

**IMPLEMENTACIÓN DE UN MÉTODO DE BÚSQUEDA EXHAUSTIVA PARA LA
RECONFIGURACIÓN DE SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA
ELÉCTRICA**

ING. JUAN RAFAEL CAMACHO MARTÍNEZ

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES
ESPECIALIZACIÓN EN SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA
ELÉCTRICA
BUCARAMANGA**

2016

**IMPLEMENTACIÓN DE UN MÉTODO DE BÚSQUEDA EXHAUSTIVA PARA LA
RECONFIGURACIÓN DE SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA
ELÉCTRICA**

ING. JUAN RAFAEL CAMACHO MARTÍNEZ

Monografía presentada como requisito para optar al título de:

Especialista en Sistemas de Distribución de Energía Eléctrica

Director:

Dr. Gilberto Carrillo Caicedo

Ingeniero Electricista – Especialista en Técnicas de Investigación –

Master of Power Engineering - Doctor en Ingeniería Industrial.

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS

**ESCUELA DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA Y DE
TELECOMUNICACIONES**

**ESPECIALIZACIÓN EN SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA
ELÉCTRICA**

BUCARAMANGA

2016

CONTENIDO

	pág.
INTRODUCCIÓN	12
1 RECONFIGURACIÓN Y EVALUACIÓN DEL ESTADO DE SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA ELÉCTRICA	14
1.1 RECONFIGURACIÓN DE SISTEMAS DE DISTRIBUCIÓN	15
1.2 EVALUACIÓN DEL ESTADO DE SISTEMAS DE DISTRIBUCIÓN	18
1.2.1 Eficiencia de sistemas eléctricos.	18
1.2.2 Porcentaje de regulación de tensión.	19
1.2.3 Flujo de carga en sistemas radiales mediante el método de barrido – suma de corrientes.	19
2 MÉTODOS DE RECONFIGURACIÓN DE SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA ELÉCTRICA	22
2.1 MÉTODOS BASADOS EN EL CONOCIMIENTO	22
2.2 MÉTODOS BASADOS EN TÉCNICAS EVOLUTIVAS	23
2.3 MÉTODOS MIXTOS	23
3 MÉTODO DE RECONFIGURACIÓN IMPLEMENTADO	25
3.1 DETERMINACIÓN EXHAUSTIVA DE LAS COMBINACIONES DE INTERRUPTORES CERRADOS BASADA EN UNA TÉCNICA DE CONTEO	27

3.2	VERIFICACIÓN DE RADIALIDAD Y DE ALIMENTACIÓN DE TODAS LAS BARRAS DEL SISTEMA	33
3.3	DIAGRAMA DE FLUJO DEL ALGORITMO DE CÁLCULO DE FLUJO DE POTENCIA, EFICIENCIA Y REGULACIÓN DE TENSIÓN	37
4	SISTEMAS DE PRUEBA	42
4.1	SISTEMA DE PRUEBA 1, REPRESENTADO POR UN SOLO HILO	43
4.2	SISTEMA DE PRUEBA 2, REPRESENTADO POR TRES HILOS	43
5	RESULTADOS	46
5.1	RESULTADOS PARA EL SISTEMA DE PRUEBA 1	46
5.2	RESULTADOS PARA EL SISTEMA DE PRUEBA 2	49
6	CONCLUSIONES	51
	BIBLIOGRAFÍA	52
	REFERENCIAS BIBLIOGRÁFICAS	54
	ANEXOS	57

LISTA DE FIGURAS

	pág.
Figura 1: Partes generales de un sistema eléctrico tradicional	14
Figura 2: Objetivos y Restricciones de la Reconfiguración	16
Figura 3: Ejemplo de sistema de distribución con tres conexiones al sistema de transmisión	16
Figura 4: Ejemplos de configuraciones para un sistema de distribución	17
Figura 5: Diagrama de flujo del método implementado	26
Figura 6: Diagrama de flujo del proceso para definir las posibles configuraciones	32
Figura 7: Diagrama de flujo del proceso para evaluar si la configuración es radial y alimenta todas las cargas	35
Figura 8: Diagrama de flujo del proceso para evaluar el flujo de carga	38
Figura 9: Representación unifilar de los sistemas de prueba, tomado de [15]	42

LISTA DE TABLAS

	pág.
Tabla 1: Relación beneficio / costo de diferentes técnicas de reducción de pérdidas	19
Tabla 2: Características del sistema de prueba 1	44
Tabla 3: Características del sistema de prueba 2	45
Tabla 4: Resultados para el sistema de prueba 1	47
Tabla 5: Resultados del sistema de prueba 1 obtenidos en [15]	48
Tabla 6: Resultados para el sistema de prueba 2	49
Tabla 7: Configuraciones en las que el flujo de potencia no converge para el sistema trifásico	50

LISTA DE ANEXOS

	pág.
ANEXO A. Código para la determinación de las posibles configuraciones	57
ANEXO B. Código para evaluar la condición de operación radial de las configuraciones	58
ANEXO C. Código para evaluar el flujo de potencias	63
ANEXO D. Código para la entrada de datos y articulación de las funciones en el sistema de prueba 1	66
ANEXO E. Código para la entrada de datos y articulación de las funciones en el sistema de prueba 2	71
ANEXO F. Código para el cálculo de la relación entre las impedancias de línea y de carga.	76

RESUMEN

TÍTULO: IMPLEMENTACIÓN DE UN MÉTODO DE BÚSQUEDA EXHAUSTIVA PARA LA RECONFIGURACIÓN DE SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA ELÉCTRICA

AUTOR: ING. JUAN RAFAEL CAMACHO MARTÍNEZ

PALABRAS CLAVE: Sistemas de distribución, reconfiguración de redes, búsqueda exhaustiva, reducción de pérdidas, operación radial, flujos de potencia.

DESCRIPCIÓN:

La reconfiguración de sistemas de distribución de energía eléctrica se realiza cambiando el estado de dispositivos de apertura o cierre presentes en la red, para modificar la topología de la misma con el fin de reducir las pérdidas de potencia en el sistema. Este trabajo presenta la implementación de un método de búsqueda exhaustiva para encontrar la configuración que maximiza la eficiencia, además se realiza la búsqueda de la configuración que minimiza el porcentaje de regulación de tensión del sistema. El algoritmo implementado establece todas las posibles combinaciones en los estados de los dispositivos de cierre y apertura, que permitan la activación de tantos tramos de línea como puntos de carga tenga el sistema, con lo que en operación radial se garantiza la alimentación a todos los puntos de carga; posteriormente se evalúa para cada posible configuración encontrada la condición de operación radial, descartando aquellas que no lo cumplan; para hallar la eficiencia y el porcentaje de regulación de tensión del sistema en cada configuración aprobada, se implementó el método de barrido – suma de corrientes para evaluar el flujo de potencia; una vez encontrada la eficiencia y el porcentaje de regulación de tensión para cada configuración se determina la que logra el cumplimiento de los objetivos planteados. El algoritmo es probado en un sistema de 33 nodos y 37 tramos de línea.

* Monografía

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Director: Dr. Gilberto Carrillo Caicedo.

ABSTRACT

TITLE: IMPLEMENTATION OF A METHOD OF EXHAUSTIVE SEARCH FOR THE RECONFIGURATION OF A DISTRIBUTION NETWORK.

AUTHOR: ENG. JUAN RAFAEL CAMACHO MARTÍNEZ

KEYWORDS: Distribution systems, network reconfiguration, exhaustive search, loss reduction, radial operation, power flows.

DESCRIPTION:

The network reconfiguration in distribution systems is realized by changing the status of sectionalizing switches on the network, to modify the topology of the same one in order to reduce the power loss in the system. This work presents the implementation of a method of exhaustive search to find the configuration that maximizes the efficiency, in addition there is realized the search of the configuration that minimizes the percentage of voltage regulation of the system. The implemented algorithm establishes all the possible combinations in the conditions of the switching devices, which allows the activation of as many sections of line as points of load the system has, with what in radial operation the supply is guaranteed to all the points of load. Later there is evaluated for every possible opposing configuration the condition of radial operation, rejecting those that do not fulfill it; to find the efficiency and the percentage of voltage regulation of the system in every approved configuration, there was implemented the method of sweep - sum of currents to evaluate the flow of power; once the efficiency is computed, the percentage of voltage regulation for every configuration determines the one that achieves the fulfillment of the raised aims. The algorithm is tested in a system of 33 nodes and 37 sections of line.

* Monograph

** Faculty of Physical-Mechanical Engineerings. School of Electrical Engineering, Electronic and Telecommunication. Specialization in Distribution Systems. Director: PhD. Gilberto Carrillo Caicedo.

INTRODUCCIÓN

Los sistemas de distribución de energía eléctrica están sometidos a múltiples variaciones en las condiciones de operación por causas como fluctuaciones de la demanda y fallas. En todas las circunstancias, es importante mantener el suministro de energía eléctrica a la mayor cantidad de usuarios posible y de la manera más eficiente que se pueda.

Para lograr esto una de las herramientas disponibles es la reconfiguración del sistema, que consiste en la conmutación de interruptores en la red, permitiendo: en condiciones de falla conectar a un alimentador zonas que han quedado sin flujo de energía eléctrica, o en condiciones normales de operación brindar caminos a los flujos de energía de tal forma que la red opere de la manera más eficiente posible. Se debe garantizar para toda configuración la operación radial de los sistemas de distribución, además se requiere la existencia de la infraestructura necesaria, de tal manera que la red esté construida de forma mallada y operada siempre de manera radial.

Cuando es posible configurar el sistema de varias maneras, esas topologías deben ser analizadas para encontrar la óptima, es decir, la que permita el suministro de energía eléctrica a la mayor cantidad de usuarios con la menor cantidad de pérdidas posibles, manteniendo los niveles de tensión dentro de los límites permitidos por el regulador. Para encontrar la configuración óptima se emplean métodos basados en la experiencia, técnicas evolutivas y métodos mixtos [1].

El problema de los métodos de búsqueda exhaustiva en la reconfiguración, consiste en la determinación adecuada del espacio de búsqueda, acotándolo según la restricción de operación radial de los sistemas de distribución, y los interruptores presentes en el sistema, con el fin de descartar rápidamente configuraciones no factibles, permitiendo el hallazgo del óptimo global de la manera más rápida posible.

La presente monografía presenta un método de búsqueda exhaustiva, mostrando su desarrollo e implementación en cinco capítulos, de la siguiente manera:

En el primer capítulo se define la función de reconfiguración de sistemas de distribución y el método de barrido – suma de corrientes para la solución de flujos de potencia de sistema radiales. El flujo de potencias permitirá calcular los niveles de tensión en cada nodo, para la evaluación del porcentaje de regulación; y las potencias en las ramas de la red, para determinar la eficiencia.

El segundo capítulo presenta una síntesis de publicaciones sobre métodos de reconfiguración de sistemas de distribución.

En el tercer capítulo se explica en detalle el método implementado basado en búsqueda exhaustiva, el cual contempla todas las posibles configuraciones que cumplan con las condiciones de operación radial y de alimentación a todos los puntos de carga. Aquí se presentan los algoritmos seguidos para el proceso de reconfiguración, la solución del flujo de potencias y el hallazgo de la configuración óptima.

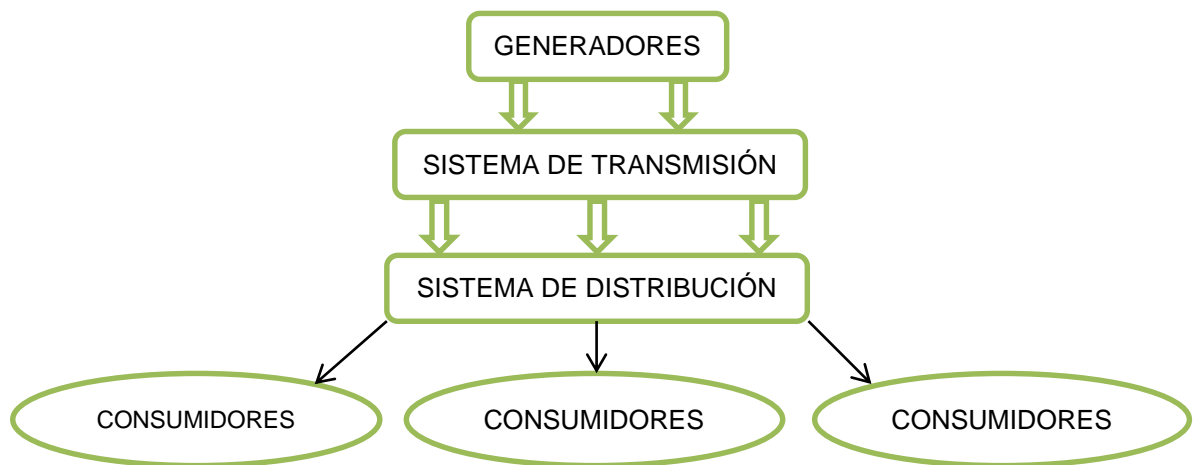
El cuarto capítulo muestra las características de los sistemas de distribución usados para las pruebas del método implementado, los cuales están constituidos por 33 barras y 37 secciones de líneas, todas con posibilidad de abrirse o cerrarse.

En el quinto capítulo se presentan los resultados obtenidos para los sistemas de prueba utilizados.

1 RECONFIGURACIÓN Y EVALUACIÓN DEL ESTADO DE SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA ELÉCTRICA

El propósito de un sistema de energía eléctrica es llevar la energía desde las fuentes energéticas hasta los puntos de consumo, el sistema se puede dividir en: generadores, sistema de transmisión, sistema de distribución y usuarios. La Figura 1 muestra la conformación general de un sistema eléctrico tradicional, donde: los generadores producen energía eléctrica a partir de otra forma de energía, tradicionalmente se encuentran ubicados lejos de los usuarios; el sistema de transmisión se encarga de la transferencia de grandes bloques de energía eléctrica desde las centrales de generación hasta las áreas de consumo; el sistema de distribución recibe la energía del sistema de transmisión y la transfiere a los consumidores dentro de un área específica; finalmente los usuarios consumen esta energía para su beneficio [2] [3].

Figura 1: Partes generales de un sistema eléctrico tradicional



Los sistemas de distribución pueden contar con distintas rutas para llevar la energía desde el sistema de transmisión a cada uno de los usuarios, es decir pueden estar contruidos de forma mallada. Sin embargo, típicamente el suministro de energía a cada usuario se realiza desde una única conexión al sistema de transmisión (considerados nodos “*slack*”), lo que implica una operación radial del sistema. La

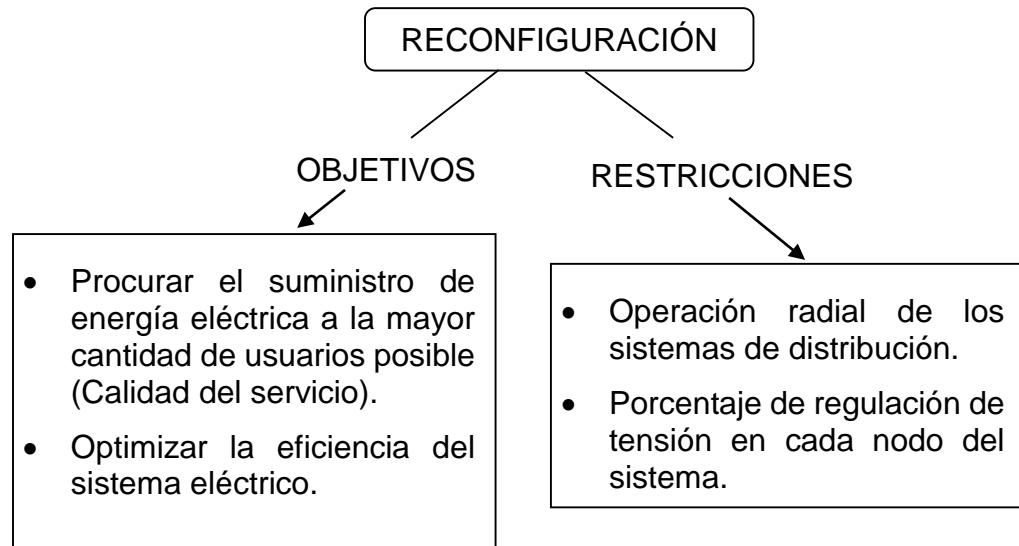
construcción mallada de la red de distribución brinda la posibilidad de escoger entre varias trayectorias para los flujos de potencia, esto se logra abriendo y/o cerrando interruptores o seccionadores que deben estar presentes en el sistema. Cada conjunto de decisiones de abrir o cerrar interruptores corresponde a una configuración del sistema, estas configuraciones presentan sus propias características al respecto de eficiencia y regulación de tensión; dependiendo de: la energía demandada en los puntos de consumo, niveles de tensión en los nodos de conexión al sistema de transmisión y los parámetros de los elementos de conexión entre cada par de barras del sistema.

