	<b>GESTIÓN DE SERVICIOS ACADÉMICOS Y BIBLIOTECARIOS</b>	<b>CÓDIGO</b>	FO-GS-15
		<b>VERSIÓN</b>	02
<b>ESQUEMA HOJA DE RESUMEN</b>		<b>FECHA</b>	03/04/2017
		<b>PÁGINA</b>	1 de 1
<b>ELABORÓ</b>	<b>REVISÓ</b>	<b>APROBÓ</b>	
Jefe División de Biblioteca	Equipo Operativo de Calidad	Líder de Calidad	

## RESUMEN TRABAJO DE GRADO

AUTOR(ES): NOMBRES Y APELLIDOS COMPLETOS

NOMBRE(S): JUAN FELIPE APELLIDOS: PEÑA BECERRA

NOMBRE(S): ANDRÉS FELIPE APELLIDOS: ESTEBAN LÓPEZ

FACULTAD: INGENIERIA

PLAN DE ESTUDIOS: INGENIERIA ELECTROMECAÁNICA

DIRECTOR:

NOMBRE(S): GLORIA ESMERALDA APELLIDOS: SANDOVAL MARTÍNEZ

NOMBRE(S): LUIS FERNANDO APELLIDOS: GRISALES NORIEGA

TÍTULO DEL TRABAJO (TESIS): Proponer y evaluar de una estrategia de flujo óptimo de potencia en redes de corriente alterna para reducción de pérdidas de energía considerando la operación de paneles fotovoltaicos.

Este trabajo de investigación, presenta la reducción de pérdidas de energía en un sistema radial de distribución de prueba, mediante la inclusión de la generación distribuida con energía solar, aplicando las técnicas de optimización PSO y SCA, el cual es simulado en MATLAB. Para la realización de este trabajo se parte conociendo la ubicación óptima de los generadores de distribución. La implementación de este trabajo es presentada con un sistema de prueba de 33 nodos, el cual se consideran tres escenarios para la simulación. El primero consiste en la simulación de flujos de potencia 24 horas donde se incluye los datos de la demanda en Colombia; en el segundo escenario se incluye la ubicación de los generadores de distribución junto con la curva de radiación de la ciudad de Medellín; en el tercer escenario se aplican las técnicas de optimización PSO y SCA para hallar la configuración de potencia adecuada de los generadores, para reducir las pérdidas de energía del sistema y se realiza una comparación para analizar cuál de ellos se adapta mejor a esta investigación. Los resultados obtenidos muestran que cuando se integra la generación distribuida en el sistema de distribución existe una reducción de pérdidas de potencia.

PALABRAS CLAVES: Pérdidas de energía, generación distribuida, optimización, energía solar, generadores de distribución.

CARACTERÍSTICAS:

PÁGINAS: 149 PLANOS: \_\_ ILUSTRACIONES: \_\_ CD ROOM: \_\_

PROPONER Y EVALUAR DE UNA ESTRATEGIA DE FLUJO ÓPTIMO DE POTENCIA EN  
REDES DE CORRIENTE ALTERNA PARA REDUCCIÓN DE PÉRDIDAS DE ENERGÍA  
CONSIDERANDO LA OPERACIÓN DE PANELES FOTOVOLTAICOS

ESTEBAN LÓPEZ ANDRÉS FELIPE

PEÑA BECERRA JUAN FELIPE

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER  
FACULTAD DE INGENIERIA  
PROGRAMA ACADÉMICO INGENIERÍA ELECTROMECHANICA  
SAN JOSÉ DE CÚCUTA

2022

PROPONER Y EVALUAR UNA ESTRATEGIA DE FLUJO ÓPTIMO DE POTENCIA EN  
REDES CORRIENTE ALTERNA PARA REDUCCIÓN DE PÉRDIDAS DE ENERGÍA  
CONSIDERANDO LA OPERACIÓN DE PANELES FOTOVOLTAICOS

ESTEBAN LÓPEZ ANDRÉS FELIPE

PEÑA BECERRA JUAN FELIPE

Proyecto de grado

Requisito para Optar al Título de Ingeniero Electromecánico

Director

Mg. GLORIA ESMERALDA SANDOVAL MARTINEZ

Codirector

Dr. LUIS FERNANDO GRISALES NOREÑA

UNIVERSIDAD FRANCISCO DE PAULA SANTANDER

FACULTAD DE INGENIERIA

PROGRAMA ACADÉMICO INGENIERÍA ELECTROMECHANICA

SAN JOSÉ DE CÚCUTA

2022

**ACTA DE SUSTENTACIÓN PROYECTO DE GRADO  
MODALIDAD TRABAJO INVESTIGATIVO**

**FECHA:** 22 de marzo de 2022

**HORA:** 06:00 p.m.

**LUGAR:** SC 301

**PLAN DE ESTUDIOS:** INGENIERÍA ELECTROMECAÁNICA

**TÍTULO DEL TRABAJO DE GRADO:** "PROPONER Y EVALUAR DE UNA ESTRATEGIA DE FLUJO ÓPTIMO DE POTENCIA EN REDES DE CORRIENTE ALTERNA PARA REDUCCIÓN DE PÉRDIDAS DE ENERGÍA CONSIDERANDO LA OPERACIÓN DE PANELES FOTOVOLTAICOS".

**JURADOS:** Mg: GERMAN ENRIQUE GALLEGO RODRÍGUEZ  
Mg: CRISTIAN LEONARDO TARAZONA CELIS

**DIRECTOR:** Mg: GLORIA ESMERALDA SANDOVAL MARTÍNEZ  
**CODIRECTOR:** Dr: LUIS FERNANDO GRISALES NOREÑA

**MERITORIA**

<b>NOMBRE DEL ESTUDIANTE:</b>	<b>CÓDIGO</b>	<b>CALIFICACION</b>
JUAN FELIPE PEÑA BECERRA	1091210	4.5
ANDRÉS FELIPE ESTEBAN LÓPEZ	1091176	4.5

**FIRMA DE LOS JURADOS:**



**VOBO. COORDINADOR COMITÉ CURRICULAR**

*Margarita Ch.*



## **Dedicatoria**

Primeramente, gracias al creador de la vida por permitirnos vivir esta experiencia próxima a culminar, al apoyo de nuestros padres por ser ese motor y apoyo durante todo el aprendizaje, a nuestros familiares y seres queridos, así también gracias a todos nuestros amigos que la vida nos ha puesto durante la etapa. Queremos hacer un agradecimiento a la ingeniera GLORIA ESMERALDA SANDOVAL, a los ingenieros LUIS FERNANDO GRISALES NOREÑA y JHON JAIRO ROJAS MONTAÑO y a todo el grupo de investigación del ITM de la ciudad de Medellín por el apoyo fundamental para la elaboración de este trabajo de grado. “Aunque nada cambie si yo cambio, todo cambia” Marcel Proust

**Andrés Felipe Esteban López**

**Juan Felipe Peña Becerra**

## Contenido

Introducción	14
1. Objetivos	16
1.1 Objetivo General	16
1.2 Objetivos Específicos	16
2. Flujos de Carga en Sistemas de Distribución y Métodos de Optimización	17
2.1 Conceptos Relevantes de un Sistema de Distribución y Flujo de Carga	17
2.1.1 Matriz de Admitancia	20
2.1.2 Ecuaciones para el Flujo de Potencia de una Red Eléctrica	21
2.2 Métodos de Flujo de Carga	23
2.2.1 Aproximaciones Sucesivas	23
2.2.2 Barrido Iterativo	27
2.3 Optimización	32
2.3.1 Variables de decisión	33
2.3.2 Restricciones	34
2.3.3 Función objetivo	35
2.4 Flujos Óptimos de Potencia	35
2.4.1 Modelo matemático	36
2.4.2 Restricciones	37
2.5 Optimización por Enjambre de Partículas (PSO)	38

2.6 Algoritmo Seno Coseno (SCA)	41
3. Programación de los Flujos de Carga de un Sistema de Distribución en Matlab	44
3.1 Elección del Sistema de Prueba con sus Variables y Parámetros	44
3.2 Representación en Programas Especializados	46
3.2.1 Ejecución del flujo de carga en un sistema radial sin GD en Digsilent	46
3.2.2 Desarrollo de la Programación del Flujo de Carga en Matlab sin GD	58
3.2.2.1 Programación del Método de Barrido Iterativo	58
3.2.2.2 Programación del Método de Aproximaciones Sucesivas	66
3.2.3 Programación de los Flujos de Carga en Matlab 24 Horas con y sin GD	70
3.2.3.1 Programación del Método Aproximaciones Sucesivas 24 Horas sin GD	70
3.2.3.2 Programación del Método Aproximaciones Sucesivas 24 Horas con GD	84
4 Flujos Óptimos de Potencia	92
4.1 Algoritmo PSO (Optimización por Enjambre de Partículas)	92
4.1.1 Método de Avance de la Técnica de Optimización	
100	
4.2 Algoritmo SCA (Senos y Cosenos)	103
4.2.1 Método de Avance de la Técnica de Optimización SCA	105
5. Análisis e Interpretación de los Resultados	113
5.1 Resultados de las Corrientes con GD Optimizada y sin GD.	115
5.2 Resultados de Voltajes con GD Optimizado y sin GD	117
5.3 Resultados de Pérdidas de Energía hora a hora con GD optimizado y sin GD	119

5.4 Resultados de las pérdidas totales con GD Optimizado y sin GD	121
6. Conclusiones	123
7. Recomendaciones	126
8. Referencias	127
9. Anexos	130



## Lista de Tablas

Tabla 1. Datos de las características de líneas y nodos	45
Tabla 2. Datos de radiación y demanda	70
Tabla 3. Datos de potencia inyectada por los generadores	72
Tabla 4. Datos de tensión sin GD	77
Tabla 5. Datos de corriente sin GD	80
Tabla 6. Datos de pérdidas de energía sin GD	81
Tabla 7. Tabla de voltajes hora a hora con generación manual	86
Tabla 8. Tabla de corrientes hora a hora con generación manual	87
Tabla 9. Tabla de pérdidas de energía hora a hora con generación manual	89
Tabla 10. Opciones para minimizar pérdidas de energía	92
Tabla 11. Tiempo de respuesta del PSO	108
Tabla 12. Configuración de potencia optima	108
Tabla 13. Datos de tensión con GD optimizada	108
Tabla 14. Datos de corriente con GD optimizada	110
Tabla 15. Datos de Energía de Energía con GD Optimizada	111
Tabla 16. Promedio del tiempo de respuesta y pérdidas de energía	113
Tabla 17. Comparación de los métodos PSO y SCA	114
Tabla 18. Comparación de la corriente sin GD y aplicando flujos óptimos	116
Tabla 19. Comparación de los voltajes sin GD y aplicando flujos óptimos	118
Tabla 20. Comparación de las pérdidas de enetgía sin GD y aplicando flujos óptimos	120
Tabla 21. Pérdidas totales con GD optimizada y sin GD	121

## Índice de figuras

Figura 1. Sistema de distribución radial de 4 nodos	18
Figura 2. Distancia entre conductores	19
Figura 3. Impedancias de un sistema de distribución	20
Figura 4. Generación y demanda en un sistema de distribución	23
Figura 5. Componentes de la matriz Ybus	24
Figura 6. Asignación de corrientes	28
Figura 7. Movimiento de la partícula	43
Figura 8. Funciones seno y coseno	43
Figura 9. Diagrama de prueba IEEE 33 nodos	44
Figura 10. Selección del nodo Slack	46
Figura 11. Interfaz del Slack	47
Figura 12. Selección de nodos	48
Figura 13. Interfaz de los nodos	49
Figura 14. Selección de líneas	49
Figura 15. Interfaz de líneas	50
Figura 16. Parámetros dentro de la interfaz de líneas	51
Figura 17. Selección de cargas	52
Figura 18. Interfaz de cargas	53
Figura 19. Configuración tipo de calculo	54
Figura 20. Configuración para ejecutar flujo de carga	55
Figura 21. Flujos de carga del diagrama de 33 nodos	56
Figura 22. Reporte de resultados	57

Figura 23. Pérdidas de energía	57
Figura 24. Cambio a por unidad	59
Figura 25. Código para crear la matriz de incidencia	60
Figura 26. Diagrama de flujo para la creación de A y Zr	61
Figura 27. Programación del método barrido iterativo	63
Figura 28. Pérdidas Sloss	64
Figura 29. Diagrama de flujo del método barrido iterativo	65
Figura 30. Formación de la Ybus	67
Figura 31. Programación del método de aproximaciones sucesivas	68
Figura 32. Diagrama de flujo para el método aproximaciones sucesivas	69
Figura 33. Radiación en Medellín	71
Figura 34. Demanda en Colombia	72
Figura 35. Ajuste de demanda de la matriz sistema y Node	74
Figura 36. Cálculo de tensiones	76
Figura 37. Tensión sin Gd	78
Figura 38. Parámetros del sistema	79
Figura 39. Grafica de la corriente sin GD	81
Figura 40. Grafica de pérdidas de energía hora a hora	82
Figura 41. Diagrama de flujo para el flujo de carga hora a hora sin GD	83
Figura 42. Inclusión de los generadores	84
Figura 43. Ciclo de inyección de potencia de los generadores	85
Figura 44. Grafica de voltajes hora a hora con generación manual	87
Figura 45. Grafica de corrientes hora a hora con generación manual	88

	Figura 46. Grafica de pérdidas de energía hora a hora con generación manual	90
	Figura 47. Diagrama de flujo para los flujos de potencia con generación manual	91
	Figura 48. Función objetivo	93
	Figura 49. Pseudocódigo PSO	93
	Figura 50. Parámetros de inicio del método PSO	95
	Figura 51. Velocidades del algoritmo	96
	Figura 52. Programación de la posición inicial $X_i$	96
	Figura 53. Evaluación de la función objetivo	97
	Figura 54. Actualización y almacenamiento de datos	98
	Figura 55. Restricción del espacio de búsqueda	98
	Figura 56. Segunda evaluación de la función objetivo	99
	Figura 57. Aprendizaje de la técnica de optimización	99
	Figura 58. Método de avance de la técnica de optimización	
100		
	Figura 59. Restricción del espacio de búsqueda actualizado	
101		
	Figura 60. Diagrama de flujo de la programación de los flujos óptimos con PSO	102
	Figura 61. Pseudocódigo del SCA	103
	Figura 62. Parámetros de inicio SCA	104
	Figura 63. Método de aprendizaje SCA	105
	Figura 64. Método de avance SCA	106
	Figura 65. Restricción del espacio de búsqueda SCA	106
	Figura 66. Diagrama de flujo del SCA	107

Figura 67. Grafica del voltaje con GD optimizada	109
Figura 68. Grafica de la corriente con GD optimizada	111
Figura 69. Grafica de las pérdidas de energía con GD optimizada	112
Figura 70. Grafica la corriente con GD optimizada y sin GD	116
Figura 71. Grafica del voltaje con GD optimizada y sin GD	118
Figura 72. Grafica de las paridas de potencia con GD optimizada y sin GD	120
Figura 73. Diagrama de barras de las pérdidas totales de potencia con GD optimizada y sin GD	122

## **Introducción**

La propuesta de una estrategia para la inyección de flujo óptimo de potencia a la red eléctrica, surge porque, el sistema energético colombiano se ha visto en la necesidad de cambiar su planeamiento en la generación de energía eléctrica, debido a la aparición de la generación distribuida, a la mano con las nuevas tecnologías que llegaron para reemplazar los sistemas convencionales, estos ayudan alivianar las cargas en red, pero si no se cuenta con un buen dimensionamiento y una buena ubicación de estos generadores, podrían causar fallos en la red. Uno de los mayores inconvenientes que presenta la red sin la inclusión de estas nuevas tecnologías, son las pérdidas de energía que están relacionadas con el transporte de energía. En este documento se observa la caracterización cada elemento de una red de distribución, se muestra una serie de pasos donde se pueda seleccionar una estrategia por medio de flujos óptimos de potencia, que permita reducir las pérdidas de energía en la red y mejorar la entrada de energía a cada consumidor, se presentan métodos para hallar flujos de carga matemáticamente, así mismo por medio de softwares especializados, identificar fácilmente los cambios que tiene la red eléctrica con generación distribuida por medio de energía solar fotovoltaica y sin la inclusión de esta misma, se observa los factores positivos y negativos al inyectar flujos óptimos, con la finalidad de obtener un sistema energético sostenible.

