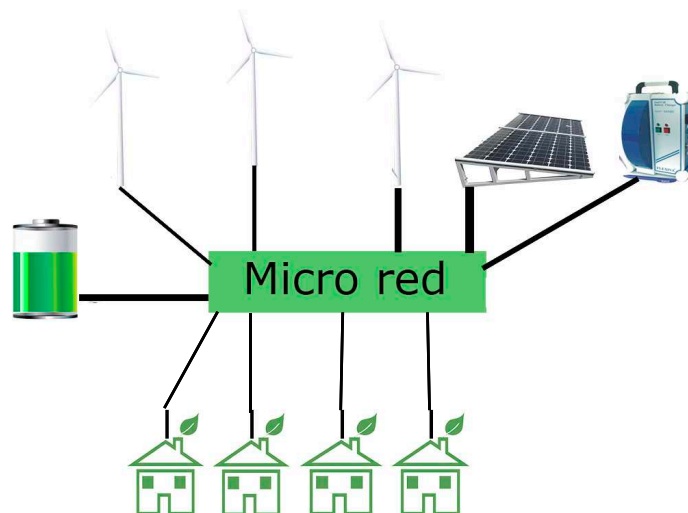


Figura 1.1: Conjunto de microgeneradores conectados a una microrred. Se tiene un conjunto de generadores eólicos, generadores solares y celdas de combustible, se cuentan con unidades de almacenamiento así como un conjunto de cargas (consumo) conectados a una microrred.

## 1.2 Pregunta de investigación

¿Es posible realizar una arquitectura descentralizada con la que se pueda resolver el problema de calendarización de ciclos de encendido y apagado en un conjunto de microgeneradores?

## 1.3 Hipótesis

Es posible formular un esquema descentralizado que pueda ser usado por uno o más algoritmos de la familia de problemas de optimización con restricciones distribuido (DCOP) para optimizar el uso de un conjunto de microgeneradores a largo plazo; cuya solución se aproxime o incluso iguale la óptima.

## 1.4 Objetivo

El objetivo general de esta tesis es diseñar soluciones descentralizadas a largo plazo para la calendarización de ciclos de encendido y apagado en un conjunto de microgeneradores en una microrred cuya solución se aproxime o inclusive iguale la solución óptima.

### 1.4.1 Objetivos específicos

- Formulación matemática del problema de (Morais et al., 2010) de manera centralizada con el fin de replicar los resultados.
- Definición del problema matemático de manera descentralizada.
- Implementación de la solución del problema computacional del dominio de las variables del problema.
- Validación del modelo.

## 1.5 Metodología

A partir de la formulación matemática dada en (Morais et al., 2010) esta se implementó usando el lenguaje de programación Java para crear los archivos para la herramienta de optimización ILOG CPLEX para replicar los resultados.

A partir de la formulación matemática se propuso una arquitectura para representar el problema de encendido y apagado en un conjunto de microgeneradores como un problema de optimización con restricciones distribuido (DCOP). En esta arquitectura se propone la inclusión de temporalidad.

Una vez definida una arquitectura, el problema puede ser resuelto utilizando el marco de trabajo de FRODO, un esquema de desarrollo que provee diversos algoritmos DCOP.

Se presentan 3 propuestas de solución al problema. La primera propuesta es dividir el problema en subproblemas (ventanas de optimización, ver sección 5.1 para explicación sobre ventanas); es decir, se optimiza varias veces de manera secuencial. Para las ventanas de tamaño mayor a 2 ya se incluye temporalidad. Cuando se aumenta el tamaño en las ventanas de optimización se va mejorando la solución. Con esta representación se puede llegar a un punto de equilibrio entre el tiempo de ejecución y la solución obtenida. Este punto de equilibrio puede ser la configuración óptima debido a que no es necesario contar con toda la información del problema para encontrar la configuración óptima.

La segunda propuesta es reducir el número de combinaciones con las que cuenta el problema. Esto debido a que es un problema combinatorio; es decir, el número de combinaciones crece de manera exponencial con el número de variables que tienen. Se propone una heurística de particionado del dominio utilizando una búsqueda primero el mejor. Al usar esta heurística se reduce el tiempo de ejecución.

La tercera solución propuesta es mejorar la solución mediante una búsqueda guiada  $A^*$ . Se propone usar la búsqueda guiada  $A^*$  con unas modificaciones para que pueda funcionar sin un objetivo. Con esta propuesta se puede mejorar la solución previamente obtenida.

Se seleccionaron los algoritmos de la familia DCOP para realizar los experimentos. Se eligieron algoritmos con diferentes características como el tipo de resolución (programación dinámica, ramificación y poda, etc.), exactitud (exactos o aproximados) y tipo de implementación (ramificación completa o con límite de memoria en ejecución). La validación de los experimentos se realizó mediante 4 métricas: error, tiempo de ejecución, número y tamaño de mensajes. Para calcular el error se utiliza error absoluto, donde el costo de la configuración obtenido de manera centralizada (costo óptimo) es comparado con el costo de la configuración obtenida mediante los algoritmos de la familia DCOP. El tiempo de ejecución, el número y tamaño de mensajes se validó contra los algoritmos de la familia DCOP.

## 1.6 Contribuciones

1. Una arquitectura de optimización a largo plazo que puede ser usada por cualquier algoritmo que pertenece a la familia de DCOPs.
2. Una heurística para hacer reducir el número de combinaciones en los dominios de las variables y con esto la reducción del tiempo computacional. Esta heurística consiste en un particionado del dominio utilizando una búsqueda primero el mejor.

3. Un modelo descentralizado de optimización a largo plazo para el problema de calendarización en microrredes.

## 1.7 Estructura de la tesis

El contenido de este documento está organizado en 7 capítulos. En el capítulo 2 se introducen los conceptos necesarios. Se expone el concepto microrred y productor de energía virtual, su funcionamiento y estructura. Se describen los recursos de energía renovables más utilizados, que además son los que marcan el desarrollo de esta tesis. Se describen los algoritmos que se utilizaron para realizar pruebas así como el marco de trabajo FRODO.

En el capítulo 3, se muestra la formulación matemática del problema. En dicho capítulo se indica la función objetivo, las restricciones del problema así como los escenarios de prueba.

En el capítulo 4, se presentan los modelos, algoritmos y soluciones que se han propuesto en la literatura, que se encuentran relacionados con la presente tesis. Se exhiben los algoritmos de optimización centralizados en el área de energía. Además se presentan los algoritmos descentralizados propuestos para la optimización en el área de energía. Se exponen algoritmos utilizados para otras áreas de aplicación. Por último se da un resumen del trabajo relacionado.

En el capítulo 5 se muestra los modelos de solución propuestos para encontrar una calendarización de los ciclos de encendido y apagado de un conjunto de microgeneradores que minimicen el costo de producción satisfaciendo la demanda.

En el capítulo 6, se muestran los resultados de las pruebas experimentales. En este capítulo se muestran tablas y gráficas para observar comportamiento de las soluciones propuestas con diferentes algoritmos. Se utilizan diferentes medidas de comparación, que son: error, tiempo, número de mensajes intercambiados entre agentes y tamaño de los mensajes intercambiados entre agentes.

Finalmente en el capítulo 7, se presentan las conclusiones y posibles áreas en que se podría extender el trabajo de investigación.



# 2

## Preliminares

En este capítulo se realiza un resumen de las bases teóricas de los diferentes elementos que enmarcan el desarrollo de esta tesis. Se da el concepto de microrred así como la explicación de por qué es importante el uso de productores de energía renovable, así como los preliminares en optimización en microrredes de manera centralizada y descentralizada.

Un algoritmo de búsqueda sirve para buscar un elemento con propiedades específicas entre una colección de diferentes elementos, al final de este capítulo se explicará la búsqueda  $A^*$ .

### 2.1 Microredes

Los incentivos económicos, tecnológicos y ambientales han estado cambiando el rumbo de la generación y transmisión de energía eléctrica. Las complicaciones de generación centralizada están dando paso a la generación descentralizada, esto gracias al cambio de la economía en el área eléctrica (Lasseter y Paigi, 2004).

La inserción indiscriminada de generadores distribuidos pueden causar muchos problemas (saturado de las líneas de transmisión, el control de generadores de la red, etcétera) que se tienen que resolver. La mejor manera con que se han resuelto estos problemas ha sido con el enfoque de asociación de generadores y cargas (consumidores) en un subsistema llamado microrred. Este enfoque permite el control local de la generación distribuida permitiendo eliminar el control centralizado.

Una microrred es un sistema de generación eléctrica a pequeña escala y bidireccional que permite que los consumidores puedan ser a la vez productores. Una microrred es autosuficiente, es decir, cuenta con la suficiente producción de energía para satisfacer su demanda. En una microrred la producción de energía se encuentra cerca de los consumidores finales. Una microrred puede estar o no conectada a la red de distribución y esta puede operar de manera independiente. Además de unidades de producción de energía en las microrredes por lo general se cuenta con unidades

de almacenamiento, esto permite que la microrred tenga mayor estabilidad. La tendencia en las microrredes es el uso de productores de energía alternativa (renovables).

En la figura 2.1 se muestra la estructura básica de una microrred. En esta estructura se pueden incluir diferentes fuentes de generación además de fuentes de almacenamiento. En la figura una microrred abarca una pequeña zona. Además se puede observar que algunos consumidores también son productores, como los que cuentan con paneles solares en techo o pequeños generadores eólicos. Como se mencionó anteriormente, las microrredes también pueden contar con productores convencionales (una planta de producción de energía atómica o generación por medio de combustibles fósiles).

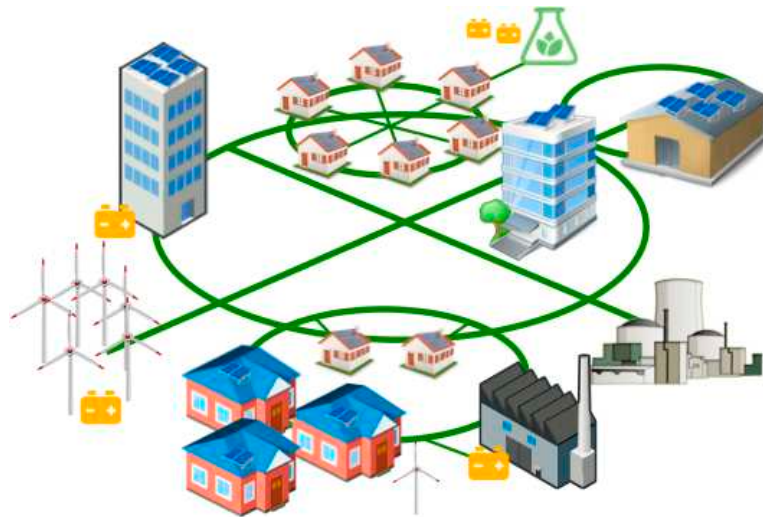


Figura 2.1: Estructura básica de una micro red. Se integran a la red de distribución fuentes de generación, fuentes de almacenamiento y cargas. Las líneas representan la red de distribución, los ventiladores representan los generadores eólicos, las pequeñas baterías representan las unidades de almacenamiento, el matraz representa la generación por medio de biomásas.

En las microrredes existe el problema de calendarización de ciclos de encendido y apagado de los microgeneradores (no es exclusivo de las microrredes). Este problema consiste en que se debe optimizar los recursos de producción de energía para satisfacer la demanda. La optimización, en general, se realiza para producir la energía necesaria al menor costo.

Debido a que la tendencia de las microrredes es el uso de recursos de producción de energía renovable a continuación se muestran los recursos más usados.

### **Energía eólica**

La fuerza del viento se ha usado para diferentes actividades, por ejemplo impulsar veleros, molinos de viento (para moler granos o bombear agua) o para prevenir que el océano inunde tierras bajas (Hansen, 2008).

Debido a las recientes crisis de combustibles y los esfuerzos para reducir la emisión de contaminantes al medio ambiente se ha reavivado el interés por el desarrollo de la tecnología de energía producida por viento (Mathema, 2008). La generación de energía eléctrica eólica se ha posicionado como la fuente de generación renovable más importante en las últimas décadas.

La turbina de eje horizontal, es la que comúnmente domina las aplicaciones de turbina de viento. Consta de una torre y una góndola. La góndola contiene el generador, la caja de reducción y el rotor.

Para sistemas de abastecimiento eléctrico rural existen modelos de generación eólica de hasta 300 kW, las turbinas de viento son usadas en combinación de algunos otros generadores (como diésel o solares) y algunas veces con un sistema de baterías.

### **Energía solar**

La energía solar puede ser usada directamente para diversas tareas domésticas. Además de esto, la energía solar puede ser convertida en eléctrica mediante diversas técnicas, principalmente termal y fotovoltaica. Las tecnologías termal convierte la energía solar en calor que puede ser almacenada o convertida en energía mecánica o eléctrica por medio de dispositivos apropiados (Mathema, 2008).

La conversión fotovoltaica se realiza directamente de la luz del sol a electricidad. Los generadores fotovoltaicos están contruidos con un diseño simple que necesita poco mantenimiento y que ofrece muchas ventajas como que pueden generar desde microwatts hasta megawatts (Parida et al., 2011).

Los sistemas de generación fotovoltaicos consisten en múltiples componentes como celdas, conexiones mecánicas y eléctricas, soportes y medios de regulación y/o modificación de la salida eléctrica (Parida et al., 2011).

### **Celdas de combustibles**

Las celdas de combustible son un aparato electroquímico que convierte la energía química del combustible a energía eléctrica sin la combustión del combustible. Las celdas de combustible toman la energía química derivada de la reacción electroquímica del combustible, la combina con un oxidante y la transforman en agua, electricidad y calor (Peighambardoust et al., 2010).

Generalmente las celdas de combustibles se clasifican en base al electrólito usado. Con base en lo anterior se obtiene la siguiente clasificación (Peighamardoust et al., 2010):

- Celdas de combustible alcalinas.
- Celdas de combustible de ácido fosfórico.
- Celdas de combustible de intercambio de protones sólidos.
- Celdas de combustible de carbonato fundido.
- Celdas de combustible de óxidos sólidos.

### **Almacenamiento de energía**

Un problema que se ha presentado a lo largo del tiempo es el almacenamiento de la energía eléctrica. Se puede almacenar mediante capacitores, pero este tipo de almacenamiento tienen la desventaja de que es mínima la energía que puede retener. Por lo general, para el almacenamiento de energía eléctrica se requiere su conversión a otro tipo de energía. Las baterías, por ejemplo, almacenan mediante el uso de componentes químicos que logran almacenar la energía. Otro proceso ocurre al descargar, que permite dar como salida energía eléctrica con un cierto voltaje. Para algunas baterías este proceso puede ser reversible y entonces esta puede ser recargada. Existen dos tipos de sistemas de baterías (Kiehne, 2003):

- Baterías primarias que son diseñadas con componentes químicos que pueden almacenar sólo una vez.
- Baterías secundarias que pueden ser recargadas.

A pesar de que existen dos tipos de baterías no hay una división clara entre ellas, debido a que las baterías secundarias tienen un uso limitado; es decir, en cierto número de recargas o tiempo de vida en que estas ya no podrán ser recargadas nuevamente.

Cuando se cuenta con baterías, lo usual es cargarlas cuando existe poca demanda de energía y descargarlas cuando existe gran demanda de esta. Las baterías usualmente poseen una gran capacidad de almacenamiento y por lo regular poseen gran capacidad de retención; es decir, no pierden mucha energía en el proceso (Khaligh y Li, 2010).

### **2.1.1 Productor de Energía Virtual**

Los Productores de Energía Virtual (VPP, *Virtual Power Producer*) son entidades que cuentan con diferentes entidades de producción, almacenamiento y consumo que están distribuidas en una pequeña zona. Un VPP encapsula un conjunto de generadores y consumidores, uno de los objetivos de este es poder trabajar de manera independiente a otros VPP's y/o a la red de distribución de

energía. El alcance de los VPP's es que los productores pueden estar seguros que los generadores son óptimamente operados. Los VPP's también trabajan en áreas desoladas. En este tipo de instalación puede tener el manejo del sistema y los equipos bajo su responsabilidad. VPP pueden reducir los costos de operación y mantenimiento además puede incrementar la eficiencia (Morais et al., 2005).

## 2.2 Optimización descentralizada

Los sistemas multiagente y los formalismos de los problemas de optimización con restricciones distribuidos (DCOP, por sus siglas en inglés) ofrecen varios algoritmos asíncronos y óptimos para naturalmente resolver problemas de optimización distribuidos de manera eficiente.

Muchos problemas en sistemas multiagente pueden ser descritos como Problemas de Satisfacción de Restricciones Distribuidas (DCSP, por sus siglas en inglés), donde el objetivo es encontrar una asignación para las variables que satisfagan todas las restricciones de los agentes. En general, los problemas de optimización son mucho más difíciles de resolver que los DCSP, ya que el objetivo es no sólo encontrar una solución, sino la mejor; esto requiere mayor exploración del espacio de búsqueda.

### 2.2.1 Problema de satisfacción de restricciones

En un problema de satisfacción de restricciones (CSP, por sus siglas en inglés), dado un conjunto de variables  $\mathcal{X} = x_1, x_2, \dots, x_m$  con dominio  $\mathcal{D} = D_1, D_2, \dots, D_m$  y un conjunto de restricciones booleanas  $\mathcal{P}$  de la forma  $p_k(x_{k1}, x_{k2}, \dots, x_{kj}) \rightarrow \{0, 1\}$ , el objetivo es encontrar asignaciones para todas las variables que no violen las restricciones.

El problema de satisfacción de restricciones más estudiado es el de *coloreo de un grafo*, ejemplo de este se muestra en la figura 2.2. En este problema dado un grafo y un conjunto de colores se debe colorear cada nodo sin que dos nodos conectados con una arista tengan el mismo color. Este problema se puede transformar a satisfacción de restricciones, donde los nodos son las variables ( $\mathcal{X}$ ) cuyo dominio ( $\mathcal{D}$ ) son el conjunto de colores y las restricciones ( $\mathcal{P}$ ) son que ningún nodo conectado tenga el mismo color.

El problema de satisfacción de restricciones es *NP-Completo*, cualquier algoritmo que se utilice tendrá comportamiento exponencial.

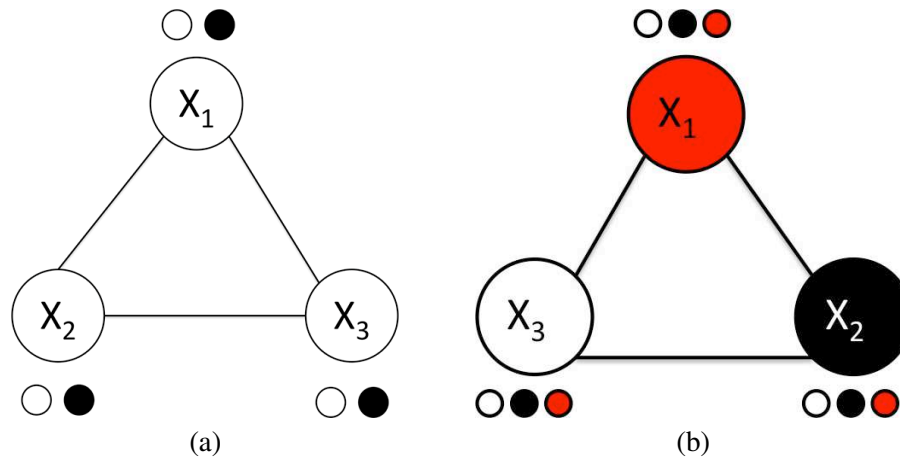


Figura 2.2: Dos ejemplos del problema de coloreo de grafos. En (a) se puede observar que no existe combinación de colores que satisfagan todas las restricciones, esto es debido a que siempre un par de nodos conectados tendrán el mismo color. En (b) se puede observar que existen combinaciones de colores que permiten no violar ninguna restricción.

En el problema de satisfacción de restricciones distribuido (DCSP, por sus siglas en inglés) cada agente posee una variable. Los agentes son responsables de asignar un valor a su variable. Como los agentes no saben los valores asignados a las otras variables, estos tienen que comunicarse con los otros agentes.

### 2.2.2 Grafos factorizados

La ley distributiva en su manera más simple indica que  $ab + ac = a(b + c)$ . El lado izquierdo implica 3 operaciones aritméticas mientras que del lado derecho sólo se necesitan 2.

Un grafo factorizado representa la factorización de una función de varias variables (Loeliger, 2004). Por ejemplo  $f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3) \cdot f_B(x_3, x_4, x_5) \cdot f_C(x_4)$ , que se presenta gráficamente en la figura 2.3. En general, esta representación consiste en nodos, aristas y *medias aristas* (que sólo están conectadas a un nodo), y sigue las siguientes reglas:

- Un nodo por cada factor,
- Una arista o *media arista* por cada variable, y
- Un nodo  $g$  esta conectado con una arista  $x$  si y sólo si la variable  $x$  aparece en el factor  $g$ .

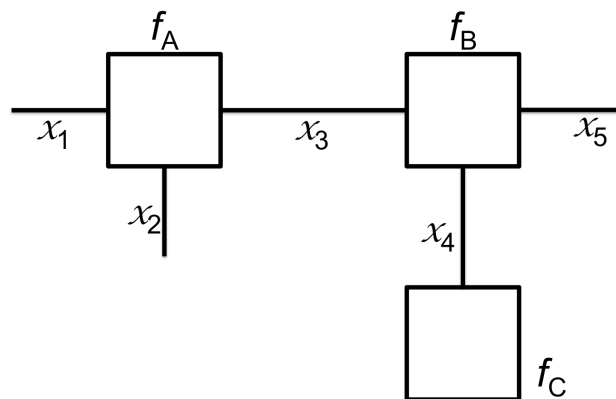


Figura 2.3: Ejemplo de grafo factorizado de la función  $f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3) \cdot f_B(x_3, x_4, x_5) \cdot f_C(x_4)$ .

### 2.2.3 Pseudoárbol

Un grafo es una representación gráfica de objetos y relaciones entre estos. Existen dos tipos de grafos, los dirigidos donde sólo puede existir relación de un objeto al otro pero no en orden inverso. Los grafos no dirigidos son aquellos donde la relación entre dos nodos es bidireccional (Brassard y Bratley, 1996).

Un árbol es un grafo sin ciclos, no dirigido y conectado. Como un árbol es definido como sin dirección en el que sólo existe una trayectoria entre cada par de nodos. Un nodo padre es aquel superior en la jerarquía de un nodo directamente conectado. Los nodos hijos son los nodos directamente conectados inferiores en la jerarquía de un nodo. En los árboles raíz existe un nodo inicial. Las hojas de un árbol son los nodos que no tienen hijos. Los árboles tiene las siguientes propiedades (Brassard y Bratley, 1996):

- Un árbol con  $n$  nodos tiene exactamente  $n - 1$  vértices.
- Si se añade un vértice se crea un ciclo.
- Si se elimina un vértice entonces el grafo se vuelve no conectado.

Un arreglo pseudoárbol de un grafo  $\mathcal{G}$  es un árbol raíz, tiene los mismos vértices que  $\mathcal{G}$  y la propiedad de que los vértices adyacentes del grafo original quedan en la misma rama del árbol. Un ejemplo de un pseudoárbol se muestra en la figura 2.4. Los elementos que componen a un pseudoárbol son los nodos, los vértices y pseudovértices. Para el control de un pseudoárbol se necesita definir lo siguiente (Petcu y Faltings, 2005):

- $P(X)$  - *padre* de un nodo  $X$ . Es el nodo superior en la jerarquía conectado directamente con el nodo  $X$  (e. g.  $P(X_4) = X_1$ ).

- $C(X)$  - *hijo* de un nodo  $X$ . Es el conjunto de nodos que inferiores en la jerarquía conectados directamente con el nodo  $X$  (e. g.  $C(X_1) = X_3, X_4$ )
- $PP(X)$  - *pseudopadre* de un nodo  $X$ . Es el nodo superior en la jerarquía conectado directamente con el nodo  $X$  (e. g.  $PP(X_4) = X_1$ ).
- $PC(X)$  - *hijo* de un nodo  $X$ . Es el conjunto de nodos que inferiores en la jerarquía conectados directamente con el nodo  $X$  (e. g.  $PC(X_1) = X_3, X_4$ )

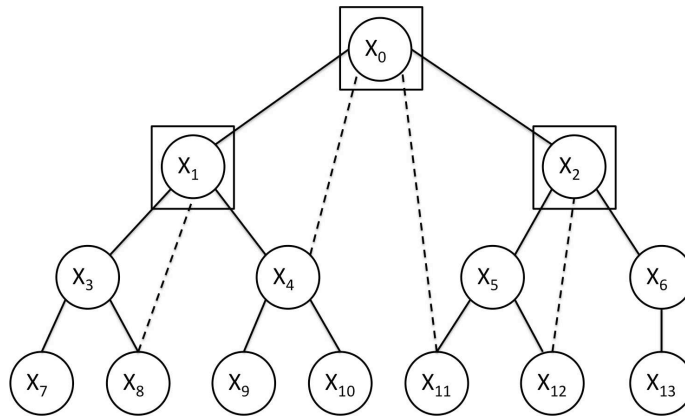


Figura 2.4: Ejemplo de un pseudoárbol. Los círculos representan los nodos, las líneas representan los vértices mientras que las líneas punteadas representan los pseudovértices. Los nodos dentro de cuadrados son los pseudopadres.

### 2.2.4 Problema de optimización con restricciones distribuido (DCOP)

El problema de optimización de restricciones distribuido (DCOP) es similar al problema de satisfacción de restricciones con la diferencia que las restricciones dan como un valor real en lugar de un valor booleano y el objetivo es minimizar el valor de las restricciones violadas. En un DCOP un grupo de agentes distribuidos eligen valores para su conjunto de variables para que el costo de la función objetivo sea minimizado o maximizado.

En un problema de optimización con restricciones (COP, por sus siglas en inglés) dado un conjunto de variables  $\mathcal{X} = x_1, x_2, \dots, x_m$  con dominio  $\mathcal{D} = D_1, D_2, \dots, D_m$  y un conjunto de restricciones  $\mathcal{P}$  de la forma  $p_k(x_{k1}, x_{k2}, \dots, x_{kj}) \rightarrow \mathbb{R}$ , encontrar asignaciones para todas las variables minimizando la suma de los valores de las restricciones.

Al igual que CSP este problema es *NP-Completo*. Esto significa que, en el peor de los casos en todos los algoritmos el tiempo computacional será exponencial (en dependencia al número de



variables).

Existen diferentes tipos de algoritmos DCOP. En cuanto a su demostración teórica de la garantía de cumplimiento de las restricciones: (i) no tienen garantía, (ii) satisfacen restricciones y (iii) optimizan. De cada uno de los tres tipos mencionados anteriormente puede haber algoritmos síncronos y asíncronos. Los algoritmos síncronos permiten la comunicación ordenada entre agentes, ya que la comunicación se realiza de manera secuencial; es decir, cada agente tiene un orden. Los algoritmos asíncronos permiten el paralelismo, sistemas de comunicación robustos y no existe un reloj global.