1.1 RECONFIGURACIÓN DE SISTEMAS DE DISTRIBUCIÓN

“La reconfiguración de alimentadores es una función que permite reducir pérdidas y mejorar la calidad del servicio alterando la estructura topológica de la red, y puede ser efectuada a través de interruptores manuales o con mando remoto. Se utiliza como una herramienta de planeación (simulación) o de control en tiempo real (operación)” [4].

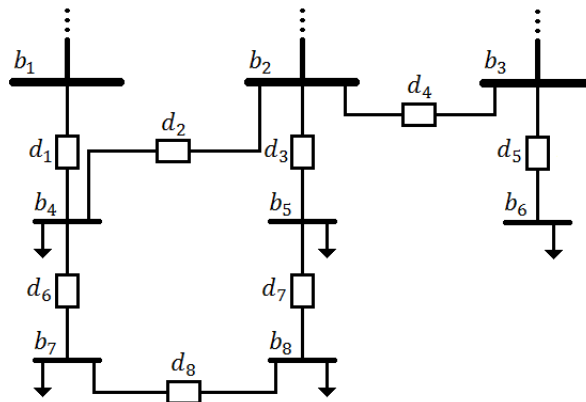
La Figura 2 muestra los objetivos que se buscan con la reconfiguración de sistemas de distribución y las restricciones que se deben tener en cuenta para cada topología probada o implementada.

Figura 2: Objetivos y Restricciones de la Reconfiguración



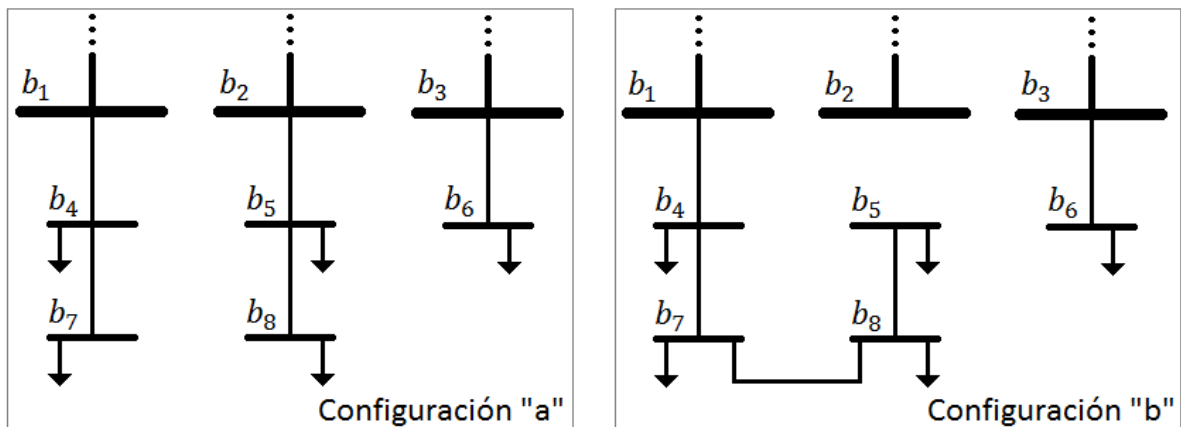
La Figura 3 muestra un sistema de distribución que cuenta con ocho barras (de la b_1 a la b_8), de las cuales b_1 , b_2 y b_3 son puntos de conexión al sistema de transmisión, de la barra b_4 a b_8 corresponden a puntos de consumo y paso de energía eléctrica; este ejemplo muestra una red de distribución mallada, sin embargo cuenta con un interruptor en cada línea (del d_1 al d_8) para permitir o no que la energía eléctrica fluya a través de ellas, lo que brinda la posibilidad de configurar el sistema de varias maneras, conservando una operación radial.

Figura 3: Ejemplo de sistema de distribución con tres conexiones al sistema de transmisión



Por ejemplo: si en el sistema mostrado en la Figura 3 se cierran todos los interruptores con excepción de d_2 , d_4 y d_8 se obtiene la configuración mostrada en la parte izquierda de la Figura 4 (configuración "a"); pero si los interruptores abiertos son d_2 , d_3 y d_4 se obtiene la configuración mostrada a la derecha de la Figura 4 (configuración "b"). Es importante resaltar que el interruptor d_4 solo debe cerrarse garantizando la ausencia de tensión desde el sistema de transmisión en la barra b_2 o b_3 , para lo cual deben existir interruptores a la llegada de dichas barras.

Figura 4: Ejemplos de configuraciones para un sistema de distribución



Las configuraciones mostradas en la Figura 4 son solo dos ejemplos de varias posibilidades que se tienen, cada una de estas configuraciones ofrecerá regulaciones de tensión y eficiencias distintas para el sistema, por lo tanto el problema de reconfiguración consiste en obtener la topología que maximice la eficiencia manteniendo la regulación de tensión dentro del rango regulatoriamente permitido.

1.2 EVALUACIÓN DEL ESTADO DE SISTEMAS DE DISTRIBUCIÓN

Para comparar objetivamente las distintas configuraciones en las que puede operar un sistema de distribución se requiere calcular el porcentaje de regulación de tensión en cada nodo y la eficiencia del sistema para todas las configuraciones analizadas, lo que requiere la solución del flujo de carga. Estos conceptos se describen a continuación.

1.2.1 Eficiencia de sistemas eléctricos. La eficiencia de un sistema eléctrico es la razón entre la energía útil y la energía que ha sido necesaria inyectar al sistema para entregar dicha energía en los puntos de consumo. Dado que la energía eléctrica es el producto comercializado en el mercado eléctrico, una red poco eficiente no es rentable para la sociedad; por lo tanto se debe procurar una configuración eficiente de la red, es decir, con la menor cantidad de pérdidas posible.

Existen varios métodos que buscan la reducción de las pérdidas técnicas. La Tabla 1 presenta la relación beneficio/costo de esos métodos, evidenciando la conveniente implementación de la reconfiguración en búsqueda de sistemas con mayor eficiencia [4].

Tabla 1: Relación beneficio / costo de diferentes técnicas de reducción de pérdidas

MÉTODO DE REDUCCIÓN DE PÉRDIDAS	BENEFICIO/COSTO
Ubicación de condensadores	2 a 8
Cambio de conductores	0.6 a 7
Elevación de los niveles de tensión de servicio	1.5 a 3
Reconfiguración	Mayor de 13

Fuente: [4]

1.2.2 Porcentaje de regulación de tensión. El nivel de tensión en un sistema eléctrico es un indicador del equilibrio en la potencia reactiva; es decir, mientras más balanceada se encuentre la potencia reactiva – inductiva y capacitiva - más estable será el nivel de tensión. Además los equipos de los usuarios se encuentran diseñados para algún nivel de tensión específico; por lo tanto, no se debe permitir que la tensión suministrada se salga de ciertos rangos con respecto del valor nominal - esos rangos están establecidos en la norma NTC 1340, que para los sistemas de distribución local (SDL) debe ser cumplida por disposición de la CREG [5]. En consecuencia, una de las restricciones en la reconfiguración la determinan los niveles de tensión de cada nodo; ya que la configuración que se establezca en el sistema debe garantizar el cumplimiento de la reglamentación.

1.2.3 Flujo de carga en sistemas radiales mediante el método de barrido – suma de corrientes. Para determinar los niveles de tensión en cada barra, las corrientes en cada una de las líneas, y permitir los cálculos del porcentaje de regulación de tensión y eficiencia del sistema, se requiere la evaluación del flujo de carga. Los métodos tradicionales para la solución del problema del flujo de carga en

sistemas eléctricos presentan un buen comportamiento en sistemas de transmisión. Sin embargo cuando se aplican a las redes de distribución típicas, caracterizadas por su operación radial y alta relación r/x , emplean demasiado tiempo en la solución o tienen problemas de convergencia, por estas razones se han desarrollado métodos adaptados a las características de las redes de distribución, como el método de barrido – suma de corrientes [6].

Este método consiste en calcular las corrientes en cada una de las líneas de una trayectoria radial para posteriormente calcular las tensiones en las barras, mediante la aplicación de la primera y segunda ley de Kirchhoff respectivamente; los cálculos de las corrientes se realizan iniciando por el nodo más lejano al punto de conexión con el sistema de transmisión, mientras que los cálculos de las tensiones se realizan en sentido contrario, es decir desde el primero hasta el último nodo de la trayectoria radial [7].

Como información de la red se debe conocer: la forma en que se conectan las barras (la configuración), el modelo de las cargas presentes en cada una de las barras, la tensión en los nodos alimentadores de la red (puntos de conexión al sistema de transmisión), y las impedancias de las líneas. Como punto de partida se asume en todos los nodos de la red la tensión presente en el nodo alimentador, posteriormente se realizan de manera iterativa los siguientes pasos:

- Se calculan las corrientes que solicitan las cargas en cada nodo, teniendo en cuenta el modelo de las cargas y las tensiones en las barras obtenidas en la iteración anterior.
- Iniciando con las ramas que terminan en las barras ubicadas en las colas de las trayectorias, se calculan las corrientes aplicando la primera ley de Kirchhoff, hasta llegar a las ramas que inician en los nodos alimentadores (“*slack*”).
- Se calculan las tensiones en las barras, aplicando la segunda ley de Kirchhoff y teniendo en cuenta que las caídas de tensión en las líneas se obtienen como el producto de las corrientes obtenidas en el paso anterior y las impedancias de las ramas (según la ley de Ohm). Este proceso debe iniciar con las barras que

reciben potencia directamente del nodo alimentador, y terminar en las barras ubicadas en las colas de la red.

- Se evalúa la convergencia teniendo en cuenta el error seleccionado, si algún valor se encuentra por encima de dicha tolerancia se repite el proceso, caso contrario se asumen los resultados de la última iteración como los valores de corriente por las ramas y tensiones en las barras.

Una vez obtenidas las tensiones y corrientes con sus respectivos ángulos, es posible calcular el porcentaje de regulación de tensión en cada una de las barras y la eficiencia del sistema. Este dato es utilizado luego para determinar las mejores opciones de configuración de la red.

2 MÉTODOS DE RECONFIGURACIÓN DE SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA ELÉCTRICA

Para encontrar la configuración del sistema que optimice la eficiencia sin violar restricciones de tensión y operación radial, se requiere la evaluación del sistema para todas las posibles combinaciones de los estados de los interruptores presentes en el sistema, con lo que se podría afirmar que se ha encontrado la mejor configuración posible con la infraestructura tenida en cuenta. En el trabajo “An Efficient Brute-Force Solution to the Network Reconfiguration Problem” [8] se presenta la implementación de una búsqueda exhaustiva que obtiene resultados en un tiempo de dos horas para un sistema de 33 barras y 37 secciones de líneas; lapso de tiempo bastante elevado para la operación de sistemas eléctricos.

En un sistema de distribución práctico, la gran cantidad de posibles configuraciones que se pueden tener hace que se requiera mucho tiempo para la realización de esta búsqueda; por esta razón han surgido métodos no exhaustivos de reconfiguración, que básicamente se constituyen en procedimientos para realizar la búsqueda de la configuración óptima de una manera más rápida, es decir sin necesidad de evaluar todas las posibles configuraciones del sistema. Los métodos de reconfiguración se pueden clasificar en tres grupos: basados en el conocimiento, basados en técnicas evolutivas y métodos mixtos [1].

2.1 MÉTODOS BASADOS EN EL CONOCIMIENTO

En este tipo de métodos se busca optimizar la operación de la red hallando una nueva configuración del sistema mediante algoritmos que se fundamentan en la experiencia de los operadores [1]. Como ejemplos de este tipo de método se tiene: La aplicación de la lógica difusa en la solución del problema de reconfiguración. Este método ha sido aplicado en trabajos como: “Logic Programming and Fuzzy

Monte Carlo for Distribution Network Reconfiguration” [9] y “Reconfiguration of Distribution Network with DG Using Fuzzy Multi-objective Method” [10].

Método heurístico basado en el intercambio de rama, el cual consiste en actuar sólo sobre un par de interruptores por cada cambio de configuración, abriendo uno y cerrando el otro; garantizando el suministro a todos los puntos de consumo y la operación radial del sistema - evaluando mediante flujos de carga las condiciones de operación de la nueva configuración, para encontrar la más adecuada [1].

2.2 MÉTODOS BASADOS EN TÉCNICAS EVOLUTIVAS

Este tipo de método inicia con una solución y la mejora iterativamente emulando procesos físicos, biológicos y neurológicos presentes en la naturaleza. Se caracterizan por la falta de una formulación matemática rigurosa [1]. Como ejemplos de este tipo de método se tiene:

Algoritmos genéticos: “Smart Grid Reconfiguration Using Simple Genetic Algorithm and NSGA-II” [11].

Optimización mediante enjambre de partículas: “Application of Enhanced Integer Coded Particle Swarm Optimization for Distribution System Feeder Reconfiguration” [12].

Red neuronal: “A Modified Artificial Neural Network Based Distribution System Reconfiguration for Loss Minimization” [13].

2.3 MÉTODOS MIXTOS

Estos métodos son combinaciones de los dos tipos anteriores, en busca de mezclar las potencialidades de esas dos técnicas de optimización y lograr mejores resultados [1].

Los métodos de reconfiguración presentados anteriormente logran mejorar la eficiencia de la red de distribución, sin embargo, no garantizan el hallazgo del óptimo global debido a que no realizan la evaluación de todas las posibles configuraciones del sistema. Para asegurar que se obtiene la configuración más eficiente se requiere la implementación de búsqueda exhaustiva, lo que presenta grandes limitaciones de tiempo para la operación de sistemas de distribución de energía eléctrica.

Debido a la gran experiencia de los operarios de los sistemas de distribución, y a la gran confianza en ello, los métodos basados en el conocimiento han sido tradicionalmente estudiados e implementados, sin embargo, la tendencia a la emulación de procesos naturales en diferentes áreas de la investigación para la optimización, ha abierto una nueva alternativa para la reconfiguración de sistemas de distribución de energía eléctrica.