La estructura de esta investigación está bajo los siguientes capítulos: En el capítulo uno se presenta los objetivos del proyecto, en el capítulo dos se muestra la profundización de los conceptos y parámetros de los flujos de carga y métodos de optimización, en el capítulo tres, se evidencia la programación de los flujos de carga de un sistema de distribución en Matlab, en el capítulo cuatro se describen los pasos para obtener los flujos óptimos de potencia, en el capítulo

cinco se observa el análisis y resultados, finalmente se presentan la conclusiones y recomendaciones del proyecto. Así como referencias bibliográficas y demás anexos.

## 1. Objetivos

### 1.1 Objetivo General

Proponer y evaluar una estrategia de flujo óptimo de potencia en redes eléctricas de corriente alterna para reducción de pérdidas de energía considerando paneles fotovoltaicos.

### 1.2 Objetivos Específicos

- Seleccionar e identificar los parámetros y variables de un sistema de prueba clásico para redes de corriente alterna.
- Seleccionar un método de flujo de carga para evaluar las pérdidas de energía en sistemas de distribución en corriente alterna.
- Proponer una estrategia teniendo en cuenta el flujo de potencia óptimo para reducción de pérdidas de energía, considerando sistemas de generación distribuida con energía solar.
- Evaluar la estrategia de flujo óptimo propuesta en los escenarios de prueba seleccionados para validar su efectividad en relación con la reducción de pérdidas de energía.



## **2. Flujos de Carga en Sistemas de Distribución y Métodos de Optimización**

En este capítulo se describen las características matemáticas de los métodos de flujo de potencia implementados en este proyecto, los cuales son aproximaciones sucesivas y barrido iterativo, además presenta una profundización sobre los conceptos de cada método utilizado y sus tiempos de respuesta. De igual forma se muestran las herramientas que brindan los métodos de optimización, se estudia el concepto de la función objetivo, las variables que la conforman y sus restricciones. De esta manera se encuentra que método de optimización se adapta mejor a este trabajo de investigación.

### **2.1 Conceptos Relevantes de un Sistema de Distribución y Flujo de Carga**

Un sistema de distribución es el encargado de transportar la energía desde el punto más grande de generación dentro de un SEP hasta las viviendas e industrias, cabe resaltar que los diseños de estos sistemas pueden ser radiales, en anillo o mayados. En este documento se analiza el comportamiento de una red radial, donde la corriente fluye por un único camino para llegar a cualquier carga conectada a los nodos, como se observa en la figura 1, estas redes radiales particularmente se analizan por métodos basados en gráficos, esto significa que la red tiene forma de árbol donde se va ramificando por el sistema, así mismo los sistemas de distribución están conformados por distintos elementos como son: SLACK, barras o nodos, cargas o consumos, líneas y transformadores, siendo estos los componentes principales que conforman su funcionamiento.

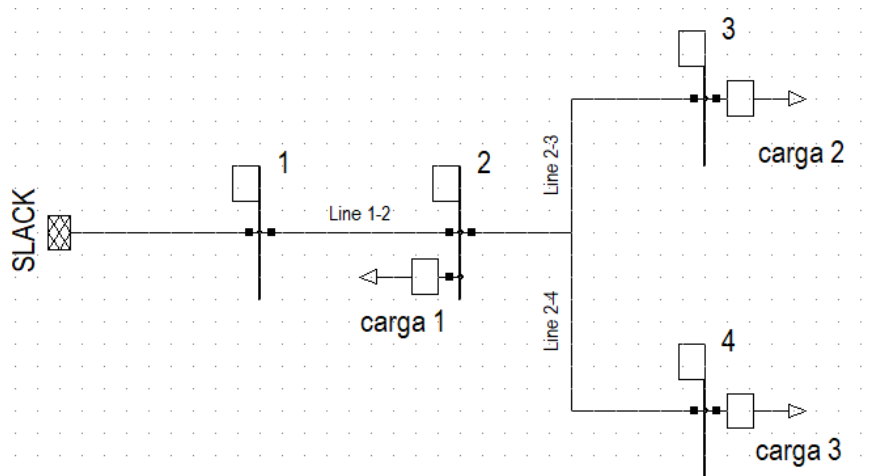


Figura 1. Sistema de distribución radial de 4 nodos

fuelle: (Schmieg, 1985)

Así mismo cuando se analiza una red de distribución se tiene en cuenta que la relación entre  $R/X \gg 1$ , es grande a comparación con la relación en redes de transmisión que es de  $X/R \gg 1$ , esto tiene como característica particular, que la función está asociada al espacio entre los conductores como se observa en la figura 2, en distribución la distancia es del orden de “cm”, como el conductor tiene un calibre, este depende del diámetro del conductor y del material, teniendo una resistencia  $R$  asociada, en redes de distribución la inductancia  $L$  y la capacitancia  $C$  dependen de la geometría de las líneas, como la distancia está en “cm” el efecto de  $L$  y  $C$  es menor, esto significa que crece la relación  $R/X$  a comparación con las redes de transmisión.



*Figura 2. Distancia entre conductores*

*fuelle: (ELECTRONICA R2M, n.d.)*

Es importante a la hora de expandir la red de distribución, establecer una acción de mejora en su operación o revisar limitaciones térmicas, tener conocimiento claro al aplicar métodos que permitan obtener los flujos de carga FC, bajo escenarios de generación y demanda conocidos en los nodos, ya que en estos puntos se centran las variables de interés de magnitud “V” y Angulo “ $\delta$ ” de todas las barras con su potencia activa y reactiva, este procedimiento llevara a encontrar y calcular las pérdidas de energía, regulación de tensión y cortocircuito, para poder encontrar estas variables se recurre a los métodos numéricos, entre los más conocidos se encuentra el Newton-Raphson, Gauss-Seidel o los nuevos como aproximaciones sucesivas, barrido iterativo entre otros, según (Gönen, 1988) alrededor de 1986 se creó dos versiones del Newton-Raphson para su solución, esto era importante en su tiempo ya que los dispositivos de cómputo no eran tan eficientes como los de la actualidad, se necesitaba algo que resolviera razonablemente rápido el problema y se inventaron las versiones desacoplado y desacoplado rápido, en la actualidad programas como DIGSILENT y ETAP internamente tienen el método completo de Newton-Raphson incluido en sus soluciones.

### 2.1.1 Matriz de Admitancia

La matriz de admitancia  $Y_{barra}$  tiene una representación física, esto significa, como es la conexión directa entre los diferentes nodos de la red, así mismo  $Y_{barra}$  es distinta de la matriz de impedancia  $Z_{barra}$  siendo una la inversa de la otra como se muestra en la ecuación 1, la matriz de impedancia se utiliza generalmente para el análisis de fallas en el sistema y la matriz de admitancia para el análisis de los flujos de carga (Grainger & Stevenson Jr, 1996).

$$Y_{barra} = Z_{barra}^{-1} \quad (1)$$

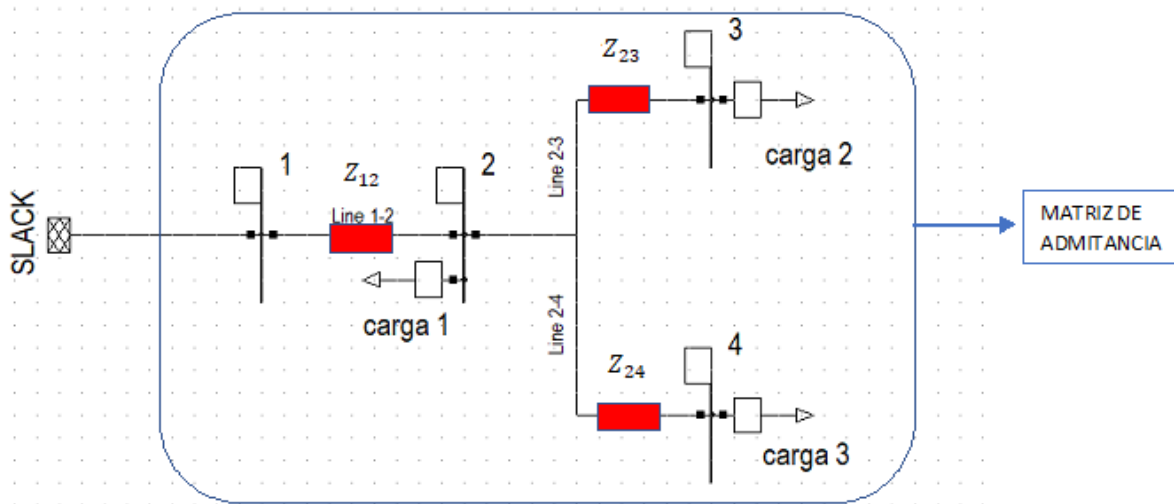


Figura 3. Impedancias de un sistema de distribución

$$Y_{13} = [0]$$

$$Y_{14} = [0]$$

Como se puede observar en la figura 3, un sistema de distribución sencillo de 4 nodos, donde se encuentran algunas características de una red eléctrica, se puede notar las impedancias de cada línea y se relaciona con la matriz de admitancia  $Y_{barra}$  al ser la inversa de la impedancia

$Z_{barras}^{-1}$ , así mismo se puede ver que no existe línea directa entre el nodo 1 y 3 o 3 y 4, esto significa que no hay una admitancia, solo existirán valores en las admitancias  $Y_{12}, Y_{23}, Y_{24}$  el resto de valores dentro de la matriz de admitancia son cero.

### 2.1.2 Ecuaciones para el Flujo de Potencia de una Red Eléctrica

$$SS_k = V_k \times I_k^* \quad (2)$$

$$I_k = I_{gk} - I_{dk} \quad (3)$$

$I_k$  = Corriente neta inyectada

$I_{gk}$  = Corriente generada del nodo k

$I_{dk}$  = Corriente demandada del nodo k

$SS_k$  = Potencia aparente en un nodo k

$V_k$  = Voltaje del nodo k

$I_k^*$  = Corriente conjugada del nodo k

$$\begin{pmatrix} Y_{11} & Y_{12} & Y_{1k} & Y_{1n} \\ Y_{21} & Y_{22} & Y_{2k} & Y_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ Y_{n1} & Y_{n2} & Y_{nk} & Y_{nn} \end{pmatrix} = Y_{bus}$$

$$\begin{bmatrix} V_1 \\ V_2 \\ V_k \\ V_n \end{bmatrix} = V \quad \text{Vector columna de voltajes del sistema}$$

$$\begin{bmatrix} I_1 \\ I_2 \\ I_k \\ I_n \end{bmatrix} = I \quad \text{Vector columna de las corrientes del sistema}$$

Ley de ohm escrita en la forma de admitancia:

$$[I] = [V] [Y_{bus}] \quad (4)$$

$$\begin{pmatrix} I_1 \\ I_2 \\ I_k \\ I_n \end{pmatrix} = \begin{pmatrix} V_1 \\ V_2 \\ V_k \\ V_n \end{pmatrix} \begin{pmatrix} Y_{11} & Y_{12} & Y_{1k} & Y_{1n} \\ Y_{21} & Y_{22} & Y_{2k} & Y_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ Y_{n1} & Y_{n2} & Y_{nk} & Y_{nn} \end{pmatrix}$$

$$I_k = Y_{k1} \times V_1 + Y_{k2} \times V_2 + \dots + Y_{kn} \times V_n = \sum_{j=1}^n Y_{kj} \times V_j \quad (5)$$

Partiendo de la ecuación 2, la cual se puede reescribir, de tal forma que al reemplazar la corriente  $I_k$  no quede conjugada, ya que ésta representa la corriente inyectada al sistema que se observa en la ecuación 6 (Grainger & Stevenson Jr, 1996).

$$SS_k^* = V_k^* \times I_k \quad (6)$$

Reemplazando la ecuación (5) en la (6).

$$SS_k^* = V_k^* \times \sum_{j=1}^n Y_{kj} \times V_j \quad (7)$$

$SS_k$  = Potencia aparente neta inyectada

$Y_{kj}$  = Matriz de admitancia que relaciona los nodos k y j

$V_k$  = Voltaje complejo del nodo k

$V_j$  = Voltaje complejo del nodo j

La ecuación 7, es la que se usa en todos los métodos de flujo de potencia; la variable “k” representan los nodos de envío y “j” los de recibo en el sistema, aprovechando las propiedades de una red de distribución radial ilustrada en la figura 4. Esta red radial se puede separar en una representación de dos colores, el rojo para la generación y el azul para la demanda, donde se

puede notar que solo hay un nodo SLACK y cuatro nodos de demanda, los cuales tienen como características la siguiente información:

- SLACK = Voltaje conocido  $1 \angle \theta$  el Angulo es constante con un valor definido.
- Demanda = Voltaje desconocido.

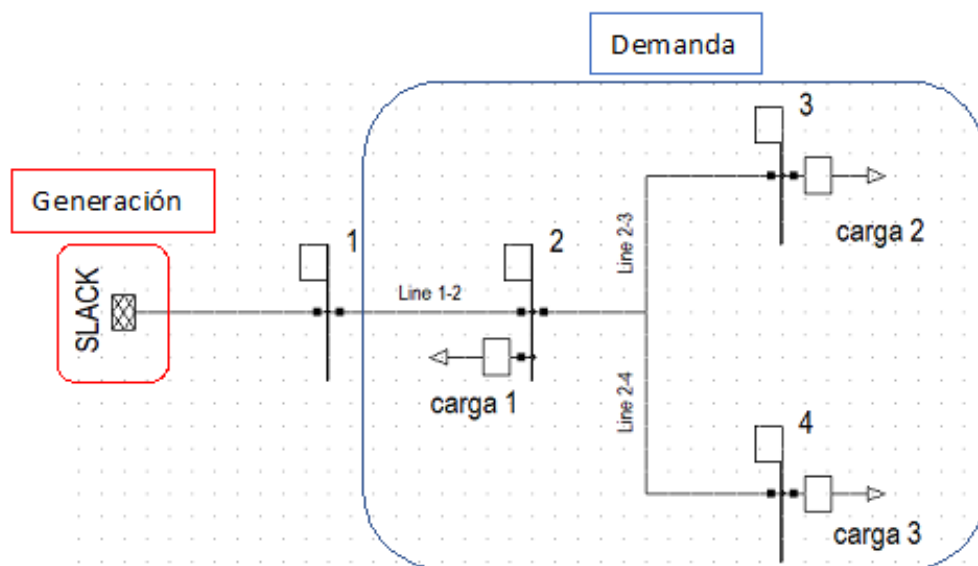


Figura 4. Generación y demanda en un sistema de distribución

## 2.2 Métodos de Flujo de Carga

La finalidad de analizar los resultados de los flujos de potencia es permitir realizar medidas correctivas para mejorar la operación del sistema. Estos métodos se clasifican en versiones, clásicas y nuevas. Este proyecto está enfocado en los métodos más actuales.