### Synchronous Branch and Bound

*Synchronous Branch and Bound* (SynchBB) (Hirayama y Yokoo, 1997), es un algoritmo que implementa de manera distribuida el método *Branch and Bound* para los problemas de satisfacción de restricciones donde existe un número permitido de restricciones que se pueden violar (*Max-CSP*). En SynchBB, se fijan los agentes, sus valores (asignaciones de su dominio), un camino y una asignación parcial para todas las variables por adelantado. Los agentes son ordenados según el problema; es decir, este orden no lo realiza el algoritmo. El proceso secuencial es el siguiente:

- El primer agente en el camino ya asignado inicia cambiando su valor al primero contenido en el dominio. Una vez asignado este valor este agente lo envía al segundo en el camino ya asignado.
- Cuando un agente recibe el valor de un agente anterior, el agente evalúa con el primer valor de su dominio y este envía al siguiente agente los valores si esta evaluación es menor que el límite superior actual (la asignación completa de todas las variables de menor valor encontrada hasta el momento), en caso contrario continua probando con los siguientes valores hasta que la evaluación es menor que el límite superior. Si no existe valor menor al límite superior regresa los valores al agente anterior.
- Cuando se reciben los valores de un agente siguiente, este agente cambia su valor al siguiente en su dominio, si el valor no supera el límite superior se lo envía al siguiente agente, de lo contrario sigue evaluando con los siguientes valores del dominio. Si se prueban todos los valores y si ninguno es menor que el límite superior se realiza otro paso atrás en los agentes.

En la figura 2.5 se muestra un ejemplo del funcionamiento del algoritmo *Synchronous Branch and Bound*. Para realizar la evaluación se utiliza la siguiente formula  $Valor(Eólica) * 0.4 + Valor(Fotovoltaica) * 0.4 + Valor(Celdas\ de\ combustible) * 0.9 - Valor(Cargar\ batería) * 0.4 + Valor(Descargar\ batería) * 0.6 + Valor(Energía\ faltante) * 1.5$ . El valor de la energía faltante es la carencia de energía producida con respecto a la carga.

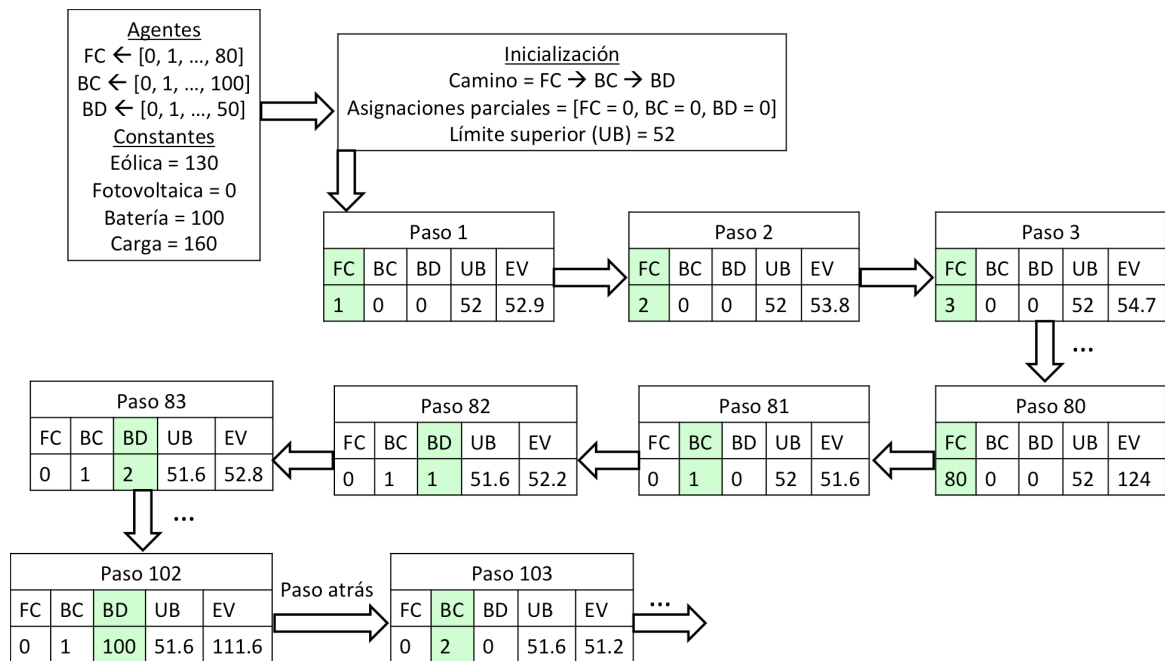


Figura 2.5: Ejemplo del funcionamiento de SychBB. Se muestra algunos pasos del funcionamiento del algoritmo SychBB. El agente FC representa las celdas de combustible, el agente BC representa la carga de la batería y el agente BD representa la descarga de la batería. EV representa la evaluación de la configuración actual de los agentes.

### Distributed Pseudotree-Optimization Procedure

*Distributed Pseudotree-Optimization Procedure* (DPOP) (Petcu y Faltings, 2005), es un método distribuido completo para optimización de restricciones basado en programación dinámica. Es un método de propagación de utilidades inspirado en el algoritmo *sum-product*. Este algoritmo requiere un número lineal de mensajes. El problema está en el requerimiento de memoria, el cual es exponencial en orden al número de nodos del grafo de restricciones. Este algoritmo consta de 3 fases:

- Los agentes establecen la estructura del pseudoárbol (donde los nodos representan las variables y las aristas las restricciones) para ser usado en las siguientes fases.
- La propagación de utilidades comienza desde las hojas del pseudoárbol y se propaga hacia arriba sólo a través de las aristas. En una red de pseudoárbol, el mensaje de utilidad enviado hacia al padre sólo depende del subárbol del nodo y las restricciones entre el nodo y su padre. En las redes con ciclos, el mensaje enviado desde un nodo hacia su padre también depende de las variables arriba de su padre.
- La etapa de propagación de valores comienza cuando el nodo raíz recibe todos los valores de sus hijos. El nodo raíz en base a los mensajes de utilidad, escoge su valor óptimo y envía a

sus hijos un mensaje informado de su decisión. Cuando los hijos van recibiendo los mensajes de sus padres, escogen su valor óptimo y envían mensajes a sus hijos hasta llegar a los nodo hoja.

En la figura 2.6 se muestra la construcción del pseudoárbol (primera etapa del algoritmo DPOP) para los agentes celdas de combustible, carga de batería y descarga de batería. En el grafo con restricciones todos los agentes tienen comunicación con todos por lo que existe un ciclo y al construir el grafo con restricciones se tiene que eliminar.

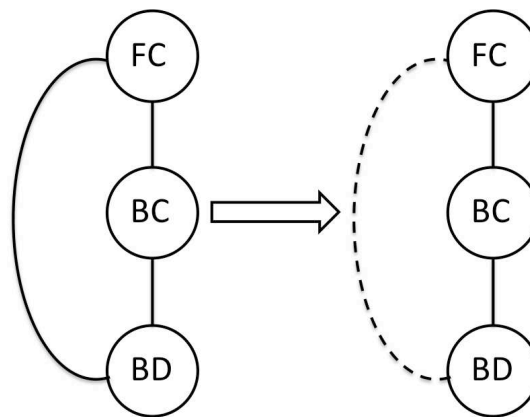


Figura 2.6: Construcción del pseudoárbol. Los nodos representan los agentes. Se muestra el grafo con restricciones donde el agente FC representa las celdas de combustible, el agente BC representa la carga de la batería y el agente BD representa la descarga de la batería. Se toma como nodo raíz a FC, entonces como existe un ciclo con los agentes BD y FC, FC pasa a ser el pseudopadre de BD.

En la figura 2.7 en la tabla (a) se muestra la utilidad generada por las posibles configuraciones de los agentes. Una vez creado el pseudoárbol se comienza la propagación de los mensajes de utilidad desde las hojas de este, en este caso el agente BD (Le et al., 2015). El agente BD comienza a crear los mensajes de utilidad con las posibles configuraciones de los agentes con los que tiene comunicación (FC y BC). Como ejemplo si el agente FC y BC son asignados como 0 se tiene que si asignamos BD 0 la relación de BD con BC vale 5 y BD con FC 5 ( $5 + 5 = 10$ ); ahora BD asignado como 1 se tiene  $BD-BC = 8$  y  $BD-FC = 8$  ( $8 + 8 = 16$ ). Como el máximo valor de que los agentes FC y BC sean asignados es 16 y ese es el valor que se envía a BC, y es lo mismo con las demás configuraciones. Una vez que BD calculó todos sus valores le envía a BC la tabla de configuraciones y el máximo valor de cada una de estas. Ahora el agente BC calcula sus mensajes de utilidad, si FC y BD se asignan como 0 el valor sería 5 y la máxima utilidad de esta configuración obtenida por el agente BD fue 16 por lo tanto es  $5 + 16 = 21$ ; si  $FC = 0$  y  $BC = 1$  entonces la utilidad sería 8 y la máxima ya calculada sería  $25$  por lo tanto  $8 + 25 = 33$ , la máxima sería 33. Por último, estos mensajes de utilidad (los creados por BC y BD) son enviados a FC.

Una vez que todos los mensajes de utilidad llegan al nodo raíz se comienza con la etapa de propagación de valor. El agente FC elige el valor que tiene mayor utilidad que es 1 por que su utilidad es 45, entonces este lo envía a BC que elige 0 debido a que su utilidad es 45 por último, BC le envía su valor a BD que con la asignación  $FC = 1$  y  $BC = 0$  elige 1 debido a que ahí es donde tiene su mayor utilidad 25.

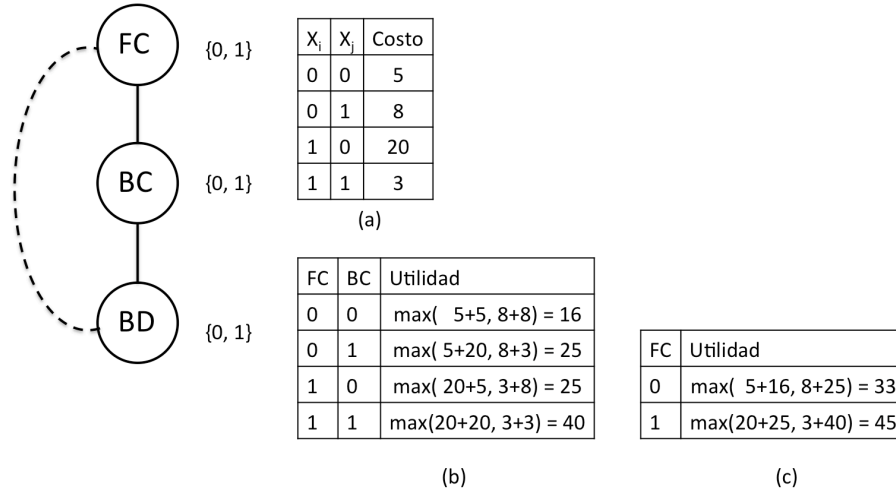


Figura 2.7: Ejemplo de ejecución de DPOP. En la tabla (a) se muestran las utilidades de las posibles configuraciones de los agentes. La tabla (b) muestra los mensajes creados por el agente BD y lo envía al agente BD. Cuando el agente BD recibe los mensajes de BD y crea los mensajes que envía a FC mostrados en la tabla (c).

### Memory Bounded - Distributed Pseudotree-Optimization Procedure

*Memory Bounded - Distributed Pseudotree-Optimization Procedure* (MB-DPOP( $k$ )) (Petcu y Faltings, 2007), es un algoritmo basado en DPOP. Es un algoritmo híbrido que puede operar con memoria limitada. Cuando el árbol tiene poca amplitud funciona igual que DPOP. Cuando existe gran amplitud, se explora con propagaciones limitadas usando la idea de *cortar ciclos*.

MB-DPOP( $k$ ), introduce un nuevo parámetro de control  $k$  que especifica la máxima dimensión de los mensajes. El algoritmo identifica los subgrafos que tienen una amplitud mayor a  $k$ . Cada área (grupo) son limitadas al máximo por el nodo más bajo en el árbol que tiene separador de tamaño  $k$ . Llamamos a estos nodos grupo, grupos raíz (CR, por sus siglas en inglés).

En la etapa de propagación de valores, para los hijos que tengan mayor amplitud (separación) que  $k$ , se crea una tabla de *Caché* que corresponde a cada instancia particular de cada separación. Se comienza a explorar a través de  $k$ -delimitada propagación sobre todos los subárboles que tienen ciclos con sus padres. Cada uno de los subárboles envían un mensaje de *contexto* a sus hijos. Los

nodos hijos propagan los mismos mensajes a sus hijos que tienen ciclos con sus padres, esto se realiza para todos los nodos que tienen una separación mayor a  $k$ .

En la figura 2.8 se muestra un ejemplo de un pseudoárbol agrupado por MB-DPOP. Si se toma  $k = 2$ ; entonces  $C_2 = \{X_9, X_{10}, X_{11}, X_{12}, X_{13}\}$  es un área con  $k$  mayor a 2.  $X_9$  es la raíz de esta agrupación. Estas agrupaciones son elegidas de abajo hacia arriba. Por ejemplo para crear  $C_3$  se tiene que  $X_{11}$  tiene  $k = 4$  y se pueden eliminar dos de esas relaciones al aislar  $X_{14}$  y  $X_{17}$  por lo que se crea un grupo que contenga esos elementos. El objetivo de crear grupos es que se pueda ejecutar de manera independiente el paso de mensajes en esa área y entonces cuando se tenga mandar los resultados al grupo o nodo padre.

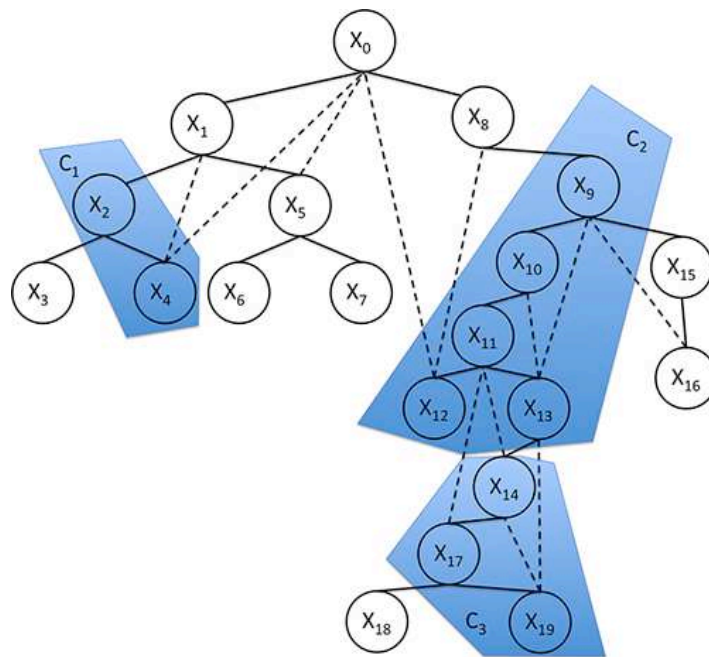


Figura 2.8: Ejemplo de un pseudoárbol agrupado por MB-DPOP.

### Asynchronous Forward Bounding

*Asynchronous Forward Bounding* (AFB) (Gershman et al., 2009), es un algoritmo de búsqueda distribuido para DisCSP. En el algoritmo AFB los agentes se asignan valores a sus variables y generan una solución parcial secuencial, esta solución parcial es propagada de manera asíncrona. La propagación de la solución parcial permite una asíncrona actualización de los límites en los costos para una temprana detección de cuando se tiene que dar marcha hacia atrás.

El algoritmo AFB está basado en el esquema *Branch and Bound*. En el proceso de búsqueda del algoritmo AFB es representado por una estructura de datos llamada Asignación Parcial Actual (CPA, por sus siglas en inglés). La CPA es inicialmente vacía, hasta que algún agente la inicializa y

la envía al siguiente agente. El costo de la CPA es la suma de los costos de las restricciones que se incluyen. Además del costo de asignación actual, los agentes mantienen en la CPA un límite inferior que es actualizado de acuerdo a la información recibida de los agentes que todavía no han sido asignados. Cada agente que recibe la CPA añade las asignaciones a sus variables, si la asignación con límite menor es menor que el actual global, el límite superior puede ser encontrado. En caso contrario, realiza marcha hacia atrás.

En la figura 2.9 se muestra un ejemplo de un grafo con restricciones que se usará para ilustrar como funciona el algoritmo AFB (Gershman et al., 2009). Para comenzar FC crea una CPA vacía y se asigna el primer valor en su dominio (0) y se la envía a BC. BC asigna el primer valor en su dominio (0) y finalmente se la envía a BD que hace lo mismo. Ahora se tiene una asignación completa y el valor es 6. Ahora que se tiene un costo este se envía a todos los agentes para actualizar su costo total. Ahora BD realiza cambia su valor a 1, como ahora el valor es 4 se vuelve a enviar el valor a todos los agentes para que actualicen su límite superior. Como BD ya no tiene más valores en su dominio da un paso hacia atrás y envía la CPA a BC. Ahora BC reasigna su valor y la evaluación da como resultado 5. Entonces este envía la CPA a BD que cambia su asignación a 0 y su evaluación da 4. Como esta asignación es igual que el límite superior se da un paso atrás y BC abandona su asignación y queda en 0 cuya evaluación es 4 y BC regresa la CPA a FC. FC asigna su siguiente valor y su evaluación da 5 por lo que envía la CPA a BC. Después de unos pasos se dan cuenta que el mejor es valor es 4 con la configuración  $FC = 0$ ,  $BC = 0$  y  $BD = 1$ .

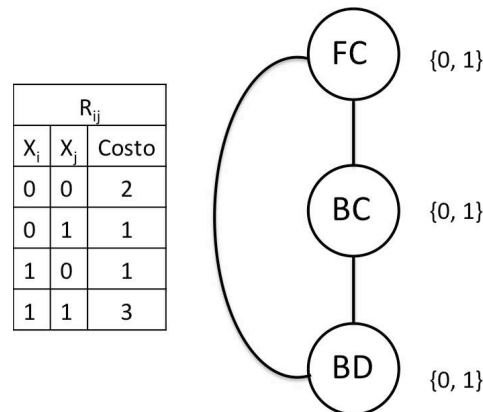


Figura 2.9: Se muestra el grafo con restricciones donde el agente FC representa las celdas de combustible, el agente BC representa la carga de la batería y el agente BD representa la descarga de la batería. El dominio de las variables puede ser 0 o 1. Las restricciones se muestran en la tabla.

## 2.3 FRODO

FRODO es un marco de trabajo de código abierto realizado en Java para la optimización combinatoria distribuida. En este trabajo de tesis FRODO sirve de base para la experimentación al contar con implementaciones de los algoritmos de optimización distribuida más conocidos. Fue desarrollado inicialmente en el Laboratorio de Inteligencia Artificial de la Escuela Politécnica Federal de Lausana, Suiza (Léauté et al., 2009). Actualmente soporta SynchBB, MGM, MGM-2, ADOPT, DSA, DPOP, S-DPOP, MPC-Dis(W)CSP4, O-DPOP, AFB, DPOP, Max-Sum, ASO-DPOP, P-DPOP, P2-DPOP, E[DPOP], Param-DPOP y  $P^{\frac{3}{2}}$ .

FRODO tiene una arquitectura multicapa que se muestra en la figura 2.10.

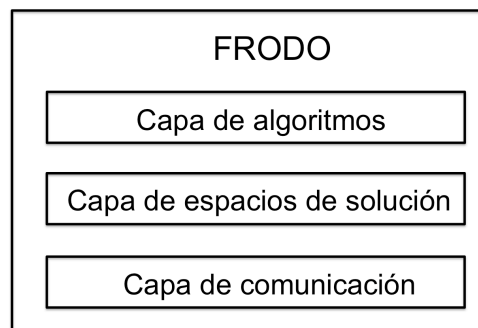


Figura 2.10: Arquitectura general de FRODO.

La capa de comunicaciones es responsable de pasar mensajes entre agentes. En el núcleo de esta capa es la clase *Queue*, que es una implementación de una cola de mensajes. La capa de espacios de solución proporciona clases que se pueden utilizar para modelar problemas de optimización combinatoria. Dado un espacio de posibles asignaciones a algunas variables, un espacio de soluciones es una representación de las tareas de interés especial, tales como las asignaciones que corresponden a soluciones de un problema dado. La capa de algoritmos se basa en la capa de espacios de soluciones y la capa de comunicación a fin de proporcionar algoritmos distribuidos para resolver DCOPs.

## 2.4 Búsqueda A\*

El algoritmo A\* (Hart et al., 1968) utiliza la *búsqueda de primero el mejor* para encontrar la ruta de menor costo desde un nodo inicial al nodo objetivo. Como A\* está diseñado para recorrer un grafo, este utiliza una cola de prioridad, donde ordena los nodos por costo (de menor a mayor).

Este algoritmo utiliza una función de costo ( $f(x)$ ) para determinar el orden de visita de los nodos en el árbol. La función de costo que tiene dos partes: (i)  $g(x)$  que es el costo del recorrido hasta el momento y (ii)  $h(x)$  que denota un estimado del futuro costo, este se estima con una heurística ad-

misible, donde una heurística admisible es aquella que no debe sobreestimar la distancia al objetivo.

El algoritmo  $A^*$  es un algoritmo de búsqueda que requiere un punto inicial y un punto final (objetivo). En nuestro caso no utilizamos un punto objetivo, se utiliza un *punto de referencia*, es decir, al encontrar una configuración de menor valor esta pasa a ser nuestro *punto de referencia*. Como estimador se utiliza la función objetivo del problema para una sola configuración (función de costo). En nuestro caso, utilizamos como punto inicial y *punto de referencia* la mejor configuración encontrada por el algoritmo *primero el mejor* en conjunto con el algoritmo DCOP. Ya determinados el punto inicial y el de referencia, se analizan todos los vecinos y entran a la lista por analizar todos aquellos nodos que cumplan la condición de ser mejores (menor costo) que el punto actual. Por lo demás, se sigue el mismo proceso que el algoritmo  $A^*$ . Como salida da un punto (configuración) que se toma como la mejor configuración. En la sección 5.3 se muestra un ejemplo del funcionamiento de esta búsqueda.

## 2.5 Resumen

En este capítulo se dieron las bases que enmarcan el desarrollo de este documento de tesis. Se introdujeron los conceptos de microrred y sus principales componentes así como que es un productor de energía virtual. Se dieron las bases de optimización descentralizada. Como base para la optimización descentralizada se explicó que es un problema de satisfacción de restricciones, un grafo factorizado y un problema de optimización con restricciones distribuido. Se explicaron los algoritmos que se utilizaron para realizar la experimentación de esta tesis así como el marco de trabajo que se utilizó para la experimentación. En el siguiente capítulo se mostrará la formalización matemática para el problema de ciclos de encendido y apagado en un conjunto de microgeneradores.



# 3

## Formalización

En este capítulo se dará la formulación matemática del problema de calendarización de ciclos de encendido y apagado descrito secciones anteriores.

### 3.1 Formulación matemática del problema

Esta formulación se encuentra en (Morais et al., 2010). Para la formulación del problema se toma el caso real del sistema de *Budapest Tech* y se crea una microrred que opera como una red de manera aislada.

Para encontrar la solución óptima al problema de calendarización que se presentará es necesario contar con información de los elementos que integran a la microrred: la cantidad de energía generada de manera eólica, fotovoltaica, por celdas de combustible y la velocidad de carga y descarga de la batería, el problema toma en consideración lo siguiente:

- La microrred es considerada un productor de energía virtual debido a que opera de manera independiente (ver sección 2.1.1).
- La generación eólica es dependiente del clima. Se cuenta con las predicciones de generación eólica de 24 horas al futuro.
- También se cuenta con las predicciones de generación fotovoltaica (24 horas al futuro), las predicciones son la entrada al problema de optimización.
- Las celdas de combustible tienen una producción limitada a largo plazo, el total de energía generada es determinada por la cantidad de hidrógeno y la capacidad de la máquina.
- La descarga de la unidad de almacenamiento está limitada por una velocidad máxima de descarga y la energía contenida.

- Para balance del sistema, el productor de energía virtual puede establecer términos de reserva (energía que se almacena).

Como ya se explicó en la sección 1.1, se cuenta con un conjunto de 3 generadores de energía renovable (turbina de viento, generador solar y celda de combustible) en donde la producción de 2 de ellos (turbina de viento y generador solar) son dependientes del clima, es decir, su producción no se puede controlar. Además de los generadores se cuenta con un dispositivo con capacidad de almacenamiento (batería), esta tiene una capacidad máxima de almacenamiento así como una capacidad máxima de carga y descarga. La formulación matemática se describe en las siguientes secciones, los costos, límites de producción y condiciones iniciales son obtenidos de (Morais et al., 2010).

### 3.1.1 Función objetivo

El objetivo es minimizar el costo de producción de energía a lo largo de 24 horas (t) mientras se mantiene el balance en la red considerando los siguientes elementos:

1. La producción de energía eólica con su respectivo costo,  $P_{Eólica(t)}$  y  $C_{Eólica}$ .
2. La producción de energía fotovoltaica tomando en cuenta su costo,  $P_{Fotovoltaica(t)}$  y  $C_{Fotovoltaica}$ .
3. La producción de energía por medio de celdas de combustible tomando en cuenta su respectivo costo,  $P_{CeldaCombustible(t)}$  y  $C_{CeldaCombustible}$ .
4. La energía que se puede cargar a la batería con su respectivo costo (consumo),  $P_{CargarBatería(t)}$  y  $C_{CargarBatería}$ .
5. La energía que se puede descargar de la batería que también tiene costo (producción),  $P_{DescargarBatería(t)}$  y  $C_{DescargarBatería}$ .
6. La energía que falta para satisfacer la demanda con su respectivo costo,  $P_{EnergíaFaltante(t)}$  y  $C_{EnergíaFaltante}$ .
7. La energía que se esta produciendo y que sobra después de atender la demanda, esta no tiene costo,  $P_{EnergíaExcedida(t)}$  y  $C_{EnergíaExcedida}$ .