3 MÉTODO DE RECONFIGURACIÓN IMPLEMENTADO

Con el propósito de asegurar la evaluación de todas las posibles configuraciones del sistema, y por lo tanto, garantizar que se encuentre el óptimo global, se realizará una búsqueda exhaustiva mediante la implementación de un algoritmo que contemple todas las posibles combinaciones en los estados de los interruptores existentes en el sistema. Dado que la implementación de este método implica tiempos de búsqueda bastante altos, no es una herramienta útil en la operación de sistemas grandes, sin embargo, permitirá dos cosas muy importantes:

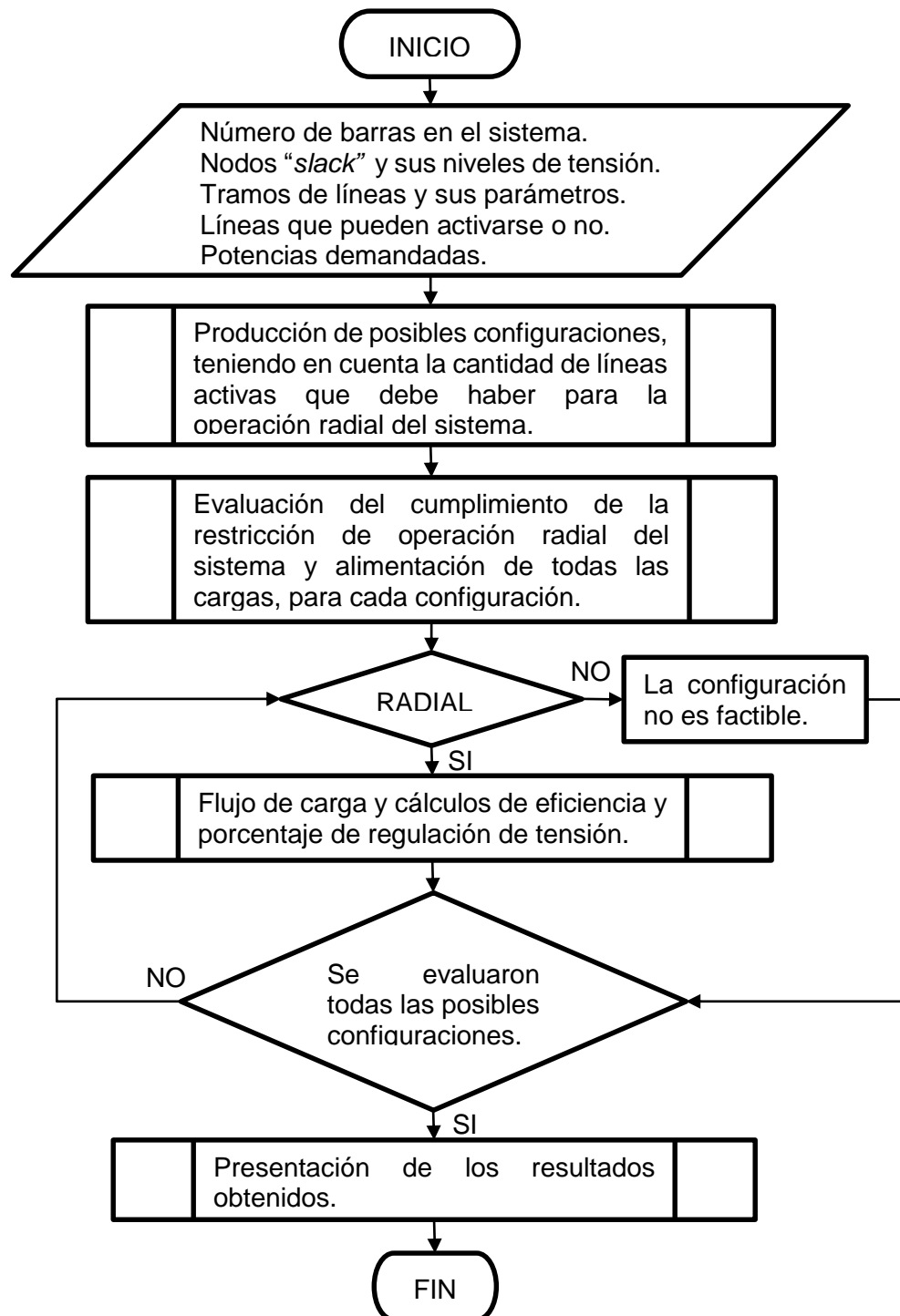
- Apoyar los procesos de expansión de las redes de distribución.
- Evaluar la eficacia en la búsqueda de la configuración óptima de sistemas de distribución al usar métodos más rápidos.

El diagrama de flujo del método implementado se presenta en la Figura 5, la composición de las variables de entrada necesarias se explicará en las siguientes secciones a medida que se vayan introduciendo en los algoritmos individuales de cada etapa.

El algoritmo implementado consta de cuatro etapas principales, descritas a continuación:

- Determinación de las combinaciones de interruptores cerrados sobre las que se efectuará la búsqueda.
- Evaluación del cumplimiento de la condición radial y alimentación total de cada una de las combinaciones encontradas.
- Evaluación del flujo de carga de cada una de las configuraciones radiales encontradas y cálculos de eficiencia y porcentaje de regulación de tensión.
- Presentación de resultados.

Figura 5: Diagrama de flujo del método implementado



3.1 DETERMINACIÓN EXAHUSTIVA DE LAS COMBINACIONES DE INTERRUPTORES CERRADOS BASADA EN UNA TÉCNICA DE CONTEO

En esta etapa se determinarán las posibles configuraciones del sistema, de acuerdo con la cantidad de líneas activas para garantizar el suministro de energía a todos los puntos de consumo, y la operación radial del sistema.

Sin importar la cantidad de nodos que tenga un sistema de distribución (nodos “*slack*” más nodos de carga o de paso de potencia), si se desea mantener la operación radial, a cada nodo de carga debe llegar una única línea que surta potencia, lo que implica que para cada configuración se debe cerrar sólo el número de interruptores necesarios para que la cantidad de líneas activas sea igual al número de barras de carga o de paso presentes en el sistema. Teniendo en cuenta las líneas que no tienen posibilidad de activarse o desactivarse, el número de interruptores que debe cerrarse para garantizar la operación radial de un sistema de “ N_T ” nodos, se puede determinar mediante la aplicación de la ecuación (1).

Una vez obtenida la cantidad de interruptores que se deben cerrar, para determinar la cantidad de posibles configuraciones con este número de interruptores cerrados, basta con calcular el número de combinaciones con esa cantidad de elementos que se pueden extraer de la totalidad de interruptores presentes en el sistema. Este cálculo se puede realizar utilizando técnicas de conteo, aplicando la ecuación (2) [14].

$$r = N_T - N_s - Lin^1 \quad (1)$$

$$Co_1 = \frac{D_T!}{(D_T - r)! r!} \quad (2)$$

En donde:

r : es la cantidad de interruptores que deben estar cerrados para garantizar la operación radial del sistema.

N_T : es el número total de nodos del sistema.

N_s : es el número de nodos “*slack*” del sistema.

Lin^1 : es el número de líneas sin interruptores que permitan abrirlas o cerrarlas, es decir que siempre están activas.

Co_1 : es la cantidad de posibles configuraciones que tienen “ r ” interruptores cerrados.

D_T : es la cantidad total de interruptores presentes en el sistema de distribución.

Ahora es necesario determinar cada una de las “ Co_1 ” combinaciones de interruptores cerrados, del total de “ D_T ” interruptores existentes en el sistema. Con esto termina esta etapa de la metodología implementada.

Para determinar cada una de las “ Co_1 ” combinaciones de interruptores cerrados, se ha implementado el siguiente algoritmo:

En forma general, para obtener cada una de las posibles combinaciones, al seleccionar “ r ” elementos de un conjunto “ N ” de “ n ” elementos, se puede observar cada combinación como un conjunto “ R^k ” de “ r ”, en donde el superíndice “ k ” varía desde uno hasta el número “ K ” de combinaciones que se pueden obtener; tal como se muestra en el conjunto de ecuaciones (3).

$$\begin{aligned}
 N &= \{A_1, A_2, \dots, A_{n-1}, A_n\} \\
 R^1 &= \{B_1^1, B_2^1, \dots, B_{r-1}^1, B_r^1\} \\
 R^2 &= \{B_1^2, B_2^2, \dots, B_{r-1}^2, B_r^2\} \\
 &\vdots \\
 R^{K-1} &= \{B_1^{K-1}, B_2^{K-1}, \dots, B_{r-1}^{K-1}, B_r^{K-1}\} \\
 R^K &= \{B_1^K, B_2^K, \dots, B_{r-1}^K, B_r^K\}
 \end{aligned} \tag{ 3 }$$

De manera sistemática y similar a como se generan los números de un sistema numérico, se pueden obtener cada uno de los conjuntos “ R^k ” de la siguiente forma:

- El conjunto “ R^1 ” se compone por los primeros “ r ” elementos del conjunto “ N ”.

- Se obtienen los conjuntos " R^k " consecutivos desde el obtenido anteriormente, haciendo variar únicamente " B_r^k " desde el elemento " A_{r+1} " hasta el elemento " A_n ".
- Una vez el elemento " B_r^k " selecciona el elemento " A_n " se incrementa en uno la selección del elemento " B_{r-1}^k ", y se hace variar el elemento " B_r^k " desde el elemento seleccionado por " B_{r-1}^k " más uno, hasta el elemento " A_n ".
- Se repite el paso anterior hasta que el elemento " B_{r-1}^k " selecciona el elemento " A_{n-1} ", situación en la cual para obtener la siguiente combinación se incrementa en uno la selección del elemento " B_{r-2}^k ".
- De manera homóloga se sigue procediendo hasta llegar al conjunto " R^K " en el que sus elementos corresponden a los últimos " r " elementos del conjunto " N ".

Se puede concluir que el elemento " B_i^k " (en donde $1 \leq i \leq r$) varía desde el elemento " A_i " hasta " $A_{i+(n-r)}$ ".

Ejemplo: Encontrar las combinaciones producidas al seleccionar tres elementos de un conjunto de cinco elementos.

Se tiene $N = \{A_1, A_2, A_3, A_4, A_5\}$; $n = 5$; $r = 3$; aplicando la ecuación (1) se obtiene:

$$\# \text{ de combinaciones} = \frac{5!}{(5-3)!*3!} = 10.$$

El elemento " B_1^k ": varía desde " A_1 " hasta " A_3 ".

El elemento " B_2^k ": varía desde " A_2 " hasta " A_4 ".

El elemento " B_3^k ": varía desde " A_3 " hasta " A_5 ".

El conjunto $R^1 = \{A_1, A_2, A_3\}$, que corresponden a los tres primeros elementos del conjunto " N ".

Ahora se hace variar el elemento " B_3^k " obteniendo:

$$R^2 = \{A_1, A_2, A_4\}, R^3 = \{A_1, A_2, A_5\}.$$

Incrementando en uno la selección de " B_2^k " y haciendo variar " B_3^k " desde " A_4 " hasta " A_5 ", se obtiene: $R^4 = \{A_1, A_3, A_4\}$, $R^5 = \{A_1, A_3, A_5\}$.

Nuevamente se incrementa en uno la selección de “ B_2^k ” y se hace variar “ B_3^k ” pero ahora desde “ A_5 ”, obteniendo: $R^6 = \{A_1, A_4, A_5\}$.

Dado que el elemento “ B_2^k ” llegó a su límite superior, ahora se incrementa en uno la selección de “ B_1^k ” y se hacen variar los otros elementos, obteniendo:

$$R^7 = \{A_2, A_3, A_4\}, R^8 = \{A_2, A_3, A_5\}, R^9 = \{A_2, A_4, A_5\}.$$

Nuevamente se incrementa en uno la selección de “ B_1^k ” obteniendo:

$R^{10} = \{A_3, A_4, A_5\}$, que corresponde a la décima y última combinación, tal como predijo la aplicación de la ecuación (1).

En resumen las diez combinaciones son:

$$R^k = \begin{bmatrix} A_1, A_2, A_3 \\ A_1, A_2, A_4 \\ A_1, A_2, A_5 \\ A_1, A_3, A_4 \\ A_1, A_3, A_5 \\ A_1, A_4, A_5 \\ A_2, A_3, A_4 \\ A_2, A_3, A_5 \\ A_2, A_4, A_5 \\ A_3, A_4, A_5 \end{bmatrix}$$

Se puede concluir que cada elemento de los conjuntos “ R^k ” varía sobre una cantidad determinada de elementos del conjunto “ N ”, esta cantidad se puede obtener aplicando la ecuación (4); de este modo se tiene que el elemento “ B_i^k ” (donde el subíndice “ i ” varía desde uno hasta “ r ”) varía desde el elemento “ A_i ” hasta el elemento “ $A_{i+ref-1}$ ”.

$$ref = n - r + 1 \quad (4)$$

En donde:

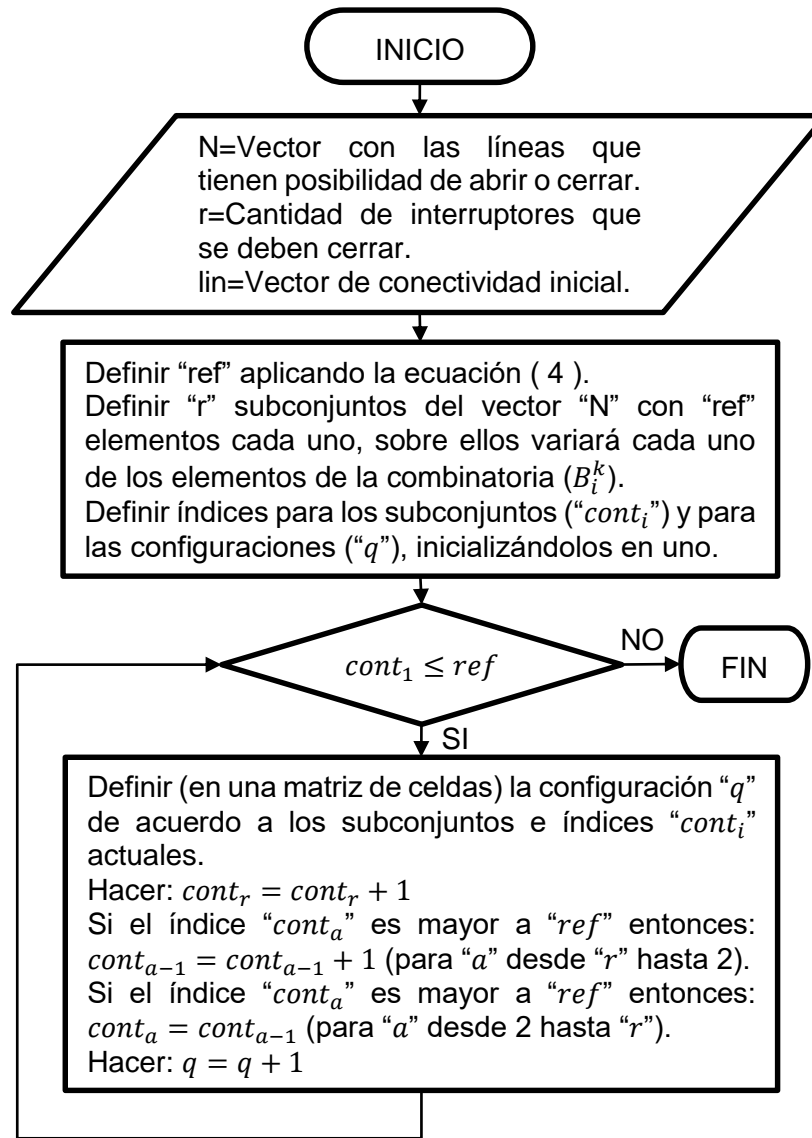
ref : corresponde a la cantidad de elementos del conjunto “ N ” sobre los que varían los elementos de los conjuntos “ R^k ”.

En la Figura 6 se muestra el algoritmo para obtener las " Co_1 " configuraciones del sistema. El código en el lenguaje de programación de Matlab como una función llamada "combinatoria" se presenta en el anexo A.