### 2.2.1 Aproximaciones Sucesivas

Este método aprovecha la estructura de un sistema radial o mayado donde hay solo un nodo SLACK y el sistema es representado por una matriz de admitancia llamada  $Y_{bus}$ . Así mismo la dimensión de la matriz dependerá de cuántos nodos tenga el sistema.

Aclarando las posiciones de las asociaciones de generación y demanda, se separa la matriz  $Y_{bus}$  en cuatro sectores como lo ilustra la figura 5, los cuales cada uno tiene una función dentro de la matriz. En el sector azul se puede encontrar  $Y_{gg}$ , siendo ésta la representación del valor de admitancia en el nodo de generación; el naranja  $Y_{gd}$  describe la admitancia entre los nodos de generación y demanda de la red; el rojo es  $Y_{dg}$  que representa la admitancia entre los nodos de demanda y generación del sistema y finalmente el verde  $Y_{dd}$  que representarán las admitancias de los nodos de demanda del sistema. Cabe resaltar que  $Y_{gg}$  y  $Y_{dd}$  son matrices cuadradas donde la dimensión de  $Y_{gg}$  es de  $g \times g$  y la de  $Y_{dd}$  es  $d \times d$ , igualmente, las matrices  $Y_{gd}$  y  $Y_{dg}$  son rectangulares las cuales se relacionan por la ecuación 8, ya que la matriz  $Y_{bus}$  es simétrica (Herrera Briñez, 2021).

$$\begin{pmatrix} \boxed{Y_{11}} & \boxed{Y_{12}} & \boxed{Y_{1k}} & \boxed{Y_{1n}} \\ \boxed{Y_{21}} & \boxed{Y_{22}} & \boxed{Y_{2k}} & \boxed{Y_{2n}} \\ \vdots & \vdots & \vdots & \vdots \\ \boxed{Y_{n1}} & \boxed{Y_{n2}} & \boxed{Y_{nk}} & \boxed{Y_{nn}} \end{pmatrix} = Y_{bus}$$

Figura 5. Componentes de la matriz  $Y_{bus}$

$$\overbrace{\begin{bmatrix} Y_{gg} & Y_{gd} \\ Y_{dg} & Y_{dd} \end{bmatrix}}^{Y_{bus}} \begin{bmatrix} V_g \\ V_d \end{bmatrix} = \begin{bmatrix} I_g \\ I_d \end{bmatrix}$$

$$Y_{gd} = Y_{dg}^T \quad (8)$$



La finalidad de este proceso es obtener los voltajes de demanda  $V_d$ , así que su enfoque está en la operación de todos los componentes de la fila donde se encuentra  $I_d$  y se expresa en la ecuación 9.

$$I_d = [Y_{bus}][V] \quad (9)$$

$$I_d = Y_{dg} \times V_g + Y_{dd} \times V_d \quad (10)$$

Como se observa en la ecuación 10, se tiene una variable de generación  $V_g$  y una variable de demanda  $V_d$ , donde  $V_d$  es un vector que incluye los componentes según las barras del sistema. Por ejemplo, si se toma como referencia la figura 1,  $V_d$  tendría tres componentes de demanda y sería un vector 3x1. En un sistema de potencia también se puede encontrar en la ecuación 11, una aplicación general para muchos nodos, partiendo de esta ecuación se aprovechan las ventajas de la red de distribución.

$$SS_g^* - SS_d^* = \text{Diag}(V)^*[Y_{bus}][V] \quad (11)$$

Teniendo en cuenta la matriz  $Y_{bus}$  que relaciona la generación con la demanda, se puede aplicar a la ecuación 11, para expandirla en 13 y 15.

- Ecuación Aplicada al Nodo SLACK

En la ecuación 13, se conoce el voltaje del nodo SLACK  $V_g$ , así como los valores de  $Y_{gg}$ ,  $Y_{gd}$ ,  $V_g$ , donde la única variable desconocida es  $V_d$ , esta ecuación es lineal porque  $SS_g^*$  absorbe lo que pide la demanda, junto con las pérdidas a un extremo de la ecuación y al otro se tiene  $V_d$  que también es variable, pero los términos son constantes. Se puede hacer una analogía utilizando la ecuación 12.

$$Y = A + BX \quad (12)$$

$$SS_g^* = \text{Diag}(V_g)^*[Y_{gg} \times V_g + Y_{gd} \times V_d] \quad (13)$$

- Ecuación Aplicada a los Nodos de Demanda

Existe una diferencia particular con la ecuación aplicada a los nodos de demanda y es que no es lineal. Si se hace una analogía con la ecuación 14 y 15, se puede evidenciar que en la ecuación 14, existen dos variables “X” y se multiplican entre ellas, dando como resultado una ecuación no lineal, siendo la variable  $V_d$  en la ecuación 15 la que se presenta en dos ocasiones.

$$C = \text{Diag}(X)[D + E(X)] \quad (14)$$

$$-SS_d^* = \text{Diag}(V_d)^*[Y_{dg} \times V_g + Y_{dd} \times V_d] \quad (15)$$

Se evidencia en la ecuación 11 la existencia de una diagonal. Si se crea un vector  $[V_1 \ V_2 \ V_3]$  dentro de una diagonal, indica que va a transformar ese vector en una matriz donde la diagonal solo sean sus componentes y el resto se rellenan con ceros. En caso de un análisis para el sistema de distribución, el vector que va a estar dentro de la diagonal va a depender de los componentes de los voltajes propios, de la dimensión de los nodos del sistema.

$$\text{Diagonal}([V_1^* \ V_2^* \ V_3^*]) = \begin{pmatrix} V_1^* & 0 & 0 \\ 0 & V_2^* & 0 \\ 0 & 0 & V_3^* \end{pmatrix}$$

Se parte de la ecuación 15 no lineal, porque se conoce la potencia demandada  $SS_d^*$  de los nodos, y se despeja el termino  $V_d$ .

$$V_d = -(Y_{dd})^{-1}(\text{Diag}^{-1}(V_d)^* SS_d^*) + Y_{dg} \times V_g \quad (16)$$

Para su solución se agrega un proceso iterativo y se define una variable “t”, que representan el número de iteraciones para el cálculo de los voltajes, de esta manera se van almacenando todos los voltajes del sistema. Esto se repite cada vez, hasta que se llegue al error

de convergencia, que tiene como característica la condición  $|V_d^{t+1} - V_d^t| \leq \varepsilon$ . Si se cumple esta condición se rompe el ciclo de iteraciones y se reporta la totalidad de los voltajes  $V_d^{t+1}$  de un sistema, en este caso radial.

$$V_d^{t+1} = -(Y_{dd})^{-1}(\text{Diag}^{-1}(V_d^t)^* SS_d^*) + Y_{dg} \times V_g \quad (17)$$

La ecuación 17 que representa el método de aproximaciones sucesivas tiene como ventaja, que la mayoría de información de la ecuación es conocida, además no utiliza derivadas como lo hace el Newton-Raphson; así que el tiempo se reduce a la inversa de una diagonal y los resultados ya están definidos en magnitud y Angulo siendo más sencillo su análisis.

Para finalizar, teniendo los flujos de potencia del sistema  $V_n$  y la  $Y_{bus}$  completa, con la definición de la ecuación 9, se calculan todas las corrientes netas inyectadas.

$$I = Y_{bus} \times V_n = I_g - I_d \quad (18)$$

Conjugando la ecuación 18 y multiplicando en ambos lados por una diagonal de todos los voltajes del sistema “V”, se obtiene la potencia neta inyectada en todos los nodos. En la ecuación 19 y 20 se tiene la potencia neta inyectada total del sistema. De esta manera se puede extraer información sobre las pérdidas de energía activa y reactiva.

$$\text{Diag}(V)I^* = \text{Diag}(V)Y_{bus}^* \times V_n^* = SS = SS_g - SS_d \quad (19)$$

$$SS_{loss} = V^T \times I^* = V^T \times (Y_{bus}^* \times V_n^*) = P_{loss} + jQ_{loss} \quad (20)$$

### 2.2.2 Barrido Iterativo

El inicio del proceso de este método es definir los sentidos de la corriente de forma arbitraria en una red de distribución, se indica con flechas rojas las corrientes y asigna un nombre a cada rama del sistema como se observa en la figura 5. Se presenta la existencia de demandas en

los nodos del sistema. Así mismo se nombran las corrientes netas inyectadas  $I_k$  ingresando a los nodos de demanda indicados con flechas azules.

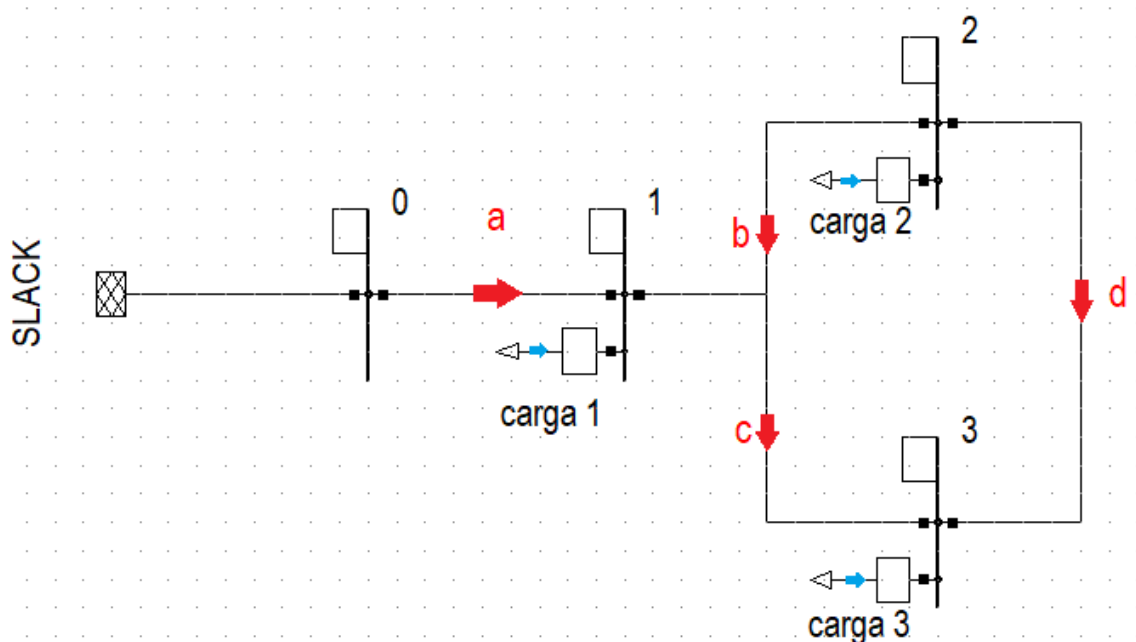


Figura 6. Asignación de corrientes

- Matriz de Incidencia A

Existen dos tipos de matrices de incidencia, la primera se establece por ser nodo-rama y la segunda rama-nodo, se relacionan por ser una la transpuesta una de la otra. Para profundizar en estos conceptos se escoge el tipo nodo-rama, teniendo en cuenta que las filas están representadas por los nodos y las columnas por las ramas; si se escoge el tipo rama-nodo es al contrario (Herrera Briñez, 2021). Esta matriz tiene como característica:

$$A_{ij} = \begin{cases} "1" & \text{si la rama } j \text{ esta conectada al nodo } i \text{ y la corriente sale del nodo } i \\ "-1" & \text{si la rama } j \text{ esta conectada al nodo } i \text{ y la corriente llega al nodo } i \\ "0" & \text{si la rama } j \text{ no esta conectada al nodo } i \end{cases}$$

Se tiene en cuenta la figura 6 para el análisis y cálculo de la matriz de incidencia nodo-rama, según las condiciones de corriente establecidas, ingresando y saliendo en los nodos. Se observa en el resultado de la matriz A, en cada columna existe una corriente saliendo "1" y una entrando "-1", por lo que solo hay una conexión directa entre dos nodos por rama.

$$A = \begin{matrix} & a & b & c & d \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{pmatrix} \end{matrix}$$

De igual manera se define la matriz A en dos componentes  $A_0$  y  $A_d$ , esto se hace para separar el nodo de generación slack del resto de nodos de demanda, indicando que las propiedades del nodo de generación son las del rectángulo azul y las de demanda en el rectángulo naranja.

$$A = \begin{pmatrix} \boxed{1 \ 0 \ 0 \ 0} \\ \boxed{-1 \ 1 \ 1 \ 0} \\ \boxed{0 \ -1 \ 0 \ 1} \\ \boxed{0 \ 0 \ -1 \ -1} \end{pmatrix}$$

$$A = \begin{pmatrix} A_0 \\ A_d \end{pmatrix}$$

Con base a la estructura de la figura 6, se calcula los voltajes de cada rama j.

$$\begin{pmatrix} \overbrace{V_a} \\ \overbrace{V_b} \\ \overbrace{V_c} \\ \overbrace{V_d} \end{pmatrix} = \begin{pmatrix} \overbrace{1 \ -1 \ 0 \ 0}^{A^T} \\ \overbrace{0 \ 1 \ -1 \ 0} \\ \overbrace{0 \ 1 \ 0 \ -1} \\ \overbrace{0 \ 0 \ 1 \ -1} \end{pmatrix} \begin{pmatrix} \overbrace{V_0} \\ \overbrace{V_1} \\ \overbrace{V_2} \\ \overbrace{V_3} \end{pmatrix}$$

$$V_r = A^T V_n \quad (21)$$

Se nota que al hacer una comparación con la matriz resultante del análisis de la diferencia de voltajes  $A^T$ , es la transpuesta de la matriz de incidencia  $A$ , también se establece el vector de voltajes de las ramas  $V_r$  y el vector de los voltajes de los nodos  $V_n$  del sistema y se representa en la ecuación 21, reescribiendo la ecuación y separando los términos de generación y demanda se tiene la ecuación 22, dejando como incógnita a  $V_d$ .

$$V_r = A_o^T V_0 + A_d^T V_d \quad (22)$$

Aplicando en el sistema mayado una sumatoria de corrientes en cada barra.

$$\sum I_1 = 0$$

$$I_1 + I_a - I_b - I_c = 0 \quad (23)$$

$$\sum I_2 = 0$$

$$I_2 + I_b - I_d = 0 \quad (24)$$

$$\sum I_3 = 0$$

$$I_3 + I_c + I_d = 0 \quad (25)$$

$$\begin{matrix} I_d \\ \overbrace{\begin{pmatrix} I_1 \\ I_2 \\ I_3 \end{pmatrix}} \end{matrix} = \begin{matrix} A_d \\ \overbrace{\begin{pmatrix} -1 & 1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & -1 \end{pmatrix}} \end{matrix} \begin{matrix} I_r \\ \overbrace{\begin{pmatrix} I_a \\ I_b \\ I_c \\ I_d \end{pmatrix}} \end{matrix}$$

$$I_d = A_d I_r \quad (26)$$

Se nota en la ecuación 27, la corriente y el voltaje en cada rama se relacionan por medio de la impedancia, independientemente del resto de las impedancias de otras líneas, esto significa

que se puede ajustar una matriz con las componentes  $V_r, Z_r, I_r$ , donde  $Z_r$  es una matriz diagonal con las impedancias de las ramas del sistema y depende de la dimensión de la red.