Se muestra la función objetivo en 3.1.  $P$  hace referencia a la producción o cantidad de energía y  $C$  hace referencia al costo de la producción de cada una de las fuentes de energía. El sufijo representa el tipo de energía. El costo de producción de energía eólica y fotovoltaica es de 0.4 por Watt/hora (W/h). El costo de producción de energía de las celdas de combustible es 0.9 por W/h. El costo de cargar la batería es 0.4 por W/h. El costo de descargar la batería es 0.6 por W/h. El costo de la energía faltante es 1.5 por W/h. Por último, el costo del exceso de energía es de 0.0 por W/h.

$$\text{Minimizar } f = \min \sum_{t=1}^{24} \left( \begin{array}{l} P_{Eólica(t)} * C_{Eólica} \\ + P_{Fotovoltaica(t)} * C_{Fotovoltaica} \\ + P_{CeldaCombustible(t)} * C_{CeldaCombustible} \\ - P_{CargarBatería(t)} * C_{CargarBatería} \\ + P_{DescargarBatería(t)} * C_{DescargarBatería} \\ + P_{EnergíaFaltante(t)} * C_{EnergíaFaltante} \\ - P_{EnergíaExcedida(t)} * C_{EnergíaExcedida} \end{array} \right) \quad (3.1)$$

### 3.1.2 Restricciones

En 3.2 se muestra la primera ley de Kirchhoff o balance de energía. Esto quiere decir que se debe mantener un balance entre lo producido y lo consumido. Aquí se introduce una nueva variable *Carga* que hace referencia al consumo de energía. La *carga* de la microrred es el consumo de energía. La variable  $P_{EnergíaFaltante(t)}$  se considera como variable de producción debido a que esa energía no se esta entregando a la red pero es la energía faltante para satisfacer la demanda, esta tiene un costo alto para evitar que esto pase. La variable *Energía Excedida* se toma en cuenta como consumo, esto debido a que esta energía ya se produjo, esta no tiene costo debido a que es mejor no producirla.

$$\left( \begin{array}{l} P_{Eólica(t)} + P_{Fotovoltaica(t)} + P_{CeldaCombustible(t)} \\ + P_{DescargarBatería(t)} + P_{EnergíaFaltante(t)} \\ = Carga_{(t)} + P_{CargarBatería(t)} + P_{EnergíaExcedida(t)} \end{array} \right); t \in \{1, 2, \dots, 24\} \quad (3.2)$$

Se muestra la restricción del límite de producción de energía eólica en la ecuación 3.3. Esta restricción esta ligada al equipo que se use, en este caso el límite es 400 W.

$$P_{Eólica(t)} \leq P_{LímiteEólica}; t \in \{1, 2, \dots, 24\} \quad (3.3)$$

Se expone la restricción del límite de producción de energía fotovoltaica en la ecuación 3.4. Esta restricción esta ligada al equipo que se use, en este caso el límite es 150 W.

$$P_{Fotovoltaica(t)} \leq P_{LímiteFotovoltaica}; t \in \{1, 2, \dots, 24\} \quad (3.4)$$

En la ecuación 3.5 se indica la restricción del límite de producción de energía por medio de las celdas de combustible. Esta restricción esta ligada al equipo que se use, en este caso el límite es 80 W.

$$P_{CeldaCombustible(t)} \leq P_{LímiteCeldaCombustible}; t \in \{1, 2, \dots, 24\} \quad (3.5)$$

Se muestra la restricción del límite de carga de la unidad de almacenamiento (batería) en la ecuación 3.6. Esta restricción esta ligada al equipo que se use, en este caso el límite es 200 W.

$$P_{Batería(t)} \leq P_{LímiteBatería}; t \in \{1, 2, \dots, 24\} \quad (3.6)$$

En la ecuación 3.7 se exhibe la restricción de lo máximo que se puede descargar de la batería en cada tiempo. Esto puede oscilar entre 0 y 50, incluidos ambos. Se introduce la variable  $X$  que es una variable de control, si es 0 significa que no se esta descargando la batería y 1 significa que se esta descargando.

$$P_{DescargarBatería(t)} \leq P_{LímiteDescargarBatería} * X(t); t \in \{1, 2, \dots, 24\}, X(t) \in \{0, 1\} \quad (3.7)$$

Se muestra la restricción de lo máximo que se puede cargar de la batería en cada tiempo 3.8. Esto puede oscilar entro 0 y 200, incluidos ambos. Aquí se introduce la variable  $Y$  que es una variable de control, si es 0 significa que no se esta cargando la batería y 1 significa que se esta cargando.

$$P_{CargarBatería(t)} \leq P_{LímiteCargarBatería} * Y(t); t \in \{1, 2, \dots, 24\}, Y(t) \in \{0, 1\} \quad (3.8)$$

Se expone la restricción que se asegura de que la batería no se cargue y descargue en un mismo tiempo en 3.9. Las variables aquí mostradas son las de control de las restricciones de las ecuaciones 3.7 y 3.8. En la restricción es menor o igual que 1 para evitar que ambas variables de control estén activadas al mismo tiempo.

$$X(t) + Y(t) \leq 1; t \in \{1, 2, \dots, 24\} \quad (3.9)$$

Se muestra la restricción para asegurar que no se descargue la batería más de la energía que tiene en la ecuación 3.10.

$$P_{DescargarBatería(t)} - P_{Batería(t-1)} \leq 0; t \in \{1, 2, \dots, 24\} \quad (3.10)$$

En la ecuación 3.11 se muestra la restricción para asegurar que la carga de la batería en cada tiempo no exceda su límite.

$$P_{CargarBatería(t)} + P_{Batería(t-1)} \leq 200; t \in \{1, 2, \dots, 24\} \quad (3.11)$$

Se muestra la restricción que actualiza el estado de la batería en cada tiempo en la ecuación 3.12.

$$\begin{aligned} P_{Batería(t)} &= P_{Batería(t-1)} \\ &- P_{DescargarBatería(t)} + P_{CargarBatería(t)}; t \in \{1, 2, \dots, 24\} \end{aligned} \quad (3.12)$$

Para los escenarios se considera como carga inicial de la batería 100 W, esto se indica en la condición inicial 3.13.

$$P_{Batería(t=0)} = 100W \quad (3.13)$$

## 3.2 Escenarios

En esta sección se muestran los escenarios de producción y consumo propuestos en (Morais et al., 2010). En estos se muestran las predicciones de producción de los generadores eólico y fotovoltaico así como del consumo (carga). En las predicciones de las cargas se tienen dos escenarios. Esto se muestra en la tabla 3.1.

Tabla 3.1: Predicciones de generadores y escenarios de consumo obtenidos de Morais et al. 2010.

Tiempo (horas)	Eólica (W/h)	Fotovoltaica (W/h)	Carga (W/h) [Esc 1]	Carga (W/h) [Esc 2]
1	130	0	160	230
2	150	0	140	190
3	140	0	150	160
4	160	0	120	150
5	100	0	110	140
6	120	5	100	180
7	150	10	170	230
8	180	30	180	250
9	170	60	200	270
10	160	100	220	280
11	120	130	230	240
12	130	140	240	300
13	150	150	240	300
14	176	140	230	280
15	185	130	220	270
16	120	100	210	280
17	130	60	210	290
18	140	30	220	300
19	170	10	230	310
20	190	5	240	320
21	120	0	250	330
22	170	0	200	300
23	130	0	190	280
24	150	0	180	270

La actualización de carga de la batería se modela con la ecuación 3.14 que indica la cantidad de energía. Esta fórmula se utiliza para calcular la cantidad de energía en la batería en cada tiempo

ya que dicho estado depende de estado inmediato anterior.

$$\begin{aligned}
 P_{Batería(t)} &= x \\
 P_{Batería(t+1)} &= P_{Batería(t)} + P_{CargarBatería(t+1)} - P_{DescargarBatería(t+1)} \\
 P_{Batería(t+2)} &= P_{Batería(t+1)} + P_{CargarBatería(t+2)} - P_{DescargarBatería(t+2)}
 \end{aligned} \tag{3.14}$$

### 3.3 Solución centralizada

En esta sección se describe la metodología para realizar la optimización de manera centralizada.

#### 3.3.1 Solución de Morais et al. 2010

El objetivo de la formulación dada en 3.1 es minimizar el costo de producción a un período de 24 horas tomando en cuenta todas las consideraciones mencionadas.

El resultado esperado debe tener las siguientes prioridades: eólica, fotovoltaica y celdas de combustible (si es necesario). Las prioridades se ha establecido así debido al costo de producción de cada generador. El exceso de energía es usado para cargar la unidad de almacenamiento (batería).

Este problema de calendarización de forma centralizada de ciclos de encendido/apagado en un conjunto de microgeneradores es resuelto mediante *ILOG CPLEX* usando la plataforma del Sistema de Modelado Algebraico en General (GAMS, por sus siglas en inglés) (Morais et al., 2010).

GAMS es un sistema de modelado matemático y optimización de alto nivel. Este se compone de un compilador de lenguaje y un conjunto de solucionadores de alto rendimiento. GAMS esta adaptado para modelado de aplicaciones a gran nivel, complejos y permite construir modelos de gran tamaño de rápida adaptación a nuevas situaciones (GAMS Development Corporation, 2014).

La calendarización óptima de ciclos de encendido/apagado en un conjunto de microgeneradores puede ser realizada para diferentes escalas de tiempo. Por ejemplo 5 minutos, 1 hora, 1 día o 1 semana.

El problema de restricciones fue elaborado tomando en consideración cinco diferentes modos de operación:

1. Si existe exceso de energía, esta se puede almacenar.
2. Si los generadores primarios (eólico, fotovoltaico) no son suficientes entonces la batería es descargada.

3. En caso de que no estén en funcionamiento los generadores primarios (no hay viento y/o sol), la batería y las celdas de combustible entran en operación.
4. En caso de que no haya energía almacenada sólo las celdas de combustible entran en operación.
5. La función objetivo del modelo lineal de enteros mixtos es encontrar el costo marginal mínimo total para un periodo de 24 horas.

### 3.3.2 Solución replicada

Se ha implementado la formulación matemática dada en este capítulo con el fin de tomar esta solución como línea base. La solución centralizada encontrada en esta propuesta es óptima y es por eso que la tomamos en este trabajo como referencia.

ILOG CPLEX es una herramienta para la optimización de problemas lineales (LP), de la forma:

$$\begin{array}{ll}
 \textit{Minimizar/Maximizar} & c_1x_1 + c_2x_2 + \dots + c_nx_n \\
 \textit{sujeto a} & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \sim b_1 \\
 & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \sim b_2 \\
 & \dots \\
 & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \sim b_m \\
 \textit{con los siguientes límites} & l_1 \leq x_1 \leq u_1 \\
 & l_2 \leq x_2 \leq u_2 \\
 & \dots \\
 & l_n \leq x_n \leq u_n
 \end{array}$$

Donde  $\sim$  puede ser  $\leq$ ,  $\geq$  o  $=$ . Los límites inferiores  $l_i$  y los límites superiores  $u_i$  pueden ser cualquier número real o incluso infinito positivo o negativo. Los elementos que se deben proveer al problema lineal son:

$$\begin{array}{ll}
 \textit{coeficientes de la función objetivo} & c_1, c_2, c_n \\
 \textit{coeficientes de las restricciones} & a_{11}, a_{12}, a_{1n} \\
 & a_{21}, a_{22}, a_{2n} \\
 & \dots \\
 & a_{m1}, a_{m2}, a_{mn} \\
 \textit{los valores del lado derecho} & b_1, b_2, \dots, b_m \\
 \textit{límites inferiores y superiores} & l_1, l_2, \dots, l_n, \textit{ y } u_1, u_2, \dots, u_n
 \end{array}$$

La solución óptima computada por ILOG CPLEX será devuelta en como valores de las variables:  $x_1, x_2, \dots, x_n$ .

La metodología que se siguió para esta implementación se enlista a continuación:

1. **Función de optimización base:** Se implementó la función de optimización para un caso básico en IBM CPLEX.
2. **Implementación en Java:** Implementación en el lenguaje de programación Java con el objetivo de poder usarla para todos los escenarios.
3. **Implementación para un escenario base:** En este punto se implemento la función base del punto 1 mediante la implementación del programa escrito en Java para un escenario base completo.
4. **Implementación para segundo escenario:** En este punto se creo la configuración para poder realizar la implementación para ambos escenarios propuestos.
5. **Comparación de resultados:** En este punto se compararon los resultados obtenidos con los mostrados en (Morais et al., 2010).

Al tener un esquema centralizado se tiene como desventaja la escalabilidad debido a que el problema combinatorio es un problema NP-Completo por lo que programación lineal de enteros mixtos también es NP-Completa. Se presenta un esquema descentralizado para pueda ser usado por uno o más algoritmos de la familia DCOP para optimizar el uso de un conjunto de microgeneradores a largo plazo obteniendo una buena configuración (de bajo costo), con bajo costo computacional, escalable y tolerante a fallos; cuya solución se aproxime o incluso iguale a la solución de los algoritmos centralizados.

### 3.4 Resumen

En este capítulo se mostró la formulación matemática al problema de calendarización de ciclos de encendido y apagado en un conjunto de microgeneradores. Se expuso la función así como las restricciones del problema. Además se indicaron los escenarios de prueba utilizados en la experimentación. Por último, se presentó la metodología utilizada en el artículo (Morais et al., 2010) para la resolución del problema antes mencionado. En el siguiente capítulo se mostrará el trabajo relacionado al presentado en este documento de tesis.



# 4

## Trabajo relacionado

En este capítulo se muestra la investigación que tiene relación con el presente trabajo de tesis. Después se aportan los trabajos en la literatura con respecto a algoritmos de optimización descentralizada. En seguida está la literatura relacionada a el problema de calendarización pero enfocada a otros campos de aplicación. Por último, se presenta un resumen del trabajo relacionado.

La industria eléctrica está cambiando su estructura tradicional (centralizada) a distribuida. Los cambios que se están realizando están enfocados principalmente a dar soporte a fuentes renovables. Las fuentes renovables se están abriendo paso debido a que emiten muy pocos contaminantes al ambiente en comparación a las plantas de energía convencionales. Los problemas que enfrentan los generadores distribuidos son (Bertani et al., 2004; Hatziaargyriou y Meliopoulos, 2002; Lasseter, 2007; Milosevic y Andersson, 2005):

1. administración integrada de diferentes generadores,
2. adopción de tecnologías de comunicación y
3. la necesidad de reglas de interconexión.

El problema de optimización de calendarización de los ciclos de encendido y apagado de microgeneradores de energía se puede atacar desde dos maneras, centralizada y descentralizada (Hiremath et al., 2007; Morais et al., 2010). De manera centralizada se tiene la ventaja de que un ente conoce todo el sistema y su funcionamiento. Al atacar el problema de manera centralizada se tiene como desventaja que es un problema exponencial, es decir, que no se pueden agregar muchos microgeneradores al sistema debido a que se vuelve intratable. Al optimizar de manera descentralizada se tiene como ventaja que se pueden agregar muchos generadores y el procesamiento se distribuye entre todos. La desventaja que tiene este enfoque es que no se tiene una visión completa del sistema (Hiremath et al., 2007; Nishimura et al., 1993).

## 4.1 Optimización de despacho centralizado en el área de energía

En (Morais et al., 2010) resuelven el problema de calendarización de ciclos de encendido y apagado de manera centralizada. Utilizan 3 diferentes microgeneradores (turbinas de viento, generadores solares y celdas de combustible) y una batería. El problema de optimización se realiza mediante el Sistema General de Modelado Algebraico (General Algebraic Modeling Systems, GAMS) utilizando programación lineal de enteros mixtos. *GAMS* permite escribir un problema de optimización y exportarlo a diferentes plataformas de optimización, en este caso utilizan *ILOG CPLEX*. El problema es aplicado a un caso real. Otro punto importante a recalcar es que una vez realizado el programa de optimización para agregar un nuevo elemento es necesario volver a generar el programa de optimización. Esta es de las primeras implementaciones de calendarización de ciclos de encendido y apagado de microgeneradores. El objetivo es satisfacer la demanda reduciendo el costo de producción por medio de la calendarización de los microgeneradores. La ventaja que tiene es que se tienen soluciones exactas. Debido a que el problema combinatorio es un problema NP-Completo por lo que programación lineal de enteros mixtos también es NP-Completa.

En (Su et al., 2014) se formula un problema estocástico para calendarizar la energía en una microrred, minimizando el costo de operación. Primero describen el problema estocástico de dos etapas de administración de energía (calendarización a un día adelante y en tiempo real de acuerdo a los errores en las predicciones). Un modelo típico de una microrred se ejecuta en dos modos de operación: conectada a la red de distribución y aislada. Proponen un modelo para poder ser resuelto por medio del planteamiento de *problema de programación de enteros mixtos no lineales*. El modelo de trabajo se muestra en la figura 4.1. La desventaja que posee este enfoque es que al ser una formulación programación de enteros mixtos no lineales este problema es NP-Completo.

En (Chaouachi et al., 2013) optimizan la calendarización de una microrred mediante el uso de técnicas de inteligencia artificial. El objetivo principal es calendarizar el uso de las baterías. Además de realizar una calendarización óptima de las baterías utilizan técnicas para la predicción de generadores solares y eólicos. Las principales aportaciones de este trabajo están en los predictores que se utilizan. La principal limitación se encuentra en que sólo optimizan el uso de la batería.

Se presentan una metodología multiobjetivo en (Bazmohammadi et al., 2012) cuyo objetivos son: (i) optimizar la operación de la microrred y (ii) calendarizar el uso de la batería para reducir los costos a la hora de comprar energía. La metodología esta basada en Optimización en Enjambre de Partículas MultiObjetivo, que minimiza el costo de operación de la red tomando en cuenta las emisiones de contaminantes al ambiente. En este trabajo sólo se reduce el costo de compra de energía mientras que se reducen las emisiones de contaminantes al medio ambiente.

En (Wang et al., 2012) se presenta un modelo centralizado estocástico para operar los generadores eólicos de una red. Optimizan el uso de los generadores eólicos dándoles un valor de riesgo a estos, esto para no darle uso excesivo. Existe una buena formulación matemática del modelo la des-

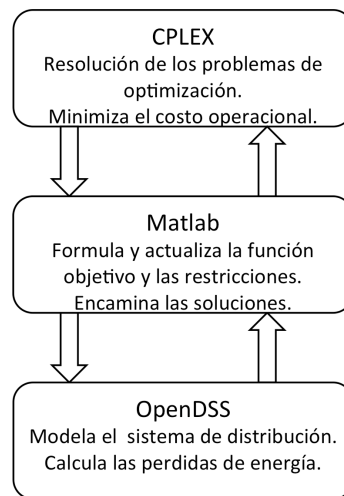


Figura 4.1: Marco de trabajo. Se marcan en cuadros los diferentes componentes de trabajo como ILOG CPLEX optimizador, Matlab que es la plataforma que ayuda a actualizar las restricciones en tiempo real y OpenDSS que permite modelar el sistema de distribución.

ventaja es que se enfoca principalmente el factor de riesgo de uso excesivo de los generadores eólicos.

En (Chen et al., 2011) presentan un sistema inteligente centralizado de manejo que optimiza la operación de una microrred. El sistema contiene módulos para cada una de las funciones: un módulo de predicción tanto para los generadores distribuidos como para la carga de la microrred, un módulo de manejo para para los dispositivos de almacenamiento y un módulo de optimización que no coordina generadores si no que se encarga de decidir cuando usar baterías y cuando comprar energía de la red inteligente con tal de minimizar los costos de operación. Este sistema optimiza la operación solamente a una hora. La principales aportaciones de este trabajo son la formulación para la administración de los componentes de la red y los predictores para los generadores renovables. La principal limitación se encuentra en que la formulación matemática únicamente esta para una hora.

En (Chen y Gooi, 2011) presentan un método multiobjetivo que: (i) minimiza el costo total de una sistema de energía inteligente y minimiza las perdidas de energía en la transmisión, (ii) realiza la calendarización de producción de energía de los generadores tradicionales, (iii) determina la carga y descarga para los dispositivos de almacenamiento, (iv) determina la energía transferida entre los dispositivos de almacenamiento y el sistema de energía inteligente y (v) determina la energía reactiva inyectada al sistema por los sistemas de energía renovables. Ellos utilizan un método centralizado implementado en *A Modeling Language for Mathematical Programming* y resuelto mediante *KNITRO*. La solución propuesta consta de subprogramas de calendarización y para resolución del problema de flujo en la red. La resolución se da hora por hora, es decir, primero resuelven la hora uno, después la hora dos así hasta llegar a la última hora, el único caso en que vuelven a

calendarizar es cuando las pérdidas en la red superan su límite definido.

En (Gil y Lopez, 2008) se presenta un esquema jerárquico para el control de múltiples microrredes. El objetivo es el control de energía inyectada al sistema. A pesar de usar un enfoque por niveles sigue siendo centralizado. Se utilizan 3 niveles de control. En el primer nivel se encuentra el Sistema de Gestión de la Distribución. En el segundo nivel se encuentra el Controlador Central de Administración Autónoma. En el tercer nivel se encuentran Controlador Central de la MicroRed, la demanda, la Generación Distribuida, Controladores de Demanda y los Controladores de Microrecursos. El esquema jerárquico presentado se muestra en la figura 4.2. Este modelo no se encarga de calendarizar el uso de los generadores sólo de la distribución de la energía.

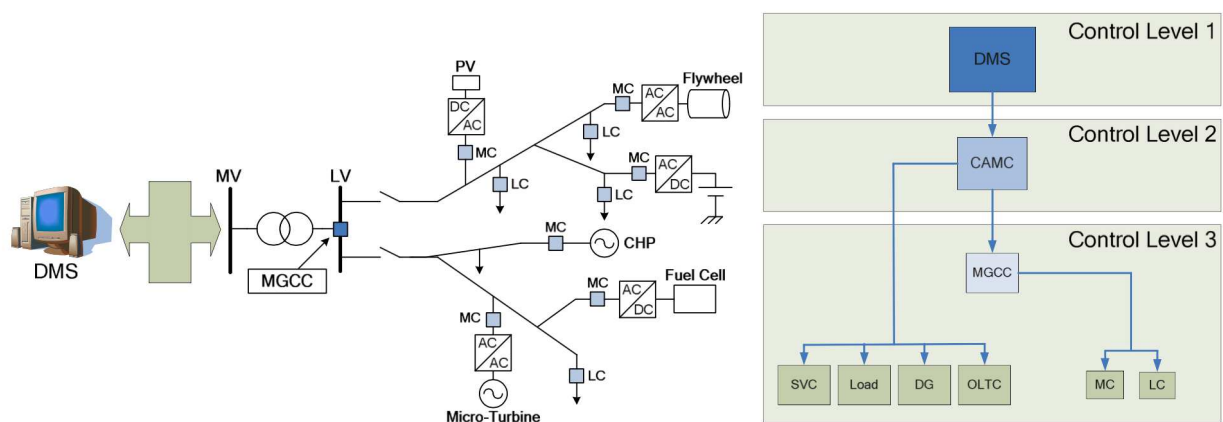


Figura 4.2: Del lado izquierdo se muestra el concepto de microrred. Del lado derecho se muestra el control jerárquico presentado en (Gil y Lopez, 2008). La notación usada es la siguiente: DMS - Sistema de Gestión de la Distribución, CAMC - Controlador Central de Administración Autónoma, MGCC - Controlador Central de la MicroRed, Load - la demanda, DG - Generación Distribuida, LC - Controladores de Demanda y MC - Controladores de Microrrecursos.

## 4.2 Optimización de despacho descentralizado en el área de energía

En (Miller et al., 2012) enfrentan el problema de optimización de generadores incrustados a una red inteligente con el objetivo de reducir las emisiones de dióxido de carbono al ambiente. Atacan el problema basado en coordinación de agentes y lo formalizan como un Problema de Optimización Descentralizada con Restricciones (DCOP, por sus siglas en inglés). Lo descomponen en un *grafo factor* y lo resuelven de manera descentralizada basada en algoritmos de la ley distributiva generalizada, en particular el algoritmo *max-sum*. En (Miller et al., 2012) se propone una modificación al algoritmo *max-sum* para reducir el número de mensajes enviados y por ende aumentar el rendimiento computacional en cuestión de tiempo. La principal diferencia con el presente trabajo de tesis es que

ellos optimizan la cantidad de energía que circula por la red de distribución. En la experimentación que se realiza en el presente trabajo de tesis no se utiliza el algoritmo *max-sum* o alguna de sus derivados debido a que este no termina su ejecución.

En (Colson y Nehrir, 2011) proponen un modelo multiagente para el control de una microrred, este modelo contiene: un agente productor, un agente consumidor y un agente observador.

El agente productor esta asociado con las fuentes de generación, sus responsabilidades son:

- Monitorear la energía activa y reactiva de los componentes de la red.
- Monitorear la energía activa y reactiva inyectada por los componentes conectados a la red.
- Determinar el costo de la energía por los elementos conectados a la red.
- Determinar rendimiento instantáneo de los elementos de la red.
- Dar los comandos de conexión/desconexión de los elementos de la red.
- Comunicar a los demás agentes la información y acciones que realice.

El agente consumidor esta asociado con los elementos de consumo, las responsabilidades de este son:

- Monitorear la energía activa y reactiva consumida por los elementos conectados a la red.
- Si existen elementos controlables, administrarlos de acuerdo a la demanda existente.
- Comunicar la información y acciones a los otros agentes.

El agente observador esta encargado de monitorear los nodos de la red, sus responsabilidades son:

- Monitorear los parámetros específicos dentro de la microrred (por ejemplo: nivel de voltaje, nivel de frecuencia, nivel de los tanques de combustible, etcétera).
- Comunicar a los demás agentes el estado de la red.

La principal diferencia de este trabajo de tesis con (Colson y Nehrir, 2011) es que ellos no realizan optimización a largo plazo si no que la realizan en tiempo real.

En (Logenthiran et al., 2011) presentan un algoritmo para el control de microrredes integradas. El algoritmo consta de tres etapas: calendarizar las microrredes de manera individual para satisfacer su demanda, buscar la manera de vender energía y recalendarizar todas las microrredes para satisfacer la demandas de todas las microrredes. La primer etapa se formula como un problema de calendarización que determina los estados (encendido/apagado) de los generadores de energía y

optimiza a un horizonte de tiempo. La segunda etapa se encarga de determinar si existen sobrantes de energía que se pueda vender al mercado. La tercera etapa se encarga de recalendarizar todas las microrredes con el fin de cubrir los faltantes que existen en algunas microrredes, si los hay. La diferencia con este trabajo de tesis es que ellos no realizan optimización a largo plazo.

En (Pipattanasomporn et al., 2009) presentan un modelo multiagente para usarse en sistema de energía autónomo distribuido. El sistema multiagente que presentan consiste en un agente de control, un agente para los recursos de energía distribuidos, un agente usuario y un agente para la base de datos. El agente de control tiene cuatro responsabilidades: (i) monitoreo de voltaje, (ii) monitoreo de frecuencia, (iii) recibir los precios de la energía eléctrica y (iv) mandar los resultados de sus monitoreos a los demás agentes. El agente para los recursos de energía distribuidos se encarga de detectar y controlar los recursos conectados a la red. El agente usuario actúa como canal de comunicación con los usuarios, a los cuales provee con información del estado de la red. Por último, el agente dedicado a base de datos se encarga de almacenar los mensajes entre los agentes con el fin de servir además de punto de acceso para los otros agentes. La principal diferencia con este trabajo de tesis es que ellos sólo proponen el modelo multiagente así como que lo proponen para ejecutarse en tiempo real.

