



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**EFFECTOS EN LOS NIVELES DE RESILIENCIA EN SISTEMAS DE  
DISTRIBUCIÓN A TRAVÉS DE RUTEOS DE CUADRILLAS DE  
RESTAURACIÓN**

**MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL ELÉCTRICO**

FABIÁN ENRIQUE BAZÁN LEIVA

**PROFESOR GUÍA:**  
ALEJANDRO NAVARRO ESPINOSA

**MIEMBROS DE LA COMISIÓN:**  
FERNANDO FLATOW GARRIDO  
ANDRÉS CABA RUTTE

SANTIAGO DE CHILE  
2018

**RESUMEN DE LA MEMORIA PARA OPTAR  
AL TÍTULO DE:** Ingeniero Civil Eléctrico.  
**POR:** Fabián Enrique Bazán Leiva  
**FECHA:** 22/10/2018  
**PROFESOR GUÍA:** Alejandro Navarro Espinosa

## **EFFECTOS EN LOS NIVELES DE RESILIENCIA EN SISTEMAS DE DISTRIBUCIÓN A TRAVÉS DE RUTEO DE CUADRILLAS DE RESTAURACIÓN**

En la actualidad, los sistemas de distribución no cuentan con la preparación necesaria para soportar eventos de baja probabilidad y alto impacto como desastres naturales, lo que trae como consecuencia daños en las redes evaluados en varios millones de dólares junto con la pérdida de suministro eléctrico y actividad económica del sector. Por ello, el concepto de resiliencia ha cobrado cada vez más relevancia, a lo que se suman varias propuestas para aumentar la resiliencia de un sistema. Sin embargo, antes de aplicar cualquiera de estas propuestas, la distribuidora necesita saber que elementos tiene a disposición y como puede utilizarlos adecuadamente para minimizar sus daños.

El objetivo de este trabajo radica precisamente en eso. Considerando que la distribuidora cuenta con su infraestructura dotada de una cierta resistencia y un equipo de restauración, se idea una estrategia que aprovecha la infraestructura de la red y despacha cuadrillas de restauración con tal de disminuir tiempos de reposición y energía no suministradas. Desarrollada la estrategia, se busca encontrar una relación de dichos parámetros con diferentes estados de preparación de la red.

Para ello, se desarrolla una metodología que contempla un evento sísmico real, el cual ejerce un cierto nivel de daño en una red de distribución realista modelada por nodos. Luego, se despachan las cuadrillas a cada nodo utilizando una estrategia que divide las redes en zonas cercanas a los centros de despachos de cuadrillas. Cada centro despacha sus cuadrillas hacia nodos agrupados por distancias, usando una estrategia de tres etapas: Atención a nodos con mayor potencial carga no suministrada, minimización de rutas y cooperación entre el resto de las cuadrillas.

Primero se compara la estrategia propuesta con otras estrategias de ruteo con un número fijo de cuadrillas. Se descubre la existencia de una relación inversa entre los tiempos de restauración de la red y energía no suministrada. La estrategia propuesta logra nivelar adecuadamente los dos parámetros a pesar de tener una predisposición mayor a disminuir tiempos de restauración.

Luego se mide el desempeño de la restauración de servicio utilizando la estrategia propuesta para diferentes estados de preparación de la red. Los resultados muestran que una mejora marginal de cualquiera de los parámetros requerirá cada vez una mayor cantidad de cuadrillas. Con ello se concluye que hay metas que la distribuidora simplemente no podrá cumplir en lo que respecta a la restauración de servicio frente a un evento de baja probabilidad y alto impacto.

*A mis padres y amigos.*

## **Agradecimientos**

En primer lugar, agradezco a mis padres María Elena y Pedro por su esfuerzo, por preocuparse de que nunca nos faltara nada y por siempre brindarnos apoyo incondicional, además de siempre darnos el amor que una madre y un padre deben entregar a sus hijos.

A mi hermano Wido y hermanas Antonia y Pabla por su compañía y por todos los lindos recuerdos que se han forjado desde nuestra infancia.

Quiero también dar gracias a mi profesor guía Alejandro Navarro por todas las oportunidades que me ha brindado, por compartir su experiencia y puntos de vista, que tanto han aportado al resultado final de este trabajo y los cuales hubiera sido imposible llegar por cuenta propia y por ser un excelente y comprensivo profesor guía y persona.

Finalmente quiero agradecer a las amistades que he hecho en la universidad por hacer más llevadero el paso por la carrera, así como las hechas fuera de ella en los últimos años por compartir penas y glorias.

# Tabla de Contenido

---

1.	Introducción.....	11
1.1	Motivación.....	11
1.2	Objetivos .....	12
1.2.1	Objetivo general.....	12
1.2.2	Objetivos específicos .....	12
1.3	Alcances .....	12
1.4	Estructura .....	13
2.	Eventos de baja probabilidad y alto impacto .....	14
2.1	Antecedentes de eventos BPAI en Chile .....	15
2.1.1	Antecedentes de terremotos .....	15
2.1.2	Antecedentes de nevazones .....	15
2.2	Modelación de eventos BPAI sísmicos.....	16
2.2.1	Modelamiento del daño en estructuras .....	18
2.2.2	Modelamiento de restauración de estructuras.....	20
3.	Concepto de resiliencia en sistemas de distribución.....	22
4.	Tópicos de ciencias de la computación y matemáticas discretas .....	26
4.1	Complejidad computacional .....	26
4.1.1	Crecimiento de funciones .....	26
4.1.2	Complejidad de algoritmos [35] .....	27
4.2	Teoría de grafos [35].....	28
4.3	Problemas en ciencias de la computación.....	29
4.3.1	Problema del vendedor viajero (TSP).....	29
4.3.2	Problema de Ruteo de Vehículos con Capacidad Acumulada .....	40
4.3.3	Problema del camino más corto.....	43
4.3.4	Problema de agrupamiento .....	44
4.4	Sistema de Información Georreferencial .....	46
4.4.1	Quantum GIS .....	47
5.	Modelo y estrategia propuesta .....	48
5.1	Introducción .....	48
5.2	Pre-Procesamiento de datos .....	49

5.2.1	Asignaciones .....	50
5.2.2	Demanda por transformador .....	52
5.2.3	Matriz de distancias mínimas .....	53
5.3	Modelo de Simulación y Ruteo .....	55
5.3.1	Carga de Parámetros de Entrada .....	55
5.3.2	Cálculo del Estado de daño de Transformadores .....	56
5.3.3	Clusterización por Zonas asignadas a centros de operaciones .....	59
5.3.4	Algoritmo de Ruteo Euclidiano .....	60
5.3.5	Algoritmo de Ruteo Cooperativo.....	63
5.3.6	Recalculo de tiempos y otros .....	66
5.3.7	Archivos de salida.....	66
5.4	Procesamiento de Resultados.....	66
6.	Metodología de estudio.....	69
6.1	Comparación con otras estrategias de ruteo .....	69
6.2	Desempeño para diferentes estados de preparación de la red .....	72
7.	Resultados y análisis.....	74
7.1	Estrategias de ruteo de cuadrillas.....	74
7.1.1	Estado de la red anclada después del evento BPAI .....	74
7.1.2	Estrategia: Caso base .....	75
7.1.3	Estrategia: Caso no cooperativo .....	80
7.1.4	Estrategia: Caso no guiado .....	82
7.1.5	Estrategia: Caso aleatorio zonal.....	84
7.1.6	Estrategia: Caso guiado zonal.....	85
7.1.7	Estrategia: Caso guiado total .....	87
7.1.8	Análisis de resiliencia .....	90
7.2	Estados de preparación de la red.....	91
7.2.1	Estado de la red no-anclada después del evento BPAI.....	92
7.2.2	Cuatro centros de operaciones .....	92
7.2.3	Elección del quinto centro de operaciones .....	95
7.2.4	Cinco centros de operaciones .....	97
7.2.5	Evolución de parámetros globales .....	99
7.2.6	Comparación de casos por nivel de resiliencia.....	102
8.	Conclusiones .....	104

8.1	Conclusiones Generales .....	104
8.2	Conclusiones sobre Aspectos Específicos del Trabajo de Título .....	104
8.3	Trabajos futuros .....	106
9.	Bibliografía .....	108
10.	Anexos .....	113
10.1	Anexo A – Histogramas de resultados globales .....	113
10.1.1	Tiempo de restauración del 100% .....	113
10.1.2	Tiempo de restauración del 98% .....	114
10.1.3	Energía no suministrada total .....	115
10.1.4	Costos de falla.....	116
10.1.5	Distancias recorridas totales .....	117
10.2	Anexo B – Curvas de Restauración.....	119
10.3	Anexo C – Porcentaje de transformadores dañados para los casos de preparación de la red 120	
10.4	Anexo D – Código fuente .....	121
10.4.1	Pre-Procesamiento de datos – cargaCapas_PASO1.py .....	121
10.4.2	Pre-Procesamiento de datos – getGraph_PASO2.py .....	121
10.4.3	Pre-Procesamiento de datos – guardarVAR_PASO3.py .....	123
10.4.4	Modelo de Simulación y Ruteo – Software_C3.py .....	124
10.4.5	Procesamiento de Resultados – Procesamiento.py.....	133
10.4.6	Procesamiento de Resultados – ProcesamientoZ.py .....	137
10.4.7	Procesamiento de Resultados – TrafosPros.py.....	141

# Índice de tablas

Tabla 2.1 Caracterización de los estados de daño en un circuito de distribución [26] .....	20
Tabla 2.2 Parámetros de las Curvas de Fragilidad para circuitos de distribución [26] .....	20
Tabla 2.3 Parámetros de las curvas de restauración para un circuito de distribución [26] .....	21
Tabla 3.1 Evaluación de Niveles de resiliencia en Chile y Japón .....	23
Tabla 3.2 Características de la región y sistemas en la Región Metropolitana y Región de Tohoku .....	24
Tabla 4.1 Comparación de rendimientos entre Concorde y LKH-2.....	40
Tabla 5.1 Tiempos de restauración por estado de daño .....	57
Tabla 5.2 Prioridades para diferentes tipos de fallas y consumos con un horizonte $T_{ev} = 3$ horas .....	58
Tabla 5.3 Prioridades para diferentes tipos de fallas y consumos con un horizonte $T_{ev} = 6$ horas .....	59
Tabla 6.1 Características las estrategias de ruteo .....	71
Tabla 6.2 Casos a evaluar de diferentes estados de preparación de la red de distribución .....	72
Tabla 7.1 Número de transformadores dañados por el sismo en una red anclada .....	74
Tabla 7.2 Resultados globales del Caso Base.....	77
Tabla 7.3 Resultados globales del Caso No Cooperativo .....	80
Tabla 7.4 Resultados globales del Caso No Guiado .....	83
Tabla 7.5 Resultados globales del Caso Aleatorio Zonal .....	84
Tabla 7.6 Resultados globales del Caso Guiado Zonal .....	85
Tabla 7.7 Resultados globales del Caso Guiado Total .....	87
Tabla 7.8 Nivel de resiliencia de las estrategias estudiadas .....	91
Tabla 7.9 Días sin suministro eléctrico equivalente para los casos de estudio.....	91
Tabla 7.10 Número de transformadores dañados por el sismo en una red no-anclada .....	92
Tabla 7.11 Resultados globales para los diferentes estados de preparación de la red con 4 centros .....	93
Tabla 7.12 Comparación de los tiempos de restauración total con su cota inferior .....	93
Tabla 7.13 Porcentaje de transformadores atendidos en cada zona por todas las cuadrillas de un centro .....	94
Tabla 7.14 Resultados globales para los casos con 4 y 5 centros de operaciones e igual número de cuadrillas .....	97
Tabla 7.15 Resultados globales para diferentes estados de preparación con 5 centros .....	98
Tabla 7.16 Nivel de resiliencia de todos los casos estudiados .....	103



# Índice de figuras

Figura 2.1 Representación de las fases de un evento BPAI [2] .....	14
Figura 2.2 Epicentros de sismos más cercanos a Chile [14].....	15
Figura 2.3 Energía suministrada en las subestaciones de Vitacura y Apoquindo durante las nevazones.....	16
Figura 2.4 Aceleración espectral para 4 sismos diferentes [23] .....	17
Figura 2.5 Mapa de contorno de intensidades macro sísmicas para el terremoto de Coquimbo 2015 [14].....	18
Figura 2.6 Curva de Fragilidad para diferentes estados de daño [23] .....	19
Figura 3.1 Evolución del número de hogares sin suministro eléctrico en Japón (a) y porcentaje de clientes sin servicio en Chile (b) [4] [28] .....	24
Figura 4.1 Concepto de la Gran-O (a) y crecimiento de funciones clásicas (b) [35] .....	26
Figura 4.2 Ejemplos de grafos y arboles [35] .....	28
Figura 4.3 Problema simétrico de 2392 nodos (a) y problema con 800 nodos dispuestos aleatoriamente en una grilla de 100x100 (b) .....	30
Figura 4.4 Movimiento 2-opt [39] .....	31
Figura 4.5 Movimiento 3-opt secuencial [39] .....	32
Figura 4.6 Secuencia de elección de links .....	32
Figura 4.7 Movimiento 4-opt no secuencial [39] .....	33
Figura 4.8 Ejemplo de un 1-tree [39].....	35
Figura 4.9 Ejemplo del algoritmo LKH-1 .....	37
Figura 4.10 Movimiento 4-opt secuencial infactible y movimiento 7-opt no-secuencial para volver al tour factible .....	39
Figura 4.11 Resultado del algoritmo LKH-2 para el problema World .....	40
Figura 4.12 Resolución del problema P-n76-k4 por medio del algoritmo LKH-3.....	43
Figura 4.13 Búsqueda del camino más corto entre a y z por medio de Dijkstra [35].....	44
Figura 4.14 Diagrama de Voronoi y distancias de los sitios a un punto [47] .....	45
Figura 4.15 Ejemplo del algoritmo K-Means [48] .....	46
Figura 5.1 Bloques que definen el modelo propuesto .....	48
Figura 5.2 Partes del bloque de Pre-Procesamiento .....	49
Figura 5.3 Elementos que componen la red de distribución y vial en Santiago .....	50
Figura 5.4 Tabla de atributos para coordenadas UTM19 (x, y).....	50
Figura 5.5 Asignación por área: Asignación de centros de operaciones (a) y subestaciones primarias (b).....	51

Figura 5.6 Curvas de contorno (SA-1.0s) para el terremoto del 27 de febrero del 2010 .....	52
Figura 5.7 Asignaciones finales a los transformadores en QGIS .....	52
Figura 5.8 Estructura de los archivos de distancias entre nodos. ....	54
Figura 5.9 Esquema general del modelo de simulación y ruteo de cuadrillas .....	55
Figura 5.10 Posibles estados de daños para un transformador sometido a un SA determinado ....	57
Figura 5.11 Curva de consumos de ejemplo .....	58
Figura 5.12 Clusterización de la Zona 1 .....	60
Figura 5.13 Resolución del problema CCVRP (a) y CVRP (b) para un mismo set de nodos [44]	62
Figura 5.14 Ruta de atención por una cuadrilla de la Zona 2 .....	63
Figura 5.15 Ilustración gráfica del Algoritmo Cooperativo .....	64
Figura 5.16 Ruta de una cuadrilla de la Zona 4 .....	66
Figura 5.17 Ejemplo grafico de energía no suministrada horaria (izquierda) y energía no suministrada acumulada por hora (derecha) .....	67
Figura 6.1 Asignación de transformadores para el Caso Guiado Zonal .....	71
Figura 7.1 Histogramas de resultados globales para el Caso Base .....	76
Figura 7.2 Curva del porcentaje de carga suministrada para el Caso Base .....	77
Figura 7.3 Curva de energía no suministrada horaria para el Caso Base .....	77
Figura 7.4 Horas de trabajo de cada cuadrilla para el Caso Base .....	78
Figura 7.5 Curva de restauración del Caso Base .....	79
Figura 7.6 Histograma de transformadores reparados por estado de daño para el Caso Base .....	79
Figura 7.7 Curva del porcentaje de carga suministrada para el Caso Base y Caso No Cooperativo .....	81
Figura 7.8 Curva de energía no suministrada horaria para el Caso Base y Caso No Cooperativo	81
Figura 7.9 Horas de trabajo de cada cuadrilla para el Caso Base y Caso No Cooperativo .....	82
Figura 7.10 Curva de restauración del Caso No Cooperativo .....	82
Figura 7.11 Curva del porcentaje de carga suministrada para el Caso Base y Caso No Guiado ..	83
Figura 7.12 Curva del porcentaje de carga suministrada para el Caso Base y Caso Aleatorio Zonal .....	85
Figura 7.13 Curva del porcentaje de carga suministrada para el Caso Base y Caso Guiado Zonal .....	86
Figura 7.14 Curva de restauración del Caso Guiado Zonal.....	87
Figura 7.15 Curva del porcentaje de carga suministrada para el Caso Base y Caso Guiado Total	88
Figura 7.16 Curva de restauración del Caso Guiado Total.....	88
Figura 7.17 Evolución de la energía no suministrada acumulada para diferentes estrategias .....	89

Figura 7.18 Curva del porcentaje de carga suministrada para los casos con 4 centros de operaciones .....	93
Figura 7.19 Energía no suministrada de transformadores atendidos (a) y distancias recorridas (b) por las cuadrillas de una determinada zona .....	95
Figura 7.20 Zonas de la red con 4 centros (izquierda) y primera localización del quinto centro (derecha) .....	95
Figura 7.21 Cambio de localización del quinto centro de operaciones .....	96
Figura 7.22 Posición final del quinto centro de operaciones .....	97
Figura 7.23 Curva del porcentaje de carga suministrada para los casos CS_100 y CS_100Z5 ....	98
Figura 7.24 Curva del porcentaje de carga suministrada para los casos con cinco centros de operaciones .....	99
Figura 7.25 Curva de tiempo de reposición total vs cuadrillas en la red .....	100
Figura 7.26 Curvas de tiempo de reposición parcial vs cuadrillas en la red .....	100
Figura 7.27 Curva de energía no suministrada vs cuadrillas en la red .....	101
Figura 7.28 Curva de costos de falla vs cuadrillas en la red .....	102

# 1. Introducción

---

## 1.1 Motivación

En los últimos años los sistemas eléctricos han sufrido severos daños frente a desastres naturales. En los Estados Unidos el azote de los huracanes ha dejado entre 8 millones de usuarios sin suministro eléctrico por varias semanas [1], mientras que otros países con niveles sísmicos similares a los de Chile, como lo son Japón y Nueva Zelanda, han experimentado un problema similar, donde los mayores daños se observaron en las áreas de distribución del sector eléctrico [2]. En el contexto nacional, desastres como nevazones y terremotos han producido prolongados cortes de suministro eléctrico, siendo nuevamente el área de distribución la más afectada [3] [4].

Lo anterior se puede explicar debido a la pasividad de la planificación de la distribución con respecto a este tema, es decir, el cómo ante un desastre natural, la distribuidora procede a reconstruir y reinstalar sus equipos de la misma forma en la que estaba antes del evento, impidiendo un aprendizaje del mismo y dejándolo en las mismas condiciones para el siguiente desastre de similares características, esto probablemente debido a la falta de incentivos regulatorios para hacerlo.

Bajo esta premisa, el concepto de *resiliencia* en redes eléctricas y otros servicios ha tomado cada vez más fuerza tanto a nivel internacional como nacional [5]. En [2] el concepto de resiliencia para sistemas de distribución se define como “*La habilidad de prepararse y adaptarse contra condiciones cambiantes y la de resistir y recuperarse rápidamente ante eventos disruptivos*”. Si bien la definición describe adecuadamente un sistema resiliente en áreas de distribución, hasta el día de hoy no ha sido posible consensuar una métrica adecuada ni una definición universal para ella.

Diferentes propuestas han surgido a lo que sería el aumento de niveles de resiliencia en el sistema eléctrico, cada uno con diferentes interpretaciones de la misma y que varían según el área del sistema eléctrico que se esté tratando. En lo que respecta a las áreas de distribución de electricidad, entre las soluciones propuestas destacan la implementación de micro-redes de emergencia [6], reforzamiento de la infraestructura de distribución [7], dar lineamientos para un número óptimo de cuadrillas [8] [9] y entregar guías a las distribuidoras para disminuir daños por causas externas, como la tala de árboles en las servidumbres de los postes [10].

A pesar de los diferentes enfoques que han tomado la industria y la academia, ambos pueden coincidir en que el nivel de resiliencia de un sistema depende fuertemente de la profundidad del daño<sup>1</sup> causado por un evento y la rapidez con el que logra recuperarse total o parcialmente. Por ello, es indispensable la buena utilización de los recursos disponibles por parte de las compañías distribuidoras. En primera instancia, una compañía poseerá su

---

<sup>1</sup> La profundidad de daño se refiere a la caída porcentual de un parámetro de interés inmediatamente después de un evento catastrófico. El parámetro depende del sistema estudiado, siendo utilizado el número de clientes o energía suministrada en sistemas de distribución.

infraestructura con determinada resistencia<sup>2</sup> y un equipo de restauración al cual se le deberá asignar un itinerario de reparaciones adecuado para evitar deficiencias en el proceso de restauración de la red. Experiencias anteriores demuestran que una adecuada coordinación y manejo de cuadrillas puede reducir los tiempos de restauración considerablemente [11], siendo este tópico el tema principal abordado en el presente trabajo de título.

## **1.2 Objetivos**

### **1.2.1 Objetivo general**

El objetivo principal de este trabajo consiste en modelar y determinar el ruteo de cuadrillas de emergencia durante un evento de baja probabilidad y alto impacto con el fin de aumentar la resiliencia del sistema.

### **1.2.2 Objetivos específicos**

Los objetivos específicos del trabajo de título son:

- Analizar la literatura concerniente a resiliencia en sistemas eléctricos de potencia, con especial atención a los sistemas de distribución, y desprender de ella métricas que permitan comparar los diferentes casos de estudio.
- Estudiar y utilizar una modelación de sismos aplicable al caso de estudio.
- Proponer y desarrollar una estrategia de ruteo de cuadrillas, rápida y eficaz, que sea aplicable a los casos de estudio para el aumento de la resiliencia del sistema.
- Recopilar información sobre programas computacionales para la implementación de la metodología propuesta.
- Comparar la estrategia de ruteo propuesta con otras estrategias de ruteo.
- Evaluar el desempeño de diferentes estados de preparación de la red utilizando la estrategia propuesta.

## **1.3 Alcances**

Tanto los conceptos de resiliencia como los tópicos que conciernen a la metodología empleada abarcan un amplio espectro de temas de la ingeniería y matemática, los cuales no pueden ser explorados a plenitud. Por ello, se definen los siguientes alcances:

- El análisis de resiliencia solo considera un único evento sísmico.
- Se utilizará solo un algoritmo de ruteo óptimo. Las estrategias a comparar harán uso o no del algoritmo dependiendo de cada caso.
- La red a estudiar solo contempla transformadores de distribución distribuidos en una zona de concesión realista de gran tamaño, las cuales modelan sus redes aguas abajo asociadas.

---

<sup>2</sup> La resistencia de un sistema es su capacidad para soportar un evento catastrófico y tiene un efecto directo sobre la profundidad de daño en el mismo.

- El levantamiento de carga a considerar es de tipo switch, es decir, no considera la componente dinámica de los sistemas de distribución ni considera la participación de la Transmisión ni Generación en el levantamiento de carga. Esto es, se asume que la carga se levanta tan pronto el sistema de distribución es recuperado.

## 1.4 Estructura

La presente memoria se compone de 8 capítulos. El segundo capítulo trata el concepto de evento de baja probabilidad y alto impacto y entrega antecedentes de tales eventos en Chile y la red de distribución de Santiago. También se presentan metodologías para modelar eventos de tipo sísmico y sus efectos en estructuras, particularmente en equipos de distribución.

El capítulo tres expone los conceptos de resiliencia y la importancia que ha cobrado en los últimos años debido al aumento de eventos de alto impacto y baja probabilidad en las redes, así como los costos asociados al mismo [12]. Luego se hace un resumen de diferentes trabajos realizados en la academia para definir el concepto de resiliencia en los sistemas eléctricos y se opta por una definición adecuada al trabajo de título.

En el capítulo cuatro se hace un repaso de definiciones relacionadas a matemáticas discretas y ciencias de la computación para comprender los problemas presentes en la implementación de la estrategia de ruteo de cuadrillas, así como los algoritmos utilizados para su resolución.

En el capítulo cinco se describe en detalle el modelo implementado para la simulación de la red ante un evento sísmico, así como la simulación del comportamiento de las cuadrillas de acuerdo a la estrategia implementada en el modelo. También se describe la estrategia propuesta para comandar las cuadrillas durante la etapa de restauración.

En el capítulo seis se describen los casos de estudio, los cuales se dividen en 2 conjuntos. El primer conjunto de casos tiene como finalidad comparar el desempeño de la estrategia propuesta con otras estrategias de ruteo que la distribuidora podría optar a utilizar. El segundo conjunto de casos tiene como finalidad conocer el comportamiento de los parámetros de desempeño de la restauración de servicio ante diferentes estados de preparación de la red, utilizando la estrategia propuesta.

El capítulo siete muestra los resultados y análisis de cada caso y finalmente en el capítulo ocho se exponen las conclusiones y discusiones finales tanto de los resultados como del trabajo de título en general.

## 2. Eventos de baja probabilidad y alto impacto

Los eventos de baja probabilidad y alto impacto (también llamados eventos BPAI) corresponden a eventos que alteran en gran medida el funcionamiento de un sistema y que su predicción es casi imposible de realizar. Generalmente a estos eventos se les asocian desastres naturales, pero también pueden abarcar ataques producidos por interferencia humana accidental o intencional. Si bien cada evento puede ser muy diferente uno de otro, es posible clasificar sus estados en 4 fases:

1. **Extreme event duration:** Corresponde al evento de alto impacto propiamente tal y tiene una duración de unos cuantos minutos hasta unos cuantos días (Ejemplos son desastres naturales como terremotos, aluviones y tormentas). Se inicia a partir de los primeros signos del evento BPAI y finaliza con la primera reparación. En esta fase, el enfoque está destinado principalmente a la reducción de la profundidad de daño sufrido por el sistema.
2. **Immediate aftermath:** Fase que dura desde unos días hasta semanas. La fase comienza apenas termina la fase 1 y finaliza con el sistema casi completamente reparado, tomándose como referencia un 98% de la carga, clientes o equipos suministrados. En esta fase se tiene un enfoque en la reposición de suministro.
3. **Intermediate aftermath:** Fase que dura desde unas semanas hasta varios meses. En ella se focalizan las adquisiciones de datos del evento, la respuesta del sistema, las curvas de levantamiento de carga, etc. En resumen, corresponde a un levantamiento de datos tanto de los daños como del proceso de restauración.
4. **Long-term aftermath:** Fase que dura desde meses hasta años. En ella se realizan todas las actividades de planificación en base a los datos obtenidos de la fase 3 con la finalidad de que el sistema aprenda de ellos y logre enfrentar el próximo evento BPAI.

Las fases descritas se encuentran dentro de un ciclo que inicia con cada evento BPAI sufrido por el sistema. La Figura 2.1 ilustra gráficamente cada fase, donde la curva azul representa el porcentaje de clientes suministrado y la curva roja el porcentaje de infraestructura dañada.

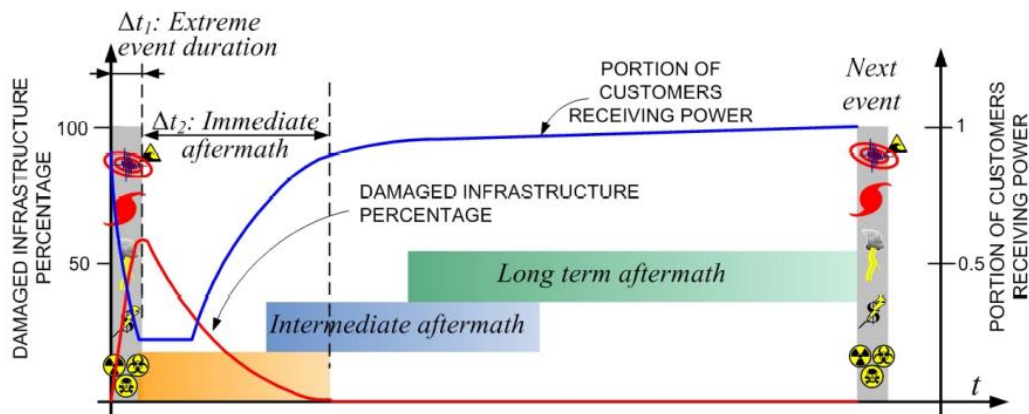


Figura 2.1 Representación de las fases de un evento BPAI [2]

## 2.1 Antecedentes de eventos BPAI en Chile

### 2.1.1 Antecedentes de terremotos

Chile, junto a Japón, es conocido como uno de los países más sísmicos del planeta. En efecto, en la lista de los veinte terremotos de mayor intensidad registrados en la historia [13], Chile no solo posiciona tres de ellos, sino que el Terremoto de Valdivia del 22 de mayo de 1960 se sitúa en el primer lugar de dicha lista con una magnitud de 9.5 grados Richter. La Figura 2.2 muestra la localización de los epicentros de los terremotos situados con intensidad superior a 8.0 (izquierda) y 7.0 (derecha) grados Richter [14].

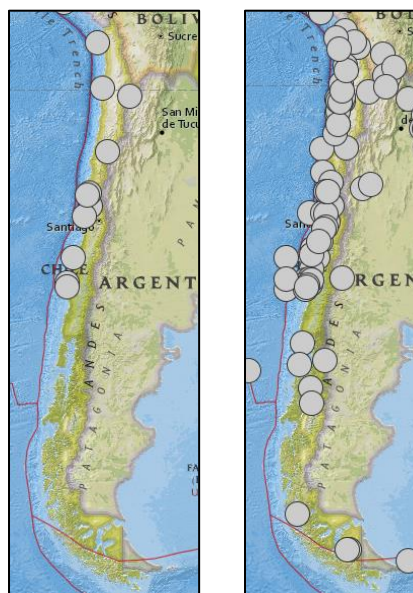


Figura 2.2 Epicentros de sismos más cercanos a Chile [14]

El territorio chileno cuenta con 11 terremotos superior a los 8.0 grados Richter y cerca de 60 terremotos con una intensidad superior a los 7.0 grados Richter, además de que gran parte de ellos se sitúan en las cercanías de la capital. A pesar de que la naturaleza de estos eventos no hace posible hacer una afirmación que relacione su localización con su intensidad, el hecho mencionado resulta de gran curiosidad, considerando el valor de la infraestructura instalada en la capital.

Lo anterior motiva a realizar el trabajo de memoria enfocándose en los impactos sísmicos que pueda sufrir la infraestructura de distribución en la Región Metropolitana, considerando que sismos superiores a 7.0 grados Richter podrían dañar gravemente los sistemas de distribución en la zona junto con la continuidad de suministro.

### 2.1.2 Antecedentes de nevazones

Entre los meses de junio y julio del año 2017 se registraron un máximo de precipitaciones de 77 milímetros de agua, superior a todo lo visto en los últimos 8 años, donde estas no alcanzaban los 45 milímetros de lluvia [15], junto con nevazones en sectores de la capital donde los últimos registros de esta clase de fenómenos datan hace más de 40 años [16], dejando un máximo de entre 430 mil y 311 mil clientes sin suministro eléctrico [3]. Debido a la magnitud del evento, las reparaciones tardaron semanas.



Otro aspecto a destacar es el hecho de que en el Terremoto del 27 de febrero del 2010 se registraron un total de 1500 fallas en Santiago, mientras que las nevazones de julio produjeron un total de 2200 fallas aproximadamente [17], siendo gran parte de ellas provocadas por la caída de árboles en el tendido eléctrico. Cabe preguntarse qué evento clasifica como falla, considerando que la distribuidora presentó daños en más de 18 mil equipos de distribución junto con 144 kilómetros de conductores [18] para el terremoto del 2010.

Una forma de caracterizar o visualizar los efectos de esta clase de fenómenos consiste en observar la evolución de la energía suministrada en las subestaciones afectadas. La Figura 2.3 muestra como la energía suministrada de las subestaciones primarias Vitacura y Apoquindo cae abruptamente en la madrugada del 15 de julio del 2017 y tardan cerca de 2 días en recuperar completamente el suministro eléctrico.

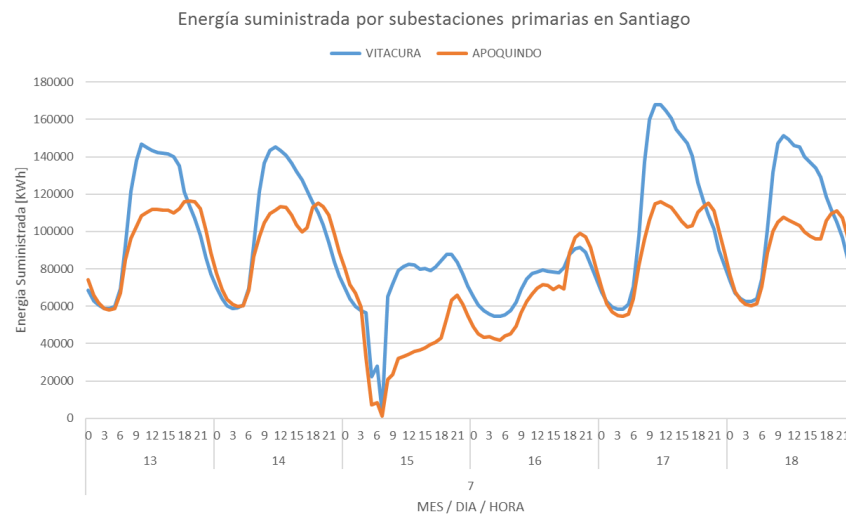


Figura 2.3 Energía suministrada en las subestaciones de Vitacura y Apoquindo durante las nevazones

A pesar de que no se cuentan con suficientes datos para modelar un evento de estas características, es de especial interés conocer la forma en como diferentes eventos BPAI pueden afectar a una misma red de distribución.

## 2.2 Modelación de eventos BPAI sísmicos

Normalmente un sismo se describe por la intensidad del mismo en términos de energía liberada, razón por la que se usa la escala Richter tanto a nivel cotidiano como en la academia. Sin embargo, en el ámbito de la ingeniería resulta más ventajoso utilizar una métrica que describa el movimiento del terreno u equipo provocado por el sismo, el cual puede ser descrito por su velocidad o por su aceleración, cada uno con diferente finalidad. Para el caso de querer conocer el comportamiento de estructuras simples y complejas frente a un sismo, se emplean las métricas de aceleración.

Las métricas de aceleración son las denominadas *Peak Ground Acceleration* (PGA) y el *Spectral Acceleration* (SA) de un periodo y amortiguación determinada y ambos medidos en Gal (g). El PGA corresponde a la máxima aceleración del terreno que siente un punto en el espacio de masa conocida en un sismo, medidas en 3 dimensiones, aunque normalmente se utiliza la máxima componente horizontal medida [19]. Por otro lado, el SA cumple la misma

función que el PGA solo que caracteriza a estructuras más complejas de frecuencia natural elevada. Como estas estructuras oscilan por un periodo finito de tiempo, también se considera una tasa de amortiguamiento que oscila entre el 1% y el 5% [20].

Tanto el PGA como el SA se determinan para un área afectada por un sismo por medio de las GMPE (Ground Motion Prediction Equation). La ecuación 2.1 muestra la forma general de un GMPE [21]:

$$\ln(PGA) = C_1 + C_2 \times M + C_3 \ln(R + C_4 \times e^{C_5 \times M}) + C_6 \times h \quad 2.1$$

- $C_i$ : Parámetros obtenidos de forma empírica.
- $M$ : Intensidad del sismo en escala Richter.
- $R$ : Distancia euclidiana desde el epicentro hacia un punto en específico en kilómetros.
- $h$ : Profundidad del epicentro en kilómetros.

Los parámetros  $C_i$  se obtienen por medio de regresiones con datos de mediciones de PGA y/o SA puntuales provenientes de estaciones sismográficas [22]. La Figura 2.4 ilustra la relación entre el PGA y el SA para diferentes sismos, donde el PGA corresponde al caso particular del SA con periodo de cero segundos.

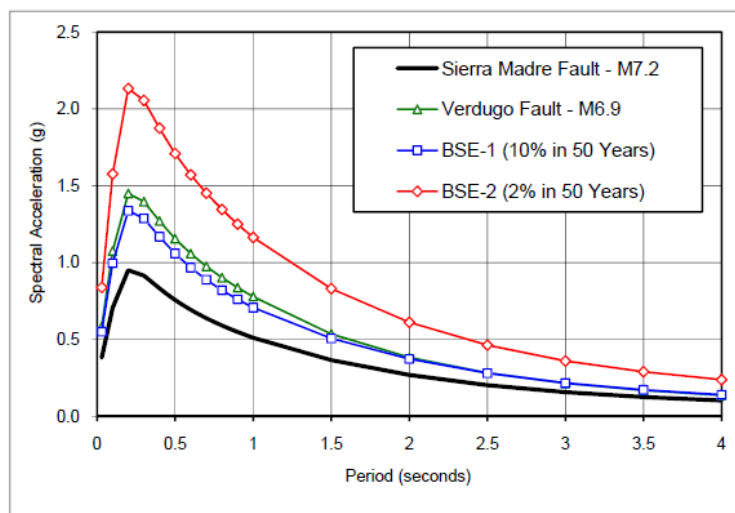


Figura 2.4 Aceleración espectral para 4 sismos diferentes [23]

La metodología anterior produce un cambio radial y atenuado de la aceleración espectral a medida que el punto de medición  $R$  se aleja de la zona de ruptura del sismo sin considerar los cambios en el terreno ni el tipo de suelo afectado. Por ello, a lo largo de los años se ha desarrollado un sistema de rápida caracterización de la distribución del movimiento del terreno para terremotos de gran envergadura denominado *ShakeMap*, los cuales entregan como resultado mapas de contorno del movimiento del terreno, entre otros, del área afectada. A modo de ejemplo, la Figura 2.5 muestra la distribución de intensidades macro sísmicas a lo largo del territorio chileno afectado en el terremoto del 16 de septiembre del 2015 en Coquimbo, de la cual se aprecia como el relieve y tipo de suelo va afectándolas a medida que ingresan al territorio chileno.

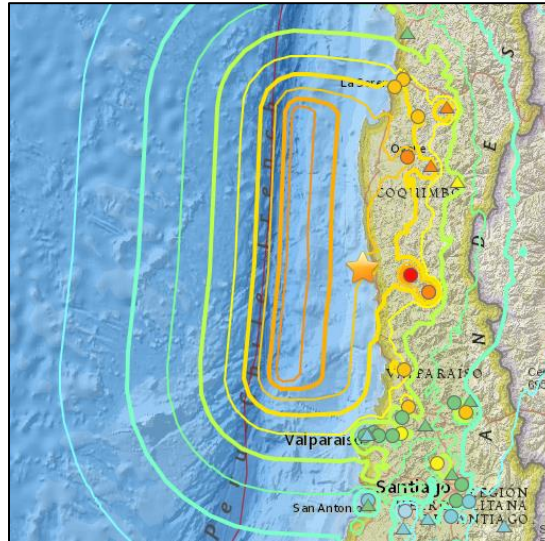


Figura 2.5 Mapa de contorno de intensidades macro sísmicas para el terremoto de Coquimbo 2015 [14]

A grandes rasgos, la obtención de estos mapas se hace de la siguiente forma [24] [25]:

1. Preparación de datos: Se remueven las estaciones sismográficas que presentan datos no confiables, se convierten datos de intensidad a movimiento de terreno y viceversa en caso de que se cuente con solo una de ellas y se ajusta el terreno a uno uniforme de tipo “roca”.
2. Elección de un GMPE: Se elige un modelo de GMPE de acuerdo a la localización del terremoto y factores de terreno para cada estación sismográfica.
3. Se realiza una interpolación de las intensidades de movimiento en el resto del terreno
4. Se aplica una corrección de terreno denominado *Amplificación de Movimiento del Terreno* que considera las irregularidades del mismo, así como el tipo de suelo.

### 2.2.1 Modelamiento del daño en estructuras

Como se mencionó en la sección anterior, la utilización de la aceleración espectral se da en el ámbito de la ingeniería ya que permite determinar el grado de resistencia sísmica y daño provocado en una estructura o equipo con cierto grado de probabilidad. Para ello, se utilizan las denominadas *Curvas de Fragilidad* que corresponden a un conjunto de curvas log-normal que describen la probabilidad de daño en alguna estructura o equipo. El tipo de daño puede ser estructural, no-estructural con componentes que puedan torcerse o descolocarse y no-estructural con elementos sensibles a la aceleración. De acuerdo a algún parámetro de entrada de fuerza o respuesta de la estructura, en nuestro caso la aceleración espectral, las curvas modelan los daños de tipo Menor, Moderado, Extensivo y Completo [23].

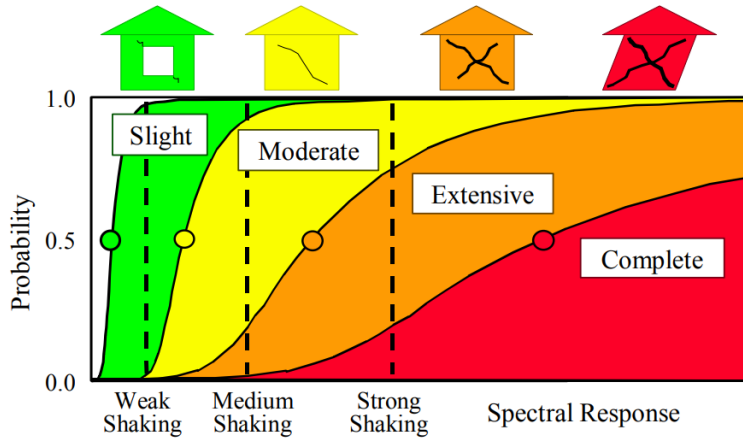


Figura 2.6 Curva de Fragilidad para diferentes estados de daño [23]

La definición formal de cada curva corresponde a la función de probabilidad acumulada condicionada a uno de los tipos de daños mencionados en el párrafo anterior, llamado  $ds$  (Damage state), ante un espectro de desplazamiento  $S_d$ . Esto es

$$P[ds|S_d] = \Phi \left[ \frac{1}{\beta_{ds}} \cdot \ln \left( \frac{S_d}{S_{d,ds}} \right) \right] \quad 2.2$$

Con  $S_{d,ds}$  el valor del espectro de desplazamiento medio con el que se llega al estado  $ds$ ,  $\beta_{ds}$  es la desviación estándar del logaritmo natural del espectro de desplazamiento para el estado de daño  $ds$  y  $\Phi(\cdot)$  es la función de probabilidad acumulada de la distribución normal estándar.

De la Figura 2.6 se puede desprender que, dado un valor de respuesta espectral, las curvas de fragilidad de una estructura dividen el intervalo  $[0,1]$ , lo cual resulta útil para modelar el estado de daño final de una estructura. Por ejemplo, para un sismo fuerte (Strong Shaking) una estructura podría tener un estado de daño Moderado, Extensivo o Completo. De hecho, teniendo en cuenta el significado de las curvas de probabilidad acumulada, en realidad las curvas indica que el daño podría ser mayor o igual al Moderado, Extensivo o Completo. Como la curva Moderada contiene el daño Extensivo y completo, entonces existe una probabilidad de tener únicamente un daño Moderado igual al valor de la Curva Moderada menos el de la Curva Extensiva. Lo mismo aplica para todas las curvas. En base a lo anterior y dado un  $x \in [0,1]$ , una estructura tendrá un determinado estado de daño si:

$0 \leq x \leq P[ds_5 S_d]$	Equipo en estado de daño $ds_5$ : Completo
$P[ds_5 S_d] \leq x \leq P[ds_4 S_d]$	Equipo en estado de daño $ds_4$ : Extensivo
$P[ds_4 S_d] \leq x \leq P[ds_3 S_d]$	Equipo en estado de daño $ds_3$ : Medio
$P[ds_3 S_d] \leq x \leq P[ds_2 S_d]$	Equipo en estado de daño $ds_2$ : Menor
$P[ds_2 S_d] \leq x \leq 1$	Equipo en estado de daño $ds_1$ : Nulo

En el caso particular y de interés para este trabajo, para los equipos de distribución las variables relevantes son la aceleración sísmica, donde este vendría siendo el espectro de desplazamiento  $S_d$  y su clasificación, las cuales pueden ser el tipo de voltaje de diseño o el

diseño sísmico del equipo en cuestión. Con respecto al diseño sísmico, también denominado estructura anclada, ello hace referencia a toda componente y/o equipo diseñado para satisfacer criterios sísmicos ya sea por reforzamiento, flexibilidad o anclaje. Un equipo con componentes estándar, también llamado no-anclado, es todo equipo diseñado sin consideraciones especiales aparte de las de manufacturación [26].

Tabla 2.1 Caracterización de los estados de daño en un circuito de distribución [26]

Estado de daño	Circuitos de distribución
$ds_1$	Ningún tipo de daño
$ds_2$	Fallan 4% de todo el circuito
$ds_3$	Falla 12% de todo el circuito
$ds_4$	Falla 50% de todo el circuito
$ds_5$	Falla 80% de todo el circuito

La Tabla 2.1 caracteriza los estados de daño para un *circuito de distribución* según la *Federal Emergency Management Agency* del gobierno de los Estados Unidos. Un sistema de distribución está compuesto por un conjunto de circuitos, los que a su vez están compuestos de postes, líneas y equipamientos en líneas propiedad de la distribuidora o privados. La Tabla 2.2 incluye los parámetros de las curvas de fragilidad para un circuito de distribución anclado y no-anclado.

Tabla 2.2 Parámetros de las Curvas de Fragilidad para circuitos de distribución [26]

Clasificación	Estado de daño	$\bar{S}_{dds} (g)$	$\beta_{ds}$
<b>CIRCUITO DE DISTRIBUCION (Anclado)</b>	Menor	0.28	0.30
	Moderado	0.40	0.20
	Extensivo	0.72	0.15
	Completo	1.10	0.15
<b>CIRCUITO DE DISTRIBUCION (No-Anclado)</b>	Menor	0.24	0.25
	Moderado	0.33	0.20
	Extensivo	0.58	0.15
	Completo	0.89	0.15

### 2.2.2 Modelamiento de restauración de estructuras

Otra característica que definen los estados de daño de los equipos es el tiempo esperado de restauración del mismo. Considerando la naturaleza aleatoria del estado de daño, es de esperar que el tiempo de restauración presente similares características.

Se definen las curvas de restauración de estructuras en un estado  $ds_i$  como una función de probabilidad con distribución normal que entregan la probabilidad de que un equipo en cierto estado tarde  $d$  días en ser reparado, dependiendo también del equipo y su clasificación. En la Tabla 2.3 se tabulan los parámetros que definen las curvas de restauración para los circuitos de distribución.

*Tabla 2.3 Parámetros de las curvas de restauración para un circuito de distribución [26]*

<b>Clasificación</b>	<b>estado de daño</b>	<b><math>\mu</math> (días)</b>	<b><math>\sigma^2</math></b>
<b>CIRCUITO DE DISTRIBUCION</b>	Menor	0.3	0.2
	Moderado	1.0	0.5
	Extensivo	3.0	1.5
	Completo	7.0	3.0

Cabe destacar que estas curvas no solo consideran la reparación misma del equipo sino también de los tiempos ajenos a la reparación, como son la detección de daños y los tiempos de viaje entre otros, las cuales se obtienen de una recopilación de datos y no obedecen ningún modelo matemático. Una de los desafíos de esta memoria es encontrar estas curvas de reparación para el caso de la red de transformadores de media tensión ante un evento sísmico determinado.

### 3. Concepto de resiliencia en sistemas de distribución

---

El concepto de resiliencia se ha estudiado ampliamente en los últimos años en varias disciplinas de la ingeniería y ciencia debido principalmente a que los análisis de confiabilidad tradicionales no logran dar señales a los sistemas para prepararlos contra eventos BPAI [2]. En [5] se define un concepto de resiliencia general como “*Las capacidades de un sistema, persona, comunidad o país expuestos a una amenaza de origen natural, para anticiparse, resistir, absorber, adaptarse y recuperarse de sus efectos de manera oportuna y eficaz, para lograr la preservación, restauración y mejoramiento de sus estructuras, funciones básicas e identidad*”.

En lo que se refiere a sistemas eléctricos de potencia, aún no ha sido posible definir una métrica universal que cuantifique esta característica en ninguna de sus partes, es decir, generación, transmisión y distribución. Más aun, recientes investigaciones han concluido que los antiguos estudios de confiabilidad en la planificación de sistemas eléctricos y sus métricas asociadas, como el *Lose of Load Probability* (LOLP) no logran dar las señales necesarias para poder conducir el sistema hacia uno más resiliente [27].

Diversas definiciones de resiliencia se han propuesto en la academia en un esfuerzo de encontrar un punto en común con todas las definiciones dadas: En [27] se propone el concepto de *Capacity Accessibility* que relaciona la probabilidad de que un nodo de la red, ante un evento catastrófico, logre situarse en una isla con capacidad de generación necesaria para poder suplir su demanda. Otro concepto de enfoque similar al anterior relaciona la capacidad de una red en dividirse en varios sub-sistemas denominados *Smart Grids* luego de un evento BPAI con la finalidad de que la caída de suministro sea mínima [6]. Por último, en [2] se logra definir un concepto de resiliencia para sistemas de distribución, el cual relaciona la profundidad de la caída de carga en el sistema con los tiempos de recuperación. A pesar de las discrepancias, todos los autores coinciden en que existe una relación directa entre el nivel de resiliencia de un sistema eléctrico y la capacidad de este de soportar el evento BPAI y la capacidad de recuperación del mismo.

Otro punto que debe ser tomado en cuenta es el daño que sufren las diferentes partes de un sistema eléctrico frente a eventos BPAI. Mediante la recolección de datos para diferentes tipos de eventos de alto impacto, ya sean terremotos, aluviones y huracanes, se han llegado a las siguientes conclusiones:

- **Transmisión:** Daños mínimos o nulo. Reposiciones tardan horas.
- **Generación:** Daños mínimos a medianos. Dependiendo del tipo de evento, puede tardar semanas en recuperar ciertas centrales y/o unidades generadoras. En lo que respecta al parque generador, los daños no comprometen en gran medida al sistema eléctrico debido a que tradicionalmente la capacidad instalada excede la demanda máxima del sistema y así en caso de que una central falle existen otras en el sistema que pueden inyectar la energía requerida.
- **Distribución:** Daños generalizados en todo el sistema de distribución para todo tipo de eventos. Dependiendo del evento, la reposición de suministro puede tardar de semanas hasta meses. Cabe destacar que varios estudios indican que un daño en un 5% de los equipos en la red de distribución pueden provocar una pérdida de hasta un 95% de la carga suministrada.