Esta función requiere como datos de entrada:

- Un vector ("N") que contenga los números que identifican las líneas que pueden ser abiertas o cerradas (las que cuenten con algún dispositivo de apertura y cierre; por ejemplo algún interruptor o seccionador).
- La cantidad de " r " interruptores que deben cerrarse para obtener una configuración con igual número de tramos de línea activos que de barras de carga y/o de paso de potencia (esto se obtiene mediante la ecuación (1).
- Un vector ("lin") con la configuración inicial del sistema, es decir, un vector de tantas posiciones como tramos de línea haya en el sistema, identificando con un uno las líneas que se encuentren activas y con cero las que no lo estén (en el caso inicial solo deben estar en uno las que no cuenten con dispositivo de cierre y apertura).

Figura 6: Diagrama de flujo del proceso para definir las posibles configuraciones



Después de la ejecución de este algoritmo se tendrán " Co_1 " vectores de configuración en una matriz de celdas, compuesta por " Co_1 " vectores de tantas posiciones como tramos de línea tenga el sistema cada uno, y que contienen la configuración; cada tramo de líneas de asocia a una posición en el vector, se indica con uno en alguna posición si el tramo correspondiente se activa, y con cero si no se activa.

3.2 VERIFICACIÓN DE RADIALIDAD Y DE ALIMENTACIÓN DE TODAS LAS BARRAS DEL SISTEMA

Una vez obtenidas las " Co_1 " configuraciones en la etapa anterior, es necesario evaluar cada una de ellas para chequear si:

- Constituye una configuración radial del sistema, se debe realizar este procedimiento ya que el criterio combinatorio usado para definir las configuraciones producirá unas radiales y otras que no lo son.
- Se alimentan todos los nodos del sistema.

El algoritmo implementado para realizar la evaluación del cumplimiento de operación radial y alimentación total consiste en lo siguiente:

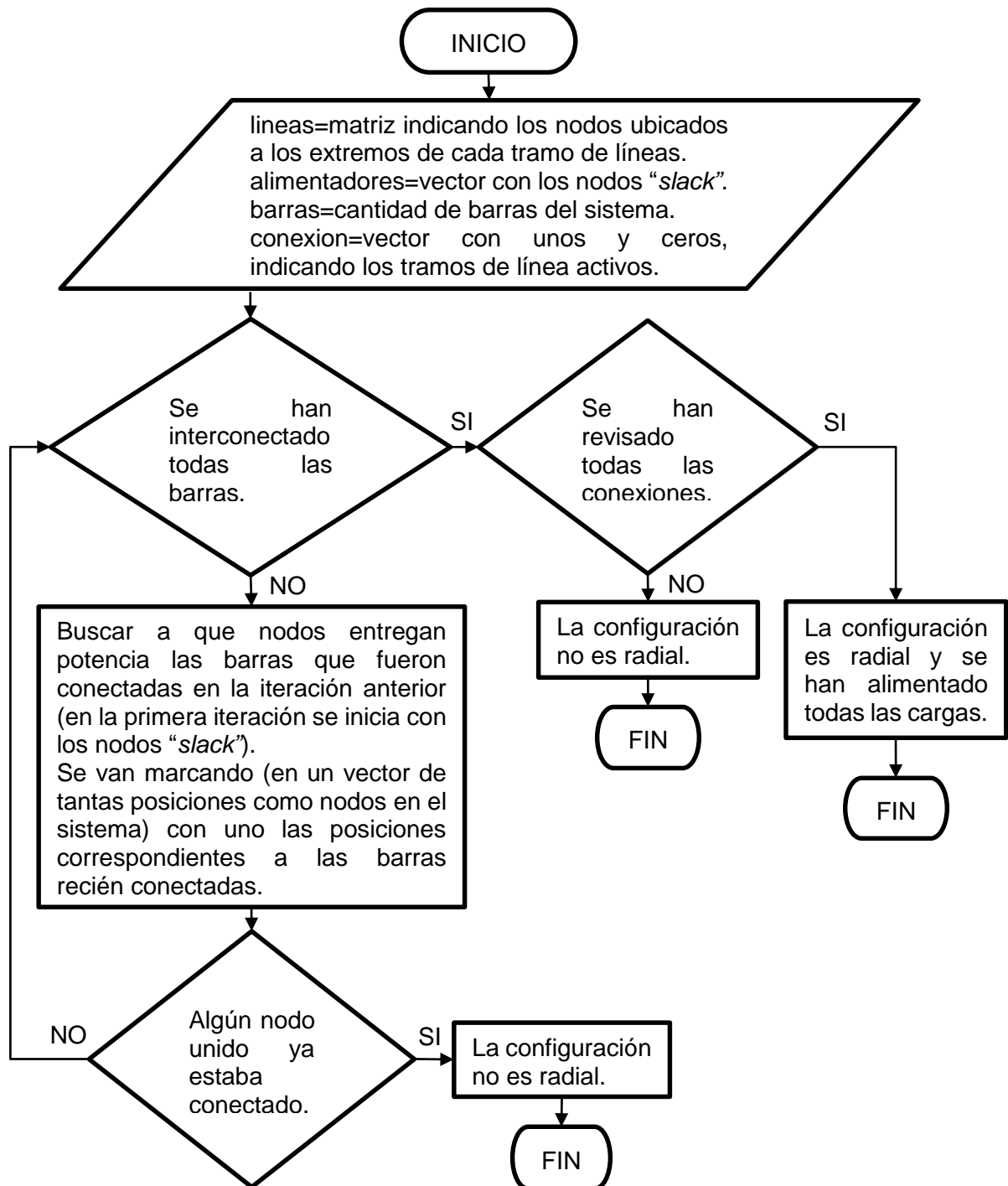
- Partiendo de los nodos "*slack*" verificar a cual nodo se entrega potencia, teniendo en cuenta el vector de tramos de líneas abiertas y cerradas, y marcando los nodos que van siendo conectados (en un vector de tantas posiciones como barras en el sistema e inicializado en ceros), cambiando a uno el elemento que corresponde a la posición de la barra conectada.
- El paso anterior se repite, buscando ahora a cuáles nodos se les entrega potencia desde los que la recibieron en la anterior iteración.
- Si se encuentra que se conecta algún nodo que previamente se había conectado, entonces, la configuración no es radial y se termina la búsqueda.
- Cuando se hayan interconectado todas la barras de consumo y/o paso de potencia desde alguno de los nodos "*slack*", y se hayan tenido en cuenta todos los tramos de línea activas indicadas en el vector de entrada, se termina el chequeo confirmando que la configuración es radial.

En la Figura 7 se muestra el algoritmo para obtener las configuraciones que cumplen la condición de operación radial y alimentan todas las cargas. El código en el lenguaje de programación de Matlab como una función llamada "radialidad" se presenta en el anexo B.

Esta función requiere como datos de entrada:

- Una matriz (“líneas”) en la que se indican los nodos ubicados a los extremos de cada tramo de línea en el sistema. Esta matriz cuenta con tantas filas como tramos tenga el sistema y cada fila cuenta con tres elementos (la matriz tiene tres columnas), en la posición uno y tres de cada fila va el número que identifica a uno de los dos nodos extremos y en la segunda posición va el número correspondiente al nodo presente en el otro extremo. Esta descripción con tres elementos permite que ubicándose en la primera o segunda posición de la fila, con tan solo sumar uno al índice de ubicación, referirse al otro extremo del tramo de línea en cuestión.
- Un vector (“alimentadores”) en el que se colocan los números que identifican los nodos slack del sistema.
- Una variable (“barras”) en la que se indica la cantidad de nodos o barras que tiene el sistema. Esta variable al igual que “líneas” y “alimentadores” son declaradas e ingresadas a la función como variables globales.
- Un vector (“conexión”) en el que se indiquen los tramos de línea que se encuentran activos para la configuración evaluada (ésta es la configuración del sistema), este vector cuenta con tantas posiciones como tramos haya en el sistema, se llenan con uno las posiciones que corresponden a los tramos activos y con cero aquellas que corresponden a los tramos no activos para la configuración en cuestión. Esta información proviene de la etapa anterior. Esta variable ingresa como argumento de la función.

Figura 7: Diagrama de flujo del proceso para evaluar si la configuración es radial y alimenta todas las cargas



Para simplificar el diagrama de flujo se han omitido algunas actividades que se realizan en esta función, pero después de la ejecución de la misma se obtienen seis variables que contienen la siguiente información:

- Salida1: es un vector fila de tres elementos, el primero es dicotómico siendo "1" si la configuración es radial o "0" si no lo es; el segundo puede ser "0" si no se alimentan todas las barras de carga del sistema, "1" si están conectadas todas o "2" si la(s) barra(s) que falta(n) por conectarse es(son) nodos "*slack*"; el tercer elemento indica cuantas veces fue necesario chequear conexiones para producir un resultado útil (número de iteraciones).
- Salida2: es un vector fila de tantos elementos como barras haya en el sistema, cada posición corresponde a una barra y el número allí contenido indica de que barra recibe potencia. Si el número es "0" significa que esa barra es un nodo "*slack*".
- Salida3: es un vector fila en el que se encuentran las barras alimentadoras ("*slack*") y posteriormente aquellas en las que se producen bifurcaciones en la topología de la configuración.
- Salida4: es un vector de celdas (con tantas celdas como elementos tiene el vector de la "salida3") en el cual se guardan las barras a las cuales entrega potencia el correspondiente elemento de "salida3".
- Salida5: es un vector fila en el que se tienen las barras finales de las trayectorias radiales del sistema.
- Salida6: es un vector de celdas (de tantas celdas como barras haya en el sistema), cada posición corresponde a una barra y el vector allí contenido indica a que barra(s) le entrega potencia. Si el número es "0" significa que esa barra finaliza una trayectoria radial.

Una vez obtenida esta información, se correrá el flujo de potencia sólo a las configuraciones en las que se encontró cumplimiento a la condición de operación radial y alimentación de todas las cargas (salida1). La información contenida en las salidas de 2 a 6 es insumo básico para correr el flujo de potencia.

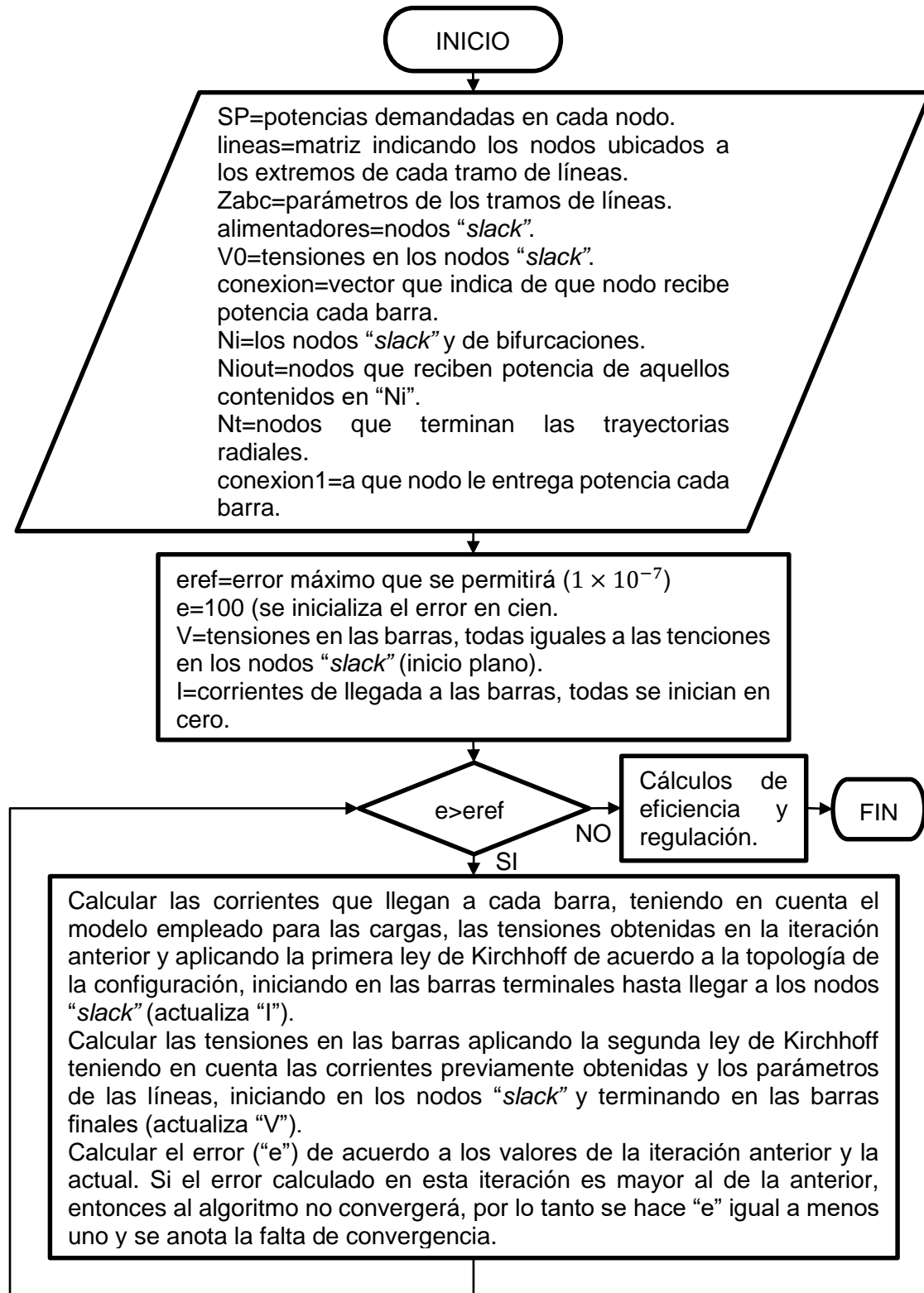
3.3 DIAGRAMA DE FLUJO DEL ALGORITMO DE CÁLCULO DE FLUJO DE POTENCIA, EFICIENCIA Y REGULACIÓN DE TENSIÓN

Para lograr la comparación objetiva de las posibles configuraciones del sistema se debe determinar la eficiencia y porcentaje de regulación de tensión, para lo cual es indispensable conocer los niveles de tensión en las barras y las corrientes a través de las líneas (como se indica en la sección 1.2.3). Este proceso se realiza mediante la evaluación del flujo de carga utilizando el método de barrido.

En la Figura 8 se muestra el algoritmo para evaluar el flujo de potencia de las configuraciones encontradas, este algoritmo se aplica sólo a aquellas que son radiales y alimentan todas las cargas (información contenida en la salida1 de la función “radialidad”). El código en el lenguaje de programación de Matlab como una función llamada “flujos” se presenta en el anexo C. Esta función requiere como datos de entrada:

- Un vector (“SP”) con las potencias demandadas en cada barra del sistema.
- Una variable (“Pout”) en la que se tiene la suma de las potencias activas demandadas por las cargas (modelo de potencia constante).
- La matriz “lineas”, que también es variable de entrada a la función “radialidad”.
- Una matriz de celda (“Zabc”) en la que se contienen los parámetros de los tramos de línea del sistema, ordenados de acuerdo con la matriz “lineas”.
- Un vector (alimentadores) en el que se tienen los números que identifican los nodos “*slack*” del sistema.
- Un vector (V0) con las tensiones en los nodos “*slack*”. Esta variable al igual que las cinco anteriores es declarada e ingresada a la función como variable global.
- El vector (“conexión”) que arroja como salida2 la función “radialidad”.
- El vector (Ni) que arroja como salida3 la función “radialidad”.
- El vector de celdas (Niout) que arroja como salida4 la función “radialidad”.
- El vector (Nt) que arroja como salida5 la función “radialidad”.
- El vector de celdas (conexion1) que arroja como salida6 la función “radialidad”.