### 4.3 Otras aplicaciones

En la toma de decisiones secuenciales, el objetivo de un agente es escoger acciones basadas en las observaciones de su ambiente que maximicen el rendimiento esperado a lo largo del tiempo. En escenarios donde las acciones no son deterministas o no se cuenta con una completa visión, los Procesos de Decisión de Markov (MDPs) y los MDPs Parcialmente Observables (POMDPs) (López et al., 2003; Spaan y Vlassis, 2004) sirven como base para los enfoques de toma de decisiones secuenciales de un agente. La extensión de estos modelos para los sistemas de múltiples agentes se ha convertido en un área de investigación creciente y activa. En la década pasada, emergieron una gran variedad de diferentes modelos multiagente para agentes cooperativos (por ejemplo: MDP Multiagentes -MMDP- (Boutilier, 1996) y POMDP Descentralizados -DecPOMDP- (Bäck y Schwefel, 1993)), para agentes egoístas (por ejemplo: POMDP Interactivos -IPOMDP- (Gmytrasiewicz y Doshi, 2005) y Juegos Estocásticos Parcialmente Observables -POSG- (Hansen et al., 2004)), además de una variedad bajo diferentes supuestos de las capacidades de los agentes para comunicarse (por ejemplo: MDP Descentralizados con Comunicación -DecMDPCom- (Goldman y Zilberstein, 2004) y Problemas de Decisión en Equipos Multiagentes Comunicativos -ComMTDP- (Pynadath y Tambe, 2002)), observar (por ejemplo: MDP Descentralizados -DecMDP- (Bernstein et al., 2002)) y influenciar a otros agentes (por ejemplo: MDP Descentralizados con Transiciones Independientes -TIDecMDP- (Becker et al., 2004) y POMDP Distribuidos en la Red -NDPOMDP- (Marecki et al., 2008)). La alta complejidad computacional ha llevado a que los investigadores desarrollen métodos multiagente para planificación y aprendizaje que exploten la estructura de interacción de los agentes, métodos orientados hacia soluciones aproximadas eficientes, métodos descentralizados que distribuyan la computación en los agentes y nuevas formas para que los agentes modelen y razonen

la interacción con otros agentes.

Se ha trabajado con diferentes aspectos con MDPs. En (Melo et al., 2012) optimizan la comunicación entre agentes. Este enfoque lo aplican para un conjunto de robots que se mueven a través de habitaciones. En el área de Dec-POMDPs, (Brafman et al., 2013) se trabajo con los dominios de los agentes, estos autores concluyen que es mejor tener una solución aproximada (buena) cuando no se cuenta con muchos recursos y/o tiempo. En (Varakantham et al., 2012) presentan un algoritmo que da prioridad a las tareas, tiene buenos resultados aunque no siempre mejora el rendimiento de su algoritmo en comparación con los que existen. En (Amato et al., 2012) presentan un enfoque donde limitan la memoria usada durante la ejecución, los autores encuentran buenas soluciones aproximadas reduciendo el costo de procesamiento.

En (Ehnberg y Bollen, 2004) por medio de simulaciones de Monte Carlo realizan la distribución de energía en pequeños lugares aislados basados únicamente en productores de energía renovables. Aquí trabajan bajo diferentes escenarios y el propósito es ver que tan confiable es generar energía solamente con generadores renovables. La conclusión es que la energía eólica tiene más disponibilidad que la solar si no se tienen unidades de almacenamiento de lo contrario se tiene mayor rendimiento con la solar. Un punto importante es la conclusión de que la combinación de diferentes generadores energías renovables pueden satisfacer de manera adecuada la demanda. En este trabajo sólo realizan simulaciones bajo Monte Carlo y el único objetivo es tratar de ver que tan confiables son los productores de energía renovables.

Algunos trabajos se enfocan en controlar una microrred de manera que esta tenga interacción con el mercado. En (Dimeas y Hatziargyriou, 2005) realizan una arquitectura completamente descentralizada para el control de una red. La arquitectura esta compuesta por operadores de la red de distribución (Distribution Operator Network, DNO), operadores de mercado (Market Operator, MO), controladores centrales de la microrred (Microgrid Central Controller, MGCC) y controladores locales (Local Controllers, LC). Los MGCC y LC están inmersos en una microrred, el DNO y MO están controlando las microrredes. Las microrredes pueden operar conectadas a una red central o aisladas. El entorno presentado se muestra en la figura 4.3. Utilizan Java Agent Development Framework (JADE).

Se ha trabajado en el control de frecuencia en una microrred. En (Black y Ilic, 2002) el enfoque tomado es del lado de la demanda, es decir, no se toma en cuenta a los microgeneradores. Aquí muestran como algunos aparatos electrónicos pueden ayudar al control de la frecuencia para así evitar apagones y caídas de tensión. La manera de resolver este problema es descentralizada.

## 4.4 Resumen

En los trabajos mencionados en este capítulo principalmente los descentralizados no se ha planteado el uso de optimización a largo plazo. Existen el planteamiento de la calendarización de los

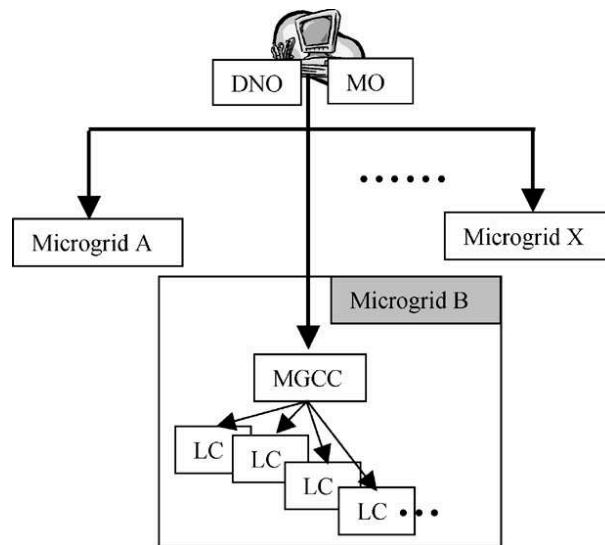


Figura 4.3: Niveles de control de un entorno microrred presentado en (Dimeas y Hatziargyriou, 2005). La arquitectura está compuesta por operadores de la red de distribución (DNO), operadores de mercado (MO), controladores centrales de la microrred (MGCC) y controladores locales (LC).

ciclos de encendido y apagado en un conjunto de generadores, pero esto se ha planteado de manera centralizada. En este trabajo se propone una arquitectura de optimización descentralizada a largo plazo que permita un mejor manejo de un conjunto de microgeneradores conectados a una microrred.

En la tabla 4.1 y 4.2 se muestra de manera resumida el trabajo en la literatura que impacta de alguna manera el presente trabajo de tesis.



Tabla 4.1: Resumen de trabajos en la literatura centralizados que enmarcan el desarrollo de la presente tesis.

Trabajo	Modelo	Aplicación
(Ehnberg y Bollen, 2004)	Centralizado	Realizan simulaciones Monte Carlo para determinar que tan confiable es producir energía eléctrica sólo con productores renovables.
(Gil y Lopez, 2008)	Centralizado	Esquema jerárquico para el control de múltiples microrredes para el control de frecuencia.
(Morais et al., 2010)	Centralizado	Optimización de calendarización de los ciclos de encendido y apagado de microgeneradores de energía eléctrica.
(Chen et al., 2011)	Centralizado	Optimiza la operación de una microrred. El sistema cuenta con un modulo de predicción, uno de administración del almacenamiento y uno de optimización.
(Chen y Gooi, 2011)	Centralizado	Calendarización de los generadores tradicionales minimizando el costo de operación y las pérdidas de energía en la red.
(Bazmohammadi et al., 2012)	Centralizado	Presentan una metodología multiobjetivo que optimiza la operación de una microrred que contiene varios tipos de generadores distribuidos y unidades de almacenamiento.
(Wang et al., 2012)	Centralizado	Optimizan el uso de la energía de viento dándole valor de riesgo por uso excesivo.
(Chaouachi et al., 2013)	Centralizado	Optimizan el uso de una microrred mediante el uso de técnicas de inteligencia artificial. Su objetivo principal es calendarizar el uso de las baterías.
(Su et al., 2014)	Centralizado	Formula un problema estocástico para calendarizar la energía en una microrred, minimizando el costo de operación.

Tabla 4.2: Resumen de trabajos en la literatura descentralizados que enmarcan el desarrollo de la presente tesis.

Trabajo	Modelo	Aplicación
(Black y Ilic, 2002)	Descentralizado	Se realiza control de demanda para mantener la estabilidad en una red.
(Dimeas y Hatziargyriou, 2005)	Descentralizado	Realizan una arquitectura para el control de una microrred.
(Pipattanasomporn et al., 2009)	Descentralizado	Presentan un modelo multiagente para usarse en sistema de energía autónomo distribuido. El sistema multiagente consiste en los siguientes agentes: de control, para los recursos de energía distribuidos, usuario y uno para la base de datos.
(Colson y Nehrir, 2011)	Descentralizado	Proponen un modelo multiagente para el control de una microrred. El modelo multiagente que proponen es: agente productor, agente consumidor y agente observador.
(Logenthiran et al., 2011)	Descentralizado	Presentan un algoritmo para el control de microrredes integradas. El algoritmo consta de tres etapas: calendarizar las microrredes de manera individual para satisfacer su demanda, buscar la manera de vender energía y recalendarizar todas las microrredes para satisfacer las demandas de todas.
(Amato et al., 2012)	Descentralizado	Enfoque donde se limita el uso de memoria al tiempo de ejecución.
(Melo et al., 2012)	Descentralizado	Optimización de la comunicación entre agentes.
(Miller et al., 2012)	Descentralizado	Optimización de generadores incrustados a una red inteligente con el objetivo de reducir las emisiones de dióxido de carbono al ambiente.
(Varakantham et al., 2012)	Descentralizado	Presentan un algoritmo que prioriza las tareas, con el objetivo de dar prioridad a las tareas más importantes primero.
(Brafman et al., 2013)	Descentralizado	Trabajo en optimización de los dominios de los agentes.

# 5

## Método propuesto

En este capítulo se describe la solución propuesta para el problema de calendarización de ciclos de encendido y apagado de microgeneradores de manera descentralizada.

Se cuenta con un conjunto de 3 generadores de energía renovable (turbina de viento, generador solar y celda de combustible) en donde la producción de 2 de ellos (turbina de viento y generador solar) son dependientes del clima, es decir, su producción no se puede controlar. Además de los generadores se cuenta con un dispositivo con capacidad de almacenamiento (batería), esta tiene una capacidad máxima de almacenamiento así como una capacidad máxima de carga y descarga. El objetivo es minimizar el costo de producción de energía a las siguientes 24 horas.

Al optimizar de manera descentralizada se tienen las siguientes ventajas: (i) reducción del costo computacional en relación al enfoque centralizado, (ii) el problema se vuelve más escalable; es decir, se pueden introducir más elementos en comparación del modelo centralizado y (iii) el sistema es tolerante a fallos, si algún elemento deja de funcionar el sistema sigue en pie sin tomar en cuenta el elemento faltante.

Lo primero que se propone es el uso de ventanas de optimización y repetir el proceso varias veces (e. g. 24, 12, 8, 6, ... veces para ventanas de optimización de tamaño 1, 2, 3, 4, ...). Cada ventana de optimización tiene diferentes condiciones iniciales (dependiente de la ventana anterior). En el esquema descentralizado no se elimina el comportamiento exponencial del problema a resolver, este enfoque permite tratar con problemas con mayor número de variables. Los algoritmos de la familia DCOP distribuyen el procesamiento, pero el problema sigue siendo NP-Completo. Debido a lo antes mencionado cualquier solución propuesta llega un momento en que se vuelve intratable.

La siguiente propuesta es particionar el dominio (mostrado en la sección 5.2) de las variables para poder reducir el tamaño del problema. Esta propuesta (*primero el mejor*) tiene como desventaja que al trabajar solamente con partes del dominio es susceptible a caer en mínimos locales.

Por último, se propone una versión modificada de la búsqueda  $A^*$  con el objetivo de mejorar la solución obtenida al utilizar la búsqueda *primero el mejor*.

Se utilizaron las librerías FRODO2 (Léauté et al., 2009) y JaCoP (Kuchcinski y Szymanek, 2014) (implementadas en Java) y se utilizaron diversos algoritmos de la familia DCOP implementados en estas plataformas. La elección de los algoritmos fue en base a la complejidad del problema a optimizar y las diferentes características que estos poseen. Otro motivo para la elección de los algoritmos fue que entre más horizontes de tiempo se optimizaban, el número de agentes aumentaba, en tanto la complejidad aumentaba también. Para las 3 propuestas se utilizaron los mismos 4 algoritmos (AFB, DPOP, MB-DPOP y SynchBB). Para la experimentación no se utilizó *max-sum* ni ninguno de sus derivados (como el propuesto en (Miller et al., 2012)) debido a que este algoritmo no termina su ejecución para este problema.

Debido a que la representación completa del problema es exponencial con respecto al número de variables (celdas de combustible, carga y descarga de batería) se propone dividirlo en subproblemas. Se dividió en 24 partes (uno por hora), después se dividió en 12 partes (dos horas al mismo tiempo) y así sucesivamente. Esto se realizó de esta manera por que así se puede observar el comportamiento algorítmico además de poder determinar el tamaño máximo de las ventanas que se puede optimizar así como poder determinar un número de ventanas con el cual existe la mejor relación tiempo y error.

## 5.1 Optimización por ventanas

El objetivo de realizar optimización por medio de ventanas es reducir el tiempo computacional. Una de las desventajas de este enfoque es que no se obtiene la solución óptima, es un aproximado. Sabiendo que el resultado que se obtiene es un aproximado se busca encontrar una relación de tiempo de ejecución y error aceptable para este problema.

Este enfoque funciona de la siguiente manera, se utilizan ventanas de optimización y se repite el proceso varias veces (e. g. 24, 12, 8, 6, ... veces para optimización a 1, 2, 3, 4, ... horizontes de tiempo). Se tiene como entrada a la primer ventana los estados iniciales de las variables y se optimiza para este tiempo. Después de optimizada la ventana inicial se procede con la siguiente, donde las condiciones iniciales dependen del resultado de optimización de la ventana anterior. Después de esto se sigue optimizando en orden hasta llegar a la ventana 24. El diagrama de flujo se muestra en la figura 5.1.

Para la optimización se utilizaron los algoritmos (descritos en la sección 2.2): AFB, DPOP, MB-DPOP y SynchBB con el objetivo encontrar un punto de equilibrio entre el error y el tiempo de ejecución. Para ventanas de tamaño uno, se optimizan celdas de combustible, carga de batería y descarga de batería tomando en cuenta la producción de energía eólica y fotovoltaica así como la carga (consumo) de ese tiempo. Para ventanas de tamaño dos es igual que para la de tamaño uno;

la diferencia radica en que el número de variables se duplica, ahora son dos celdas de combustible (tiempo 1 y tiempo 2) dos cargas de batería (tiempo 1 y tiempo 2), etcétera.

Al pasar a un esquema descentralizado el problema a resolver sigue siendo NP-Completo. Los algoritmos de la familia DCOP distribuyen el procesamiento, pero el problema sigue siendo NP-Completo. Debido a lo mencionado anteriormente cualquier solución propuesta llega un momento en que se vuelve intratable. Es por eso que se propone utilizar una búsqueda basada en un heurística que permita reducir el dominio de las variables y así poder tener mayor escalabilidad.

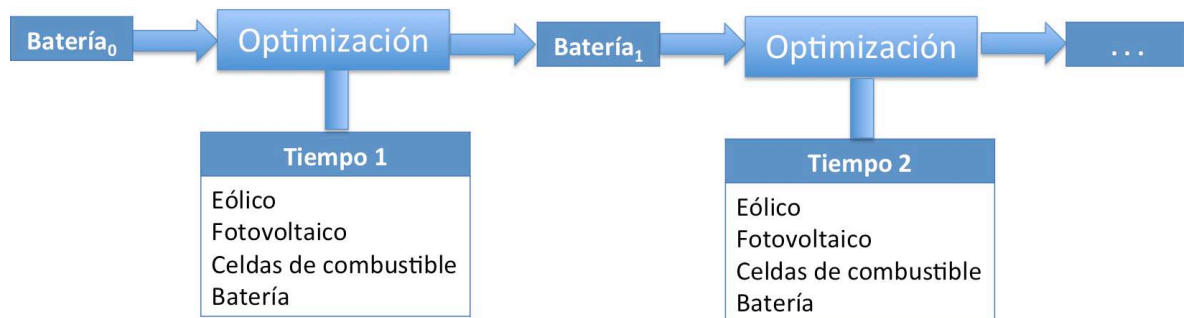


Figura 5.1: Optimización por medio de ventanas. Los rectángulos marcados como *Optimización* representan al algoritmo utilizado, este está relacionado con las variables a optimizar. Los rectángulos marcados como *Batería<sub>t</sub>* representan el estado de la batería al entrar y salir de la optimización.

## 5.2 Optimización utilizando búsqueda primero el mejor

Debido a que al aumentar el tamaño de las ventanas de optimización el tiempo de ejecución crece, se propone reducir el dominio de las variables. Esto se realiza mediante un proceso de particionado del dominio. Para particionar el dominio utilizando *primero el mejor* se propone lo siguiente:

- para inicializar el procedimiento se recibe como entrada el dominio completo de la variable,
- si se solicita particionado del dominio,
  - recibe como entrada el valor medio de la mejor partición hasta el momento, determinada por un algoritmo DCOP,
  - toma la partición correspondiente al valor de entrada y eliminan las demás particiones,
  - particiona en  $N$  partes el dominio,
  - si no existen más particiones del dominio, la variable se marca como terminada,
  - el valor de entrada se guarda como mejor valor encontrado

- si se solicitan valores de la partición y no esta marcado como terminado,
  - retorna el valor medio de cada una de las  $N$  particiones en un arreglo.
- si se solicitan valores de la partición y está marcado como terminado,
  - retorna el mejor valor encontrado

En la figura 5.2 se muestra el diagrama de flujo para realizar el particionado del dominio. El caso ilustrado en la figura es para cuando se utilizan ventanas de tamaño 1, pero la entrada es dependiente del tamaño de la ventana. En el diagrama al algoritmo DCOP entran un vector de dos valores por cada variable, este elige la mejor combinación y da como salida por cada variable un valor. Una vez elegido este valor, al particionado entra el nuevo dominio de la variable que se elige como ya se indicó anteriormente. En la figura 5.3 se muestra un ejemplo de particionado.

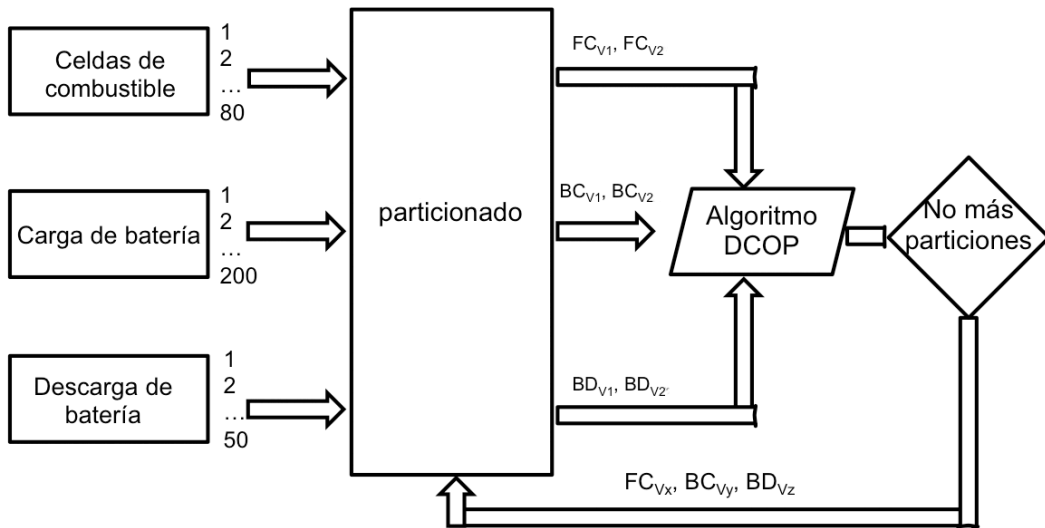


Figura 5.2: Diagrama de flujo de la realización del particionado del dominio.  $FC_{V1}$ ,  $FC_{V2}$ ,  $BC_{V1}$ ,  $BC_{V2}$ ,  $BD_{V1}$  y  $BD_{V2}$  representan los dos valores obtenidos del particionado y  $FC_{Vx}$ ,  $BC_{Vy}$  y  $BD_{Vz}$  representan la mejor configuración (menor valor) decidida por el algoritmo DCOP.

Al utilizar esta técnica se sigue obteniendo un valor aproximado de la solución óptima por lo que se decidió utilizar una búsqueda guiada  $A^*$ . Esta elección fue para mejorar la solución ya encontrada.

### 5.3 Optimización utilizando $A^*$

Además de los enfoques mencionados en las secciones anteriores, se utilizaron heurísticas para ayudar a reducir el espacio de posibles configuraciones particionando el dominio de cada variable

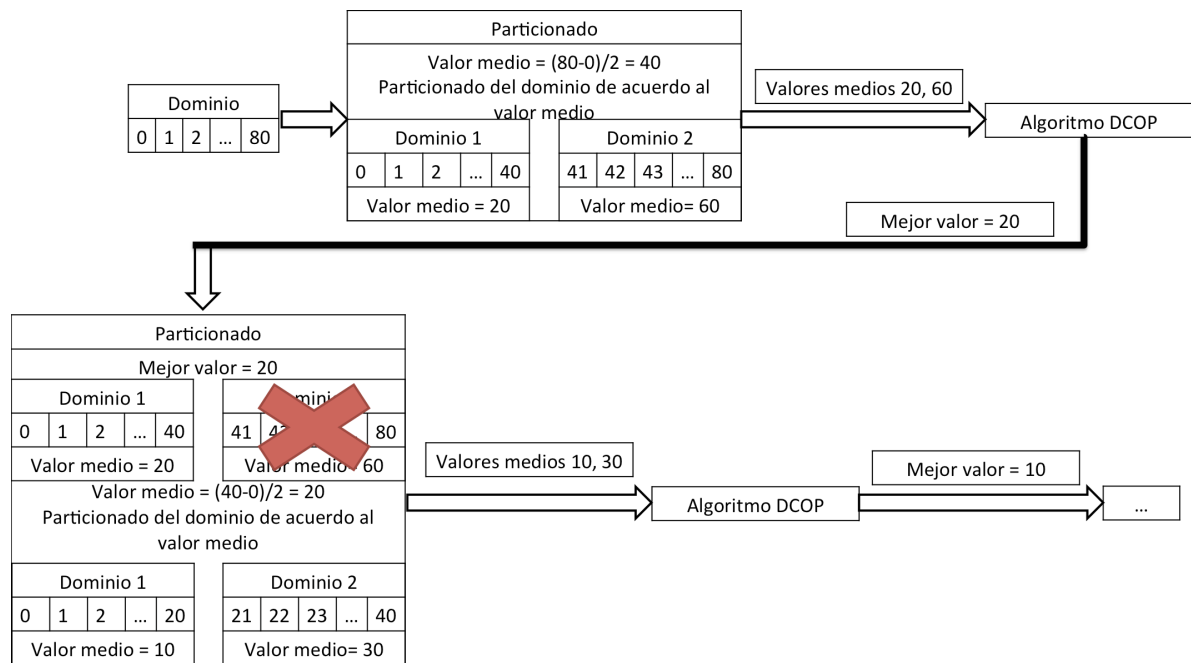


Figura 5.3: Ejemplo de particionado. El dominio de entrada consta de los valores del 0 al 8, el particionado elige el valor medio (40), se crean dos nuevos dominios en base al valor medio y se obtiene el valor medio de cada dominio. Se envían los valores medios de las dos particiones del dominio al algoritmo DCOP para optimizar y elegir el mejor valor. Se regresa el mejor valor al particionado y este elige la partición perteneciente a este valor medio y desecha la otra y repite el proceso con la partición elegida.

controlable.

El algoritmo completo se muestra en el algoritmo 5.1. El algoritmo toma como entrada un grafo con restricciones (formalización para los algoritmos DCOP) para un problema de coordinación multiagente. Como entrada se tiene lo siguiente:  $\mathcal{A}$ ,  $\mathcal{X}$ ,  $\mathcal{D}$ ,  $\mathcal{F}$  y  $N$ , donde  $N$  es el número de divisiones,  $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$  es un conjunto de agentes y  $\mathcal{X} = \{x_1, x_2, \dots, x_m\}$  es un conjunto de variables, cada variable  $x_i$  pertenece sólo a un agente.  $\mathcal{D} = \{D_1, D_2, \dots, D_m\}$  es un conjunto de dominios discretos y finitos para las variables. Cada variable  $x_i$  puede tomar valores en el dominio  $D_i$ . Finalmente,  $\mathcal{F} = \{F_1, F_2, \dots, F_p\}$  es un conjunto de funciones que describen las restricciones de las variables. Como salida, el algoritmo da una configuración optimizada de sus variables  $\mathcal{C}$ .

Para realizar la optimización se utilizaron los siguientes pasos (Algoritmo 5.1):

1. Como entrada se definen todos los agentes y sus dominios (en este caso en particular celdas de combustible, carga de batería y descarga de batería),
2. Cada variable (e. g. celdas de combustible, carga de batería, descarga de batería) crea una instancia de un algoritmo *primero el mejor* para controlar el particionado de su dominio,

3. Cada variable particiona su dominio, después manda los valores medios de cada partición a algún algoritmo DCOP (todas las variables utilizan el mismo algoritmo DCOP),
4. El algoritmo DCOP optimiza hasta encontrar la configuración óptima con el conjunto de valores con los que cuenta,
5. Mientras los algoritmos *primero el mejor* de todas las variables no hayan terminado, se repiten los pasos 3 y 4.
6. Una vez terminado de particionar los dominios, la asignación encontrada se envía al algoritmo  $A^*$  para encontrar una mejor solución.