Por los puntos anteriores resulta de mayor interés buscar medidas que logren aumentar los niveles de resiliencia en áreas de distribución del sistema eléctrico, razón por la que se ahondará en el concepto de resiliencia definido en [2].

En su forma más simple, el nivel de resiliencia de un sistema de distribución se define matemáticamente como

$$R_I = \frac{T_U}{T} = \frac{T_U}{T_U + T_D} \quad 3.1$$

Donde  $T_U$  es al tiempo de subida,  $T_D$  al tiempo de bajada y  $T$  el horizonte de evaluación. Más concretamente,  $T_U$  (Time Up) se define como el tiempo que una carga está siendo suministrada durante el horizonte de evaluación y se le relaciona con la capacidad del sistema para soportar el evento BPAI. Por otro lado,  $T_D$  (Time Down) corresponde al tiempo en que la carga se encuentra sin suministro y se le asocia a los tiempos de recuperación de servicio. Otro punto a destacar es que a diferencia del análisis de confiabilidad en sistemas eléctricos, el horizonte  $T$  es flexible, es decir, puede ser definido en un intervalo que abarque cualquiera de las fases de un evento de baja probabilidad y alto impacto. Notar también que el concepto de *carga* puede referirse a un único cliente como a la de una red de distribución completa. En este último caso, se asocian los tiempos  $T_D$  y  $T_U$  a porcentajes de carga no suministrada y suministrada respectivamente en intervalos de tiempo, los cuales suelen ser medidos en horas o días.

Otros conceptos de importancia son el porcentaje de clientes no suministrados  $\theta(t)$  y la resiliencia dependiente de una carga  $R_L$ , que relaciona la resiliencia base  $R_I$  con su capacidad de almacenamiento de energía con un tiempo de operación  $T_S$ , donde estas actúan como *buffers* de propagación de carga no suministrada en la red eléctrica.

A modo de ejemplo, en [2] se aplican dichos conceptos para varios eventos ocurridos en diferentes países del mundo. En la Tabla 3.1 se muestran los resultados de resiliencia para los Terremotos del 27 de febrero del 2010 en Chile y el del 11 de marzo del 2011 en Japón, de intensidad 8.8 y 9.0 grados Richter respectivamente.

Tabla 3.1 Evaluación de Niveles de resiliencia en Chile y Japón

Event	Area	Area Type	$\theta_{\max}$	$T_U$ (Days)	$T_D$ (Days)	$R_B$	$R_L$ ( $T_S = 1$ Day)
Chile Earthquake (EQ)	Region VIII	Total	1.00	11.42	5.58	0.67	0.73
Japan EQ	Tohoku affected area	Total	1.00	37.25	2.71	0.93	0.95

A pesar de que ambos eventos produjeron un corte total del suministro eléctrico en el área de evaluación, el tiempo de recuperación en el área de Tohoku es menor a la de la Región del Biobío. Otro aspecto a considerar es el mayor horizonte de evaluación en el área de Tohoku debido a una réplica de alta intensidad cerca de un mes después del evento en cuestión que dejó sin suministro a gran parte de sus clientes. La Figura 3.1 muestra la evolución de hogares sin suministro y porcentaje de clientes no suministrados respectivamente.



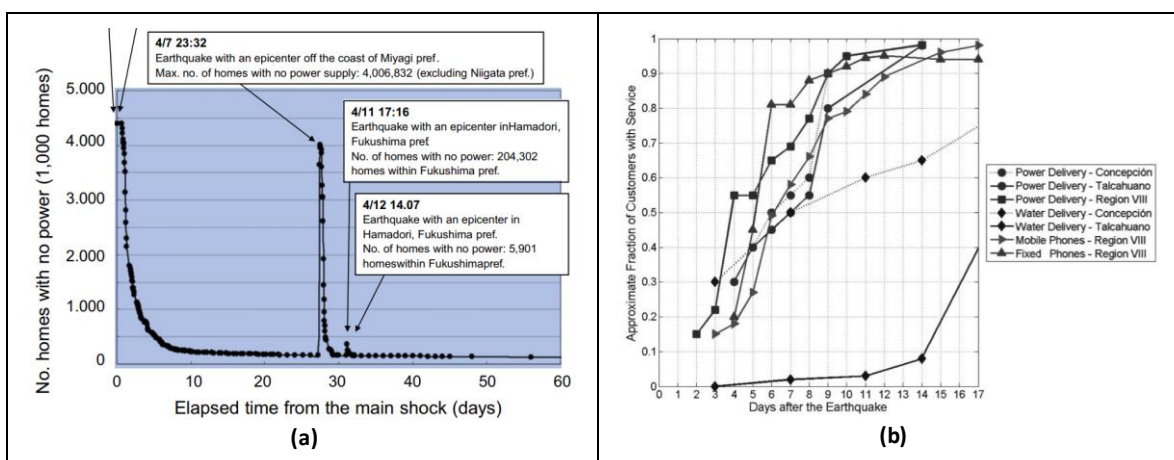


Figura 3.1 Evolución del número de hogares sin suministro eléctrico en Japón (a) y porcentaje de clientes sin servicio en Chile (b) [4] [28]

Para el caso del área de Tohoku no solo el sistema logro recuperarse en poco tiempo pasado el terremoto propiamente tal, sino que también logró evolucionar de tal forma que, frente al segundo evento BPAI, la pérdida de hogares suministrados y tiempos de reposición fueron menores al primero. En contraste con la región del Biobío donde se observa como no se tuvo información de la carga no suministrada durante dos días junto con una reposición de suministro lenta comparada a la del caso anterior.

Para poder comprender la razón del comportamiento diferente en la restauración de ambos sistemas para eventos BPAI de similares características, es necesario conocer ciertos aspectos tanto del sistema como de la demografía y geografía del lugar. En la Tabla 3.2 se ponen a disposición ciertos datos concernientes a lo mencionado anteriormente [29] [30] [31] [32] [33] [34]

Tabla 3.2 Características de la región y sistemas en la Región Metropolitana y Región de Tohoku

	Región Metropolitana	Región de Tōhoku
Área [ $km^2$ ]	15.400	79.500
Población [millones]	6,310	9,021
Máxima demanda [GW]	3,0	13,3
Costo Fijo en tarifa R. [CLP]	683	5.625
Costo Variable en tarifa R. [CLP/kWh]	111,66	130,52
Valor de Instalaciones [millones de USD]	1.385	13.375

En este caso se compara la región de Tohoku con la Región Metropolitana, que tuvo una reposición de suministro similar al a región del Biobío, solo que con una duración de 12 días. Se observa en primer lugar que el área abarcada por la distribuidora de Tohoku, *Tohoku*

*Electric Power Company*, es hasta 5 veces mayor a la Región Metropolitana. Por otro lado, el valor de las instalaciones es hasta 10 veces mayor en Tohoku, pero a su vez el número de clientes es casi 2 veces mayor al de la RM (mayor densidad de carga), por lo que es posible distribuir de mejor manera el cobro por inversión en instalaciones hacia los clientes regulados. En efecto, se observa que el costo fijo de la tarifa residencial es casi 10 veces mayor a la de la tarifa residencial en Santiago, mientras que el cobro por energía consumida solo es superado por cerca de 20 CLP/kWh.

De lo anterior, nace la importancia conocer el comportamiento de los tiempos de reposición y energía no suministrada para diferentes estados de preparación de los sistemas de distribución para que la Distribuidora pueda decidir adecuadamente la inversión más costo-eficiente en equipamiento y personal.

## 4. Tópicos de ciencias de la computación y matemáticas discretas

En esta sección se hace una revisión básica sobre teoría de grafos y complejidad computacional para que el lector pueda entender a cabalidad los algoritmos utilizados en el trabajo de título. Luego se explican problemas clásicos en ciencias de la computación y que a su vez se presentan al momento de buscar una estrategia para comandar cuadrillas de restauración, ellos junto con los algoritmos disponibles para su resolución. Por último, se hace una revisión sobre los sistemas GIS, que permiten resolver problemas y visualizar información espacial rápida y eficazmente.

### 4.1 Complejidad computacional

#### 4.1.1 Crecimiento de funciones

Antes de abarcar el concepto de complejidad, se requiere comprender el concepto de crecimiento de funciones. La notación de la *Gran-O* se define como sigue [35]:

- Sea  $f$  y  $g$  funciones de dominio entero o real hacia el conjunto de los reales. Se dice que  $f(x)$  es  $O(g(x))$  si existen constantes  $C$  y  $k$  tal que  $|f(x)| \leq C|g(x)|$  para  $x > k$ .

En palabras simples, la notación define un conjunto de funciones  $f$  que crece mas lentamente que  $g$  multiplicado por una constante  $C$ . Bajo esta definición,  $f$  es  $O(g(x))$  o cualquier función con mayor crecimiento que  $f$ . Por convención, la familia que define la cota superior de  $f$  es el menor  $g$  posible. La Figura 4.1 ilustra el concepto de la *Gran-O* así como el crecimiento de funciones clásicas.

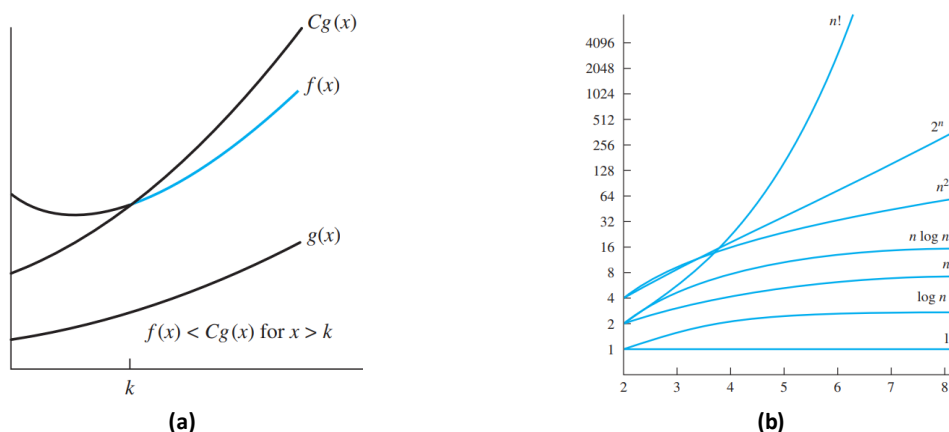


Figura 4.1 Concepto de la *Gran-O* (a) y crecimiento de funciones clásicas (b) [35]

La notación de la *Gran-O* solo define una cota superior para funciones. Cotas inferiores y doble cotas se definen por medio de la *Gran-Ω* y *Gran-Θ*.

- Sea  $f$  y  $g$  funciones de dominio entero o real hacia el conjunto de los reales. Se dice que  $f(x)$  es  $\Omega(g(x))$  si existen constantes  $C$  y  $k$  tal que  $|f(x)| \geq C|g(x)|$  para  $x > k$ .
- Sea  $f$  y  $g$  funciones de dominio entero o real hacia el conjunto de los reales. Se dice que  $f(x)$  es  $\Theta(g(x))$  si  $f(x)$  es  $O(g(x))$  y  $f(x)$  es  $\Omega(g(x))$ .

#### 4.1.2 Complejidad de algoritmos [35]

El estudio de complejidad computacional se refiere a la agrupación y clasificación de algoritmos de acuerdo a su rendimiento en lo que respecta a recursos computacionales. Un algoritmo corresponde a una serie de pasos a seguir para resolver un problema, por lo que también puede ser definido como una función de una o varias entradas, que puede o no retornar una o varias salidas.

Existen dos puntos de vista en este estudio, la complejidad de tiempo y la complejidad de espacio de un algoritmo. En lo que respecta en la complejidad de tiempo, un algoritmo puede ser descrito por el número de operaciones realizadas de acuerdo al tamaño de un input general. El término *número de operaciones* se utiliza en vez del tiempo real de ejecución pues dicho tiempo dependerá del hardware en el que esté siendo ejecutado. De acuerdo a esto, cada algoritmo está asociado a una función  $f(n)$  definida como el número de operaciones realizadas para tratar un input de tamaño  $n$ . En base a lo anterior, los algoritmos pueden ser clasificados por medio de la notación de la *Gran- $\theta$*  o *Gran- $O$* . Ejemplos de este método de clasificación son:

- Algoritmos que realizan  $m$  número de operaciones independiente de  $n$  tienen complejidad  $\theta(1)$ . Ejemplos son operaciones matemáticas entre números, asignaciones, etc.
- Algoritmos que realizan un número de operaciones igual al tamaño de su input tienen complejidad  $\theta(n)$ . Ejemplos son las operaciones entre vectores.
- Algoritmos que recorren elementos de un input dentro de otro recorrido anidado tienen complejidad  $\theta(n^2)$ . Ejemplos son las operaciones de matrices.
- Algoritmos de alta eficiencia son generalmente de tipo  $\theta(\log(n))$  u  $\theta(n \log(n))$ . Ejemplos de estos algoritmos son búsquedas binarias y ordenamiento.
- Algoritmos que requieran calcular combinaciones y/o permutaciones entre elementos de un input tendrán complejidad  $\theta(n!)$  y  $\theta(c^n)$ . Por ejemplo, un algoritmo de fuerza bruta (enumeración completa) que resuelva el problema del vendedor viajero es de complejidad  $\theta(n!)$ , mientras que un algoritmo que use programación dinámica será de complejidad  $\theta(c^n)$ . Otros ejemplos son la enumeración de todas las particiones de un set y la determinación de si dos enunciados lógicos son equivalentes.

Bajo esta clasificación, los algoritmos son catalogados como problemas **tratables** e **intratables**. Un problema se llama **tratable** si existe un algoritmo capaz de resolverlo en tiempo polinomial, es decir, que sea  $\theta(n^c)$ . Por otro lado, un problema será **intratable** si no se conoce ningún algoritmo que pueda resolverlo en tiempo polinomial.

## 4.2 Teoría de grafos [35]

Un grafo corresponde a un elemento  $G$  definido por el par de conjuntos  $(N, E)$  de nodos y aristas respectivamente y denotado generalmente como  $G = (N, E)$ . Cada elemento en  $E$  tiene al menos uno o dos nodos asociados a él. Normalmente el conjunto de nodos se define como  $N = \{1, 2, \dots, n\}$  y el conjunto de aristas como  $E = \{(i, j) | i \in N, j \in N\}$ . La Figura 4.2 ilustra ejemplos de grafos.

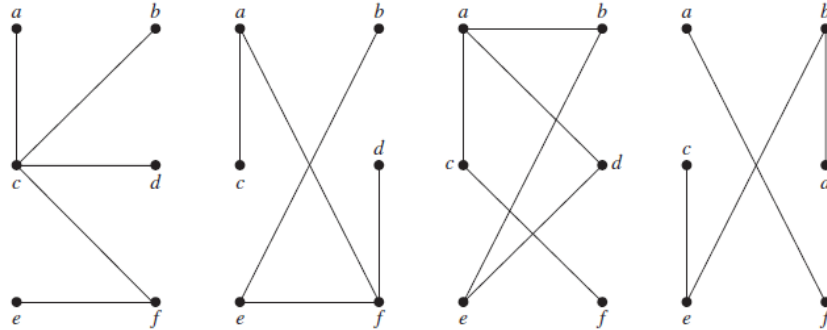


Figura 4.2 Ejemplos de grafos y arboles [35]

A continuación, se describen ciertos elementos que caracterizan a un grafo:

- Un grafo es llamado *con peso* si existe una función  $f: E \rightarrow \mathbb{R}$  tal que a cada arista le asigna un valor real denominado *peso*. Por convención la función será simbolizada como  $|\cdot|$ .
- Un grafo se dice *no-dirigido* si sus aristas no tienen orientación. Esto es  $E = \{(i, j) | i \in N, j \in N\}$ . En el caso contrario se le llama *grafo dirigido*.
- Un grafo se dice *completo* si para todo par de nodos en  $N$ , existe una arista que los une.
- El *grado* de un nodo es el número de aristas incidentes a ese nodo.
- Un *camino* es un conjunto de aristas  $\{(i_1, i_2), (i_2, i_3), \dots, (i_{k-1}, i_k)\}$  con  $i_p \neq i_q \forall p \neq q$ .
- Un *ciclo* es un conjunto de aristas  $\{(i_1, i_2), (i_2, i_3), \dots, (i_k, i_1)\}$  con  $i_p \neq i_q \forall p \neq q$ .
- Un grafo se dice *conexo* si entre cualquier par de nodos existe un camino que los une.
- Un *tour* es un ciclo donde  $k = n$ .
- En un grafo con peso, el *largo* de  $S \subseteq E$  es  $L(S) = \sum_{(i,j) \in S} |(i,j)|$ .
- Un *árbol* es un grafo conexo sin ciclos. El *árbol de expansión* de un grafo es un árbol con  $n-1$  aristas de  $G$ , donde  $n$  es el número de nodos del grafo. Un *árbol de mínima expansión* es un *árbol de expansión* de largo mínimo.

La forma en que se define un grafo lo hace ideal para modelar rutas viales, procesos secuenciales, jerarquizaciones, etc. En particular, un grafo es ideal para modelar un problema de restauración donde cuadrillas deban moverse por una red vial (aristas o arcos) con el fin de visitar transformadores dañados (nodos).

Generalmente un grafo se utiliza para modelar flujos y relaciones entre puntos. Los arboles corresponden a un caso particular de un grafo, y se utilizan principalmente como estructuras de datos de alta eficiencia. Los conceptos de grafo y árbol son fundamentales para comprender a cabalidad el problema del vendedor viajero explicado a continuación y los algoritmos utilizados para su resolución.

### 4.3 Problemas en ciencias de la computación

En esta sección se describen problemas presentados en ciencias de la computación que son relevantes para el desarrollo de la estrategia de ruteo propuesta en este trabajo junto con los algoritmos utilizados para su resolución.

#### 4.3.1 Problema del vendedor viajero (TSP)

El problema se plantea de la siguiente forma: Un vendedor requiere visitar  $n$  ciudades una única vez, es decir, realizando un tour en el grafo. El problema consiste en encontrar el tour de costo mínimo, donde el costo representa las distancias entre ciudades.

Matemáticamente se plantea de la siguiente forma: Sea  $C = c_{ij}$  la matriz de costos que representa viajar desde el nodo  $i$  hasta un nodo  $j$  en un grafo  $G = (N, E)$ . Se desea encontrar la permutación  $(i_1, i_2, i_3, \dots, i_n)$  con  $i_k \in \{1, \dots, n\}$  tal que minimice la suma  $c_{i_1 i_2} + c_{i_2 i_3} + \dots + c_{i_n i_1}$ . Dependiendo de  $C$ , el problema se clasificara como:

- **Simétrico** si  $c_{ij} = c_{ji}$ , de lo contrario será llamado **asimétrico**.
- **Métrico** si se respeta la desigualdad triangular ( $c_{ik} \leq c_{ij} + c_{jk} \forall i, j, k$ ).
- **Euclidiano** si el problema puede proyectarse en un plano, lo que es equivalente a que un problema sea simétrico y métrico.

A pesar de la simpleza del problema, encontrar un óptimo requiere un tiempo de computación prohibitivo por medio de métodos clásicos, que consisten en la resolución del siguiente problema de programación entera [36]:

$$\text{Minimizar} \quad \sum_{i \neq j} c_{ij} x_{ij} \quad 4.1$$

$$\text{Sujeto a} \quad \sum_{j=1} x_{ij} = 1, \quad i = 1, \dots, n \quad 4.2$$

$$\sum_{i=1} x_{ij} = 1, \quad j = 1, \dots, n \quad 4.3$$

$$u_i - u_j + (n - 1)x_{ij} \leq n - 1 \quad i, j = 2, \dots, n - 2, \quad i \neq j \quad 4.4$$

$$1 \leq u_i \leq n - 1 \quad i = 2, \dots, n \quad 4.5$$

$$x_{ij} \in \{0,1\}$$

4.6

La variable de decisión corresponde a  $x_{ij}$  binaria, que indica si se escogió un camino desde el nodo  $i$  hasta el nodo  $j$ . La restricción (4.2) y (4.3) son de *grados* e imponen que de un nodo solo puede entrar y salir de una arista respectivamente. Las restricción (4.4) impiden que sub-tours se puedan generar mediante la variable auxiliar  $u_i$  con dominio definido en (4.5). Por último, la restricción (4.6) restringe el dominio de  $x_{ij}$  a variables binarias.

Resolver el problema mediante el método de fuerza bruta o enumeración completa implica buscar los  $(n - 1)!$  tours factibles y comparar sus costos totales. A modo de ejemplo, si una computadora de 10 GHz se dedicara a resolver un problema TSP de 500 nodos, tardaría miles de millones de años en explorar los más de  $10^{1100}$  tours. Hasta la fecha no existe algoritmo que pueda resolver el problema TSP en tiempo polinomial, siendo parte de los problemas **intratables** con complejidad  $O(n!)$  por medio de un ILP.

A pesar de lo anterior, existen algoritmos exactos que permiten resolver el problema del vendedor viajero (desde ahora TSP) para un número relativamente grande de ciudades en tiempo razonable. Estos algoritmos utilizan técnicas avanzadas de resolución de problemas de variable entera como los métodos *branch-and-cut* y requieren recursos computacionales elevados para su ejecución, siendo de complejidad exponencial  $O(c^n)$ . A modo de ejemplo, **Concorde**, software que utiliza el método de *branch-and-cut* y considerado como una implementación en el estado del arte de algoritmo de resolución del TSP [37], logra resolver un problema simétrico de 2392 nodos en menos de 1 minuto [38], la cual se muestra en la Figura X. Para problemas que no presentan algún tipo de simetría, los cuales son de mayor interés para el presente trabajo, ya con 800 nodos el algoritmo puede tardar más de 10 minutos en resolverlo.

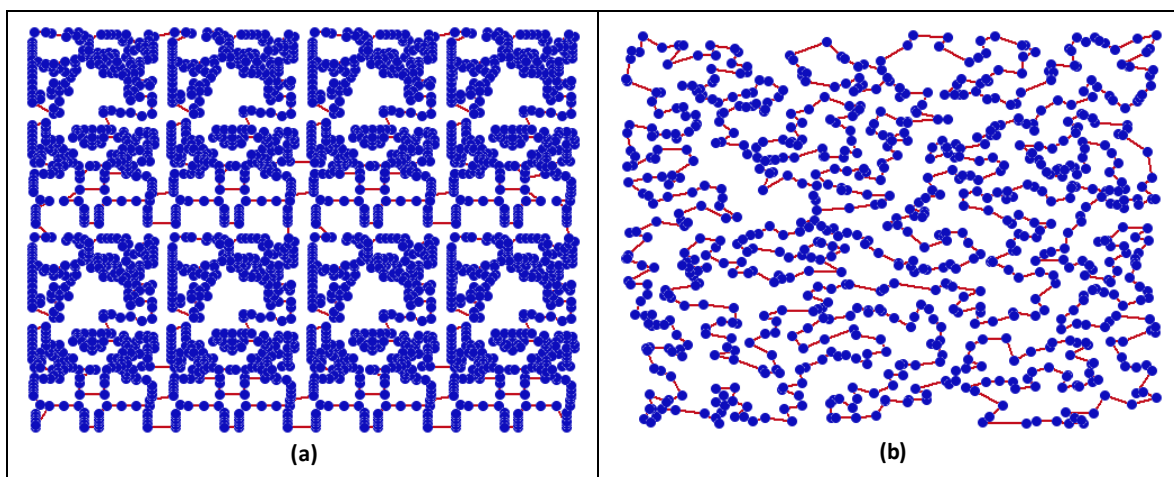


Figura 4.3 Problema simétrico de 2392 nodos (a) y problema con 800 nodos dispuestos aleatoriamente en una grilla de  $100 \times 100$  (b)

Por otro lado, también se han desarrollado algoritmos heurísticos enfocados en intentar resolver problemas de varios miles de nodos en pocos segundos. Estos algoritmos requieren pocos recursos computacionales, aunque su eficacia en términos de optimalidad dependerá

de la heurística utilizada. Se dividen en tres tipos de algoritmos: De construcción de tour, de mejoramiento de tours y compuestos.

Un algoritmo de construcción de tours parte con un sub-tour e intenta decidir cuál es la mejor forma de agregar otro nodo al tour, finalizando cuando todos los nodos pertenecen al tour. Un algoritmo de mejoramiento de tours crea un tour inicial con todos los nodos del grafo e intenta decidir de qué forma es posible intercambiar los links entre tours para crear un tour de menor largo. Un algoritmo compuesto ejecuta un algoritmo de construcción y un algoritmo de mejoramiento de tours secuencialmente.

Hasta el momento, los algoritmos que presentan mejor rendimiento tanto en tiempo como en resultados corresponden a los de mejoramiento. En particular, los algoritmos que han sobresalido en esta rama ejecutan un procedimiento llamado *movimiento  $k$ -opt*, que corresponde al mejoramiento del tour intercambiando  $k$  links. A modo de ejemplo, la Figura 4.4 muestra un *movimiento 2-opt* [39].

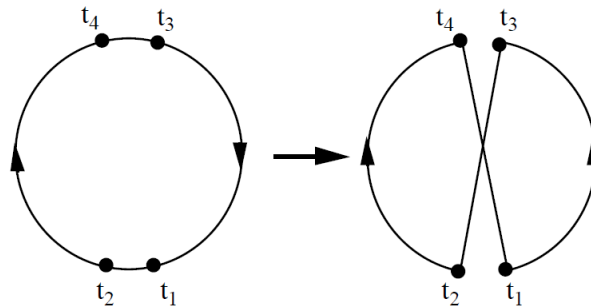


Figura 4.4 Movimiento 2-opt [39]

La idea del método se basa en el concepto de  $\lambda$ -optimalidad o simplemente  $\lambda$ -opt. Un tour es llamado  $\lambda$ -opt si no es posible obtener un tour de menor largo al intercambiar cualquier  $\lambda$  cantidad de links por otro conjunto de links de igual largo. Se puede demostrar que si un tour es  $\lambda$ -opt, entonces será  $k$ -opt para  $k \leq \lambda$ . Claramente un tour será óptimo si y solo si es  $n$ -opt. Este concepto es base para el algoritmo Lin-Kernighan, el cual será explicado a continuación.

#### 4.3.1.1 Algoritmo Lin-Kernighan [40]

Los algoritmos de mejoramiento de ruta generalmente utilizan valores  $k$  fijos para realizar *movimientos  $k$ -opt*. El Algoritmo Lin-Kernighan implementa un algoritmo con  $k$  variable de acuerdo con las circunstancias en donde se encuentre.

El algoritmo esencialmente ejerce una secuencia de links a romper y agregar a medida que avanza de nivel o iteración. Más formalmente, sea  $T \subset E$  un tour generado aleatoriamente sobre todo los nodos del grafo, con  $E$  el conjunto de aristas entre todos los nodos del problema. Sea  $X \subseteq T$  un conjunto de  $r$  links pertenecientes a un tour,  $X$  puede ser definido alternativamente como  $X = \{x_1, x_2, \dots, x_r\}$  y sea  $Y \subseteq E - T$  conjunto de  $r$  links no pertenecientes al tour con  $Y = \{y_1, y_2, \dots, y_r\}$  tal que si se rompen los links de  $X$  y se unen los links de  $Y$  a  $T - X$ , el resultado es un tour más corto. En este paso el algoritmo se encontrará en el nivel  $r$  y podrá realizar un *movimiento  $r$ -opt* si es necesario. A modo de ejemplo la Figura 4.5 muestra un movimiento 3-opt con sus links a romper ( $X$ ) y agregar ( $Y$ ).



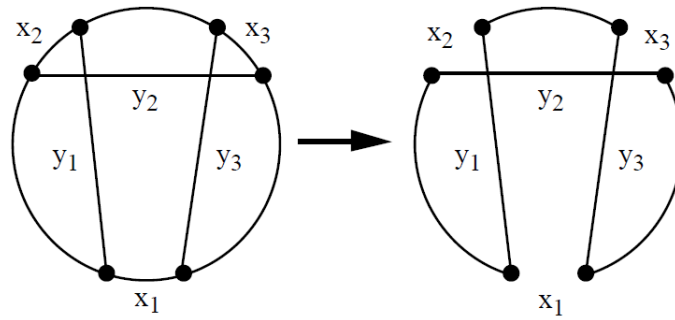


Figura 4.5 Movimiento 3-opt secuencial [39]

Al ser un proceso iterativo, parte desde la selección de un  $x_1$ , sigue con una selección de  $y_1$ , procede con una nueva selección de  $x_2$  y así sucesivamente. En resumen, en cada iteración  $i$  un par de links  $(x_i, y_i)$  son agregados a  $X$  e  $Y$  respectivamente. El algoritmo cuenta con criterios para agregar elementos a  $X$  e  $Y$  pues un par de links podría hacer que el tour fuera imposible de cerrar u otro par de links podría empeorar el tour final. Los siguientes criterios de ingreso de links deben cumplirse para evitar estas situaciones

1. **Criterio de intercambio secuencial:**  $x_i$  e  $y_i$  deben tener un nodo en común, así como  $y_i$  y  $x_{i+1}$ . A modo de entendimiento y siguiendo los pasos de la Figura 4.6, en la primera iteración ( $i = 1$ ), si se escoge el nodo  $t_1$ , existen dos opciones para escoger  $t_2$ , luego tenemos que  $x_1 = (t_1, t_2)$ . En la misma iteración se debe escoger  $t_3$  para la cual hay múltiples opciones, entonces  $y_1 = (t_2, t_3)$ . Para la segunda iteración,  $x_2 = (t_3, t_4)$  y así sucesivamente. El proceso de inserción generalizado implica que  $x_i = (t_{2i-1}, t_{2i})$  e  $y_i = (t_{2i}, t_{2i+1})$ .

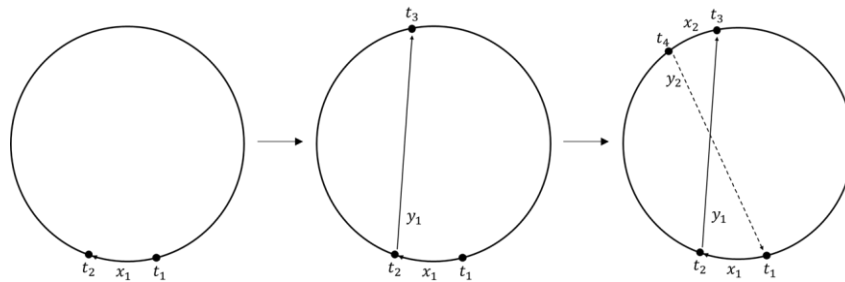


Figura 4.6 Secuencia de elección de links

Por cada iteración se genera una secuencia  $(x_1 y_1 x_2 y_2 \dots x_r y_r)$ . Si  $y_r = (t_{2r}, t_1)$ , a este intercambio se le llama *secuencial*. No todos los tipos de intercambios son secuenciales, por ejemplo en la Figura 4.7,  $y_r = (t_8, t_5)$  con  $r = 4$ .

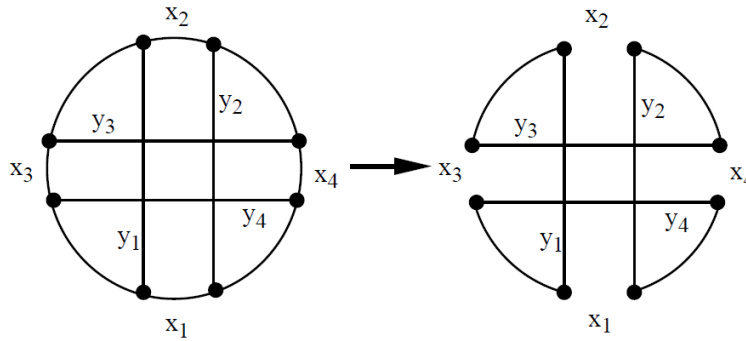


Figura 4.7 Movimiento 4-opt no secuencial [39]

2. **Criterio de factibilidad:** Para cualquier iteración  $i \geq 3$ ,  $x_i = (t_{2i-1}, t_{2i})$  debe escogerse tal que si se decide crear el link  $y_i = (t_{2i}, t_1)$ , este resulta en un tour. En palabras simples, para cada iteración, la elección de  $x_i$  debe garantizar que el tour pueda cerrarse (todos los nodos se encuentran conectados) en la misma iteración aun para el caso en que se decida seguir iterando.
3. **Criterio de ganancia positiva:** Sea  $|x_i|$  e  $|y_i|$  el costo de los respectivos links,  $g_i = |x_i| - |y_i|$  la ganancia por intercambiar  $x_i$  por  $y_i$  y  $G_i = \sum g_i$  la ganancia total de la iteración  $i$ . El criterio indica que en cada iteración se debe cumplir que  $G_i \geq 0$ .
4. **Criterio de disyunción:** El criterio de disyunción indica que  $X \cap Y = \emptyset$ , es decir, que un link a romper no puede volver a ser unido por cada iteración. Cuando se ejecuta el movimiento secuencial,  $X$  e  $Y$  vuelven a ser vacíos.

El algoritmo propiamente tal se describe a continuación

1. Generar un tour aleatorio  $T$
2. Sea  $i = 1$ , escoger  $t_1$
3. Escoger  $x_1 = (t_1, t_2) \in T$
4. Escoger  $y_1 = (t_2, t_3) \notin T$  tal que  $G_i > 0$   
Si no es posible hacer esto, ir al *Paso 12*
5. Sea  $i = i + 1$
6. Escoger  $x_i = (t_{2i-1}, t_{2i}) \in T$  tal que
  - a. Si  $t_{2i}$  se une a  $t_1$ , el resultado final es un tour  $T'$
  - b.  $x_i \neq y_s \forall s < i$
 Si  $T'$  es un mejor tour que  $T$ , entonces  $T = T'$  y volver al *Paso 2*
7. Escoger  $y_i = (t_{2i}, t_{2i+1}) \notin T$  tal que
  - a.  $G_i > 0$
  - b.  $y_i \neq x_s \forall s < i$
  - c.  $x_{i+1}$  exista
 Si existe tal  $y_i$ , volver al *Paso 5*
8. Si existe una alternativa para  $y_2$  no probada, ajustar  $i = 2$  e ir al *Paso 7*
9. Si existe una alternativa para  $x_2$  no probada, ajustar  $i = 2$  e ir al *Paso 6*
10. Si existe una alternativa para  $y_1$  no probada, ajustar  $i = 1$  e ir al *Paso 4*

11. Si existe una alternativa para  $x_1$  no probada, ajustar  $i = 1$  e ir al *Paso 3*
12. Si existe una alternativa para  $t_1$  no probada, ir al *Paso 2*
13. Detenerse (o ir al *Paso 1*)
14. Escoger el tour de menor largo en caso de haber realizado múltiples instanciaciones de tours aleatorios

El algoritmo también consta de una serie de refinamientos encargados de limitar y guiar la búsqueda de links y hacer una búsqueda de posibles soluciones que contemplen movimientos no secuenciales. Los refinamientos intentan mejorar los tiempos de ejecución del algoritmo, así como hacer frente a casos particulares del mismo. A continuación, se describen estos refinamientos:

- **Limitación en la búsqueda de links:**
  - La búsqueda del link  $y_i = (t_{2i}, t_{2i+1})$  a ingresar en  $T$  está limitada a los nodos más cercanos a  $t_{2i}$ .
  - Para  $i \geq 4$ , ningún link  $x_i$  debe ser roto si pertenece a la solución común encontrada en 2 a 5 tours diferentes.
  - La búsqueda de mejoramiento es detenida si el tour  $T$  es igual a un tour encontrado en previas soluciones
- **Dirección en la búsqueda de links:**
  - Cuando el link  $y_i$  ( $i \geq 2$ ) debe ser escogido de entre múltiples opciones. Cada link posee la prioridad  $|x_{i+1}| - |y_i|$ . Dependiendo de los parámetros impuestos en el algoritmo, se podrá escoger solo la de mayor prioridad o se podrá hacer *backtracking* por prioridad
  - Si hay dos alternativas para  $x_4$ , escoger la que tenga un  $|x_4|$  mayor.
- **Movimientos no-secuenciales post-iteración:**
  - Cuando no es posible encontrar mejores tours por medio de cambios secuenciales, el algoritmo prueba movimientos 4-opt no secuenciales en links permitidos a romper.

#### 4.3.1.2 Algoritmo Lin-Kernighan-Helsgaun-1 [39]

El algoritmo Lin-Kernighan (LK) fue bien recibido en su época pues lograba encontrar soluciones óptimas rápidamente para problemas de hasta 300 nodos. A pesar de ello, el algoritmo era difícil de implementar y la calidad de sus soluciones caía rápidamente para problemas de mayor tamaño, lo que motivó a científicos a crear sus propias versiones del algoritmo LK.

Uno de las modificaciones que ha tenido mayor éxito ha sido el algoritmo Lin-Kernighan-Helsgaun (LKH) el cual en su primera versión es capaz de resolver problemas del orden de los 10.000 nodos en minutos con resultados cercanos al óptimo. De hecho, el algoritmo fue capaz de encontrar rutas mínimas desconocidas en su época. Si bien el algoritmo sigue los mismos pasos que el algoritmo LK, este realiza ciertas modificaciones en los criterios y refinamientos originales.

##### 4.3.1.2.1 Búsqueda de links $y_i$

En el algoritmo LK la búsqueda de links  $y_i = (t_{2i}, t_{2+1})$  está restringida principalmente por los 5 nodos más cercanos a  $t_{2i}$ . El algoritmo LKH-1 modifica el concepto de *cercanía* por medio del concepto  $\alpha$ -nearness. Para comprender este criterio primero es necesario hacer las siguientes definiciones:

- Un **1-tree** del grafo  $G = (N, E)$  es un árbol de mínima expansión en el conjunto  $N \setminus \{1\}$  combinado con dos aristas incidentes al nodo 1, donde el nodo 1 es de elección arbitraria.
- Un 1-tree mínimo es un 1-tree de largo mínimo.

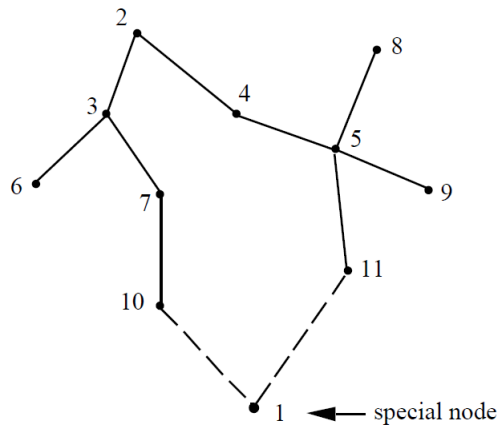


Figura 4.8 Ejemplo de un 1-tree [39]

La definición de  $\alpha$ -nearness es la siguiente:

- Sea  $T$  un 1-tree mínimo de largo  $L(T)$  y sea  $T^+(i, j)$  un 1-tree de largo mínimo que incorpora la arista  $(i, j)$ , entonces el  $\alpha$ -nearness de  $(i, j)$  se define como la cantidad  $\alpha(i, j) = L(T^+(i, j)) - L(T)$ .

Las razones de la utilización de esta métrica son varias, entre ellas se señala que un tour óptimo contiene cerca de un 70% a 80% de las aristas de un 1-tree, así como el hecho de que un tour óptimo es un 1-tree con todos sus nodos de grado 2. De hecho, si un 1-tree corresponde a un tour, entonces este será el tour óptimo del problema TSP. De acuerdo a la definición de  $\alpha(i, j)$ , si  $(i, j)$  es parte del 1-tree, entonces tendrán distancia 0. De lo contrario tendrá distancia mayor a cero.

Los resultados preliminares del algoritmo LKH muestran que la métrica  $\alpha$ -nearness logra mejores resultados que la métrica euclidiana convencional. Además, el algoritmo está implementado para que el cálculo de las distancias  $\alpha$  sea en tiempo  $O(n^2)$ , utilizando espacio  $O(n)$ . El algoritmo LKH no solo se queda con la definición  $\alpha$ -nearness para apoyar el criterio de búsqueda, sino que también la utiliza en un proceso de pre-optimización para encontrar el óptimo en caso de ser posible, considerando lo siguiente:

- Todo tour es un 1-tree. Por consiguiente, un 1-tree corresponde a una cota inferior del largo del tour óptimo.
- Variar el largo de las aristas incidentes a un mismo nodo en una constante mantiene la solución del problema TSP. Luego, si definimos la transformación de la matriz de costo  $D$  con respecto a la matriz original  $C$  como  $d_{ij} = c_{ij} + \pi_i + \pi_j$ , el tour óptimo

de  $D$  es igual al tour óptimo de  $C$ , con un largo incrementado por  $2\sum\pi_i$ . Por otro lado, las aristas que componen al 1-tree de  $D$  si cambian con respecto a los de  $C$ .

- Si  $T_\pi$  es el 1-tree mínimo de  $D$ , entonces  $L(T_\pi)$  es una cota inferior del tour óptimo de  $D$ . Por lo tanto,  $w(\pi) = L(T_\pi) - 2\sum\pi_i$  es una cota inferior del largo del tour óptimo de  $C$ .

El primer problema a resolver por el algoritmo LKH-1 es el de encontrar el vector  $\pi = (\pi_1, \dots, \pi_n)$  que maximice la función  $w(\pi) = L(T_\pi) - 2\sum\pi_i$ . Si en el proceso de  $T_\pi$  es un tour, entonces el tour óptimo se habrá encontrado. En caso contrario, los valores  $\alpha(i, j)$  calculados de  $D$  corresponden a mejores candidatos a ser óptimos que los calculados de  $C$ .

El cálculo de los valores en  $\pi$  se realiza por medio de una optimización por sub-gradientes, donde los cambios en  $\pi$  hacen énfasis en penalizar aristas del 1-tree asociadas a nodos de grado mayor a 2. En términos simples, por cada iteración las aristas con grado mayor a 2 en el 1-tree se hacen más grandes, las asociadas a nodos con un grado se hacen más pequeñas y las asociadas a nodos de 2 grados se mantienen igual. El cálculo de  $\pi$  se hace en pocas iteraciones ya que el objetivo principal de esta etapa es encontrar mejores valores de  $\alpha(i, j)$  a los obtenidos directamente de  $C$ .

#### 4.3.1.2.2 Eliminación de aristas

En lo que respecta a la elección de candidatos para el set  $X$ , se siguen las siguientes reglas:

- La primera arista a romper  $x_1$  no debe pertenecer a la mejor solución encontrada. En el caso en que no se conozca dicha solución, no debe pertenecer a su 1-tree.
- La última arista a ser ingresada no debe pertenecer a una arista previamente ingresada. Esta regla corresponde a una relajación del criterio de disyunción.

#### 4.3.1.2.3 Movimientos básicos

En el algoritmo LK cada movimiento  $r$ -opt de cada iteración puede descomponerse en movimientos 2-opt y 3-opt secuenciales. El algoritmo LKH-1 tiene como movimiento básico al 5-opt secuencial. Otro aspecto de este movimiento 5-opt es que los movimientos anteriores a este pueden no ser secuenciales.

#### 4.3.1.2.4 Tour inicial

El algoritmo LKH-1 genera un tour inicial de acuerdo con los valores  $\alpha(i, j)$  generados en el proceso de optimización por sub-gradiente. El tour inicial intenta contener el mayor número de candidatos posibles de acuerdo con el ordenamiento dado por  $\alpha$ -nearness. Si esto no es posible, se elige una arista al azar.

El ejemplo de la Figura 4.9 se muestra para que el lector pueda entender a cabalidad el algoritmo LKH. Los pasos seguidos son los siguientes

- 1) Se instancian los nodos del problema TSP. El problema consta de 11 nodos dispuestos aleatoriamente
- 2) Se asigna una enumeración a los nodos y se encuentra su 1-tree mínimo
- 3) En base al 1-tree se genera un tour inicial. El tour inicial es de largo 4937.

- 4) Se escoge  $x_1$  no perteneciente al 1-tree ya que este es el primer tour. Luego se escoge el nodo con el menor  $\alpha$ -nearness, colocados en rojo.
- 5) Se escoge  $y_1$  y se calcula la ganancia. Se tiene que  $G_1 = 410 > 0$
- 6) Se escoge  $x_2$  y se calculan los  $\alpha$ -nearness, colocados en rojo
- 7) Se escoge  $y_2$  y se calcula la ganancia. Se tiene que  $G_2 = 410 + (-61) = 349$
- 8) Como  $y_2$  termina en  $t_1 = 10$ , se realiza el movimiento secuencial de tipo 2-opt. El nuevo tour es de largo 4588 y es óptimo.

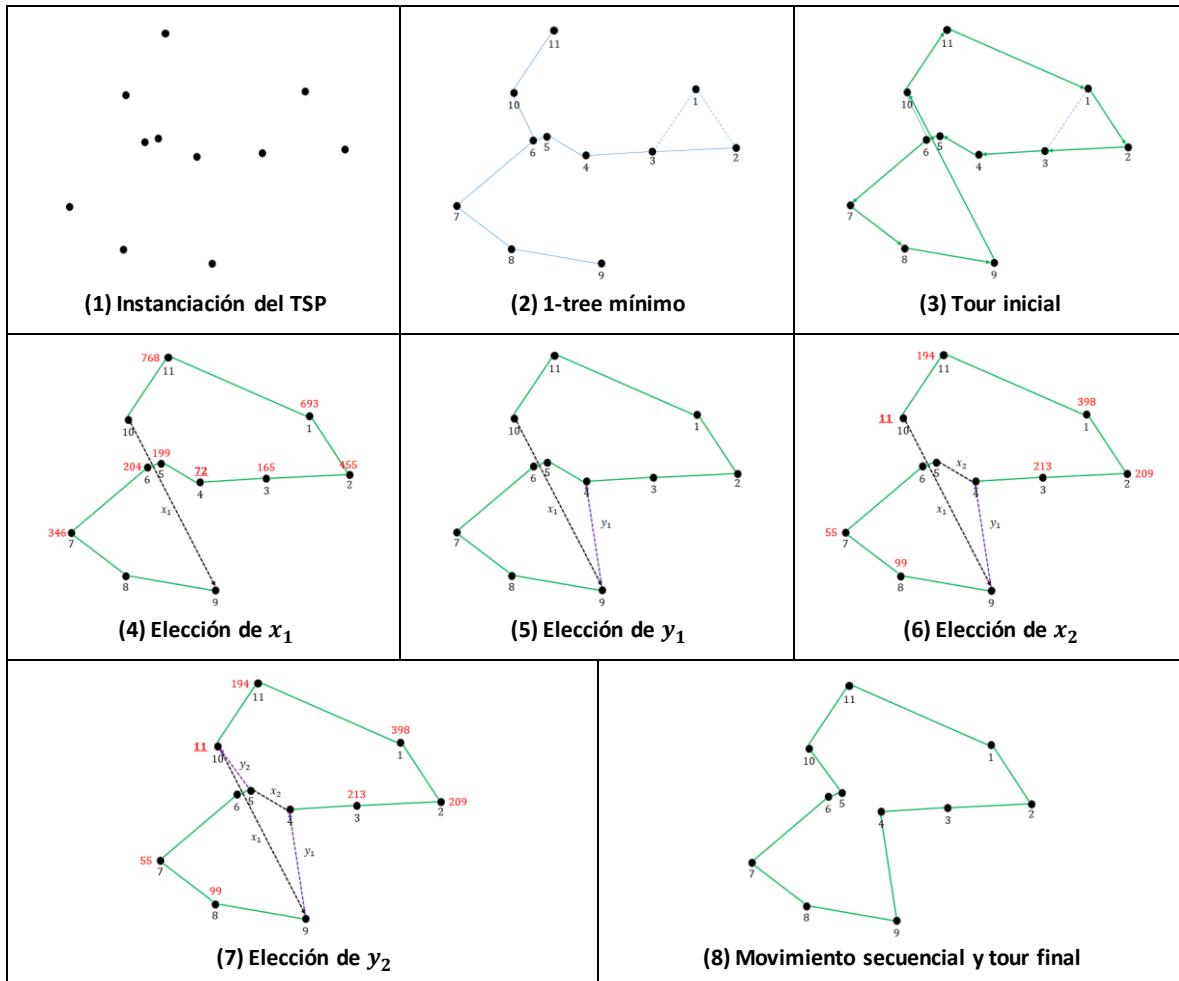


Figura 4.9 Ejemplo del algoritmo LKH-1

#### 4.3.1.3 Algoritmo Lin-Kernighan-Helsgaun-2 [41]

Como se mencionó en la sección anterior, el algoritmo LKH-1 fue capaz de encontrar tours factibles con un grado de error de hasta un 1% con respecto al óptimo e incluso fue capaz de encontrar mejores sub-óptimos de problemas no resueltos en esa época, todo ello con una complejidad en tiempo de  $O(n^{2.2})$  en promedio. A modo de potenciar aún más este algoritmo, se crea la segunda versión LKH-2 capaz de resolver problemas de hasta 100.000 nodos en pocos minutos. Los cambios con respecto a LKH-1 están centrados principalmente en

implementaciones más eficientes de los puntos anteriores y otros aspectos. Los cambios de mayor importancia se mencionan a continuación.

#### 4.3.1.3.1 Movimientos básicos

El algoritmo puede realizar movimientos  $k$ -op básicos, donde  $k$  es un entero entregado por el usuario. El algoritmo también es capaz de examinar movimientos no secuenciales y utilizarlos para mejorar un tour. Esto se debe principalmente a que, si un conjunto de links  $X$  e  $Y$  da un tour infactible, este puede volverse factible por medio de una combinación de movimientos secuenciales por cada sub-tour generado. La Figura 4.10 ilustra como un tour puede volverse infactible y como un movimiento no-secuencial puede volverlo factible:

- 1) Se instancia el problema de 20 nodos y se crea un tour inicial
- 2) Se indican los links a romper y unir para un movimiento 4-opt secuencial
- 3) Se realiza el movimiento 4-opt secuencial, lo que da como resultado un tour infactible. En concreto, se tienen 4 sub-tours conformado por todos los nodos del grafo.
- 4) Se indican los links a romper y unir para volver el tour factible. La secuencia de links  $\{x_1, y_1, x_2, y_2\}$  y  $\{x_3, y_3, x_4, y_4, x_5, y_5, x_6, y_6, x_7, y_7\}$  corresponden a movimientos 2-opt y 5-opt secuenciales. Por otro lado, el inicio de  $x_1$  es diferente al fin de  $y_7$ , dando como resultado un movimiento final de tipo 7-opt no-secuencial.
- 5) Se ejecuta el movimiento 7-opt no-secuencial, lo que da como resultado un tour factible.