$$V_r = Z_r I_r \quad (27)$$

$$\overbrace{\begin{pmatrix} V_a \\ V_b \\ V_c \\ V_d \end{pmatrix}}^{V_r} = \overbrace{\begin{pmatrix} Z_a & 0 & 0 & 0 \\ 0 & Z_b & 0 & 0 \\ 0 & 0 & Z_c & 0 \\ 0 & 0 & 0 & Z_d \end{pmatrix}}^{Z_r} \overbrace{\begin{pmatrix} I_a \\ I_b \\ I_c \\ I_d \end{pmatrix}}^{I_r}$$

Se necesita hallar una relación que despeje a  $V_d$  en función de  $I_d$ , como  $Z_r$  es invertible por ser una matriz cuadrada, se despeja  $I_r$  de la ecuación 27 y se sustituye en la ecuación 26.

$$I_r = Z_r^{-1} V_r \quad (28)$$

$$I_d = A_d Z_r^{-1} V_r \quad (29)$$

Se reemplaza la ecuación 22 en la 29.

$$I_d = A_d Z_r^{-1} (A_0^T V_0 + A_d^T V_d) \quad (30)$$

Reorganizando.

$$I_d = \underbrace{A_d Z_r^{-1} A_0^T}_{Y_{d0}} V_0 + \underbrace{A_d Z_r^{-1} A_d^T}_{Y_{dd}} V_d \quad (31)$$

Simplificando la ecuación 31, se establece que  $Y_{d0}$  es reemplazado por " $A_d Z_r^{-1} A_0^T$ ", donde relaciona todas las demandas con el nodo SLACK y  $Y_{dd}$  es sustituido por " $A_d Z_r^{-1} A_d^T$ ", porque ambos términos relacionan voltajes demandados  $A_d$  con corrientes demandadas  $A_d^T$  y la simplificación se muestra en la siguiente ecuación.

$$I_d = Y_{d0} V_0 + Y_{dd} V_d \quad (32)$$

De esta manera se puede despejar la variable  $V_d$ , al notar que  $Y_{dd}$  es invertible por ser una matriz cuadrada como evidencia en la ecuación 33.

$$V_d = Y_{dd}^{-1}(I_d - Y_{d0}V_0) \quad (33)$$

Por último, se identifica  $I_d$ , depende del término de voltajes y potencias, está definida por la ecuación 34 y se reemplaza en 33, para finalizar el procedimiento se procede a aplicar los contadores para las iteraciones y de esta manera obtener todos los flujos de carga del sistema, como muestra en la ecuación 36, de esta manera se tiene la ecuación que representa el método barrio iterativo fundamentalmente expresado de forma general. Así mismo se pueden realizar para redes radiales y mayadas.

$$I_d = -\text{Diag}^{-1}(V_d^*)SS_d^* \quad (34)$$

$$V_d = -Y_{dd}^{-1}(\text{Diag}^{-1}(V_d^*)SS_d^* + Y_{d0}V_0) \quad (35)$$

$$V_d^{t+1} = -Y_{dd}^{-1}(\text{Diag}^{-1}(V_d^*)^tSS_d^* + Y_{d0}V_0) \quad (36)$$

### 2.3 Optimización

Uno de los aportes más importantes en este trabajo está relacionado con la aplicación de modelos de optimización, es una de las ramas de las matemáticas conocida como investigación de operaciones, aplicando y formulando métodos matemáticos cuya implementación se enfoca en los procesos de toma de decisiones (Cordero et al., 2019). En la actualidad existen diversos métodos de optimización, muchas de las técnicas se han obtenido mediante la interpretación en el comportamiento tanto de seres vivo, así como de leyes universales que rigen el universo (Maynard Smith, 1978). Muchos autores suelen dividir la optimización en dos principales técnicas, las cuales pueden ser determinista (heurística) como estocástica (metaheurística), cuyo fin busca hacer optimo los recursos disponibles (Pérez Peña, 2019).

La mayoría de los problemas no se pueden resolver empleando una sola técnica de estudio, uno de estos factores se debe a que las soluciones a estos problemas pueden ser de un



solo objetivo a solucionar (mono objetivo) o a múltiples objetivos (multiobjetivo), por ende, estas categorías presentan principios diferente a la hora de encontrar la mejor respuesta (Pérez Peña, 2019). A la hora de realizar estas técnicas, es importante conocer los términos como mejor solución o solución óptima, el cual es el final del proceso de cualquier algoritmo empleado; sin embargo, existen muchos factores en los cuales podría determinar si estos métodos producirán éxito o fracaso, esto depende de una gran cantidad de variables que a lo mejor no se tuvieron en cuenta pero eran necesaria para que el método funcionara correctamente, también influye mucho el conocimiento que se tenga sobre el tema y el no saber programarlos en los softwares especializados (Pérez Peña, 2019).

En general las técnicas de optimización basadas en la población inician el proceso de optimización con un conjunto aleatorio de soluciones. Este conjunto aleatorio es evaluado varias veces por una función objetivo y mejorado por un conjunto de reglas que están en el núcleo de estos métodos (Mirjalili, 2016). Estos métodos se utilizan cuando no existe un algoritmo específico para este fin, se deben utilizar varios casos de prueba para validar el rendimiento de un algoritmo (G.-G. Wang et al., 2014).Es recomendable a hora de realizar cualquier tipo de técnicas, contar con un número suficiente de soluciones aleatorias y pasos de iteraciones necesarias, aumenta la probabilidad de encontrar el valor óptimo general (Schweickardt & Miranda, 2010). Su principal desventaja es que no garantizan que se logre el óptimo absoluto.

Los elementos fundamentales que se deben tener en cuenta a la hora de realizar cualquier técnica de optimización son las siguientes:

### ***2.3.1 Variables de decisión***

Para la elaboración de cualquier método de optimización como primer paso, se deben seleccionar las suficientes variables independientes que permitan describir los posibles diseños candidatos y las condiciones de operación del sistema, el objetivo de este ítem es seleccionar las que tengan un impacto significativo en la función objetivo (Meneu Gaya et al., 1999).

Las variables independientes se representarán mediante vectores columna de  $\mathbb{R}^n$

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

o vectores fila

$$x^T = (x_1, \dots, x_n)$$

### 2.3.2 Restricciones

Siguiendo el orden propuesto, el siguiente paso es establecer las relaciones entre las variables de decisión mediante ecuaciones o desigualdades, estas variables están obligadas a satisfacer las condiciones que el problema a optimizar requiera (Ramos et al., 2010). Existen muchos tipos de restricciones las cuales van desde las leyes de la naturaleza, restricciones tecnológicas, entre otras cosas (Meneu Gaya et al., 1999), Podemos distinguir dos tipos de restricciones:

Restricciones de igualdad: Son ecuaciones entre las variables de la forma

$$h(x) = h(x_1, \dots, x_n) = 0 \quad (37)$$

donde  $h = A \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  es una función real de variables reales definidas en el conjunto de  $A$  de los números reales.

Restricciones de desigualdad: Son inecuaciones entre las variables de la forma

$$g(x) = g(x_1, \dots, x_n) \leq 0 \quad (38)$$

Donde  $g = A \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  es una función real de variables reales definidas en el conjunto de  $A$  de los números reales.

### **2.3.3 Función objetivo**

Para obtener una respuesta en un problema de optimización es necesario emplear la función objetivo, también conocida como índice de desempeño o criterio de selección. Es el factor que se utiliza para decidir los valores adecuados de las variables de decisión que se desea optimizar, para determinar los mejores valores, muchos de estos problemas buscan la minimización o maximización de un problema (Ramos et al., 2010).

Antes de profundizar ampliamente sobre las técnicas de optimización empleadas en este trabajo, se debe tener en cuenta antes de la realización de estas, existe una serie de parámetros y restricciones de los flujos óptimos de potencia, el cual, a continuación, se mencionará cada uno de los ítems para tener en cuenta para así aplicar la optimización.

## **2.4 Flujos Óptimos de Potencia**

Los flujos óptimos de potencia cuyas siglas en inglés son (OPF, Optimal Power Flows), son uno de los problemas fundamentales en el análisis de sistemas de energía. Algunas de las investigaciones necesarias para la operación y planificación de la red eléctrica en general requieren una gran cantidad de soluciones OPF iterativas. En estos análisis, la velocidad de convergencia de las soluciones de los flujos óptimos, así como su precisión, son dos objetos importantes (Mosquera & Carrión, 2019).

El objetivo de OPF es encontrar la configuración óptima de una red de sistema de energía dada para optimizar las funciones objetivo del sistema, tales como costos totales de producción, pérdidas del sistema, desviaciones de voltaje de bus, etc., emisión de unidades de producción,

cantidad de acción de control y reducción de carga mientras se satisfacen las ecuaciones actuales, los límites de seguridad del sistema y la operación del equipo (Mosquera & Carrión, 2019).

Los flujos de energía óptimo tienen problemas matemáticos complejos para resolver, debido a que corresponden a fórmulas matemáticas no lineales y no convexas, esto conlleva a un alto esfuerzo computacional para resolverlo. La solución del problema de los flujos óptimos involucra la solución del flujo de carga, así como las características intrínsecas del sistema (López Lezama & Gallego Pareja, 2008). En el caso de los flujos óptimos de potencia, la función objetivo más utilizada por los modelos matemáticos propuestos en la literatura especializada es la minimización de pérdidas de energía. Presentado como restricciones de presupuesto de energía nodal, regulación de voltaje, capacidad de generación distribuida, entre otros (Velasquez Rodríguez, 2019).

#### ***2.4.1 Modelo matemático***

El modelo matemático de la ubicación y el tamaño óptimos de los generadores distribuidos (DG) en una red de distribución radial (RDN) corresponde a un problema de flujos óptimos de potencia del modelo mixto de programación entera no lineal (MINLP). Las variables (binarias) representan las variables de decisión asociadas con la posición o no de un DG en la red, mientras que las variables continuas están asociadas con la fórmula clásica de flujo de potencia, expresada por cantidad y ángulo de tensión en cada nodo (Grisales Noreña & Giraldo Montoya, 2020). A continuación, se representa el modelo matemático de la función objetivo del sistema

$$\min z = \sum_{i \in \Omega_N} V_i \left( \sum_{j \in \Omega_N} V_j Y_{ij} \cos(\theta_i - \theta_j - \phi_{ij}) \right) \quad (39)$$

Donde  $z$  es el valor de la función objetivo, correspondiente a la pérdida de potencia en todas las ramas de la red en la situación de carga máxima;  $\Omega_N$ , el conjunto asociado a los nodos de la red;  $V_i$  y  $V_j$ , las magnitudes de las tensiones en los nodos  $i$  y  $j$ ,  $\theta_i$  y  $\theta_j$  son los ángulos de voltaje en los nodos  $i$  y  $j$ ;  $Y_{ij}$  es la magnitud de la admitancia la cual está asociada con la línea conectada entre  $i$  y  $j$  nodos;  $\phi_{ij}$ , su ángulo.

#### 2.4.2 Restricciones

$$P_i^{CG} + P_i^{DG} = V_i \sum_{j \in \Omega_N} V_j Y_{ij} \cos(\theta_i - \theta_j - \phi_{ij}) + P_i^D, \{\forall i \in \Omega_N\} \quad (40)$$

$P_i^{CG}$  representa la potencia activa generada en el nodo  $i$  por un generador convencional;  $P_i^{DG}$ , la potencia activa generada por un DG ubicado en el nodo  $i$ ; y  $P_i^D$ , la potencia activa total de demanda en el nodo  $i$ .

$$Q_i^{CG} + Q_i^D = V_i \sum_{j \in \Omega_N} V_j Y_{ij} \sin(\theta_i - \theta_j - \phi_{ij}), \{\forall i \in \Omega_N\} \quad (41)$$

$Q_i^{CG}$  indica la potencia reactiva generada en el nodo  $i$  por un generador convencional;  $Q_i^{DG}$ , la potencia reactiva generada por un GD ubicado en el nodo  $i$ ; y  $Q_i^D$ , la potencia reactiva total de demanda en el nodo  $i$ .

$$V_i^{min} \leq V_i \leq V_i^{max}, \{\forall i \in \Omega_N\} \quad (42)$$

$V_i^{min}$  y  $V_i^{max}$  representan los valores de voltaje mínimo y máximo permitidos en cada nodo.

$$0 \leq P_i^{DG} \leq x_i P_i^{DG,max}, \{\forall i \in \Omega_N\} \quad (43)$$

$P_i^{DG,max}$  maxi es la inyección de potencia activa máxima permitida en el nodo  $i$  por un GD y  $x_i$  representa la variable de decisión, que toma un valor de 1 si el GD está ubicado en el nodo  $i$  y 0 en caso contrario. Esta ecuación muestra la posibilidad de ubicar y dimensionar una GD en cualquier nodo de la red de distribución radial. Solo consideramos la inyección de potencia activa en las GD, haciendo  $Q_i^{DG} = 0$ .

$$\sum_{i \in \Omega_N} Vi \leq N_{ava}^{GD} \quad (44)$$

$N_{ava}^{GD}$  es el número de DG disponibles, lo que implica que limita el número de posibilidades de ubicación para la generación distribuida en la red de distribución radial.

Teniendo claro los conceptos de flujo óptimo de potencia se procede a realizar las técnicas de optimización empleadas en este trabajo.

## 2.5 Optimización por Enjambre de Partículas (PSO)

La optimización por enjambre de partículas cuyas siglas en inglés es (PSO, Particle Swarm Optimizarían Algorithm), es una metaheurística basado en poblaciones inspirado en los principios de investigación utilizados por los enjambres de insectos, manadas, aves y peces, en busca de alimento (Jatmiko et al., 2007). Desde su propuesta inicial en 1995, esta técnica se ha ido perfeccionando y aplicando a multitud de problemas por su facilidad de programación, flexibilidad, rapidez y buenas características de convergencia (Eberhart & Kennedy, 1995).

El PSO es un tipo de proceso de búsqueda basado en enjambres en el que cada individuo llamado partícula se identifica como una solución potencial, del problema de optimización en el espacio de búsqueda de dimensión  $D$ , y en cual puede memorizar la posición óptima del enjambre y la propia, así como la velocidad. La información de las partículas se combina para ajustar la velocidad de cada dimensión con cada generación, que se utiliza para calcular la nueva posición de la partícula. Las partículas cambian continuamente de estado en el espacio de búsqueda multidimensional, hasta que alcanzan el equilibrio o el estado óptimo, o superan el límite de los cálculos. Las conexiones únicas entre las diferentes dimensiones del espacio del problema se introducen a través de la función objetivo.

El método inicia cuando cada individuo está representado por un punto en el sistema de coordenadas cartesianas, asignado aleatoriamente con una velocidad y posición iniciales (Clerc

& Kennedy, 2002). Se agrega una variable aleatoria al ítem de velocidad, para lograr que este método obtenga mejores respuestas, se asume que las coordenadas de posición son  $(x_0, y_0)$ , las coordenadas de posición y las coordenadas de velocidad de cada individuo son  $(x, y)$  y  $(v_x, v_y)$ , respectivamente.