Figura 8: Diagrama de flujo del proceso para evaluar el flujo de carga



Se ha encontrado que cuando la distancia eléctrica entre los nodos “*slack*” y los nodos ubicados en la cola de los tramos radiales del sistema se hace demasiado grande el método aquí implementado pierde convergencia, por lo cual a partir de la segunda iteración cuando se encuentra un error superior al de la iteración anterior, se concluye que no habrá convergencia, y no se continúa la búsqueda de la solución, marcando cero en la sexta posición de la salida uno de la función. Esto se ilustrará con los datos de las configuraciones que no convergen para los sistemas de prueba (en la sección 5).

Después de evaluar el flujo de potencia se obtienen las tensiones en cada barra y las corrientes por cada tramo de líneas, con esta información es posible calcular la eficiencia del sistema, el porcentaje de regulación de tensión en cada barra e incluso el modelo de impedancia de las cargas para las tensiones encontrada. Estos cálculos se realizan dentro de la misma función “flujos” programada en MATLAB.

Para el cálculo de la eficiencia es necesario determinar la potencia que ingresa al sistema de distribución por los nodos “*slack*”, para lo cual se utiliza la información de las corrientes que llegan a estos nodos y sus niveles de tensión, estos cálculos se realizan aplicando el conjunto de ecuaciones (5).

$$P_{in} = \sum_{i=1}^a |V_i| |I_i| \cos(\theta_{v_i} - \theta_{I_i})$$

$$\% \eta = \frac{P_{out}}{P_{in}} 100\%$$
(5)

En donde:

P_{in} : es la potencia de entrada al sistema de distribución.

a : es la cantidad de nodos “*slack*” en el sistema.

V_i e I_i : corresponden respectivamente a la tensión y corriente en el nodo “*slack*” “ i ”.

$\% \eta$: es el porcentaje de eficiencia del sistema de distribución.

P_{out} : es la suma de la potencia consumida por las cargas.

El cálculo del porcentaje de regulación de tensión se realiza aplicando la ecuación (6) para las tensiones en cada una de las barras del sistema, posteriormente se determina el máximo valor obtenido.

$$\%R.V. = \frac{|V_0| - |V_i|}{|V_0|} 100\% \quad (6)$$

En donde:

$\%R.V.$: es el porcentaje de regulación de tensión.

V_0 : es la tensión en los nodos slack del sistema.

V_i : es la tensión encontrada para la barra “i” (para la cual se realiza el cálculo).

Después de ejecutar la función “flujos” en MATLAB se obtienen las siguientes variables:

- Salida1: es un vector de seis posiciones con la siguiente información respectivamente: eficiencia, porcentaje de regulación de tensión máxima, la barra en la que se tiene el porcentaje de regulación máxima, error con el que se terminó el algoritmo del flujo de carga, número de iteraciones realizadas, uno si el algoritmo convergió o cero si no lo hizo.
- Salida2: es una matriz con los niveles de tensión calculados por fase y para cada una de las barras del sistema.
- Salida3: es una matriz con las corrientes obtenidas a la llegada de cada barra y por fase.
- Salida4: es una matriz que contiene el valor de la impedancia que modela las cargas por fase en cada barra, teniendo en cuenta las potencias demandadas y los niveles de tensión calculados.

Una vez obtenida la eficiencia y el porcentaje de regulación de tensión de cada configuración, estos índices son comparados para encontrar la configuración con

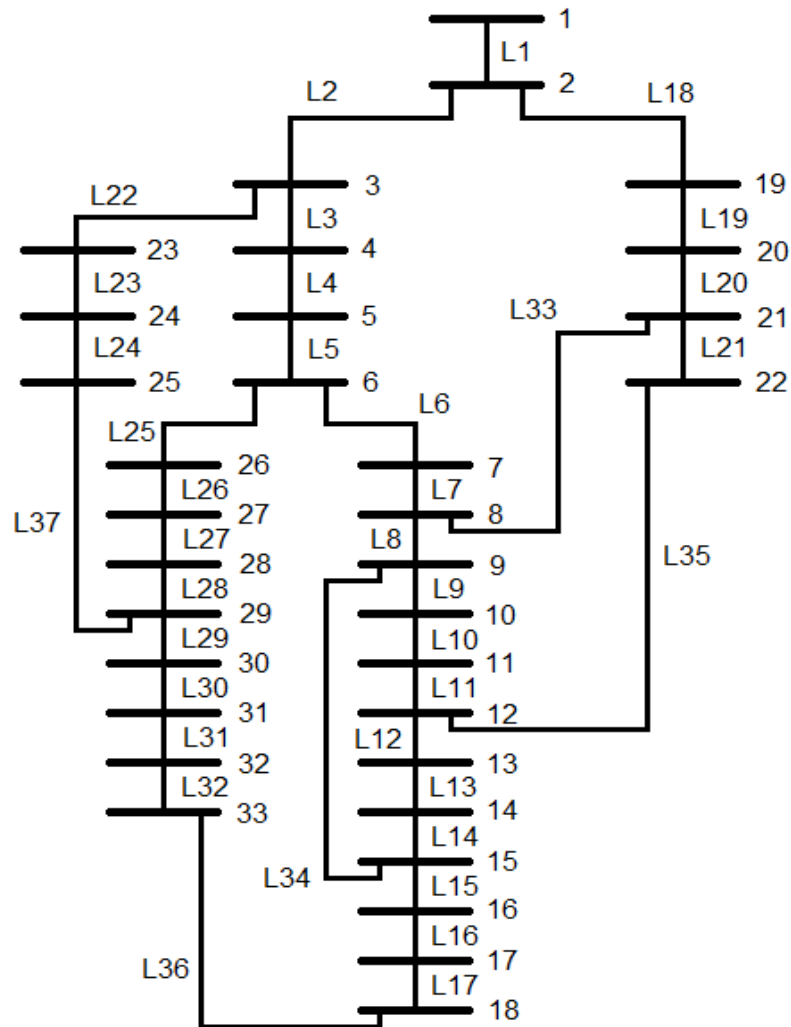
mayor eficiencia y la de menor porcentaje de regulación, estas configuraciones son presentadas en pantalla.

El ingreso de los datos del sistema, la articulación de las funciones y la presentación de resultados son realizados en un archivo llamado “reconfiguracion1” o “reconfiguracion2” que se presentan en el anexo D y E, para el sistema de prueba uno y dos respectivamente.

4 SISTEMAS DE PRUEBA

El método implementado se ha probado en dos sistemas de distribución, uno de ellos representado por un sistema unifilar, y el otro por tres hilos. En la Figura 9 se presenta la representación unifilar de las conexiones de ambos sistemas, esta topología es tomada de [15]. La descripción de los sistemas de prueba se presenta a continuación.

Figura 9: Representación unifilar de los sistemas de prueba, tomado de [15]



4.1 SISTEMA DE PRUEBA 1, REPRESENTADO POR UN SOLO HILO

Este sistema ha sido tomado de [15], está constituido por 33 nodos, de los cuales el nodo 1 es la única conexión al sistema de transmisión (nodo “*slack*”) con una tensión de 12660V; los otros 32 son nodos de carga, la cual se modelada como una demanda de potencia activa y reactiva constante; el sistema cuenta con 37 tramos de línea y en cada una de ellas se considera que existe algún dispositivo que permite conectarlo o desconectarlo. En la Tabla 2 se presentan: los nodos ubicados en los extremos de cada uno de los tramos de línea y las impedancias que caracterizan dichos tramos, y las potencias demandadas en los nodos del sistema.

4.2 SISTEMA DE PRUEBA 2, REPRESENTADO POR TRES HILOS

Este sistema es igual al anterior en cuanto a su topología, sin embargo, las líneas son modeladas de modo trifásico por medio de una matriz de impedancias que por unidad de longitud es igual para todos los tramos, dicha matriz se presenta en la ecuación (7). Los nodos en los extremos de los tramos, sus longitudes y las potencias demandadas (balanceadas) en los nodos se presentan en la Tabla 3.

$$Z_{abc} = \begin{bmatrix} 0,5416 + 0,7129 * j & 0,2512 + 0,3512 * j & 0,2590 + 0,3125 * j \\ 0,2512 + 0,3512 * j & 0,5318 + 0,7365 * j & 0,2200 + 0,3012 * j \\ 0,2590 + 0,3125 * j & 0,2200 + 0,3012 * j & 0,5211 + 0,7218 * j \end{bmatrix} \frac{\Omega}{km} \quad (7)$$

Tabla 2: Características del sistema de prueba 1

Número de línea	Nodo de inicio	Nodo fin	R [Ω]	X [Ω]	P y Q demandada en el nodo fin	
					P	Q
1	1	2	0,0922	0,0477	100	60
2	2	3	0,4930	0,2511	90	40
3	3	4	0,3660	0,1840	120	80
4	4	5	0,3811	0,1941	60	30
5	5	6	0,8190	0,0700	60	20
6	6	7	0,1872	0,6188	200	100
7	7	8	1,7114	1,2351	200	100
8	8	9	1,0300	0,7400	60	20
9	9	10	1,0400	0,7400	60	20
10	10	11	0,1966	0,0650	45	30
11	11	12	0,3744	0,1238	60	35
12	12	13	1,4680	1,1550	60	35
13	13	14	0,5416	0,7129	120	80
14	14	15	0,5910	0,5260	60	10
15	15	16	0,7463	0,5450	60	20
16	16	17	1,2890	1,7210	60	20
17	17	18	0,7320	0,5740	90	40
18	2	19	0,1640	0,1565	90	40
19	19	20	1,5042	1,3554	90	40
20	20	21	0,4095	0,4784	90	40
21	21	22	0,7089	0,9373	90	40
22	3	23	0,4512	0,3083	90	50
23	23	24	0,8980	0,7091	420	200
24	24	25	0,8960	0,7011	420	200
25	6	26	0,2030	0,1038	60	25
26	26	27	0,2842	0,1447	60	25
27	27	28	1,0590	0,9337	60	20
28	28	29	0,8042	0,7006	20	70
29	29	30	0,5075	0,2585	200	600
30	30	31	0,9744	0,9630	150	70
31	31	32	0,3105	0,3619	210	100
32	32	33	0,3410	0,5902	60	40
33	21	8	2,0000	2,0000		
34	9	15	2,0000	2,0000		
35	12	22	2,0000	2,0000		
36	18	33	0,5000	0,5000		
37	25	29	0,5000	0,5000		

Tabla 3: Características del sistema de prueba 2

Número de línea	Nodo de inicio	Nodo fin	Longitud del tramo [km]	Potencia trifásica de la carga en el nodo fin	
				P [kW]	Q [kVAr]
1	1	2	0,1	100	60
2	2	3	0,6	90	40
3	3	4	0,45	120	80
4	4	5	0,47	60	30
5	5	6	1,2	60	20
6	6	7	0,7	200	100
7	7	8	0,83	200	100
8	8	9	1,4	60	20
9	9	10	1,4	60	20
10	10	11	0,2	45	30
11	11	12	0,4	60	35
12	12	13	2	60	35
13	13	14	1	120	80
14	14	15	1	60	10
15	15	16	1	60	20
16	16	17	2,4	60	20
17	17	18	1	90	40
18	2	19	0,25	90	40
19	19	20	2,3	90	40
20	20	21	0,8	90	40
21	21	22	1,3	90	40
22	3	23	0,9	90	50
23	23	24	1,25	420	200
24	24	25	1,2	420	200
25	6	26	0,25	60	25
26	26	27	0,25	60	25
27	27	28	1,5	60	20
28	28	29	1,2	120	70
29	29	30	1	200	600
30	30	31	1,2	150	70
31	31	32	0,5	210	100
32	32	33	0,6	60	40
33	21	8	3		
34	9	15	3		
35	12	22	3		
36	18	33	0,7		
37	25	29	0,7		

5 RESULTADOS

Dado que los datos que caracterizan al sistema bajo prueba se deben introducir en el archivo “reconfiguracion” en Matlab, se ha creado un archivo de estos para cada sistema: “reconfiguracion1” (presentado en el anexo D) para el primer sistema de prueba y “reconfiguracion2” (presentado en el anexo E) para el segundo sistema; esto con el fin de mostrar como introducir datos para sistemas representados por uno y por tres hilos. Todos los demás códigos son genéricos.

Dado que ambos sistemas tienen la misma composición de nodos y posibles conexiones, se ha encontrado para los dos sistemas 435897 combinaciones de líneas activas, es decir, posibles configuraciones; de las cuales solo 50751 satisfacen la condición de operación radial y de alimentación a todos los puntos de carga.

Después de ejecutar los algoritmos planteados para los sistemas de prueba se han obtenido los siguientes resultados.

5.1 RESULTADOS PARA EL SISTEMA DE PRUEBA 1

Los resultados obtenidos para este sistema se muestran en la Tabla 4, en este caso se obtiene la misma configuración para la máxima eficiencia y para el mínimo porcentaje de regulación de tensión. Estos resultados se obtienen después de 26 minutos y 32 segundos. De las 50751 configuraciones a las que se le evaluó el flujo de carga se han encontrado 6438 en las que el algoritmo no converge, estas configuraciones se caracterizan por tener tramos radiales con relación $\frac{Z_{\text{Líneas}}}{Z_{\text{Carga}}}$ superiores a 0,4961; las relaciones entre las impedancias de carga y de las líneas de los tramos radiales son calculadas mediante la ejecución del código de Matlab presentado en el anexo F, el cual debe ser ejecutado después de correr las funciones “reconfiguracion1”. Estas configuraciones no se presentan aquí por ser

muchas, pero se puede tener acceso a ellas en la variable “Diver_inac” después de ejecutar el código presentado en el anexo D.

Tabla 4: Resultados para el sistema de prueba 1

CRITERIO DE OPTIMIZACIÓN	SECCIONES DE LÍNEA INACTIVAS	EFICIENCIA [%]	REGULACIÓN DE TENSIÓN MÁXIMA [%]	NODO DE TENSIONES MÍNIMAS Y VALORES [P.U.]
		POTENCIA DE PÉRDIDAS [kW]		
MÁXIMA EFICIENCIA	7, 9, 14, 32, 37	96,4803	5,4572	32 0,9454
		131,8785		
MÍNIMO %R.V.	7, 9, 14, 32, 37	96,4803	5,4572	32 0,9454
		131,8785		

En la publicación [15] obtienen para este sistema los resultados mostrados en la parte izquierda de la Tabla 5. Con esta configuración y mediante el algoritmo implementado en esta monografía para evaluar el flujo de potencia, se obtienen los resultados presentados en la parte derecha de la Tabla 5, estos datos se obtienen con base en los resultados que se encuentran en la fila 11236 de la matriz de celdas “resultado” después de ejecutar el código “reconfiguracion1” mostrado en el anexo D. En la última fila de la Tabla 5 se presenta la diferencia porcentual entre los resultados en la publicación [15] y los obtenidos para esa misma configuración mediante el algoritmo de flujo de potencia aquí implementado. Se puede apreciar que son resultados similares, cuya diferencia puede deberse al criterio de parada

implementado, que para el algoritmo implementado en este trabajo es obtener un error inferior a 1×10^{-7} , no se tiene conocimiento del criterio de parada para el algoritmo implementado en la publicación [15].

De acuerdo a los resultados presentados en la Tabla 4, mediante la búsqueda exhaustiva aquí desarrollada se ha encontrado una configuración mucho más eficiente que la propuesta en [15], observando una reducción en las pérdidas de más de 7 kW, en tanto que la diferencia debida al método de cálculo del flujo de carga es de 250 W aproximadamente. Sin embargo, el tiempo empleado por el método propuesto en [15] ha sido de tan solo 4,3 segundos, mucho menor al requerido por el método propuesto en este trabajo, que ha sido superior a los 26 minutos.