El algoritmo  $A^*$  (explicado en la sección 2.4) es un algoritmo de búsqueda que requiere una configuración inicial y una final (objetivo). En nuestro caso no utilizamos una configuración objetivo (debido a que no se conoce y es a lo que se quiere llegar), se utiliza una *configuración de referencia*, es decir, las configuraciones mejores que este punto son analizadas y tomadas en cuenta como posible valor “óptimo”. Como estimador se utiliza la función objetivo del problema para una sola configuración (función de costo). En nuestro caso, utilizamos como configuración inicial y *configuración de referencia* la mejor configuración encontrada por el algoritmo *primero el mejor* en conjunto con el algoritmo DCOP. Ya determinado la configuración inicial y de referencia, se analizan todos los vecinos y entran a la lista por analizar todos aquellos valores que cumplan la condición de ser mejores (menor costo) que el punto analizado. Por lo demás, se sigue el mismo proceso que en el algoritmo  $A^*$ . Como salida da una configuración que se toma como la mejor configuración (más cercana al punto óptimo), este algoritmo se muestra en 5.2.

La búsqueda guiada realizada por  $A^*$  también se hace por medio de ventanas.

---

**Algoritmo 5.1:** Particionado $A^*$ 


---

**entrada:**  $\mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{F}, N$

**salida** : la configuración de las variables del problema de optimización  $\mathcal{C}$

- 1: **for**  $x_i \in \mathcal{X}$  **do**
  - 2:     ┌ se crea instancia del procedimiento de particionado
  - 3: **while** los algoritmos *primero el mejor* de todas las variables no hayan terminado de particionar su dominio **do**
  - 4:     ┌ **for**  $x_i \in \mathcal{X}$  **do**
  - 5:         ┌ se particiona el dominio
  - 6:         └ se obtienen los valores medios del particionado del dominio
  - 7:     └  $\mathcal{C} \leftarrow \text{DCOP}(\mathcal{A}, \mathcal{X}, \mathcal{D}, \mathcal{F})$
  - 8: DCOP- $A^*(\mathcal{C})$
-



**Algoritmo 5.2:** DCOP- $A^*(\mathcal{C})$ 

**entrada:**  $\mathcal{C}$ : la configuración encontrada por la búsqueda *primero el mejor* y el algoritmo DCOP

**salida** :  $\mathcal{C}^*$ : una nueva configuración (mejor) de las variables

```

1: objetivo  $\leftarrow \mathcal{C}$ 
2: conjuntoCerrado  $\leftarrow \phi$ 
3: conjuntoAbierto  $\leftarrow \mathcal{C}$ 
4: costo[ $\mathcal{C}$ ] = calculaCosto( $\mathcal{C}$ )
5: while conjuntoAbierto no esta vacío do
6:     actual  $\leftarrow$  configuración en el conjuntoAbierto que tiene el menor valor de costo[]
7:     conjuntoAbierto = conjuntoAbierto - actual
8:     conjuntoCerrado = conjuntoCerrado + actual
9:     for vecino  $\in$  nodosVecinos(actual) do
10:         if vecino  $\in$  conjuntoCerrado then
11:              $\lfloor$  continuar
12:         costoVecino = calculaCosto(vecino)
13:         if vecino  $\notin$  conjuntoAbierto and costoVecino < costo[actual] then
14:             costo[vecino] = costoVecino
15:             if vecino  $\notin$  conjuntoAbierto then
16:                  $\lfloor$  conjuntoAbierto = conjuntoAbierto + vecino
17:  $\mathcal{C}^* \leftarrow$  el nodo en el conjuntoCerrado que tiene el menor valor de costo[]
18: return  $\mathcal{C}^*$ 

```

En la figura 5.4 se un ejemplo con 3 iteraciones del funcionamiento de la búsqueda  $A^*$  en nuestro dominio. En la figura se puede observar que algunas configuraciones no se abren, esto puede ser por dos motivos, esta se encuentra en el conjunto cerrado o se viola alguna restricción.

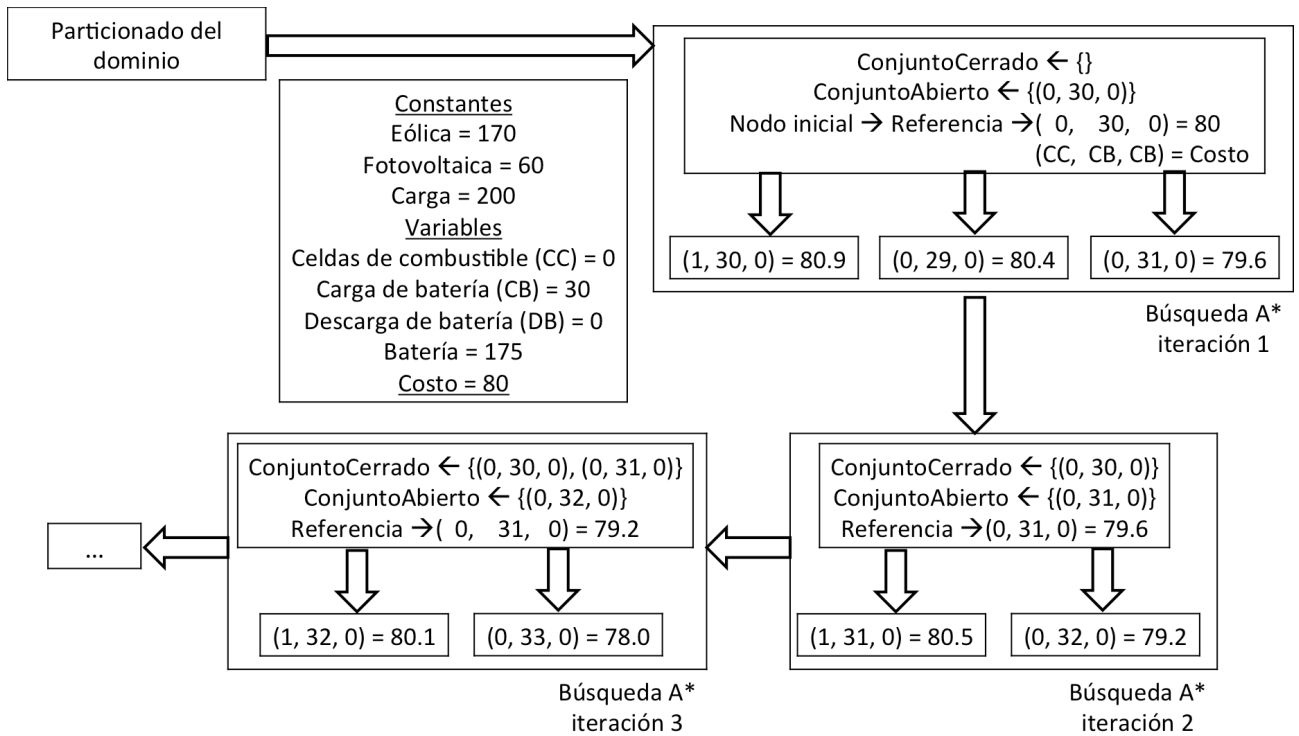


Figura 5.4: Ejemplo de funcionamiento de la búsqueda  $A^*$ . La configuración inicial es obtenida por medio de particionado de dominio, las constantes son las predicciones con las que se cuenta (escenarios). Se muestran 3 iteraciones de la búsqueda  $A^*$ , el costo es calculado de acuerdo a la función objetivo.

## 5.4 Resumen

En este capítulo se describieron los métodos propuestos para la solución del problema de calendarización de ciclos de encendido y apagado en un conjunto de microgeneradores. La primera solución propuesta es la optimización por medio de ventanas. Se propone dividir el problema en pequeños subproblemas y resolverlos individualmente pero de manera secuencial; es decir, se resuelve primero el primer subproblema después el segundo así hasta resolver el último.

La segunda propuesta es la optimización utilizando una búsqueda primero el mejor. Se propone atacar el problema combinatorio de los dominios utilizando una búsqueda primero el mejor para así reducir el dominio de las variables.

Por último, se muestra la propuesta de mejorar la solución obtenida de utilizar optimización por medio de ventanas y búsqueda primero el mejor al utilizar un búsqueda guiada  $A^*$ . En el siguiente capítulo se mostrará las pruebas experimentales para la comparación de cada uno de los modelos propuestos en error, tiempo de ejecución, número y tamaño de mensajes.

# 6

## Pruebas experimentales

En este capítulo se muestran los resultados obtenidos de la experimentación. El capítulo está dividido en dos partes, los resultados obtenidos con una solución centralizada y los obtenidos con una solución descentralizada.

En la primera parte se muestran los resultados cuyo objetivo es replicar y verificar los resultados obtenidos de manera centralizada contra los expuestos en (Morais et al., 2010). Se muestran una comparación de resultados, aquí se muestran los resultados obtenidos de manera centralizada utilizando la formulación de (Morais et al., 2010).

En la segunda parte de este capítulo se muestran los resultados de las pruebas experimentales de manera descentralizada. Esta segunda parte se divide en tres secciones. En la primera sección se muestran los resultados con la primera propuesta de esta tesis, que trata de optimizar por ventanas. En la segunda sección se muestran los resultados obtenidos mediante el uso de una búsqueda *primero el mejor*. En la tercera sección se muestran los resultados donde se aplica una heurística para guiar una búsqueda. El objetivo de los experimentos es validar el porcentaje error contra el modelo centralizado. El error es calculado mediante error relativo, donde la base es el valor obtenido mediante la optimización de manera centralizada. Otro de los objetivos de estos experimentos es validar el rendimiento de la solución descentralizada con diferentes algoritmos de optimización multiagente. Se utilizan 4 medidas de validación: error, tiempo de ejecución, promedio de mensajes intercambiados por los agentes y promedio de tamaño de los mensajes intercambiados por los agentes.

Para la experimentación tanto centralizada como descentralizada se utilizó una PC con procesador Intel(R) Core(TM) i7-4770 @ 3.40GHz con 16 GB de RAM.

## 6.1 Solución centralizada

En esta sección se muestran, explican y comparan los resultados dados en (Morais et al., 2010) contra los replicados. Los resultados mostrados en esta sección se obtuvieron de manera centralizada. Se muestran los resultados para los dos escenarios mostrados en la tabla 6.1.

Tabla 6.1: Predicciones de generadores y escenarios de consumo obtenidos de Morais et al. 2010.

Tiempo (horas)	Eólica (W/h)	Fotovoltaica (W/h)	Carga (W/h) [Esc 1]	Carga (W/h) [Esc 2]
1	130	0	160	230
2	150	0	140	190
3	140	0	150	160
4	160	0	120	150
5	100	0	110	140
6	120	5	100	180
7	150	10	170	230
8	180	30	180	250
9	170	60	200	270
10	160	100	220	280
11	120	130	230	240
12	130	140	240	300
13	150	150	240	300
14	176	140	230	280
15	185	130	220	270
16	120	100	210	280
17	130	60	210	290
18	140	30	220	300
19	170	10	230	310
20	190	5	240	320
21	120	0	250	330
22	170	0	200	300
23	130	0	190	280
24	150	0	180	270

En los resultados reportados en (Morais et al., 2010), cada hora se satisface la demanda para el escenario 1, mientras que existen deficiencias de entrega de energía para el escenario 2. Esta deficiencia se debe a que a pesar de tener estas fuentes de generación (eólica, fotovoltaica y celdas

de combustible) y almacenamiento (batería) no se tiene suficiente capacidad para satisfacer la demanda con las condiciones planteadas.

El replicado de los resultados se realizó debido a que se debía verificar la formulación y los resultados. Para obtener estos resultados se aplicaron las restricciones y consideraciones mostradas en la sección 3.1 utilizando la metodología mostrada en la sección 3.3.2, que es replicar los resultados usando Java e ILOG CPLEX. Al existir varios mínimos las configuraciones obtenidas en los resultados replicados difieren un poco con los reportados en (Morais et al., 2010) pero tienen el mismo costo global (valor óptimo).

En la tabla 6.2 se muestran los resultados replicados de la optimización para el escenario 1. En la tabla 6.3 se muestran los resultados replicados de la optimización para el escenario 2.

Se muestran diferentes resultados para cada escenario (con respecto al artículo (Morais et al., 2010)) debido a que no sólo existe una configuración óptima que satisface las restricciones del problema. Se encontró que existen múltiples configuraciones que satisfacen las restricciones que además tienen el costo mínimo, esto se conoce en la literatura como problema degenerado.

En las tablas se muestra la configuración óptima de los microgeneradores. En la columna de *Costo* se muestra el costo de generación en cada tiempo  $T$ . En la fila siguiente a cada configuración se muestra el costo total de generación en las 24 horas (costo de la configuración), que es la función objetivo a minimizar. La simbología utilizada es la siguiente:

- $T$  → tiempo,
- $EO$  → generación eólica,
- $FO$  → generación fotovoltaica,
- $CC$  → generación por celdas de combustible,
- $CB$  → carga de batería,
- $DB$  → descarga de batería,
- $CA$  → carga,
- $NE$  → energía no entregada, y
- $EX$  → exceso de energía generada.

Tabla 6.2: Resultados obtenidos mediante Java-ILOG CPLEX en escenario 1, dos configuraciones óptimas.

T	EO	FO	CA	CC	CB	DB	NE	EX	Costo	CC	CB	DB	NE	EX	Costo		
1	130	0	160	0	0	30	0	0	70	0	0	30	0	0	70		
2	150	0	140	0	10	0	0	0	56	0	0	0	0	10	60		
3	140	0	150	0	0	10	0	0	62	0	0	10	0	0	62		
4	160	0	120	0	40	0	0	0	48	0	0	0	0	40	64		
5	100	0	110	0	0	10	0	0	46	0	0	10	0	0	46		
6	120	5	100	0	25	0	0	0	40	0	0	0	0	25	50		
7	150	10	170	0	0	10	0	0	70	0	0	10	0	0	70		
8	180	30	180	0	30	0	0	0	72	0	0	0	0	30	84		
9	170	60	200	0	30	0	0	0	80	0	0	0	0	30	92		
10	160	100	220	0	0	0	0	40	104	0	0	0	0	40	104		
11	120	130	230	0	20	0	0	0	92	0	0	0	0	20	100		
12	130	140	240	0	5	0	0	25	106	0	0	0	0	30	108		
13	150	150	240	0	0	0	0	60	120	0	50	0	0	10	100		
14	176	140	230	0	0	0	0	86	126.4	0	50	0	0	36	106.4		
15	185	130	220	0	0	0	0	95	126	0	50	0	0	45	106		
16	120	100	210	0	0	0	0	10	88	0	10	0	0	0	84		
17	130	60	210	20	0	0	0	0	94	0	0	20	0	0	88		
18	140	30	220	0	0	50	0	0	98	0	0	50	0	0	98		
19	170	10	230	0	0	50	0	0	102	0	0	50	0	0	102		
20	190	5	240	25	0	20	0	0	112.5	45	0	0	0	0	118.5		
21	120	0	250	80	0	50	0	0	150	80	0	50	0	0	150		
22	170	0	200	30	0	0	0	0	95	30	0	0	0	0	95		
23	130	0	190	60	0	0	0	0	106	60	0	0	0	0	106		
24	150	0	180	0	0	30	0	0	78	0	0	30	0	0	78		
									Total	2141.9						Total	2141.9

Tabla 6.3: Resultados obtenidos mediante Java-ILOG CPLEX en escenario 2, dos configuraciones óptimas.

T	EO	FO	CA	CC	CB	DB	NE	EX	Costo	CC	CB	DB	NE	EX	Costo		
1	130	0	230	80	0	20	0	0	136	80	0	20	0	0	136		
2	150	0	190	40	0	0	0	0	96	40	0	0	0	0	96		
3	140	0	160	20	0	0	0	0	74	20	0	0	0	0	74		
4	160	0	150	19	29	0	0	0	69.5	19	29	0	0	0	69.5		
5	100	0	140	40	0	0	0	0	76	40	0	0	0	0	76		
6	120	5	180	55	0	0	0	0	99.5	55	0	0	0	0	99.5		
7	150	10	230	70	0	0	0	0	127	70	0	0	0	0	127		
8	180	30	250	40	0	0	0	0	120	40	0	0	0	0	120		
9	170	60	270	40	0	0	0	0	128	40	0	0	0	0	128		
10	160	100	280	20	0	0	0	0	122	20	0	0	0	0	122		
11	120	130	240	0	10	0	0	0	96	0	10	0	0	0	96		
12	130	140	300	30	0	0	0	0	135	30	0	0	0	0	135		
13	150	150	300	0	0	0	0	0	120	0	0	0	0	0	120		
14	176	140	280	0	36	0	0	0	112	0	36	0	0	0	112		
15	185	130	270	0	45	0	0	0	108	0	45	0	0	0	108		
16	120	100	280	60	0	0	0	0	142	60	0	0	0	0	142		
17	130	60	290	80	0	0	20	0	178	80	0	0	20	0	178		
18	140	30	300	80	0	0	50	0	215	80	0	0	50	0	215		
19	170	10	310	80	0	50	0	0	174	80	0	0	50	0	219		
20	190	5	320	80	0	45	0	0	177	80	0	30	15	0	190.5		
21	120	0	330	80	0	55	75	0	265.5	80	0	100	30	0	225		
22	170	0	300	80	0	50	0	0	170	80	0	0	50	0	215		
23	130	0	280	80	0	0	70	0	229	80	0	70	0	0	166		
24	150	0	270	80	0	0	40	0	192	80	0	0	40	0	192		
									Total	3361.5						Total	3361.5

En las figuras 6.1 y 6.2 se muestran los resultados replicados (mostrados en la tabla 6.2) obtenidos de la optimización centralizada para el escenario 1. En el primer gráfico se muestran los resultados de dos diferentes configuraciones óptimas para las restricciones en lo referido a producción. En el segundo gráfico se muestra el consumo. De estos gráficos se puede observar que existen diferentes configuraciones óptimas que cumplen con las restricciones. Como ya se mencionó anteriormente el costo de la configuración final (24 horas) es lo que determina que configuración es mejor.

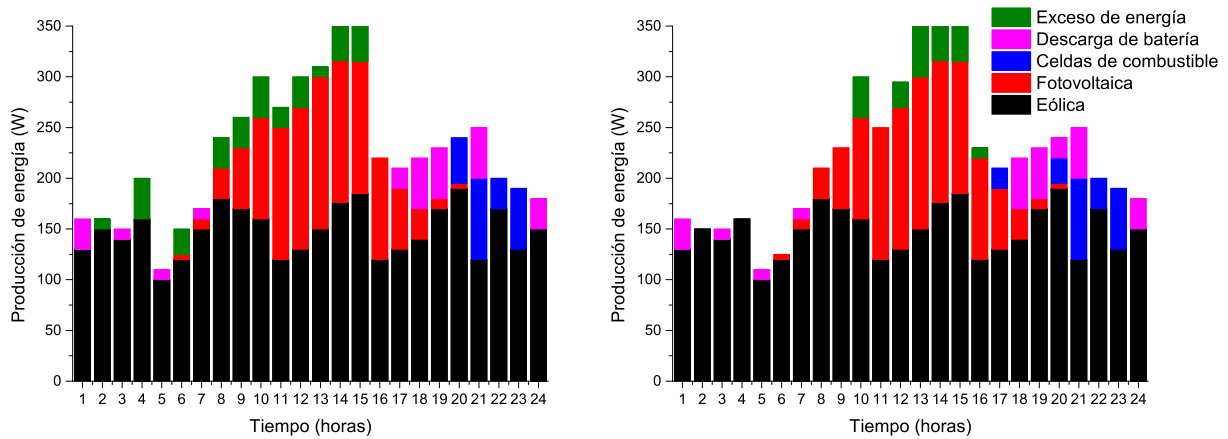


Figura 6.1: Resultados de dos diferentes configuraciones de producción óptimas para el escenario 1.

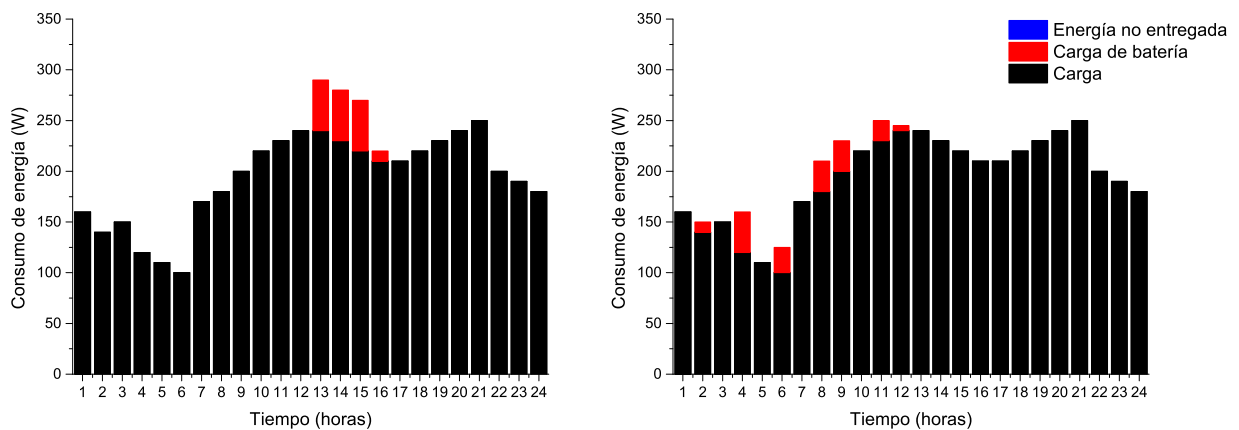


Figura 6.2: Resultados de dos diferentes configuraciones de consumo óptimas para el escenario 1.

En las figuras 6.3 y 6.4 se muestran los resultados obtenidos de la optimización centralizada para el escenario 2, en el primer gráfico se muestran los resultados de dos diferentes configuraciones óptimas para las restricciones en lo referido a producción. En el segundo gráfico se muestra el consumo. Al igual que el escenario 1, en el segundo también podemos obtener diferentes configuraciones

óptimas que satisfagan todas las restricciones. Como ya se mencionó anteriormente el costo de la configuración final (24 horas) es lo que determina que configuración es mejor.

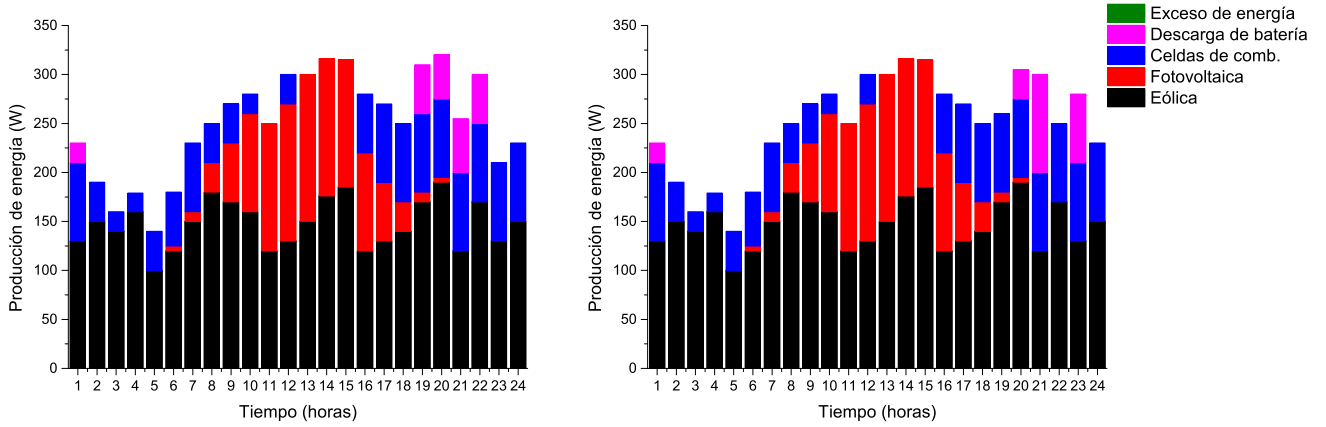


Figura 6.3: Resultados de dos diferentes configuraciones de producción óptimos para el escenario 2.

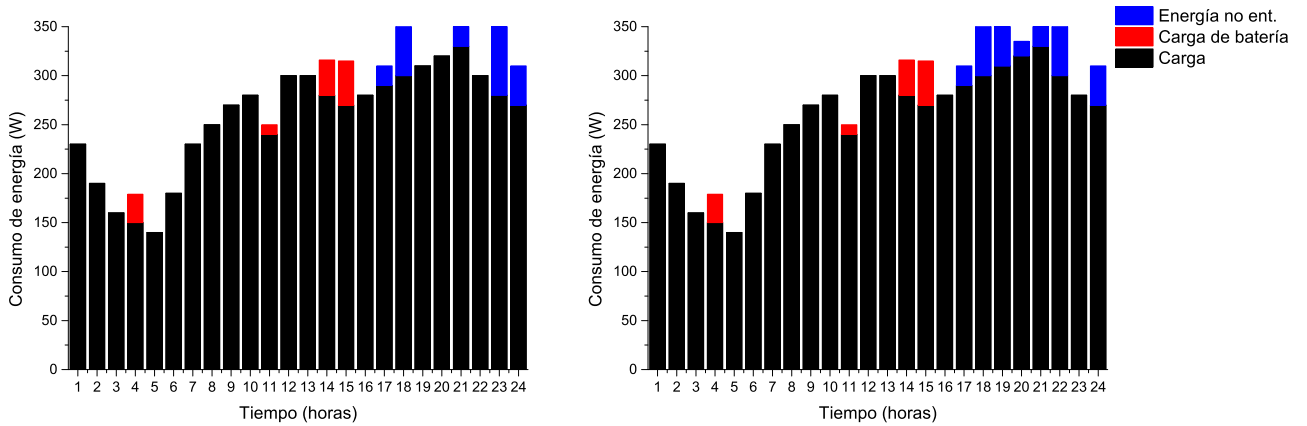


Figura 6.4: Resultados de dos diferentes configuraciones de consumo óptimos para el escenario 2.

Los tiempos de ejecución al replicar los resultados es de 70 milisegundos con una desviación estándar de 10 milisegundos, esto se obtuvo realizando 1000 experimentos.

Se logró replicar los resultados de (Morais et al., 2010) de manera centralizada, obteniendo configuraciones óptimas que tienen el costo más bajo, es decir nosotros encontramos otras configuraciones óptimas de costo más bajo, cuyo costo concuerda con lo reportado en (Morais et al., 2010). Con esto se encontró que no sólo existe una configuración óptima, el llegar a una solución u otra depende del algoritmo utilizado.



Con lo mostrado en las tablas y en las figuras se demuestra que dada las condiciones del escenario 2, no se tiene suficiente capacidad de producción y/o almacenamiento de energía para poder satisfacer la demanda que se tiene. Esto se puede observar en la tabla 6.3 y en la figura 6.4, ya que dada las restricciones siempre existe un faltante en la entrega de energía a ciertas horas. Lo único cambiante son los montos de energía no entregada y en que instantes no se entrega. Dependiendo de las condiciones con las que se cuenten se puede restringir o parcializar la falta de energía a través del tiempo para tener muchos tiempos con poco faltante o pocos tiempos con mucho faltante.