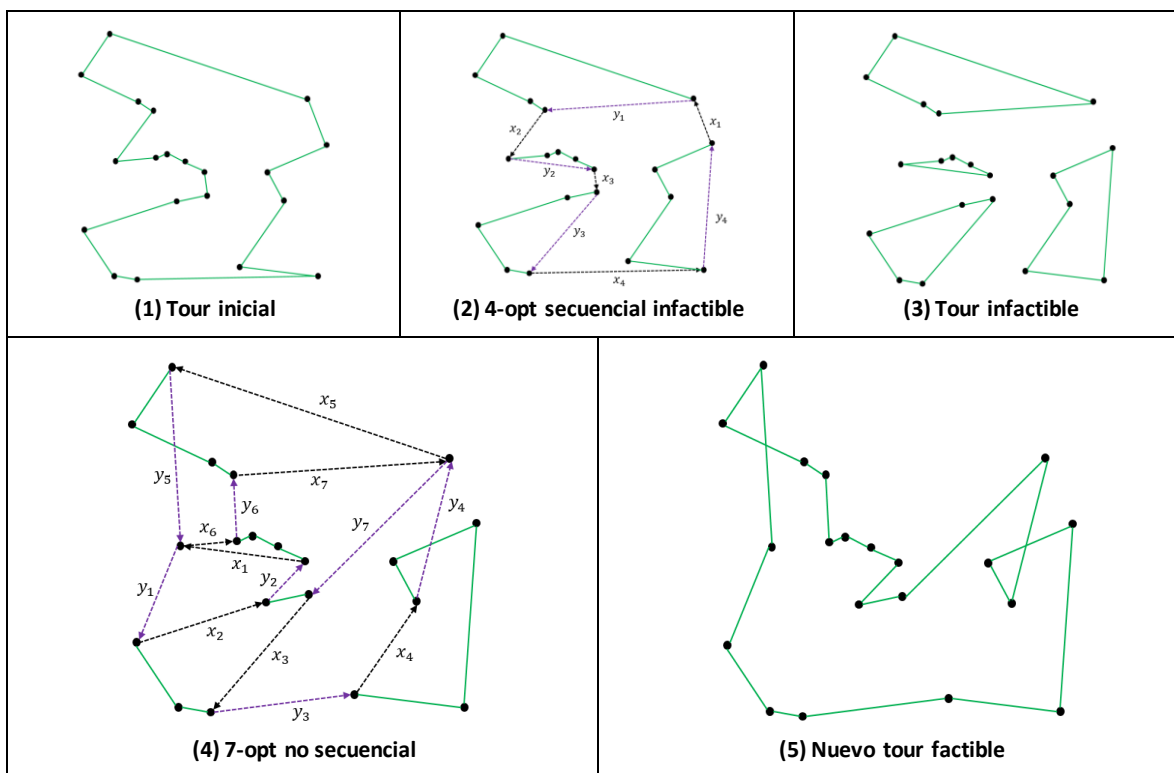


Figura 4.10 Movimiento 4-opt secuencial infactible y movimiento 7-opt no-secuencial para volver al tour factible

La implementación de movimientos no-secuenciales no solo mejora sustancialmente los resultados del algoritmo, sino que también logra reducir en promedio los tiempos en un 40% con respecto a la búsqueda de tours óptimos por medio de intercambios únicamente secuenciales.

#### 4.3.1.3.2 Métodos de particionamiento

Se agregan hasta 6 diferentes métodos de particionamiento para reducir la complejidad de problemas extremadamente grandes. Debido a que se divide el problema, el autor también implementa métodos para unir dichos sub-problemas.

La implementación de todos los métodos y funciones no son mayores a  $O(\sqrt{n})$ , por lo que la complejidad del algoritmo original sigue siendo  $O(n^{2.2})$ . El algoritmo logró encontrar mejores soluciones a problemas no resueltos, como el problema de todas las ciudades del mundo llamado *World*. Este problema consiste en un TSP de 1.904.711 nodos que representan localizaciones en el mundo. Su solución se muestra en la Figura 7.14.



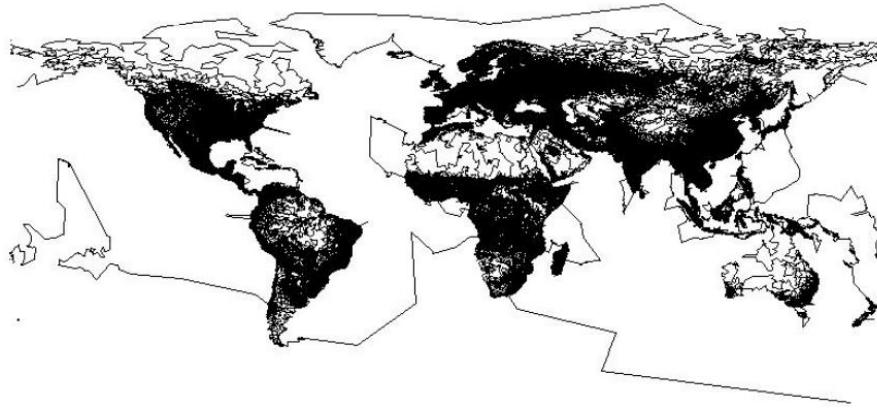


Figura 4.11 Resultado del algoritmo LKH-2 para el problema World

A modo de ejemplo, se comparan los tiempos y porcentaje de error entregados por 1000 corridas de **LKH-2** para diferentes tipos de problemas con respecto al resultado entregado por el algoritmo **Concorde**. Los problemas se corrieron en un computador con procesador de 3.1 GHz de 4 núcleos, 16 GB de memoria RAM y en el sistema operativo Windows 8.1 de 64 bits.

Tabla 4.1 Comparación de rendimientos entre Concorde y LKH-2

Problema	Nodos	Resultado Concorde	Resultado LKH-2 (Max)	Error porcentual	Tiempo Concorde [s]	Tiempo LKH-2 [s]
<b>kroA100</b>	100	21.282	21282	0,00%	0,28	0,06
<b>lin318</b>	318	42.029	42.454	1,01%	3,67	0,54
<b>rat575</b>	575	6.773	6.795	0,32%	25,64	0,15
<b>u724</b>	724	41.910	42.035	0,29%	24,97	0,26
<b>pr1002</b>	1002	259.045	260.877	0,70%	28,72	0,96

Los problemas de la Tabla 4.1 corresponden a TSPs clásicos estudiados en la academia. El problema **kroA100** es un TSP de 100 nodos utilizado para probar un algoritmo heurístico [42], mientras que el resto de los casos corresponden a los denominados *Drilling Problems*, que consiste en un problema aplicado donde se desea saber el recorrido que debe realizar una máquina perforadora para unir circuitos en una placa PCB [43].

#### 4.3.2 Problema de Ruteo de Vehículos con Capacidad Acumulada

También llamado CCVRP (Cumulative Capacitated Vehicle Routing Problem), corresponde a una extensión relativamente nueva del problema TSP en donde se pretende minimizar el tiempo de espera total de los clientes representados por los nodos. Cada cliente posee una demanda a ser cubierta por un número de vehículos con capacidad definida, lo cual lo hace ideal para modelar problemas de ayuda humanitaria en donde se desea atender zonas de riesgo en el menor tiempo posible. La diferencia fundamental con el problema TSP es el

hecho de que se minimice la suma de las distancias vistas por todos los nodos, en vez de la distancia total vista únicamente por el nodo inicial.

Sea  $G = (V, E, W)$  un grafo no-dirigido con  $V = \{0, \dots, n, n + 1\}$  el conjunto de nodos donde 0 y  $n - 1$  corresponden ambos al depósito (lugar de inicio),  $E$  el conjunto de aristas asociadas a la matriz de costos  $W$  con  $w_{ij} = w_{ji}$  para  $i, j \in E$ . Sea  $Q$  la capacidad de los vehículos,  $R$  el tamaño de la flota de vehículos y  $q_i$  la demanda de los clientes en  $V' = V \setminus \{0, n + 1\}$ . La modelación del problema como un problema de programación lineal entero se muestra a continuación [44]:

$$\text{Minimizar} \quad \sum_{k=1}^R \sum_{i \in V'} t_i^k \quad 4.7$$

$$\text{Sujeto a} \quad \sum_{i \in V} x_{ij}^k = \sum_{i \in V} x_{ji}^k, \quad \forall j \in V', k \in \{1, \dots, R\} \quad 4.8$$

$$\sum_{k=1}^R \sum_{j \in V} x_{ij}^k = 1 \quad \forall i \in V' \quad 4.9$$

$$\sum_{i \in V'} \sum_{j \in V} x_{ij}^k q_i \leq Q \quad \forall k \in \{1, \dots, R\} \quad 4.10$$

$$\sum_{j \in V} x_{0j}^k = 1 \quad \forall k \in \{1, \dots, R\} \quad 4.11$$

$$\sum_{i \in V} x_{i, n+1}^k = 1 \quad \forall k \in \{1, \dots, R\} \quad 4.12$$

$$t_i^k + w_{ij} - (1 - x_{ij}^k)T \leq t_j^k \quad \forall i \in V \setminus \{n + 1\}, \forall j \in V, \quad 4.13$$

$$\forall k \in \{1, \dots, R\}$$

$$t_i^k \geq 0 \quad \forall i \in V, \forall k \in \{1, \dots, R\} \quad 4.14$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i \in V, \forall j \in V, i \neq j, \forall k \in \{1, \dots, R\} \quad 4.15$$

Las variables de decisión corresponden a  $x_{ij}^k$ , variable binaria, que es la ruta a tomar por el vehículo  $k$  y la variable positiva  $t_i^k$  que es el tiempo de espera del nodo  $i$  al ser visitado por el vehículo  $k$ . La restricción (4.8) especifica que un vehículo entrando al nodo  $i$  debe salir de él. La restricción (4.9) impone que un cliente es atendido únicamente por un vehículo. La restricción (4.10) corresponde a la restricción de capacidad. Las restricciones (4.11) y (4.12) aseguran que el nodo inicial y final de un vehículo es asignado únicamente al depósito. La restricción (4.13) define la variable  $t_i$  de acuerdo a la matriz  $W$  y al nodo previamente

visitado por un vehículo  $k$ , por medio de una variable  $T$  grande. Por último, las restricciones (4.14) y (4.15) definen los dominios de las variables de decisión del problema.

El problema CCVRP resulta ser una complejización del problema TSP y solo se conocen soluciones para problemas del orden de pocas decenas de nodos [45].

#### 4.3.2.1 Algoritmo Lin-Kernighan-Helsgaun-3 [46]

La tercera versión del algoritmo LKH con su última actualización lanzada en julio del 2018 cuenta con una serie de procedimientos que le permiten resolver diferentes tipos de extensiones del problema TSP, entre ellas el problema CCVRP. Esto se realiza de la siguiente forma:

##### 4.3.2.1.1 Transformaciones de problemas

Un problema asimétrico con  $n$  nodos se transforma en un problema simétrico con  $2n$  nodos. Esto se hace para problemas con ventanas de tiempos y restricciones similares. Por otro lado, un problema con  $m$  vendedores o vehículos puede transformarse en un TSP con repetición  $m - 1$  veces del nodo origen.

##### 4.3.2.1.2 Funciones de penalización

El algoritmo utiliza funciones de penalización para los casos donde se tengan restricciones de otro tipo no relacionados al TSP como por ejemplo demanda en los nodos. Las funciones de penalización reciben como input un tour y entregan un valor real en función de que tanto se ha violado una restricción. Cada tour tiene asociado un par  $(P, C)$  con  $P$  la penalización del tour y  $C$  su costo. Se dice que  $T_1$  es mejor que  $T_2$  si  $(P_1 < P_2) \vee (P_1 = P_2 \wedge C_1 < C_2)$ .

Los resultados con LKH-3 muestran que el algoritmo es capaz de encontrar mejores rutas a las ya conocidas en la gran mayoría de las pruebas propuestas. Por ejemplo, el problema **P-n76-k4** de 76 nodos y 4 vehículos, no es posible resolver en menos de una hora con un algoritmo exacto, mientras que LKH-3 es capaz de encontrar una mejor cota superior (una mejora del 25%) en ese tiempo. La resolución del problema por medio de LKH-3 se muestra en la Figura 4.12.

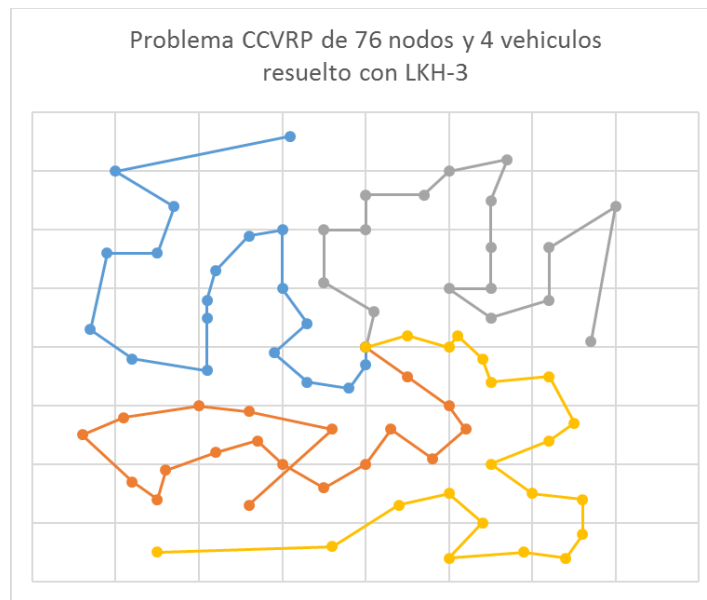


Figura 4.12 Resolución del problema P-n76-k4 por medio del algoritmo LKH-3

### 4.3.3 Problema del camino más corto

El problema se plantea de la siguiente forma: Sea  $G = (N, E)$  un grafo no-dirigido con peso y conexo. Se desea conocer el camino más corto entre los nodos  $a, z \in N$ . A diferencia del problema TSP, el camino recorrido no necesariamente debe pasar por todos los nodos, el grafo no necesariamente debe ser completo y  $a \neq z$ . En palabras simples, se desea conocer el camino más corto entre dos nodos de un grafo.

Esta clase de problemas es de especial interés para el presente trabajo de título pues, sabiendo que una red vial puede ser representado por un grafo, el largo del camino que uno dos nodos de interés es único. Si el set de nodos de interés es fijo y de menor orden al conjunto de nodos que conforman la red vial, entonces el ruteo de cuadrillas puede ser representado por la elección de dichos caminos en vez de una secuencia de nodos, haciendo el algoritmo de ruteo más rápido y ligero computacionalmente.

#### 4.3.3.1 Algoritmo de Dijkstra [35]

Este algoritmo es de tipo *greedy*, es decir, en cada paso intenta buscar la mejor solución posible. Normalmente los algoritmos de tipo greedy convergen en óptimos locales, pero dada las características del problema su correctitud<sup>3</sup> está asegurada. El algoritmo es de complejidad en tiempo de  $O(|E| + |N| \log(|N|))$  y sigue los siguientes pasos

1. Asignar a cada nodo del grafo con un valor infinito e inicializa el conjunto  $S = \emptyset$ .
2. Asignar al nodo inicial con valor cero

<sup>3</sup> Un programa  $S$  se dice correcto para toda entrada  $p$  verdadera, el programa termina y entrega una salida  $q$  verdadera. La demostración es llamada *Verificación formal* y se hace por inferencia, inducción o contradicción lógica. Los modelos matemáticos de  $S$ ,  $p$  y  $q$  dependen del tipo de demostración [63]

3. Asignar  $u$  al nodo con menor valor y que no esté en  $S$
4. Para cada nodo  $v$  adyacente a  $u$  y que no este en  $S$ 
  - a. Si el valor de  $v$  es mayor a  $c_{u,v}$  mas el valor de  $u$ , actualizar el valor de  $v$
5. Agregar  $u$  a  $S$  y volver a (3)
6. Finalizar cuando  $z \in S$

La Figura 4.13 muestra un ejemplo de la ejecución del algoritmo de Dijkstra para un grafo dirigido no completo.

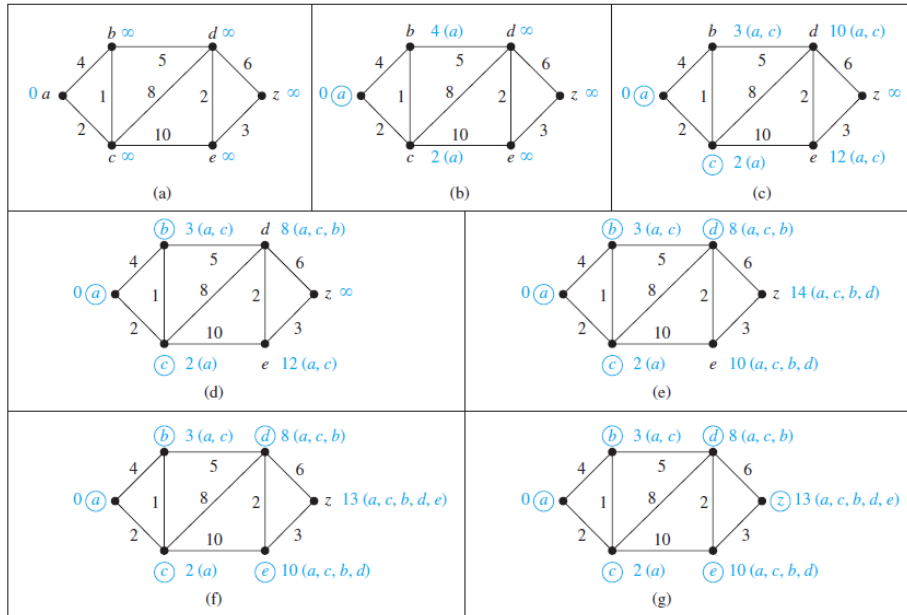


Figura 4.13 Búsqueda del camino más corto entre  $a$  y  $z$  por medio de Dijkstra [35]

### 4.3.4 Problema de agrupamiento

El problema de agrupamiento consiste en determinar la clasificación de un set de datos, ya sea en un espacio o con características que definen cada uno de sus elementos. El problema de agrupamiento resulta ser complejo ya que a priori no se sabe la cantidad de grupos que representa a un set de datos. Por ello, los algoritmos de agrupamiento generalmente dejan al usuario decidir el número de grupos impuesto al set.

La resolución de este problema no solo permite clasificar datos concentrados espacialmente, sino que también permite implementar los algoritmos de tipo “*Dividir para conquistar*”, es decir, dividir un problema de gran tamaño en pequeños sub-problemas, los que se resuelven y al ser unidos logran resolver el problema principal. En lo que concierne el problema TSP, esto sería transformar un TSP del orden de los 20 mil nodos en varios del orden de los mil nodos y uniendo cada uno de ellos de forma conveniente.

#### 4.3.4.1 Diagrama de Voronoi

El diagrama de Voronoi, para el espacio en  $\mathbb{R}^2$ , de un conjunto de “sitios” o “puntos generadores” es una colección de regiones que dividen el espacio, tal que cada región

contiene uno de los “puntos generadores” y además cada punto dentro de una región asociada a ese sitio está más cercano a dicho sitio que a cualquier otro. La Figura 4.14 ilustra este concepto gráficamente [47].

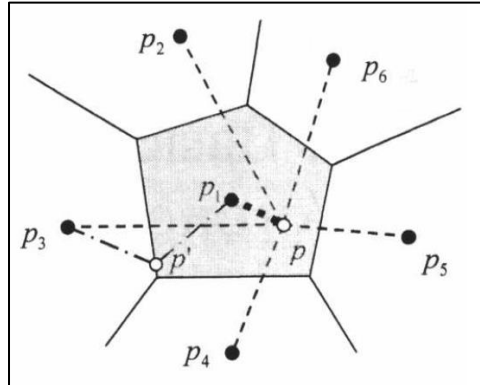


Figura 4.14 Diagrama de Voronoi y distancias de los sitios a un punto [47]

Formalmente, sea  $p_i$  un sitio con coordenadas  $(x_{i1}, x_{i2})$ , con su vector asociado  $\vec{x}_i$ . Sea  $P = \{p_1, p_2, \dots, p_n\} \in \mathbb{R}^2$  el conjunto de puntos generadores o sitios en el plano, donde además se cumple que  $p_i \neq p_j$  si  $i \neq j$ . Una región de Voronoi de  $p_i$  se define como

$$V(p_i) = \{\vec{x} \mid \|\vec{x} - \vec{x}_i\| \leq \|\vec{x} - \vec{x}_j\| \forall j \in P - \{p_i\}\}$$

Los polígonos de Voronoi son particularmente útiles para agrupar elementos de acuerdo con su cercanía a los puntos generadores. Los algoritmos encargados de encontrar polígonos de Voronoi son de complejidad  $O(n \log(n))$  con  $n$  el número de puntos generadores.

#### 4.3.4.2 Algoritmo K-means

El algoritmo K-means consiste principalmente en agrupar un conjunto de datos en un número de clusters ya definido previamente por el usuario. El número de clusters está definido por los centroides, los cuales clasifican los elementos a agrupar de acuerdo a la distancia que se encuentren de él. A grandes rasgos, para datos dispuestos espacialmente, el algoritmo trabaja de la siguiente forma [48]:

1. Se definen los  $k$  centroides aleatoriamente asociado a los clusters indexados por ellos  $\{m^{(k)}\}$
2. Se minimiza la varianza intra-grupo, es decir, minimizar la distancia al cuadrado de cada punto a los centroides. Matemáticamente se resuelve el siguiente problema de optimización.

$$\operatorname{argmin}_{\{m^{(k)}\}} \left\{ \sum_{i=1}^k \sum_{x \in \{m^{(k)}\}} \|x - m^{(k)}\|^2 \right\} = \operatorname{argmin}_{\{m^{(k)}\}} \left\{ \sum_{i=1}^k |\{m^{(k)}\}| \operatorname{Var}(\{m^{(k)}\}) \right\}$$

En palabras simples, se ingresan los puntos al conjunto  $\{m^{(k)}\}$  al centroide más cercano.

3. Se actualizan los centroides como el punto medio de todos los elementos que conforman su respectivo cluster  $\{m^{(k)}\}$ , es decir, para cada  $k$  centroide se calcula

$$m^{(k)} = \frac{\sum_{x \in \{m^{(k)}\}} x}{|\{m^{(k)}\}|}$$

4. Se vuelve a (2) hasta que converja el algoritmo por número de iteraciones máximas o por convergencia del centroide.

El agrupamiento K-Means es utilizado ampliamente para efectos de clusterización de datos discretos repartidos espacialmente. La Figura 4.15 muestra cómo opera el algoritmo siguiendo una secuencia de pasos de izquierda a derecha, donde el paso de asignación (Assignment) resuelve el problema de optimización, es decir, asigna los  $n$  puntos a diferentes centroides representados por los círculos de mayor tamaño, y el paso de actualización (Update) recalcula la posición de los centroides.

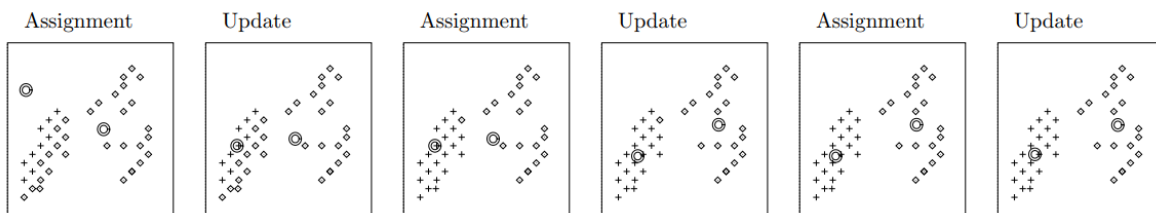


Figura 4.15 Ejemplo del algoritmo K-Means [48]

El algoritmo K-Means logra resolver el problema de agrupamiento en un tiempo  $O(nkdi)$  con  $n$  el número de datos de  $d$  dimensiones a ser clusterizados,  $k$  el número de clusters e  $i$  el número de iteraciones a realizar.

#### 4.4 Sistema de Información Georreferencial

Un sistema de información geográfica es un conjunto de herramientas que integra y relaciona diversos componentes que permiten la organización, almacenamiento, manipulación, análisis y modelación de grandes cantidades de datos procedentes del mundo real que están vinculados a una referencia espacial, facilitando la incorporación de aspectos sociales-culturales, económicos y ambientales que conducen a la toma de decisiones de una manera más simple y eficaz.

Este tipo de programas permiten, por tanto, realizar análisis de datos alfanuméricos asociados a una componente espacial, con representación de los mismos, y realizar operaciones sobre las imágenes existentes. El hecho de trabajar sobre un sistema de representación geoespacial implica procesar una gran cantidad de datos (calles, localizaciones, zonas de interés) lo cual es posible lograr mediante la estructura de tipo *Shapefile*. El formato *shapefile* (*.shp*) permite almacenar la ubicación, forma y atributos de entidades geográficas. Estos formatos son ampliamente utilizados por programas de georreferenciación gracias a la capacidad de almacenamiento en forma de tablas, que permiten realizar *consultas* (*queries*) al igual que en lenguajes de bases de datos, los que a su vez permiten procesar grandes volúmenes de datos a gran velocidad [49].

#### **4.4.1 Quantum GIS**

También conocido como QGIS, es un sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. Permite a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones. Se caracteriza principalmente por ser un software libre y de código abierto y que es posible automatizar procesos por medio de la herramienta como PyQGIS, razón por la que se utilizará para el trabajo de memoria.

El uso de esta herramienta es esencial en el trabajo de memoria pues permite procesar grandes volúmenes de datos con propiedades espaciales a gran velocidad y aplicar algoritmos que relacionen diferentes elementos tales como nodos y caminos.



# 5. Modelo y estrategia propuesta

---

## 5.1 Introducción

En el presente capítulo se describirá el modelo y estrategia propuesta para realizar ruteo de cuadrillas. A modo de introducción, la Figura 5.1 muestra los tres grandes bloques por los que pasa el modelo de ruteo de cuadrillas.

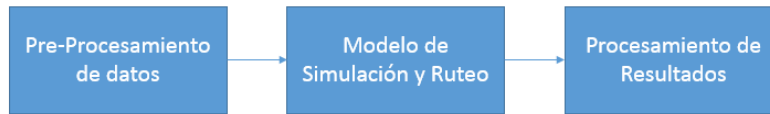


Figura 5.1 Bloques que definen el modelo propuesto

A grandes rasgos, el bloque **Pre-Procesamiento** trata todos los datos obtenidos y disponibles públicamente para que el Modelo de Simulación y Ruteo pueda procesarlos y utilizarlos adecuadamente. El bloque **Modelo de Simulación y Ruteo**, tal como su nombre lo indica, simula el comportamiento de las cuadrillas y transformadores referenciales ante un evento sísmico y entrega resultados en formato de texto plano. Los archivos de salida del bloque son tratados por el bloque **Procesamiento de Resultados** que finalmente desprende las métricas de interés. El espíritu del modelo y estrategia de ruteo propuesta se describe a continuación:

1. Las localizaciones de los transformadores de Media Tensión referenciales modelan el conjunto de equipos aguas debajo del transformador propiedad de la distribuidora, incluyendo al mismo transformador. A estos puntos se les llamará simplemente *transformadores*.
2. Los transformadores tienen una probabilidad de sufrir un cierto estado de daño de acuerdo con la aceleración sísmica al que es sometido. La aceleración sísmica está dada por las características del terremoto y el estado de daño es determinado aleatoriamente según las probabilidades dadas por su Curva de Fragilidad.
3. Conocido el estado de daño de los transformadores, estos son colocados en una lista de prioridad que depende de su consumo esperado de energía en un horizonte de tiempo dado, su estado de falla y su tiempo de reparación.
4. El conjunto total de transformadores se divide en zonas igual al número de centro de operaciones de cuadrilla. Cada zona es atendida por su correspondiente centro y cada centro posee el mismo número de cuadrillas de restauración. Cada cuadrilla atiende una sección de la zona por medio de algoritmos de clusterización.
5. El ruteo de una cuadrilla hacia los transformadores de su cluster asignado está determinado primero por su prioridad y luego por la ruta óptima entre el centro de operaciones y un conjunto de transformadores de igual prioridad.
6. La primera cuadrilla que termina de atender su cluster se dedica a ayudar el cluster que tarde más en ser reparado, dividiéndose el tiempo de reparación en partes iguales.

7. Cuando finaliza la simulación, el programa entrega los transformadores atendidos por tiempo, falla, capacidad y otros, los cuales son procesados para entregar la energía no suministrada total, su evolución en el tiempo, distancias recorridas, etc.
8. Debido a la naturaleza aleatoria del estado de daño, se realizan 1000 simulaciones de Montecarlo para obtener una estimación de los valores esperados que puedan ser comparados.

El modelo propuesto contempla la recolección del estado de daño de los transformadores, la simulación de la estrategia propuesta y el procesamiento de sus resultados, mientras que la estrategia propuesta se describe entre los puntos (3) y (6). Los códigos correspondientes a todos los bloques se encuentran en el Anexo D, en lenguaje Python.

A continuación, se describirán cada uno de los bloques en detalle junto con las decisiones de diseño a medida que corresponda.

## 5.2 Pre-Procesamiento de datos

En esta etapa se procesan todos los datos relevantes para la simulación de un evento sísmico y la modelación del comportamiento de las cuadrillas. En la Figura 5.2, el bloque se divide en dos sub-bloques, los **Cálculos** y las **Asignaciones**. Los **Cálculos** son procesamientos matemáticos que hacen uso de uno o más elementos y que entregan nuevos elementos relevantes para el Modelo. Por ejemplo, una relación entre dos transformadores y la red vial de Santiago sería el camino más corto entre ellos. Por su parte, las **Asignaciones** corresponden a asignar a un determinado elemento otro elemento de interés, tal que compartan un mismo identificador. Por ejemplo, un transformador podría tener asociado una ID (Número entero único), su capacidad, sus coordenadas (x, y), etc.

El Pre-procesamiento de datos fue hecho principalmente en QGIS pues gran parte de la información relevante de referencia se encontraban en formato de *shapefiles* (.shp). Además, se escribieron códigos en Python que permitieron automatizar procesos que de otra forma hubiesen requerido tiempos extremadamente largos en realizar, los cuales se disponen en el Anexo D.

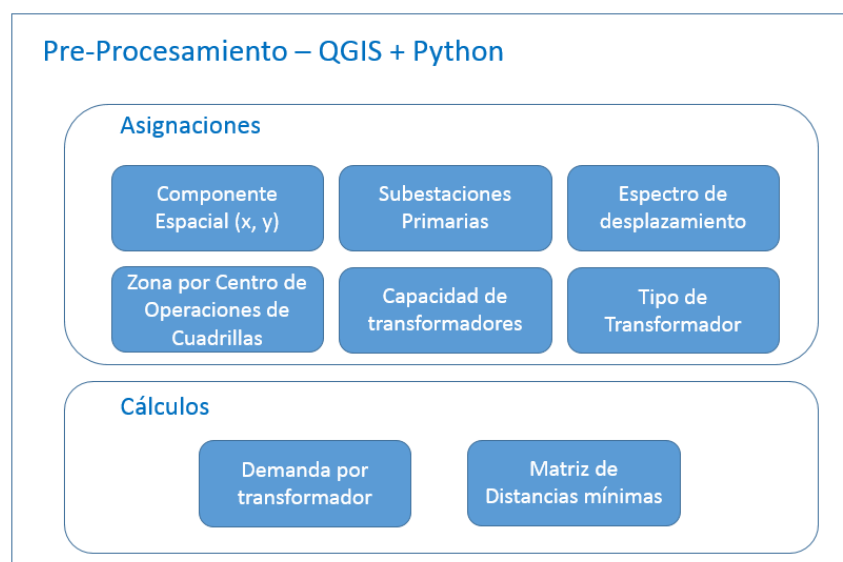


Figura 5.2 Partes del bloque de Pre-Procesamiento

## 5.2.1 Asignaciones

Los datos encontrados en formato *.shp* son las componentes (x,y) de los transformadores referenciales de media tensión, subestaciones primarias de la misma, la capacidad de los transformadores, la red vial de Santiago y el desplazamiento sísmico en forma de PGA y SA de 0.3s, 1.0s y 3.0s. La Figura 5.3 muestra una imagen de las comunas de Santiago que pertenecen al área de concesión de ENEL Distribución con los datos mencionados anteriormente a excepción del desplazamiento sísmico, donde las comunas de Santiago son representados por polígonos de color verde, la red vial de Santiago por trazos de color azul, los transformadores referenciales de MT por puntos de color morado, las subestaciones primarias de distribución por puntos rojos y los centros de operaciones de cuadrillas por puntos cuadrados de color cian.



Figura 5.3 Elementos que componen la red de distribución y vial en Santiago

La **Asignación** de valores se hace por medio de la tabla de **Atributos** que cada elemento con formato *.shp* posee. La tabla de Atributos contiene una ID, que corresponde a un número único para cada elemento que sirve como identificador, y otros atributos asociados a dicho elemento. En un principio, los transformadores de MT solo poseían sus coordenadas en UTM19 (Sistema de coordenadas universal transversal de Mercator), mientras que sus capacidades se encontraban en planillas ajenas al archivo *.shp*. En los casos donde se tiene acceso a las IDs, la asignación es simplemente por correspondencia de IDs. La correspondencia es posible gracias a las herramientas que posee el software QGIS.

	ID	Coord_X	Coord_Y
1	41	347384.24600	6301639.55000
2	97	345448.85000	6302370.46800
3	222	327728.99800	6313367.65900
4	356	355209.50100	6302170.15300
5	393	341227.80500	6302670.05300

Figura 5.4 Tabla de atributos para coordenadas UTM19 (x, y)

Cuando no se cuentan con las IDs, se deben hacer correspondencias por área. En el caso de las zonas de restauración asignadas a cada centro de operaciones, la asignación se realiza por

medio de polígonos de Voronoi (detallado en la Sección 4.3.4.1). Esta decisión de diseño asegura que cada transformador correspondiente a una zona generada por su centro de operaciones se encuentra más cerca de él que a cualquier otro centro. La Figura 5.5 (a) muestra la división de los transformadores, los que serán nombrados por *Zona i*, con *i* la numeración de la zona correspondiente.

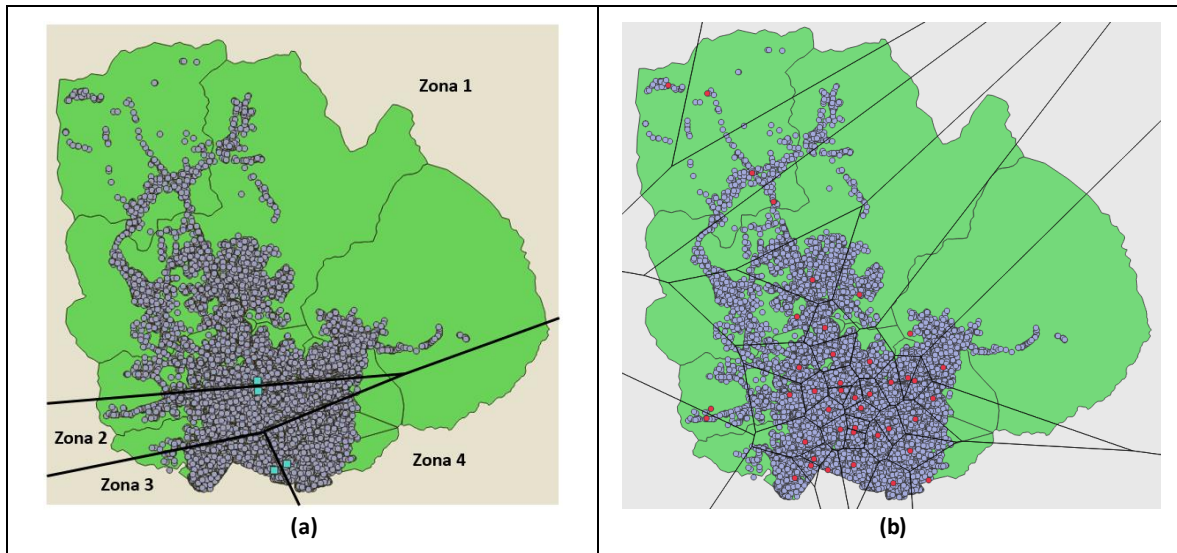


Figura 5.5 Asignación por área: Asignación de centros de operaciones (a) y subestaciones primarias (b)

Un procedimiento similar se hace con las subestaciones primarias de distribución. Como no se conoce la correspondencia Transformador – Subestación Primaria en el sistema de referencia utilizado, la asignación aproximada se hace de acuerdo a la cercanía del transformador a las subestaciones por medio de polígonos de Voronoi. La razón de esta división se hace para asignar en términos aproximados la demanda correspondiente a cada transformador de acuerdo a la demanda real de su subestación primaria, algo que se discutirá en las secciones siguientes.

Para el caso de la aceleración sísmica, los datos fueron obtenidos desde el Shakemap de la USGS<sup>4</sup> [50], escogiendo el terremoto del 27 de febrero del 2010. De los archivos se extrajeron curvas de contorno de PGA, y SA 3.0s, 1.0s y 0.5s. Finalmente se decidió utilizar como input de aceleración sísmica la Aceleración Espectral de 1.0 segundos por las siguientes razones:

- De acuerdo a la Federal Emergency Management Agency [26], es posible utilizar una de las 4 medidas de aceleración sísmica como input en las curvas de fragilidad.
- Estudios muestran que los postes de distribución de hormigón poseen frecuencias naturales de entre 1 Hz a 8 Hz [51], equivalente a periodos de 1 a 0,125 segundos con *peaks* cercanos al segundo. Otros estudios señalan que postes de acero tubular presentan periodos de entre 1.0 segundos a 3.0 segundos [52].

<sup>4</sup> United State Geological Survey

- Un estudio similar [53] concluyó que una aceleración espectral de 3.0s modelaba de mejor forma el comportamiento de su sistema de distribución real ante una perturbación sísmica, arguyendo que los postes y líneas de distribución son estructuras de alto periodo cuyo nivel de daño en un sismo está asociado principalmente al movimiento entre postes y consumos que hacen que los cables se “suelten” cuando no existe flexibilidad suficiente. En este caso, pruebas preliminares mostraron que un SA de 1.0 segundos modela de mejor forma el estado de daño de los transformadores.

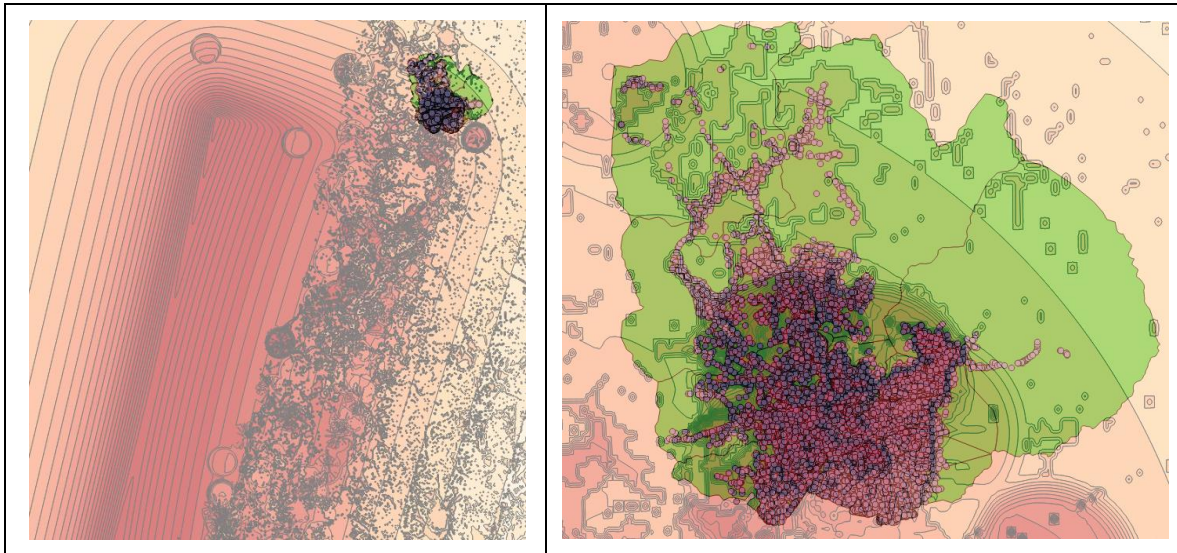


Figura 5.6 Curvas de contorno (SA-1.0s) para el terremoto del 27 de febrero del 2010

Con las áreas de contorno cargadas en QGIS, la asignación del movimiento sísmico a cada punto o transformador referencial se hace según el polígono en el que se encuentren inmersos. Finalmente, estos puntos son guardados en archivos de texto plano, preservando su orden. Esto último es de suma importancia pues indica que una fila del archivo que guarda la aceleración sísmica corresponde al mismo elemento para diferentes archivos con la misma fila, lo cual permite prescindir de identificadores en el Modelo de Ruteo.

ID	Coord_X	Coord_Y	psa10	psa30	psa03	PARAMVALUE	TrafoCap_C	Trafos_BIN	Trafos_Z_I	
1	41	347384.24600	6301639.55000	0.72000	0.12000	0.80000	0.36000	500	1	17
2	97	345448.85000	6302370.46800	0.92000	0.16000	0.88000	0.40000	1000	1	14
3	222	327728.99800	6313367.65900	0.56000	0.08000	0.56000	0.28000	10	1	41
4	356	355209.50100	6302170.15300	0.60000	0.12000	0.76000	0.32000	150	1	12
5	393	341227.80500	6302670.05300	0.84000	0.16000	0.80000	0.40000	150	1	14
6	414	334257.36600	6304153.40100	1.04000	0.16000	0.84000	0.44000	45	1	9
7	420	336248.19400	6304575.28300	1.08000	0.20000	0.88000	0.44000	30	1	9
8	425	349192.55800	6302799.51900	0.64000	0.12000	0.72000	0.36000	150	1	13

Figura 5.7 Asignaciones finales a los transformadores en QGIS

## 5.2.2 Demanda por transformador

La demanda aproximada asociada a cada transformador se determina de la siguiente forma:

1. Se adquiere la demanda real horaria de energía por subestación primaria de distribución para todo el año 2017. La información fue recabada del reporte de medidas históricas del Coordinador Eléctrico Nacional [54].

2. Se determina la capacidad máxima de los transformadores referenciales de distribución para cada zona asignada a una subestación de distribución. Las zonas de las subestaciones fueron determinadas por polígonos de Voronoi, procedimiento explicado en la sección anterior, ilustrada en la Figura 5.5.
3. Se verifica que, por cada zona definida por las subestaciones primarias, la demanda máxima del año no sea mayor al 70% de su capacidad en transformadores. En caso de que no se cumpla esta condición, se multiplica la curva de demanda por un factor menor a uno con el fin de que se cumpla este criterio. Esto, dado que es práctica habitual de las compañías de distribución tener una holgura relevante en las subestaciones primarias para prestar respaldo en caso de contingencia a subestaciones vecinas, ya sea a través de las redes de distribución o de conexiones entre subestaciones primarias.
4. Se calculan los porcentajes de asignación de demanda de la subestación en función de la capacidad del transformador y la capacidad total del área asignada, es decir

$$PD_i = C_i / \sum_{k \in SE_j} C_k \quad 5.1$$

Con  $PD_i$  el porcentaje de demanda del transformador  $i$  que pertenece al área de la subestación  $SE_j$  con capacidad  $C_i$ .

5. Los porcentajes se guardan en archivos siguiendo el orden en que se guardaron las asignaciones del capítulo anterior.

A modo de resumen se listan las simplificaciones hechas hasta el momento

- Se asigna la demanda leída por subestación primaria a sus transformadores más cercanos.
- La demanda por potencia se asume igual a la demanda de energía en una determinada hora.
- La demanda de una subestación primaria (AT/MT) se distribuye a prorrata de las capacidades de los transformadores de distribución (MT/BT). Los transformadores nunca superan el 70% de su capacidad máxima.

### 5.2.3 Matriz de distancias mínimas

El grafo que representa a la red vial consta de más de 700.000 nodos y 100.000 aristas sin considerar los transformadores de distribución, aunque mucho de estos nodos son duplicados para unir vectores (calles) en QGIS. Debido a este tipo de deficiencias, resolver un problema de distancias mínimas o asignación de trabajos en un grafo de esta magnitud en tiempo real resulta computacionalmente infactible, quedando descartada esta idea.

Para asegurar que el algoritmo se acerque a un procedimiento de ruteo óptimo de cuadrillas, primero se requiere que el camino que tome entre dos puntos sea el de menor distancia. Debido a que se está utilizando la red vial de Santiago, existen múltiples caminos que van desde un transformador a otro, o desde un centro de mantenimiento a un transformador. El cálculo del camino óptimo entre dos puntos se realiza por medio del algoritmo de Dijkstra, el cual se encuentra incorporado en QGIS.

El resultado final de este proceso es un grafo completo  $G = (N, E)$  con  $N$  el conjunto de transformadores y centros de operaciones, y  $E$  las aristas que representan la distancia entre los nodos  $i, j$  al tomar el camino más corto entre ellos. El grafo no varía por simulación ya que los centros y número de transformadores son fijos, razón que permite guardar el resultado en memoria física para su posterior uso.

La estructura de datos del grafo corresponde a la de una matriz triangular superior, donde el elemento  $c_{ij}$  representa la distancia en metros que hay entre el nodo  $i$  y el nodo  $j$ . La matriz omite las diagonales debido a que la distancia de un punto consigo mismo es cero. Ahora bien, el procedimiento de almacenamiento de esta estructura de dato se realizó por medio de archivos de texto plano “ci” donde  $i$  corresponde al nodo de origen, el cual contiene todas las distancias hacia nodos  $j$  mayores a  $i$ . Esta estructura de datos se escogió por el siguiente motivo

- **Es de alta velocidad:** Solo requiere abrir un archivo que contiene una línea, transformar la línea en una lista y obtener la posición  $(i, j)$  deseada. El algoritmo tarda cerca de 1,5 minutos en recorrer todos los nodos (21.701) en un procesador de 3.1 GHz.
- **Liviano en memoria RAM:** Solo se requiere una función que abra un archivo, obtenga un valor y lo cierre, operación que presenta un consumo de memoria RAM despreciable. El peso de la matriz es de 1.9 GB, por lo que almacenar toda esa memoria en una lista de dos dimensiones resulta costoso, además de que la búsqueda en listas tarda  $O(n)$ . Implementar otra estructura de datos de alta rapidez como un look-up por diccionarios requeriría utilizar hasta 9.5 GB en memoria RAM para un único caso, lo que impediría realizar simulaciones en paralelo.

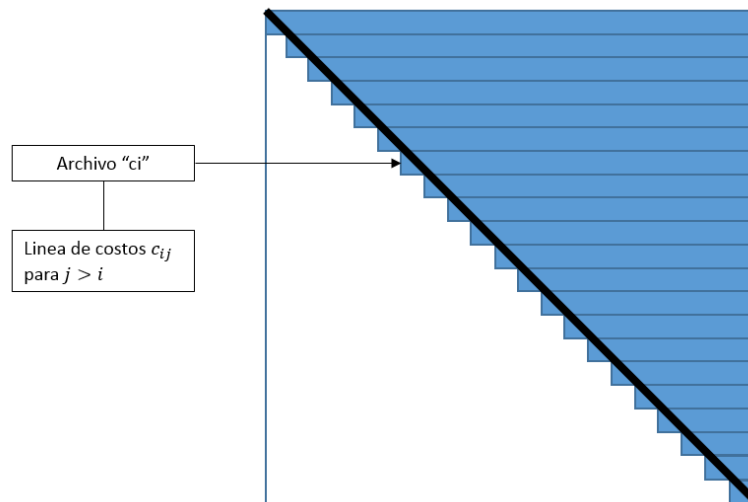


Figura 5.8 Estructura de los archivos de distancias entre nodos.

Un aspecto a mencionar en el diseño de la red es que el grafo resulta ser no-dirigido, es decir,  $c_{ij} = c_{ji}$  o en términos verbales, que el camino más corto es independiente de donde se inicie la ruta. Si bien esto no es cierto en la vida real, se optó por esta simplificación ya que no fue posible acceder a las direcciones de las calles en los archivos *.shp* de la red vial de Santiago.

Otro punto no menor es el orden en que se guardaron los recorridos  $c_{ij}$  tal que la fila  $c_{i,j}$  coincidiera con el orden de las asignaciones del capítulo anterior. Con ello, utilizar una ID resulta innecesario ya que existe una correspondencia 1:1 entre el índice de la fila y el índice de todas las demás asignaciones estudiadas. El hecho de que no exista una ID permite que el Modelo no requiera de un algoritmo de búsqueda, lo que aumenta su rapidez.

### 5.3 Modelo de Simulación y Ruteo

En el Modelo de Simulación y Ruteo tiene como objetivo simular el comportamiento de la red ante el evento BPAI cargado, así como simular el comportamiento de las cuadrillas según la estrategia implementada dentro del mismo. Este se divide en 7 etapas, los que serán explicados a continuación.

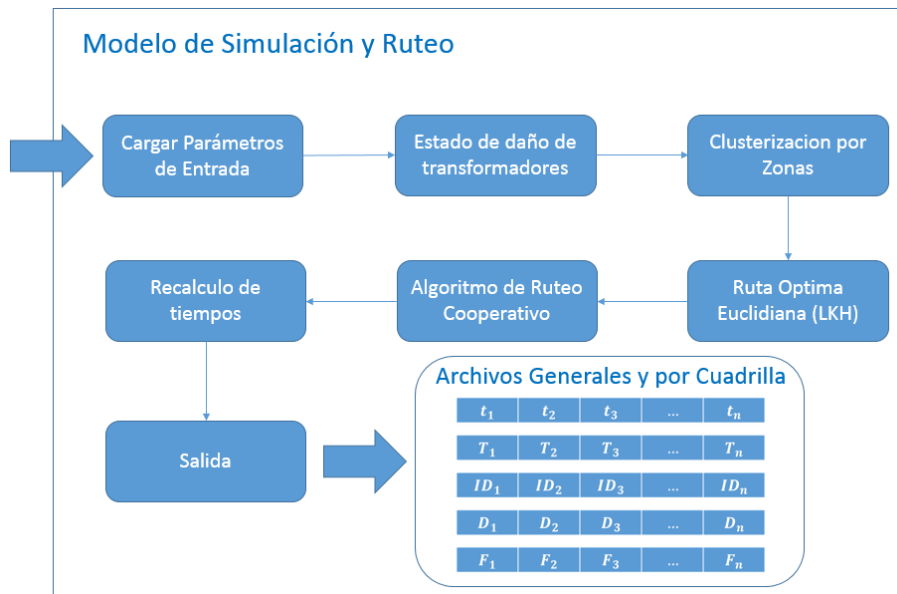


Figura 5.9 Esquema general del modelo de simulación y ruteo de cuadrillas

#### 5.3.1 Carga de Parámetros de Entrada

Los parámetros a cargar en el modelo son todas las asignaciones realizadas en una estructura de dato denominada *Lista*. Una lista cualitativamente es similar a un vector, es decir, almacena datos en una única dimensión en la forma  $v_i$  donde  $i$  es la posición  $i$ -ésima de la lista. Sin embargo, no se debe confundir un vector con una lista, pues si bien cualitativamente son similares, un vector corresponde a una estructura de datos fija que almacena elementos de un mismo tipo de datos, mientras que una lista puede crecer o disminuir su tamaño dinámicamente, junto con almacenar diferentes tipos de datos. Los elementos cargados en listas son

- Coordenada  $(x, y)$  de los nodos del grafo
- Zonas a las que pertenecen los nodos
- Subestación a la que pertenecen los transformadores
- Porcentaje de demanda asignada a cada transformador
- Capacidad de los transformadores
- Aceleración sísmica de cada transformador



Gracias al procedimiento descrito en la sección de Pre-Procesamiento de datos, se puede asegurar lo siguiente: El transformador con coordenadas  $Coord_i$  posee sus atributos guardados en cualquier posición de una lista genérica  $L_i$ .