Tabla 5: Resultados del sistema de prueba 1 obtenidos en [15]

RESULTADOS DE [15]			RESULTADOS PARA LA MISMA CONFIGURACIÓN MEDIANTE EL ALGORITMO PROPUESTO	
SECCIONES DE LÍNEA INACTIVAS	POTENCIA DE PÉRDIDAS [kW]	NODO CON TENSIÓN MÍNIMA [P.U.]	POTENCIA DE PÉRDIDAS [kW]	NODO CON TENSIÓN MÍNIMA
8, 14, 28, 32, 33	139,5	0,9437 (Nodo 33)	139,24	0,9393 (Nodo 33)
DIFERENCIA PORCENTUAL			0,1867%	0,4684%

5.2 RESULTADOS PARA EL SISTEMA DE PRUEBA 2

Los resultados obtenidos para este sistema se muestran en la Tabla 6. Estos resultados se obtienen después de 26 minutos y 38 segundos. En este caso 39 de las configuraciones no presentan convergencia al evaluar el flujo de potencia mediante el método implementado. La Tabla 7 presenta las líneas inactivas de estas configuraciones, esta información es almacenada en la variable “Diver_inac” en el código del anexo E.

Tabla 6: Resultados para el sistema de prueba 2

CRITERIO DE OPTIMIZACIÓN	SECCIONES DE LÍNEA INACTIVAS	EFICIENCIA [%]	REGULACIÓN DE TENSIÓN MÁXIMA [%]	NODO DE TENSIONES MÍNIMAS Y VALORES [P.U.]
		POTENCIA DE PÉRDIDAS [kW]		
MÁXIMA EFICIENCIA	7, 9, 14, 32, 37	98,5806	3,373	0,9663
		53,4909632		32 0,9672 0,9679
MÍNIMO %R.V.	6, 9, 14, 32 Y 37	98,5252	3,1906	0,9681
		55,6096859		32 0,9689 0,9696

Tabla 7: Configuraciones en las que el flujo de potencia no converge para el sistema trifásico

Líneas inactivas	Líneas inactivas	Líneas inactivas
2 11 33 34 37	2 10 15 28 33	2 8 13 27 33
2 11 28 33 34	2 10 15 27 33	2 8 13 24 33
2 11 17 25 33	2 10 15 26 33	2 8 12 33 37
2 11 16 27 33	2 10 15 25 33	2 8 12 28 33
2 11 16 26 33	2 9 33 34 37	2 8 12 27 33
2 11 16 25 33	2 9 16 25 33	2 8 12 26 33
2 11 15 28 33	2 9 15 28 33	2 8 12 24 33
2 11 15 27 33	2 9 15 27 33	2 8 11 33 37
2 11 15 26 33	2 9 15 26 33	2 8 10 33 37
2 11 15 25 33	2 9 15 25 33	2 8 9 33 37
2 10 33 34 37	2 8 14 33 37	2 7 34 35 37
2 10 16 26 33	2 8 13 33 37	2 7 21 34 37
2 10 16 25 33	2 8 13 28 33	2 7 21 24 34

6 CONCLUSIONES

La mayoría de los métodos de reconfiguración requieren iniciar la búsqueda con una configuración del sistema (configuración base), por lo cual la velocidad con la que se encuentre el óptimo, e incluso, las posibilidades de converger al óptimo global quedan supeditadas a la cercanía de la configuración base con la configuración óptima. El método de búsqueda exhaustiva implementado en este trabajo no requiere una configuración base por lo tanto se garantiza el hallazgo del óptimo global.

Se observa que en la evaluación de los flujos de potencia en sistemas radiales mediante el método de barrido – suma de corrientes, la convergencia depende de la razón entre la impedancia que modela las líneas y la que modela las cargas, es decir, entre más comparable sea la impedancia de las líneas con la de las cargas menor será la convergencia del algoritmo. Para los casos analizados en este trabajo la relación $\frac{Z_{Líneas}}{Z_{Carga}}$ más alta que presentó convergencia fue de 0,4961.

El tiempo empleado en la búsqueda exhaustiva para ambos sistemas de prueba ha estado alrededor de 26 minutos y medio, lo que imposibilita la implementación de este método para la operación de sistemas de distribución de energía eléctrica.

Al evaluar todas las posibles configuraciones del sistema este método garantiza que se encuentra la configuración óptima del sistema considerado, por lo cual se constituye en una herramienta para la planeación de sistemas de distribución y para la evaluación de la efectividad en la búsqueda de métodos de reconfiguración más veloces.

BIBLIOGRAFÍA

C. d. R. d. E. y. G. Colombia, *Resolución CREG 070 - Reglamento de Distribución de Energía Eléctrica*, Bogotá, 1998.

CHANG, CHU, S. y WANG, H. «An Improved Backward/Forward Sweep Load Flow Algorithm for Radial Distribution Systems,» *Power Systems, IEEE Transactions on*, vol. XXII, nº 2, pp. 882-884, 2007.

GARCES RUIZ, A. y GUTIERREZ MONSALVE, G. A. Estudio comparativo sobre metodologías de flujo de potencia, Pereira: Universidad Tecnológica de Pereira, 2008.

GONZALEZ QUINTERO, J. A., LISAN MESA, I. y HIFIKEPUNJE KANDJUNGULUME, J. «Algoritmo Heurístico para la Reconfiguración de Sistemas de Distribución Mediante Intercambio de Ramas,» *Ingeniería Energética*, vol. XXXIII, nº 3, pp. 196-204, 2012.

GRANOBLES, R., CAICEDO, G., DIAZ J. F. y GERS, J. M. «Estado del arte en reconfiguración de sistemas de distribución,» *Energía y Computación*, vol. X, nº 1, pp. 50-60, 2001.

M. d. M. y. E. Colombia, *Reglamento Técnico de Instalaciones Eléctricas - RETIE*, Bogotá: Seguridad Eléctrica SAS, 2013.

MONTGOMERY D. y RUNGER, G. Probabilidad y estadística aplicada a la ingeniería, México: MCGRAW-HILL.

MORTON A. y MAREELS. «An Efficient Brute-Force Solution to the Network Reconfiguration Problem,» *Power Delivery, IEEE Transactions on*, vol. XV, nº 3, pp. 996-1000, 2000.

S. R. R, S. Narasimham y R. M, «Optimization of Distribution Network Configuration for Loss Reduction Using Artificial Bee Colony Algorithm,» *World Academy of Science, Engineering and Technology*, nº 45, pp. 708-714, 2008.

WILDI, T. «Sistemas de Potencia Eléctrica, Generación, Transmisión y Distribución,» de *Máquinas Eléctricas y Sistemas de Potencia*, México, Pearson Educación, 2007, pp. 665-905.

REFERENCIAS BIBLIOGRÁFICAS

- [1] J. A. González Quitero, I. Lisan Mesa y J. Hifikepunje Kandjungulume, «Algoritmo Heurístico para la Reconfiguración de Sistemas de Distribución Mediante Intercambio de Ramas,» *Ingeniería Energética*, vol. XXXIII, nº 3, pp. 196-204, 2012.
- [2] T. Wildi, «Sistemas de Potencia Eléctrica, Generación, Transmisión y Distribución,» de *Máquinas Eléctricas y Sistemas de Potencia*, México, Pearson Educación, 2007, pp. 665-905.
- [3] M. d. M. y. E. Colombia, *Reglamento Técnico de Instalaciones Eléctricas - RETIE*, Bogotá: Seguridad Eléctrica SAS, 2013.
- [4] R. Granobles, G. Caicedo, J. F. Díaz y J. M. Gers, «Estado del arte en reconfiguración de sistemas de distribución,» *Energía y Computación*, vol. X, nº 1, pp. 50-60, 2001.
- [5] C. d. R. d. E. y. G. Colombia, *Resolución CREG 070 - Reglamento de Distribución de Energía Eléctrica*, Bogotá, 1998.
- [6] A. Garces Ruiz y G. A. Gutiérrez Monsalve, Estudio comparativo sobre metodologías de flujo de potencia, Pereira: Universidad Tecnológica de Pereira, 2008.
- [7] G. Chang, S. Chu y H. Wang, «An Improved Backward/Forward Sweep Load Flow Algorithm for Radial Distribution Systems,» *Power Systems, IEEE Transactions on*, vol. XXII, nº 2, pp. 882-884, 2007.

- [8] A. Morton y Mareels, «An Efficient Brute-Force Solution to the Network Reconfiguration Problem,» *Power Delivery, IEEE Transactions on*, vol. XV, nº 3, pp. 996-1000, 2000.
- [9] Z. Vale, B. Canizes, J. Soares, P. Oliveira, T. Sousa y T. Pinto, «Logic programming and fuzzy Monte Carlo for distribution network reconfiguration,» de *Intelligent System Application to Power Systems (ISAP), 2011 16th International Conference on*, Hersonissos, 2011.
- [10] R. Syahputra, I. Robandi y M. Ashari, «Reconfiguration of distribution network with DG using fuzzy multi-objective method,» de *Innovation Management and Technology Research (ICIMTR), 2012 International Conference on*, Malacca, 2012.
- [11] P. Ramaswamy y G. Deconinck, «Smart grid reconfiguration using simple genetic algorithm and NSGA-II,» de *Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on*, Berlin, 2012.
- [12] W.-C. Wu y M.-S. Tsai, «Application of Enhanced Integer Coded Particle Swarm Optimization for Distribution System Feeder Reconfiguration,» *Power Systems, IEEE Transactions on*, vol. XXVI, nº 3, pp. 1591-1599, 2011.
- [13] K. Kumar, K. Rajalakshmi y S. Karthikeyan, «A modified artificial neural network based Distribution System reconfiguration for loss minimization,» de *Advances in Electrical Engineering (ICAEE), 2014 International Conference on*, Vellore, 2014.
- [14] D. Montgomery y G. Runger, Probabilidad y estadística aplicada a la ingeniería, México: McGRAW-HILL.

- [15] S. R. R, S. Narasimham y R. M, «Optimization of Distribution Network Configuration for Loss Reduction Using Artificial Bee Colony Algorithm,» *World Academy of Science, Engineering and Technology*, nº 45, pp. 708-714, 2008.

ANEXOS

ANEXO A. Código para la determinación de las posibles configuraciones

Este código corresponde al diagrama de flujo presentado en la Figura 6.

```
%Esta función genera todas las posibles combinaciones de interruptores
%cerrados que se pueden tener, que hagan que la cantidad de tramos de
línea
%activos sea igual a la cantidad de barras de carga o de paso de potencia
%en el sistema. Se recibe:
%En la variable "N": un vector que contiene los números que identifican
las líneas que pueden ser abiertas o cerradas.
%En la variable "r": la cantidad de interruptores que deben ser cerrados.
%En la variable "lin": un vector de tantos elementos como líneas en el
%sistema, se reciben en "1" aquellas líneas que no tienen la posibilidad
de abrir o cerrar, es decir están siempre cerradas; en "0" aquellas que
%pueden activarse o desactivarse mediante algún dispositivo de
%seccionamiento.
function salida=combinatoria(N,r,lin)
n=length(N);
l=round(factorial(n)/(factorial(n-r)*factorial(r)));
salida=zeros(l,length(lin));
q=1;
ref=n-r+1;
cont=ones(1,r);
rn=zeros(r,ref);
for a=1:r%Creando los indicadores sobre los que se moverán los "r"
dígitos seleccionados.
    rn(a,:)=N(a:a-1+ref);
end
while cont(1)<=ref
    salida(q,:)=lin;
    for a=1:r
        salida(q,rn(a,cont(a)))=1;
    end
    cont(r)=cont(r)+1;
    for a=r:-1:2
        if cont(a)>ref
            cont(a-1)=cont(a-1)+1;
        end
    end
    for a=2:r
        if cont(a)>ref
            cont(a)=cont(a-1);
        end
    end
    q=q+1;
end
```

ANEXO B. Código para evaluar la condición de operación radial de las configuraciones

Este código corresponde al diagrama de flujo presentado en la Figura 7.

```
%Este código permite en base a la información de líneas cerradas y
%abiertas (información contenida en el argumento - conexion), determinar
%en
%cada salida:
%salida1: es un vector fila de tres elementos, el primero es dicotómico
%siendo "1" si la configuración es radial o "0" si no lo es; el segundo
%puede ser "0" si no se alimentan todas las barras de carga del sistema,
%"1" si están conectadas todas o "2" si la(s) barra(s) que falta(n) por
%conectarse es(son) alimentadores; el tercer elemento indica cuantas
%veces
%fue necesario chequear conexiones para producir un resultado útil.
%salida2: es un vector fila de tantos elementos como barras haya en el
%sistema, cada posición corresponde a una barra y el número allí
%contenido
%indica de que barra recibe potencia. Si el número es "0" significa que
%esa
%barra es alimentadora (nodo slack).
%salida3: es un vector fila en el que se encuentran las barras
%alimentadoras y posteriormente aquellas en las que se producen
%bifurcaciones.
%salida4: es un vector de celdas (con tantas celdas como elementos tiene
%el
%vector de la "salida3") en el cual se guardan las barras a las cuales
%entrega potencia el correspondiente elemento de "salida3".
%salida5: es un vector fila en el que se tienen las barras finales de las
%trayectorias radiales del sistema.
%salida6: es un vector celdas (de tantas celdas como barras haya en el
%sistema), cada posición corresponde a una barra y el vector allí
%contenido
%indica a que barra(s) le entrega potencia. Si el número es "0" significa
%que esa barra finaliza una trayectoria radial.
function
[salida1,salida2,salida3,salida4,salida5,salida6]=radialidad(conexion)
global barras alimentadores lineas

conexion1=zeros(1,barras);
conexion2{barras}=0;
for a=1:length(conexion2)
    conexion2{a}=0;
end
vercone=sum(conexion);
conectividad{1}=alimentadores;
verificacion=zeros(1,barras); %vector para realizar chequeo de barras
conectadas
radial=1;
```



```

barras1=alimentadores;%para guardar las barras en su orden sucesivo.
barrasno(1)=0;
barrasnoali(1)=0;
conali=0;
unialimen=0;
Ni=alimentadores;
aa(4)=length(Ni)+1;
Niout={};
aa(5)=1;
finales=[];
contfinales=length(alimentadores);%En esta variable se guarda el número
de barras terminales de trayectorias radiales que se deben tener, de
acuerdo a las divisiones que se van presentando.
fin=0;
veces=0;
ram=length(conectividad);
aconec(1)=ram;
bconec=0;
aa(3)=1;
while fin==0
    for a=1:aconec(length(aconec))
        ram=length(conectividad);
        for b=1:length(conectividad{bconec+a})
            aa(1)=1;
            aa(6)=1;
            for c=1:length(conexion)
                for d=1:2
                    if veces==0%Verificar si se han conectado
directamente dos barras alimentadoras
                        for dd=1:length(conectividad{1})
                            for ddd=1:length(conectividad{1})
                                if conexion(c)==1 &&
lineas(c,d)==conectividad{1}(dd) && lineas(c,d+1)==conectividad{1}(ddd)
                                    unialimen=1;
                                end
                            end
                        end
                    end
                end
                if conexion(c)==1 &&
conectividad{bconec+a}(b)==lineas(c,d)
                    conexion(c)=2;%para indicar que esta conexión ya
fue tomada en cuenta
                    verificacion(conectividad{bconec+a}(b))=1;
                    if verificacion(lineas(c,d+1))==0
                        verificacion(lineas(c,d+1))=1;
                    end
                    conexion1(lineas(c,d+1))=conectividad{bconec+a}(b);
                    conexion2{conectividad{bconec+a}(b)}(aa(6))=lineas(c,d+1);
                    aa(6)=aa(6)+1;
                    barras1(length(barras1)+1)=lineas(c,d+1);
                    conectividad{ram+b}(aa(1))=lineas(c,d+1);
                    aa(1)=aa(1)+1;
                else