## 6.2 Solución descentralizada

Al tener un esquema centralizado se tiene como desventaja que es muy poco escalable, es decir, que con pocos elementos en el sistema el problema se vuelve intratable esto debido a que es exponencial con respecto al número de variables. Se presenta un esquema descentralizado para ser usado por uno o más algoritmos de la familia DCOP para optimizar el uso de un conjunto de microgeneradores a largo plazo obteniendo una buena configuración (de bajo costo), con bajo costo computacional, escalable y tolerante a fallos; cuya solución se aproxime o incluso iguale a la solución (óptima) de los algoritmos centralizados. La primera propuesta es ventanas de optimización, la segunda propuesta es reducción de dominio con una búsqueda primero el mejor, la tercera propuesta es mejorar la solución utilizando una búsqueda guiada  $A^*$ .

El objetivo de los experimentos es validar el error en comparación contra el modelo centralizado. Otro de los objetivos de estos experimentos es validar el rendimiento de la solución descentralizada con diferentes algoritmos de optimización multiagente, se utilizan 4 medidas de comparación: error, tiempo de ejecución, promedio del número mensajes intercambiados por los agentes y promedio de tamaño de los mensajes intercambiados por los agentes.

### 6.2.1 Ventanas de optimización

Los resultados obtenidos en esta sección corresponden a los obtenidos mediante al uso de ventanas de optimización y repetir el proceso varias veces (p. e. 24, 12, 8, 6, ... veces para optimización a 1, 2, 3, 4, ... horizontes de tiempo). Cada ventana de optimización tiene diferentes condiciones iniciales (que dependen de la ventana anterior). Los algoritmos utilizados son: *AFB*, *DPOP*, *MB-DPOP* y *Synchronous Branch and Bound*. Estos algoritmos se eligieron debido a que atacan el problema de distinta manera, ya sea programación lineal o técnicas asíncronas, así como que unos proporcionan solución exacta mientras que otros dan una aproximada. Se realizaron 100 ejecuciones de cada simulación a excepción del algoritmo *AFB* con 3 horizontes de tiempo debido al crecimiento exponencial (mostrado en secciones siguientes) tiempo de ejecución.

Para el cálculo del error se utilizó como referencia el costo de las configuraciones óptimas (función objetivo), donde este costo representa el 100% y la diferencia del valor óptimo contra el

obtenido representa el error (error relativo). Los resultados para las medidas de comparación para el escenario 1 se muestran en la tabla 6.4. En la tabla 6.5 se muestran los resultados para las medidas de comparación para el escenario 2. Los algoritmos marcados en las tablas con “—” es debido a que no terminaron su ejecución en 384 horas.

Al aumentar las ventanas de optimización disminuye considerablemente el porcentaje de error. Para el escenario 1, al optimizar con una ventana de tamaño 1 el error es relativamente pequeño (depende de la aplicación), mientras que para el escenario 2 es relativamente grande. La diferencia grande en el error para la optimización con una ventana de tamaño 1 en ambos escenarios se debe a que para el escenario 2 se necesita más información para tratar el problema. Al aumentar de una ventana de optimización a tamaño 2 el error para el escenario 2 disminuye significativamente, pero a un costo de un incremento considerable en el tiempo de ejecución. Esto por el hecho de que se cuenta con mayor información del futuro.

En general, al aumentar las ventanas de optimización el espacio de búsqueda aumenta exponencialmente y al aumentar el espacio de búsqueda aumenta el tiempo de ejecución. Este crecimiento es debido a que la temporalidad no es manejada de manera natural por los algoritmos de la familia *DCOP*. Como ejemplo, el número de combinaciones posibles para la optimización con ventana de tamaño 1 es 800,000 (domino celdas de combustible 80 por dominio de carga de batería 200 por dominio de descarga de batería 50) mientras que para la optimización con ventanas de tamaño 2 son 640,000,000,000 (se tiene el doble de variables).

Otra de las medidas de comparación es el promedio de mensajes intercambiados entre los agentes durante la ejecución de cada algoritmo. Además del promedio de mensajes intercambiados entre los agentes es importante saber en promedio que tamaño tienen estos mensajes. Esto debido a que los algoritmos pueden intercambiar muchos mensajes pequeños y otros pocos mensajes muy grandes.

En cuestión del número y tamaño de los mensajes enviados entre agentes para ciertos algoritmos no aumentan en gran medida mientras que para otros sí. Como ejemplo, *DPOP* y *MB-DPOP* fueron elegidos debido a que el número y tamaño de los mensajes que envían es pequeño. A pesar de que estos algoritmos envían pocos mensajes, el número de cálculos realizados es grande.

Algo que se puede observar en los resultados mostrados es que no hay relación entre el número de mensajes y el tiempo de ejecución. Lo anterior se debe a la naturaleza del algoritmo, algunos algoritmos realizan sus cálculos y después envían mensajes para que otros realicen sus cálculos, mientras que otros envían sus mensajes y después realizan los cálculos y solamente realizan este proceso pocas veces. Otro punto importante a resaltar es que al no tratar la temporalidad de manera natural los algoritmos de la familia *DCOP*, cuando se intenta agregar esta al problema, este crece de manera exponencial.

Uno de los puntos principal a resaltar es que para el escenario 1 al utilizar ventanas de tamaño

3 se puede alcanzar el óptimo. Con esto se puede decir que para este escenario la información necesaria para alcanzar el óptimo global se obtiene ventanas de tamaño 3.

Tabla 6.4: Medidas de comparación de optimización a largo plazo utilizando ventanas de optimización para el escenario 1. El tiempo de ejecución mostrado es en segundos, mientras el tamaño de mensajes es presentado en megabytes.

Error								
Ventana	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD
Tamaño 1	6.34 %	0.0	3.64 %	0.0	3.64 %	0.0	3.64 %	0.0
Tamaño 2	8.30 %	8.3	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0
Tamaño 3	0 %	—	—	—	—	—	—	—

Tiempo								
Tamaño								
Tamaño 1	8.1	0.1413	17.3	0.2218	17.2	0.1941	12.3	0.2815
Tamaño 2	2101.3	13.3349	176.6	0.5569	296.7	56.9301	2741.2	137.5269
Tamaño 3	757754.7	—	—	—	—	—	—	—

Número de mensajes								
Tamaño								
Tamaño 1	52785	5.5959	480	0.0	528	0.0	107076	0.0
Tamaño 2	314767	394.3478	628	0.0	688	0.0	5492875	0.0
Tamaño 3	185255337	—	—	—	—	—	—	—

Tamaño de mensajes								
Tamaño								
Tamaño 1	2.6	$3.87 \times 10^{-4}$	0.8	$7.58 \times 10^{-6}$	0.8	$6.75 \times 10^{-6}$	2.6	$8.56 \times 10^{-6}$
Tamaño 2	28	0.0396	2.8	$8.35 \times 10^{-6}$	2.9	$8.98 \times 10^{-6}$	154.7	0.0
Tamaño 3	2351.7	—	—	—	—	—	—	—

En las figuras 6.5, 6.6, 6.7 y 6.8 se muestran las gráficas de las diferentes configuraciones obtenidas con aplicando esta técnica de optimización para el escenario 1.

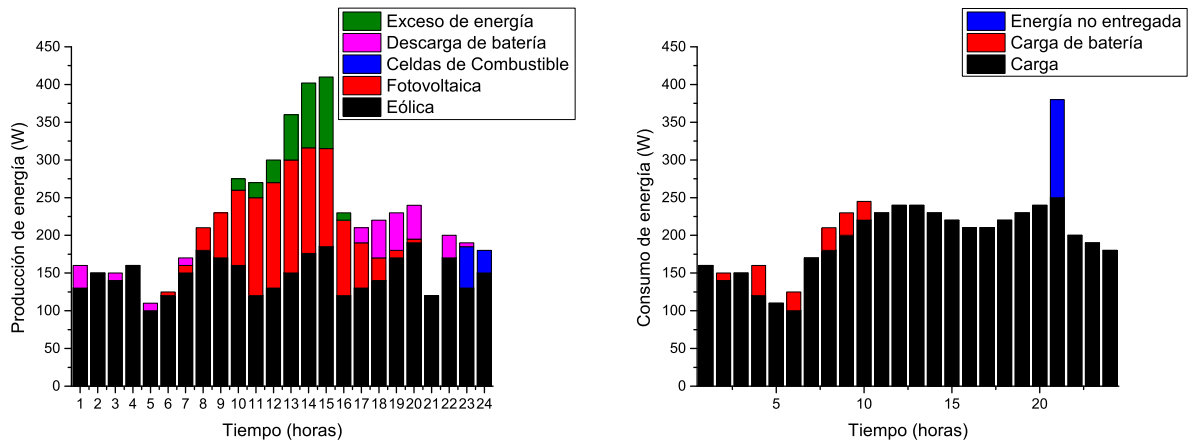


Figura 6.5: Se muestran los resultados de producción y consumo de energía para el escenario 1, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden al algoritmo AFB con ventana de optimización de tamaño 1.

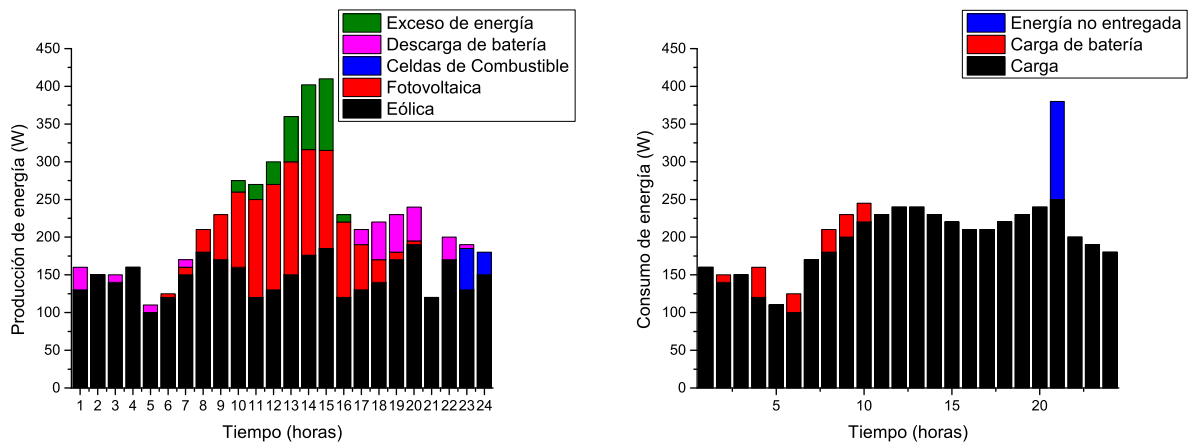


Figura 6.6: Se muestran los resultados de producción y consumo de energía para el escenario 1, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden a los algoritmos DPOP, MB-DPOP y SynchBB utilizando una ventana de optimización de tamaño 1.

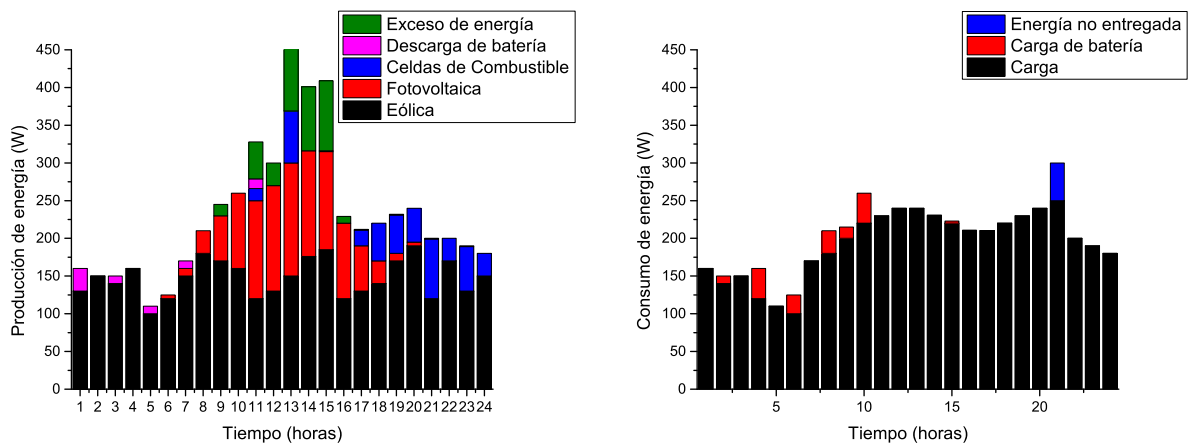


Figura 6.7: Se muestran los resultados de producción y consumo de energía para el escenario 1, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden al algoritmo AFB con ventana de tamaño 2.

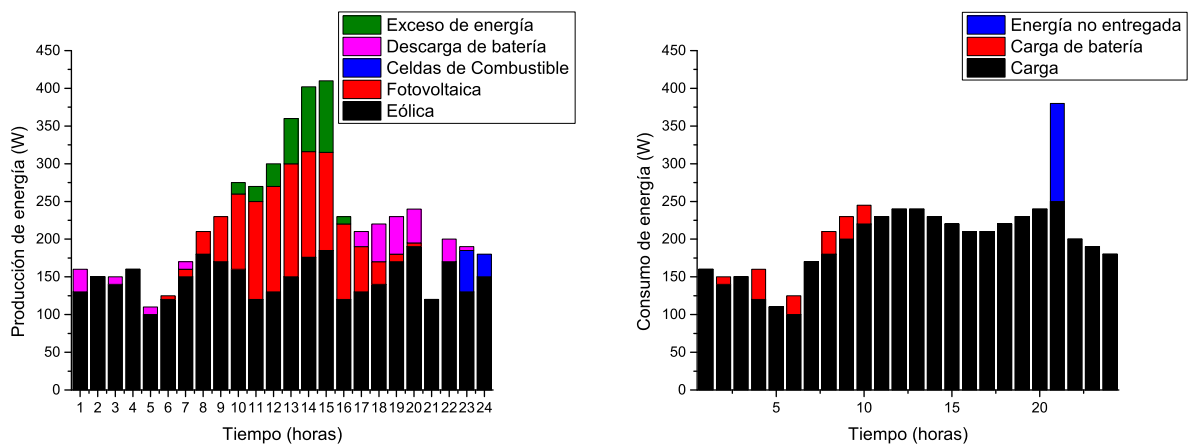


Figura 6.8: Se muestran los resultados de producción y consumo de energía para el escenario 1, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden a los algoritmos DPOP, MB-DPOP y SynchBB utilizando una ventana de optimización de tamaño 2.

Tabla 6.5: Medidas de comparación de optimización a largo plazo utilizando ventanas de optimización para el escenario 2. El tiempo mostrado es en segundos, mientras el tamaño de mensajes es presentado en megabytes.

Error								
Ventana	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD
Tamaño 1	12.3 %	0.0	12.29 %	0.0	12.29 %	0.0	12.29 %	0.0
Tamaño 2	2.14 %	0.0	2.14 %	0.0	2.14 %	0.0	2.14 %	0.0

Tiempo								
Tamaño 1	31.9	0.4410	11.3	0.3035	11.2	0.2124	18.2	0.2602
Tamaño 2	19376.8	795.6866	554.2	4.9534	549.8	5.7209	3551.6	23.1514

Número de mensajes								
Tamaño 1	84615	2.6218	480	0.0	528	0.0	66562	0.0
Tamaño 2	17038724	497.2982	624	0.0	684	0.0	8746644	0.0

Tamaño de mensajes								
Tamaño 1	4.9	$1.80 \times 10^{-4}$	0.4	$7.25 \times 10^{-6}$	0.4	$8.18 \times 10^{-6}$	1.5	$6.37 \times 10^{-6}$
Tamaño 2	1702.2	0.0484	2.5	$1.17 \times 10^{-5}$	2.6	$8.27 \times 10^{-6}$	247.7	0.0

En las figuras 6.9, 6.10 y 6.11 se muestran las gráficas de las diferentes configuraciones obtenidas con aplicando esta técnica de optimización para el escenario 2.

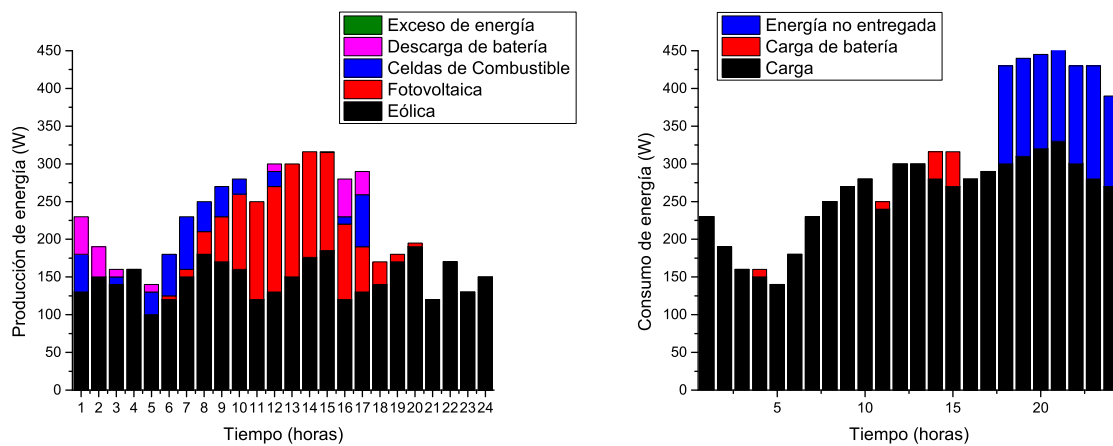


Figura 6.9: Se muestran los resultados de producción y consumo de energía para el escenario 2, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden a los algoritmo AFB con ventana de tamaño 1.

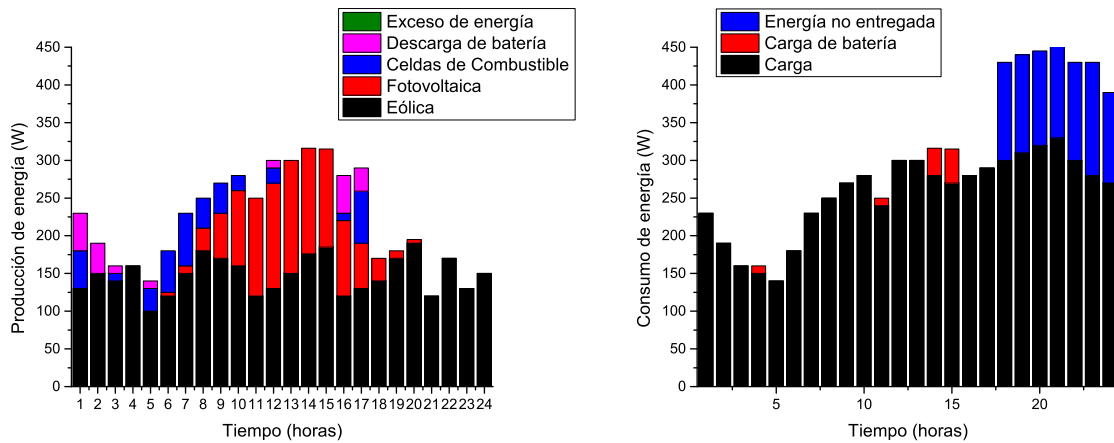


Figura 6.10: Se muestran los resultados de producción y consumo de energía para el escenario 2, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden a los algoritmos DPOP, MB-DPOP y SynchBB utilizando una ventana de optimización de tamaño 1.

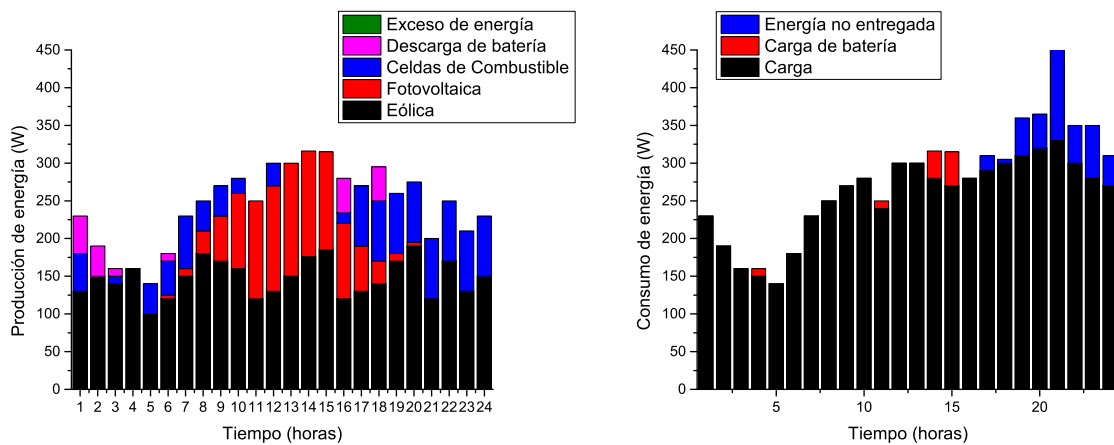


Figura 6.11: Se muestran los resultados de producción y consumo de energía para el escenario 2, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden a los algoritmos AFB, DPOP, MB-DPOP y SynchBB utilizando una ventana de optimización de tamaño 2.

De la implementación del modelo descentralizado basado en división del problema general en subproblemas (ventanas de optimización) se obtienen las siguientes conclusiones:

- Si el algoritmo de optimización es exacto, el error decrece al aumentar las ventanas de optimización. Esta disminución se debe a que al aumentar las ventanas de optimización se cuenta con más información del problema. Al contar con mayor información del problema es posible calendarizar los elementos de la microrred no sólo estimando los costos individuales por

período de tiempo, sino que permite tener un objetivo global.

- Si el algoritmo no es exacto como en el caso de AFB, el error aumenta debido a que cae en mínimos locales. Esto se repite a través de los experimentos debido a que su ejecución es consistente (siempre da el mismo resultado).
- El tiempo crece de manera exponencial, una exponencial que tiende a infinito muy rápido. El crecimiento se debe a que al agregar un nuevo horizonte de tiempo aumenta el número de nodos del problema además que las relaciones entre nodos aumenta de manera considerable. Esto se debe a que la representación con temporalidad no es natural en la representación de los problemas de optimización con restricciones distribuido.
- El número de mensajes intercambiados entre agentes aumenta al agregar más horizontes de tiempo. Al igual que el tiempo, el número de mensajes crece de manera exponencial, la diferencia principal es que la rapidez de crecimiento depende del algoritmo, esto debido a que algunos algoritmos envían muchos mensajes pequeños y otros pocos mensajes muy grandes.
- El tamaño de los mensajes al igual que el número de mensajes crece, la diferencia principal recae en que esté tamaño crece en mayor medida.

## 6.2.2 Solución descentralizada usando búsqueda primero el mejor

Para estos experimentos se utilizó la búsqueda primero el mejor. La metodología utilizada se detalla en la sección 5.2. Se inicializarán todas las variables y a cada una se le asigno un algoritmo de búsqueda primero el mejor, que realiza particiones al dominio, se seleccionan 2 valores y se envían al algoritmo *DCOP*, estos pasos se realizan hasta que no existan más particiones en ninguna de las variables. La partición del dominio se realiza de la siguiente manera: se divide el dominio de la variable en dos partes iguales, de cada partición se selecciona el valor medio y en caso de ser repetición se recibe como entrada un valor correspondiente a una partición y esa partición se toma como el dominio de la variable.

El concentrado de los resultados obtenidos para el escenario 1 se muestra en la tabla 6.6. En la tabla 6.7 se muestra el concentrado de los resultados obtenidos para el escenario 2. Se realizó la experimentación para comparar el rendimiento de la propuesta en cuestión de tiempo de ejecución con los algoritmos mencionados anteriormente.

Al utilizarse está metodología para la partición del dominio es muy fácil caer en mínimos locales. Por lo antes mencionado se puede observar en los resultados que al aumentar las ventanas de optimización el tiempo no aumenta mucho pero el error si aumenta.

En las figuras 6.12, 6.13 y 6.14 se muestran las gráficas de las diferentes configuraciones obtenidas con aplicando esta técnica de optimización para el escenario 1.



Tabla 6.6: Medidas de comparación de optimización a largo plazo con búsqueda primero el mejor para el escenario 1. El tiempo de ejecución mostrado es en segundos, mientras el tamaño de mensajes es presentado en megabytes.

Error									
Ventana	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD	
Tamaño 1	0.42 %	0.0	0.64 %	0.0	0.64 %	0.0	0.64 %	0.0	
Tamaño 2	3.39 %	1.9	0.36 %	0.0	0.38 %	0.0	0.42 %	0.0	
Tamaño 3	9.28 %	0.91	9.28 %	1.02	9.28 %	1.02	9.28 %	0.91	

Tiempo									
Tamaño	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD	
Tamaño 1	1.2	0.1289	0.2	0.0517	1.4	0.1318	2.1	0.1552	
Tamaño 2	2.4	0.2530	0.2	0.0372	2.7	0.2574	6.5	0.2389	
Tamaño 3	63.5	1.0635	0.6	0.1011	0.6	0.1080	249.5	13.8779	

Número de mensajes									
Tamaño	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD	
Tamaño 1	149389	2.2495	3060	0.0	149389	2.2495	292197	2.2495	
Tamaño 2	286637	2224.7405	5856	0.0	286637	2224.3051	562421	2224.3051	
Tamaño 3	658099	2810.2592	7446	0.1	7942	0.1	944264	0	

Tamaño de mensajes									
Tamaño	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD	
Tamaño 1	3.9	$1.76 \times 10^{-4}$	0.7	$2.05 \times 10^{-5}$	3.9	$1.76 \times 10^{-4}$	6.6	$1.75 \times 10^{-4}$	
Tamaño 2	8.9	0.0554	1.5	$2.51 \times 10^{-5}$	8.9	0.0554	15.1	0.0554	
Tamaño 3	40.7	0.3582	2.2	$1.64 \times 10^{-4}$	2.3	$1.50 \times 10^{-4}$	25.5	$2.24 \times 10^{-5}$	

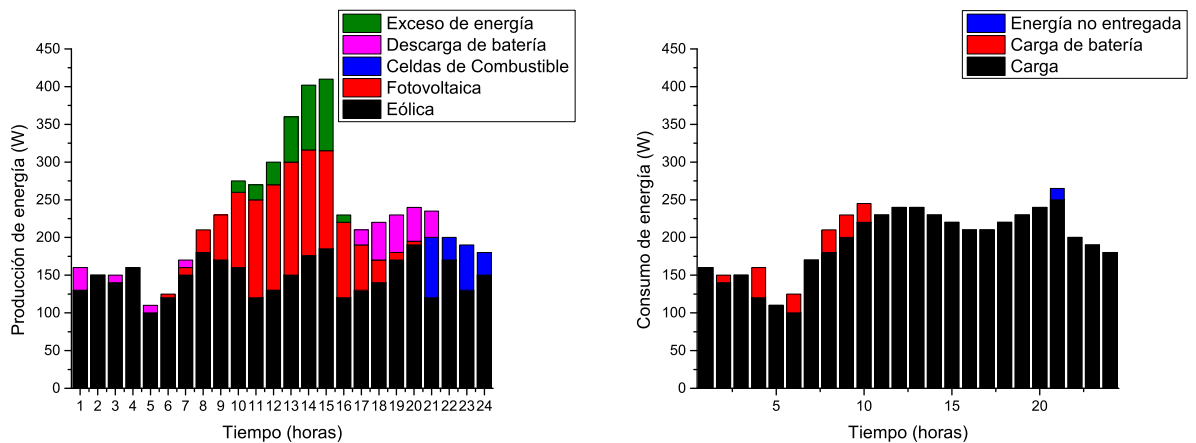


Figura 6.12: Se muestran los resultados de producción y consumo de energía para el escenario 1, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden a los algoritmos DPOP, MB-DPOP y SynchBB con ventana de tamaño 1.