Otros parámetros cargados al modelo y que no corresponden a elementos procesados son los siguientes:

- Nombre del archivo de salida
- Fecha del evento BPAI con formato de mes, día y hora: **La fecha corresponde al 27 de febrero a las 3:00 AM.**
- Número de simulaciones: **Se realizaron 1000 simulaciones por cada caso.**
- Número de clusters totales: El número de clusters, los cuales serán atendidos por cuadrillas de diferentes centros de operaciones. Se requiere que el número de cluster totales sea un múltiplo del número de centro de operaciones de cuadrillas.
- Parámetros de las curvas de fragilidad: Puede ser con componente estándar o con reforzamiento sísmico.
- Horas de trabajo y horas de descanso: Una cuadrilla no puede trabajar sin parar. **Las horas de trabajo y descanso son de 12 horas en ambos casos**, teniendo en cuenta el estado de emergencia en el que se encuentra la red.
- Horizonte de evaluación de prioridades: El horizonte corresponde a un lapso de tiempo medido en horas que pretende modelar la energía total que podría suministrar cada transformador desde el inicio del evento BPAI hasta el fin del horizonte de tiempo si es que no hubiera falla alguna en el sistema. Este parámetro permite asignarle prioridades de atención a cada transformador con un estado de falla determinado, procedimiento que se explicará en las siguientes secciones. **El horizonte utilizado es de 168 horas, es decir, una semana.**
- Velocidad de las cuadrillas: Ingresadas en kilómetros por hora. **La velocidad utilizada es de 40 km/h**, que es aproximadamente la velocidad promedio de las cuadrillas de distribución en Chile de acuerdo al último estudio VAD [55].

### 5.3.2 Cálculo del Estado de daño de Transformadores

Fase aleatoria del Modelo y la que impone la necesidad de realizar múltiples simulaciones para llegar a parámetros esperados. El procedimiento consiste en explotar la definición de Curvas de Fragilidad dado un desplazamiento, sabiendo que existe cierta probabilidad de que un transformador se encuentre en un estado de daño determinado. Se modela con  $x = U(0,1)$ , con  $U(0,1)$  la distribución uniforme entre 0 y 1, que definirá el estado de daño final del transformador en cada simulación.

A modo de ejemplo, supongamos que se está calculando el estado de daño del transformador  $i$ , en primer lugar se debe calcular la función de probabilidad acumulada de dicho transformador para diferentes estados de daño, conociendo su curva de fragilidad (Reforzamiento Estándar o Sísmico) y la aceleración espectral de 1.0s a la que está sometida en galios. En la Figura 5.10, se ve que con una aceleración espectral determinada, el transformador  $i$  tiene una probabilidad del 80% de tener un estado de daño mayor al 4%, una probabilidad del 30% de tener un estado de daño mayor al 12% y una probabilidad del 0% de tener un estado de daño mayor al 50% y 80%. Como el estado de daño mínimo es del 4%,

entonces el transformador tiene un 20% de probabilidades de no sufrir daño alguno ante esta aceleración espectral. Por último, el estado de daño final del transformador se escoge de un valor obtenido de una distribución uniforme entre 0 y 1. Claramente, el estado de daño que tenga más probabilidad y, por lo tanto, abarque un intervalo mayor en [0,1] tendrá más probabilidades de ser escogido por cada simulación.

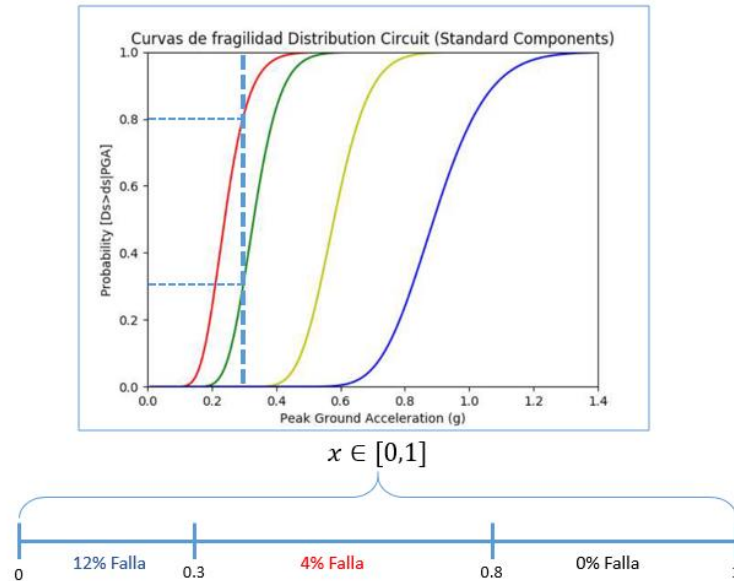


Figura 5.10 Posibles estados de daños para un transformador sometido a un SA determinado

Con respecto a los tiempos de reparación por estado de daño, de acuerdo al último estudio VAD entregado por la CNE [55], las tareas de reparación realizadas por una cuadrilla tardan entre 15 minutos a 2 horas con 20 minutos, aunque este último es el peor caso de una reparación de empalme y acometida, donde el promedio corresponde a una hora y 40 minutos. Estos tiempos solo contemplan la tarea de reparación en sí misma, es decir, no considera los tiempos de viaje, detección de falla, etc. A modo de simplificación, se escogieron los rangos de restauración de falla entre estos valores en intervalos crecientes. Los tiempos de reparación dependen del tipo de falla que sufra un elemento, los que se describen en la Tabla 5.1.

Tabla 5.1 Tiempos de restauración por estado de daño

Estado de daño	Tiempo de restauración [horas]
4% del circuito	0.25
12% del circuito	0.5
50% del circuito	1.0
80% del circuito	2.0

Finalmente, conocido los estados de daño y los tiempos de restauración de cada transformador, se procede a calcular la prioridad del transformador. La prioridad corresponde a una medida de levantamiento de carga no suministrada por unidad de tiempo en un **horizonte** de tiempo dado. La fórmula utilizada es la siguiente:

$$PriorTrafo_i = \left[ \left( \sum_{t \geq T_{Ev}}^{T_{Ev} + T_h} E_{i,t} \right) \cdot \frac{Falla_i}{Rest_i} \right] \quad 5.2$$

Donde  $T_{Ev}$  es la fecha de ocurrencia del evento BPAI,  $T_h$  es el horizonte ingresado al modelo,  $E_{i,t}$  es la energía suministrada horaria esperada del transformador  $i$  en el tiempo  $t$ ,  $Falla_i$  es el estado de falla del transformador en porcentaje y  $Rest_i$  es el tiempo de restauración del transformador. El objetivo de esta fórmula es dar prioridad a las fallas que además de privar de una gran porción de energía al sistema, también puedan ser reparadas en un tiempo razonable. La función *techo* se aplica al resultado ya que para los transformadores de baja capacidad las diferencias de prioridad varían en decimales. El hecho de que la variación sea tan pequeña, hace preferir un ruteo óptimo entre estos transformadores.

A modo de ejemplo, se presenta dos consumos ficticios en la Figura 5.11, los que se utilizan de para calcular las diferentes prioridades que tendrían transformadores dependiendo de su falla en un horizonte de evaluación de 3 horas, tabulados en la Tabla 5.2.

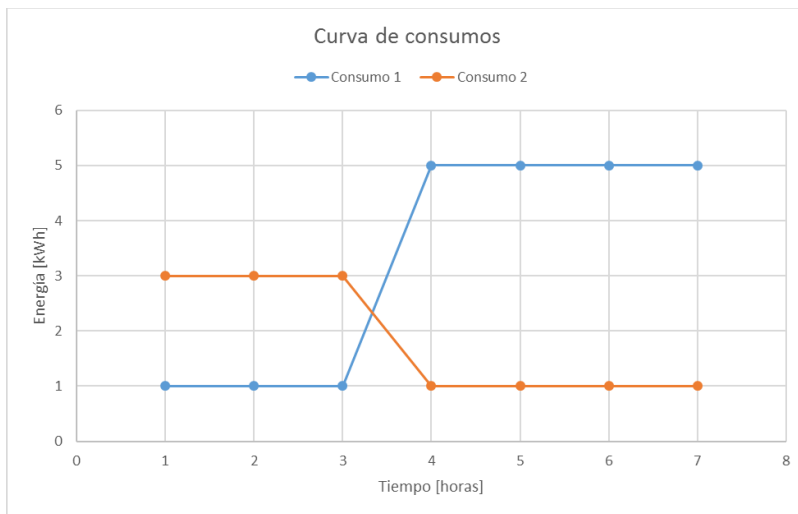


Figura 5.11 Curva de consumos de ejemplo

Tabla 5.2 Prioridades para diferentes tipos de fallas y consumos con un horizonte  $T_{ev} = 3$  horas

Falla [%]	Tiempo de Reparación [horas]	Prioridad (Consumo 1) $E_{T_{ev}} = 3$ [kWh]	Prioridad (Consumo 2) $E_{T_{ev}} = 9$ [kWh]
4	0,25	0,48	1,44
12	0,50	0,72	2,16
50	1,00	1,50	4,50

80	2,00	1,20	3,60
----	------	------	------

Según esta metodología, para un mismo consumo la prioridad sigue la secuencia decreciente de falla de un 50%, 80%, 12% y 4%. Esto quiere decir que levanto más energía por unidad de tiempo al reparar una falla del 50% en 1 hora que al levantar un 4% de la misma carga en solo 15 minutos. Otro aspecto de esta metodología es que prioriza la pérdida de consumos más elevados al comparar las prioridades del Consumo 1 con los del Consumo 2.

Tabla 5.3 Prioridades para diferentes tipos de fallas y consumos con un horizonte  $T_{ev} = 6$  horas

Falla [%]	Tiempo de Reparación [horas]	Prioridad (Consumo 1) $E_{T_{ev}} = 18 [kWh]$	Prioridad (Consumo 2) $E_{T_{ev}} = 12 [kWh]$
4	0,25	2,88	1,92
12	0,50	4,32	2,88
50	1,00	9,00	6,00
80	2,00	7,20	4,8

Por otro lado, de la Tabla 5.3 se aprecia como la elección del horizonte puede afectar la priorización de transformadores y, por lo tanto, del ruteo de cuadrillas.

### 5.3.3 Clusterización por Zonas asignadas a centros de operaciones

Luego del paso anterior, se procede a hacer una clusterización de cada zona en el número de cuadrillas que contenga cada centro de operaciones. El Modelo asume que cada centro de operaciones tiene el mismo número de cuadrillas, más precisamente, cada uno de los 4 centros de operaciones cuenta con 25 cuadrillas inicialmente.

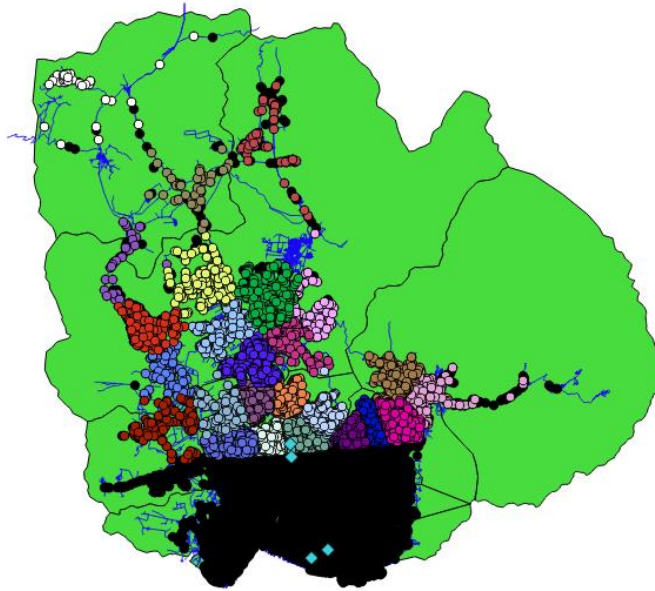


Figura 5.12 Clusterización de la Zona 1

La Figura 5.12 muestra una clusterización de los transformadores de la Zona 1, donde los clusters corresponden a todos los puntos coloreados a excepción de los puntos negros, asignados a transformadores que no sufrieron daños ni son parte de la Zona 1. Los rombos cian identifican a los centros de operaciones en cada zona.

La idea de clusterizar zonas radica en la utilización eficiente de las cuadrillas de restauración. Sabiendo que cada cluster posee transformadores clasificados de acuerdo a su cercanía espacial, se disponen cada una de las cuadrillas disponibles en cada cluster con la finalidad de reparar los transformadores que la conforman en el menor tiempo posible. Bajo este esquema aseguramos que todas las cuadrillas estén siendo despachadas durante la emergencia y estén reparando transformadores cercanos unos a otros.

### 5.3.4 Algoritmo de Ruteo Euclidiano

Esta fase se aplica a cada cluster del sistema y se divide en varias etapas. Para cada cluster con un centro de operaciones correspondiente a su zona:

1. Se determinan las prioridades de los transformadores en el cluster de acuerdo al estado de falla de cada transformador, y se ordenan de mayor a menor. Luego se crea una lista auxiliar, llamada lista de prioridades, que tiene la finalidad de almacenar todas las diferentes prioridades en el cluster. Esta lista mantiene el orden de mayor a menor y elimina prioridades repetidas (no así las prioridades intrínsecas de cada transformador).
2. Se inicializa una lista vacía de transformadores prioritarios
3. Se recorre la lista de prioridades, para cada prioridad:
  - a. Si solo un transformador posee esa prioridad, se colocan directamente en la lista de transformadores prioritarios. Esto pues el tour de dos nodos es único.
  - b. Si más de dos transformadores tienen esa prioridad, se procede a resolver el problema CCVRP (LKH-3) para todos los transformadores con la misma

prioridad. La demanda de los transformadores para el problema CCVRP corresponde a sus tiempos de restauración en minutos, la capacidad del vehículo es el tiempo total que tomaría restaurar el conjunto de transformadores y el depósito se coloca en el último transformador atendido en la iteración anterior. El ejecutable LKH-3 entrega un archivo de texto plano con el orden de ruta de los transformadores de acuerdo a la convención TSPLIP [56], el cual se ingresa de igual forma a la lista de transformadores prioritarios

4. Se calculan los costos de viaje haciendo uso de la matriz de distancias mínimas desde el centro de operaciones siguiendo la lista de transformadores prioritarios considerando que:
  - a. Una cuadrilla puede estar activa tanto tiempo como su máxima cantidad de horas de trabajo lo permita. Esto considera tiempos de viaje y tiempos de restauración.
  - b. Excedida las horas de trabajo, la cuadrilla vuelve a su centro de operaciones y descansa las horas de descanso ingresadas al Modelo.
  - c. La cuadrilla vuelve al trabajo iniciando su ruta desde el centro de operaciones hasta el siguiente transformador en la lista de prioridades.
5. El cálculo del punto (4) guarda en listas separadas lo siguiente:
  - a. Tiempo creciente entre el viaje del nodo origen hasta la reparación del nodo destino.
  - b. Distancia recorrida entre el nodo destino y el nodo origen.
  - c. Estado de falla del nodo destino.
  - d. Capacidad del nodo destino.
  - e. Índice del nodo destino. El índice corresponde al índice de las listas de atributos del transformador determinado, las cuales coinciden para todas las listas.

El almacenamiento de la falla, capacidad e índice de cada transformador permiten recrear la simulación en caso de que se requiera. El resto de las variables se obtienen para calcular los parámetros de rendimiento deseados.

Un punto a destacar es el paso (3.b), donde se indica que la capacidad del vehículo corresponde a la suma de todos los tiempos que llevaría reparar el conjunto de transformadores. Esta decisión de diseño se llevó a cabo debido a resultados de pruebas preliminares donde efectivamente se ajustaba la capacidad del vehículo a sus horas de descanso y el depósito en el centro de operaciones. El algoritmo LKH-3 en este caso crea  $K = \lceil T_R/T_D \rceil$  vehículos ficticios con  $T_R$  el tiempo total de reparación y  $T_D$  las horas de descanso, lo que es correcto en el sentido de que cada vehículo modela una jornada de trabajo o sub-circuito del único vehículo atendiendo el cluster. El problema recae en la formulación del problema CCVRP, donde se minimiza la suma total de tiempos de espera de cada nodo en el sistema, lo que a su vez implica la minimización de la suma de los tiempos de espera de cada nodo en cada sub-circuito atendido por un vehículo ficticio, es decir, se intentara minimizar los tiempos de espera de los nodos en cada sub-circuito asignado al vehículo real

como si estuvieran siendo resueltos en paralelo, lo que provoca que al inicio de cada sub-circuito se atiendan la mayor cantidad de nodos en el menor tiempo posible. Como el inicio de cada sub-circuito implica una espera igual al tiempo de descanso del vehículo real, esta metodología no entrega resultados satisfactorios.

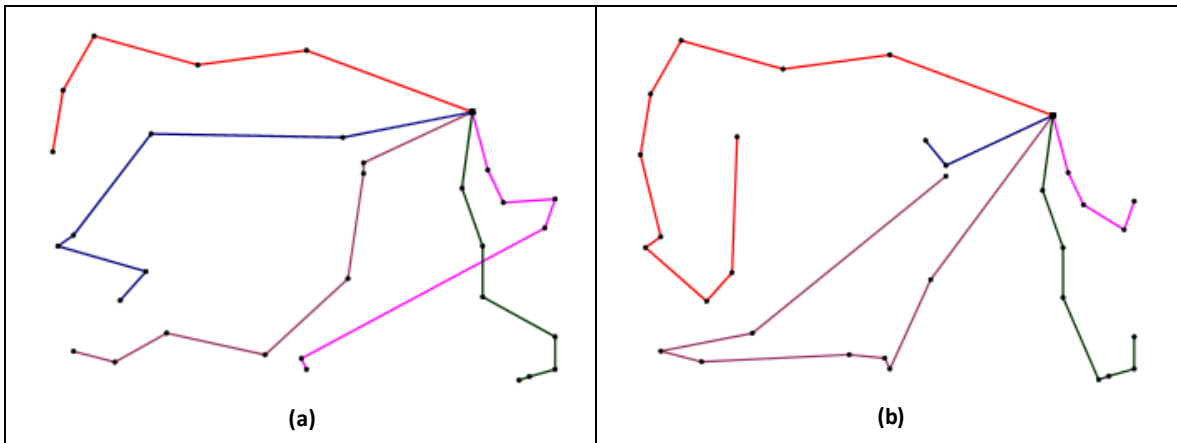


Figura 5.13 Resolución del problema CCVRP (a) y CVRP (b) para un mismo set de nodos [44]

Un ejemplo de cómo actúa el algoritmo CCVRP se muestra en la Figura 5.13 (a), donde cada vehículo atiende una cantidad no menor de nodos en los primeros pasos de cada sub-circuito. Recordar que el CCVRP minimiza la suma de las distancias recorridas por vehículos en cada nodo, que no es equivalente a minimizar la distancia total recorrida por cada vehículo (Problema CVRP).

Para lograr que el vehículo atiende la mayor cantidad de transformadores en sus primeras jornadas, se ajustó la capacidad a la descrita en el punto (3.b). Además, como el vehículo se devuelve al depósito únicamente al atender el último transformador en la iteración, el depósito<sup>5</sup> es colocado en el último transformador atendido en la iteración anterior de la lista de prioridades (si es la primera iteración, entonces el depósito es el centro de operaciones) para que se escojan los transformadores más cercanos a la posición actual del vehículo.

La metodología del párrafo anterior no presenta problemas debido a que el archivo de salida al seguir la norma TSPLIB, entrega la lista secuencial de nodos atendidos no repetidos con el depósito en la primera posición de la lista. La inserción del centro de operaciones considerando tiempos de trabajo y descanso se realiza en el paso (4).

<sup>5</sup> No confundir “depósito” con el centro de operaciones. El depósito es el nodo inicial de un problema CCVRP que varía de problema en problema, mientras que el centro de operaciones es la localización real donde parten y descansan todas las cuadrillas. Notar que solo se habla de depósito cuando se menciona el problema CCVRP.

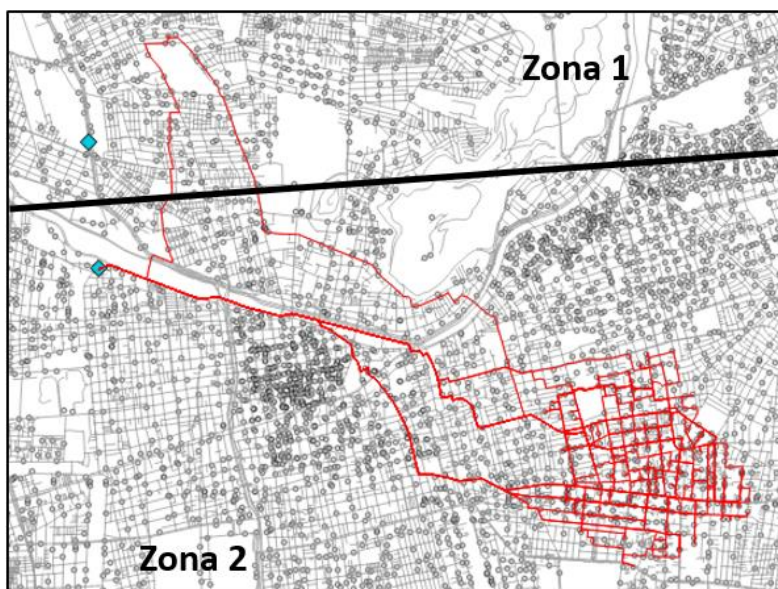


Figura 5.14 Ruta de atención por una cuadrilla de la Zona 2

La Figura 5.14 muestra las rutas recorridas por una de las cuadrillas de la Zona 2 en rojo, donde se aprecia la atención al cluster asignado así como múltiples rutas que van hacia su centro de operaciones, indicando la necesidad de descanso de las cuadrillas. También se aprecia una desviación hacia otra zona del área de concesión (Zona 1) debido al algoritmo de Ruteo Cooperativo, explicado a continuación.

### 5.3.5 Algoritmo de Ruteo Cooperativo

Debido a que los clusters contienen diferente cantidad de transformadores dañados, que a su vez puede tener diferentes estados de daño, es inevitable que unas cuadrillas terminen de atenderlos antes que otras. Teniendo en cuenta que la distribuidora se encuentra en un estado de emergencia, no es aceptable que queden cuadrillas ociosas o al menos no en las primeras etapas de la restauración de servicio.

El algoritmo desarrollado para abordar la tarea de reasignación de nodos a atender recibe el nombre de Algoritmo Cooperativo. Supongamos que se calcularon los tiempos de reposición de cada cluster, entonces:

1. Se eligen los clusters de menor y mayor tiempo de reposición, representados por su lista de transformadores prioritarios  $c_i$  y  $c_j$  respectivamente.
2. Partiendo desde el transformador de menor prioridad del cluster  $c_j$ 
  - a. Si es un centro de operaciones, quitarlo de  $c_j$ .
  - b. Si es un transformador, entonces si el tiempo total de reposición acumulado de  $c_j$  es mayor al tiempo acumulado de  $c_i$  más el tiempo en viajar y reparar el transformador de  $c_j$ , se elimina el último elemento de  $c_j$  y se agrega al final de  $c_i$ .
  - c. Se actualizan todas las listas asociadas a los clusters en caso de que se cumpla la condición de (2.b) y se vuelve al paso (2.a).



Este proceso se hace tantas veces como se desee con el fin de asegurar una igualdad de tiempos de trabajo entre todas las cuadrillas de restauración. Pruebas preliminares aseguran que repetir este algoritmo  $2N_c$  veces con  $N_c$  el número total de cuadrillas disponibles en la red asegura una diferencia máxima entre cuadrillas no mayor a 15 horas. La Figura 5.15 muestra gráficamente como opera el algoritmo, donde el gradiente indica la prioridad de los transformadores atendidos en un tiempo determinado. El algoritmo básicamente iguala el tiempo de reposición de ambos clusters.

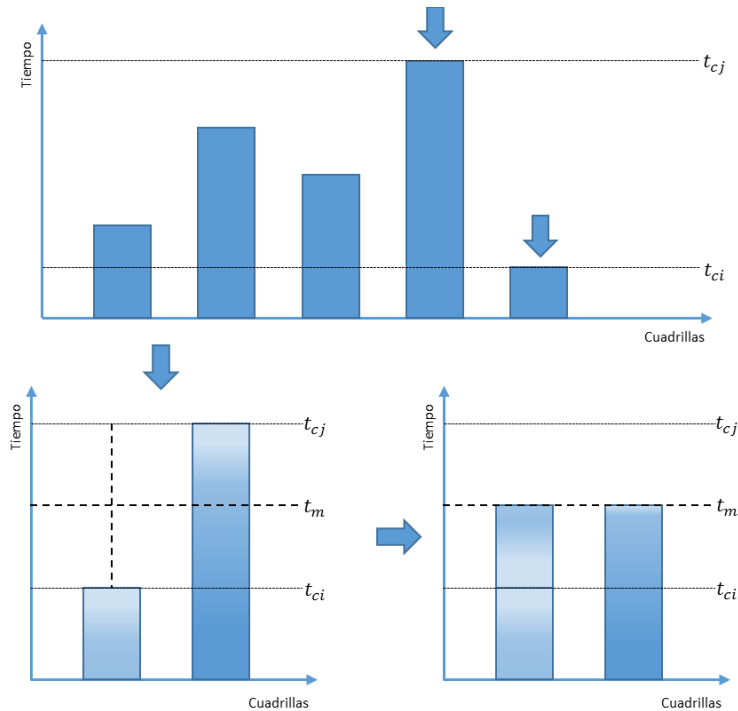


Figura 5.15 Ilustración gráfica del Algoritmo Cooperativo

Las siguientes decisiones de diseño se tomaron en la implementación del Algoritmo Cooperativo:

- **Ningún transformador de  $c_j$  ya atendido cuando  $c_i$  finalizó puede ser atendido por  $c_i$ .** Gráficamente significa que ningún transformador de  $c_j$  bajo la línea  $t_{ci}$  puede ser atendido por  $c_i$ , tal como se aprecia en la Figura 5.15. Esto apunta a que las cuadrillas deben finalizar sus clusters en primer lugar, y luego cooperar con el trabajo de otras cuadrillas sin importar la zona a la que pertenezca, asegurando que se estén aprovechando la distancia de los puntos que conforman los clusters. Si no se respetara esto, entonces hacer clusterización no tendría ningún objetivo y simplemente se haría un algoritmo de ruteo basado únicamente en transformadores prioritarios. Más aun, considerando que en esta etapa las zonas no tienen relevancia, no respetar esto implicaría restarle importancia a la zonificación de la red de transformadores.
- **El algoritmo es de tipo greedy en tiempo.** La elección de  $c_i$  y  $c_j$  como las cuadrillas que menos y más tardan en atender a sus clusters corresponde al mejor paso a seguir en su iteración, pero no necesariamente el paso hacia el mejor resultado final. En lo que respecta a la elección de cuadrillas, una implementación que recorrió todas las cuadrillas e igualaba sus tiempos con la cuadrilla de mayor diferencia temporal fue

implementado en modelos preliminares de este trabajo de memoria, dando como resultado un algoritmo lento y deficiente, además de que la cuadrilla elegida era  $c_i$  o  $c_j$  en la mayoría de los casos. Otro aspecto del algoritmo es que elige sus cuadrillas de acuerdo a los tiempos que tardan en atender a sus clusters, cuando podrían existir otras alternativas, por ejemplo, ser uno de menor tiempo  $c_i$  y otro que proporcione un mayor levantamiento de energía no suministrada. Un algoritmo de este estilo también se implementó en modelos preliminares de este trabajo de título, el cual resultó ser efectivo pero prohibitivamente lento para efectos de esta simulación de gran tamaño. En cualquier caso, teniendo en cuenta que la norma técnica de calidad de servicio para sistemas de distribución [57], en lo que se refiere a continuidad de servicio, considera como uno de los indicadores de calidad el tiempo de interrupción promedio (SAIDI), es razonable que la distribuidora desee restaurar el servicio en el menor tiempo posible.

- **Se comienza la colaboración por los transformadores menos prioritarios.** En primer lugar, el algoritmo tiene como objetivo hacer que ambas cuadrillas tarden la misma cantidad de tiempo en restaurar sus transformadores, lo que provoca que el intervalo de tiempo desde  $t_{cj}$  hasta  $t_{ci}$  se divida en  $t_m$ . Luego, solo tiene sentido elegir una de las dos partes a agregar en  $c_i$ . Si  $c_i$  fuera de una zona diferente a  $c_j$ , entonces este último tiene la ventaja de atender transformadores cercanos a su centro de operaciones, lo que implica que  $c_j$  debería atender la primera mitad de su exceso de tiempo de restauración, aprovechando tiempos mínimos de viaje, mientras que  $c_i$  debería atender la segunda mitad, considerando que ellos pertenecen a transformadores de baja prioridad y la penalización por viaje no afectaría mayormente la restauración del sistema. Con lo anterior, aseguramos que  $c_i$  debe partir por los transformadores de menor prioridad de  $c_j$ . Una desventaja de este método es que cuando  $c_i$  termina de atender a su cluster, comienza con los transformadores de menor prioridad en  $c_j$  cuando podría comenzar desde el inicio de  $t_m$ , lo cual se puede ver en el degradado azul de las barras finales en la Figura 5.15. Este problema se trata en la siguiente fase del algoritmo.

La Figura 5.16 muestra de mejor forma la cooperación de una cuadrilla de la Zona 4. La ruta tomada por la cuadrilla, delineada en rojo, primero se encarga de restaurar los transformadores en su cluster asignado, para luego ofrecer ayuda a una de las cuadrillas de la Zona 2. Se aprecia también que la cuadrilla siempre vuelve a su centro de operaciones independiente de la zona que se esté ayudando, lo que es coherente con el hecho de que los centros de operaciones no cuentan con capacidad infinita de vehículos.



Figura 5.16 Ruta de una cuadrilla de la Zona 4

### 5.3.6 Recalculo de tiempos y otros

Esta parte tiene como objetivo resolver las deficiencias del último punto listado en las decisiones de diseño del Algoritmo Cooperativo. Básicamente se invierte la sección agregada al cluster  $c_i$  con la finalidad de que este atienda a los transformadores inmediatamente después de  $t_m$ , es decir, se atienden los transformadores de mayor prioridad pertenecientes anteriormente a la cuadrilla  $c_j$ . Dado que las distancias recorridas (que se leen de la matriz de distancias) en la mayoría de las veces se repiten con el paso anterior, se utiliza una metodología Look-up Table para no tener acceder a la matriz de distancias nuevamente. Luego de realizar la inversión, se actualizan las listas de las demás variables asociados al cluster.

### 5.3.7 Archivos de salida

Esto corresponde a la parte final del Modelo de Simulación y Ruteo donde se guardan las variables relevantes del ruteo final de las cuadrillas. Las salidas se dividen en dos grupos, uno de resultados globales y otro de resultados por cuadrilla. Los resultados globales muestran la evolución de la restauración global del sistema, mientras que los archivos por cuadrilla muestran la evolución de la restauración de los transformadores de cada cuadrilla.

Los archivos a guardar en ambos casos son las listas de la etapa 5 del Algoritmo de Ruteo Euclidiano, modificadas por las dos etapas posteriores a ella. Los archivos corresponden archivos de texto plano separados por coma, con un número de filas igual al número de simulaciones.

## 5.4 Procesamiento de Resultados

En el procesamiento de resultados se trabajan los archivos de salida de las simulaciones para obtener parámetros que permitan dar cuentas del desempeño de la red en el proceso de restauración. Los archivos en la última etapa del Modelo de Simulación y Ruteo básicamente corresponden itinerarios de transformadores a atender globalmente y por cuadrilla, asociados su estado de daño, tiempo de reparación, distancias recorridas y tiempos acumulados. Del itinerario se extraen parámetros globales y por cuadrilla, que permitirán medir el desempeño del modelo y tomar decisiones en base a ellos. Los parámetros son

- Parámetros Globales por simulación:
  - Tiempo de restauración del 100%
  - Tiempo de restauración del 98%
  - Energía no suministrada total
  - Evolución de la energía no suministrada acumulada
  - Evolución de la energía no suministrada horaria
  - Costo de falla
  - Suma de distancias totales recorrida por cuadrillas
- Parámetros por Cuadrillas asignadas a un centro de operaciones:
  - Energía no suministrada total de los transformadores atendidos
  - Suma de distancias por cuadrillas
  - Suma de transformadores atendidos por zona

En los Parámetros Globales, los tiempos totales de restauración, de energía no suministrada horaria y total tienen directa relación con los parámetros  $T_U$  y  $T_D$  mencionados en el capítulo donde se trata el concepto de resiliencia (detallado en la Sección 3). La evolución de la energía no suministrada acumulada permite visualizar el comportamiento de la restauración del sistema y entre que rangos se mueven dichos parámetros, mientras que la evolución de energía no suministrada horaria muestra la profundidad de la falla en conjunto del sistema. Básicamente la evolución de energía no suministrada acumulada corresponde a la suma acumulada de la energía no suministrada horaria, lo cual se puede visualizar de la Figura X.

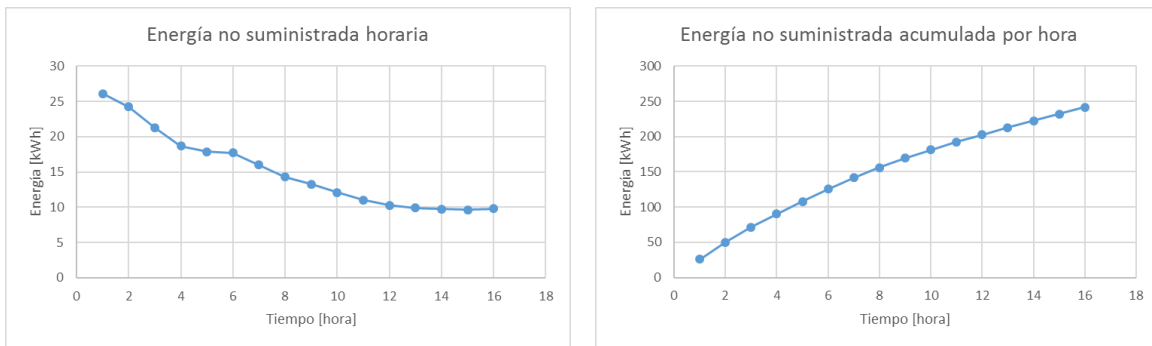


Figura 5.17 Ejemplo grafico de energía no suministrada horaria (izquierda) y energía no suministrada acumulada por hora (derecha)

El costo de falla tiene directa relación con los tres puntos anteriores, calculado utilizando los Costos de Falla de Largo plazo, de acuerdo a lo mencionado en [58], donde se cita que “Sin perjuicio de las sanciones que correspondan, la interrupción o suspensión del suministro de energía eléctrica no autorizada en conformidad a la ley y los reglamentos, que afecte parcial o íntegramente una o más áreas de concesión de distribución, dará lugar a una compensación a los usuarios sujetos a regulación de precios afectados, de cargo del concesionario, **equivalente al duplo del valor de la energía no suministrada durante la interrupción o suspensión del servicio, valorizada a costo de racionamiento**”. Esto implica que la valorización del costo social producido por una interrupción no autorizada en una o más áreas

de concesión se calcula como doble del Costo de Falla de Largo plazo. Si bien durante eventos de fuerza mayor no se compensan a los clientes, es de importancia tener un método de cuantificación de todos los costos asociados a la pérdida económica y social producidos por el corte. Los costos se calculan por profundidad para cada hora de acuerdo a la actualización de costos de falla de largo plazo de marzo del 2018 [59]. Las sumas de distancias totales permiten aproximar el tiempo utilizado en viajes con respecto al tiempo total de restauración, así como detectar deficiencias en comparación con otras estrategias de ruteo.

Los resultados por cuadrillas asignadas miden el desempeño individual de las cuadrillas de un centro de operaciones determinado. La energía no suministrada por transformadores atendidos contempla tanto transformadores de la zona que debieron atender las cuadrillas como los transformadores de otras zonas atendidas en cooperación con otras cuadrillas. La suma de distancias por cuadrilla muestra si es que las cuadrillas de una zona deben realizar viajes más largos y, por lo tanto, perder más tiempo en ida y vuelta entre transformadores y centros de operaciones. Por último, la suma de transformadores atendidos indica cuantos transformadores de una determinada zona fueron atendidos por las cuadrillas de otra zona. Este último parámetro es de suma importancia pues indica si una zona esta con sobrepersonal o, por el contrario, requiere más personal. Por ejemplo, si la mitad de los transformadores de una zona fueron atendidos por cuadrillas de otra zona, esto quiere decir que la zona no posee la suficiente cantidad de cuadrillas como para atenderse a sí misma adecuadamente.

## 6. Metodología de estudio

---

Primero se evalúa el desempeño de la estrategia propuesta con otras estrategias de ruteo. La idea es verificar que las diferentes etapas del modelo efectivamente logran aportar a la restauración de servicio total.

Luego de esto, se evalúan diferentes estados de preparación del sistema considerando un número de cuadrillas realista. Los estados de preparación del sistema dependen del tipo de estructura de la red y de la cantidad de cuadrillas de restauración que posee cada centro de operaciones. En base a los resultados de esta etapa se coloca un nuevo centro de operaciones y se hace un análisis similar.

Las métricas de comparación serán todas o algunos de los parámetros obtenidos del bloque de Procesamiento de Resultados del Modelo de Ruteo de Cuadrillas y derivadas de ellas. También se utilizará la métrica de resiliencia adoptada en [2] para comparar el desempeño de cada caso, aunque esto se hará críticamente teniendo en cuenta que no se ha adoptado universalmente y por lo tanto, representa sólo una métrica de referencia que permite la comparación de los resultados.

Todos los casos fueron ejecutados en computadoras con el sistema operativo Ubuntu 16.04 por a las siguientes razones:

- No es posible modificar el número de archivos abiertos en Windows por proceso (máximo 1024).
- No es posible hacer conversar exitosamente el Modelo con el ejecutable de LKH-3 por factores de seguridad en Windows.
- La cantidad de procesos ejecutados en segundo plano en Windows ralentizan el Modelo más de lo que debería.
- Consola inflexible (Símbolos de sistema) en Windows, no es posible tener privilegios tipo *root*<sup>6</sup> en el sistema.

### 6.1 Comparación con otras estrategias de ruteo

El objetivo de esta etapa es comparar la estrategia de ruteo de cuadrillas propuesta con otras posibles estrategias que podría adoptar la distribuidora, ya sea por conveniencia o por falta de recursos. Las estrategias se evalúan bajo los siguientes parámetros comunes:

- **Número de zonas:** El número de zonas es igual al número de centros de operaciones. En este caso se cuentan con 4 centros de operaciones, centros con los que se sabe cuenta la distribuidora actualmente.
- **Número de Cuadrillas:** El número de cuadrillas es de 25 cuadrillas por centro, haciendo un total de 100 cuadrillas en el sistema. El número fue escogido teniendo en cuenta que el número de cuadrillas totales disponibles para el terremoto del 27 de

---

<sup>6</sup> Los privilegios tipo Root o de Administrador en un sistema operativo otorgan al usuario todos los derechos sobre el sistema.

febrero del 2010 [17]. Sin embargo, es un número meramente referencial y no corresponde necesariamente al número real de cuadrillas que a la fecha tiene disponible la empresa distribuidora.

- **Tipo de estructura:** El tipo de estructura es anclada o con consideraciones sísmicas en su diseño. La decisión de este parámetro se debe a que la distribuidora toma en cuenta diseños sísmicos en la elección de equipos [60], así como la norma NMT-001 indica que la colocación de redes públicas y de servicio debe considerar grados de flexibilidad ante oscilaciones de estructuras por movimientos sísmicos, junto con dar otras indicaciones a instalaciones eléctricas [61]. Por último, la CIGRE corrobora una adecuada normativa de construcción en distribución debido a la baja tasa de fallas en infraestructuras eléctrica [62] frente al sismo del 27 de febrero del 2010.

Las estrategias a comparar son las siguientes:

- **Caso Base (C\_B):** Corresponde a la metodología de ruteo de cuadrillas propuesta en la sección anterior: Zonificación por centro de operaciones, clusterización por número de cuadrillas, resolución del problema CCVRP según transformadores prioritarios y cooperación entre cuadrillas. Este es la estrategia que se comparara con el resto.
- **Caso No Cooperativo (C\_NC):** Las cuadrillas no pueden cooperar unas con otras, estando relegadas únicamente a reparar sus clusters. Se mantiene la zonificación por centros de operaciones, la clusterización por número de cuadrillas y la resolución del problema CCVRP. Esto quiere decir que cada cuadrilla únicamente atiende los transformadores de su cluster asignado y ningún otro más.
- **Caso No Guiado (C\_NG):** Las cuadrillas no poseen una lista de prioridades por transformador, por lo que solo se encargan de reparar sus transformadores usando únicamente algoritmo LKH-3 y la cooperación entre cuadrillas. En este caso existe zonificación, clusterización, resolución del problema CCVRP y cooperación de cuadrillas, aunque todo esto se hace sin considerar la cantidad de energía no suministrada y solo se ven distancias entre transformadores. Esto quiere decir que se intenta restaurar el servicio en el menor tiempo posible sin considerar ningún otro factor adicional.
- **Caso Aleatorio Zonal (C\_RZ):** Las cuadrillas atienden transformadores de forma aleatoria o por orden de llamada en su zona de operación. Existe zonificación, no así clusterización ni se resuelve el problema CCVRP. En este caso todas las cuadrillas en una zona se reparten los transformadores dañados en partes iguales y de forma aleatoria. Debido a que el tiempo de reparación de cada transformador depende de su falla, no todas las cuadrillas repararan todos sus transformadores asignados al mismo tiempo, por lo que se acepta cooperación entre ellas.
- **Caso Guiado Zonal (C\_GZ):** Las cuadrillas atienden los transformadores de su zona siguiendo únicamente la lista de prioridades, sin considerar ruteo óptimo ni clusterización, por lo que se pretende minimizar la energía no suministrada dentro de cada zona. En este caso existe zonificación, no así clusterización ni resolución del problema CCVRP. Al existir la lista de prioridades de los transformadores en cada zona, el despacho de cada una de sus cuadrilla se da en paralelo a los primeras  $N_c$

transformadores con mayor prioridad, luego se atienden en paralelo los  $N_c$  transformadores siguientes y así sucesivamente. Explicado de otra forma, si  $c_k$  es la cuadrilla  $k$  de una zona arbitraria y  $\{T_i\}_{i=1}^{N_T}$  es el conjunto ordenado de transformadores en la misma zona, entonces el despacho de las  $N_c$  cuadrillas en la zona será por medio de la asignación  $\{(c_1, T_1), (c_2, T_2), \dots, (c_{N_c}, T_{N_c}), (c_1, T_{N_c+1}), \dots\}$ . La Figura X ilustra este despacho gráficamente este ruteo para 4 cuadrillas y 12 transformadores por zona, omitiendo la posible cooperación entre ellas.

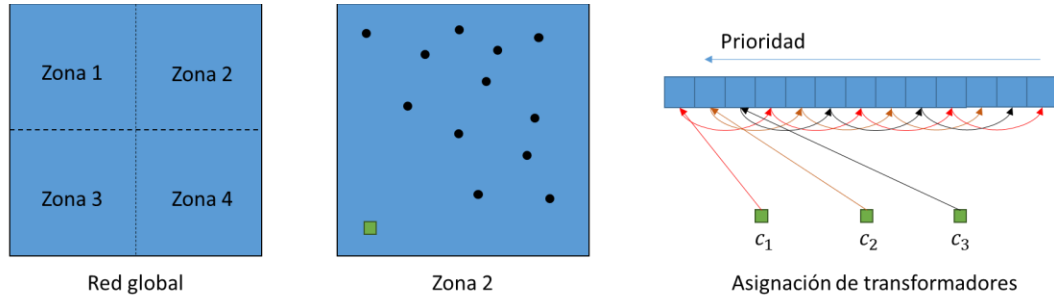


Figura 6.1 Asignación de transformadores para el Caso Guiado Zonal

- **Caso Guiado Total (C\_GT):** Las cuadrillas atienden los transformadores siguiendo únicamente la lista de prioridades global de la red. No se considera zonificación, clusterización ni resolución del problema CCVRP y se admite cooperación entre cuadrillas. Es similar al Caso Guiado Zonal, solo que esta vez se considera la lista de prioridades de todos los transformadores que conforman la red, y ahora  $N_c$  es el número de total de cuadrillas disponibles en toda la red. La idea en este caso es minimizar la energía no suministrada de la red ante un evento BPAI.

La Tabla 6.1 resume las características de cada estrategia.

Tabla 6.1 Características las estrategias de ruteo

Casos	Zonificación	Clusterización	LKH	Prioridad	Cooperativo
C_B	SI	SI	SI	SI	SI
C_NC	SI	SI	SI	SI	NO
C_NG	SI	SI	SI	NO	SI
C_RZ	SI	NO	NO	NO	SI
C_GZ	SI	NO	NO	SI	SI
C_GT	NO	NO	NO	SI	SI



## 6.2 Desempeño para diferentes estados de preparación de la red

Se comparan diferentes estados de preparación del sistema distribución variando el número de cuadrillas totales, el tipo de infraestructura de los circuitos de distribución y el número de centros de operaciones.

En primer lugar, se hacen pruebas con 4 centros de operaciones para los dos tipos de estructuras, con 100 y 148 cuadrillas respectivamente. Las 148 cuadrillas se deben al número de cuadrillas disponibles por la distribuidora durante las nevazones del año 2017, que si bien fueron 150, se redujeron a 148 ya que como supuesto cada centro debe tener la misma cantidad de cuadrillas [17]. Utilizando este valor como referencia y al supuesto de que cada centro de operaciones posee el mismo número de cuadrillas, asumimos que 37 cuadrillas es el tope máximo que puede soportar un centro de operaciones. Los parámetros de comparación son los globales obtenidos por el modelo y de resiliencia.

Luego, por medio de los parámetros obtenidos por cuadrilla se decide una localización para un nuevo centro de operaciones [17] y se compara este con el caso base, el cual sería **CS\_100**<sup>7</sup>, es decir, con 5 centros de operaciones, componentes ancladas y 20 cuadrillas por centro. Se pretende verificar que la nueva localización mejora los parámetros de comparación independiente del número de cuadrillas en la red. Esto da paso a comparar el desempeño de la restauración de la red con este nuevo centro, siguiendo los mismos parámetros utilizados con 4 centros, es decir, 25 y 37 cuadrillas por centro de operaciones.

Con estos resultados se obtiene una línea de tendencias de los resultados globales por número total de cuadrillas en la red de distribución de referencia. También se realizan simulaciones con 40 y 400 cuadrillas para los dos tipos de estructuras para obtener resultados globales en casos extremos. Estos casos no se analizan en detalle debido a que no corresponden a un número realista de cuadrillas por centro de operaciones y solo pretenden aproximar las curvas de los parámetros globales por cuadrillas que serán calculadas más adelante.

Finalmente se hace un análisis de resiliencia sistémica para todos los casos estudiados y se comparan las estrategias del caso anterior con diferentes estados de preparación de la red.

La Tabla 6.2 resume los casos a estudiar en esta sección

*Tabla 6.2 Casos a evaluar de diferentes estados de preparación de la red de distribución*

Casos	N° de Centros	N° de Cuadrillas	Estructura
CE_100	4	100	Estándar
CE_148	4	148	Estándar
CS_100	4	100	Sísmica
CS_148	4	148	Sísmica

---

<sup>7</sup> Los casos con diferentes estados de la red poseen nombres de tipo CX\_YYY, con X = E (Componente Estándar) o S (Componente Sísmica) e YYY = número de cuadrillas totales.

<b>CS_100Z5</b>	5	100	Sísmica
<b>CE_125</b>	5	125	Estándar
<b>CE_185</b>	5	185	Estándar
<b>CS_125</b>	5	125	Sísmica
<b>CS_185</b>	5	185	Sísmica
<b>CE_40</b>	4	40	Estándar
<b>CE_400</b>	4	400	Estándar
<b>CS_40</b>	4	40	Sísmica
<b>CS_400</b>	4	400	Sísmica

## 7. Resultados y análisis

---

### 7.1 Estrategias de ruteo de cuadrillas

En esta sección se comparan las estrategias de ruteo con la metodología diseñada a lo largo de este trabajo de memoria. Primero se describe el estado de daño de la red provocado por el sismo, luego se analiza el desempeño del Caso Base lo que da pie a la comparación con el resto de las estrategias. Finalmente se comparan todos los casos por medio de la métrica de resiliencia escogida y se hace un análisis crítico de sus resultados.

Todos los resultados mostrados corresponden a valores obtenidos de 1000 simulaciones, es decir, cada caso consta de mil simulaciones del evento BPAI de tipo sísmico junto con el ruteo correspondiente.

#### 7.1.1 Estado de la red anclada después del evento BPAI

La Tabla 7.1 resume el estado esperado de la red después del evento BPAI, correspondiente al sismo del 27 de febrero del 2010 modelado por la aceleración sísmica en forma de aceleración espectral de periodo 1.0 segundos, considerando circuitos de distribución con componentes sísmicas. De ella se extrae la siguiente información:

- La Zona 1 y Zona 2 son las zonas que cuentan con el mayor número de transformadores dañados. También son las zonas que poseen la mayor cantidad de transformadores en estado de daño Extensivo y Completo.
- La Zona 3 y 4 presentan daños mayormente Menores y Moderados. Esto se suma a que ambas zonas poseen el menor número de transformadores dañados.
- Los daños globales de la red son principalmente Moderados y Extensivos. Además, los daños Moderados presentan una repartición medianamente uniforme para todas las zonas.