```

```

        radial=0;
    end
    if unialimen~=0
        radial=0;
    end
    end
    unialimen=0;%Es necesario?
end
end
if aa(1)>1
    contfinales=contfinales+aa(1)-2;%Si aa(1)==2 la barra
analizada actualmente sólo entrega potencia a otra, por tanto no se crea
una nueva trayectoria radial(con su correspondiente final).
end
if veces==0
    if aa(1)==1
        Niout{aa(5)}=0;
        aa(5)=aa(5)+1;
    else
        Niout{aa(5)}=conectividad{ram+b};
        aa(5)=aa(5)+1;
    end
end
if aa(1)>2
    Ni(aa(4))=conectividad{bconec+a}(b);%Guardando los nodos
que inician trayectorias radiales
    aa(4)=aa(4)+1;
    if veces>0
        Niout{aa(5)}=conectividad{ram+b};%Guardando los nodos
a los que entregan potencia los iniciales de trayectorias radiales
        aa(5)=aa(5)+1;
    end
end
if aa(1)==1%La barra analizada actualmente no entrega
potencia a otra (es una barra que termina una trayectoria radial)
    conectividad{ram+b}(aa(1))=0;
    if conectividad{bconec+a}(b)~=0
        finales(aa(3))=conectividad{bconec+a}(b);
        aa(3)=aa(3)+1;
    end
end
if veces==0 && aa(1)==1%Contar las barras alimentadoras que
no se conectaron,pero si fueron chequeada
    conali=conali+1;
end
end
end
ram=length(conectividad);
bconec=sum(aconec);
aconec(length(aconec)+1)=ram-bconec;
if sum(verificacion)+conali==barras && radial==1%Para agregar a
finales las barras del último chequeo
    for a=bconec+1:ram
        for b=1:length(conectividad{a})

```

```

        if conectividad{a}(b)~=0
            finales(aa(3))=conectividad{a}(b);
            aa(3)=aa(3)+1;
        end
    end
end
if radial==0%No es radial
    fin=1;
end
if sum(verificacion)+conali==barras%Todas las barras han sido
conectadas
    fin=1;
    barrastodas=1;
    if vercone*2~=sum(conexion)%Faltan conexiones por tener en
cuenta, las cuales dañaran la radialidad
        radial=0;
    end
else
    barrastodas=0;
end
if vercone*2==sum(conexion)%Todas las conexiones han sido tenidas en
cuenta
    fin=1;
    if sum(verificacion)~=barras%Falta por lo menos una barra de
carga por conectar
        barrastodas=0;
        aa(1)=1;
        for a=1:length(verificacion)%Buscando las barras que no se
conectaron
            if verificacion(a)==0
                barrasno(aa(1))=a;
                aa(1)=aa(1)+1;
            end
        end
        aa(1)=1;
        for a=1:length(barrasno)%Encontrando las barras no conectadas
que también son alimentadoras
            for b=1:length(conectividad{1})
                if barrasno(a)==conectividad{1}(b)
                    barrasnoali(aa(1))=barrasno(a);
                    aa(1)=aa(1)+1;
                end
            end
        end
        if length(barrasnoali)==length(barrasno) && barrasnoali(1)~=0
            barrastodas=2;%Este 2 indica que todas las barras no
conectadas son alimentadoras, por tanto se cumplen restricciones de
suministro
        end
    end
end
if contfinales==aa(3)-1%Se terminaron las conexiones que provienen de
alimentadores

```

```
        fin=1;
    end
    veces=veces+1;
end
salida1=[radial barrastodas veces];
salida2=conexion1;
salida3=Ni;
salida4=Niout;
salida5=finales;
salida6=conexion2;
```

ANEXO C. Código para evaluar el flujo de potencias

Este código corresponde al diagrama de flujo presentado en la Figura 8.

```
%Esta función determina en cada salida:
%Salida1: es un vector de seis posiciones con la siguiente información
%respectivamente: eficiencia, porcentaje de regulación de tensión máxima,
%la barra en la que se tiene el porcentaje de regulación máxima, error
con
%el que se terminó el algoritmo del flujo de carga, número de iteraciones
%realizadas, uno si el algoritmo convergió o cero si no lo hizo.
%Salida2: es una matriz con los niveles de tensión calculados por fase y
%para cada una de las barras del sistema.
%Salida3: es una matriz con las corrientes obtenidas a la llegada de cada
%barra y por fase.
%Salida4: es una matriz que contiene el valor de la impedancia que modela
%las cargas por fase en cada barra, teniendo en cuenta las potencias
%demandadas y los niveles de tensión calculados.
function
[salida1,salida2,salida3,salida4,salida5]=flujos(conexion,Ni,Niout,Nt,con
exion1)
global barras alimentadores V0 V01 SP Pout Zabc lineas sis
V=V01;
I=zeros(sis,barras);
Nti=Ni;
for a=1:length(Nt)
    Nti(length(Nti)+1)=Nt(a);
end
eref=0.0000001;
e=100;
conver=1;
veces=0;
while e>eref
    III=I;%En las variables III y VV1 se guardan los resultados de la
    VV1=V;%iteración anterior para el posterior cálculo del error.
    %Cálculo de las corrientes que solicitan las cargas en cada nodo.
    I=conj(SP./V);
    %Cálculo de las corrientes que entregan los nodos que terminan las
    %trayectorias radiales, y a los que los alimentan sucesivamente hasta
    %llegar a los nodos que inician dichas trayectorias.
    for a=length(Nti):-1:1
        if a<=length(Ni)
            for b=1:length(Niout{a})
                I(:,Nti(a))=I(:,Nti(a))+I(:,Niout{a}(b));
            end
        end
        c(1)=Nti(a);
        c(2)=conexion(c(1));
        fin=0;
        if c(2)==0
            fin=1;
        end
    end
end
```

```

end
for b=1:length(Nti)
    if c(2)==Nti(b)
        fin=1;
    end
end
while fin==0
    I(:,c(2))=I(:,c(2))+I(:,c(1));
    c(1)=c(2);
    c(2)=conexion(c(1));
    if c(2)==0
        fin=0;
    end
    for b=1:length(Nti)
        if c(2)==Nti(b)
            fin=1;
        end
    end
end
end
%Cálculo de las tensiones teniendo en cuenta las corrientes obtenidas
%anteriormente.
for a=1:length(Ni)
    for b=1:length(Niout{a})
        c(1)=Ni(a);
        c(2)=conexion1{c(1)}(b);
        fin=0;
        if c(2)==0
            fin=1;
        end
        while fin==0
            d=0;
            dd=0;
            while dd==0
                d=d+1;
                for k=1:2
                    if lineas(d,k)==c(1) && lineas(d,k+1)==c(2)
                        dd=1;
                    end
                end
            end
            V(:,c(2))=V(:,c(1))-Zabc{d}*I(:,c(2));
            c(1)=c(2);
            for d=1:length(Ni)
                if c(2)==Ni(d)
                    fin=1;
                end
            end
            c(2)=conexion1{c(1)}(1);
            if c(2)==0
                fin=1;
            end
        end
    end
end
end

```

```

end
%cálculo del error
veces=veces+1;
er=e;
e1=abs((V-VV1)*100)./abs(V0(1));
e2=abs((I-III1)*100)./abs(I);
if sis==3
    [me1 pe1]=max(max(e1));
    [me2 pe2]=max(max(e2));
else
    [me1 pe1]=max(e1);
    [me2 pe2]=max(e2);
end
[e ppe]=max([me1,me2]);
if ppe==1
    pe(veces)=pe1;
else
    pe(veces)=pe2;
end
if e>er && veces>1
    conver=0;
    V=VV1;
    I=III1;
    e=-1;
end
end
Zl=(abs(V).^2)./(conj(SP));
%Cálculo de pérdidas y eficiencia.
Pin=0;
for a=1:length(alimentadores)
    for b=1:sis

Pin=Pin+(abs(V(b,alimentadores(a)))*abs(I(b,alimentadores(a)))*cos(angle(
V(b,alimentadores(a))-angle(I(b,alimentadores(a)))));
    end
end
Ploss=Pin-Pout;
eficiencia=(Pout/Pin)*100;
%Cálculos de regulación de tensión en cada nodo y por cada fase.
for a=1:length(conexion)
    for b=1:sis
        Reg(b,a)=((abs(V0(1))-abs(V(b,a)))/(abs(V0(1))))*100;
    end
end
if sis==3
    [Regmax, PosRegmax]=max(max(Reg));
else
    [Regmax, PosRegmax]=max(Reg);
end
salida1=[eficiencia Regmax PosRegmax e veces conver];
salida2=V;
salida3=I;
salida4=Zl;
salida5=pe;

```

ANEXO D. Código para la entrada de datos y articulación de las funciones en el sistema de prueba 1

Este código corresponde al diagrama de flujo presentado en la Figura 5, con los datos de entrada del sistema de prueba 1.

```
%Las barras deben ser numeradas consecutivamente iniciando por el uno.
clc
clear all

tinicio=cputime;
tcombinatoria=tinicio;
global barras alimentadores lineas V0 V01 P Pout Q SP Zabc sis
sis=1;%Sistema monofásico
%Matrices de impedancias de las líneas (longitudes en la matriz S), en
este
%caso todos los tramos de líneas tienen la misma característica por
unidad
%de longitud.
Zabc={0.0922+0.0477*1j 0.4930+0.2511*1j 0.3660+0.1840*1j
0.3811+0.1941*1j...
0.8190+0.0700*1j 0.1872+0.6188*1j 1.7114+1.2351*1j
1.0300+0.7400*1j...
1.0400+0.7400*1j 0.1966+0.0650*1j 0.3744+0.1238*1j
1.4680+1.1550*1j...
0.5416+0.7129*1j 0.5910+0.5260*1j 0.7463+0.5450*1j
1.2890+1.7210*1j...
0.7320+0.5740*1j 0.1640+0.1565*1j 1.5042+1.3554*1j
0.4095+0.4784*1j...
0.7089+0.9373*1j 0.4512+0.3083*1j 0.8980+0.7091*1j
0.8960+0.7011*1j...
0.2030+0.1034*1j 0.2842+0.1447*1j 1.0590+0.9337*1j
0.8042+0.7006*1j...
0.5075+0.2585*1j 0.9744+0.9630*1j 0.3105+0.3619*1j
0.3410+0.5302*1j...
2.0000+2.0000*1j 2.0000+2.0000*1j 2.0000+2.0000*1j
0.5000+0.5000*1j...
0.5000+0.5000*1j};
barras=33;
if sis==3
    a=cosd(120)+sind(120)*1j;
    V0=(12660/(3^0.5))*[1;conj(a);a];
else
    V0=12660;
end
V01=zeros(sis,barras);
for a=1:barras
```



```

V01(:,a)=V0;
end
%Cargas trifásicas balanceadas en los nodos
P=[0 100 90 120 60 60 200 200 60 60 45 60 60 120 60 60 60 90 90 90 90 90
...
90 420 420 60 60 60 20 200 150 210 60]*1000;
Q=[0 60 40 80 30 20 100 100 20 20 30 35 35 80 10 20 20 40 40 40 40 40 50
...
200 200 25 25 20 70 600 70 100 40]*1000;
Pout=sum(P);
for a=1:sis
    SP(a,:)=(P+Q*1j)/sis;
end
alimentadores=[1];%Define las barras fuente (Slack).
% 1 2 3 4 5 6 7 8 9 10
lineas=[1 2 1;2 3 2;3 4 3;4 5 4;5 6 5;6 7 6;7 8 7;8 9 8;9 10 9;10 11
10;...
% 11 12 13 14 15 16 17
11 12 11;12 13 12;13 14 13;14 15 14;15 16 15;16 17 16;17 18
17;...
% 18 19 20 21 22 23 24
2 19 2;19 20 19;20 21 20;21 22 21;23 23 3;23 24 23;24 25 24;...
% 25 26 27 28 29 30 31
6 26 6;26 27 26;27 28 27;28 29 28;29 30 29;30 31 30;31 32 31;...
% 32 33 34 35 36 37
32 33 32;8 21 8;9 15 9;12 22 12;18 33 18;25 29 25];
% 1 2 3 4 5 6 7 8 9 13 17 20 24 28 30
%Aquí se escriben las líneas que cuentan con interruptores.
inter=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
...
26 27 28 29 30 31 32 33 34 35 36 37];
%inter=[34 35 36 37 20 15 27 5 24];
[flin clin]=size(lineas);
lin=ones(sis,flin);
%Dejando desactivandas solo las líneas que tienen interruptor.
for c=1:length(inter)
    lin(inter(c))=0;%Configuración inicial - en uno las líneas sin
posibilidad de activar o desactivar
end
line=combinatoria(inter,barras-length(alimentadores)-(flin-
length(inter)),lin);
tcombinatoria=cputime-tcombinatoria;
disp('El tiempo (en segundos) empleado para encontrar todas las posibles
configuraciones del sistema ha sido: ')
disp(tcombinatoria)
[k kn]=size(line);
disp(['Se han encontrado ',num2str(k),' configuraciones posibles.'])
tradicional=cputime;
a1=zeros(k,3);
b1=zeros(k,barras);
c1=cell(k,1);
d1=cell(k,1);
e1=cell(k,1);
f1=cell(k,1);

```

```

radiales=zeros(k,1);
n=0;
contt=1;
divergentes=[];
contdiver=0;
varioseficientes=0;
contvarefi=0;
variosRegmin=0;
contvarReg=0;
%Evaluando la radialidad de las posibles configuraciones.
for b=1:length(line)
    if cputime-tinicio>180*contt
        clc
        disp('El programa se está ejecutando.')
        disp('Han transcurrido: ')
        disp(fix((cputime-tinicio)/60))
        disp('minutos, y se ha realizado el: ')
        disp(b*85/k)
        disp('Por ciento de los cálculos.')
        contt=contt+1;
    end
    [a1(b,:),b1(b,:),c1{b},d1{b},e1{b},f1{b}]=radialidad(line(b,:));
    if a1(b,1)==1 && a1(b,2)~=0
        n=n+1;
        radiales(n)=b;
    end
end
tradialidad=cputime-tradialidad;
tflujos=cputime;
resultado=cell(n,5);
eficienciamax=0;
optimoe=0;
Regmin=100;
optimor=0;
%Realizando los flujos de carga de las configuraciones radiales.
for b=1:n
    if cputime-tinicio>180*contt
        clc
        disp('El programa se está ejecutando.')
        disp('Han transcurrido: ')
        disp(fix((cputime-tinicio)/60))
        disp('minutos, y se ha realizado el: ')
        disp(85+(15*b/n))
        disp('Por ciento de los cálculos.')
        contt=contt+1;
    end

    [resultado{b,1},resultado{b,2},resultado{b,3},resultado{b,4},resultado{b,5}]=flujos(b1(radiales(b,:),:),c1{radiales(b)},d1{radiales(b)},e1{radiales(b)},f1{radiales(b)});
    %Buscando la configuración más eficiente
    if resultado{b,1}(1)>eficienciamax
        eficienciamax=resultado{b,1}(1);
        optimo(1)=b;
    end
end

```

```

        varioseficientes=0;
        contvarefi=1;
elseif resultado{b,1}(1)==eficienciamax && b~=1
    contvarefi=contvarefi+1;
    varioseficientes(contvarefi)=b;
end
inde=0;
%Buscando la configuración con menor %R.V.
if resultado{b,1}(2)<Regmin
    Regmin=resultado{b,1}(2);
    PRegmin=resultado{b,1}(3);
    optimo(2)=b;
    variosRegmin=0;
    contvarReg=1;
elseif resultado{b,1}(2)==Regmin && b~=1
    contvarReg=contvarReg+1;
    variosRegmin(contvarReg)=b;
end
%Anotando las configuraciones que no presentan convergencia.
if resultado{b,1}(6)==0
    contdiver=contdiver+1;
    divergentes(contdiver)=b;
end
end
tflujos=cputime-tflujos;
tttotal=cputime-tinicio;
%Presentación de resultados.
for a=1:2
    c=0;
    for b=1:flin
        if line(radiales(optimo(a)),b)==0
            c=c+1;
            R_inac(a,c)=b;
        end
    end
end
for a=1:contdiver
    c=0;
    for b=1:flin
        if line(radiales(divergentes(a)),b)==0
            c=c+1;
            Diver_inac(a,c)=b;
        end
    end
end
horas=fix(tttotal/3600);
minutos=fix((tttotal-horas*3600)/60);
segundos=fix(tttotal-horas*3600-minutos*60);
clc
disp(['Los cálculos han sido finalizados exitosamente, en un tiempo
(h:m:s) de: ',num2str(horas),':',num2str(minutos),':',num2str(segundos)])
disp(['Se han encontrado ',num2str(k),' posibles configuraciones, de las
cuales ',num2str(n),' son radiales y alimentan todas las cargas.'])