La distancia entre la posición actual y el espacio de búsqueda se usa para medir el rendimiento de la posición actual y la velocidad (Clerc & Kennedy, 2002), cuanto mayor sea la distancia al espacio de búsqueda, mejor es el rendimiento, por el contrario, peor es el rendimiento. Cada individuo tiene la capacidad de memorizar y puede recordar la mejor posición que ha alcanzado, designada como  $pbest$ ,  $a$  es la constante de ajuste de velocidad,  $rand$  representa un número aleatorio en  $[0,1]$ , el cambio del elemento de velocidad se puede configurar de acuerdo con las siguientes reglas:

$$Si x > pbestx, v_x = v_x - rand \times a, de lo contrario, v_x = v_x + rand \times a.$$

$$Si y > pbesty, v_y = v_y - rand \times a, de lo contrario, v_y = v_y + rand \times a.$$

A continuación, el enjambre puede comunicarse de alguna manera, en el cual cada individuo puede conocer y recordar la mejor posición (marcada como  $gbest$ ) en el enjambre total hasta la fecha. Donde  $b$  es la constante de control de velocidad, después de ajustar el elemento de velocidad de acuerdo con las reglas anteriores, también debe actualizarse nuevamente de acuerdo con las siguientes reglas:

$$Si x > gbestx, v_x = v_x - rand \times b, de lo contrario, v_x = v_x + rand \times b.$$

$$Si y > gbesty, v_y = v_y - rand \times b, de lo contrario, v_y = v_y + rand \times b.$$

relativamente grande, los individuos se reunirán más rápidamente en el espacio de búsqueda, y si  $a/b$  es pequeño, las partículas se unirán de manera inestable y lenta alrededor del espacio de búsqueda. La corrección que se realizó al algoritmo fue la siguiente manera:

$$v_x = v_x + 2 * rand * (pbestx - x) + 2 * rand * (gbestx - x) \quad (45)$$

$$x = x + v_x \quad (46)$$

Logrando así que cada individuo para que fuera una partícula sin masa ni volumen, con solo velocidad y posición. Una vez tenido en cuenta las anteriores ecuaciones se procede a describir la fórmula de actualización de la posición óptima del individuo la cual es:

:

$$P_{i,t+1}^d = \begin{cases} x_{i,t+1}^d, & \text{Si } f(x_{i,t+1}) < f(p_{i,t}) \\ P_{i,t}^d, & \end{cases} \quad (47)$$

La posición óptima del enjambre es la de todas las posiciones óptimas del individuo. La fórmula de actualización de velocidad y posición se indica de la siguiente manera:

$$v_{i,t+1}^d = v_{i,t}^d + c1 * rand * (P_{i,t}^d - x_{i,t}^d) + c2 * rand * (P_{g,t}^d - x_{i,t}^d) \quad (48)$$

$$x_{i,t+1}^d = x_{i,t}^d + v_{i,t+1}^d \quad (49)$$

Al método tradicional se le agrego unos coeficientes  $c1$  (coeficiente de aprendizaje cognitivo) y  $c2$  (coeficiente social), estos coeficientes representan los pesos de los términos de aceleración aleatoria que atraen cada partícula a  $pBest$  y  $gBest$  (o  $nBest$ ). En muchos casos,  $c1$  y  $c2$  se establecen en 2.0, por lo que la búsqueda incluye el área de enfoque en  $pBest$  y  $gBest$ . Otro valor común es 1.49445 que puede garantizar la convergencia del algoritmo PSO el cual fue propuesto en (Clerc & Kennedy, 2002), para efectos de este trabajo se asumió el valor anteriormente mencionado.

El peso inercial se ingresa en la fórmula de actualización de velocidad, la nueva fórmula de actualización de velocidad se convierte en:

$$v_{i,t+1}^d = \omega * v_{i,t}^d + c1 * rand * (P_{i,t}^d - x_{i,t}^d) + c2 * rand * (P_{g,t}^d - x_{i,t}^d) \quad (50)$$



Los estudios actuales creen que el peso de inercia tiene la mayor influencia en el rendimiento del algoritmo PSO, también se utiliza para equilibrar la búsqueda global, la búsqueda local y las ponderaciones de inercia más altas tienden a la búsqueda global mientras que las ponderaciones son más altas. En (Shi & Eberhart, 1998) sugirieron que establecer el peso inercial en [0.9, 1.2], para este trabajo se tomó un peso de inercia de 0.7 cumpliendo con lo establecido anteriormente.

El enjambre de partículas además de buscar a su propia posición óptima  $pBest$ , la partícula no busca a la posición óptima del cúmulo  $gBest$ , sino a la posición óptima  $nBest$  de todas las partículas de la vecindad. La ecuación de actualización de velocidad se convirtió en:

$$v_{i,t+1}^d = \omega * v_{i,t}^d + c1 * rand * (P_{i,t}^d - x_{i,t}^d) + c2 * rand * (P_{l,t}^d - x_{i,t}^d) \quad (51)$$

Donde  $pl$  era la posición óptima en el vecindario local.

## 2.6 Algoritmo Seno Coseno (SCA)

El algoritmo seno coseno (SCA) es un nuevo algoritmo de optimización estocástica que basa sus actualizaciones en las funciones seno y coseno, el algoritmo se basa en la población, inicia con un conjunto aleatorio de agentes de búsqueda de soluciones colocados al azar en el espacio de búsqueda de un problema de optimización (Hafez et al., 2016). El agente de búsqueda es guiado a la solución de puntos óptima en el espacio de búsqueda mediante la función de adecuación que evalúa a cada agente de búsqueda en cada iteración del algoritmo (Hafez et al., 2016).

SCA mantiene un conjunto de  $m$  agentes de búsqueda, y cada agente está representado por un vector de variables de decisión  $n$ -dimensiones  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$  donde  $X_i$  es el  $i$ -ésimo agente de búsqueda en el conjunto. Además, el algoritmo realiza un seguimiento de la

posición  $P$  de la mejor solución lograda por todos los agentes de búsqueda en la población en cada iteración. El modelo matemático utilizado en el algoritmo SCA se basa en la siguiente función de actualización para cualquier agente de búsqueda  $X_i$ : todo esto va del nuevo SCA.

$$X_i^{t+1} = \begin{cases} x_i^t + r1 * \sin(r2) * |r3P_i^t - x_i^t| & \text{Si } r4 < 0.5 \\ x_i^t + r1 * \cos(r2) * |r3P_i^t - x_i^t| & \text{Si } r4 \geq 0.5 \end{cases} \quad (52)$$

Donde  $P_i$  es la posición del punto de destino. El parámetro  $r1$  dicta la región de la siguiente posición (o dirección de movimiento) este podría estar dentro o fuera del espacio de solución, este parámetro se determina por medio de la siguiente ecuación:

$$r1 = \alpha - t \frac{\alpha}{T} \quad (53)$$

Donde  $\alpha$  tiene un valor de 2,  $t$  es el contador y  $T$  es el número máximo de iteraciones y  $a$  es una constante.

El parámetro  $r2$  especifica la distancia recorrida hacia o desde el destino, los valores van entre  $[0, 2\pi]$  garantizando la exploración y explotación del espacio de búsqueda respectivamente. El parámetro  $r3$  proporciona un peso aleatorio para el destino con el fin de enfatizar estocásticamente ( $r3 > 1$ ) o restar importancia ( $r3 < 1$ ) la influencia del destino en la definición de la distancia. Finalmente, el parámetro  $r$  también varía entre los componentes seno y coseno donde es un número aleatorio en  $[0, 1]$ .

Los efectos la ecuación 52 se expresa en la figura 7, la cual muestra cómo la ecuación propuesta define un espacio entre dos soluciones en el espacio de búsqueda. Esta ecuación se puede extender a dimensiones más grandes. El modelo periódico de funciones seno y coseno permite que una solución cambie alrededor de otra. Esto puede garantizar la explotación del espacio definido entre dos soluciones. Para examinar el espacio de búsqueda, las soluciones

también deben poder buscar fuera del espacio entre sus respectivos destinos. Esto se puede lograr variando el rango de las funciones seno y coseno como se muestra en la figura 8.

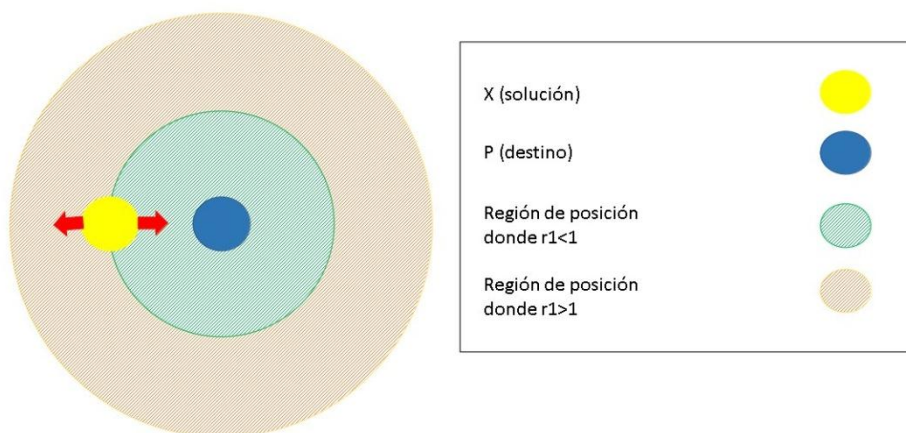


Figura 7. Movimiento de la partícula

fuelle: (Mirjalili, 2016)

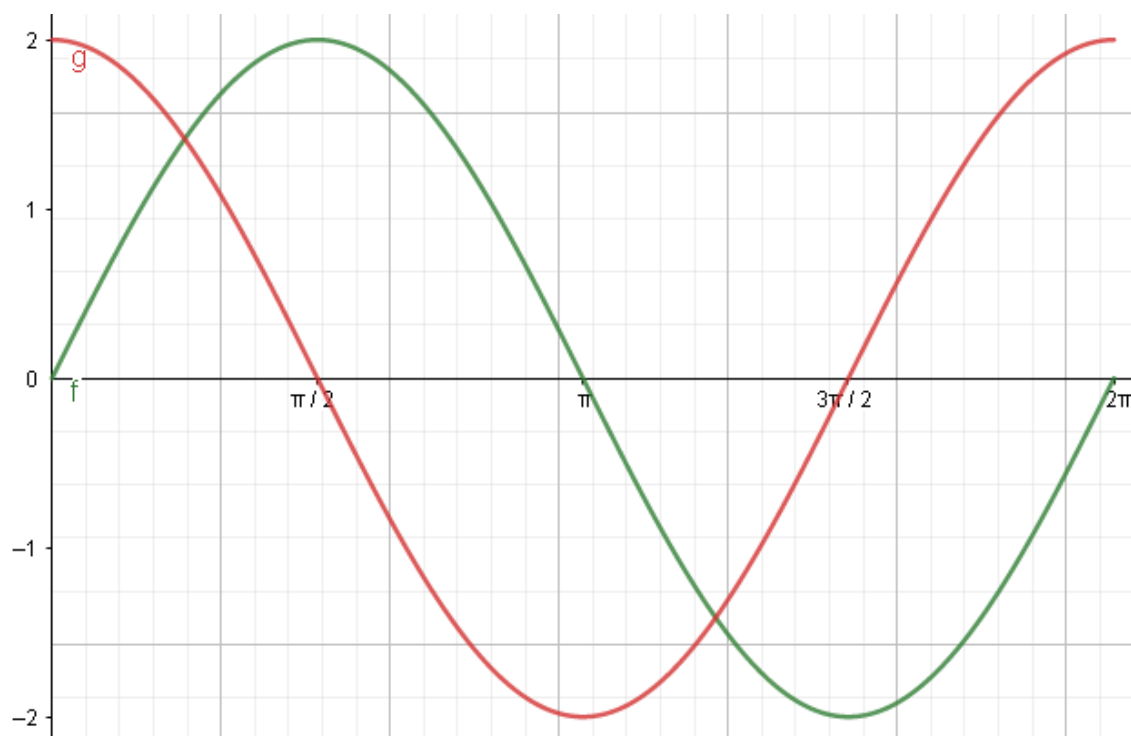


Figura 8. Funciones seno y coseno

fuelle: (Mirjalili, 2016)

### 3. Programación de los Flujos de Carga de un Sistema de Distribución en Matlab

#### 3.1 Elección del Sistema de Prueba con sus Variables y Parámetros

Para la realización de este proyecto se toma como referencia varios artículos, inicialmente la selección del sistema de 33 nodos y los métodos de solución de flujo se carga. Los datos de las líneas y nodos del sistema radial, con sus características de carga y valores nominales (potencia aparente, voltaje base) se visualiza en la tabla 1 y se extraen de (Herrera Briñez, 2021), se trasladan estos datos a los programas Digsilent y Matlab para evaluar y comparar sus resultados de pérdidas de energía y voltajes de los diferentes métodos de flujo de carga, ya que Digsilent trabaja con el método de Newton Raphson y para Matlab se aplica el método de barrido iterativo y aproximaciones sucesivas.

En la figura 9, se muestra la estructura que conforma el sistema de prueba de 33 barras y 32 líneas con una potencia y tensión base de  $1000kVA$  y  $12.66kV$  respectivamente, el nodo Slack se localiza en el nodo 1.

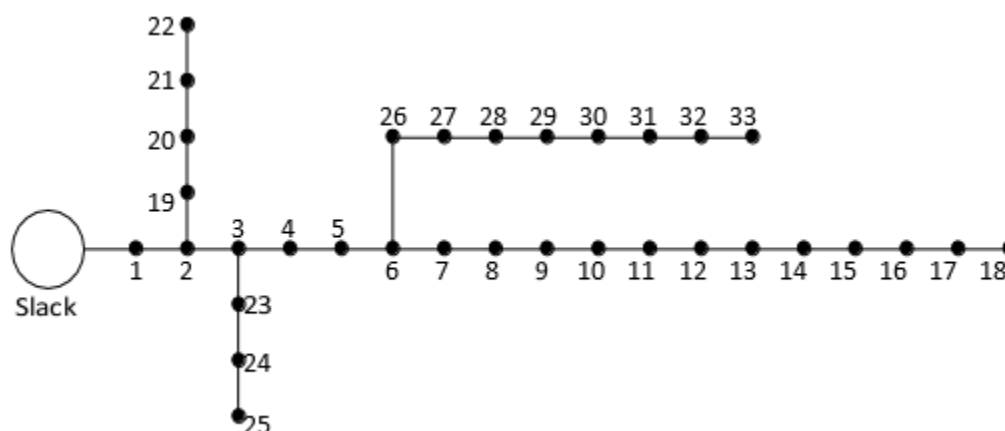


Figura 9. Diagrama de prueba IEEE 33 nodos

fuelle: (Herrera Briñez, 2021)

Tabla 1. Datos de las características de líneas y nodos

<b>Nodo <math>i</math></b>	<b>Nodo <math>j</math></b>	<b><math>R_{ij}</math> [<math>\Omega</math>]</b>	<b><math>X_{ij}</math> [<math>\Omega</math>]</b>	<b><math>P_j</math> [kW]</b>	<b><math>Q_j</math> [kvar]</b>
1	2	0.0922	0.0477	100	60
2	3	0.4930	0.2511	90	40
3	4	0.3660	0.1864	120	80
4	5	0.3811	0.1941	60	30
5	6	0.8190	0.7070	60	20
6	7	0.1872	0.6188	200	100
7	8	1.7114	1.2351	200	100
8	9	1.0300	0.7400	60	20
9	10	1.0400	0.7400	60	20
10	11	0.1966	0.0650	45	30
11	12	0.3744	0.1238	60	35
12	13	1.4680	1.1550	60	35
13	14	0.5416	0.7129	120	80
14	15	0.5910	0.5260	60	10
15	16	0.7463	0.5450	60	20
16	17	1.2890	1.7210	60	20
17	18	0.7320	0.5740	90	40
2	19	0.1640	0.1565	90	40
19	20	1.5042	1.3554	90	40
20	21	0.4095	0.4784	90	40
21	22	0.7089	0.9373	90	40
3	23	0.4512	0.3083	90	50
23	24	0.8980	0.7091	420	200
24	25	0.8960	0.7011	420	200
6	26	0.2030	0.1034	60	25
26	27	0.2842	0.1447	60	25
27	28	1.0590	0.9337	60	20
28	29	0.8042	0.7006	120	70
29	30	0.5075	0.2585	200	600
30	31	0.9744	0.9630	150	70
31	32	0.3105	0.3619	210	100
32	33	0.3410	0.5302	60	40

Fuente: (Herrera Briñez, 2021)

## 3.2 Representación en Programas Especializados

### 3.2.1 Ejecución del flujo de carga en un sistema radial sin GD en Digsilent

Para implementar el diagrama de 33 nodos en el programa Digsilent, lo primero que se hace es configurar los parámetros de inicio del programa, donde la frecuencia de operación sea de 60 Hz.