*Tabla 7.1 Número de transformadores dañados por el sismo en una red anclada*

Zona\Daño	Menor	Moderado	Extensivo	Completo	TOTAL
Zona 1	472	2.583	3.113	776	6.944
Zona 2	43	2.618	3.545	237	6.443
Zona 3	234	2.574	447	1	3.256
Zona 4	747	3.244	418	3	4.412
TOTAL	1.496	11.019	7.523	1.017	

Conocer el estado de la red permite prever las dificultades que presentara cada zona. Por ejemplo, se espera que la Zona 1 y Zona 2 requieran de la cooperación de las cuadrillas de la Zona 3 y 4 para reparar el total de sus transformadores en un tiempo razonable.

Otro resultado destacable es que de la tabla se puede extraer una cota inferior del tiempo total de restauración de la red. Si suponemos que las cuadrillas pueden llegar instantáneamente a su destino y entre cada una de ellas se distribuyen los tiempos de reparación en partes iguales, el tiempo teórico esperado en reparar la red completamente está dado por la siguiente fórmula:

$$T_{teo}^C = \frac{[(\sum_{i \in DS} N_i \cdot T_i^{Re}) + [(\sum_{i \in DS} N_i \cdot T_i^{Re})/0,5] \cdot 0,5]}{C} \quad 7.1$$

Con  $C$  el número de cuadrillas,  $N_i$  el número de transformadores con un estado de daño  $i$  y  $T_i^{Re}$  el tiempo, en días, necesario para reparar un transformador con un estado de daño  $i$ . El lado izquierdo de la suma contempla todas las horas de trabajo necesarias para reparar la red, mientras que el lado derecho considera las horas de descanso de las cuadrillas. La función *piso* se aplica para conocer el número de descansos realizados, la cual se multiplica por las horas de descanso. El tiempo teórico de reparación para una red de estructura anclada y 100 cuadrillas con el 97% de sus transformadores con algún tipo de daño es de **12,43 días**.

### 7.1.2 Estrategia: Caso base

Los parámetros fundamentales a comparar entre casos corresponden a los resultados globales por simulación: Tiempo de reposición del 100% (También llamado tiempo de reposición total), tiempo de reposición del 98% (También llamado tiempo de reposición parcial), energía no suministrada total, costo de falla total y distancia total recorrida por las cuadrillas. Los histogramas de la Figura 7.1 muestran que cada uno de los parámetros presenta una distribución gaussiana a excepción del tiempo de reposición parcial, el cual posee dos distribuciones gaussianas en intervalos de tiempos distantes. Este fenómeno se debe principalmente a las horas de descanso necesarias que debe cumplir cada cuadrilla, pues teniendo en cuenta que cada cuadrilla inicia su jornada a la misma hora, es esperable que la gran mayoría de ellas trabaje en intervalos de tiempos similares. Bajo este argumento, uno podría concluir que el mismo resultado debería darse para el tiempo de reposición total de la red, cosa que efectivamente sucede en otros casos estudiados más adelante, pero que se evitan en la mayoría de los casos gracias a la cooperación entre cuadrillas. De hecho, los tiempos de reposición parciales de la red tiene una mayor relación con la forma en que se levanta carga que con un ruteo óptimo.

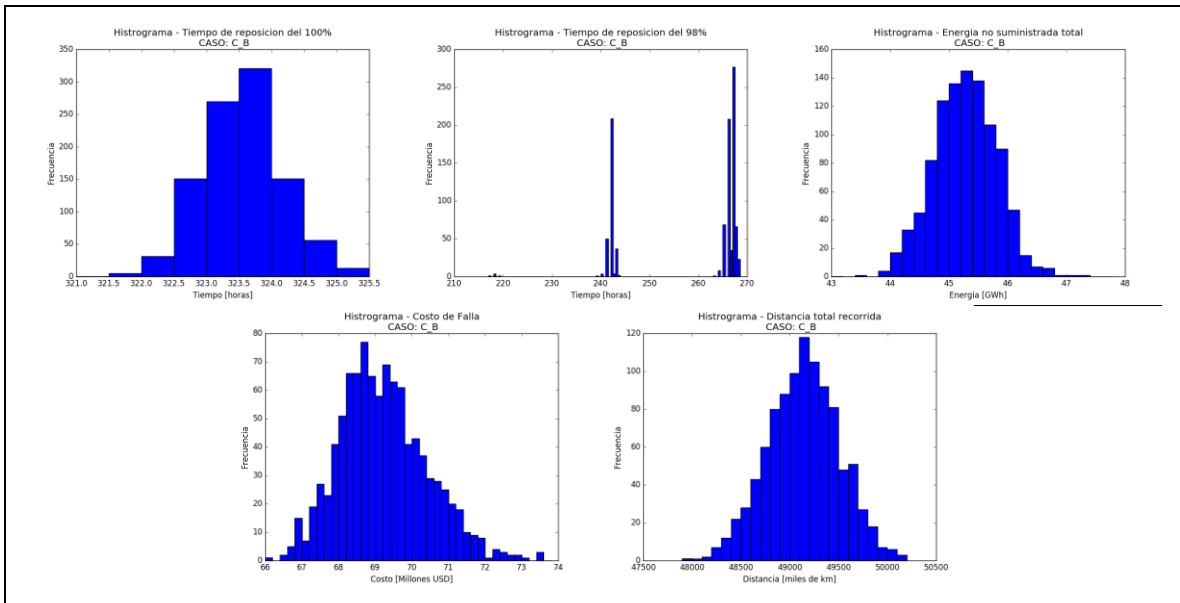


Figura 7.1 Histogramas de resultados globales para el Caso Base

El hecho de que los resultados globales presenten distribuciones gaussianas permite simplificar el análisis de cada caso a sus parámetros esperados, es decir, utilizando el promedio de todas las simulaciones para cada parámetro global. Por ello, desde ahora cada vez que se mencione un parámetro ya sea global, entre cuadrillas o derivada de ellas, se estará hablando de su valor esperado a menos que se indique lo contrario. Los histogramas del resto de los casos se encuentran en el Anexo A.

Concluido el punto anterior, la Tabla 2.1 muestra los resultados globales de cada parámetro para el Caso Base. Un punto destacable de los resultados es el hecho de que el tiempo de reposición total de la red es mayor al tiempo teórico de reparación de la red en solo 1,05 días. Esto se debe a las siguientes razones

- **Tiempo de viaje:** Si se divide la distancia recorrida total por el número de cuadrillas junto a la velocidad de viaje, el resultado es un tiempo de viaje de 12,28 horas. A ello debemos sumarle el tiempo de descanso que se da al superar las 12 horas de trabajo. En total, el tiempo extra producido por el ruteo de las cuadrillas es de 1,01 días.
- **Deficiencias en la cooperación:** Las deficiencias en cooperación se deben principalmente a que no es posible dividirse los tiempos de restauración del sistema en partes iguales. Como los puntos a reparar representan cantidades discretas de tiempos de restauración, se requeriría encontrar, si es que existe, la permutación que hiciera que todas las cuadrillas terminaran al mismo tiempo. La deficiencia en cooperación se calcula como el resto entre los 1,05 días de diferencia con el tiempo teórico y el tiempo de viaje, lo que da como resultado un total de 57 minutos. Esto no implica que el algoritmo esta solo 57 minutos sobre el óptimo, ya que el óptimo depende de la ruta escogida, la cual afecta directamente el tiempo de viaje.

Tabla 7.2 Resultados globales del Caso Base

	Tiempo 100% [días]	Tiempo 98% [días]	Energía NS [GWh]	Costo Falla [Mill USD]	Distancia Rec. [km]
<b>C_B</b>	13,48	10,79	45,28	138,44	49.142

El resto de los parámetros se discutirá más adelante al comparar el caso base con el resto de las estrategias. Por el momento, un análisis de la restauración de servicio de Figura 7.2 permite visualizar como éste presenta un comportamiento convexo, con una pendiente pronunciada en las primeras horas que va disminuyendo a medida que avanza el tiempo. Los pequeños picos en los intervalos se deben a cambios en la distribución de energía por la red a medida que pasan las horas del día. A ciertas horas los transformadores con fallas no suministran suficiente energía en condiciones normales al sistema comparado con otros transformadores operativos, haciendo que suba el porcentaje de energía suministrada momentáneamente, lo que se puede observar en la Figura 7.3.

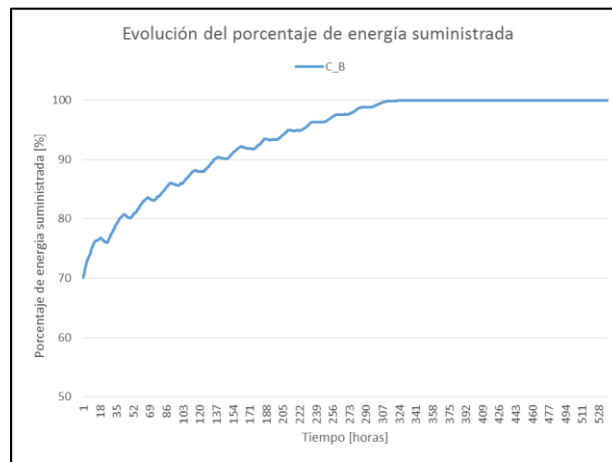


Figura 7.2 Curva de porcentaje de carga suministrada para el Caso Base

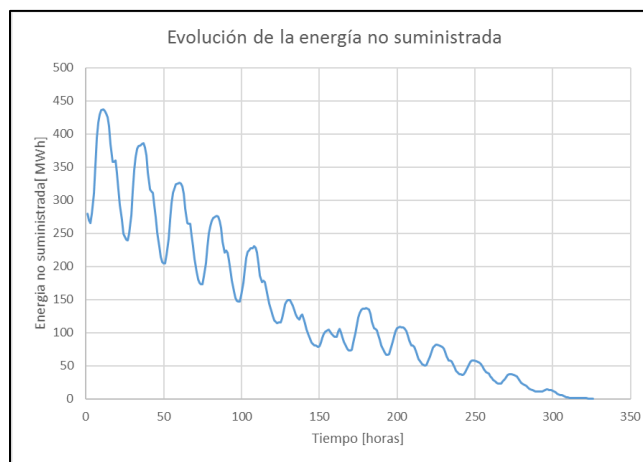


Figura 7.3 Curva de energía no suministrada horaria para el Caso Base

Un análisis más detallado del comportamiento de las cuadrillas se puede visualizar en la Figura 7.4, que ilustra los tiempos promedios que toma cada cuadrilla en finalizar su trabajo. Como se ha mencionado anteriormente, las cuadrillas se enumeran del 0 hasta el número total de cuadrillas menos uno, donde la división por zonas se da por intervalos consecutivos de cuadrillas, es decir, si cada zona cuenta con 25 cuadrillas, entonces la Zona 1 tendrá asignadas las cuadrillas 0 al 24, luego la Zona 2 tendrá las cuadrillas del 25 al 49 y así sucesivamente. La curva de la Figura 7.4 preserva esta partición, no así la asignación de cuadrillas dentro de ella, es decir, la cuadrilla 1 de la simulación 0 no necesariamente es la misma que la siguiente simulación, ya que la asignación se realiza por medio del algoritmo de clusterización que difieren por cada simulación. La curva se obtiene ordenando los tiempos de trabajo de cada cuadrilla en cada zona y se promedian por simulación, dando como resultado una curva creciente por intervalos.

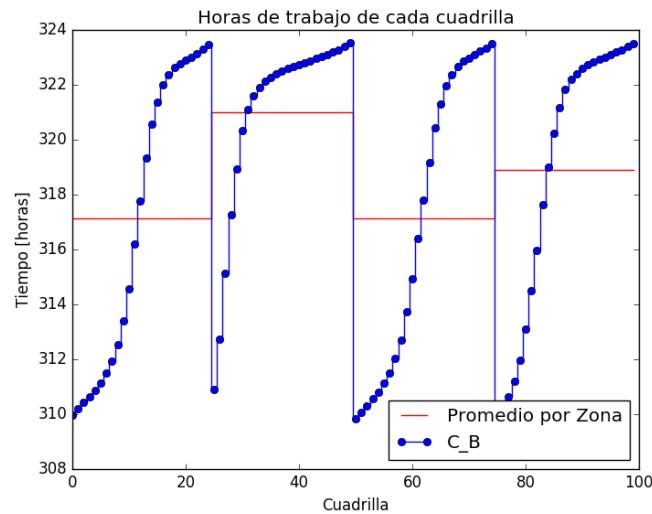


Figura 7.4 Horas de trabajo de cada cuadrilla para el Caso Base

Los resultados indican que existe una diferencia de 13,86 de horas entre la cuadrilla que menos tarda en reparar sus transformadores asignados y la que más lo hace, encontrándose en los márgenes de error previstos del algoritmo cooperativo. Esto se debe a que el algoritmo es de tipo greedy, convergiendo en un mínimo local. Más precisamente, la cuadrilla con menor tiempo de trabajo  $c_i$  requiere descansar para poder reparar el último transformador de la cuadrilla con mayor cantidad de horas de trabajo  $c_j$ , lo que a su vez, sumando los tiempos de viaje y reparación, aumentarían las horas de trabajo de  $c_i$  por sobre los de  $c_j$ . En el mejor de los casos se podría reducir las horas de reposición total en 6,93 horas, valor despreciable considerando que el 98% de la energía suministrada fue reestablecida hace 3 días. De cualquier modo, la diferencia de tiempos va cobrando mayor relevancia a medida que los tiempos de restauración total van decreciendo.

Como análisis final sobre el Caso Base, se muestra la curva de probabilidad de que un transformador sea reparado en cierta hora (también llamado Curva de Restauración). Esta curva se deriva de los resultados del Modelo de Ruteo de Cuadrillas y se puede interpretar de dos formas:

- A nivel unitario: Como lo dice su nombre, es la probabilidad de que un transformador sea reparado luego de cierta cantidad de horas pasado el evento BPAI.
- A nivel sistémico: Como utiliza valores hasta la unidad y se deriva de un histograma del estado de los transformadores de todas las simulaciones, se puede interpretar como el porcentaje esperado de transformadores reparados con un cierto nivel de daño en una cierta hora luego del evento BPAI.

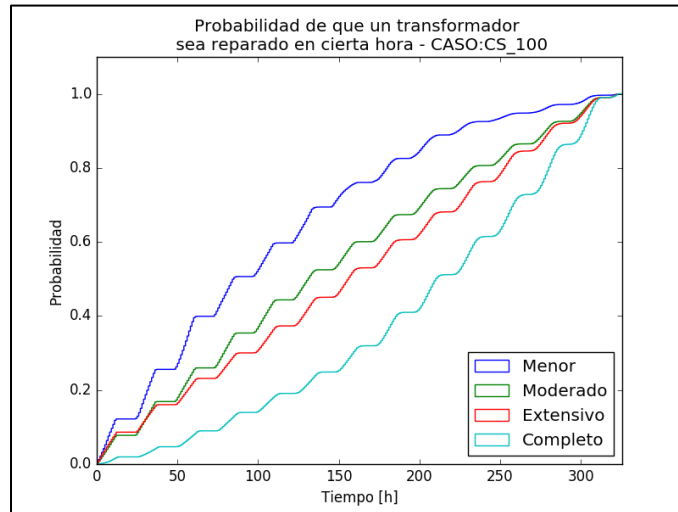


Figura 7.5 Curva de restauración del Caso Base

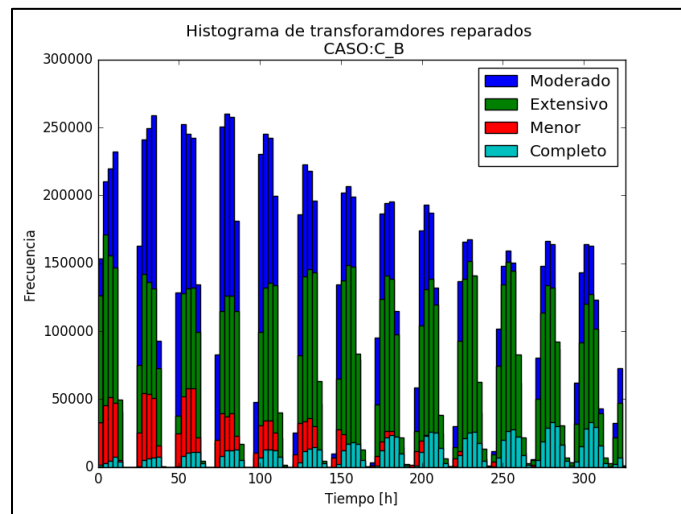


Figura 7.6 Histograma de transformadores reparados por estado de daño para el Caso Base

Para comprender el comportamiento de la estrategia, resulta más conveniente utilizar la interpretación sistémica de la curva de restauración. Bajo esa interpretación, el algoritmo prioriza la restauración de transformadores con un estado de daño Menor por sobre el resto, seguido por los transformadores con daños Moderados, Extensivos y Completos. Ello podría conducir a la creencia de que el algoritmo Euclidiano da prioridades a las cuadrillas en el orden mostrado, cosa que no necesariamente es cierta, ya que existen más restricciones a las que están sujetas las cuadrillas en las primeras horas de ruteo. Esto queda reforzado con el histograma de la Figura 7.6, donde se muestra la frecuencia de reparación de los



transformadores según su estado de daño para todas las simulaciones. En ella se observa como gran parte de los transformadores con estado de daño Menor son recuperados cerca de las 150 primeras horas. Algo similar ocurre con los transformadores con estado de daño Moderado, aunque estos requieren seguir siendo atendidos a lo largo del periodo de restauración. Por otro lado, los transformadores con daño extensivo son atendidos de manera constante durante toda la restauración, y por último los transformadores con daño completo son atendidos en su mayoría cerca del final del periodo de restauración.

El análisis de la siguiente estrategia permite determinar la razón del comportamiento de la curva de restauración.

### 7.1.3 Estrategia: Caso no cooperativo

La idea de este caso es mostrar la importancia de que ninguna cuadrilla quede ociosa en el proceso de restauración de la red. La Tabla 7.3 muestra los resultados del Caso No Cooperativo, así como la diferencia entre este caso y el Caso Base. De ella se desprende lo siguiente:

- Hay cerca de dos meses de diferencia en la restauración total de la red, lo que indica que a ciertas cuadrillas se les asignan clusters que no son capaces de resolver por sí solos en tiempo razonable
- La diferencia en la restauración parcial es de solo una semana. Este resultado se atribuye a que en cada cluster existe un ruteo óptimo y prioritario, permitiendo restaurar el servicio en un tiempo medianamente razonable.
- La no cooperación tiene graves efectos en la energía no suministrada total y con ello, en los costos de falla.
- Las distancias recorridas son menores al Caso Base en cerca de 8000 [km] (se recorren menos kilómetros, pero la restauración de servicio tarda significativamente más). Esto implica que la cooperación entre cuadrillas le cuesta esa cifra al Caso Base, que son 2 horas más de viaje para cada cuadrilla.

Tabla 7.3 Resultados globales del Caso No Cooperativo

	Tiempo 100% [días]	Tiempo 98% [días]	Energía NS [GWh]	Costo Falla [Mill USD]	Distancia Rec. [km]
<b>C_NC</b>	61,97	17,12	62,36	170,20	40.330
<b>C_B</b>	13,48	10,79	45,28	138,44	49.142
<b>Comp. C_B</b>	<b>48,49</b>	<b>6,33</b>	<b>17,08</b>	<b>31,76</b>	-8.812

Los efectos de la no cooperación de cuadrillas se pueden apreciar tanto en la Figura 7.7 como en la Figura 7.8, donde en las primeras 100 horas pasado el evento BPAI ambas curvas presentan el mismo comportamiento de carga suministrada. Pasada estas horas, el Caso No Cooperativo presenta una disminución en su pendiente cada vez más notoria, lo que decanta en tiempos excesivos en la restauración del servicio. Esto se aprecia mejor en la curva de energía no suministrada horaria, donde luego de las 160 horas no parece haber una mejora en la restauración de suministro eléctrico.

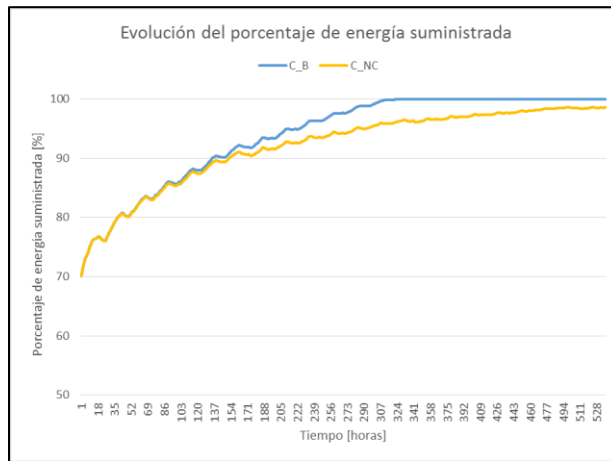


Figura 7.7 Curva del porcentaje de carga suministrada para el Caso Base y Caso No Cooperativo

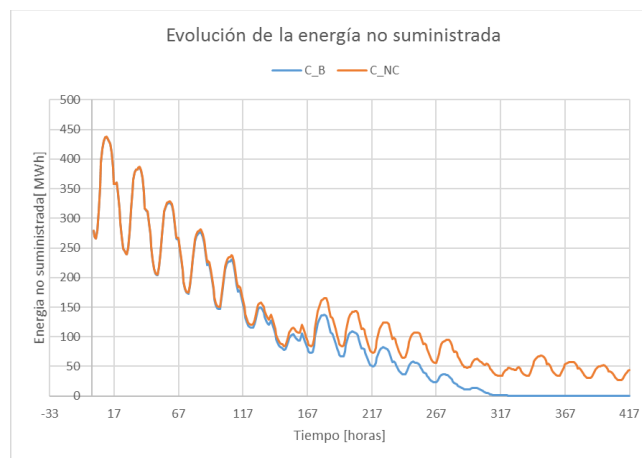


Figura 7.8 Curva de energía no suministrada horaria para el Caso Base y Caso No Cooperativo

El comportamiento de la curva de porcentaje de carga suministrada se justifica con los resultados obtenidos de todas las horas de trabajo por cuadrilla de la Figura 7.9. De este resultado es posible confirmar lo siguiente: Las Zonas 3 y 4 finalizan la restauración de sus transformadores mucho antes que las Zonas 1 y 2, por lo que son fundamentales en la cooperación de cuadrillas. Este resultado da pie para un análisis más detallado de la restauración de la red por medio de la utilización de las curvas de restauración de la Figura 7.10.

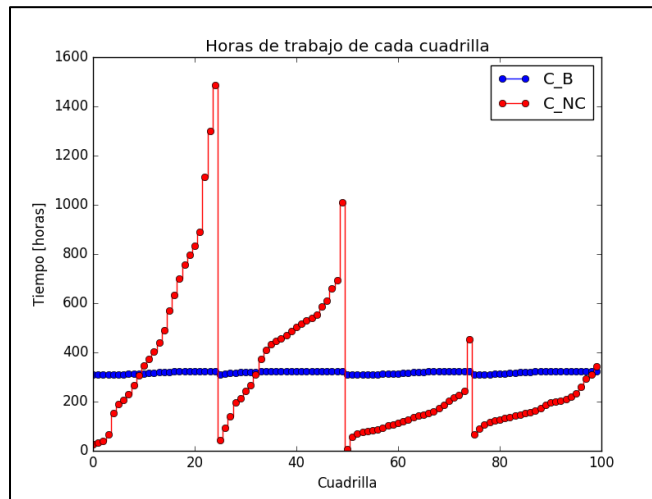


Figura 7.9 Horas de trabajo de cada cuadrilla para el Caso Base y Caso No Cooperativo

En las primeras 300 horas de restauración la gran mayoría de los transformadores con daños Menores y Moderados ya han sido reparados. Al mismo tiempo, en las primeras 300 horas la gran mayoría de las cuadrillas de la Zona 3 y 4 ya han finalizado sus deberes. Esto indica lo siguiente: La forma de la curva de restauración del Caso Base es producto de la zonificación de la red. Al mismo tiempo, cerca del 60% de los transformadores con daños Extensivos ya han sido reparados, lo que da indicios de la priorización de transformadores en las Zonas 1 y 2. El tópico de transformadores prioritarios será explorado en detalle en el análisis de estrategias de tipo guiadas.

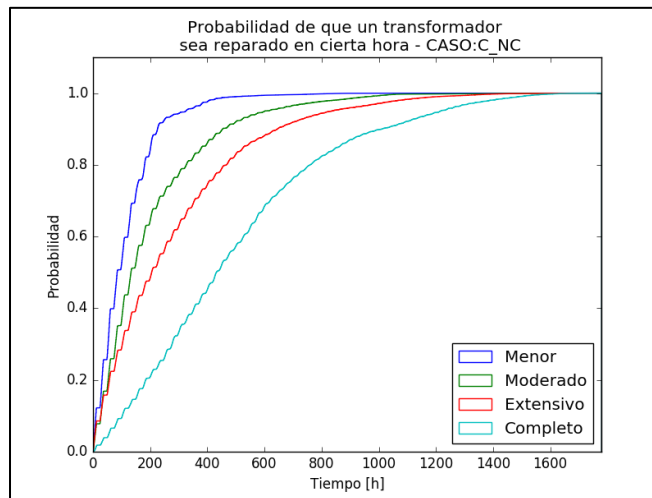


Figura 7.10 Curva de restauración del Caso No Cooperativo

#### 7.1.4 Estrategia: Caso no guiado

El caso no guiado consiste en un caso donde solo el ruteo óptimo de cuadrillas cobra importancia a la hora de restaurar el servicio de la red, sin importar el nivel de carga que pueda satisfacer un transformador al ser restaurado.

La Tabla 7.4 muestra que bajo esta estrategia, los tiempos de restauración total logran bajar a casi 7 horas, así como las distancias recorridas logran reducirse en 12.000 kilómetros. Esto se debe principalmente a que el modelo únicamente intenta resolver el problema de distancias

entre transformadores, es decir, intenta minimizar el tiempo de restauración global de la red sin otras consideraciones. Sin embargo, el tiempo parcial de restauración aumenta en cerca de 14 horas, así como la energía no suministrada y los costos de falla. El hecho de que el tiempo parcial de restauración aumente se debe precisamente a que las cuadrillas no estén siendo guiadas, asignando rutas a las cuadrillas hacia transformadores cercanos unos con otros de baja carga.

Tabla 7.4 Resultados globales del Caso No Guiado

	Tiempo 100% [días]	Tiempo 98% [días]	Energía NS [GWh]	Costo Falla [Mill USD]	Distancia Rec. [km]
<b>C_NC</b>	13,20	11,41	68,62	246,82	36.965
<b>C_B</b>	13,48	10,79	45,28	138,44	49.142
<b>Comp. C_B</b>	-0,28	0,62	23,34	108,38	-12.177

La Figura 7.11 refuerza esta observación, donde la curva de porcentaje de energía suministrada posee una pendiente constante, es decir, las cuadrillas no están considerando reparar sectores con concentraciones de carga considerables en las primeras etapas de la restauración de la red. El Caso No Guiado al presentar zonificación posee una curva de restauración similar a la del Caso Base, por lo que no se presentará ya que no aporta información adicional a lo ya mencionado. Las curvas de restauración de todos los casos estudiados pueden ser consultadas en el Anexo B.

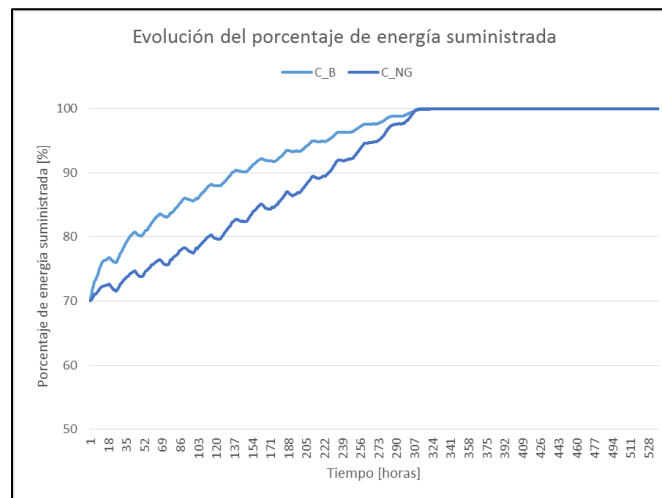


Figura 7.11 Curva del porcentaje de carga suministrada para el Caso Base y Caso No Guiado

El hecho de que la estrategia no guiada logre acercarse 7 horas al tiempo de restauración teórico es un resultado relevante. Aun así, el costo incurrido para lograr dicho cometido resulta ser demasiado alto como para considerarlo una estrategia viable. La distribuidora no solo desea minimizar los tiempos totales de restauración, sino que también desea levantar la mayor cantidad de carga en el menor tiempo posible.

### 7.1.5 Estrategia: Caso aleatorio zonal

La idea del caso aleatorio zonal es modelar el comportamiento de las cuadrillas de restauración sin protocolos definidos ante eventos BPAI, donde no se tienen conocimientos de cargas y por lo tanto no hay forma de hacer una priorización. A este caso también se le denomina en este trabajo “orden de llamada”, ya que la cuadrilla irá al primer lugar donde se reporte una falla, que en este caso será de forma aleatoria determinado una vez por simulación en cada cuadrilla. Una cuadrilla atenderá primero las llamadas de su zona y cooperará con el resto de las cuadrillas después.

Los resultados de la Tabla 7.5 indican que este método resulta tener un mal rendimiento en lo que es la restauración de servicio en general. Existen tiempos de viajes excesivos al no realizar una planificación de ruta con sobre 200.000 kilómetros recorridos al Caso Base, lo que son casi 4 días desperdiciados únicamente viajes y los descansos que implican. Otro aspecto a destacar es el hecho de que solo supera en tiempo de restauración parcial al caso no cooperativo en 2 días y posee un peor rendimiento en energía no suministrada a los Casos No Guiado y No Cooperativo.

Tabla 7.5 Resultados globales del Caso Aleatorio Zonal

	Tiempo 100% [días]	Tiempo 98% [días]	Energía NS [GWh]	Costo Falla [Mill USD]	Distancia Rec. [km]
<b>C_NC</b>	17,67	15,1	88,86	314,28	261.611
<b>C_B</b>	13,48	10,79	45,28	138,44	49.142
<b>Comp. C_B</b>	4,19	4,31	43,58	175,84	212.469

Con respecto a la restauración del servicio en general, la curva de porcentaje de carga suministrada presenta características similares a la del Caso No Guiado, es decir, una pendiente constante aunque de menor inclinación, lo que supone una mayor cantidad de energía no suministrada total. La curva de restauración es similar al del Caso Base dada la presencia de zonificación y no aporta mayor información a la rescatada de los resultados mostrados hasta ahora.

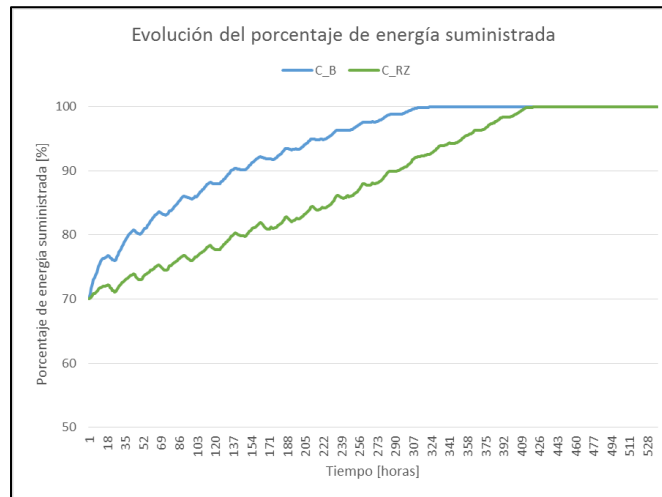


Figura 7.12 Curva del porcentaje de carga suministrada para el Caso Base y Caso Aleatorio Zonal

A modo de análisis final, es fundamental que existan planes y protocolos de acción ante eventos BPAI por parte de la distribuidora para evitar cualquier improvisación en lo que es la restauración de servicio y con ello, evitar deficiencias de este tipo. Incluso un plan de acción que únicamente minimice distancias recorridas logra reducir la energía no suministrada de la red en cerca de 20 [GWh] comparados a este caso.

### 7.1.6 Estrategia: Caso guiado zonal

La idea del Caso Guiado Zonal es comparar el desempeño de una estrategia que despacha las cuadrillas hacia transformadores de mayor prioridad sin resolver el problema CCVRP, sino que las despacha hacia los transformadores de mayor prioridad en paralelo con la finalidad de disminuir la mayor cantidad de energía no suministrada, es decir, el algoritmo es de tipo greedy en energía. Los centros tienen nociones de cercanías de los transformadores, por lo que cada cuadrilla atenderá primero su zona y luego cooperará con el resto.

Los resultados muestran una baja en el desempeño de los tiempos de restauración total y, sorpresivamente, en los tiempos de restauración parcial. A pesar ello, logra ahorrarle a la sociedad unos 16,74 millones de dólares en energía no suministrada con respecto al Caso Base. El hecho de que los tiempos de restauración parcial aumenten mientras que la energía no suministrada disminuya implica que existen focos concentrados de transformadores que producen grandes cantidades de energía no suministrada. Al existir estos focos, el Caso Base los agrupa en pocos clusters atendidos por una cuadrilla cada uno. Como no existe clusterización en el Caso No Guiado Zonal, cada transformador parte del foco es atendido en paralelo por todas las cuadrillas disponibles en la zona.

Tabla 7.6 Resultados globales del Caso Guiado Zonal

	Tiempo 100% [días]	Tiempo 98% [días]	Energía NS [GWh]	Costo Falla [Mill USD]	Distancia Rec. [km]
C_NC	16,38	11,16	41,67	121,70	195.285
C_B	13,48	10,79	45,28	138,44	49.142

Comp. C_B	2,90	0,37	-3,61	-16,74	146.143
-----------	------	------	-------	--------	---------

Por otro lado, al no existir un algoritmo de ruteo óptimo, las cuadrillas pierden cerca de 3,5 días en viajes y los descansos que implica. La Figura 7.13 muestra que, a pesar de que el Caso Guiado Zonal logra levantar más carga durante las primeras horas de restauración, esta se va perdiendo eficiencia hasta el punto de cruzar la curva del Caso Base, teniendo como consecuencia tiempos de restauración mayores.

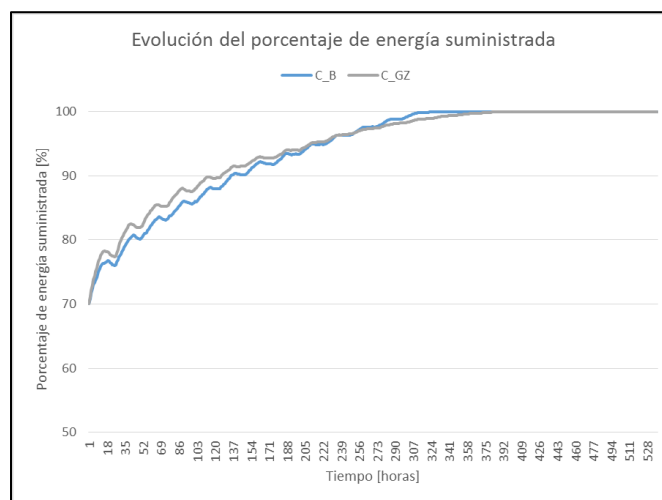


Figura 7.13 Curva del porcentaje de carga suministrada para el Caso Base y Caso Guiado Zonal

Resulta interesante analizar la curva de restauración considerando la ausencia de clusterización y ruteo óptimo. Durante las primeras 100 horas, resulta que los transformadores con daños Extensivos y Moderados son de mayor prioridad que los que poseen daños Menores y Completos, lo que indica que la distribución de falla depende de cada zona, así como la distribución de energía. Cerca de las 200 horas, que es cuando la mayoría de las cuadrillas de la Zona 3 y 4 concluyen con la reparación de sus respectivas zonas, se aprecian grandes escalones en las curvas de restauración para transformadores con daños Menores, lo que implica y refuerza nuevamente que la forma de las curvas de restauración del Caso Base se debe principalmente a la zonificación de la red. Pasada las 200 horas, la curva Moderada se vuelve relativamente plana, mientras que las de daño Completo aumentan su pendiente. Con respecto a las curvas de daño Moderado y Completo, estas presentan una pendiente relativamente constante a lo largo del periodo de restauración.

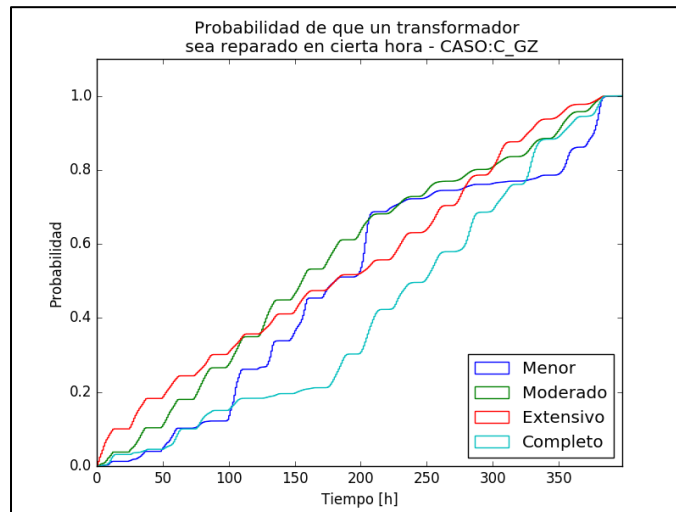


Figura 7.14 Curva de restauración del Caso Guiado Zonal

Resulta interesante el hecho de que, aun disminuida la energía no suministrada, los tiempos hayan aumentado con respecto al Caso Base. Esto implica que una estrategia de tipo guiada zonificada no explota ninguna de sus ventajas a cabalidad, es decir, el hecho de atender transformadores cercanos a su zona y el de atender transformadores con mayor incidencia en el suministro de energía de la red.

### 7.1.7 Estrategia: Caso guiado total

Esta estrategia corresponde al caso extremo del Caso Guiado Zonal, donde las cuadrillas ya no están restringidas a sus zonas y pueden reparar los focos de mayor daño en la red en conjunto. Como se vio en el caso anterior, la zonificación de la red implica una restricción muy fuerte para este tipo de estrategias, por lo que se espera una disminución de los tiempos de restauración parciales, así como el de la energía no suministrada, lo que se confirma de acuerdo a los resultados de la Tabla 7.7.

En efecto, los tiempos de restauración parcial disminuyen en un día y medio con respecto al caso base junto con la disminución de la energía no suministrada en casi 11 [GWh]. Sin embargo, el hecho de no optimizar rutas hace que las cuadrillas pierdan cerca de 5,5 días en viajes y sus descansos correspondientes, resultando en un tiempo de restauración total excesivo.

Tabla 7.7 Resultados globales del Caso Guiado Total

	Tiempo 100% [días]	Tiempo 98% [días]	Energía NS [GWh]	Costo Falla [Mill USD]	Distancia Rec. [km]
<b>C_NC</b>	19,05	9,22	34,24	99,86	335.908
<b>C_B</b>	13,48	10,79	45,28	138,44	49.142
<b>Comp. C_B</b>	<b>5,57</b>	<b>-1,57</b>	<b>-11,04</b>	<b>-38,58</b>	<b>286.766</b>



De las curvas de porcentaje de carga se extrae que, efectivamente, las cuadrillas focalizan sus esfuerzos en reparar los focos de mayor daño en la energía suministrada de la red, lo que implica una curva de porcentaje de carga más pronunciada con respecto al Caso Base.

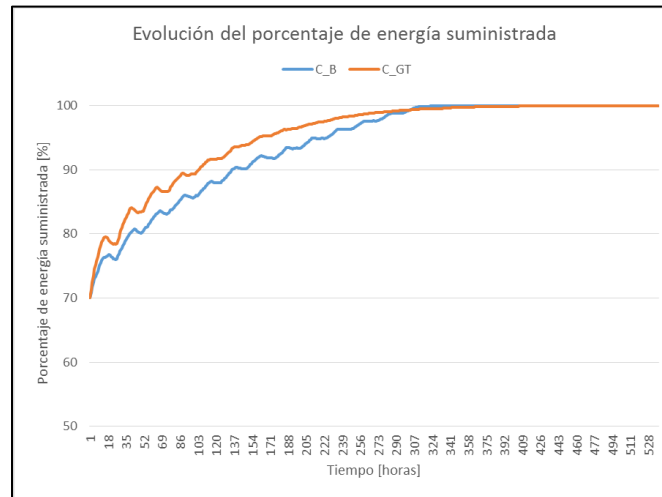


Figura 7.15 Curva del porcentaje de carga suministrada para el Caso Base y Caso Guiado Total

Al existir solo un ruteo de cuadrillas en base a la lista de prioridades de la red global, es posible extraer la clasificación de transformadores prioritarios en base a la curva de restauración del Caso Guiado Total, la que se muestra en la Figura 7.16. Ya de ella es posible concluir con toda seguridad lo siguiente: La priorización de transformadores sigue el orden de daño Extensivo, Completo, Moderado y Menor. La razón de esto es simple: Si un transformador entregara  $E_h$  [MWh] a la red en un horizonte de tiempo dado, entonces de la Ecuación (5.2) es posible deducir que su prioridad será igual a dicha energía multiplicada por los factores 40, 50, 24 y 16 para estados de daño Completo, Extensivo, Moderado y Menor, es decir, seguiría el orden de la curva de restauración del Caso Guiado Total. Como la carga de los transformadores depende de su capacidad y zona definida por las subestaciones primarias de distribución en la red, las cuadrillas también hacen esfuerzos para reparar transformadores con daños Moderados y Menores.

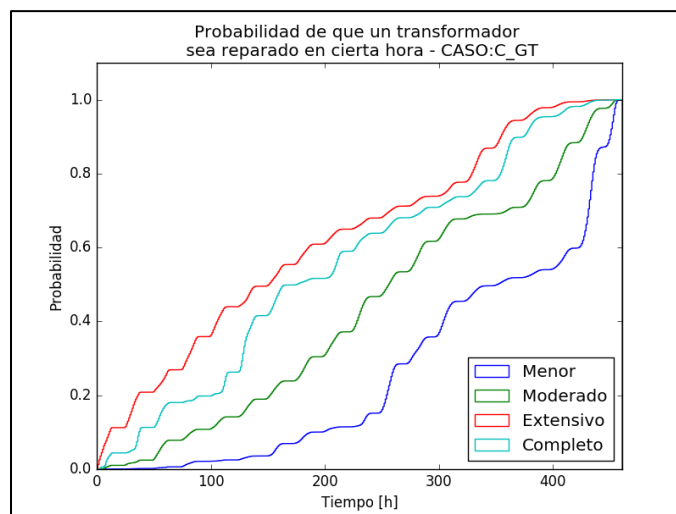


Figura 7.16 Curva de restauración del Caso Guiado Total

De todos los casos extremos estudiados, el hecho de intercambiar buenos resultados entre lo que es energía no suministrada y tiempo de reposición total de la red. La Figura 7.17 compara la evolución de la energía no suministrada acumulada por hora para los Casos Base, No Guiado y Guiado Total, donde la curva sólida representa la energía acumulada horaria promedio y la curva interlineada sus rangos máximos y mínimos extraídos de una de las 1000 simulaciones. La curva indica la existencia de *trade off* en entre ambos parámetros. Además, el *trade off* no resulta tener un comportamiento lineal, es decir, una disminución de los tiempos resulta en un aumento considerable de energía no suministrada y viceversa.

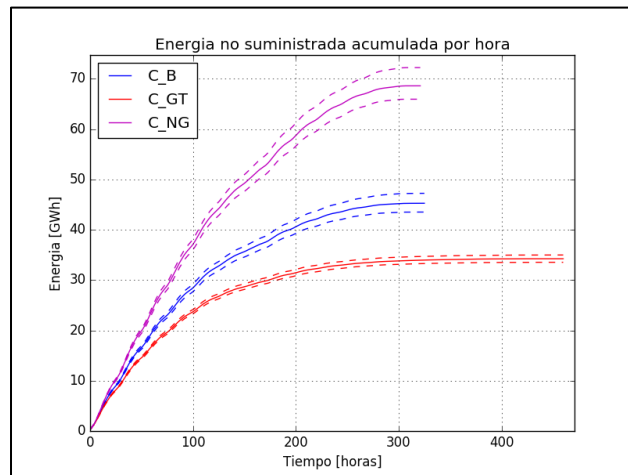


Figura 7.17 Evolución de la energía no suministrada acumulada para diferentes estrategias

Dado a lo anterior, la distribuidora deberá decidir que es mejor para la compañía y sus clientes. Si decide minimizar costos totales de falla, entonces será conveniente minimizar la energía no suministrada en desmedro de los tiempos de reposición total y por ende, los tiempos de espera que deberán soportar algunos de sus clientes. Esta no es una decisión simple dado que la zona de referencia estudiada alberga a casi 5 millones de habitantes, eso sumado a que los focos con mayor carga se concentran en sectores industriales y comerciales, se estarían dejando de lado las zonas residenciales en la restauración de servicio, resultando en miles de habitantes (no así clientes) sin energía durante varias semanas. Por otro lado, si la distribuidora pretende minimizar sus tiempos de restauración totales, deberá escoger una de las estrategias que combinen ruteo óptimo de cuadrillas con el guiado hacia zonas críticas de la red para no incurrir en costos de falla excesivos para la sociedad.

Como comentario final a la estrategia implementada en el Caso Base, resulta ser la que transa de mejor forma los tiempos de restauración total junto con la energía no suministrada total. El hecho de hacer zonificación y clusterización produce tiempos de viajes comparables a los obtenidos por una estrategia enfocada únicamente en el ruteo óptimo de cuadrillas. Por otro lado, la priorización de transformadores logra reducir enormemente los costos de falla comparados con la estrategia no guiada. Sin embargo, presenta costos sociales superiores en 38.58 millones de dólares con respecto a una estrategia de única priorización. Más adelante se verá cómo es posible lograr una restauración similar a la estrategia guiada total por otro tipo de medios, sin comprometer los tiempos de restauración de la red.

### 7.1.8 Análisis de resiliencia

El análisis de resiliencia se realiza por medio de la métrica de resiliencia estudiada en el Capítulo 3, donde se definen las variables  $T_U$  y  $T_D$ . Para el cálculo de resiliencia, se considera  $T_{U,h}$  y  $T_{D,h}$  como el porcentaje de carga suministrada y no suministrada en cierta hora. Luego se tiene que

$$R = \frac{T_U}{T} = \frac{\sum_h T_{U,h}}{\sum_h (T_{U,h} + T_{D,h})} \quad 7.2$$

El cálculo parece sencillo, pero la métrica presenta dificultades al momento de realizar comparaciones entre casos. En primer lugar, el autor de [2] afirma que una de las ventajas de la métrica es el hecho de tener un  $T$  móvil y relativo al caso de estudio. Supongamos que  $T$  se fija al momento donde se logró un 100% de la restauración, analicemos los siguientes casos de porcentaje de falla horaria:

- Caso A: {100, 90, 10, 0}
- Caso B: {100, 90, 10, 10, 0}

Si escogemos un  $T_{100}$  para cada caso, es decir  $T_{100}^A = 4$  horas y  $T_{100}^B = 6$  horas, entonces el nivel de resiliencia de los sistemas es  $R_{100}^A = 50\%$  y  $R_{100}^B = 63\%$ , es decir, el Caso B resulto ser más resiliente al Caso A aun teniendo tiempos de restauración y carga no suministrada mayores. Si fijamos  $T_{100}$  al máximo tiempo de restauración de ambos casos, es decir,  $T_{100} = 6$  horas, entonces este resulta que  $R_{100}^A = 66\%$  y  $R_{100}^B = 63\%$ , entregando un resultado coherente. A pesar de ello, el hecho de fijar  $T$  como el máximo de los casos a estudiar implica los siguientes inconvenientes:

- **Nivel de resiliencia relativo:** El nivel de resiliencia depende de los casos a estudiar y por lo tanto son relativos a sus tiempos. El nivel de resiliencia de un set de casos podría variar entre 90% y 80% y para otro set de casos variar entre los 50% y 60%. La consecuencia final de esto es la incerteza de saber qué nivel de resiliencia es un buen nivel de resiliencia, relegando la métrica únicamente a comparar casos y no a dar señales de desempeño de los sistemas estudiados.
- **Enfoque en el porcentaje de carga total:** La métrica no da cuenta del progreso de la curva de carga y solo se enfocada en el porcentaje de carga total no suministrada en el horizonte de evaluación. De hecho, permutar cualquiera de los niveles de carga suministrada horaria da un mismo nivel de resiliencia, así como concentrar las cargas no suministradas en cada hora. Este problema no considera la profundidad de falla horaria ni los costos sociales y económicos que implican.

Previstos los posibles problemas que podrían darse al comparar casos, se utiliza una métrica de resiliencia del 98% de carga suministrada, denominada  $R_{98}$  y se usa un tiempo de 538 horas, la cual corresponde a la hora donde se alcanza el 98% de la carga para el caso de CE\_40 estudiado más adelante. La elección de  $T_{98}$  se debe a la diferencia de tiempos entre el caso C\_NC y el resto de los casos, lo que provocaría niveles de resiliencia muy elevados para una comparación adecuada.

Tabla 7.8 Nivel de resiliencia de las estrategias estudiadas

	C_B	C_NC	C_NG	C_RZ	C_GZ	Z_GT
<b>R<sub>98</sub> [%]</b>	93,97	92,06	90,81	87,97	94,42	95,44

Los resultados de la Tabla 7.8 muestran que los casos No Cooperativo, No Guiado y Aleatorio Zonal poseen un nivel de resiliencia inferior al Caso Base. Por otro lado, los casos Guiados Zonales y Totales presentan niveles de resiliencia superiores a él. Notemos que el Caso Guiado Zonal presenta un nivel de resiliencia superior al Caso Base aun cuando este resultado tener un tiempo de restauración parcial inferior.

Nuevamente, esto se debe a que la métrica de resiliencia solo considera niveles de carga levantados efectivamente, esto es, convierte todos los casos a uno donde toda la carga no suministrada se concentra en las primeras horas. La Tabla 7.9 muestra el equivalente en días de carga no suministrada para cada estrategia estudiada.

Tabla 7.9 Días sin suministro eléctrico equivalente para los casos de estudio

	C_B	C_NC	C_NG	C_RZ	C_GZ	Z_GT
<b>T<sub>D</sub> [días]</b>	1,35	1,77	2,06	2,89	1,25	1,02

La métrica compara los días equivalentes sin suministro eléctrico con el horizonte de evaluación y entrega un resultado. Cuando el horizonte es fijo para cada caso, esto es equivalente a evaluar la energía no suministrada total de cada caso. Cuando el horizonte no es fijo, la métrica entrega malas evaluaciones cuando se da una restauración con pendientes de energía suministrada pronunciadas a las pocas horas, ya que considera que la razón entre días equivalentes con suministro al 100% y días evaluados es pequeño, independiente de la duración de la restauración. Por esta razón, la métrica se utilizará para comparar el desempeño del levantamiento de carga equivalente en el resto de las secciones.