```

```

disp(['La configuración de líneas abiertas y cerradas con mayor
eficiencia (' ,num2str(eficienciamax), '%, con unas pérdidas de
' ,num2str((Pout*100/eficienciamax)-Pout), 'W) es:'])
disp(line(radiales(optimo(1)),:))
disp(['La cual tiene los siguientes tamos de línea inactivos:
' ,num2str(R_inac(1,:)), ', y presenta un %R.V. máximo de:
' ,num2str(resultado{optimo(1),1}(2)), '%'])
disp(['El nodo con la menor tensión es el
' ,num2str(resultado{optimo(1),1}(3)), ', con una tensión de:'])
disp(num2str(abs(resultado{optimo(1),2}(:,resultado{optimo(1),1}(3)))))
disp(['La configuración de líneas abiertas y cerradas con menor %R.V.
(' ,num2str(Regmin), '%) es:'])
disp(line(radiales(optimo(2)),:))
disp(['La cual tiene los siguientes tamos de línea inactivos:
' ,num2str(R_inac(2,:)), ', y presenta una eficiencia de:
' ,num2str(resultado{optimo(2),1}(1)), '%, que corresponde a unas pérdidas
de ' ,num2str((Pout*100/resultado{optimo(2),1}(1))-Pout), 'W'])
disp(['El nodo con la menor tensión es el
' ,num2str(resultado{optimo(2),1}(3)), ', con una tensión de:'])
disp(num2str(abs(resultado{optimo(2),2}(:,PRegmin))))

```

ANEXO E. Código para la entrada de datos y articulación de las funciones en el sistema de prueba 2

Este código corresponde al diagrama de flujo presentado en la Figura 5, con los datos de entrada del sistema de prueba 2.

```
%Las barras deben ser numeradas consecutivamente iniciando por el uno.
clc
clear all

tinicio=cputime;
tcombinatoria=tinicio;
global barras alimentadores lineas V0 V01 P Pout Q SP Zabc sis
sis=3;%sistema trifásico
%Matrices de impedancias de las líneas (longitudes en la matriz S), en
este
%caso todos los tramos de líneas tienen la misma característica por
unidad
%de longitud.
Zabc1=[0.5416+0.7129*1j 0.2512+0.3512*1j 0.259+0.3125*1j;...
        0.2512+0.3512*1j 0.5318+0.7365*1j 0.22+0.3012*1j;...
        0.259+0.3125*1j 0.22+0.3012*1j 0.5211+0.7218*1j];

S=[0.1;0.6;0.45;0.47;1.2;0.7;0.83;1.4;1.4;0.2;0.4;2;1;1;1;2.4;1;0.25;...
2.3;0.8;1.3;0.9;1.25;1.2;0.25;0.25;1.5;1.2;1;1.2;0.5;0.6;3;3;3;0.7;0.7];
for a=1:length(S)
    Zabc{a}=S(a)*Zabc1;
end
barras=33;
if sis==3
    a=cosd(120)+sind(120)*1j;
    V0=(12660/(3^0.5))*[1;conj(a);a];
else
    V0=12660;
end
V01=zeros(sis,barras);
for a=1:barras
    V01(:,a)=V0;
end
%Cargas trifásicas balanceadas en los nodos
P=[0 100 90 120 60 60 200 200 60 60 45 60 60 120 60 60 60 90 90 90 90
...
    90 420 420 60 60 60 120 200 150 210 60]*1000;
Q=[0 60 40 80 30 20 100 100 20 20 30 35 35 80 10 20 20 40 40 40 40 50
...
    200 200 25 25 20 70 600 70 100 40]*1000;
```

```

Pout=sum(P);
for a=1:sis
    SP(a,:)=(P+Q*1j)/sis;
end
alimentadores=[1];%Define las barras fuente (Slack).
% 1 2 3 4 5 6 7 8 9 10
lineas=[1 2 1;2 3 2;3 4 3;4 5 4;5 6 5;6 7 6;7 8 7;8 9 8;9 10 9;10 11
10;...
% 11 12 13 14 15 16 17
11 12 11;12 13 12;13 14 13;14 15 14;15 16 15;16 17 16;17 18
17;...
% 18 19 20 21 22 23 24
2 19 2;19 20 19;20 21 20;21 22 21;22 23 23;23 24 23;24 25 24;...
% 25 26 27 28 29 30 31
6 26 6;26 27 26;27 28 27;28 29 28;29 30 29;30 31 30;31 32 31;...
% 32 33 34 35 36 37
32 33 32;8 21 8;9 15 9;12 22 12;18 33 18;25 29 25];
% 1 2 3 4 5 6 7 8 9 13 17 20 24 28 30
%Aquí se escriben las líneas que cuentan con interruptores.
inter=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
...
26 27 28 29 30 31 32 33 34 35 36 37];
%inter=[34 35 36 37 20 15 27 5 24];
[flin clin]=size(lineas);
lin=ones(1,flin);
%Dejando desactivandas solo las líneas que tienen interruptor.
for c=1:length(inter)
    lin(inter(c))=0;%Configuración inicial - en uno las líneas sin
posibilidad de activar o desactivar
end
line=combinatoria(inter,barras-length(alimentadores)-(flin-
length(inter)),lin);
tcombinatoria=cputime-tcombinatoria;
disp('El tiempo (en segundos) empleado para encontrar todas las posibles
configuraciones del sistema ha sido: ')
disp(tcombinatoria)
[k kn]=size(line);
disp(['Se han encontrado ',num2str(k),' configuraciones posibles.'])
tradicidad=cputime;
a1=zeros(k,3);
b1=zeros(k,barras);
c1=cell(k,1);
d1=cell(k,1);
e1=cell(k,1);
f1=cell(k,1);
radiales=zeros(k,1);
n=0;
contt=1;
divergentes=[];
contdiver=0;
varioseficientes=0;
contvarefi=0;
variosRegmin=0;
contvarReg=0;

```

```

%Evaluando la radialidad de las posibles configuraciones.
for b=1:length(line)
    if cputime-tinicio>180*contt
        clc
        disp('El programa se está ejecutando.')
        disp('Han transcurrido: ')
        disp(fix((cputime-tinicio)/60))
        disp('minutos, y se ha realizado el: ')
        disp(b*85/k)
        disp('Por ciento de los cálculos.')
        contt=contt+1;
    end
    [a1(b,:),b1(b,:),c1{b},d1{b},e1{b},f1{b}]=radialidad(line(b,:));
    if a1(b,1)==1 && a1(b,2)~=0
        n=n+1;
        radiales(n)=b;
    end
end
tradialidad=cputime-tradialidad;
tflujos=cputime;
resultado=cell(n,5);
eficienciamax=0;
optimoe=0;
Regmin=100;
optimor=0;
%Realizando los flujos de carga de las configuraciones radiales.
for b=1:n
    if cputime-tinicio>180*contt
        clc
        disp('El programa se está ejecutando.')
        disp('Han transcurrido: ')
        disp(fix((cputime-tinicio)/60))
        disp('minutos, y se ha realizado el: ')
        disp(85+(15*b/n))
        disp('Por ciento de los cálculos.')
        contt=contt+1;
    end

    [resultado{b,1},resultado{b,2},resultado{b,3},resultado{b,4},resultado{b,5}]=flujos(b1(radiales(b,:),:),c1{radiales(b)},d1{radiales(b)},e1{radiales(b)},f1{radiales(b)});
    %Buscando la configuración más eficiente
    if resultado{b,1}(1)>eficienciamax
        eficienciamax=resultado{b,1}(1);
        optimo(1)=b;
        varioseficientes=0;
        contvarefi=1;
    elseif resultado{b,1}(1)==eficienciamax && b~=1
        contvarefi=contvarefi+1;
        varioseficientes(contvarefi)=b;
    end
    inde=0;
    %Buscando la configuración con menor %R.V.
    if resultado{b,1}(2)<Regmin

```

```

        Regmin=resultado{b,1}(2);
        PRegmin=resultado{b,1}(3);
        optimo(2)=b;
        variosRegmin=0;
        contvarReg=1;
elseif resultado{b,1}(2)==Regmin && b~=1
    contvarReg=contvarReg+1;
    variosRegmin(contvarReg)=b;
end
%Anotando las configuraciones que no presentan convergencia.
if resultado{b,1}(6)==0
    contdiver=contdiver+1;
    divergentes(contdiver)=b;
end
end
tflujos=cputime-tflujos;
tttotal=cputime-tinicio;
%Presentación de resultados.
for a=1:2
    c=0;
    for b=1:flin
        if line(radiales(optimo(a)),b)==0
            c=c+1;
            R_inac(a,c)=b;
        end
    end
end
for a=1:contdiver
    c=0;
    for b=1:flin
        if line(radiales(divergentes(a)),b)==0
            c=c+1;
            Diver_inac(a,c)=b;
        end
    end
end
horas=fix(tttotal/3600);
minutos=fix((tttotal-horas*3600)/60);
segundos=fix(tttotal-horas*3600-minutos*60);
clc
disp(['Los cálculos han sido finalizados exitosamente, en un tiempo
(h:m:s) de: ',num2str(horas),':',num2str(minutos),':',num2str(segundos)])
disp(['Se han encontrado ',num2str(k),' posibles configuraciones, de las
cuales ',num2str(n),' son radiales y alimentan todas las cargas.'])
disp(['La configuración de líneas abiertas y cerradas con mayor
eficiencia (',num2str(eficienciamax),'%, con unas pérdidas de
',num2str((Pout*100/eficienciamax)-Pout),'W) es:'])
disp(line(radiales(optimo(1)),:))
disp(['La cual tiene los siguientes tamos de línea inactivos:
',num2str(R_inac(1,:)),' y presenta un %R.V. máximo de:
',num2str(resultado{optimo(1),1}(2)),'%'])
disp(['El nodo con la menor tensión es el
',num2str(resultado{optimo(1),1}(3)),' con una tensión de:'])
disp(num2str(abs(resultado{optimo(1),2}(:,resultado{optimo(1),1}(3)))))

```



```

disp(['La configuración de líneas abiertas y cerradas con menor %R.V.
(', num2str(Regmin), '%) es:'])
disp(line(radiales(optimo(2)), :))
disp(['La cual tiene los siguientes tramos de línea inactivos:
', num2str(R_inac(2, :)), ', y presenta una eficiencia de:
', num2str(resultado{optimo(2), 1}(1)), '%, que corresponde a unas pérdidas
de ', num2str((Pout*100/resultado{optimo(2), 1}(1))-Pout), 'W'])
disp(['El nodo con la menor tensión es el
', num2str(resultado{optimo(2), 1}(3)), ', con una tensión de:'])
disp(num2str(abs(resultado{optimo(2), 2}(:, PRegmin))))

```

ANEXO F. Código para el cálculo de la relación entre las impedancias de línea y de carga.

Este código debe ser ejecutado después de correr la función “reconfiguracion1”.

```
Z1=cell(n,25);
SP2=cell(n,25);
nradial=zeros(n,1);
v=1;
for z=1:n
    conexion=[];
    Ni=[];
    Niout=[];
    conexion1=[];
    conexion=b1(radiales(z),:);
    Ni=c1(radiales(z));
    Niout=d1(radiales(z));
    conexion1=f1(radiales(z));
    contZ1=0;
    for a=1:25
        Z1{z,a}=0;
        SP2{z,a}=0;
    end
    for a=1:length(Ni)
        for b=1:length(Niout{a})
            contZ1=contZ1+1;
            c(1)=Ni(a);
            c(2)=conexion1{c(1)}(b);
            fin=0;
            if c(2)==0
                fin=1;
            end
            while fin==0
                d=0;
                dd=0;
                while dd==0
                    d=d+1;
                    for k=1:2
                        if lineas(d,k)==c(1) && lineas(d,k+1)==c(2)
                            dd=1;
                        end
                    end
                end
                Z1{z,contZ1}=Z1{z,contZ1}+Zabc{d};
                if resultado{z,5}(length(resultado{z,5}))==c(1) ||
resultado{z,5}(length(resultado{z,5}))==c(2)
                    nradial(z,1)=contZ1;
                end
            end
        end
    end
end
```

```

        SP2{z,contZ1}=SP2{z,contZ1}+P(c(2))+Q(c(2))*1j;
        c(1)=c(2);
        for d=1:length(Ni)
            if c(2)==Ni(d)
                fin=1;
            end
        end
        c(2)=conexion1{c(1)}(1);
        if c(2)==0
            fin=1;
        end
    end
end
end
if z==10000*v
    z
    v=v+1;
end
end
Zmax=zeros(n,1);
nradialZ=zeros(n,1);
raro=[];
contraro=0;
Z1d=cell(contdiver,25);
Z1c=cell(n-contdiver,25);
SP1d=cell(contdiver,25);
SP1c=cell(n-contdiver,25);
SP2d=cell(contdiver,25);
SP2c=cell(n-contdiver,25);
Reld=cell(contdiver,25);
Relc=cell(n-contdiver,25);
cd=1;
cc=1;
for z=1:n
    vec=[];
    a=1;
    while Z1{z,a}~=0
        vec(a)=Z1{z,a};
        a=a+1;
    end
    [Zmax(z) nradialZ(z)]=max(vec);
    if nradialZ(z)~=nradial(z)
        contraro=contraro+1;
        raro(contraro)=z;
    end
    if z==divergentes(cd)
        for a=1:25
            Z1d{cd,a}=Z1{z,a};
            SP1d{cd,a}=conj((V0^2)/Z1d{cd,a});
            SP2d{cd,a}=SP2{z,a};
            Reld{cd,a}=abs(SP2d{cd,a}/SP1d{cd,a});
        end
        cd=cd+1;
    else

```

```

        for a=1:25
            Z1c{cc,a}=Z1{z,a};
            SP1c{cc,a}=conj((V0^2)/Z1c{cc,a});
            SP2c{cc,a}=SP2{z,a};
            Relc{cc,a}=abs(SP2c{cc,a}/SP1c{cc,a});
        end
        cc=cc+1;
    end
end
MaxReld=zeros(1,contdiver);
MinReld=zeros(1,contdiver);
Pomaxd=zeros(1,contdiver);
Pomind=zeros(1,contdiver);
for a=1:contdiver
    vec=[];
    b=1;
    while Reld{a,b}~=0
        vec(b)=Reld{a,b};
        b=b+1;
    end
    [MaxReld(a) Pomaxd(a)]=max(vec);
    [MinReld(a) Pomind(a)]=min(vec);
end
MaxRelc=zeros(1,n-contdiver);
MinRelc=zeros(1,n-contdiver);
Pomaxc=zeros(1,n-contdiver);
Pominc=zeros(1,n-contdiver);
for a=1:n-contdiver
    vec=[];
    b=1;
    while Relc{a,b}~=0
        vec(b)=Relc{a,b};
        b=b+1;
    end
    [MaxRelc(a) Pomaxc(a)]=max(vec);
    [MinRelc(a) Pominc(a)]=min(vec);
end
end

```