Al ingresar a la interfaz principal, se elige el nodo Slack en las opciones ubicadas en la parte derecha de la hoja de trabajo del programa como lo indica la siguiente figura.



Figura 10. Selección del nodo Slack

fuentes: (Schmieg, 1985)

Después de seleccionar la opción señalada se configura de la siguiente manera: Al ingresar a la interfaz del Slack, se procede a ponerle un nombre en la opción de Basic Data, después es necesario darle clic a Load Flow y de ahí cambiar la configuración del Bus Type y colocar la opción de (SL). Este es el encargado de controlar el voltaje, así mismo se fija el voltaje de 1 p.u y el ángulo en 0 grados.

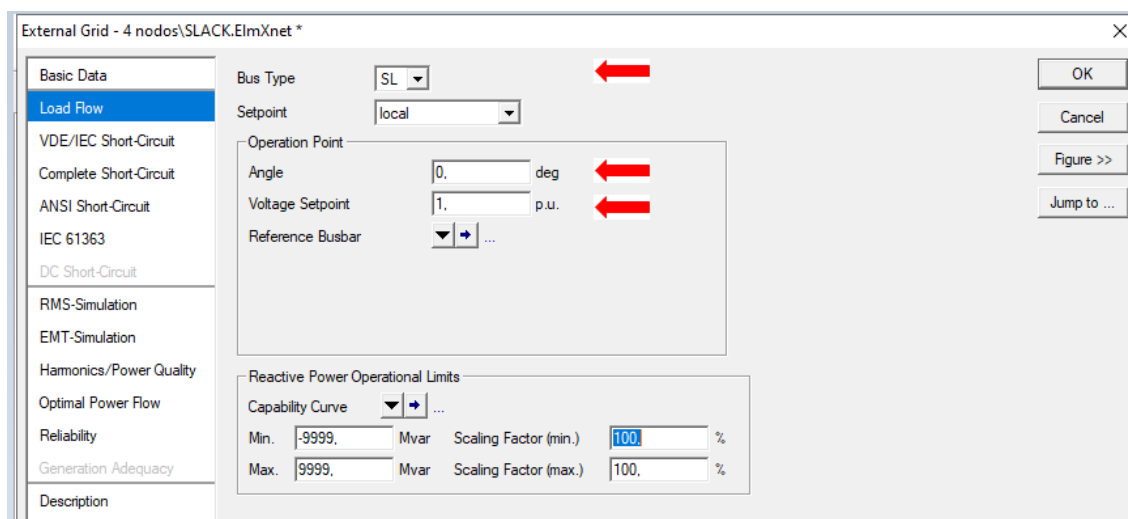


Figura 11. Interfaz del Slack

fuentes: (Schmieg, 1985)

Se procede a configurar los nodos del sistema, es necesario conocer cuantos se utilizan y a su vez saber la configuración de la distribución de estos, ya que es importante seguir el orden en el cual se muestra en la figura 9. Para la selección de nodos, se encuentran en la siguiente figura.



Figura 12. Selección de nodos

fuelle: (Schmiegl, 1985)

Una vez abre la interfaz del nodo se debe indicar el nombre en cada nodo según su secuencia mostrada en la figura 9 y a su vez seleccionar que tipo de sistema es y cuantas fases lo conforman, en este caso es trifásico CA, así mismo se agrega la tensión línea a línea del sistema.



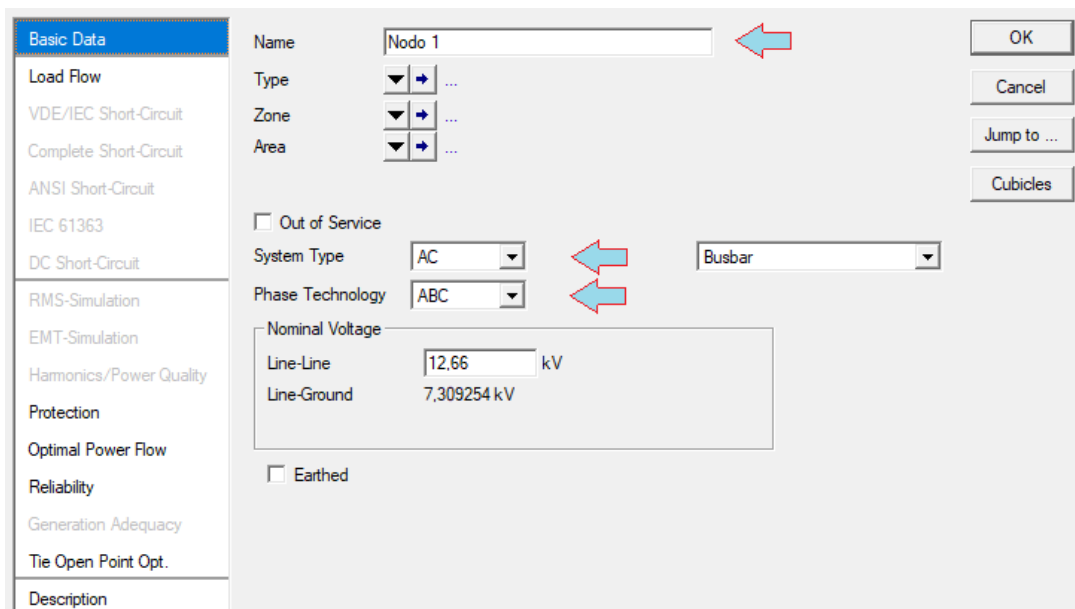


Figura 13. Interfaz de los nodos

fuentes: (Schmieg, 1985)

Una vez configuradas las barras del sistema se dispone a unir las por medio de las líneas en forma radial, la opción se encuentra en la siguiente figura.

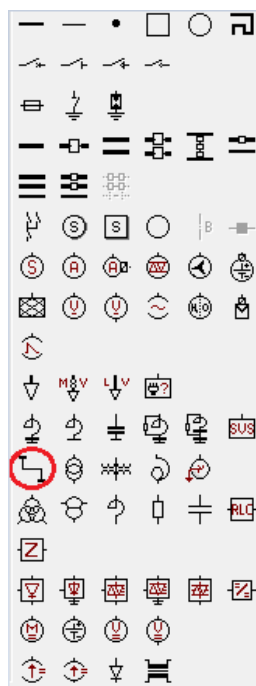


Figura 14. Selección de líneas

fuelle: (Schmieg, 1985)

Para configurar las líneas de la red se asigna un nombre a cada rama, esto depende de los nodos involucrados en la unión directa, en la opción Type se crea la nueva línea para ingresar los datos básicos, es necesario hacer clic primero en New Project Type, una vez allí se oprime la opción Line Type (TypLne).

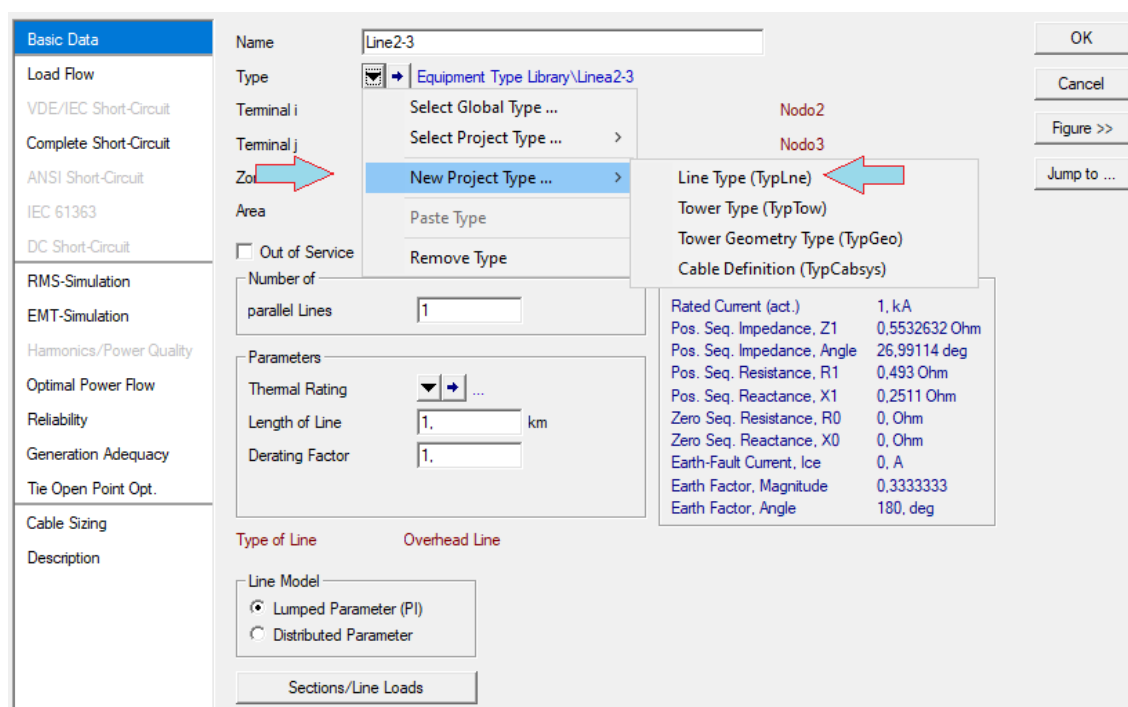


Figura 15. Interfaz de líneas

fuelle: (Schmieg, 1985)

Una vez realizado estos pasos, se configuran los valores básicos de la línea tales como: voltaje operativo, frecuencia, el tipo de sistema (AC) con tres líneas y cero neutros, las respectivas inductancias y reactancias que se encuentran la tabla 1, por último, se configura el tipo de línea (Cable/OHL) el cual se establece que es aéreo (Overhead línea). Debido a que en las condiciones iniciales no se suministra el dato de la corriente, se deja libre con el valor

establecido del programa, además tampoco se cuenta con la información de las distancias de las líneas, por ende, se deja establecida en la configuración inicial del programa que son de 1km.

The image shows a software interface for configuring line parameters. The 'Basic Data' tab is active. The configuration includes:

- Name: Linea2-3
- Rated Voltage: 12.66 kV
- Rated Current: 1 kA
- Nominal Frequency: 60 Hz
- Cable / OHL: Overhead Line
- System Type: AC
- Number of Neutrals: 3

Parameters per Length 1,2-Sequence:

- AC-Resistance R'(20°C): 0.493 Ohm/km
- Reactance X': 0.2511 Ohm/km

Parameters per Length Zero Sequence:

- AC-Resistance R0': 0 Ohm/km
- Reactance X0': 0 Ohm/km

Figura 16. Parámetros dentro de la interfaz de líneas

fuelle: (Schmieg, 1985)

Hasta el momento se observa que aún no se han ingresado los valores de las cargas (P y Q) dentro del diagrama, para ingresar esos valores se selecciona la opción donde indica la siguiente figura.



Figura 17. Selección de cargas

fuelle: (Schmiegl, 1985)

Se selecciona la opción Load Flow en la interfaz de la carga como se observa en la figura 18, allí se procede a ingresar los datos de las cargas P y Q que se requieren en cada nodo que se encuentran en la tabla 1, cabe resaltar que el sistema se encuentra balanceado.

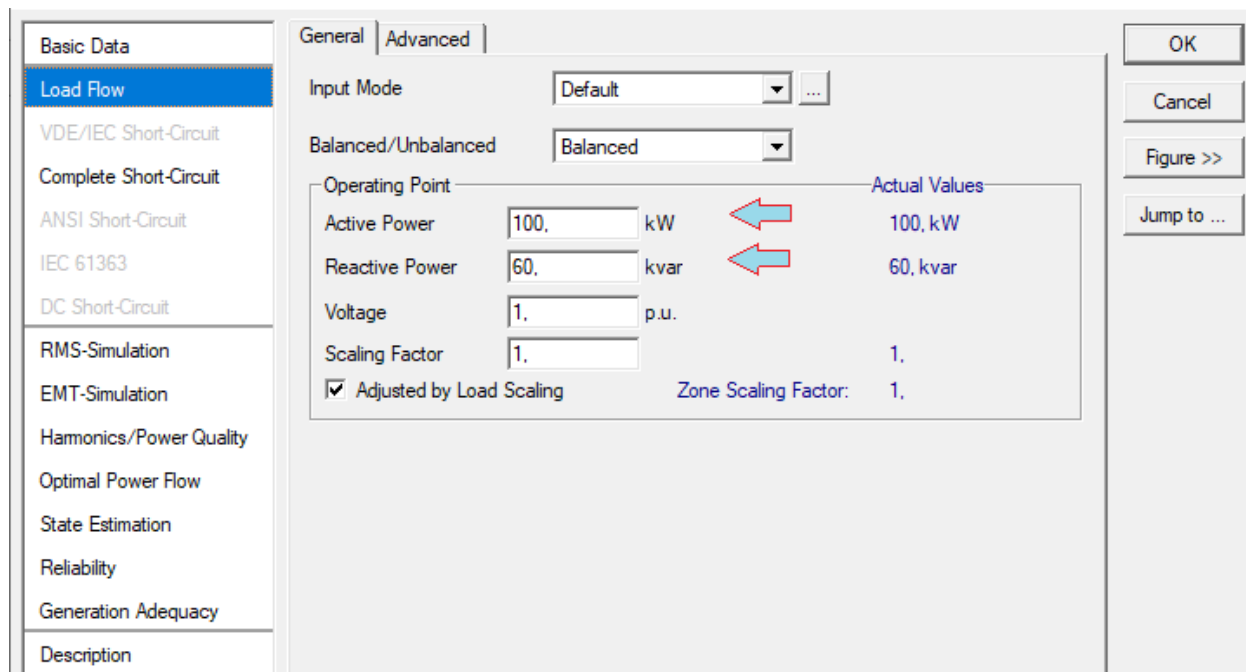


Figura 18. Interfaz de cargas

fuelle: (Schmieg, 1985)

Después de haber colocado los datos del sistema e identificado cada característica, se evaluará el flujo de potencia de la red, para extraer la información de voltajes en cada nodo y las pérdidas de energía del sistema, se necesita antes oprimir la opción calculation, a su vez se busca Optimal Power Flow como muestra la figura a continuación.