## 7.2 Estados de preparación de la red

En esta sección se analizan los resultados para diferentes estados de preparación de la red bajo la estrategia utilizada en el Caso Base. El estado de preparación de la red dependerá del tipo de estructura y número de cuadrillas a disposición de la distribuidora.

Primero se analiza el estado de la red con estructura no-anclada después del evento BPAI y se compara con el su homólogo para un sistema anclado. Luego se comparan diferentes estados de preparación de la red con 4 centros de operaciones y se buscan señales para localizar un quinto centro de operaciones. Emplazado el quinto centro de operaciones, se compara el desempeño del Caso Base con su homólogo de 5 centros, es decir, una red anclada con 20 cuadrillas por zona y se hace un análisis similar al realizado con 4 centros de operaciones. Finalmente, se busca una relación entre parámetros globales y número de centros de cuadrillas, agregando cuatro casos extremos con 40 y 400 cuadrillas en la red.

### 7.2.1 Estado de la red no-anclada después del evento BPAI

La Tabla 7.10 resume el estado en el que queda una red no-anclada ante el evento BPAI, del cual se puede extraer lo siguiente:

- Las Zonas 1 y 2 presentan la mayor cantidad de transformadores dañados, así como la mayor cantidad de transformadores con daños Completos y Extensivos.
- Las Zonas 3 y 4 presentan en su mayoría daños Moderados y Extensivos.
- A nivel global, la red presenta daños mayormente Extensivos, seguido por daños moderados.

*Tabla 7.10 Número de transformadores dañados por el sismo en una red no-anclada*

Zona\Daño	Menor	Moderado	Extensivo	Completo	TOTAL
Zona 1	320	1.460	3.027	2.310	7.117
Zona 2	9	746	4.246	1.456	6.457
Zona 3	46	1.791	1.425	51	3.313
Zona 4	243	3.071	1.216	73	4.603
<b>TOTAL</b>	618	7.068	9.914	3.890	

Con respecto a los daños percibidos por la red con infraestructura anclada, los transformadores que anteriormente sufrían daños Moderados y Menores ahora sufren daños Extensivos y Completos, aumentando en cerca de 2500 transformadores para cada uno de ellos. Por lo anterior, se prevé una profundidad de falla mayor a la de los sistemas anclados, así como tiempos de restauración mayores. Además, el tiempo teórico de restauración de la red es de 17,41 días.

### 7.2.2 Cuatro centros de operaciones

Los resultados para este set de casos se encuentran en la Tabla 7.11, de lo que se desprende lo siguiente:

- Las distancias recorridas no cambian significativamente entre casos con igual tipo de estructura. La disminución de distancias recorridas al mejorar la estructura de la red se debe a dos factores:
  - Menor número de transformadores a reparar, lo que tiene un impacto directo en las distancias a recorrer.
  - Menor número de transformadores con estado de daño Extensivo y Completo, lo que implica menos horas de reparación en un único nodo y por lo tanto, menos horas de viaje de descanso y vuelta al trabajo.
- En general, el resto de los parámetros baja a medida que aumenta el número de cuadrillas y a medida que mejora la infraestructura de la red.

- Una red con infraestructura estándar es capaz de llegar a tiempos de restauración similares a la de una red con infraestructura reforzada aumentando su número de cuadrillas, como se aprecia en los casos CE\_148 y CS\_100. Sin embargo, un mayor esfuerzo se requiere para disminuir la energía no suministrada total, así como el costo de falla. Ello debido a la profundidad de falla inicial de un sistema no-anclado, muy superior al de un sistema anclado. Este hecho se puede apreciar en la Figura 7.18, donde en la hora cero, la energía no suministrada ya es mayor a los casos con redes ancladas.

Tabla 7.11 Resultados globales para los diferentes estados de preparación de la red con 4 centros

	Tiempo 100% [días]	Tiempo 98% [días]	Energía NS [GWh]	Costo Falla [Mill USD]	Distancia Rec. [km]
<b>CE_100</b>	18,51	13,91	95,00	340,28	60.509
<b>CS_100</b>	13,48	10,79	45,28	138,44	49.142
<b>CE_148</b>	12,49	9,95	65,53	237,96	60.476
<b>CS_148</b>	8,87	7,06	31,00	97,30	47.589

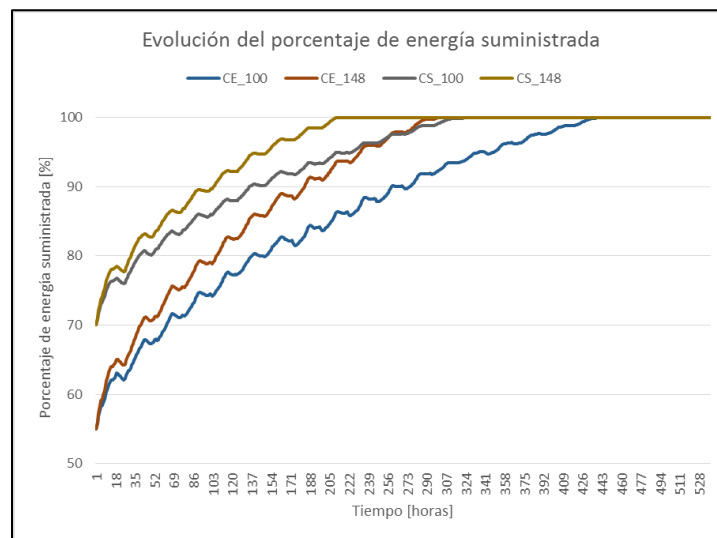


Figura 7.18 Curva del porcentaje de carga suministrada para los casos con 4 centros de operaciones

Un aspecto destacable en todos los casos es que se mantienen relativamente cercanos a sus tiempos teóricos de restauración, siendo un resultado característico de la estrategia desarrollada a lo largo del trabajo de memoria.

Tabla 7.12 Comparación de los tiempos de restauración total con su cota inferior

	CE_100	CS_100	CE_148	CS_148
$T_{teo}^C$ [días]	17,41	12,43	12,02	8,35

<b>Dif. Tiempo 100% [días]</b>	1,10	1,05	0,47	0,52
------------------------------------	------	------	------	------

Ahora que se conoce el desempeño de la restauración de la red con los centros a máxima capacidad, se propone colocar un nuevo centro de operaciones. Para ello, es necesario conocer el comportamiento de las cuadrillas de distribución en todos los casos, tal que la nueva localización pueda ser útil para todos los casos ya estudiados.

De la sección anterior, se mostró en base a resultados cualitativos que las zonas de mayor saturación pertenecían a las Zonas 1 y 2 de la red. La Tabla 7.13, que representa el porcentaje de atención de las cuadrillas a diferentes zonas, obtenida como un promedio de atención de todos los casos, pues la distribución de cuadrillas se mantiene en cada uno de ellos. En el Anexo C se pueden consultar las distribuciones de cuadrillas por zona para todos los casos estudiados en esta sección.

El resultado indica que las cuadrillas de la Zona 3 y 5 logran atender cerca del 98% de su zona, y atienden entre el 13% y 14% de las Zonas 1 y 2. Por otro lado, Las cuadrillas de las Zonas 1 y 2 no logran atender la totalidad de los transformadores de su zona. Un hecho curioso es que dichas zonas cooperan en menor medida unas con otras, mostrando la falta de personal en sus respectivos centros de operaciones.

*Tabla 7.13 Porcentaje de transformadores atendidos en cada zona por todas las cuadrillas de un centro*

		ZONA1	ZONA2	ZONA3	ZONA4
Promedio [%]	CUADRILLAS Z1	62,30	4,65	0,37	0,23
	CUADRILLAS Z2	4,76	67,65	0,30	0,24
	CUADRILLAS Z3	18,74	14,36	98,15	0,36
	CUADRILLAS Z4	14,2	13,34	1,18	99,19

A modo de complemento, La Figura 7.19 (a) grafica la cantidad de energía no suministrada por todos los transformadores atendidos por las cuadrillas de un determinado centro, mientras que Figura 7.19 (b) indica las distancias totales recorridas por las cuadrillas de cada centro. Los resultados indican lo siguiente:

- La Zona 2 posee la mayor cantidad de energía no suministrada y a su vez posee la menor cantidad de distancia recorrida. Esto implica que es la zona con mayor fuga de energía suministrada. Las distancias recorridas indican que las cuadrillas de su centro de operaciones están relegadas únicamente a recuperar el servicio de su zona.
- La Zona 1 es la siguiente en poseer mayor fuga de energía suministrada, así como también de poseer los tiempos recorridos más largos. La Tabla 7.13 muestra que sus cuadrillas se encuentran relegadas a reparar únicamente sus zonas, por lo que se concluye que los altos tiempos de viaje se deben principalmente a que la Zona 1 también posee una cantidad no menor de transformadores alejados de su centro de operaciones, lo cual se puede visualizar en la Figura 7.20.
- Las Zonas 3 y 4 presentan la menor fuga de energía en todos los casos estudiados. Otro aspecto a destacar es que la energía no suministrada graficada corresponde

también a la aportada por transformadores de otras zonas, es decir, gracias a la cooperación de cuadrillas. El hecho de que la energía no suministrada de los transformadores atendidos por las cuadrillas de las Zonas 3 y 4 sea tan baja implica que estas apoyan de forma eficiente a las Zonas 1 y 2.

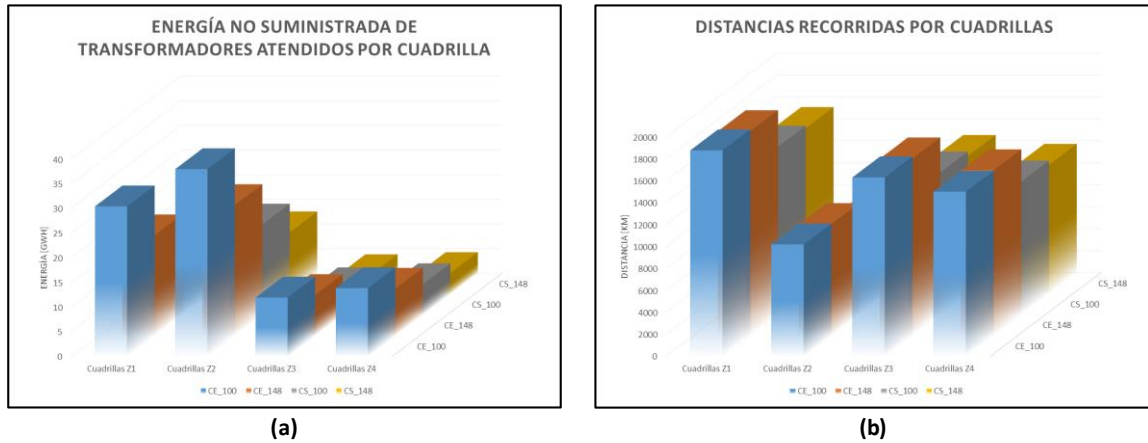


Figura 7.19 Energía no suministrada de transformadores atendidos (a) y distancias recorridas (b) por las cuadrillas de una determinada zona

### 7.2.3 Elección del quinto centro de operaciones

En base a los resultados anteriores se busca localizar el nuevo centro de operaciones con la finalidad de disminuir la carga de trabajo de las Zonas 1 y 2. Esta sección tiene como objetivo mostrar que no solo es posible mejorar el rendimiento de la restauración con un nuevo centro, sino también mostrar la importancia de su ubicación. Al inicio de este trabajo, y siguiendo resultados de trabajos similares [17], resultaba atractivo colocar una nueva zona por sobre la Zona 1 ya que se apreciaban viajes de largas distancias desde los transformadores de la zona norte de la red hacia el resto de los centros de operaciones.

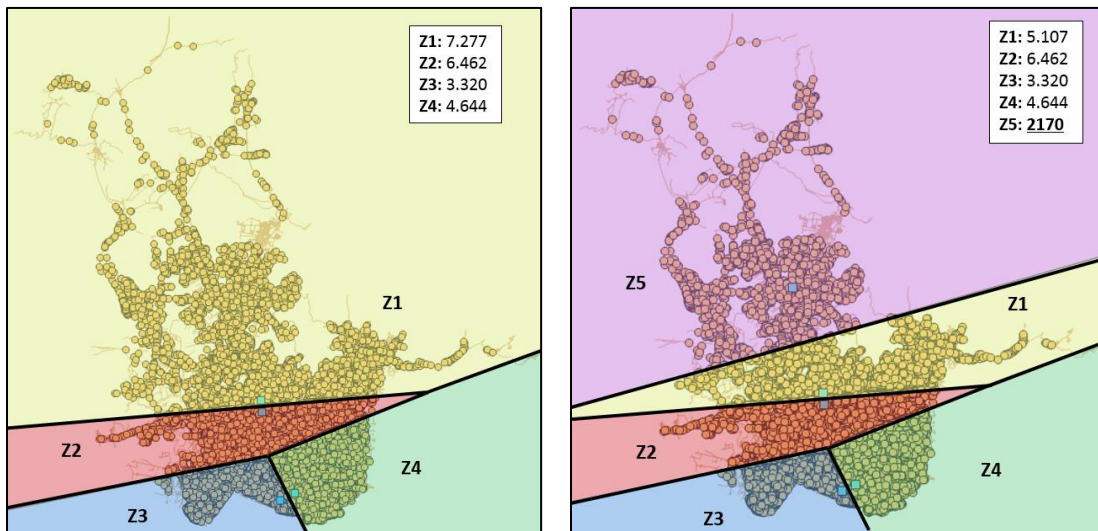


Figura 7.20 Zonas de la red con 4 centros (izquierda) y primera localización del quinto centro (derecha)

La Figura 7.20 muestra la división de la red por zonas así como el número de transformadores asociado a cada una de ellas antes y después de colocar el quinto centro de operaciones



siguiendo dicha suposición inicial. La posición del nuevo centro adquiere transformadores únicamente de la Zona 1 y crea una zona con menor cantidad de transformadores al resto. En definitiva, este criterio no aportara a des-saturar las Zonas 1 y 2.

El criterio elegido consiste en balancear el número de transformadores entre la Zona 1, la Zona 2 y la nueva zona, la que llamaremos Zona 5 y se basa en un criterio de proximidad de distancia euclidiana (Formación de nuevos polígonos de Voronoi). Bajo este criterio, se siguió el procedimiento de bajar el centro de operaciones y probar localizaciones en sentido anti-horario del reloj, procedimiento que se ilustra en la Figura 7.21. Un centro de operaciones demasiado cerca del centro de la Zona 1 adquiere transformadores de la Zona 1 y 2, pero genera un desbalance entre ellos, mientras que un centro muy cercano al centro de la Zona 2 adquiere transformadores de la Zona 1, Zona 2 y Zona 3, generando un desbalance entre ambas zonas.

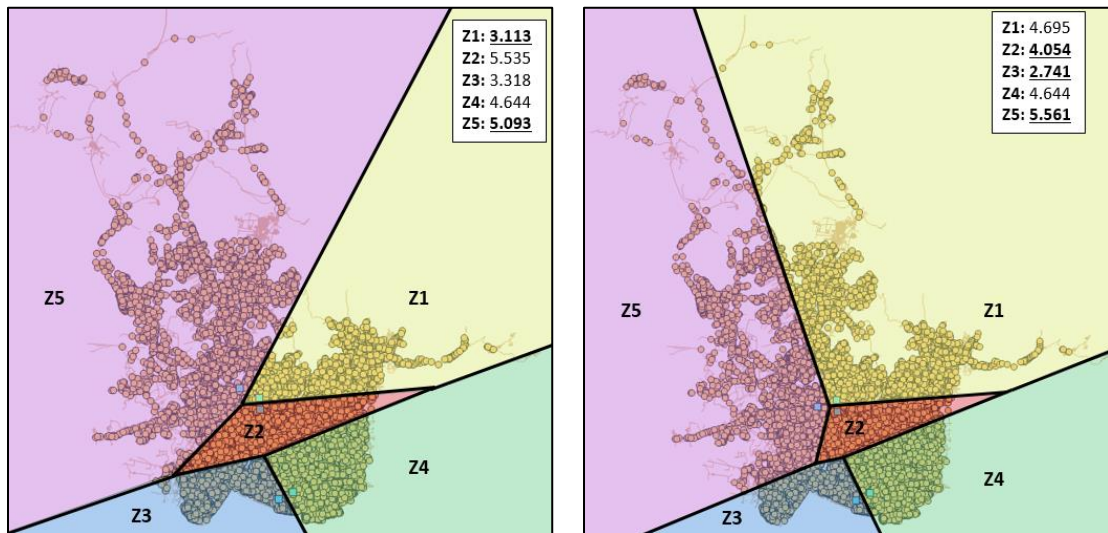


Figura 7.21 Cambio de localización del quinto centro de operaciones

En base a prueba y error se decidió alejar el quinto centro de operaciones de los centros de operaciones de las Zonas 1 y 2 y colocarlo a la altura de la Zona 1, lo que logra generar un balance adecuado entre ellas. Sin embargo, la Zona 5 también adquiere transformadores de la Zona 3, ya que subir el centro provoca un desbalance en la Zona 1, esto dado que el criterio de distancias se mantiene inalterado

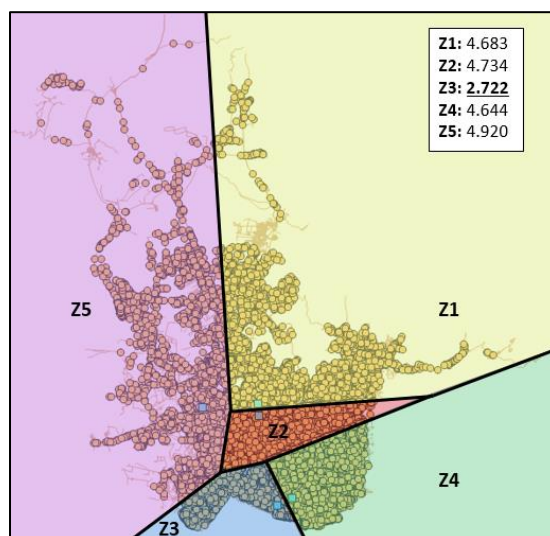


Figura 7.22 Posición final del quinto centro de operaciones

Los resultados de la restauración de servicio con esta nueva disposición de centros se disponen en la siguiente sección.

#### 7.2.4 Cinco centros de operaciones

Para demostrar que el nuevo centro de operaciones logra des-saturar las zonas, se compara el Caso Base con su homólogo de 5 centros de operaciones. Los resultados se observan en la Tabla 7.14, donde la nueva inserción mejora todos los parámetros con respecto al Caso Base. La mejora más destacable resulta ser el de la energía no suministrada total, donde se logró disminuir la energía no suministrada en 2,24 [GWh], generado un ahorro referencial (bajo los supuestos ya mencionados) de 7,94 millones de dólares para la sociedad, logrado únicamente por colocar un nuevo centro de operaciones y manteniendo la cantidad de personal disponible.

Tabla 7.14 Resultados globales para los casos con 4 y 5 centros de operaciones e igual número de cuadrillas

	Tiempo 100% [días]	Tiempo 98% [días]	Energía NS [GWh]	Costo Falla [Mill USD]	Distancia Rec. [km]
<b>CS_100</b>	13,48	10,79	45,28	138,44	49.142
<b>CS_100Z5</b>	13,45	10,33	43,04	130,50	46.105
<b>Diferencia</b>	-0,03	-0,46	-2,24	-7,94	-3.037

La curva de la Figura 7.23 muestra como el porcentaje del caso CS\_100Z5 se mantiene por debajo del Caso Base, lo que se debe principalmente a que el nuevo centro de operaciones logra clusterizar de mejor forma la red de transformadores. Solo con la Zona 1 y 2, grandes focos de energía no suministrada son cubiertos por pocos clusters, mientras que con la Zona 5, las Zonas 1 y 2 concentran sus esfuerzos en dichos focos dividiéndolos en más partes dada su reducción de área.

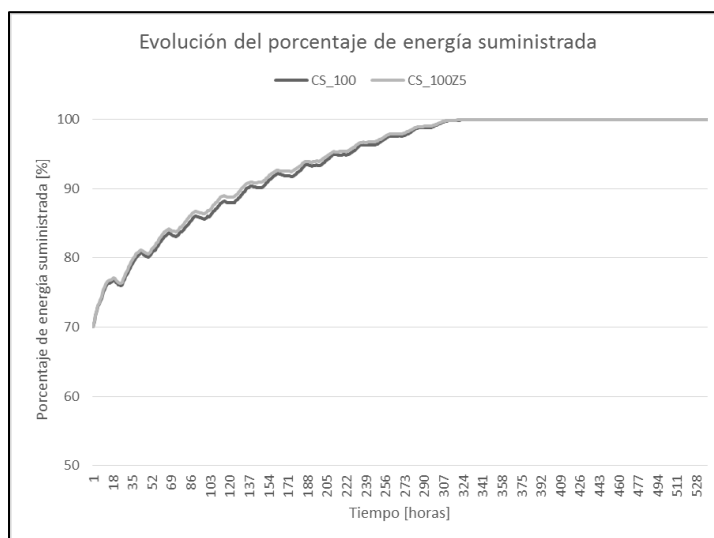


Figura 7.23 Curva del porcentaje de carga suministrada para los casos CS\_100 y CS\_100Z5

Los resultados con 5 centros de operaciones para diferentes estados de preparación de la red se muestran en la Tabla 7.15. Una observación radica en la disminución de la distancia recorrida con respecto a los valores obtenidos para 4 centros de operaciones, lo que se atribuye al quinto centro de operaciones de su zona. El comportamiento cualitativo del resto de los parámetros es similar al caso con 4 centros.

Tabla 7.15 Resultados globales para diferentes estados de preparación con 5 centros

	Tiempo 100% [días]	Tiempo 98% [días]	Energía NS [GWh]	Costo Falla [Mill USD]	Distancia Rec. [km]
<b>CE_125</b>	14,69	10,99	73,66	132,08	57.216
<b>CS_125</b>	10,63	8,07	34,73	53,18	45.799
<b>CE_185</b>	9,71	8,00	50,21	90,40	55.972
<b>CS_185</b>	7,27	5,57	23,68	36,78	45.369

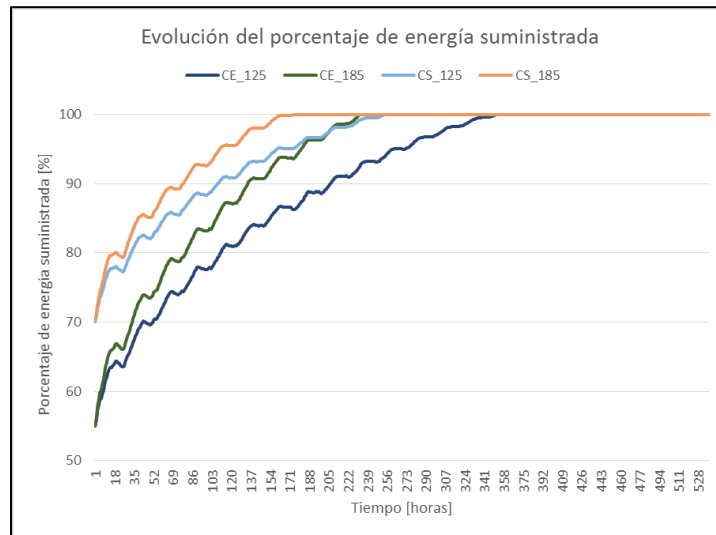


Figura 7.24 Curva del porcentaje de carga suministrada para los casos con cinco centros de operaciones

### 7.2.5 Evolución de parámetros globales

Con los datos de parámetros globales obtenidos, se procede a buscar una relación entre ellos y el número de cuadrillas en la red. Al set de resultados se le suman 2 casos extremos por tipo de estructura donde la red cuenta con una cantidad de 40 y 400 cuadrillas distribuidas en 4 centros de operaciones, es decir, son sensibilidades para la metodología de la estrategia propuesta en el caso base.

Las relaciones obtenidas son todas hipérbolas de tipo  $y = A \cdot x^{-c}$ , lo que da indicios de que una disminución marginal de alguno de los parámetros globales requerirá costos cada vez más elevados.

Las Figura 7.26 y Figura 7.26 muestran la evolución de los tiempos de reposición total y parcial por número de la red. El índice de determinación del tiempo de reposición total es casi del 100% para la ecuación determinada. Tal como se ha mencionado anteriormente, el tiempo de reposición total solo depende del tipo de infraestructura de la red y del número de cuadrillas, siendo afectado de forma insignificante por otros factores como el número de centro de operaciones en la red. Por otro lado, el tiempo de reposición parcial al estar ligado más con el levantamiento eficiente de suministro eléctrico, si se ve afectado por el número y localización de nuevos centros de operaciones, cosa que se puede apreciar en la disminución del índice de determinación así como descalces con la curva entre los puntos de 4 y 5 centros de operaciones.

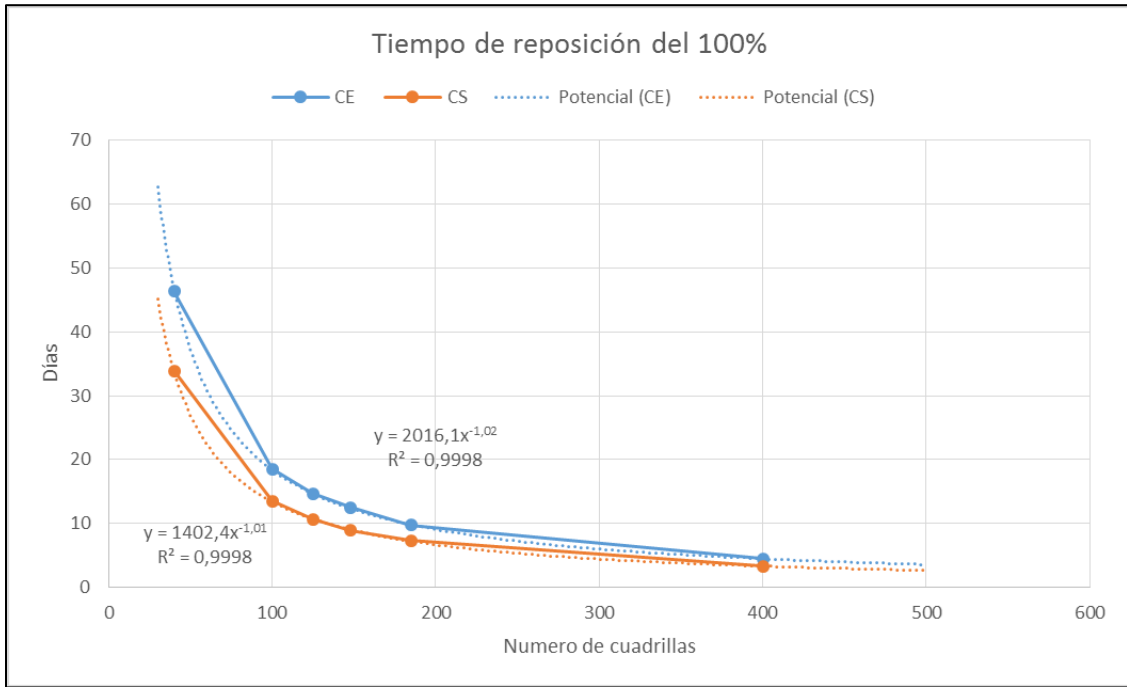


Figura 7.25 Curva de tiempo de reposición total vs cuadrillas en la red

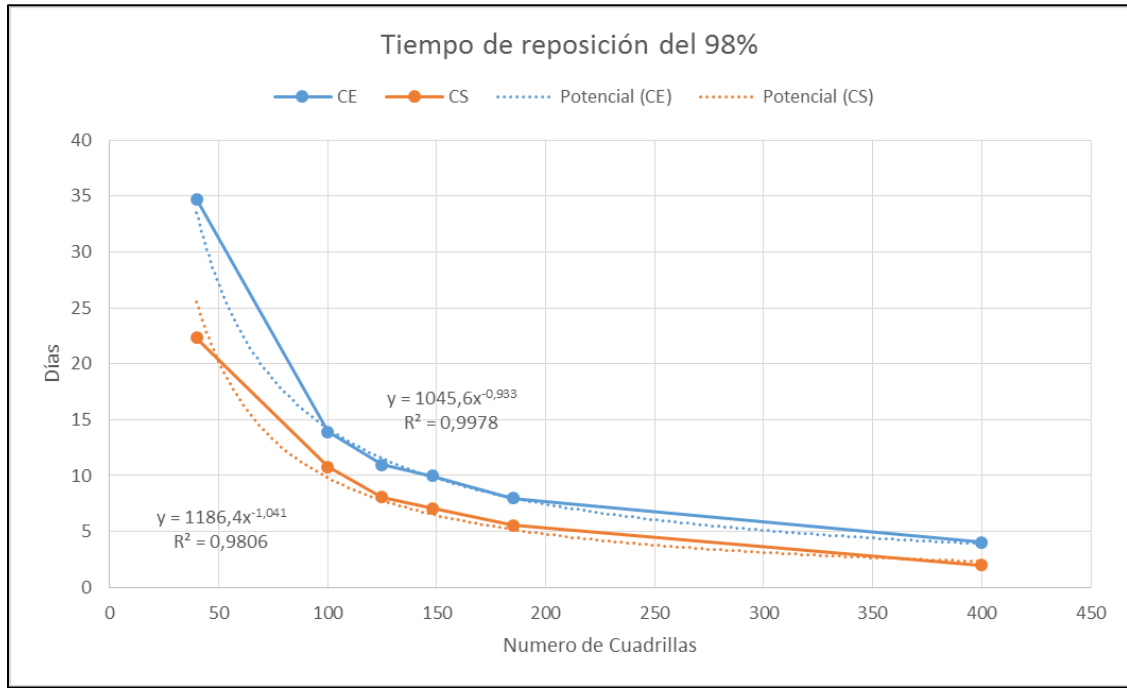


Figura 7.26 Curvas de tiempo de reposición parcial vs cuadrillas en la red

El comportamiento de los tiempos de reposición con respecto al número de cuadrillas disponibles da cuenta de que hay ciertas metas que la distribuidora no serán capaces de alcanzar a menos que desembolsen una cantidad prohibitiva de dinero. Si por ejemplo, se dictara que el sistema deba estar levantado al 100% en un tiempo máximo de 5 días frente un

sismo de igual características al utilizado, la distribuidora en el Caso Base requeriría de 166 nuevas cuadrillas de restauración, lo que implicaría la instalación de 4 nuevos centros de operaciones a máxima capacidad, y esto suponiendo que la distribuidora es capaz de ejecutar la estrategia propuesta en el presente trabajo de título.

Un caso similar se presenta en las curvas de energía no suministrada y costo de falla total en la Figura 7.27 y Figura 7.28. Del Caso Base al Caso CS\_185 (5 centro de operaciones y 185 cuadrillas) la distribuidora logra ahorrar en términos de costo social estimado, un total de 32,44 millones de dólares instalando un nuevo centro de operaciones y contratando 85 cuadrillas. Sin embargo, desde este caso (CS\_185) hasta el caso con 400 cuadrillas, es decir, contratando 215 cuadrillas solo se logran ahorrar 20,81 millones de dólares. Si bien este análisis no contempla la localización óptima de nuevos centros de operaciones, los cuales afectarían positivamente el monto final en los costos de falla y energía no suministrada, estos no afectarían mayormente la forma de la curva inversa en si, como se puede ver en los casos de 100 cuadrillas (4 centros) junto con los de 185 (5 centros). En definitiva, hay costos que no pueden permitirse la distribuidora para prepararse contra un evento que no se sabe a ciencia a cierta cuando sucederá.

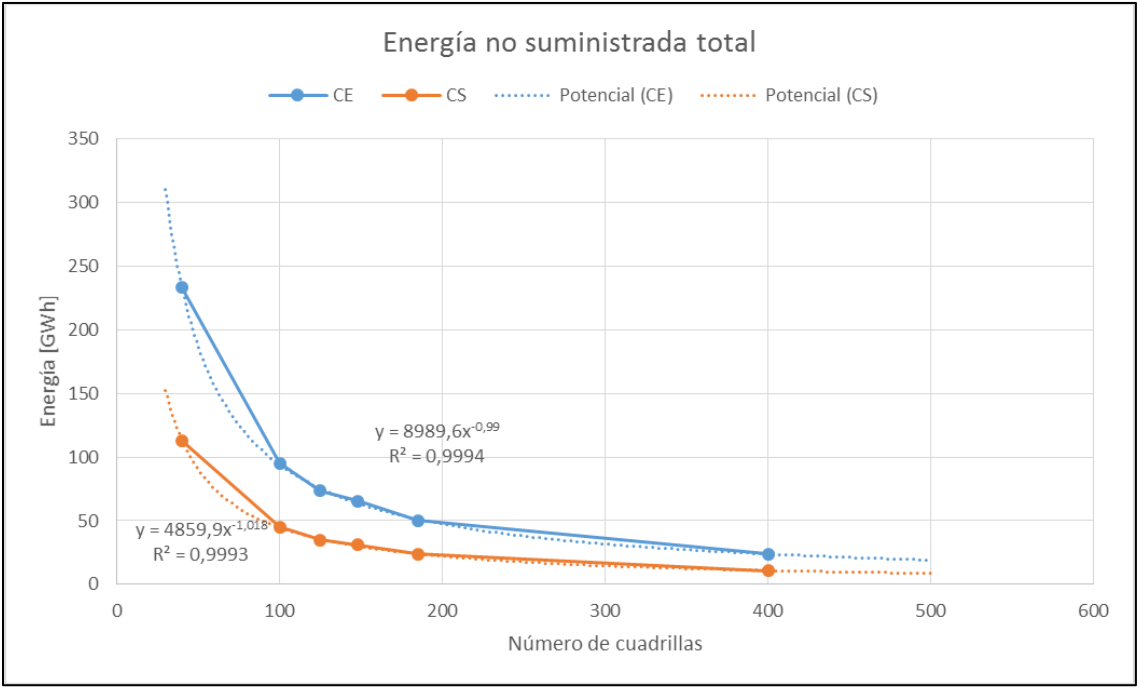


Figura 7.27 Curva de energía no suministrada vs cuadrillas en la red

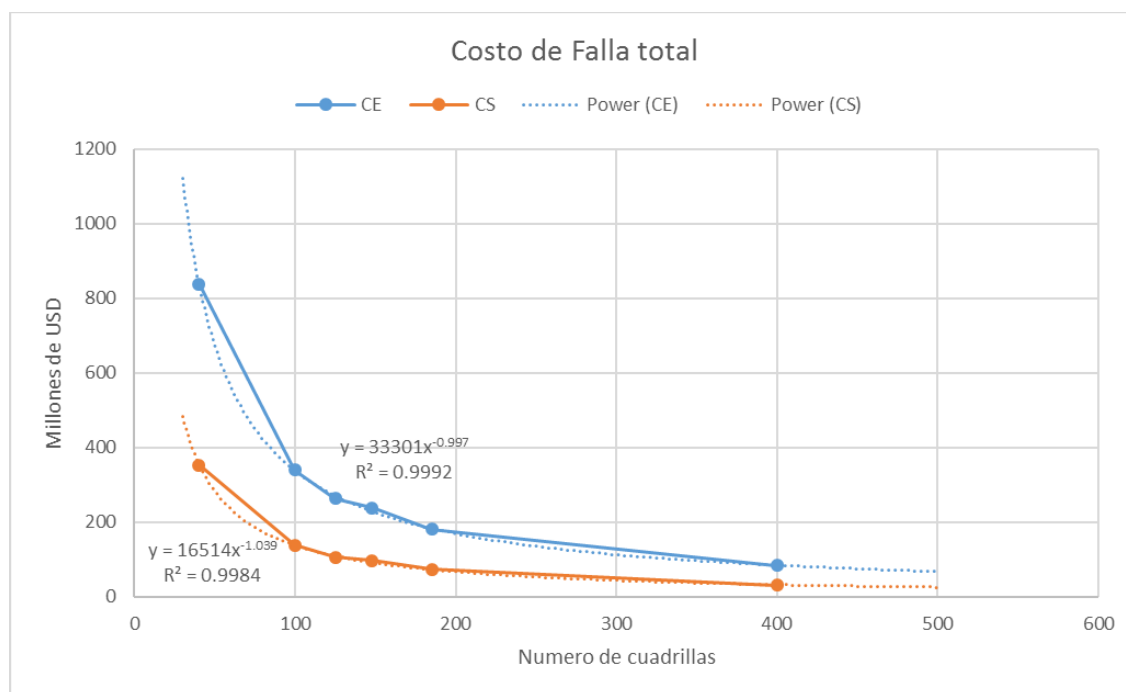


Figura 7.28 Curva de costos de falla vs cuadrillas en la red

A modo de resumen se listan las ideas principales extraídas de la sección

- Las distancias recorridas disminuyen gracias al nuevo centro de operaciones, pero siguen manteniéndose relativamente constantes bajo una misma infraestructura sísmica.
- Los tiempos de reposición, la energía no suministrada y el costo de falla presentan curvas de tipo hipérbola con respecto al número de cuadrillas, lo que implica una mejora marginal de cada una de ellas a medida que aumenta las cuadrillas en la red.
- Frente a lo anterior, se deben exigir metas realistas la distribuidora con respecto a su desempeño ante eventos BPAI de tipo sísmicos.

### 7.2.6 Comparación de casos por nivel de resiliencia

En esta sección se evalúan las estrategias analizadas anteriormente con los diferentes estados de preparación de la red por medio de la métrica de resiliencia escogida.

En primer lugar, de la Tabla 7.16 se observa como las estrategias Sin Cooperación, No Guiado y Aleatorio Zonal presentan niveles de resiliencia comparables a los obtenidos por una red sin componentes sísmicas. Ello da cuenta de que estas estrategias no logran aprovechar el nivel de infraestructura instalado en la red, siendo el peor de ellos el Caso Aleatorio Zonal, lo que refuerza nuevamente la necesidad de que existan protocolos de acción de cuadrillas ante eventos BPAI.

Por otro lado, los casos Guiados Zonal y Total superan a los Casos Base (CS\_100) y es comparable al caso de 125 cuadrillas. Este resultado es importante ya que, sabiendo que la métrica de resiliencia propuesta da más énfasis al levantamiento de carga que a los tiempos de restauración, un aumento de 25 cuadrillas (6-7 cuadrillas por centro) con respecto al caso

base, sin necesidad de un nuevo centro de operaciones, logra un levantamiento de carga similar al Caso Guiado Total sin comprometer los tiempos de restauración total de la red.

*Tabla 7.16 Nivel de resiliencia de todos los casos estudiados*

Estrategias		Componente Estándar		Componente Sísmica	
Casos	$R_{98}$ [%]	Casos	$R_{98}$ [%]	Casos	$R_{98}$ [%]
C_NC	92,06	CE_100	87,24	CS_100	93,97
C_NG	90,81	CE_125	90,17	CS_100Z5	94,27
C_RZ	87,97	CE_148	91,30	CS_125	95,39
C_GZ	94,42	CE_185	93,36	CS_148	95,90
C_GT	95,44	–	–	CS_185	96,90

Por último, cabe destacar de los resultados de resiliencia que recién con un número de 185 cuadrillas, el caso con componente estándar es capaz de compararse al caso con componente sísmica de solo 100 cuadrillas, es decir, sin infraestructura sísmica se requerirían de 185 cuadrillas para lograr levantamiento de carga similar al de una red con infraestructura sísmicas y solo 100 cuadrillas.



## 8. Conclusiones

---

### 8.1 Conclusiones Generales

Se realizó una revisión bibliográfica sobre tópicos que conciernen al concepto de resiliencia y se escogió una métrica adecuada para el trabajo de título, haciendo un análisis crítico del mismo y detectando sus posibles falencias.

Se recopilaron antecedentes de eventos BPAI en la red estudiada y se escogió el evento de tipo sísmico para modelar el estado de daño de la red a estudiar junto con la modelación del sismo, el daño provocado en los puntos en la red y la restauración de las mismas.

Se propuso una estrategia de ruteo teniendo como foco tanto la restauración de transformadores prioritarios en el suministro de energía de la red como la minimización de tiempos de espera de cada nodo a atender. La estrategia consta de las siguientes etapas: Zonificar la red por cercanía de nodos a centros de operaciones, clusterizar cada zona para que las cuadrillas atiendan transformadores cercanos entre sí, priorizar transformadores y minimizar tiempos de espera y finalmente cooperar con cuadrillas que aún no han terminado sus labores. La estrategia fue implementada en un Modelo computacional para modelar el estado de daño del sistema y simular el despacho de cuadrillas dada una estrategia determinada.

Se recopiló información sobre programas computacionales necesarios para implementar el Modelo de simulación, tanto a nivel de pre-procesamiento de datos como a nivel de ejecución, lo que dio como resultado un programa de alta eficiencia y rapidez.

Se comparó la estrategia de ruteo propuesta con diferentes estrategias que podría adoptar la distribuidora, encontrando relaciones inversas entre la energía no suministrada total y los tiempos de reposición total.

Se evaluaron diferentes estados de la red con 4 centros de operaciones para distinta disponibilidad de recursos (estructural y por cantidad de personal), lo que permitió decidir la localización de un nuevo centro de operación. Con los resultados de cinco centros de operaciones, se obtuvo la curva de parámetros en función de número de cuadrillas.

Finalmente se realizó un análisis global de todos los casos estudiados por medio de la métrica de resiliencia escogida, lo que permitió hacer comparaciones y equivalencias entre casos.

### 8.2 Conclusiones sobre Aspectos Específicos del Trabajo de Título

A continuación se realizan conclusiones detalladas sobre resultados específicos del trabajo de título.

- **Comparación de estrategias**

El hecho de comparar estrategias no solo permite entender el comportamiento del algoritmo propuesto si no de dar a conocer la existencia de estrategias que son claramente inferiores a otras. El hecho de no existir cooperación entre cuadrillas hace que se excedan los tiempos de reposición por fallas concentradas en la red. Por otro lado, la ausencia de protocolos de acción produce el peor desempeño de todas las estrategias estudiadas. Sin duda, esta es la estrategia que ninguna distribuidora debiera usar.

Por otro lado, las estrategias que minimizan solo un aspecto de la restauración de la red, esto es, tiempos de restauración total o energía no suministrada, presentan falencias en el otro aspecto no tratado. La estrategia que solo trata el aspecto temporal posee dos veces la energía no suministrada a la estrategia energética, mientras que esta última tarda un 50% más en reparar la red a la estrategia temporal. Todo ello más las curvas estudiadas en los capítulos anteriores implican una relación inversa entre la cantidad de energía no suministrada y el tiempo de restauración de la red.

Por su parte, la estrategia propuesta logra nivelar adecuadamente estos dos aspectos, teniendo un incremento del 2% en los tiempos de restauración y de un 32% en energía no suministrada en comparación a sus respectivos casos. Otra ventaja de la estrategia propuesta radica en que al estar zonificada la red y al saber que existen zonas que logran reponerse antes que otras, se asegura que dichas zonas tendrán todas sus áreas operativas en algún momento de la restauración de servicio, esto es, comercio, industria y residencia, algo que no se puede asegurar en ninguno de los algoritmos de enfoque único.

- **Instalación de nuevo centro de operaciones**

Si bien la instalación de un nuevo centro de operaciones obedece a la necesidad de aumentar el personal del equipo de restauración de la distribuidora, que por consiguiente tendrá efectos positivos en la restauración de servicio, su localización debe resultar ser conveniente para aprovechar la distribución de nodos dañados a lo largo de la red.

Si bien se logró localizar el centro por medio de las tablas de atención de cuadrillas por zona, este solo corresponde a un único evento sísmico. Para otro evento sísmico, la distribución podría cambiar y con ello disminuir el rendimiento del nuevo centro, por lo que la localización del quinto centro requiere un estudio de resiliencia y optimización más detallado. De cualquier forma, el hecho de que el nuevo centro logre agrupar una gran cantidad de transformadores al oeste de la red y considerando que la gran mayoría de los epicentros se encuentra en ese lado del territorio chileno, se espera que la localización del nuevo centro logre cubrir adecuadamente esa clase de eventos sísmos.

- **Comportamiento de parámetros por cuadrilla**

El comportamiento de todos los parámetros a excepción de las distancias recorridas presenta una curva hipérbola o inversa. Esto implica que los beneficios de aumentar el nivel de resiliencia del sistema se van haciendo cada vez más pequeño en comparación con los costos que acarrea dicho aumento. Este comportamiento se repite para los dos tipos de estructuras (estándar y sísmica) en la red, además de que las curvas se van acercando a medida que aumenta el número de cuadrillas. Con respecto a las distancias recorridas, estos no cambian significativamente por número de cuadrillas y si lo hace por número de centros de operaciones, ya que más centros de operaciones implican más división de zonas y por lo tanto, viajes más cortos de trabajo y descanso.

- **Análisis de resiliencia**

Para realizar un correcto análisis de resiliencia, es necesario fijar el horizonte de tiempo para todos los casos a estudiar. Esto trae inconvenientes, pues el horizonte de tiempo cambia de acuerdo a cada caso y por lo tanto no indica que tan bueno es el desempeño de la red, relegándola únicamente a una métrica de comparación.

Cuando es fijado el horizonte de tiempo, la métrica se acerca más a un medidor de desempeño por levantamiento de carga, sin darle mayor peso a la componente temporal en la restauración de servicio. Bajo esta idea, es posible comparar todos los casos de estudio y realizar equivalencias entre cada uno de ellos.

Se concluye que las estrategias con deficiencias en la coordinación de cuadrillas resultan en un desaprovechamiento de la resistencia de la infraestructura instalada. Por otro lado, para lograr levantamientos similares a los casos con enfoque en carga no suministrada, solo es necesario aumentar en 7 cuadrillas cada centro de operaciones utilizando la estrategia propuesta, sin la necesidad de instalar un nuevo centro de operaciones ni comprometer tiempos de restauración.

### 8.3 Trabajos futuros

Por último, diversos desafíos nacen del trabajo de título, ya sea en lo que es el refinamiento de elementos actuales o la incorporación de nuevas funcionalidades al modelo. Los trabajos a futuro propuestos son:

- **Variación en la distribución de cuadrillas por centro:** Debido a limitaciones del modelo implementado, el número de cuadrillas de la red debe estar distribuido en partes iguales por centro. Sabiendo que las cuadrillas de las Zonas 3 y 4 son las que cooperan con las Zonas 1 y 2, hubiera sido interesante conocer el comportamiento de la restauración si se cargan más cuadrillas a las Zonas con deficiencia de personal. Se propone implementar un proceso que permita distribuir el número de cuadrillas a voluntad y realizar pruebas trasladando cuadrillas de las Zona 3 y 4 hacia las Zonas 1 y 2 para el Caso Base.
- **Justificar el nuevo centro bajo un análisis de resiliencia más completo:** En el presente trabajo de título solo se utilizó un evento BPAI para diferentes fines, entre ellos localizar un nuevo centro de operaciones, permitiendo mejorar el desempeño del levantamiento de la red. Se propone evaluar el desempeño del nuevo centro para diferentes tipos de sismos, ya sea por medio de GMPEs o Shakemaps disponibles.
- **Mejoramiento del Algoritmo Cooperativo:** Existe un gap de hasta 15 horas entre las cuadrillas con mayor y menor horas de trabajo que se va haciendo cada vez más importante a medida que los tiempos de restauración disminuyen. Se propone implementar un Algoritmo Cooperativo con  $N$  pasos de tipo greedy seguido de  $N$  pasos de tipo secuencial, con  $N$  el número de cuadrillas en la red, con la finalidad de obtener un algoritmo rápido (paso greedy) y que explore todas las posibles cuadrillas a cooperar (paso secuencial) en su última etapa.
- **Refinamiento en tiempos de restauración:** Un resultado interesante es el hecho de que el tiempo teórico de restauración del Caso Base haya sido de 12,43 días siendo que para el sismo del 27 de febrero del 2010 la distribuidora tardó solo 12 días en restaurar el servicio completamente. Ello podría implicar una sobreestimación de los tiempos de restauración de los transformadores por estado de daño, por lo que se propone re-estudiar dichos tiempos en base a un análisis más detallado de los tiempos de reparación ofrecidos por los estudios VAD. También cabe la posibilidad de que se hayan sobreestimado a la cantidad de transformadores dañados. En tal caso se recomienda hacer un análisis utilizando varios modelos de movimientos espectrales encontrados en la academia.

- **Incorporación de equipos de respaldo:** Un equipo de respaldo puede modelar sistemas de almacenamiento como baterías o generadores diésel o sistemas complejos como Smart Grids de emergencia. Se propone incorporar estos equipos al modelo, modificando la lista de prioridades en el Algoritmo Euclidiano y dar señales de la existencia de equipos de respaldo en la etapa de Procesamiento de Resultados del Modelo.
- **Realizar un análisis costo-eficiente más detallado:** En nuestro análisis costo-eficiente solo se compararon los costos de falla considerándolos como un costo referencial al costo en el que podría incurrir la sociedad. Un análisis más detallado también contemplaría los costos asociados a la instalación de nuevos centros de operaciones, aumento de cuadrillas en cada centro de operaciones, aumento de costo en la instalación de infraestructura anclada, costos asociados a los viajes de cada cuadrilla, etc.

## 9. Bibliografía

---

- [1] D. Henry y J. R.-M. Emmanuel, «On the Impacts of Power Outages during Hurricane Sandy-A Resilience-Based Analysis,» *Systems Engineering*, vol. 19, nº 1, pp. 59-75, 2016.
- [2] A. Kwasinski, Quantitative Model and Metrics of Electrical Grids' Resilience Evaluated at a Power Distribution Level, Department of Electrical and Computer Engineering, Swanson School of Engineering, University of Pittsburgh, 2015.
- [3] Los temas de fondo que dejó la caída de suministro en distribución, *Revista Electricidad*, agosto del 2017.
- [4] A. K. Tang y J. M. Eidiger, Chile Earthquake of 2010 Lifeline Performance, American Society of Civil Engineers, 2013.
- [5] Hacia un Chile resiliente frente a desastres: Una oportunidad, Comisión Nacional para la Resiliencia frente a Desastres de Origen Natural, 2016.
- [6] G. J. Estévez, A. N. Espinosa, R. P. Behnke, L. Lanuzza y N. Velázquez, Achieving Resilience at the Distribution Level, 2017.
- [7] A. Navarro-Espinosa, R. Moreno, T. Lagos, F. Ordoñez, R. Sacaan y H. Rudnick, Improving Distribution Network Resilience Against Earthquakes, 2017.
- [8] R. Bent, P. v. Hentenryck y C. Coffrin, Vehicle Routing for the Last Mile of Power System Restoration, Los Alamos National Laboratory, 2011.
- [9] Y. Tan, F. Qiu, A. K. Das, D. S. Kirschen, P. Arabshahi y J. Wang, Scheduling Post-Disaster Repairs in Electricity Distribution Networks, 2017.
- [10] Cómo lograr un sistema de distribución más resiliente, *Revista Electricidad*, ver septiembre, 2017.
- [11] CHILECTRA (Actual ENEL Distribución), Informe de Sostenibilidad 2010, Santiago, 2010.
- [12] H. Ritchie y M. Roser, «Natural Catastrophes,» Our World in Data, 2017. [En línea]. Available: <https://ourworldindata.org/natural-catastrophes>.
- [13] 20 Largest Earthquakes in the World, <https://earthquake.usgs.gov/earthquakes/browse/largest-world.php>.