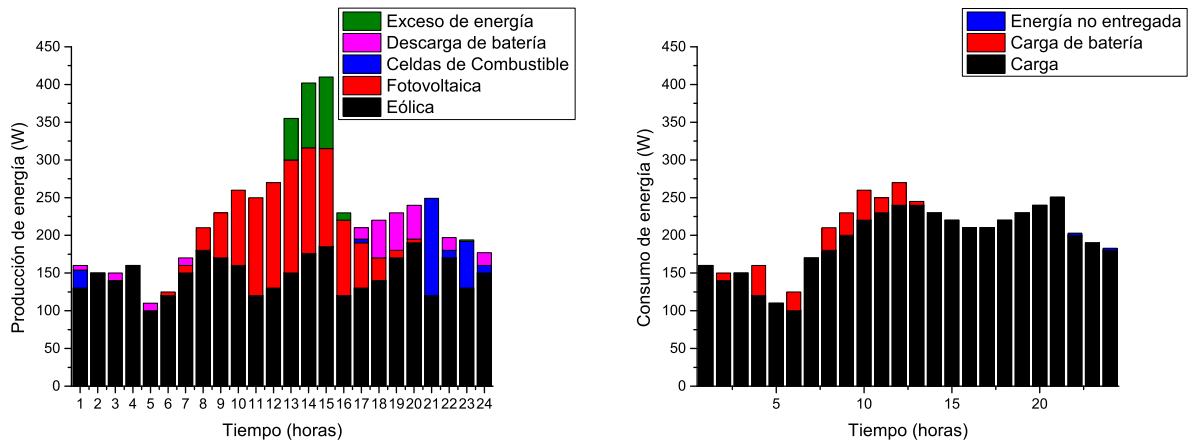


Figura 6.13: Se muestran los resultados de producción y consumo de energía para el escenario 1, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden al algoritmo DPOP utilizando una ventana de optimización de tamaño 2.

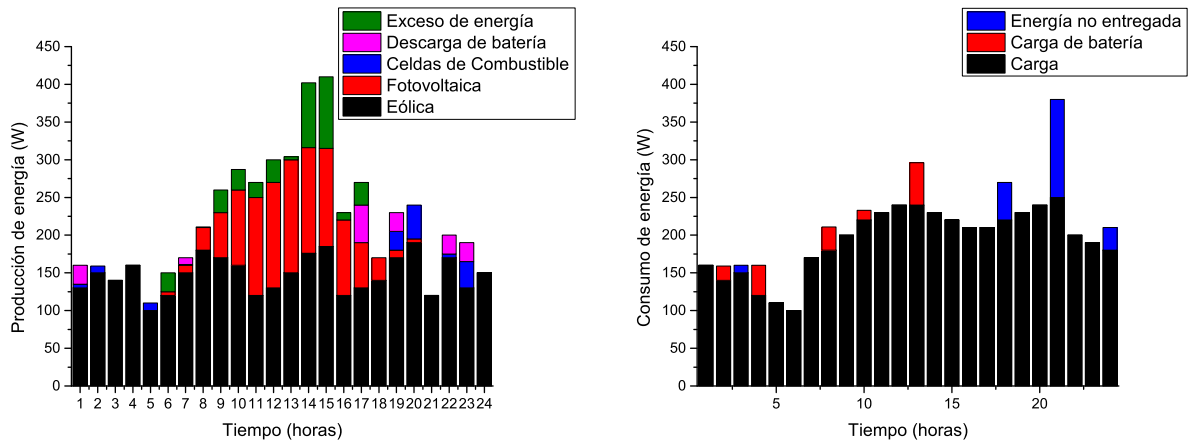


Figura 6.14: Se muestran los resultados de producción y consumo de energía para el escenario 1, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden a los algoritmos DPOP, MB-DPOP y SynchBB utilizando una ventana de optimización de tamaño 3.

Tabla 6.7: Medidas de comparación de optimización a largo plazo para el escenario 2 con búsqueda primero el mejor. El tiempo de ejecución mostrado es en segundos, mientras el tamaño de mensajes es presentado en megabytes.

Error									
Ventana	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD	
Tamaño 1	13.62 %	0.0	16.77 %	0.0	16.77 %	0.0	16.77 %	0.0	
Tamaño 2	9.75 %	9.4746	5.12 %	0.0	5.12 %	0.0	5.12 %	0.0	
Tamaño 3	9.95 %	0.0	9.96 %	0.0	9.96 %	0.0	9.95 %	0.0	
Tiempo									
Tamaño 1	1.8	1.3761	0.3	0.1212	0.3	0.1363	1.2	0.3040	
Tamaño 2	2.9	2.0807	0.3	0.4139	0.3	0.1047	2.1	0.4151	
Tamaño 3	324.7	2.1545	0.8	0.0852	0.8	0.0171	2002.6	54.5778	
Número de mensajes									
Tamaño 1	150288	7.6079	3220	0.0	3542	0.0	150484	0.0	
Tamaño 2	249875	873.8785	5088	0.0	5562	0.0	239769	0.0	
Tamaño 3	1196487	2741.1896	7474	0.0	7979	0.0	5318396	0.0	
Tamaño de mensajes									
Tamaño 1	3.2	$5.21 \times 10^{-4}$	0.8	$2.11 \times 10^{-5}$	0.8	$2.14 \times 10^{-5}$	0.1	$1.93 \times 10^{-5}$	
Tamaño 2	7.0	0.0508	1.3	$2.41 \times 10^{-5}$	1.4	$2.65 \times 10^{-5}$	5.3	$2.27 \times 10^{-4}$	
Tamaño 3	107.1	0.3509	2.2	$2.45 \times 10^{-5}$	2.3	$2.44 \times 10^{-5}$	152.6	$1.20 \times 10^{-5}$	

En las figuras 6.15, 6.17 y 6.17 se muestran las gráficas de las diferentes configuraciones obtenidas con aplicando esta técnica de optimización para el escenario 2.

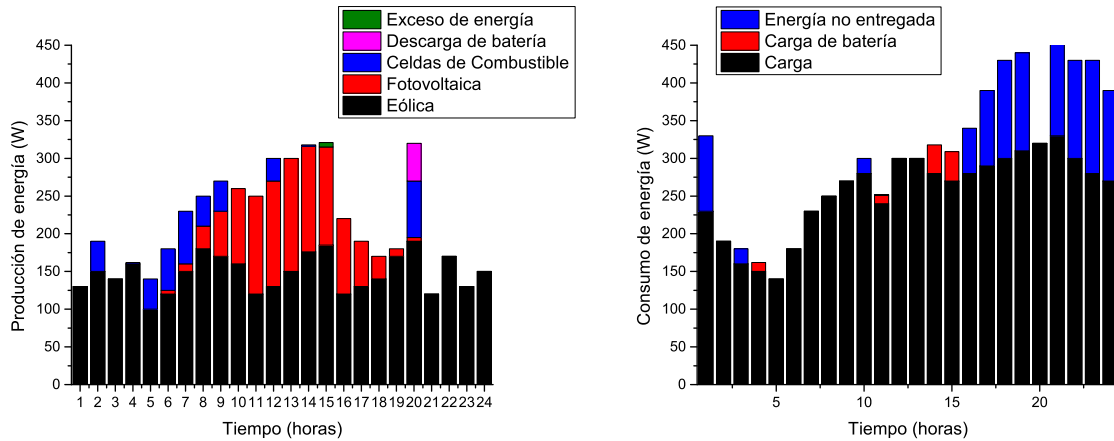


Figura 6.15: Se muestran los resultados de producción y consumo de energía para el escenario 2, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden a los algoritmos DPOP, MB-DPOP y SynchBB con ventana de tamaño 1.

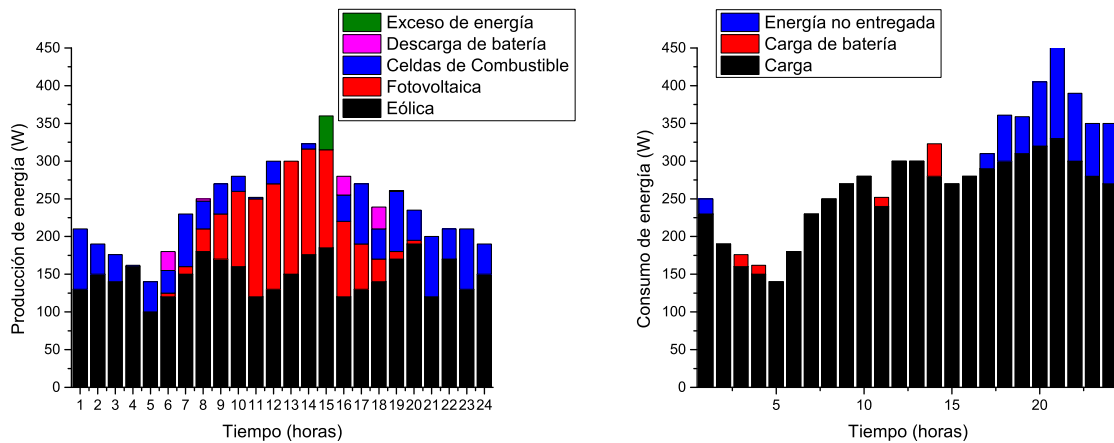


Figura 6.16: Se muestran los resultados de producción y consumo de energía para el escenario 2, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden al algoritmos DPOP, MB-DPOP y SynchBB utilizando una ventana de optimización de tamaño 2.

Sólo se realizaron experimentos hasta ventanas de tamaño 3 debido a que el error aumenta cuando el número de ventanas de optimización crece. Con la optimización descentralizada utilizando técnicas de *primero el mejor* para la reducción de los dominios de las variables en conjunto con las ventanas de optimización se puede concluir lo siguiente:

- Los mínimos locales individuales ayudan a encontrar mejores soluciones que al utilizar mínimos locales colectivos. Con un elemento esta búsqueda ayuda a caer en mínimos locales que

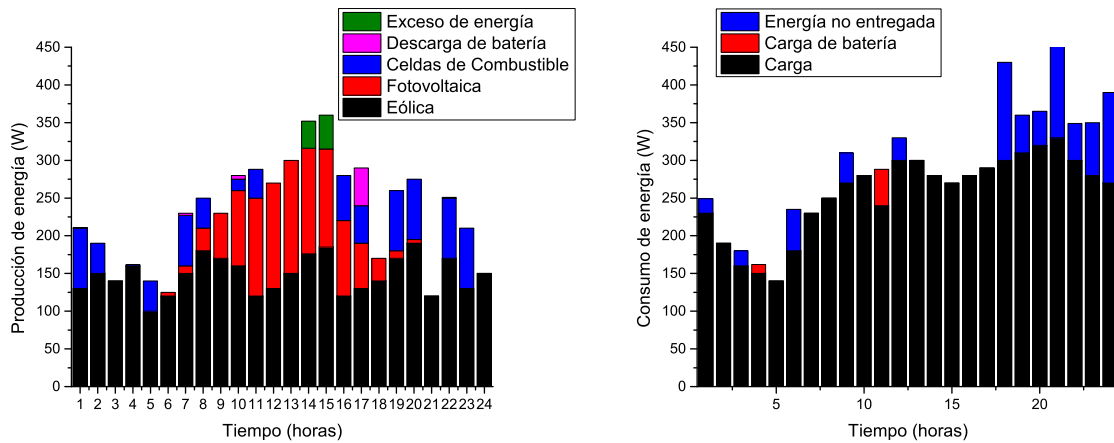


Figura 6.17: Se muestran los resultados de producción y consumo de energía para el escenario 2, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden al algoritmos DPOP y MB-DPOP utilizando una ventana de optimización de tamaño 3.

los algoritmos de optimización óptimos no llegan, esto debido a que este realiza búsquedas y se desvía por donde ve un buen camino y al final eso ayuda a la solución global.

- El error aumenta con el aumento de ventanas de optimización debido a que al tener demasiadas variables se tienen demasiadas combinaciones que nos llevan a mínimos locales que parecen buenos pero no lo son.
- El tiempo de optimización se reduce drásticamente en comparación de la optimización con sólo ventanas, esto se debe a que se optimiza muchas veces pero con pocos elementos en el dominio de cada variable. Otra cosa a resaltar es que el tiempo ya no crece de manera exponencial agresiva, sino que crece de manera mucho más lenta.
- En cuestión a los mensajes intercambiados entre agentes, el número de mensajes es mayor al sólo utilizar ventanas de optimización, esto se debe a que se optimiza con dominios pequeños pero esta optimización se realiza múltiples veces, es decir, se particiona el dominio y se optimiza y esto se realiza hasta que no existen más particiones. Al aumentar el tamaño de las ventanas de optimización el número de mensajes no crece de manera agresiva pero crece a mayor medida que al sólo utilizar ventanas de optimización.

### 6.2.3 Solución descentralizada usando una heurística para guiar la búsqueda

En esta sección se muestran los resultados utilizando una heurística primero el mejor que da como salida el punto inicial y de referencia para  $A^*$ . Los resultados presentados son para diferentes ventanas de optimización a largo plazo. El objetivo de estos experimentos es mostrar como se comportan la solución descentralizada usando sólo ventanas de optimización contra la propuesta usando heurística. Se usan cuatro medidas de comparación: *error*, *tiempo de ejecución*, *número de mensajes intercambiados* y *tamaño promedio de los mensajes*. Se realizaron 100 ejecuciones de cada experimento.

Debido al aumento del error en la propuesta mencionada en la sección 5.2 se decidió utilizar una ventana de optimización para la búsqueda primero el mejor y mejorar esa solución con más ventanas.

El concentrado de los resultados de error y tiempo de ejecución para el escenario 1 se muestra en la tabla 6.8. En la tabla 6.9 se muestra el concentrado de los resultados de error y tiempo para el escenario 2.

En las tablas mostradas se puede observar que al aumentar el número de horizontes de tiempo se reduce el error. Para el escenario 1 con ventanas de optimización de tamaño 3 se puede llegar al valor óptimo, sin embargo, al seguir aumentando los horizontes de optimización aumenta el espacio de búsqueda por lo que el error aumenta. Para el escenario 2 al llegar al ventanas de tamaño 3 se alcanza el mínimo error y se mantiene al seguir aumentando las ventas de optimización.

Al aumentar las ventanas de ejecución el tiempo aumenta de manera considerable. Al llegar a ventana de tamaño 6, el tiempo de ejecución estalla. A pesar de que el tiempo de ejecución aumenta considerablemente este sigue estando en el orden de las horas lo cual permite que esta solución siga siendo viable esto debido a que se optimiza por periodos de 24 horas. Como se puede observar en las tablas anteriores para el escenario 1 con ventana de tamaño 3 existe un comportamiento anormal, esto se debe a que con una ventana de tamaño 3 no puede encontrar fácilmente una solución mejor. Dicho en otras palabras lo anterior,  $A^*$  al no encontrar mejor solución para muy pronto. Por la misma razón mencionada anteriormente para el escenario 2 con optimización a 6 horizontes se ejecuta en menor tiempo que para el escenario 1.

El concentrado de los resultados sobre los mensajes intercambiados se muestran en la tabla 6.10 para el escenario 1 y en la tabla 6.11 para el escenario 2. El tamaño de los mensajes se mantiene relativamente constante debido a que siempre se ejecuta la optimización para el tiempo 1 y esa es la entrada para  $A^*$ .

Tabla 6.8: Error en predicciones y tiempo de ejecución usando heurística para guiar una búsqueda para el escenario 1. El tiempo de ejecución mostrado es en segundos.

Error								
Ventana	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD
Tamaño 1	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0
Tamaño 2	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0
Tamaño 3	0 %	0.0	0 %	0.0	0 %	0.0	0 %	0.0
Tamaño 4	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0
Tamaño 6	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0

Tiempo								
Tamaño 1	11.46	0.7606	3.47	0.6304	2.84	0.1297	9.46	0.2393
Tamaño 2	11.31	0.4886	3.76	0.7895	2.84	0.1150	9.42	0.2138
Tamaño 3	17.35	0.6774	12.19	1.0598	12.07	0.5959	17.85	1.3758
Tamaño 4	11.05	0.3296	3.69	0.9231	3.01	0.1323	9.41	0.2557
Tamaño 6	58878.79	113.4864	54860.71	313.5843	55949.12	1199.4771	55872.33	272.5515

Tabla 6.9: Error en predicciones y tiempo de ejecución usando heurística para guiar una búsqueda para el escenario 2. El tiempo ejecución mostrado es en segundos.

Error								
Ventana	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD
Tamaño 1	3.09 %	0.0	3.09 %	0.0	3.09 %	0.0	3.09 %	0.0
Tamaño 2	3.09 %	0.0	3.09 %	0.0	3.09 %	0.0	3.09 %	0.0
Tamaño 3	2.21 %	0.0	2.21 %	0.0	2.21 %	0.0	2.21 %	0.0
Tamaño 4	2.21 %	0.0	2.21 %	0.0	2.21 %	0.0	2.21 %	0.0
Tamaño 6	2.21 %	0.0	2.21 %	0.0	2.21 %	0.0	2.21 %	0.0

Tiempo								
Tamaño 1	9.11	0.5987	3.17	0.6164	2.89	0.1172	8.89	0.4536
Tamaño 2	9.39	0.4768	3.28	0.7596	2.98	0.1448	9.15	0.4843
Tamaño 3	16.52	0.8615	11.64	1.2002	11.44	0.3791	17.39	0.9736
Tamaño 4	72.56	3.2392	67.78	1.3468	64.85	1.4102	73.64	3.7596
Tamaño 6	16851.21	272.8472	14338.33	32.9307	15589.51	1762.9741	17184.52	2671.1055

En las figuras 6.18 y 6.19 se muestran las gráficas de las diferentes configuraciones obtenidas con aplicando esta técnica de optimización para el escenario 1.

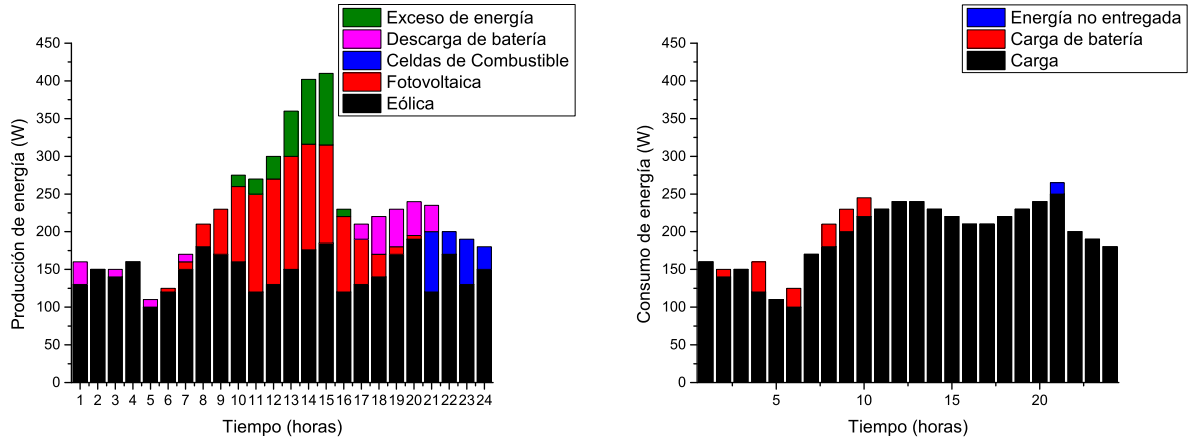


Figura 6.18: Se muestran los resultados de producción y consumo de energía para el escenario 1, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden al algoritmo AFB, DPOP, MB-DPOP y SynchBB con ventanas de optimización de tamaño 1, 2, 4 y 6.

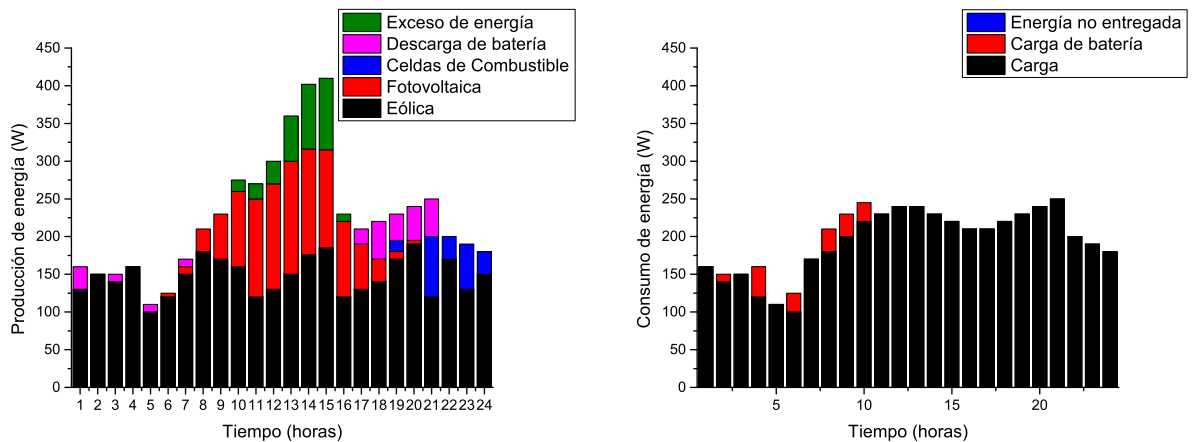


Figura 6.19: Se muestran los resultados de producción y consumo de energía para el escenario 1, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden al algoritmos AFB, DPOP, MB-DPOP y SynchBB utilizando una ventana de optimización de optimización de tamaño 3.



En las figuras 6.20 y 6.21 se muestran las gráficas de las diferentes configuraciones obtenidas con aplicando esta técnica de optimización para el escenario 2.

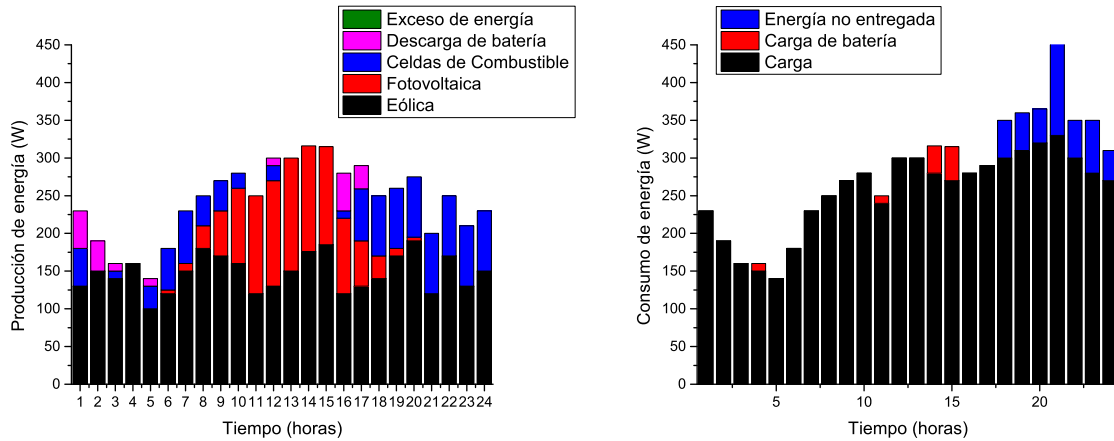


Figura 6.20: Se muestran los resultados de producción y consumo de energía para el escenario 2, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden al algoritmo AFB, DPOP, MB-DPOP y SynchBB con ventanas de optimización de tamaño 1 y 2.

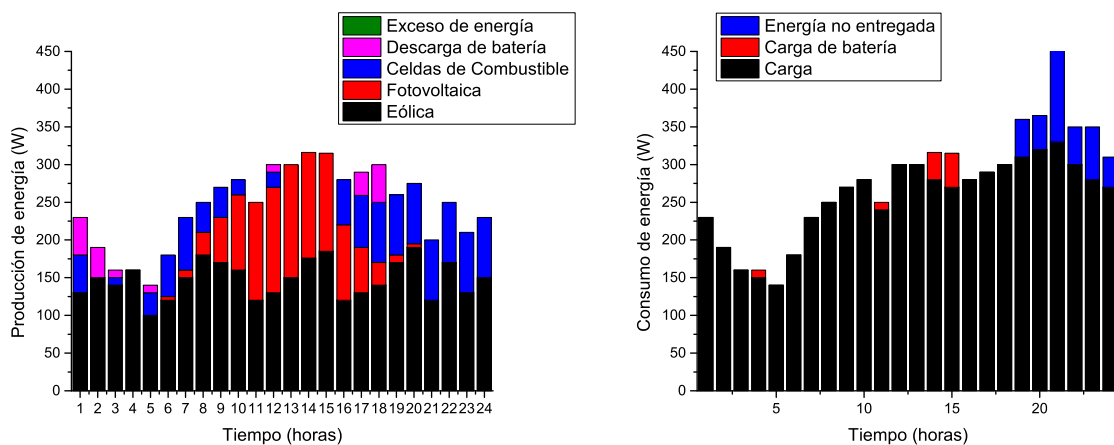


Figura 6.21: Se muestran los resultados de producción y consumo de energía para el escenario 2, tomando en cuenta que el estado inicial de la batería es de 100 W. Estos resultados corresponden al algoritmos AFB, DPOP, MB-DPOP y SynchBB utilizando ventanas de optimización de tamaño 3, 4 y 6.

De las tablas de número y tamaño de mensajes se puede observar que los algoritmos que no son deterministas como *AFB* existen variaciones. Ya que *DPOP* es determinista, siempre tiene el mismo número de mensajes en cada ejecución y por lo tanto el mismo tamaño. Los demás algoritmos al

no ser deterministas en su ejecución, el número de mensajes así como su tamaño es variable.

Tabla 6.10: Promedio del número y tamaño de mensajes intercambiados entre agentes usando heurística para guiar una búsqueda para el escenario 1. El tamaño de mensajes se muestra en megabytes.