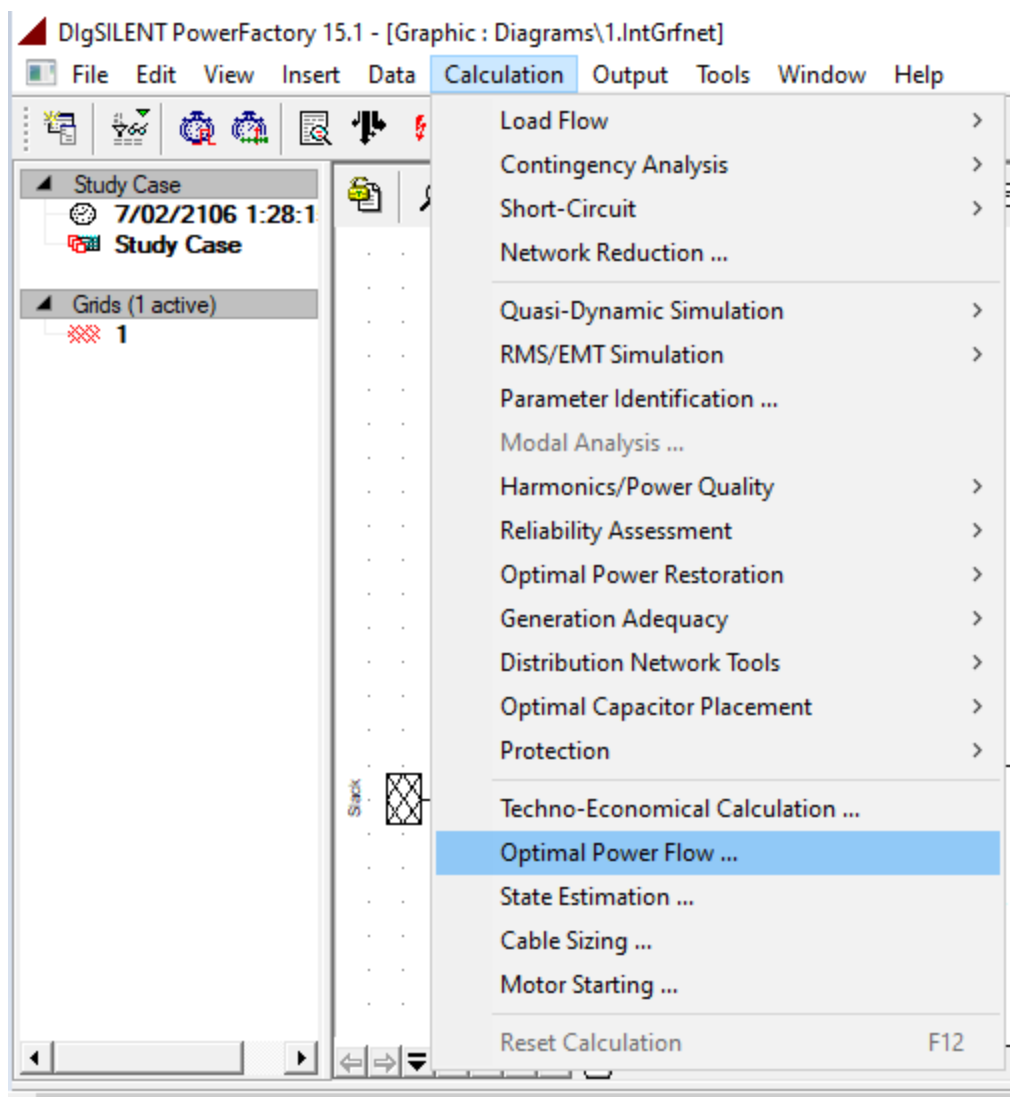


Figura 19. Configuración tipo de calculo

fuate: (Schmieg, 1985)

Aparece una pestaña en la cual elige la opción AC Optimization (Interior Point Method) y se le da Execute, como se muestra en la siguiente figura.

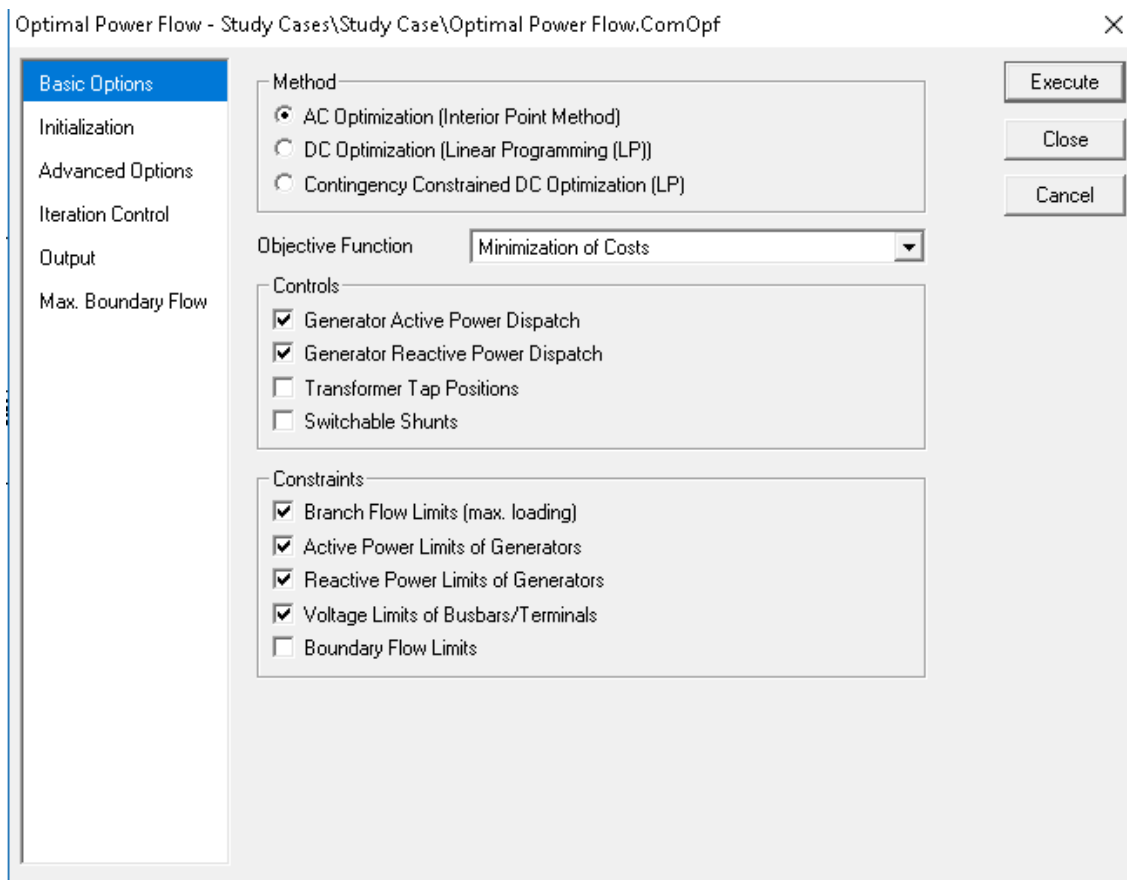


Figura 20. Configuración para ejecutar flujo de carga

fuentes: (Schmiege, 1985)

Se identifican todos los parámetros dentro del sistema de los 33 nodos y se encuentra el resultado de las variables de voltaje, corriente y potencia en cada nodo.

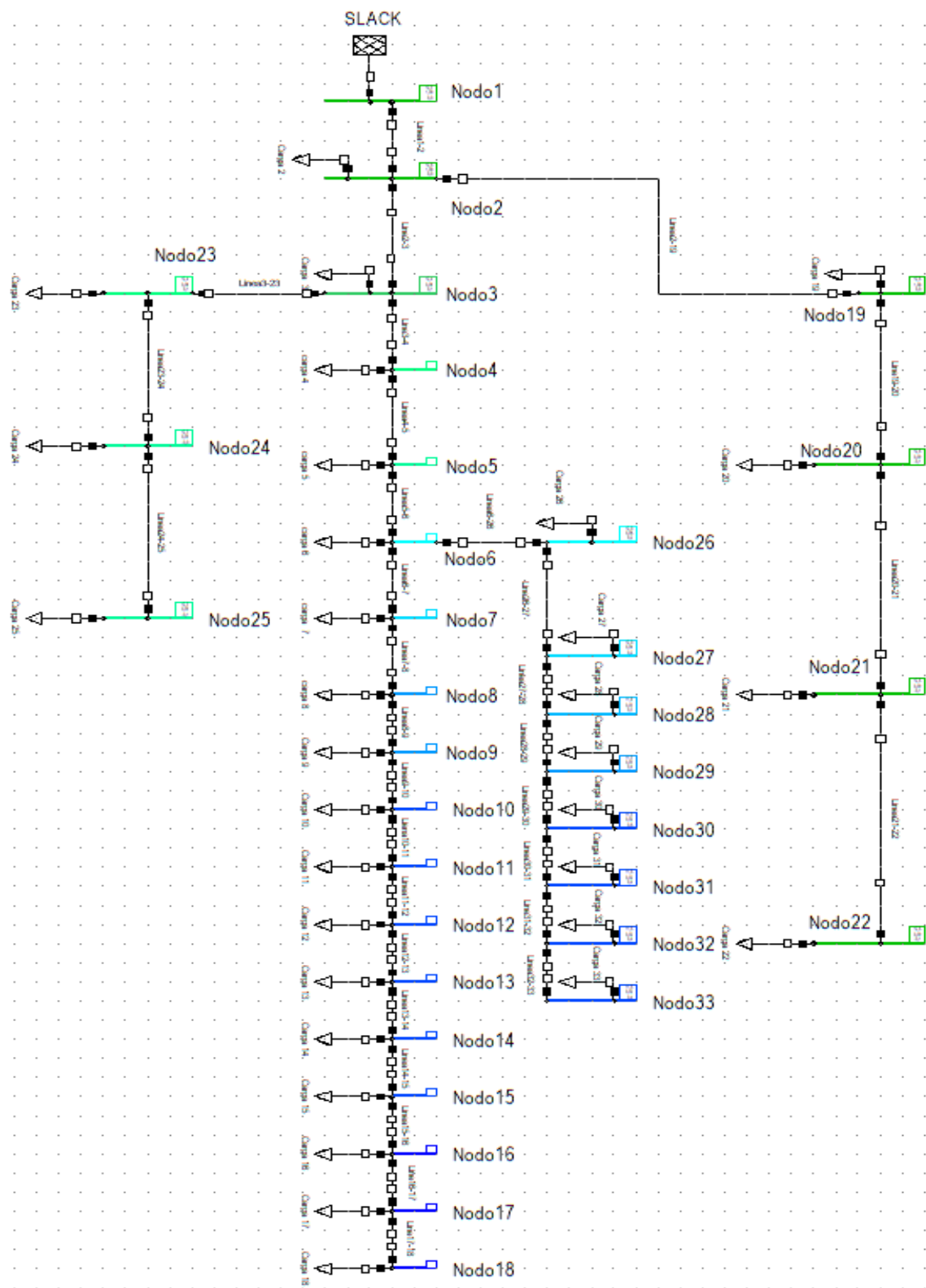


Figura 21. Flujos de carga del diagrama de 33 nodos

fuelle: (Schmiegl, 1985)



Se muestra un reporte detallado del resultado de todas las características del sistema, donde aparecen los valores de voltaje en *p.u.*, sus voltajes nominales con sus respectivos ángulos.

Grid: 1		System Stage: 1				Study Case: Study Case		Annex: / 1		
	rtd.V [kV]	Bus - [p.u.]	voltage [kV]	[deg]	-10	-5	Voltage - 0	Deviation [%]	+5	+10
Nodo2	12,66	0,997	12,62	0,01						
Nodo1	12,66	1,000	12,66	0,00						
Nodo3	12,66	0,983	12,44	0,10						
Nodo4	12,66	0,975	12,35	0,16						
Nodo5	12,66	0,968	12,25	0,23						
Nodo6	12,66	0,949	12,02	0,13						
Nodo7	12,66	0,946	11,98	-0,10						
Nodo8	12,66	0,932	11,80	-0,25						
Nodo9	12,66	0,926	11,72	-0,32						
Nodo10	12,66	0,920	11,65	-0,39						
Nodo11	12,66	0,919	11,64	-0,38						
Nodo12	12,66	0,918	11,62	-0,37						
Nodo13	12,66	0,912	11,54	-0,46						
Nodo14	12,66	0,909	11,51	-0,54						
Nodo15										

Figura 22. Reporte de resultados

fuelle: (Schmieg, 1985)

Por último se muestra las pérdidas totales del sistema, tanto activas como reactivas y se comprueba que son los mismos valores mostrados en el artículo (Herrera Briñez, 2021).

Total Losses [kW]/ [kvar]
210,99 143,13

Figura 23. Pérdidas de energía

Una vez realizado estos pasos, se toma la información sobre la ubicación de los generadores distribuidos en el artículo (Grisales-Noreña et al., 2018a).

### 3.2.2 *Desarrollo de la Programación del Flujo de Carga en Matlab sin GD*

#### 3.2.2.1 Programación del Método de Barrido Iterativo

Para la realización de estos dos métodos se tiene en cuenta:

- **Paso 1**

Identificar parámetros en sus estados de tensiones nominales de alimentación, se considera que el circuito se encuentra balanceado en la red radial de 33 nodos.

- **Paso 2**

Se procede a incluir la información de cada línea y nodo de la tabla 1, en una matriz de nodo que contenga las variables: nodo de recibo ( $j$ ), nodo de envío ( $i$ ), resistencia ( $R$ ), reactancia ( $X$ ) en cada columna con una dimensión de  $33 \times 4$ , de igual forma se crea una matriz de línea con las variables:  $i$  (número de nodos de la matriz), tipo (representa el tipo de nodo, 0 para el Slack y 1 para las demandas),  $V_0$  (voltaje de todos los nodos de demanda),  $P$  (potencia activa),  $Q$  (potencia reactiva) en cada columna de la matriz con dimensión  $32 \times 5$ .