- [14] Earthquake Lists, Maps, and Statistics, <https://earthquake.usgs.gov/earthquakes/browse/>.
- [15] Dirección Meteorológica de Chile, «Explorador Climático,» 10 2018. [En línea]. Available: <http://explorador.cr2.cl/>.
- [16] F. Labarca, «Así fueron nevazones de los últimos 46 años en Santiago,» *La Tercera*, 15 07 2017.
- [17] E. de-los-Ríos-Cruz, J. C. Araneda, A. Santander y A. Navarro, Resiliencia en el Sistema Eléctrico de Potencia de Chile, Modelación Simplificada de la Respuesta ante Terremotos del Sistema de Distribución, Centro de Energía, Universidad de Chile, 2017.
- [18] ENEL Distribución Chile, *Efectos del terremoto en el sistema de Chilectra S.A.*, Santiago, 2010.
- [19] G. Solomos, A. Pinto y S. Dimova, A Review of the Seismic Hazard Zonation in National Building Codes in the Context of Eurocode 8, JRC European Commission, 2008.
- [20] Federal Emergency Management Agency, *Designing for Earthquakes*, 2006.
- [21] S. GUPTA y I. D. GUPTA, The Prediction of Earthquake Peak Ground Acceleration in Koyna Region, India, 2004.
- [22] B. Idini, F. Rojas, S. Ruiz y C. César, «Ground motion prediction equations for the Chilean subduction zone,» *Bulletin of Earthquake Engineering*, vol. 15, n° 5, 2017.
- [23] Hazus®-MH MR5, Earthquake Loss Estimation Methodology, Department of Homeland Security, Federal Emergency Management Agency, 2001.
- [24] C. B. Worden, D. J. Wald, T. I. Allen, K. Lin, D. Garcia y G. Cua, «A revised ground-motion and intensity interpolation scheme for shakemap,» *Bulletin of the Seismological Society of America*, vol. 100, n° 6, pp. 3083-3096, 2010.
- [25] United States Geological Survey, «ShakeMap 3.5 Manual,» [En línea]. Available: [http://usgs.github.io/shakemap/manual3\\_5/index.html](http://usgs.github.io/shakemap/manual3_5/index.html).
- [26] Hazus®-MH 2.1, Multi-hazard Loss Estimation Methodology, Earthquake Model, Department of Homeland Security, Federal Emergency Management Agency.
- [27] B. Zhang, P. Dehghanian y M. Kezunovic, Optimal Allocation of PV Generation and Battery Storage for Enhanced Resilience, 2017.

- [28] M. Kazama y T. Noda, Damage statistics (Summary of the 2011 off the Pacific Coast of Tohoku Earthquake damage), Soils and Foundations, 2012.
- [29] Memoria Anual, Enel Distribución Chile, 2016.
- [30] Tarifas de Suministro Electrico Destinadas a Usuarios Residenciales, ENEL Distribucion, Noviembre del 2017.
- [31] Financial Results for the Second Quarter of Fiscal, Tohoku Electric Power CO., INC., 2017.
- [32] 従量電灯A (Plan del Medidor A), [http://www.tohoku-epco.co.jp/dprivate/menu/menu\\_lighta.html](http://www.tohoku-epco.co.jp/dprivate/menu/menu_lighta.html), 2017.
- [33] Organisation for Economic Co-Operation and Development, «Regional income per capita,» [En línea]. Available: <https://stats.oecd.org/>.
- [34] The Electric Power System -Japan-, CIGRE.
- [35] K. Rosen, Discrete Mathematics and Its Applications, 2012.
- [36] G. Laporte, The Traveling Salesman Problem, An overview of exact and approximate algorithms, Centre de recherche sur les transports, Université de Montréal, 1992.
- [37] D. Applegate, R. Bixby, V. Chvátal y W. Cook, «TSP Cuts Which Do Not Conform to the Template Paradigm,» *Computational Combinatorial Optimization*, vol. 2241, pp. 261-303, 2001.
- [38] University of Waterloo, «Benchmark TSP Information,» [En línea]. Available: <http://www.math.uwaterloo.ca/tsp/concorde/benchmarks/bench99.html>.
- [39] K. Helsgaun, An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic, Roskilde: Roskilde University, 1998.
- [40] S. Lin y B. W. Kernighan, «An Effective Heuristic Algorithm for the Traveling-Salesman Problem,» *Operations Research*, vol. 21, n° 2, pp. 498-516, 1973.
- [41] K. Helsgaun, An Effective Implementation of K-opt Moves for the Lin-Kernighan TSP Heuristic, Roskilde: Roskilde University, 2006.
- [42] P. Krolak, W. Felts y G. Marble, «A Man-Machine Approach Toward Solving the Traveling Salesman Problem,» *Communications of the ACM*, vol. 14, n° 5, pp. 327-334, 1971.

- [43] M. Junger, G. Reinelt y G. Rinaldi, *The Traveling Salesman Problem*, 1994.
- [44] S. U. Nogueira, C. Prins y R. W. Calvo, «An effective memetic algorithm for the cumulative capacitated vehicle routing problem,» *Computers & Operations Research*, vol. 37, nº 11, pp. 1877-1885, 2010.
- [45] J. Lysgaard y S. Wøhlk, «A Branch-and-Cut-and-Price Algorithm for the Cumulative Capacitated Vehicle Routing Problem,» *European Journal of Operational Research*, vol. 236, nº 3, pp. 800-810, 2014.
- [46] K. Helsgaun, *An Extension of the Lin-Kernighan-Helsgaun TSP Solver for Constrained Traveling Salesman and Vehicle Routing Problem*, Roskilde: Roskilde University, 2017.
- [47] A. Dobrin, *A Review of Properties and Variations of Voronoi Diagram*.
- [48] D. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.
- [49] Environmental Systems Research Institute, «What is a shapefile?,» [En línea]. Available: <http://desktop.arcgis.com/en/arcmap/10.3/manage-data/shapefiles/what-is-a-shapefile.htm>.
- [50] United State Geological Survey, «Earthquake Hazard Programs,» 12 03 2018. [En línea]. Available: [https://earthquake.usgs.gov/earthquakes/eventpage/official20100227063411530\\_30/shakemap/intensity](https://earthquake.usgs.gov/earthquakes/eventpage/official20100227063411530_30/shakemap/intensity).
- [51] N. Orai, K. Ichii y S. Ishii, «A trial of simple and easy health monitoring for power poles by using ‘ipod’,» *Science China Technological Sciences*, vol. 58, nº 4, p. 712–719, 2015.
- [52] E. Tapia-Hernández, «Tubular steel poles under lateral load patterns,» *Advanced Steel Construction*, vol. 12, nº 4, pp. 428-445, 2016.
- [53] J. Eidinger, «Fragility of Electric Power Distribution Systems,» de *16th World Conference on Earthquake Engineering*, Santiago, 2017.
- [54] Coordinador Electrico Nacional, «Reporte Medidas Historicas PRMTE,» [En línea]. Available: <https://medidas.coordinador.cl/index.php/reportes-medidas-publicas/>.
- [55] Comisión Nacional de Energía, «Estudio de Costos de Servicios Asociados al Suministro de Electricidad en Distribución - Memoria de Calculo,» [En línea]. Available: <https://www.cne.cl/tarificacion/electrica/valor-agregado-de-distribucion/>.

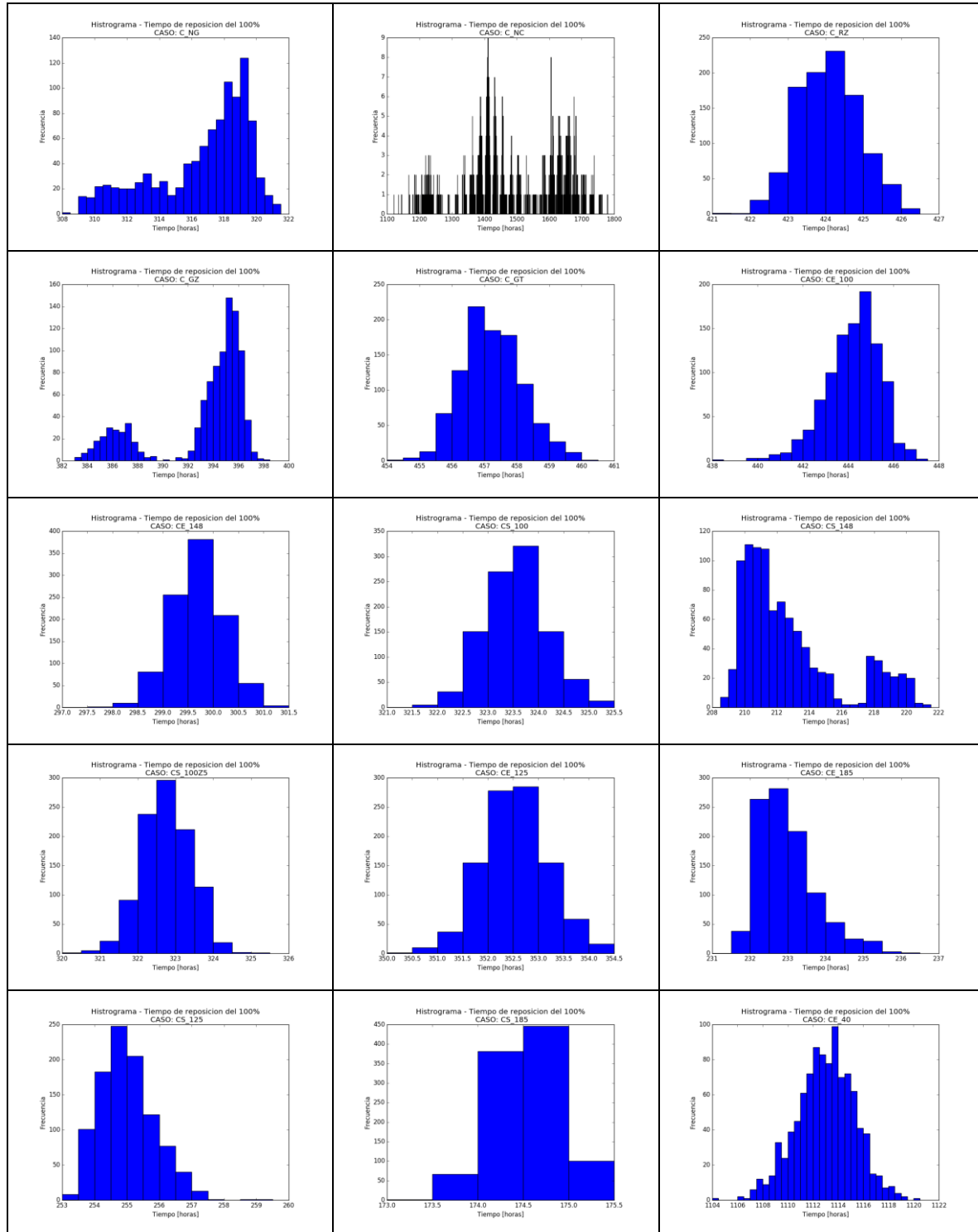


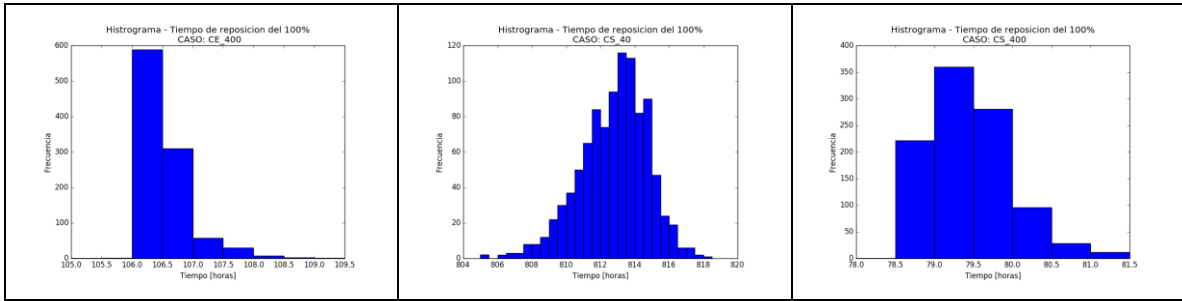
- [56] G. Reinelt, «TSPLIB. A traveling salesman problem library,» *INFORMS Journal on Computing*, vol. 3, n° 4, pp. 376-384, 1991.
- [57] Comisión Nacional de Energía, Norma Técnica de Calidad de Servicio para Sistemas de Distribución, Santiago de Chile, 2017.
- [58] Ministerio de Economía, Gobierno de Chile, Ley 18.410, Crea una Superintendencia de Electricidad y Combustibles, 2012.
- [59] Ministerio de Energía, Gobierno de Chile, Informa y comunica nuevos valores del Costo de Falla de Corta y Larga Duracion en el Sistema Eléctrico Nacional y Los Sistemas Medianos, Santiago, 2018.
- [60] Enersis, Especificacion Tecnica: Acción Sísmica en Equipos Electricos y Mecanicos (E-SE-010), 2003.
- [61] Ministerio de Vivienda y Urbanismo, Gobierno de Chile, NTM001-Diseño sísmico de componentes y sistemas no estructurales, Santiago, 2013.
- [62] CIGRE, Experiencias y desafíos en el sistema eléctrico de CGE Distribución a raíz del terremoto 27 de febrero, 2017.
- [63] R. Rosenfeld y J. Irazábal, Computabilidad, complejidad computacional y verificación de programas, Universidad Nacional de La Plata, 2013.

# 10. Anexos

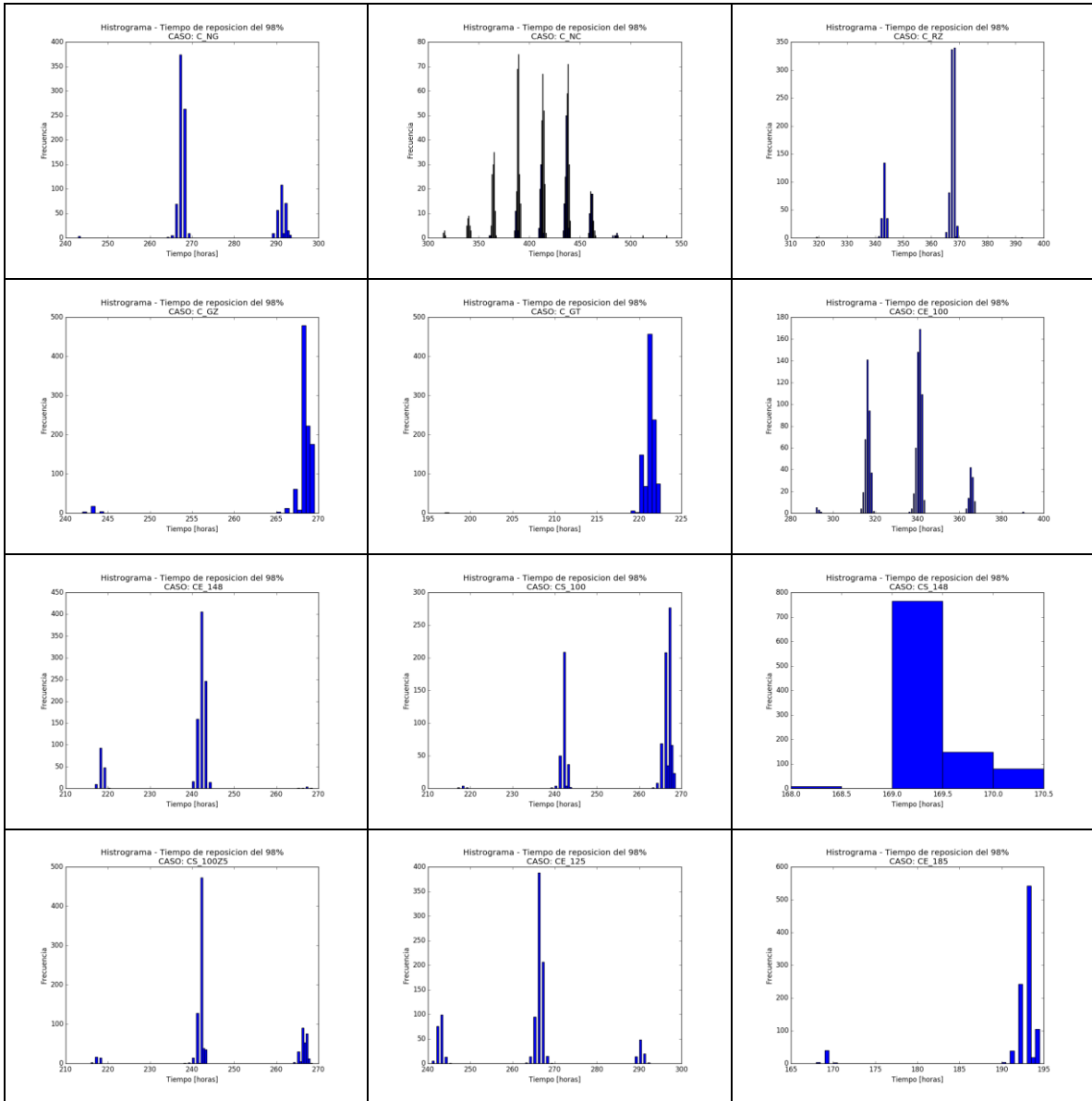
## 10.1 Anexo A – Histogramas de resultados globales

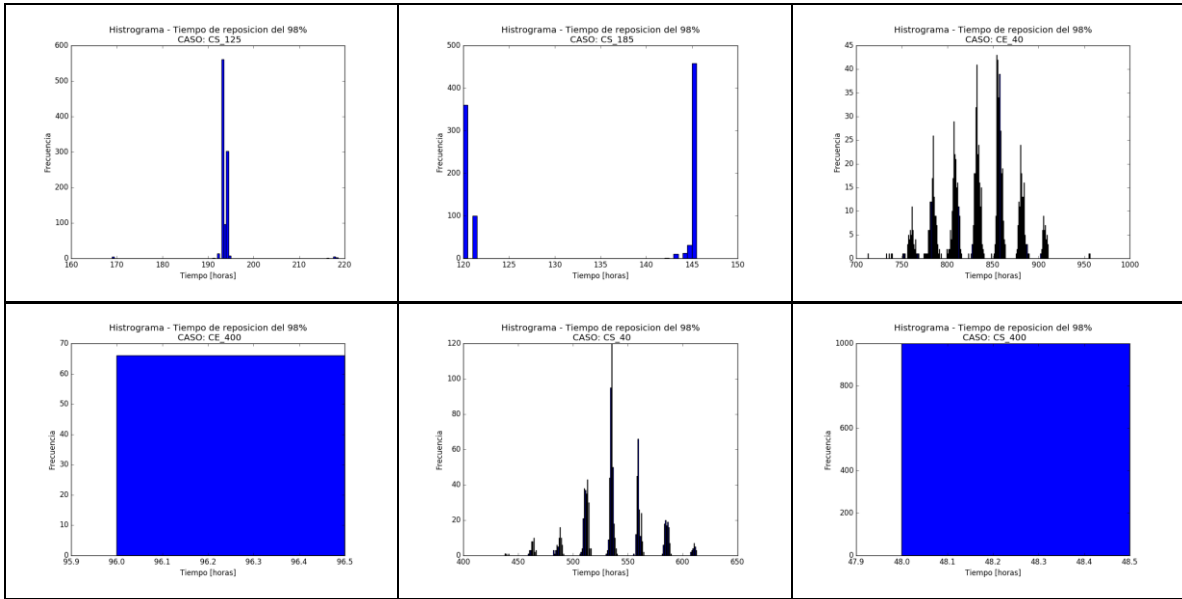
### 10.1.1 Tiempo de restauración del 100%



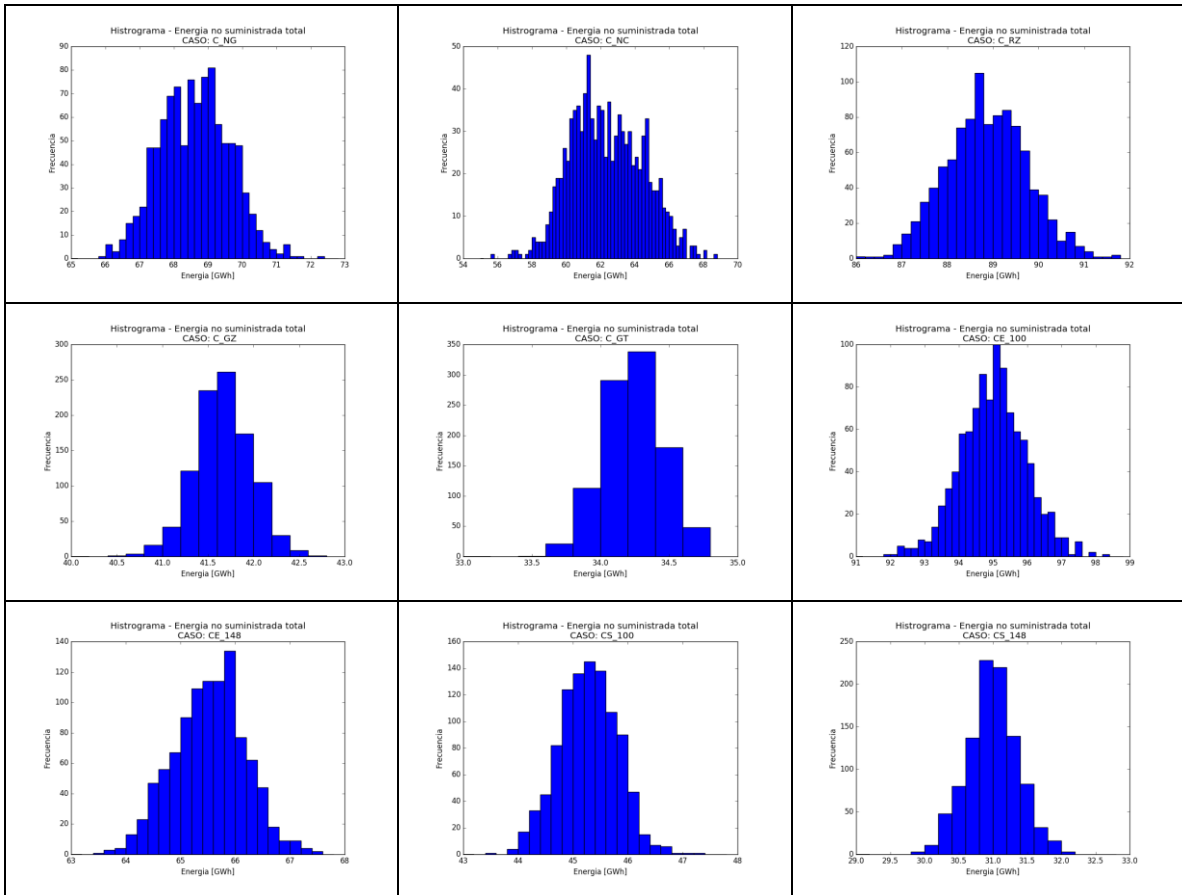


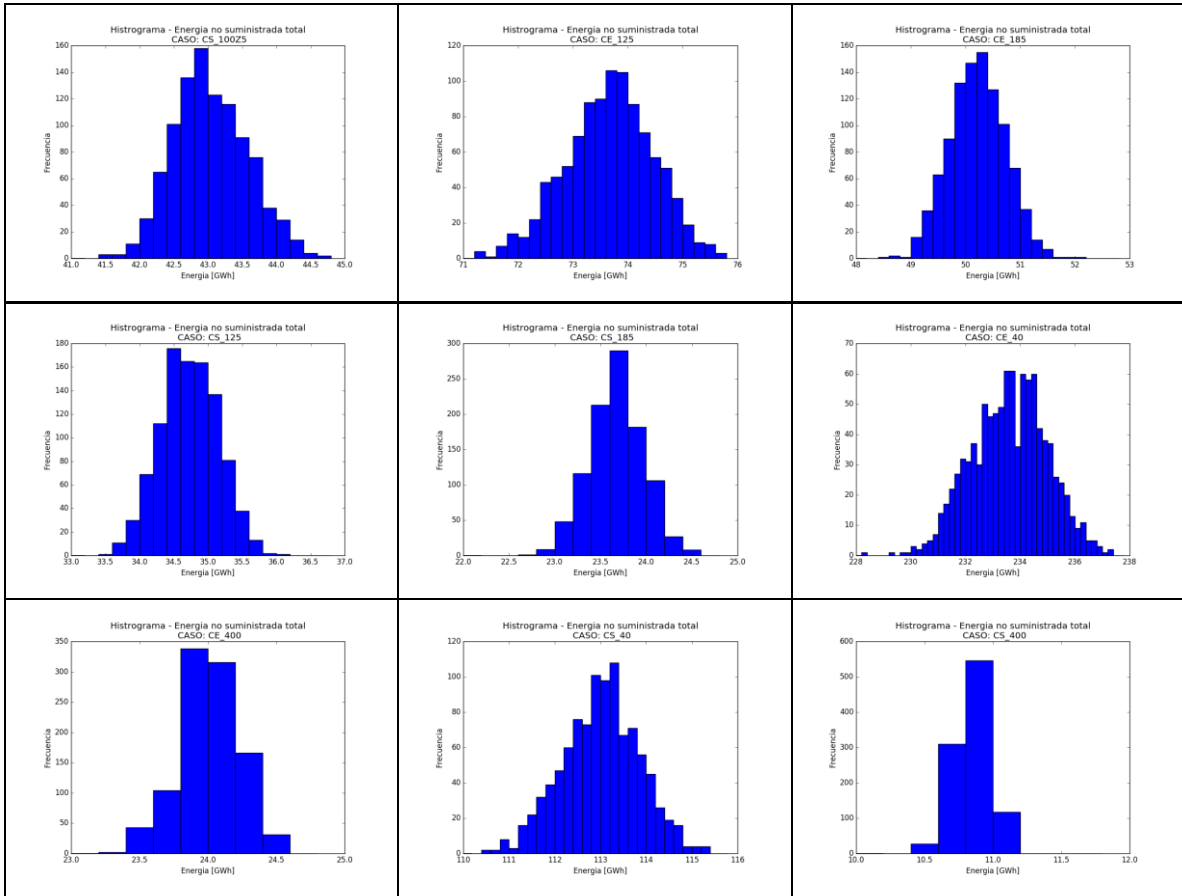
### 10.1.2 Tiempo de restauración del 98%



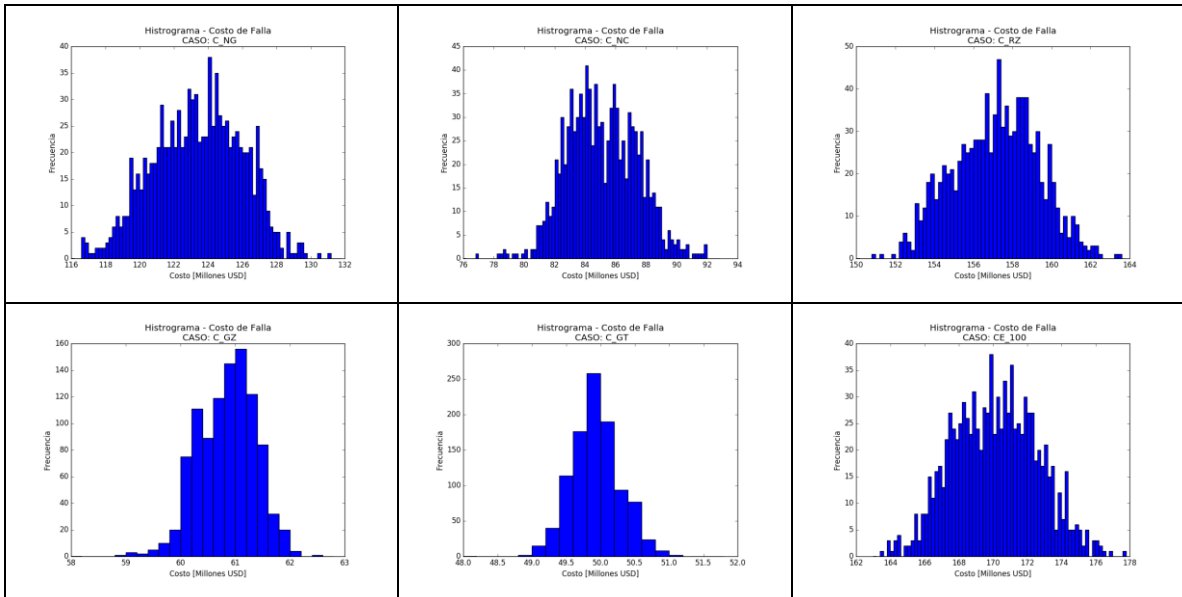


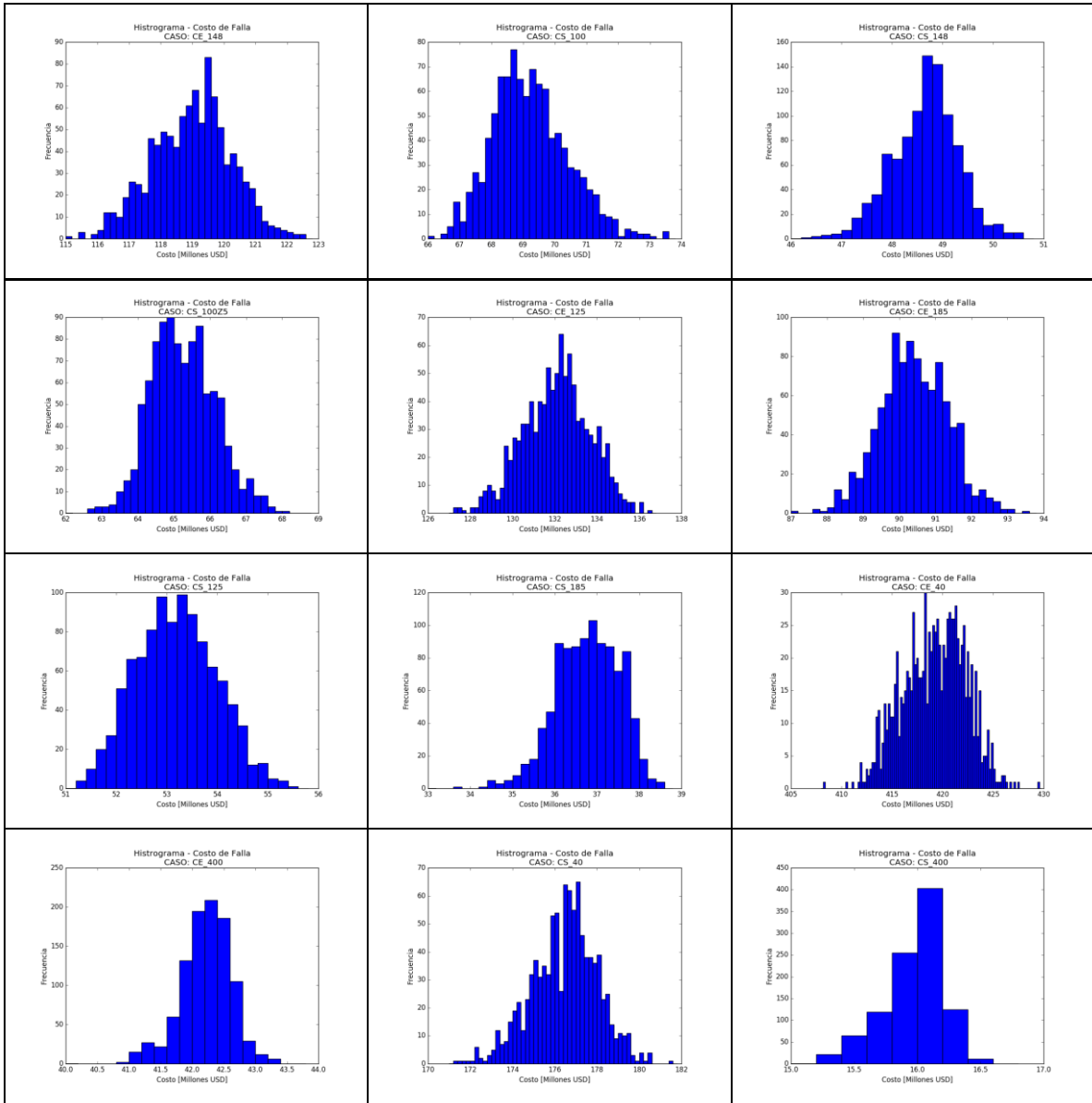
### 10.1.3 Energía no suministrada total



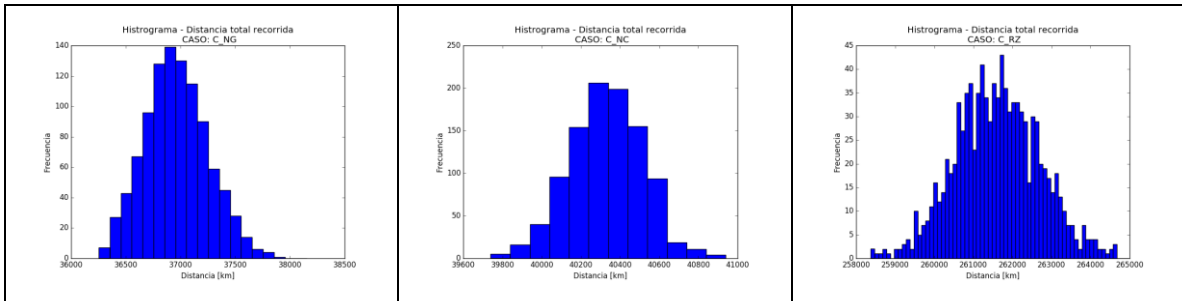


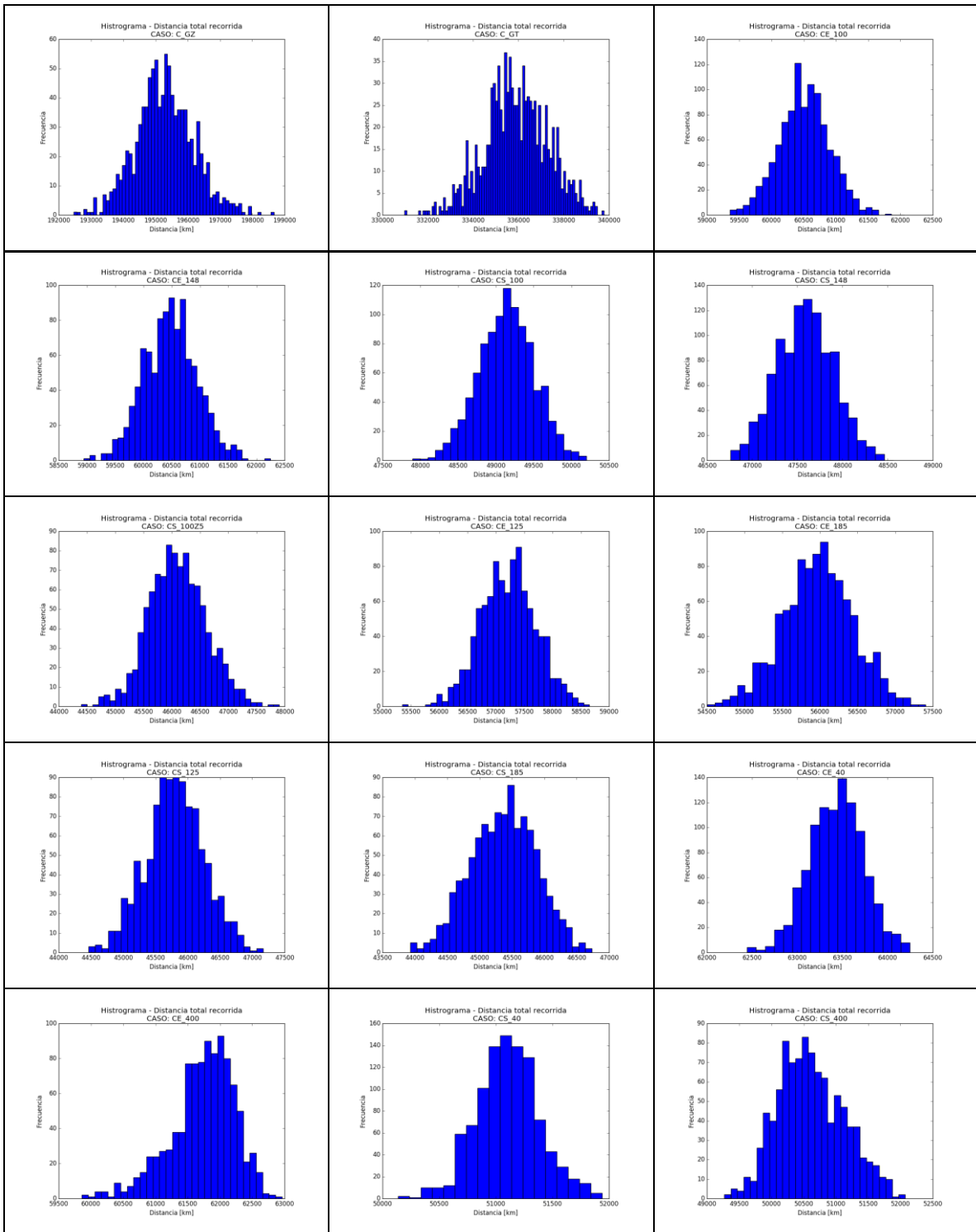
### 10.1.4 Costos de falla



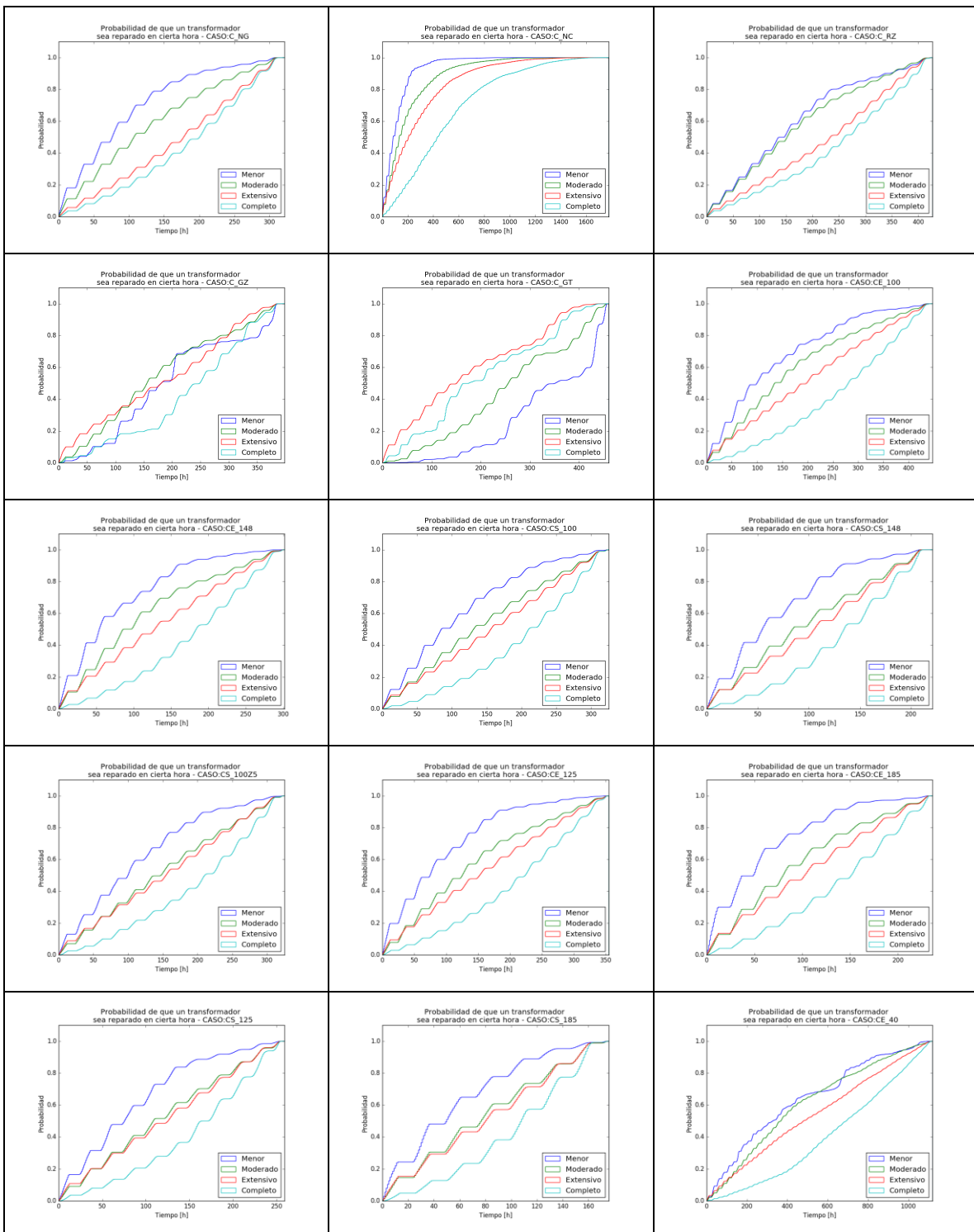


### 10.1.5 Distancias recorridas totales

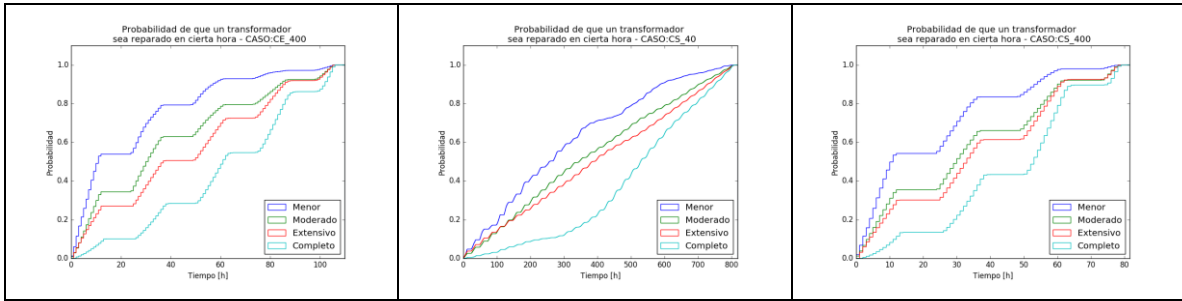




## 10.2 Anexo B – Curvas de Restauración







### 10.3 Anexo C – Porcentaje de transformadores dañados para los casos de preparación de la red

		ZONA1	ZONA2	ZONA3	ZONA4
CE_100	CUADRILLA Z1	63,44	3,87	0,39	0,17
	CUADRILLA Z2	4,66	67,43	0,45	0,13
	CUADRILLA Z3	18,11	15,01	97,89	0,20
	CUADRILLA Z4	13,79	13,69	1,27	99,50
CE_148	CUADRILLA Z1	62,15	5,54	0,30	0,30
	CUADRILLA Z2	4,54	68,48	0,21	0,33
	CUADRILLA Z3	18,66	13,66	98,58	0,59
	CUADRILLA Z4	14,65	12,33	0,91	98,78
CS_100	CUADRILLA Z1	62,45	4,00	0,46	0,14
	CUADRILLA Z2	5,29	66,63	0,37	0,16
	CUADRILLA Z3	18,69	14,87	97,70	0,16
	CUADRILLA Z4	13,57	14,50	1,47	99,55
CS_148	CUADRILLA Z1	61,17	5,17	0,31	0,29
	CUADRILLA Z2	4,54	68,07	0,18	0,32
	CUADRILLA Z3	19,51	13,91	98,43	0,48
	CUADRILLA Z4	14,78	12,84	1,07	98,91

		ZONA1	ZONA2	ZONA3	ZONA4	ZONA5
CE_125	CUADRILLA Z1	68,71	2,45	0,00	0,33	9,69
	CUADRILLA Z2	2,08	79,62	0,00	0,83	1,90
	CUADRILLA Z3	14,99	7,30	100,00	1,22	15,05
	CUADRILLA Z4	8,25	6,36	0,00	97,09	7,88
	CUADRILLA Z5	5,97	4,28	0,00	0,54	65,49
CE_185	CUADRILLA Z1	69,59	2,71	0,11	0,48	9,07
	CUADRILLA Z2	2,60	78,12	0,15	0,59	2,64
	CUADRILLA Z3	13,68	8,85	99,48	1,94	14,79
	CUADRILLA Z4	8,20	6,76	0,18	96,28	8,04
	CUADRILLA Z5	5,93	3,56	0,07	0,72	65,47
CS_125	CUADRILLA Z1	70,71	2,60	0,00	0,41	8,08
	CUADRILLA Z2	2,43	79,16	0,00	0,91	2,41

	CUADRILLA Z3	13,99	7,65	100,00	1,07	16,60
	CUADRILLA Z4	7,52	6,54	0,00	97,17	8,06
	CUADRILLA Z5	5,35	4,05	0,00	0,45	64,86
CS_185	CUADRILLA Z1	71,12	2,84	0,04	0,52	7,89
	CUADRILLA Z2	2,67	77,59	0,08	0,59	3,14
	CUADRILLA Z3	12,90	8,96	99,77	1,88	16,60
	CUADRILLA Z4	7,44	6,88	0,08	96,25	8,30
	CUADRILLA Z5	5,87	3,72	0,04	0,75	64,06

## 10.4 Anexo D – Código fuente

### 10.4.1 Pre-Procesamiento de datos – cargaCapas\_PASO1.py

```

##CARGA LOS LAYERS EN QGIS A USAR
##CASO DE 4 ZONAS
import time
import math
import random
import subprocess

from PyQt4.QtCore import *
from PyQt4.QtGui import *

from qgis.core import *
from qgis.gui import *

from qgis.networkanalysis import *

import numpy as np
import scipy.stats as stats
import scipy.special as special
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from operator import itemgetter

##CARGAMOS LOS LAYERS
##Se requiere que se haga de forma separada al resto del codigo para poder
##carga adecuadamente las metricas a utilizar en el grafo
RedVial = iface.addVectorLayer("./Software_Memoria/LayersQGIS/RedVial/vial_vad.shp",
"", "ogr")
TrafosT =
iface.addVectorLayer("./Software_Memoria/LayersQGIS/Trafos/TrafosTotales_5Z_LITE.shp",
"", "ogr")
CentroM =
iface.addVectorLayer("./Software_Memoria/LayersQGIS/CentroOperaciones/CentroOpZ5.shp",
"", "ogr")

##INICIALIZO VALORES INAMOVIBLES
ZONAS = 5

```

### 10.4.2 Pre-Procesamiento de datos – getGraph\_PASO2.py

```

##CREA GRAFO Y OBTIENE OTROS ARREGLOS RELEVANTES

##Inicializo los constructores del grafo
director = QgsLineVectorLayerDirector(RedVial, -1, '', '', '', 3)
properter = QgsDistanceArcProperter ()

```

```

director.addProperter (properter)
crs = qgis.utils.iface.mapCanvas().mapRenderer().destinationCrs ()
builder = QgsGraphBuilder (crs)

#Obtengo la informacion relevante de la Capa de Trafos en QGIS
#para el grafo y para los arreglos con los que se trabajara
L=[]
Nodos = []
TrafosID = []
TrafosZona4 = []
TrafosZona5 = []
PGA = []
Capacidad = []
SSEE = []

feat = TrafosT.getFeatures ()
for feature in feat:
    L.append(QgsPoint (feature [1], feature [2]))
    TrafosID.append (feature [0])
    Nodos.append ([feature [1], feature [2]])
    PGA.append (feature [3])
    TrafosZona4.append (feature [10])
    TrafosZona5.append (feature [11])
    Capacidad.append (feature [7])
    SSEE.append (feature [9])

for i in range (ZONAS):
    feat = CentroM.getFeatures ()
    for feature in feat:
        if feature [3]==(i+1):
            L.append(QgsPoint (feature [1], feature [2]))
            Nodos.append ([feature [1], feature [2]])
            break

##ARREGLOS CONSTRUIDOS HASTA AHORA
## Nodos = [[CoordX, CoordY],...]
##     Primeros elementos corresponden a los trafos
##     Ultimos "ZONAS" elementos corresponde a los Centros de
##     Mantenimiento, ordenados de Zonal a Zona4
##
## TrafosID = [ID,...]
##     La ID del trafa en QGIS asociada a cada trafa en "Nodos"
##     La posicion de la ID en este arreglo corresponde a la posicion
##     del trafa en el arreglo "Nodos"
##
## CentroMID = [ID,...]
##     La Zona a la que pertenece el centro de mantenimiento. La posicion
##     de la ID corresponde a la posicion "len(TrafosID)-1+ID" en Nodos.

##Uno los puntos de los trafos a la red vial
tiedPoints = director.makeGraph (builder, L)

##Creo el grafo
grafo = builder.graph ()

##Guardo el indice del grafo
indexGrafo=[]
for i in range (len (tiedPoints)):
    indexGrafo.append (grafo.findVertex (tiedPoints [i]))

##     Creamos len(Nodos)-1 archivos que contienen los costos del par de
##     nodos (i,j), donde j>i. Los costos estan en metros
## Este bloque no es necesario correrlo si ya se cuenta con el archivo de costos
for i in range (len (Nodos)-1):
    (tree, cost) = QgsGraphAnalyzer.dijkstra (grafo, indexGrafo [i], 0)
    DistanciaN=[]
    for j in range (len (tiedPoints)):
        if j>i:
            DistanciaN.append (round (cost [indexGrafo [j]], 0))

```

```

path = "./Software_Memoria/Recursos/Lite/MatrizCostos/c"+str(i)
f=open(path,"w")
f.write(" ".join(str(x) for x in DistanciaN))
f.close()

#####
CapSSEE = []
for i in range(42):
    CapSSEE.append(0)

for i in range(len(SSEE)):
    for j in range(42):
        if int(SSEE[i])==j:
            CapSSEE[j]=CapSSEE[j]+Capacidad[i]
            break

ProrTrafo = []
for i in range(len(SSEE)):
    ProrTrafo.append(round(float(Capacidad[i])/CapSSEE[int(SSEE[i])],10))