Número de mensajes								
Ventana	AFB	STD	DCOP	STD	MB-DPOP	STD	SynchBB	STD
Tamaño 1	149393	5.0598	3060	0.0	6426	0.0	149234	5.0598
Tamaño 2	149393	3.5824	3060	0.0	6426	0.0	149234	3.5824
Tamaño 3	149393	0.8283	3060	0.0	6426	0.0	149234	0.0
Tamaño 4	149393	1.3997	3060	0.0	6426	0.0	149234	0.0
Tamaño 6	149393	1.8522	3060	0.0	6426	0.0	149234	0.0

Tamaño de mensajes								
Tamaño 1	3.89	$3.29 \times 10^{-4}$	0.78	$1.80 \times 10^{-5}$	1.63	$3.27 \times 10^{-5}$	4.37	$3.27 \times 10^{-5}$
Tamaño 2	3.89	$2.39 \times 10^{-4}$	0.78	$1.61 \times 10^{-5}$	1.63	$2.44 \times 10^{-5}$	4.37	$2.44 \times 10^{-5}$
Tamaño 3	3.89	$6.47 \times 10^{-5}$	0.78	$1.61 \times 10^{-5}$	1.63	$2.76 \times 10^{-5}$	4.37	$3.54 \times 10^{-5}$
Tamaño 4	3.89	$9.80 \times 10^{-5}$	0.78	$1.99 \times 10^{-5}$	1.63	$2.60 \times 10^{-5}$	4.37	$3.50 \times 10^{-5}$
Tamaño 6	3.89	$1.15 \times 10^{-4}$	0.78	$2.41 \times 10^{-5}$	1.63	$1.08 \times 10^{-5}$	4.37	$1.87 \times 10^{-5}$

Tabla 6.11: Promedio del número y tamaño de mensajes intercambiados entre agentes usando heurística para guiar una búsqueda para el escenario 2. El tamaño de mensajes se muestra en megabytes.

Número de mensajes								
Ventana	AFB	STD	DPOP	STD	MB-DPOP	STD	SynchBB	STD
Tamaño 1	159919	1.9090	3360	0.0	7056	0.0	163192	1.9090
Tamaño 2	159919	1.6762	3360	0.0	7056	0.0	163192	1.6762
Tamaño 3	159919	1.1430	3360	0.0	7056	0.0	163192	0.0
Tamaño 4	159919	6.4230	3360	0.0	7056	0.0	163192	0.0
Tamaño 6	159919	1.2849	3360	0.0	7056	0.0	163192	0.0

Tamaño de mensajes								
Tamaño 1	4.13	$1.33 \times 10^{-4}$	0.86	$2.21 \times 10^{-5}$	1.78	$1.29 \times 10^{-5}$	4.77	$1.33 \times 10^{-4}$
Tamaño 2	4.13	$1.14 \times 10^{-4}$	0.86	$1.98 \times 10^{-5}$	1.78	$1.14 \times 10^{-5}$	4.77	$1.14 \times 10^{-4}$
Tamaño 3	4.13	$2.23 \times 10^{-4}$	0.86	$2.18 \times 10^{-5}$	1.78	$2.82 \times 10^{-5}$	4.77	$3.34 \times 10^{-5}$
Tamaño 4	4.13	$4.46 \times 10^{-4}$	0.86	$2.24 \times 10^{-4}$	1.78	$3.48 \times 10^{-5}$	4.77	$4.86 \times 10^{-5}$
Tamaño 6	4.13	$1.15 \times 10^{-4}$	0.86	$1.24 \times 10^{-5}$	1.78	$3.42 \times 10^{-5}$	4.77	0.0

Al utilizar ventanas de optimización en conjunto de una búsqueda guiada  $A^*$  apoyada por una heurística *primero el mejor* se puede concluir lo siguiente:

- Al aumentar el número de ventanas el error disminuye, se estanca y vuelve a subir. Esto se debe a que en  $A^*$  cada variable se considera una dimensión en el espacio de búsqueda y

con muchas dimensiones le cuesta moverse al punto óptimo, ya que el camino se vuelve más largo. Como ejemplo al utilizar ventanas de tamaño 1 se tienen 3 dimensiones de búsqueda (celdas de combustible, carga y descarga de batería), al tener ventanas de tamaño 2 se tienen 6 dimensiones (existen 2 variables por cada agente) y para ventanas de tamaño 3 existen 9 dimensiones (3 variables por cada agente).

- Al contar con menos recursos se requiere un mayor número de ventanas de optimización. Lo anterior quiere decir que si no se cuentan con los recursos suficientes para satisfacer la demanda en todos los tiempos se necesita mayor visión del sistema para poder aplicar los recursos con los que se cuenta de manera efectiva.
- El tiempo de optimización crece de manera suave lo que permite poder agregar varias dimensiones al espacio de búsqueda, es decir, se pueden agregar más tiempos para poder mejorar su solución.
- El número de mensajes intercambiado entre agentes así como el tamaño de estos se mantiene a través del incremento de tiempos de optimización debido a que lo único cambiante es la implementación de  $A^*$ .

### 6.3 Resumen

Este capítulo se enfocó en mostrar los resultados de las diversas técnicas propuestas para resolver el problema de calendarización de ciclos de encendido y apagado de microgeneradores.

La técnica de ventanas de optimización resuelve de manera natural el problema. La desventaja que tiene este enfoque es que después de cierto número de ventanas el tiempo computacional es mucho. Para el escenario 1 optimizando con ventanas de tamaño 3 se puede observar que se llega a la configuración óptima. Con esto se puede decir que no es necesario implementar las 24 horas para encontrar el valor óptimo. La cantidad de información (tamaño de las ventanas de optimización) necesaria para encontrar el valor óptimo es diferente para cada problema.

Con el enfoque utilizando búsqueda *primero el mejor* se reduce considerablemente el tiempo de optimización dando una configuración cercana al óptimo. La desventaja más grande de este enfoque es que al aumentar en número de ventanas se corre mayor riesgo de caer en un mínimo/máximo local, es decir, que al aumentar el número de ventanas el error aumenta.

El enfoque utilizando una búsqueda guiada  $A^*$ , ayuda a mejorar los mínimos/máximos locales encontrados por una búsqueda *primero el mejor* y con esto poder llegar al óptimo.

La mejor solución es utilizar búsqueda *primero el mejor* y  $A^*$ . Esto es debido a que  $A^*$  permite fortalecer las debilidades de la búsqueda *primero el mejor*. En este enfoque al aumentar el número de ventanas el error disminuye, se estanca y vuelve a subir. Esto se debe a que en  $A^*$  cada variable

se considera una dimensión en el espacio de búsqueda y con muchas dimensiones le cuesta moverse al punto óptimo, ya que el camino se vuelve más largo.

En los resultados mostrados el error en algunos casos aumenta al aumentar las ventanas de optimización, este aumento se debe a diferentes razones. Al utilizar de ventanas de optimización se debe a que el algoritmo que presenta este comportamiento no es exacto pero es consistente, es decir; su ejecución es determinista. Al utilizar búsqueda primero el mejor este comportamiento se debe a que al aumentar el número de variables se tiende a caer en mínimos locales. Para el caso de búsqueda  $A^*$  este comportamiento se debe a que la búsqueda se realiza en varias dimensiones.

Otras conclusiones se enlistan a continuación:

- El problema a pesar de proponer soluciones que reduzcan el espacio de búsqueda sigue siendo exponencial, por lo que al llegar a cierto número de ventanas de optimización el problema se vuelve intratable.
- Teniendo las especificaciones y predicciones de la generación y consumo así como las especificaciones de almacenamiento se puede determinar si es suficiente la capacidad de producción y almacenamiento para satisfacer la demanda con la que se cuenta, esto como medida preventiva, es decir, antes de adquirir los elementos que conformen la microrred.
- Con la optimización de los elementos controlables de la microrred además se optimiza su uso, esto quiere decir que los utilizamos solamente lo necesario.
- Con la calendarización de los elementos de la microrred se cuenta además con los elementos que sirvan como antecedentes para poder dar mantenimiento preventivo así como por desgaste.

# 7

## Conclusiones y trabajo futuro

En este capítulo final se dará de manera resumida las conclusiones y contribuciones de este trabajo de tesis así como el trabajo futuro.

### 7.1 Conclusiones

La calendarización de ciclos de encendido de los generadores es un problema presente en todas las redes de distribución de energía. Con el aumento de consumo de energía se deben agregar más generadores a la red. Debido a la inserción indiscriminada de generadores se ha recurrido a crear pequeñas redes autosuficientes que pueden estar o no conectadas a la red de distribución. Con la creación de estas redes se está descentralizando el enfoque de distribución de energía. En las microrredes también existe el problema de calendarización de ciclos de encendido y apagado de sus microgeneradores.

Los enfoques centralizados permiten tener un mayor control de las redes de distribución ya que se tiene una visión completa del sistema. La desventaja de estos enfoques es que al crecer las redes se vuelve imposible realizar la administración así como si existe falla en el sistema central todo el sistema falla. Por las desventajas mencionadas anteriormente, en las microrredes existe la tendencia a la administración descentralizada. Además la transición de modelos centralizados a modelos descentralizados de optimización es un hecho en los últimos años es debido a que los problemas han ido creciendo de manera significativa y traen consigo el requerimiento de mayor procesamiento computacional.

Muchos problemas en sistemas multiagentes pueden ser descritos como Problemas de Satisfacción de Restricciones Distribuidas, donde el objetivo es encontrar una asignación para las variables que satisfagan todas las restricciones de los agentes. En general, los problemas de optimización son mucho más difíciles de resolver que los problemas de satisfacción de restricciones, el objetivo es no sólo encontrar una solución, sino la mejor. Esto requiere mayor exploración del espacio de búsqueda.

El objetivo de este trabajo de tesis es utilizar un modelo descentralizado para la calendarización de ciclos de encendido y apagado de un conjunto de microgeneradores. Se realizaron tres propuestas: ventanas de optimización, una búsqueda primero el mejor para reducir el dominio de las variables y una modificación a la búsqueda  $A^*$  para mejorar la solución obtenida. Al utilizar en conjunto las tres propuestas se puede optimizar a largo plazo encontrando un balance entre tiempo de ejecución y error relativo.

Debido a que los algoritmos multiagentes actuales no integran la temporalidad se propone una arquitectura de optimización a largo plazo que puede ser usada por cualquier algoritmo de la familia de DCOP. Esta arquitectura que se ha llamado *optimización por ventanas* permite representar problemas que incluyen temporalidad. Esta arquitectura permite optimizar un problema completo o por medio de subproblemas. Cuando un problema es muy grande se propone optimizar por medio de subproblemas (ventanas de optimización), con esto se puede obtener una solución aproximada al valor óptimo pero con menor costo computacional. La utilización de este enfoque permite encontrar una relación entre el tamaño de las ventanas de optimización y la solución óptima, esto debido a que la para realizar una optimización global (en el caso de esta tesis 24 horizontes) no es necesario utilizar toda la información (ventana de tamaño 24) para encontrar el valor óptimo.

Como existen problemas de optimización con una gran cantidad de variables se propone una heurística para reducir el dominio de las variables. Esta heurística consiste en el particionado del dominio de las variables con elección primero el mejor. Este particionado permite que se reduzca de manera significativamente el número de combinaciones de las variables del problema. Además se obtienen reducción del tiempo computacional así como el número y tamaño de los mensajes enviados. Este enfoque permite la utilización de ventanas de optimización de mayor tamaño aunque no en todos los problemas es conveniente usar ventanas de optimización grandes con este enfoque.

La última propuesta es una modificación a la búsqueda guiada  $A^*$  para mejorar la solución obtenida mediante la búsqueda primero el mejor realizadas por ventanas de optimización. Esta implementación permite que una vez obtenida una solución parcial utilizar más información (aumento de las ventanas de optimización) para mejorarla. Con esta implementación se puede obtener mejores resultados con menor costo computacional que el utilizar ventanas de optimización por si solas.

Los tres enfoques juntos permiten obtener resultados comparables con la solución óptima. Además se reduce el tiempo computacional y con esto se permite utilizar ventanas de optimización de mayor tamaño y por consiguiente se cuenta con mayor información global para la optimización.

Otro enfoque de este trabajo se encuentra en la creación de nuevas microrredes. Esto debido a que teniendo un perfil de consumo se pueden realizar las simulaciones antes de crear la infraestructura para crearla de tal manera que siempre satisfaga las necesidades de la demanda.

## 7.2 Contribuciones

Las contribuciones de esta tesis se enlistan a continuación:

1. Un modelo descentralizado para la calendarización de los ciclos de encendido y apagado en un conjunto de microgeneradores.
2. Una heurística para reducir el dominio de las variables.
3. Una modificación a la búsqueda  $A^*$  para mejorar la solución obtenida por algoritmos no exactos de optimización.

## 7.3 Trabajo futuro

De lo realizado en este trabajo de tesis se abren nuevas direcciones hacia donde realizar investigación:

- **En el área de algoritmos**, la implementación de una nueva gama de algoritmos que permitan trabajar con la temporalidad naturalmente. Una posible opción sería un algoritmo eficiente que permita ingresar un problema que incluya temporalidad y se trate de manera secuencial y después optimización conjunta (que optimice todos los horizontes a la vez). Otra opción podría ser el manejo y reducción del dominio del problema y trabajar con secciones de este, que partiendo de esas soluciones parciales (mínimo/máximo local) se pueda partir hasta encontrar la solución general (mínimo/máximo global).
- **En el área de representación DCOP**, crear una representación que permita la creación de problemas distribuidos que incluyan temporalidad de manera compacta y natural. Para lo antes mencionado, se necesitaría la creación de nuevas restricciones que permitan incluir temporalidad en un problema distribuido, en donde se pueda especificar cuáles son y cómo están conectadas las variables a través del tiempo.
- **En el área de los dominios de las variables de un sistema multiagente**, trabajar con nuevas técnicas que ayuden a reducir el dominio en escenarios complicados. Una manera de atacar el problema es realizar heurísticas que permitan ayudar a los algoritmos a encontrar de manera eficiente soluciones óptimas o muy cercanas al óptimo. Otra manera sería implementar algoritmos que permitan reducir la dimensionalidad del dominio con que se trabaja como preprocesamiento antes de ingresar a un algoritmo de optimización.





# Referencias

- Amato, C., Bernstein, D. S. y Zilberstein, S.: 2012, Optimizing Memory-Bounded Controllers for Decentralized POMDPs, *Autonomous Agents and Multi-Agent Systems* **21**(3), 293–320.
- Awad, B., Wu, J. y Jenkins, N.: 2008, Control of Distributed Generation, *e&i Elektrotechnik und Informationstechnik* **125**, 409–414.
- Bäck, T. y Schwefel, H.-P.: 1993, An overview of evolutionary algorithms for parameter optimization, *Evolutionary computation* **1**(1), 1–23.
- Bazmohammadi, N., Karimpour, A. y Bazmohammadi, S.: 2012, Optimal operation management of a microgrid based on MOPSO and Differential Evolution algorithms, *Smart Grids (ICSG), 2012 2nd Iranian Conference on*, pp. 1–6.
- Becker, R., Zilberstein, S. y Lesser, V.: 2004, Decentralized Markov decision processes with event-driven interactions, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems- Volume 1*, IEEE Computer Society, pp. 302–309.
- Bernstein, D. S., Givan, R., Immerman, N. y Zilberstein, S.: 2002, The complexity of decentralized control of Markov decision processes, *Mathematics of operations research* **27**(4), 819–840.
- Bertani, A., Bossi, C., Massucco, S., Spelta, S. y Tivegna, F.: 2004, A Microturbine Generation System for Grid Connected and Islanding Operation, *IEEE PES Power Systems Conference and Exposition*. pp. 822–827.
- Black, J. W. y Ilic, M.: 2002, Demand-Based Frequency Control For Distributed Generation, *Power Engineering Society Summer Meeting, 2002 IEEE*, Vol. 1, pp. 427–432.
- Boutilier, C.: 1996, Planning, Learning and Coordination in Multiagent Decision Processes, *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '96, pp. 195–210.
- Brafman, R. I., Shani, G. y Zilberstein, S.: 2013, Qualitative Planning under Partial Observability in Multi-Agent Domains, *Association for the Advancement of Artificial Intelligence*.
- Brassard, G. y Bratley, P.: 1996, *Fundamentals of Algorithmics*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Chaouachi, A., Kamel, R., Andoulsi, R. y Nagasaka, K.: 2013, Multiobjective Intelligent Energy Management for a Microgrid, *Industrial Electronics, IEEE Transactions on* **60**(4), 1688–1699.
- Chen, C., Duan, S., Cai, T., Liu, B. y Hu, G.: 2011, Smart energy management system for optimal microgrid economic operation, *Renewable Power Generation, IET* **5**(3), 258–267.
- Chen, S. y Gooi, H.: 2011, Jump and Shift Method for Multi-Objective Optimization, *Industrial Electronics, IEEE Transactions on* **58**(10), 4538–4548.

- Colson, C. y Nehrir, M.: 2011, Algorithms for distributed decision-making for multi-agent microgrid power management, *Power and Energy Society General Meeting, 2011 IEEE*, pp. 1–8.
- Dimeas, A. L. y Hatziargyriou, N. D.: 2005, Operation of a Multiagent System for Microgrid Control, *IEEE Transactions on Power Systems* **20**(3), 1447–1455.
- Ehnberg, J. S. G. y Bollen, M. H. J.: 2004, Generation Reliability for Small Isolated Power Systems entirely based on Renewable Sources, *Power Engineering Society General Meeting, 2004. IEEE* pp. 2322–2327.
- GAMS Development Corporation: 2014, GAMS - A User's Guide.  
**URL:** <http://www.gams.com>
- Gershman, A., Meisels, A. y Zivan, R.: 2009, Asynchronous Forward Bounding for Distributed COPs, *Journal of Artificial Intelligence Research* **34**, 61–88.
- Gil, N. J. y Lopez, J. A. P.: 2008, Exploiting Automated Demand Response, Generation and Storage Capabilities for Hierarchical Frequency Control in Islanded Multi-Microgrids, *Proceedings of PSCC2008, 16th Power System Computation Conference* .
- Gmytrasiewicz, P. J. y Doshi, P.: 2005, A framework for sequential planning in multi-agent settings, *Journal of Artificial Intelligence Research (JAIR)* pp. 49–79.
- Goldman, C. V. y Zilberstein, S.: 2004, Decentralized control of cooperative systems: Categorization and complexity analysis, *Journal of Artificial Intelligence Research (JAIR)* **22**, 143–174.
- Hansen, E. A., Bernstein, D. S. y Zilberstein, S.: 2004, Dynamic programming for partially observable stochastic games, *American Association for Artificial Intelligence (AAAI)*, Vol. 4, pp. 709–715.
- Hansen, M. O. L.: 2008, *Aerodynamics of Wind Turbines*, second edn, Earthscan.
- Hart, P., Nilsson, N. y Raphael, B.: 1968, A formal basis for the heuristic determination of minimum cost paths, *Systems Science and Cybernetics, IEEE Transactions on* **4**(2), 100–107.
- Hatziargyriou, N. D. y Meliopoulos, A. P. S.: 2002, Distributed Energy Sources: Technical Challenges, *Power Engineering Society Winter Meeting, 2002. IEEE* **2**, 1017–1022.
- Hirayama, K. y Yokoo, M.: 1997, Distributed Partial Constraint Satisfaction Problem, *Principles and Practice of Constraint Programming-CP97*, Vol. 1330 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 222–236.
- Hiremath, R., Shikha, S. y Ravindranath, N.: 2007, Decentralized Energy Planning; Modeling and Application - a Review, *Renewable and Sustainable Energy Reviews* **11**(5), 729–752.
- Khaligh, A. y Li, Z.: 2010, Battery, Ultracapacitor, Fuel Cell, and Hybrid Energy Storage Systems for Electric, Hybrid Electric, Fuel Cell, and Plug-In Hybrid Electric Vehicles: State of the Art, *IEEE Transactions on Vehicular Technology* **59**(6), 2806–2814.
- Kiehne, H. A.: 2003, *Battery Technology Handbook*, Marcel Dekker, Inc.

- Kuchcinski, K. y Szymanek, R.: 2014, JaCoP - Java Constraint Programming Solver.  
**URL:** <http://www.jacop.eu>
- Lasseter, R. H.: 2007, Microgrids and Distributed Generation, *Journal of Energy Engineering* pp. 144–149.
- Lasseter, R. H. y Paigi, P.: 2004, Microgrid: a Conceptual Solution, *2004 IEEE 35th Annual Power Electronics Specialists Conference* **6**, 4285–4290.
- Le, T., Son, T. C., Pontelli, E. y Yeoh, W.: 2015, Solving distributed constraint optimization problems using logic programming, *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Léauté, T., Ottens, B. y Szymanek, R.: 2009, FRODO 2.0: An Open-Source Framework for Distributed Constraint Optimization, *Proceedings of the IJCAI'09 Distributed Constraint Reasoning Workshop (DCR'09)*, Pasadena, California, USA, pp. 160–164. URL: <http://frodo2.sourceforge.net>.
- Loeliger, H.-A.: 2004, An introduction to factor graphs, *Signal Processing Magazine, IEEE* **21**(1), 28–41.
- Logenthiran, T., Srinivasan, D. y Khambadkone, A. M.: 2011, Multi-agent system for energy resource scheduling of integrated microgrids in a distributed system, *Electric Power Systems Research* **81**(1), 138 – 148.
- López, M. E., Barea, R., Bergasa, L. M. y Escudero, M. S.: 2003, *Visually Augmented POMDP for Indoor Robot Navigation*, Vol. 21, pp. 183–187.
- Marecki, J., Gupta, T., Varakantham, P., Tambe, M. y Yokoo, M.: 2008, Not all agents are equal: scaling up distributed POMDPs for agent networks, *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, International Foundation for Autonomous Agents and Multiagent Systems, pp. 485–492.
- Mathema, P.: 2008, *Optimization of Integrated Renewable Energy System - Micro Grid*, Master's thesis, Tribhuvan University. Kathmandu, Nepal.
- Melo, F. S., Spaan, M. T. J. y Witwicki, S. J.: 2012, QueryPOMDP: POMDP-Based Communication in Multiagent Systems, in M. Cossentino, M. Kaisers, K. Tuyls y G. Weiss (eds), *Multi-Agent Systems*, Vol. 7541 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 189–204.
- Miller, S., Ramchurn, S. D. y Rogers, A.: 2012, Optimal Decentralised Dispatch of Embedded Generation in the Smart Grid, *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* pp. 281–288.
- Milosevic, M. y Andersson, G.: 2005, Generation Control in Small Isolated Power Systems, *Power Symposium, 2005. Proceedings of the 37th Annual North American* pp. 524–529.
- Morais, H., Cardoso, M., Castanheira, L., Vale, Z. y Praca, I.: 2005, A Decision-Support Simulation Tool for Virtual Power Producers, *Future Power Systems, 2005 International Conference on* pp. 1–96.

- Morais, H., Kádár, P., Faria, P., Vale, Z. A. y Khodr, H. M.: 2010, Optimal Scheduling of a Renewable Micro-Grid in an Isolated Load Area using Mixed-Integer Linear Programming, *Renewable Energy* **35**, 151–156.
- Nishimura, F., Tabors, R. D., Ilic, M. D. y Lacalle-Melero, J. R.: 1993, Benefit Optimization of Centralized and Decentralized Power Systems in a Multi-Utility Environment, *Power Systems, IEEE Transactions on* **8**(3), 1180–1186.
- Parida, B., Iniyar, S. y Goic, R.: 2011, A Review of Solar Photovoltaic Technologies, *Renewable and Sustainable Energy Reviews* **15**(3), 1625–1636.
- Peighambaroust, S. J., Rowshanzamir, S. y Amjadi, M.: 2010, Review of the Proton Exchange Membranes for Fuel Cell Applications, *International Journal of Hydrogen Energy* **35**(17), 9349–9384.
- Petcu, A. y Faltings, B.: 2005, DPOP: A Scalable Method for Multiagent Constraint Optimization, *International Joint Conference on Artificial Intelligence (IJCAI 05)*, pp. 266–271.
- Petcu, A. y Faltings, B.: 2007, MB-DPOP: A New Memory-Bounded Algorithm for Distributed Optimization, *International Joint Conference on Artificial Intelligence (IJCAI 07)*, pp. 1452–1457.
- Pipattanasomporn, M., Feroze, H. y Rahman, S.: 2009, Multi-agent systems in a distributed smart grid: Design and implementation, *Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES*, pp. 1–8.
- Praça, I., Ramos, C., Vale, Z. A. y Cordeiro, M.: 2003, MASCEM: A Multiagent System That Simulates Competitive Electricity Markets, *IEEE Intelligent Systems* **18**, 54–60.
- Pynadath, D. V. y Tambe, M.: 2002, Multiagent Teamwork: Analyzing the Optimality and Complexity of Key Theories and Models, *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2, AAMAS '02*, pp. 873–880.
- Spaan, M. T. J. y Vlassis, N.: 2004, A point-based POMDP algorithm for robot planning, *Proceedings - IEEE International Conference on Robotics and Automation* **2004**(3), 2399–2404.
- Su, W., Wang, J. y Roh, J.: 2014, Stochastic Energy Scheduling in Microgrids With Intermittent Renewable Energy Resources, *Smart Grid, IEEE Transactions on* **5**(4), 1876–1883.
- Varakantham, P., Yeoh, W., Velagapudi, P., Sycara, K. y Scerri, P.: 2012, Prioritized Shaping of Models for Solving DEC-POMDPs, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3, AAMAS '12, International Foundation for Autonomous Agents and Multiagent Systems*, pp. 1269–1270.
- Wang, Q., Guan, Y. y Wang, J.: 2012, A Chance-Constrained Two-Stage Stochastic Program for Unit Commitment With Uncertain Wind Power Output, *Power Systems, IEEE Transactions on* **27**(1), 206–215.

# Acrónimos

**AFB**

Cota Delantera Asíncrona. 19–21, 42, 55, 57, 60–63, 65, 69–72

**COP**

Problema de Optimización con Restricciones. 14

**CSP**

Problema de Satisfacción de restricciones. 11, 14

**DCOP**

Optimización con Restricciones Distribuido. 2, 4, 5, 11, 14, 15, 21, 22, 30, 41–47, 55, 56, 62, 72, 76

**DCSP**

Problemas Satisfacción de Restricciones Distribuidas. 11, 12

**DPOP**

Procedimiento de optimización distribuida de pseudoárboles. 16–18, 21, 42, 55–61, 63–67, 69–72

**MB-DPOP**

Procedimiento de optimización distribuida de pseudoárboles con Límite de Memoria. 18, 19, 42, 55–61, 63–67, 69–72

**SynchBB**

Ramificación y Cota Síncrona. 15, 16, 21, 42, 57–61, 63–66, 69–72

**VPP**

Productor de Energía Virtual. 10, 11