```

### 10.4.3 Pre-Procesamiento de datos – guardarVAR\_PASO3.py

```

path1 = "./Software_Memoria/Recursos/Lite/TrafosID"
path2 = "./Software_Memoria/Recursos/Lite/TrafosZona4"
path3 = "./Software_Memoria/Recursos/Lite/TrafosZona5"
path4 = "./Software_Memoria/Recursos/Lite/PGA"
path5 = "./Software_Memoria/Recursos/Lite/Capacidad"
path6 = "./Software_Memoria/Recursos/Lite/SSEE"
path7 = "./Software_Memoria/Recursos/Lite/ProrTrafo"
f1=open(path1,"w")
f2=open(path2,"w")
f3=open(path3,"w")
f4=open(path4,"w")
f5=open(path5,"w")
f6=open(path6,"w")
f7=open(path7,"w")
for i in range(len(TrafosID)):
    f1.write(str(TrafosID[i])+"\n")
    f2.write(str(TrafosZona4[i])+"\n")
    f3.write(str(TrafosZona5[i])+"\n")
    f4.write(str(PGA[i])+"\n")
    f5.write(str(Capacidad[i])+"\n")
    f6.write(str(SSEE[i])+"\n")
    f7.write(str(ProrTrafo[i])+"\n")
f1.close()
f2.close()
f3.close()
f4.close()
f5.close()
f6.close()
f7.close()

path1 = "./Software_Memoria/Recursos/Lite/Nodos"
path2 = "./Software_Memoria/Recursos/Lite/indexGrafo"
f1=open(path1,"w")
f2=open(path2,"w")
for i in range(len(Nodos)):
    f1.write(" ".join(str(x) for x in Nodos[i])+"\n")
    f2.write(str(indexGrafo[i])+"\n")
f1.close()
f2.close()

```

## 10.4.4 Modelo de Simulación y Ruteo – Software\_C3.py

```
import time
import math
import random
import os
import subprocess

import numpy as np
import scipy.special as special
from sklearn.cluster import KMeans
from operator import itemgetter

#Entrega los costos en metros
def getCost(i,j):
    if i==j:
        costo = 0
    elif i<j:
        path = "./Recursos/Lite/MatrizCostos/c"+str(i)
        f = open(path,"r")
        linea = f.read()
        f.close()

        Aux = map(float, linea.split())
        costo = Aux[j-i-1]
    else:
        path = "./Recursos/Lite/MatrizCostos/c"+str(j)
        f = open(path,"r")
        linea = f.read()
        f.close()

        Aux = map(float, linea.split())
        costo = Aux[i-j-1]
    if np.isinf(costo):
        return -1
    else:
        return costo

def LookUpTable(TablaD,i,j):
    if i==j:
        return [TablaD,0]
    if i<j:
        ti = i
        tj = j
    else:
        ti = j
        tj = i

    if not TablaD:
        C = getCost(ti,tj)
        TablaD.append((ti, tj, C))
        return [TablaD,C]
    else:
        for k in range(len(TablaD)):
            if TablaD[k][0]==ti:
                if TablaD[k][1]==tj:
                    return [TablaD, TablaD[k][2]]
                elif TablaD[k][1]>=tj:
                    C = getCost(ti,tj)
                    TablaD.insert(k,(ti, tj, C))
                    return [TablaD, C]
            elif TablaD[k][0]>=ti:
                C = getCost(ti,tj)
                TablaD.insert(k,(ti, tj, C))
                return [TablaD, C]

    C = getCost(ti,tj)
    TablaD.insert(k+1,(ti, tj, C))
    return [TablaD, C]
```

```

def phi(x):
    return (1.0 + special.erf(x/np.sqrt(2.0)))/2.0

def lognormalcdf(x, mu, sigma):
    return phi((np.log(x)-mu)/sigma)

def getCurvFra(PGA, FraParam):
    IntFra=np.zeros((PGA.size,4))
    for i in range(4):
        IntFra[:,i] = lognormalcdf(PGA,
np.log(np.ones(PGA.size)*FraParam[i][0]), np.ones(PGA.size)*FraParam[i][1])
    return IntFra

def getFalla(InterFra, NT):
    val = np.random.rand(NT)
    Res = np.zeros(NT, dtype=int)
    s = val<InterFra[:,3]
    Res[s] = (Res[s]+1)*80
    s = np.logical_and(val<InterFra[:,2], val>=InterFra[:,3])
    Res[s] = (Res[s]+1)*50
    s = np.logical_and(val<InterFra[:,1], val>=InterFra[:,2])
    Res[s] = (Res[s]+1)*12
    s = np.logical_and(val<InterFra[:,0], val>=InterFra[:,1])
    Res[s] = (Res[s]+1)*4
    #s = Res[val>=InterFra[:,0]]
    #Res[s] = (Res[s]+1)*0
    return Res

def getRestV4(Falla):
    Res = np.zeros((Falla.size))
    s = Falla==4
    Res[s] = (Res[s]+1)*0.25
    s = Falla==12
    Res[s] = (Res[s]+1)*0.5
    s = Falla==50
    Res[s] = (Res[s]+1)*1
    s = Falla==80
    Res[s] = (Res[s]+1)*2.0
    return Res

def getEnergia(mes, dia, hora, horizonte):
    path1 = "./Recursos/Lite/EnergiaNorm.csv"
    f1=open(path1,"r")
    flagS = 0
    flagE = 0
    EnergiaHor = np.zeros(42)

    flagL0=0
    for lineal in f1:
        if flagL0==0:
            flagL0 = 1
            continue
        Aux = map(float, lineal.split(","))
        if mes==int(Aux[0]) and dia==int(Aux[1]) and hora==int(Aux[2]):
            flagS = 1
            if flagS==1:
                EnergiaHor = EnergiaHor + np.array(Aux[3:])
                flagE = flagE+1
            if flagE==horizonte:
                break
    f1.close()
    return EnergiaHor

def getRutaV5(CyTrafosXY, TasaTrafos, RestTrafos, CyTrafosIndx, Cluster, TasaSet,
MAXTRAB):
    DESCANSO = RestTrafos[0]

    IndxRuta = np.zeros(TasaTrafos.size, dtype=int)-1
    IndxRuta[0] = CyTrafosIndx[0]

    SubIndx = np.arange(TasaTrafos.size)

```

```

NodoInicio = CyTrafosIndx [0]
NodoInicioXY = CyTrafosXY [0]
j=1

ResAux = []
for f in range(TasaSet.size):
    s = TasaTrafos==TasaSet[f]
    TrafoXY = CyTrafosXY[s]
    TAMANO = TrafoXY[:,1].size

    if TAMANO==0:
        continue
    elif TAMANO==1:
        IndxRuta[j] = CyTrafosIndx[s]

        j=j+TAMANO
        NodoInicio = CyTrafosIndx[s][0]
        NodoInicioXY = TrafoXY
    else:
        TrafosAux = np.zeros((TAMANO+1,2))
        IndxAux = np.zeros(TAMANO+1)
        DemAux = np.zeros(TAMANO+1)
        TrafosAux[0] = NodoInicioXY
        IndxAux[0] = NodoInicio
        DemAux[0] = 0

        TrafosAux[1:] = TrafoXY
        IndxAux[1:] = CyTrafosIndx[s]
        DemAux[1:] = RestTrafos[s]*60

        res = getRutaLKHV5(TrafosAux, DemAux, IndxAux, Cluster, MAXTRAB*60)

        IndxRuta[j:j+TAMANO] = res

        j=j+TAMANO
        NodoInicio = res[TAMANO-1]
        NodoInicioXY = CyTrafosXY[SubIndx[CyTrafosIndx==NodoInicio]]

return IndxRuta

##PARA RUTEO CCVRP!!!
def getRutaLKHV5(CyTrafosXY, RestTrafos, CyTrafosIndx, Cluster, Capacidad):
    CapTotal = int(Capacidad*np.ceil(np.sum(RestTrafos)/Capacidad))
    IndxTSP = np.arange(RestTrafos.size)+1
    COORDXY = np.zeros((RestTrafos.size,3))
    COORDXY[:,0] = IndxTSP
    COORDXY[:,1] = CyTrafosXY[:,0]
    COORDXY[:,2] = CyTrafosXY[:,1]

    DEM = np.zeros((RestTrafos.size,2))
    DEM[:,0] = IndxTSP
    DEM[:,1] = RestTrafos

    ListCOORDXY = COORDXY.tolist()
    ListCOORDXY = [map(int, x) for x in ListCOORDXY]
    ListCOORDXY = [map(str, subCOORD) for subCOORD in ListCOORDXY]
    ListCOORDXY = [' '.join(x) for x in ListCOORDXY]
    ListCOORDXY = "\n".join(ListCOORDXY)+"\n"

    ListDEM = DEM.tolist()
    ListDEM = [map(int, x) for x in ListDEM]
    ListDEM = [map(str, subDEM) for subDEM in ListDEM]
    ListDEM = [' '.join(x) for x in ListDEM]
    ListDEM = "\n".join(ListDEM)+"\n"

    path = "./LKH/LKH-3.0.2-C1"
    archivePAR = "C1" + str(Cluster) + ".par"
    archiveTSP = "C1" + str(Cluster) + ".vrp"
    strPAR = "SPECIAL\nPROBLEM_FILE = ./Entrada/"+archiveTSP+"\nSALESMEN =
1\nMAX TRIALS = 1000\nRUNS = 1\nTRACE LEVEL = 0\nTIME_LIMIT =
0.05\nOUTPUT_TOUR_FILE=./Resultado/C1"+str(Cluster)
fl=open(path+"/Entrada/"+archivePAR,"w")

```

```

f1.write(strPAR)
f1.close()

f1=open(path+"/Entrada/"+archiveTSP,"w")
strTSP = "NAME : C1"+str(Cluster)+"\nTYPE : CCVRP\nDIMENSION :
"+str(RestTrafos.size)+"\nEDGE_WEIGHT_TYPE : EUC_2D\nCAPACITY :
"+str(CapTotal)+"\nNODE_COORD_SECTION\n"
f1.write(strTSP)
f1.write(ListCOORDXY)
f1.write("DEMAND_SECTION\n")
f1.write(ListDEM)
f1.write("DEPOT_SECTION\n1\n-1\nEOF")
f1.close()

subprocess.call("cd ./LKH/LKH-3.0.2-C1; ./LKH ./Entrada/C1"+str(Cluster)+".par",
shell=True)

res = np.zeros(RestTrafos.size-1, dtype=int)
f1 = open(path+"/Resultado/C1"+str(Cluster), "r")
count = 0
j=0
for linea in f1:
    if count<7:
        count=count+1
    else:
        Aux = int(linea)
        if int(Aux)==-1:
            break
        else:
            res[j]=CyTrafosIndx[Aux-1]
            j=j+1
return res

def getCostTV5(Ruta, Rest, z, DESCANSO, TRABAJO, VELOCIDAD, NT):
    NTRAFOS = NT-1
    sumTrab = 0
    TiempoT = []
    DistanciaT = []
    IndexInsrt = []
    ListTrab = []
    TiempoSum = 0.0
    for k in range(len(Ruta)):
        if k==0:
            i = NTRAFOS+z
            j = Ruta[k]
            costoViaje = getCost(i,j)
            Tiempo_Aux = costoViaje/VELOCIDAD/1000.0 + Rest[k]
            sumTrab = sumTrab + Tiempo_Aux
            TiempoSum = TiempoSum + Tiempo_Aux
            TiempoT.append(TiempoSum)
            DistanciaT.append(costoViaje)
        elif k!=0 and sumTrab<TRABAJO:
            i = Ruta[k-1]
            j = Ruta[k]
            costoViaje = getCost(i,j)
            Tiempo_Aux = costoViaje/VELOCIDAD/1000.0 + Rest[k]
            sumTrab = sumTrab + Tiempo_Aux
            TiempoSum = TiempoSum + Tiempo_Aux
            TiempoT.append(TiempoSum)
            DistanciaT.append(costoViaje)
        elif k!=0 and TRABAJO<=sumTrab:
            i = Ruta[k-1]
            j = NTRAFOS+z
            costoViaje = getCost(i,j)
            Tiempo_Aux = costoViaje/VELOCIDAD/1000.0 + DESCANSO
            ListTrab.append(sumTrab)
            sumTrab = 0
            TiempoSum = TiempoSum + Tiempo_Aux
            TiempoT.append(TiempoSum)
            DistanciaT.append(costoViaje)
            IndexInsrt.append(len(TiempoT)-1)

```



```

        i = NTRAFOS+z
        j = Ruta[k]
        costoViaje = getCost(i,j)
        Tiempo_Aux = costoViaje/VELOCIDAD/1000.0 + Rest[k]
        sumTrab = sumTrab + Tiempo_Aux
        TiempoSum = TiempoSum + Tiempo_Aux
        TiempoT.append(TiempoSum)
        DistanciaT.append(costoViaje)
    ListTrab.append(sumTrab)
    return [TiempoT, DistanciaT, IndexInsrt, ListTrab]

if __name__ == '__main__':

    #----- PARAMETROS A AJUSTAR -----
    NOMBREFILE = "CASO3/CASO3"

    MES = 2
    DIA = 27
    HORA = 3
    HORIZONTE = 168

    ZONAS = 4
    NTRAFOS = 21703

    DESCANSO = 12
    TRABAJO = 12
    VELOCIDAD = 40.0

    CUADRILLA_Z = 25
    CLUSTER = ZONAS*CUADRILLA_Z

    SIM = 10
    #Componentes sismicas
    CURV_FRA_P = [[0.28, 0.3], [0.4, 0.2], [0.72, 0.15], [1.10, 0.15]]

    #Componentes estandars
    #CURV_FRA_P = [[0.24, 0.25], [0.33, 0.2], [0.58, 0.15], [0.89, 0.15]]

    #-----

    Nodos = []
    TrafoZona = []
    PGA = []
    Capacidad = []
    SSEE = []
    ProrTrafo = []

    path1 = "./Recursos/Lite/Nodos"
    f1=open(path1,"r")
    for i in range(NTRAFOS+ZONAS):
        linea1 = map(float, (f1.readline()).split())
        Nodos.append(linea1)
    f1.close()

    path1 = "./Recursos/Lite/TrafosZona"+str(ZONAS)
    path2 = "./Recursos/Lite/PGA"
    path3 = "./Recursos/Lite/Capacidad"
    path4 = "./Recursos/Lite/SSEE"
    path5 = "./Recursos/Lite/ProrTrafo"
    f1=open(path1,"r")
    f2=open(path2,"r")
    f3=open(path3,"r")
    f4=open(path4,"r")
    f5=open(path5,"r")
    for i in range(NTRAFOS):
        linea1 = int(f1.readline())
        linea2 = float(f2.readline())
        linea3 = float(f3.readline())
        linea4 = int(f4.readline())

```

```

        linea5 = float(f5.readline())
        TrafoZona.append(linea1)
        PGA.append(linea2)
        Capacidad.append(linea3)
        SSEE.append(linea4)
        ProrTrafo.append(linea5)
f1.close()
f2.close()
f3.close()
f4.close()
f5.close()

#Convertir en numpy array para flexibilizar y acelerar calculos
Nodos = np.array(Nodos)
TrafoZona = np.array(TrafoZona, dtype=int)
PGA = np.array(PGA)
Capacidad = np.array(Capacidad)
SSEE = np.array(SSEE, dtype=int)
ProrTrafo = np.array(ProrTrafo)

#Indice de los arreglos asociados a los trafos
IndxTrafos = np.arange(NTRAFOS)

#Energia suministrada en el horizonte de tiempo dada la fecha de inicio
EnergiaEv = getEnergia(MES, DIA, HORA, HORIZONTE)
EnergiaEvTrafo = np.ones(NTRAFOS)*EnergiaEv[SSEE]*ProrTrafo

#Puntos de la curva de fragilidad de acuerdo al PGA
CurvFraTraf = getCurvFra(PGA, CURV_FRA_P)

#Indexadores para moverse en clusters
Z = np.arange(ZONAS)+1
C = (np.arange(ZONAS)+1)*CUADRILLA_Z

pathT = "./Resultados/Raw/"+NOMBREFILE+"Tiempo"
pathF = "./Resultados/Raw/"+NOMBREFILE+"Falla"
pathTT = "./Resultados/Raw/"+NOMBREFILE+"TrafoTipo"
pathN = "./Resultados/Raw/"+NOMBREFILE+"TrafoID"
pathR = "./Resultados/Raw/"+NOMBREFILE+"TiempoRuta"
fT=open(pathT,"w")
fF=open(pathF,"w")
fTT=open(pathTT,"w")
fN=open(pathN,"w")
fR=open(pathR,"w")

paths = []
archC = []
for i in range(CLUSTER):
    paths.append([])
    archC.append([])
    paths[i].append("./Resultados/Raw/"+NOMBREFILE+"Tiempo_CUADRILLA_"+str(i))
    paths[i].append("./Resultados/Raw/"+NOMBREFILE+"Falla_CUADRILLA_"+str(i))

paths[i].append("./Resultados/Raw/"+NOMBREFILE+"TrafoTipo_CUADRILLA_"+str(i))

paths[i].append("./Resultados/Raw/"+NOMBREFILE+"TrafoID_CUADRILLA_"+str(i))

paths[i].append("./Resultados/Raw/"+NOMBREFILE+"TiempoRuta_CUADRILLA_"+str(i))

    archC[i].append(open(paths[i][0],"w"))
    archC[i].append(open(paths[i][1],"w"))
    archC[i].append(open(paths[i][2],"w"))
    archC[i].append(open(paths[i][3],"w"))
    archC[i].append(open(paths[i][4],"w"))

start_time = time.time()

for sim in range(SIM):
    print("INICIANDO SIMULACION: "+str(sim) + "-- EN EL MINUTO:
"+str(round((time.time()-start_time)/60.0,4)))

```

```

#Se calcula la falla por trafo y su tiempo de restauracion por simulacion
- Parte aleatoria
FallaSim = getFalla(CurvFraTraf, NTRAFOS)
ResSim = getRestV4 (FallaSim)

TrafoCluster = np.zeros (NTRAFOS, dtype=int)-1
for centro in range(ZONAS):
    s = np.logical_and (FallaSim!=0, TrafoZona==(centro)+1)
    TrafoAux = Nodos[:NTRAFOS][s]
    IndxAux = IndxTrafos[s]
    kmeans = KMeans(n_clusters=CUADRILLA_Z, max_iter=300).fit(TrafoAux)
    TrafoCluster[IndxAux] = (kmeans.labels_)+CUADRILLA_Z*(centro)

TasaRest = np.zeros (NTRAFOS)
s = FallaSim!=0
TasaRest[s] = np.ceil (EnergiaEvTrafo[s]*FallaSim[s]/100.0/ResSim[s])

#Ruteo - Solo considero las que fueron clusterisadas
RutaC=[]
CostC=[]
for clu in range (CLUSTER):
    z = Z[clu<C]
    s = TrafoCluster==clu
    TamanoClu = TrafoCluster[s].size

    ClustAux = np.zeros ((TamanoClu+1,2))
    IndxAux = np.zeros (TamanoClu+1,dtype=int)
    TasaAux = np.zeros (TamanoClu+1,dtype=int)
    ResAux = np.zeros (TamanoClu+1)

    #Agregamos el centro de distribucion primero
    ClustAux[0] = Nodos[NTRAFOS-1+z[0]]
    IndxAux[0] = NTRAFOS-1+z[0]
    TasaAux[0] = 0
    ResAux[0] = DESCANSO

    #Agregamos el resto de los Nodos asociados al cluster
    ClustAux[1:] = Nodos[:NTRAFOS][s]
    IndxAux[1:] = IndxTrafos[s]
    TasaAux[1:] = TasaRest[s]
    ResAux[1:] = ResSim[s]

    #Creamos el set de prioridades
    TasaSet = np.unique(TasaAux[1:])[:-1]

    Ruta = getRutaV5 (ClustAux, TasaAux, ResAux, IndxAux, clu, TasaSet,
TRABAJO)

    RutaC.append(Ruta)

#Tiempo total de trabajo de la Ruta i
TRutaFin = np.zeros (len (RutaC))
#Tiempo de trabajo permitido de la Ruta i
TRutaTrab = []
#Variables a guardar por Cuadrilla
TiempoC = []
FallaC = []
CapacidadC = []
RutaIDC = []
DistanciaC = []

for i in range (len (RutaC)):
    z = Z[i<C]
    AuxFalla = FallaSim[RutaC[i][1:]].tolist ()
    AuxRest = ResSim[RutaC[i][1:]].tolist ()
    AuxCap = Capacidad[RutaC[i][1:]].tolist ()
    AuxRuta = RutaC[i][1:].tolist ()
    T_D_K_ST = getCostTV5 (AuxRuta, AuxRest, z[0], DESCANSO, TRABAJO,
VELOCIDAD, NTRAFOS)

    for k in T_D_K_ST[2]:

```

```

        AuxFalla.insert(k,0)
        AuxCap.insert(k,0)
        AuxRuta.insert(k,NTRAFOS-1+z[0])

TAMANO = len(T_D_K_ST[0])
TRutaFin[i] = T_D_K_ST[0][TAMANO-1]
TRutaTrab.append(T_D_K_ST[3])

TiempoC.append(T_D_K_ST[0])
FallaC.append(AuxFalla)
CapacidadC.append(AuxCap)
RutaIDC.append(AuxRuta)
DistanciaC.append(T_D_K_ST[1])

#RuteoCooperativo MINMAX
LookUp = []
count = 0
fix = 2*CLUSTER
while count<fix:
    ci = np.argmin(TRutaFin)
    cj = np.argmax(TRutaFin)

    li = len(RutaIDC[ci])-1
    lj = len(RutaIDC[cj])-1

    pi = RutaIDC[ci][li]
    pj = RutaIDC[cj][lj]

    zi = Z[ci<C][0]
    zj = Z[cj<C][0]

    TRutaTrabi=TRutaTrab[ci][len(TRutaTrab[ci])-1]
    TRutaTrabj=TRutaTrab[cj][len(TRutaTrab[cj])-1]

    RutaAux = []
    Tiempoi = TiempoC[ci][li]
    Tiempoj = TiempoC[cj][lj]

    if TRABAJO<=TRutaTrabi:
        LookUpAux = LookUpTable(LookUp,pi,NTRAFOS-1+zi)
        LookUp = LookUpAux[0]
        C1 = LookUpAux[1]/VELOCIDAD/1000.0
        LookUpAux = LookUpTable(LookUp,NTRAFOS-1+zi,pj)
        LookUp = LookUpAux[0]
        C2 = LookUpAux[1]/VELOCIDAD/1000.0
        COMP = C1 + DESCANSO + C2 + Tiempoi + ResSim[pj]
    else:
        LookUpAux = LookUpTable(LookUp,pi,pj)
        LookUp = LookUpAux[0]
        C1 = LookUpAux[1]/VELOCIDAD/1000.0
        COMP = C1 + Tiempoi + ResSim[pj]

    li=-1
    while (COMP<Tiempoj):
        RutaAux.append([FallaC[cj].pop(lj), CapacidadC[cj].pop(lj),
RutaIDC[cj].pop(lj)])

        TAux = TiempoC[cj].pop(lj)
        DistanciaC[cj].pop(lj)

        #Actualizacion del Maximo (j)
        lj = lj-1
        pj = RutaIDC[cj][lj]
        if NTRAFOS<=pj:
            FallaC[cj].pop(lj)
            CapacidadC[cj].pop(lj)
            RutaIDC[cj].pop(lj)
            TiempoC[cj].pop(lj)
            DistanciaC[cj].pop(lj)
            TRutaTrab[cj].pop(len(TRutaTrab[cj])-1)

            lj = lj-1

```

```

        pj = RutaIDC[cj][lj]
        TRutaTrabj=TRutaTrab[cj][len(TRutaTrab[cj])-1]
        TiempoC[cj][lj]

    else:
        TiempoC[cj][lj]
        TRutaTrabj = TRutaTrabj - TAux + TiempoC[cj][lj]

#Actualizacion virtual del Minimo (i)
    li = li+1
    pi = RutaAux[li][2]
    TiempoC[ci] = COMP
    if TRABAJO<=TRutaTrabi:
        TRutaTrabi = C2 + ResSim[pi]
    else:
        TRutaTrabi = TRutaTrabi + C1 + ResSim[pi]

    if TRABAJO<=TRutaTrabi:
        LookUpAux = LookUpTable(LookUp,pi,NTRAFOS-1+zi)
        LookUp = LookUpAux[0]
        C1 = LookUpAux[1]/VELOCIDAD/1000.0
        LookUpAux = LookUpTable(LookUp,NTRAFOS-1+zi,pj)
        LookUp = LookUpAux[0]
        C2 = LookUpAux[1]/VELOCIDAD/1000.0
        COMP = C1 + DESCANSO + C2 + TiempoC[pj] + ResSim[pj]
    else:
        LookUpAux = LookUpTable(LookUp,pi,pj)
        LookUp = LookUpAux[0]
        C1 = LookUpAux[1]/VELOCIDAD/1000.0
        COMP = C1 + TiempoC[pj] + ResSim[pj]

#Actualizacion real del minimo - Inversion de la lista de prioridad
    li = len(RutaIDC[ci])-1
    pi = RutaIDC[ci][li]
    TRutaTrabi=TRutaTrab[ci][len(TRutaTrab[ci])-1]
    for T in reversed(RutaAux):
        if TRABAJO<=TRutaTrabi:
            LookUpAux = LookUpTable(LookUp,pi,NTRAFOS-1+zi)
            LookUp = LookUpAux[0]
            Dist = LookUpAux[1]

TiempoC[ci].append(TiempoC[ci][li]+Dist/VELOCIDAD/1000.0+DESCANSO)
    FallaC[ci].append(0)
    CapacidadC[ci].append(0)
    RutaIDC[ci].append(NTRAFOS-1+zi)
    DistanciaC[ci].append(Dist)
    TRutaTrab[ci][len(TRutaTrab[ci])-1] = TRutaTrabi
    li=li+1

    LookUpAux = LookUpTable(LookUp,NTRAFOS-1+zi,T[2])
    LookUp = LookUpAux[0]
    Dist = LookUpAux[1]

TiempoC[ci].append(TiempoC[ci][li]+Dist/VELOCIDAD/1000.0+ResSim[T[2]])
    FallaC[ci].append(T[0])
    CapacidadC[ci].append(T[1])
    RutaIDC[ci].append(T[2])
    DistanciaC[ci].append(Dist)

    TRutaTrabi = Dist/VELOCIDAD/1000.0+ResSim[T[2]]
    TRutaTrab[ci].append(TRutaTrabi)
    li=li+1
    pi = RutaIDC[ci][li]

    else:
        LookUpAux = LookUpTable(LookUp,pi,T[2])
        LookUp = LookUpAux[0]
        Dist = LookUpAux[1]

TiempoC[ci].append(TiempoC[ci][li]+Dist/VELOCIDAD/1000.0+ResSim[T[2]])
    FallaC[ci].append(T[0])
    CapacidadC[ci].append(T[1])

```

```

RutaIDC[ci].append(T[2])
DistanciaC[ci].append(Dist)

TRutaTrabi = TRutaTrabi +
Dist/VELOCIDAD/1000.0+ResSim[T[2]]
    li=li+1
    pi = RutaIDC[ci][li]
    TRutaTrab[ci][len(TRutaTrab[ci])-1] = TRutaTrabi
    TRutaTrab[cj][len(TRutaTrab[cj])-1] = TRutaTrabj

    TRutaFin[cj]=TiempoC[cj][lj]
    TRutaFin[ci]=TiempoC[ci][li]
    count = count+1

for i in range(CLUSTER):
    for k in range(len(TiempoC[i])):
        CostC.append([TiempoC[i][k], FallaC[i][k],
CapacidadC[i][k], RutaIDC[i][k], DistanciaC[i][k]])

    archC[i][0].write(str(sim)+" "+str(" ".join(str(round(x,4)) for x in
TiempoC[i]))+"\n")
    archC[i][1].write(str(sim)+" "+str(" ".join(str(x) for x in
FallaC[i]))+"\n")
    archC[i][2].write(str(sim)+" "+str(" ".join(str(x) for x in
CapacidadC[i]))+"\n")
    archC[i][3].write(str(sim)+" "+str(" ".join(str(x) for x in
RutaIDC[i]))+"\n")
    archC[i][4].write(str(sim)+" "+str(" ".join(str(x) for x in
DistanciaC[i]))+"\n")

    Aux = sorted(CostC, key=itemgetter(0))

    TiempoT = [row[0] for row in Aux]
    FallaT = [row[1] for row in Aux]
    CapT = [row[2] for row in Aux]
    NodoT = [row[3] for row in Aux]
    CostoRT = [row[4] for row in Aux]

    fT.write(str(sim)+" "+str(" ".join(str(round(x,4)) for x in TiempoT)+"\n")
    fF.write(str(sim)+" "+str(" ".join(str(x) for x in FallaT)+"\n")
    fTT.write(str(sim)+" "+str(" ".join(str(x) for x in CapT)+"\n")
    fN.write(str(sim)+" "+str(" ".join(str(x) for x in NodoT)+"\n")
    fR.write(str(sim)+" "+str(" ".join(str(x) for x in CostoRT)+"\n")

    fT.close()
    fF.close()
    fTT.close()
    fN.close()
    fR.close()

for i in range(CLUSTER):
    archC[i][0].close()
    archC[i][1].close()
    archC[i][2].close()
    archC[i][3].close()
    archC[i][4].close()

print("--- %s seconds ---" % (time.time() - start_time))

```

#### 10.4.5 Procesamiento de Resultados – Procesamiento.py

```

#####
##CARGA DE RESULTADOS
##CASO DE 4 ZONAS - C1
import time
import math
import random

```

```

import numpy as np
from operator import itemgetter
from copy import deepcopy

#-----Parametros a Modificar-----
NOMBREFILE = "CASO2/CASO2"
NOMBREFILEOUT = "CASO2/C2"
ZONAS = 4
'''
NOMBREFILE = "CASO4/CASO4"
NOMBREFILEOUT = "CASO4/C4"
ZONAS = 4

NOMBREFILE = "CASO7/CASO7"
NOMBREFILEOUT = "CASO7/C7"
ZONAS = 5

NOMBREFILE = "CASO8/CASO8"
NOMBREFILEOUT = "CASO8/C8"
ZONAS = 5
'''

SIM=1000

MES = 2
DIA = 27
HORA = 3

NTRAFOS = 21703

PRECIOF = np.array([2055.13, 1507.11, 1043.62, 717.31])
PORFALLA = np.array([20, 10, 5, 0])
#-----

SSEE = []
ProrTrafo = []
path5 = "../Recursos/Lite/SSEE"
path6 = "../Recursos/Lite/ProrTrafo"
f5=open(path5,"r")
f6=open(path6,"r")
for i in range(NTRAFOS):
    linea5 = int(f5.readline())
    linea6 = float(f6.readline())
    SSEE.append(linea5)
    ProrTrafo.append(linea6)
f5.close()
f6.close()

for i in range(ZONAS):
    SSEE.append(0)
    ProrTrafo.append(0)

SSEE = np.array(SSEE, dtype=int)
ProrTrafo = np.array(ProrTrafo)

pathT = "../Resultados/Raw/"+NOMBREFILE+"Tiempo"
pathF = "../Resultados/Raw/"+NOMBREFILE+"Falla"
pathTT = "../Resultados/Raw/"+NOMBREFILE+"TrafoTipo"
pathTID = "../Resultados/Raw/"+NOMBREFILE+"TrafoID"
pathTR = "../Resultados/Raw/"+NOMBREFILE+"TiempoRuta"
fT=open(pathT,"r")
fF=open(pathF,"r")
fTT=open(pathTT,"r")
fTID=open(pathTID,"r")
fTR=open(pathTR,"r")

CostoT = []
FallaT = []

```

```

CapT = []
NodoT = []
CostoRT = []
for i in range(SIM):
    Aux = map(float, (fT.readline()).split(","))
    CostoT.append(Aux[1:len(Aux)-1])
    Aux = map(int, map(float, (fF.readline()).split(",")))
    FallaT.append(Aux[1:len(Aux)-1])
    Aux = map(int, map(float, (fTT.readline()).split(",")))
    CapT.append(Aux[1:len(Aux)-1])
    Aux = map(int, map(float, (fTID.readline()).split(",")))
    NodoT.append(Aux[1:len(Aux)-1])
    Aux = map(float, (fTR.readline()).split(","))
    CostoRT.append(Aux[1:len(Aux)-1])
fT.close()
fF.close()
fTT.close()
fTID.close()
fTR.close()

EnergiaNoSum = []
#Porcentaje de energia no suministrada de acuerdo al tiempo
TiempoNoSumPor98 = []
CostoFalla = []
TiempoNoSumPor100 = []
CostoCamino = []
ENSHorMean = []

#Archivo donde se guarda la energia suministrada en condiciones normales
path1 = "../Recursos/Lite/EnergiaNorm.csv"

#Tiempo de inicio de procesamiento
time_is = time.time()

for CASO in range(SIM):
    #Se imprime el caso procesado para verificar que el algoritmo esta funcinando
    print "CASO",CASO, "PROCESADO EN EL MINUTO", round((time.time()-time_is)/60,2)

    #Se calcula la energia por en las horas del problema
    f1=open(path1,"r")
    flagS = 0
    flagE = 0
    EnergiaHorizonte = []
    horizonte = math.ceil(CostoT[CASO][len(CostoT[CASO])-1])+1
    ENSHorMean.append([])

    flag=0
    for lineal in f1:
        if flag==0:
            flag=1
            continue
        Aux = map(float, lineal.split(","))
        if MES==int(Aux[0]) and DIA==int(Aux[1]) and HORA==int(Aux[2]):
            flagS = 1
        if flagS==1:
            EnergiaHorizonte.append(Aux[3:45])
            flagE = flagE+1
        if flagE==horizonte:
            break
    f1.close()

    EnergiaHorizonte = np.array(EnergiaHorizonte)

    ACostoT = np.array(CostoT[CASO])
    AFallaT = np.array(FallaT[CASO], dtype=int)
    ANodoT = np.array(NodoT[CASO], dtype=int)
    ACostoRT = np.array(CostoRT[CASO])

    finNodo = ANodoT[-1]
    finTiempo = ACostoT[-1]

```



```

flagS=0
flag98=0
CostoFallaTotal=0
t=0
EnergiaNS=0
Tiempo98 = -1

while(t<horizonte and flagS==0):
    s1 = ACostoT>t
    s2 = np.logical_and(ACostoT<t+1,s1)

    EnergiaAux = EnergiaHorizonte[t,:]
    AnergiaNST =
np.sum(AFallaT[s1]/100.0*EnergiaAux[SSEE[ANodoT[s1]]]*ProrTrafo[ANodoT[s1]])

    if np.sum(s2)!=0:
        IndxAux = ANodoT[s2]
        #Obtengo intervalos de tiempo entre reparaciones
        interTime = np.ediff1d(ACostoT[s2], to_begin=ACostoT[s2][0]-t, to_end=t+1-
ACostoT[s2][-1])
        #Obtengo intervalos de energia restaurada entre reparaciones
        interEnergA =
AFallaT[s2]/100.0*EnergiaAux[SSEE[ANodoT[s2]]]*ProrTrafo[ANodoT[s2]]
        interEnerg = np.zeros(interEnergA.size+1)
        interEnerg[1:] = interEnergA
        #Calculo el area bajo la curva de energia no suministrada en el intervalo
[t,t+1]
        EnergiaNST = (AnergiaNST - np.cumsum(interEnerg))*interTime
    else:
        interTime = np.array([1])
        EnergiaNST = np.array([AnergiaNST])

    #print np.sum(EnergiaNST), '-', np.sum(EnergiaAux), '-', t
    #print np.cumsum(interEnerg)*interTime

    #Determinar el tiempo de restauracion 98%
    EvEnergia98 = EnergiaNST/np.sum(EnergiaAux)*100
    s98 = EvEnergia98<=2.0
    if np.sum(s98)!=0 and flag98==0:
        Tiempo98 = interTime[s98][0]+t
        flag98=1
    elif np.sum(s98)==0 and flag98==1:
        flag98=0
    elif np.sum(s98)!=s98.size and flag98==1:
        Tiempo98 = interTime[s98][0]+t

    EnergiaNS = EnergiaNS + np.sum(EnergiaNST)

    #Calculo de la costo de falla por hora
    AuxPor = np.sum(EnergiaNST)/np.sum(EnergiaAux)*100
    CostoFallaTotal = CostoFallaTotal +
np.sum(EnergiaNST)/1000.0*PRECIOf[AuxPor>=PORFALLA][0]

    #Guardo Evolucion de ENS horaria por caso
    ENSHorMean[CASO].append(np.sum(EnergiaNST))

    t=t+1
    sFin = np.array([0])
    if np.sum(s2)!=0:
        sFin = np.logical_and(IndxAux==finNodo, ACostoT[s2]==finTiempo)
    if np.sum(sFin)!=0:
        flagS = 1
    EnergiaNoSum.append(EnergiaNS)
    TiempoNoSumPor98.append(Tiempo98)
    TiempoNoSumPor100.append(finTiempo)
    CostoFalla.append(CostoFallaTotal)
    CostoCamino.append(np.sum(ACostoRT))

path1 = "../Resultados/Procesado/"+NOMBREFILEOUT+" EnergiaNS"

```

```

path2 = "../Resultados/Procesado/"+NOMBREFILEOUT+"_Tiempo98"
path3 = "../Resultados/Procesado/"+NOMBREFILEOUT+"_Tiempo100"
path4 = "../Resultados/Procesado/"+NOMBREFILEOUT+"_CostoF"
path5 = "../Resultados/Procesado/"+NOMBREFILEOUT+"_CostoR"
path6 = "../Resultados/Procesado/"+NOMBREFILEOUT+"_EvENS"
f1=open(path1,"w")
f2=open(path2,"w")
f3=open(path3,"w")
f4=open(path4,"w")
f5=open(path5,"w")
f6=open(path6,"w")

for i in range(SIM):
    f1.write(str(EnergiaNoSum[i])+"\n")
    f2.write(str(TiempoNoSumPor98[i])+"\n")
    f3.write(str(TiempoNoSumPor100[i])+"\n")
    f4.write(str(CostoFalla[i])+"\n")
    f5.write(str(CostoCamino[i])+"\n")
f1.close()
f2.close()
f3.close()
f4.close()
f5.close()

maxtime = 0
for i in range(len(ENSHorMean)):
    t_ev = len(ENSHorMean[i])
    if maxtime<t_ev:
        maxtime=t_ev

for i in range(len(ENSHorMean)):
    while(len(ENSHorMean[i])<maxtime):
        ENSHorMean[i].append(0)

ENSH_Array = np.array(ENSHorMean)
ENSCum_Array = ENSH_Array.cumsum(axis=1)

ENS_H_Prom = (ENSH_Array.sum(axis=0)/SIM).tolist()
ENSCum_Max = ENSCum_Array.max(axis=0).tolist()
ENSCum_Min = ENSCum_Array.min(axis=0).tolist()
ENSCum_Prom = (ENSCum_Array.sum(axis=0)/SIM).tolist()

for i in range(len(ENS_H_Prom)):

f6.write(str(ENS_H_Prom[i])+", "+str(ENSCum_Min[i])+", "+str(ENSCum_Prom[i])+", "+str(ENSCum
_Max[i])+"\n")

f6.close()

```

#### 10.4.6 Procesamiento de Resultados – ProcesamientoZ.py

```

#####
#CARGA DE RESULTADOS
#CASO DE 4 ZONAS - C1
import time
import math
import random

import numpy as np
from operator import itemgetter
from copy import deepcopy

def calcFallaT(Falla, ITrafo, ProrTrafo, SETrafo, EnergiaSE):
    EnergiaNS = 0
    for i in range(len(Falla)):
        indiceTrafo = ITrafo[i]
        EnergiaNS = EnergiaNS +
(Falla[i]/100.0)*(EnergiaSE[SETrafo[indiceTrafo]]*ProrTrafo[indiceTrafo])

```

```

return Energians

#-----Parametros a Modificar-----
NOMBREFILE = "CASO3_Z5/CASO3_Z5"
NOMBREFILEOUT = "CASO3_Z5/C3_Z5"
ZONAS = 5
CUADRILLA_Z = 20
'''
NOMBREFILE = "CASO4/CASO4"
NOMBREFILEOUT = "CASO4/C4"
ZONAS = 4
CUADRILLA_Z = 37

NOMBREFILE = "CASO7/CASO7"
NOMBREFILEOUT = "CASO7/C7"
ZONAS = 5
CUADRILLA_Z = 25

NOMBREFILE = "CASO8/CASO8"
NOMBREFILEOUT = "CASO8/C8"
ZONAS = 5
CUADRILLA_Z = 37
'''

SIM=1000
MES = 2
DIA = 27
HORA = 3
NTRAFOS = 21703

#-----

CLUSTER = ZONAS*CUADRILLA_Z

#Indexadores para moverse en clusters
Z = np.arange(ZONAS)+1
C = (np.arange(ZONAS)+1)*CUADRILLA_Z

SSEE = []
ProrTrafo = []
path5 = "../Recursos/Lite/SSEE"
path6 = "../Recursos/Lite/ProrTrafo"
f5=open(path5,"r")
f6=open(path6,"r")
for i in range(NTRAFOS):
    linea5 = int(f5.readline())
    linea6 = float(f6.readline())
    SSEE.append(linea5)
    ProrTrafo.append(linea6)
f5.close()
f6.close()

TrafoZona = []
path2 = "../Recursos/Lite/TrafosZona"+str(ZONAS)
f2=open(path2,"r")
for i in range(NTRAFOS):
    linea2 = int(f2.readline())
    TrafoZona.append(linea2)
f2.close()

for i in range(ZONAS):
    SSEE.append(0)
    ProrTrafo.append(0)
    TrafoZona.append(0)

CostoTC = []
FallaTC = []

```

```

NodoTC = []
CostoRTC = []

for NCUADRILLA in range(CLUSTER):

    CUADRILLA = "_CUADRILLA_"+str(NCUADRILLA)

    pathT = "../Resultados/Raw/"+NOMBREFILE+"Tiempo"+CUADRILLA
    pathF = "../Resultados/Raw/"+NOMBREFILE+"Falla"+CUADRILLA
    pathTT = "../Resultados/Raw/"+NOMBREFILE+"TrafoTipo"+CUADRILLA
    pathTID = "../Resultados/Raw/"+NOMBREFILE+"TrafoID"+CUADRILLA
    pathTR = "../Resultados/Raw/"+NOMBREFILE+"TiempoRuta"+CUADRILLA
    fT=open(pathT,"r")
    fF=open(pathF,"r")
    fTT=open(pathTT,"r")
    fTID=open(pathTID,"r")
    fTR=open(pathTR,"r")

    CostoT = []
    FallaT = []
    CapT = []
    NodoT = []
    CostoRT = []
    for i in range(SIM):
        Aux = map(float, (fT.readline()).split(","))
        CostoT.append(Aux[1:len(Aux)-1])
        Aux = map(int, map(float, (fF.readline()).split(",")))
        FallaT.append(Aux[1:len(Aux)-1])
        Aux = map(int, map(float, (fTT.readline()).split(",")))
        CapT.append(Aux[1:len(Aux)-1])
        Aux = map(int, map(float, (fTID.readline()).split(",")))
        NodoT.append(Aux[1:len(Aux)-1])
        Aux = map(float, (fTR.readline()).split(","))
        CostoRT.append(Aux[1:len(Aux)-1])
    fT.close()
    fF.close()
    fTT.close()
    fTID.close()
    fTR.close()

    CostoTC.append(CostoT)
    FallaTC.append(FallaT)
    NodoTC.append(NodoT)
    CostoRTC.append(CostoRT)

EnergiaNoSum = []
#Porcentaje de energia no suministrada de acuerdo al tiempo
CostoCamino = []
TrafoZ = []
time_is = time.time()
for CASO in range(SIM):
    print "CASO",CASO, "PROCESADO EN EL MINUTO", round((time.time()-time_is)/60,2)
    EnergiaNSZ = np.zeros(ZONAS).tolist()
    DistanciaZ = np.zeros(ZONAS).tolist()
    TipoTrafoZ = np.zeros((ZONAS,ZONAS+1)).tolist()

    for NCUADRILLA in range(CLUSTER):

        path1 = "../Recursos/Lite/EnergiaNorm.csv"
        f1=open(path1,"r")
        flagS = 0
        flagE = 0
        EnergiaHorizonte = []
        horizonte = math.ceil(CostoTC[NCUADRILLA][CASO][len(CostoTC[NCUADRILLA][CASO])-
1])+1

        flag=0
        for lineal in f1:
            if flag==0:
                flag=1

```

```

        continue
    Aux = map(float, lineal.split(", "))
    if MES==int(Aux[0]) and DIA==int(Aux[1]) and HORA==int(Aux[2]):
        flagS = 1
    if flagS==1:
        EnergiaHorizonte.append(Aux[3:45])
        flagE = flagE+1
    if flagE==horizonte:
        break
f1.close()

ACostoT = CostoTC [NCUADRILLA][CASO]
AFallaT = FallaTC [NCUADRILLA][CASO]
ANodoT = NodoTC [NCUADRILLA][CASO]

t=0
Tiempo = []
ATiempo = 0
EnergiaNS = []
AEnergiaNST = 0
ATiempo = 0
ATiempoP = 0

z = Z [NCUADRILLA<C][0]

while(t<horizonte and ACostoT!=[]):
    EnergiaNST = 0
    AEnergiaNST = calcFallaT(AFallaT, ANodoT, ProrTrafo, SSEE,
EnergiaHorizonte [t])
    ATiempo = t
    ATiempoP = t
    #print "Tiempo: ", t

    flagT=0
    while(flagT==0):
        if ACostoT==[]:
            break
        elif ACostoT!=[] and ACostoT[0]<t+1:
            ATiempo = ACostoT.pop(0)
            indiceTrafo = ANodoT.pop(0)
            indiceFalla = AFallaT.pop(0)
            EnergiaNST = AEnergiaNST*(ATiempo-ATiempoP) + EnergiaNST
            ATiempoP = ATiempo
            AEnergiaNST = AEnergiaNST -
            (indiceFalla/100.0)*(EnergiaHorizonte[t] [SSEE [indiceTrafo]]*ProrTrafo[indiceTrafo])

            TipoTrafoZ [z-1] [TrafoZona [indiceTrafo]]=TipoTrafoZ [z-
1][TrafoZona[indiceTrafo]]+1

            elif ACostoT[0]>=t+1:
                flagT=1
                #SE calcula la energia no suministrada en dicha hora
                EnergiaNST = AEnergiaNST*(t+1-ATiempo) + EnergiaNST
                EnergiaNS.append(EnergiaNST)

            t=t+1
            if ACostoT==[]:
                break

    EnergiaNSZ [z-1]=EnergiaNSZ [z-1] + sum(EnergiaNS)
    DistanciaZ [z-1]=DistanciaZ [z-1] + sum(CostoRTC [NCUADRILLA][CASO])

    EnergiaNoSum.append(EnergiaNSZ)
    CostoCamino.append(DistanciaZ)
    TrafoZ.append(TipoTrafoZ)

path1 = "../Resultados/Procesado/"+NOMBREFILEOUT+"_ZEnergiaNS"
path2 = "../Resultados/Procesado/"+NOMBREFILEOUT+"_ZCostoR"
path3 = "../Resultados/Procesado/"+NOMBREFILEOUT+"_ZTrafoZ"
f1=open(path1,"w")
f2=open(path2,"w")

```

```

f3=open(path3,"w")
for i in range(SIM):
    f1.write(",".join(str(v) for v in EnergiaNoSum[i])+"\n")
    f2.write(",".join(str(v) for v in CostoCamino[i])+"\n")
    f3.write(",".join(str(int(r)) for v in TrafoZ[i] for r in v)+"\n")
f1.close()
f2.close()
f3.close()

```

### 10.4.7 Procesamiento de Resultados – TrafosPros.py

```

import time
import math
import random
import matplotlib.pyplot as pyplot

import numpy as np
from operator import itemgetter
from copy import deepcopy

NOMBRECASO = 'CASO3_NC'
NOMBRETABLA = 'C_NC'
NOMBREFILE = NOMBRECASO+'/' +NOMBRECASO
TiempoDS = 324

SIM=1000

CostoT = []
FallaT = []
CapT = []

pathT = "../Resultados/Raw/"+NOMBREFILE+"Tiempo"
pathF = "../Resultados/Raw/"+NOMBREFILE+"Falla"
pathTT = "../Resultados/Raw/"+NOMBREFILE+"TrafoTipo"
fT=open(pathT,"r")
fF=open(pathF,"r")
fTT=open(pathTT,"r")

for i in range(SIM):
    Aux = map(float, (fT.readline()).split(","))
    CostoT.append(Aux[1:len(Aux)-1])
    Aux = map(int, map(float, (fF.readline()).split(",")))
    FallaT.append(Aux[1:len(Aux)-1])
    Aux = map(int, map(float, (fTT.readline()).split(",")))
    CapT.append(Aux[1:len(Aux)-1])
fT.close()
fF.close()
fTT.close()

CostoT = [item for sublist in CostoT for item in sublist]
FallaT = [item for sublist in FallaT for item in sublist]
CapT = [item for sublist in CapT for item in sublist]

CostoT = np.array(CostoT)
FallaT = np.array(FallaT)
CapT = np.array(CapT)

ds1 = CostoT[np.logical_and(0<CapT, 4==FallaT)]
ds2 = CostoT[np.logical_and(0<CapT, 12==FallaT)]
ds3 = CostoT[np.logical_and(0<CapT, 50==FallaT)]
ds4 = CostoT[np.logical_and(0<CapT, 80==FallaT)]

TAMANO = 1

MaxTime = max([np.amax(ds1), np.amax(ds2), np.amax(ds3), np.amax(ds4)])

```

```

bins=np.arange(0, MaxTime+TAMANO, TAMANO)

counts1, bins1, bars1 = pyplot.hist(ds1, bins, alpha=1, label='Menor', normed=1,
histtype='step', cumulative=True)
counts2, bins2, bars2 = pyplot.hist(ds2, bins, alpha=1, label='Moderado', normed=1,
histtype='step', cumulative=True)
counts3, bins3, bars3 = pyplot.hist(ds3, bins, alpha=1, label='Extensivo', normed=1,
histtype='step', cumulative=True)
counts4, bins4, bars4 = pyplot.hist(ds4, bins, alpha=1, label='Completo', normed=1,
histtype='step', cumulative=True)
pyplot.legend(loc='lower right')
pyplot.title('Probabilidad de que un transformador \n sea reparado en cierta hora -
CASO: '+NOMBRETABLA)
pyplot.xlabel('Tiempo [h]')
pyplot.ylabel('Probabilidad')
pyplot.ylim(ymax = 1.1)
pyplot.xlim(xmax = MaxTime)
pyplot.savefig('./FotosMem/CurvasProb/C_PRT_' +NOMBRETABLA+'.png')

print round(counts1[TiempoDS]*100,2)
print round(counts2[TiempoDS]*100,2)
print round(counts3[TiempoDS]*100,2)
print round(counts4[TiempoDS]*100,2)

```