- **Paso 3**

Se convierten los valores reales a por unidad, el cual es la representación adimensional de todas las variables del sistema eléctrico de potencia (SEP), la finalidad de esta conversión es representar el sistema real, a un sistema en porcentaje y se realiza por medio de la fórmula de la impedancia base ( $Z_{base}$ ), es igual a la división entre el voltaje base ( $V_{base}$ ) al cuadrado y la potencia aparente base ( $S_{base}$ ), luego se extraen los valores de las columnas 3 y 4 de la matriz

de línea y se dividen por  $Zbase$ , de igual forma se utilizan los datos de las columnas 4 y 5 de la matriz de nodos y se dividen por  $Sbase$ .

$$Zbase = Vbase^2 / Sbase \quad (54)$$

```

70      %% Cambio a por Unidad
71 -    Zbase= ((1000*Vbase)^2)/(1000*Sbase);
72 -    Lineas(:,3:4)=Lineas(:,3:4)/Zbase;
73 -    Nodos(:,4:5)=Nodos(:,4:5)/Sbase;

```

Figura 24. Cambio a por unidad

- **Paso 4**

Se busca cuantas filas tienen las matrices de nodos y líneas, después verificar que dimensiones tiene la matriz A (matriz de incidencia), así mismo se ocupa la matriz A por medio de un ciclo *for* que se ejecuta con el contador *i* hasta el número de líneas en el código, dependiendo de la corriente en los nodos de envío y de recibo dentro del diagrama de la figura 9 de la siguiente manera:

$$A_{ij} = \begin{cases} "1" & \text{si la rama } j \text{ esta conectada al nodo } i \text{ y la corriente sale del nodo } i \\ "-1" & \text{si la rama } j \text{ esta conectada al nodo } i \text{ y la corriente llega al nodo } i \\ "0" & \text{si la rama } j \text{ no esta conectada al nodo } i \end{cases}$$

```
74  %% Formación Matriz de Incidencia (Matriz A)
75 - NN=size(Nodos,1);
76 - NL=size(Lineas,1);
77 - A=zeros(NN,NL);
78 - Zr=zeros(NL,NL);
79 - for i=1:NL
80 -     Ni=Lineas(i,1); Nj=Lineas(i,2);
81 -     A(Ni,i)=1;
82 -     A(Nj,i)=-1;
83 -     Zr(i,i)=Lineas(i,3)+1j*Lineas(i,4);
84 - end
```

Figura 25. Código para crear la matriz de incidencia

Se extraen las impedancias de las columnas tres y cuatro de la matriz de nodos y se incluye en el ciclo *for*. En la figura 26, se presenta el diagrama de flujo donde se describe el proceso para el desarrollo de la creación de la matriz de incidencia A y la matriz primitiva de las impedancias Zr.

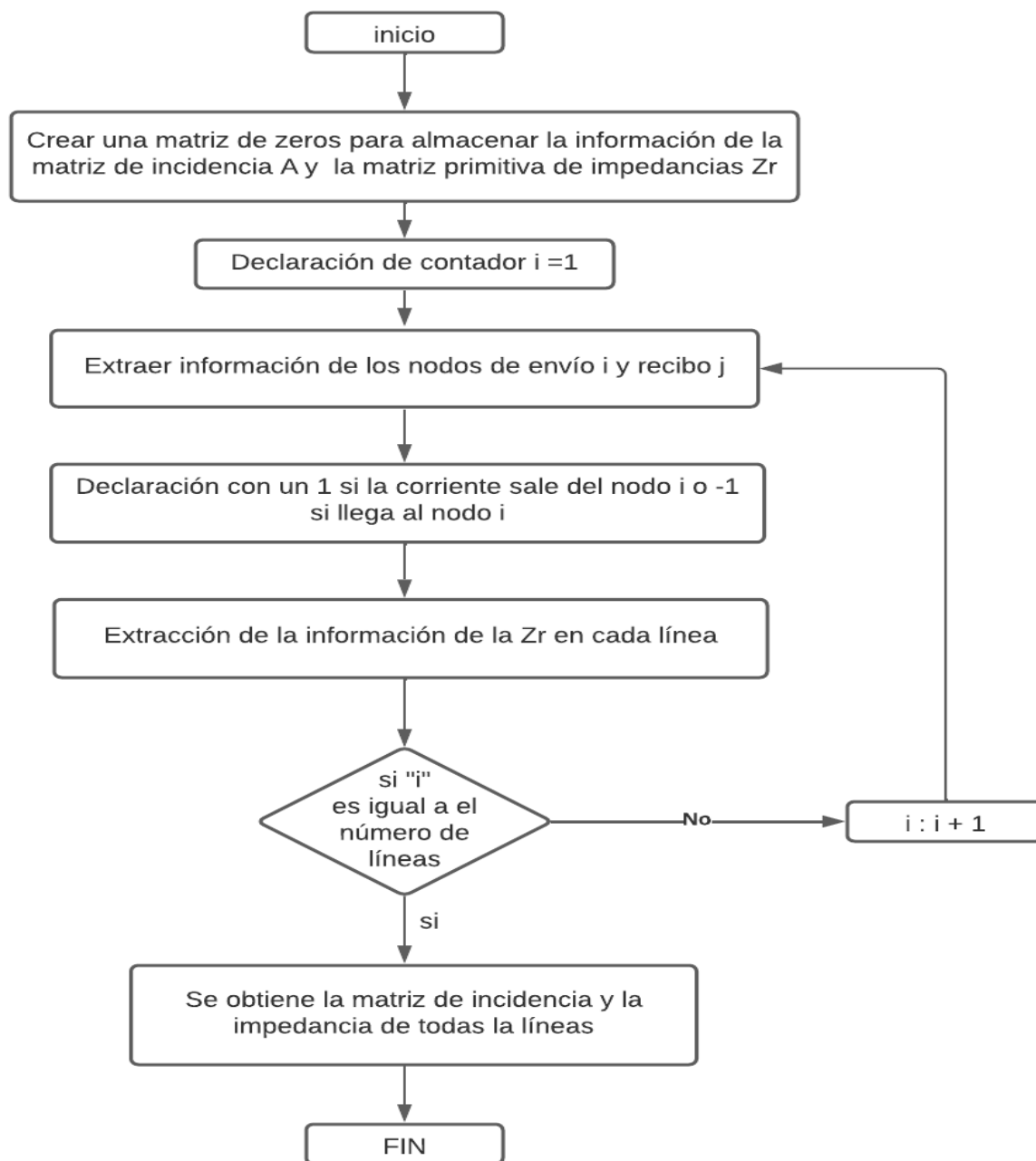


Figura 26. Diagrama de flujo para la creación de A y Zr

- Paso 5

Se extrae la variable  $Ao$  de la matriz de incidencia, que es equivalente a la fila 1 de dicha matriz, así mismo se presenta  $Ad$  que es el restante de la matriz de incidencia de todas las filas y columnas exceptuando la fila 1, obteniendo estas variables se aplica la fórmula para hallar  $Ydd$ .

$$Ydd = Ad * (inv (Zr)) * Ad.', \quad (55)$$

Se agregan los voltajes de las barras de demanda, ubicados en la columna 3 de la matriz de nodos y la potencia de demandada activa y reactiva. Se define una tolerancia de  $1 * 10^{-6}$  con interacciones máximas de 100, para la etapa final del código se aplica la siguiente formula:

$$Vd = -IYdd * ((inv (diag (conj (Vdo)))) * conj (Sd) + Ydg * Vg) \quad (56)$$

Se obtiene el error más grande en las interacciones por medio de los valores absolutos, debido a que se trabaja con números complejos y se necesita sacar la magnitud de  $Vd - Vdo$ , cuyo resultado tiene que ser menor a la tolerancia.

Se introduce en valor absoluto  $Vd$  y  $Vdo$ , porque  $Vd$  genera la magnitud de la iteración actual y  $Vdo$  la magnitud de la iteración anterior, se pone un valor absoluto entre la resta de las dos para eliminar cualquier resultado negativo, como esto es un vector se indica en el código el error máximo, para compararlo con la tolerancia y es así como finaliza el proceso, sino siguen las iteraciones hasta que se cumpla esta condición.

```

%% Barrido iterativo
tol=1e-6;
Ao=A(1,:);
Ad=A(2:end,:);
Ydd=Ad*(inv(Zr))*Ad.';
IYdd=inv(Ydd);
Ydg=Ad*(inv(Zr))*Ao.';
Vdo=Nodos(2:end,3);
Vo=Nodos(1,3);
Sd=Nodos(2:end,4)+1j*Nodos(2:end,5);
itermax=100;
for t=1:itermax
    Vd=-IYdd*(inv(diag(conj(Vdo))))*conj(Sd)+Ydg*Vo;
    if max(abs((abs(Vd)-abs(Vdo))))<=tol
        V=[Vo;Vd];
        break
    else
        Vdo=Vd;
    end
end
end

```

Figura 27. Programación del método barrido iterativo

- **Paso 6**

Finalmente se hayan las pérdidas del sistema, para realizar esto primero se aplican las fórmulas de las tensiones y corrientes en las ramas que se expresan a continuación:

$$Vr = Ao.' * Vo + Ad.' * Vd \quad (57)$$

$$Ir = (inv(Zr)) * Vr \quad (58)$$

Una vez hallado estos valores, se procede a incluir la fórmula de pérdidas de energía basada en la teoría, se agrega *sum* para obtener la pérdida total del sistema, de lo contrario se muestran las pérdidas de cada línea, este método no depende de la *Ybus* para el cálculo de las pérdidas. Al hallar las pérdidas se debe tener cuidado con las dimensiones del resultado, ya que *Zr* tiene una dimensión de  $Nr \times Nr$  y el vector *Ir* es un vector de  $Nr \times 1$ , como los vectores no se pueden elevar al cuadrado, se eleva cada elemento del vector agregando  $.^2$  en el código, como

el resultado del producto de  $(Zr * (abs(Ir)).^2)$  es un vector de  $Nrx1$ , por último se suma cada componente con la función *sum*.

```
106 %%Pérdidas Sloss %%  
107 - Vr=Ao.'*Vo+Ad.'*Vd;  
108 - Ir=(inv(Zr))*Vr;  
109 - Sloss=(sum(Zr*(abs(Ir)).^2))*Sbase;
```

Figura 28. Pérdidas Sloss

En la figura 29, se presenta el diagrama de flujo donde se describe el proceso para el desarrollo del método barrido iterativo.



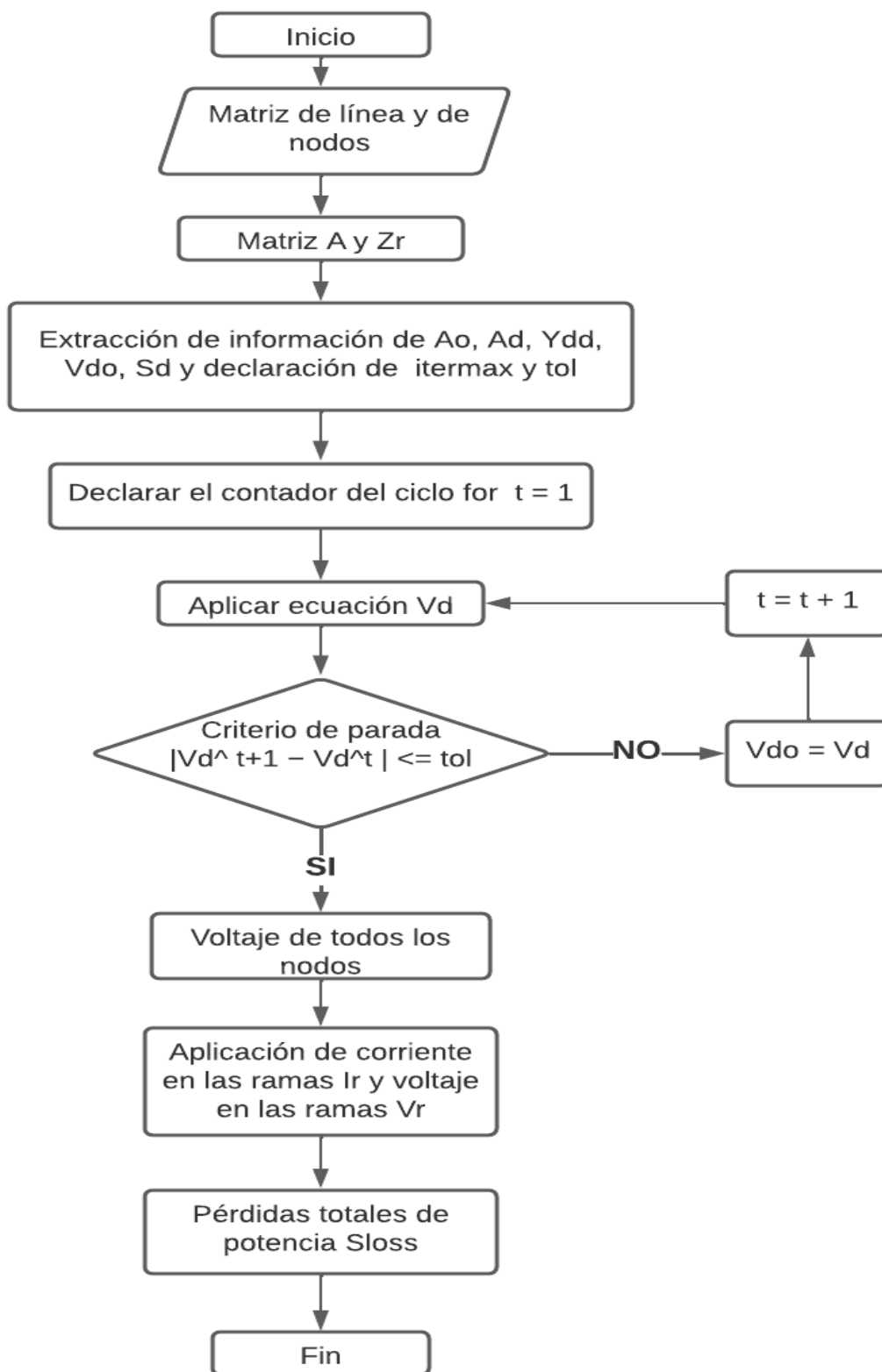


Figura 29. Diagrama de flujo del método barrido iterativo

### 3.2.2.2 Programación del Método de Aproximaciones Sucesivas

Para la realización del proceso de programación de este código, se deben seguir los primeros tres pasos del método de barrido iterativo, ya que estos no tienen ningún cambio al comparar ambos métodos. Después de tener esta información, se muestra continuación los pasos a seguir:

- **Paso 1**

Para encontrar la  $Y_{bus}$  se tiene que contar, cuántas filas tiene la matriz de nodos ( $NN$ ) y cuántas filas tiene la matriz de líneas ( $NL$ ), se define el tamaño de la matriz  $Y_{bus}$  rellenándola inicialmente de ceros, indicando que es una matriz de  $NN \times NN$ , esto depende del número de nodos que tenga el sistema. La matriz  $Y_{bus}$  se hace por medio de un ciclo *for* con un contador  $i$ , desde la línea 1 hasta el final de las líneas, se determinan los elementos de la matriz de línea extrayendo la información de los nodos de envío y recibo, ubicados en las columnas 1 y 2 de toda la matriz, igualmente se extrae la  $ZL$  su parte real ( $R$ ) e imaginaria ( $X$ ) ubicadas en las columnas 3 y 4, aplicando el concepto de la matriz  $Y_{bus}$  se realiza en los nodos de envío ( $i$ ) y de recibo ( $j$ ) del sistema, una vez realizado este procedimiento se invierten las posiciones de dichos nodos, debido a que la matriz es simétrica y se procede a crear la diagonal de la matriz de la siguiente manera:

$$Yb(Ni, Ni) = Yb(Ni, Ni) + 1/ZL \quad (59)$$

$$Yb(Nj, Nj) = Yb(Nj, Nj) + 1/ZL \quad (60)$$

Lo que se acaba de realizar es la creación  $Y_{bus}$  cuando no tienen elementos o líneas acopladas a tierra con efecto capacitivo, si se llega a tener un capacitor en el sistema se le agrega una nueva columna a la matriz de línea y se corre normalmente.

```

79      %Formacion de la Ybus
80 -    NN = size(Nodos,1);
81 -    NL = size(Lineas,1);
82
83      %tamaño de la matriz ybus
84 -    Yb = zeros(MN,NN);
85
86 -    for i=1:NL
87 -        Ni = Lineas(i,1); Nj = Lineas(i,2);
88 -        ZL = Lineas(i,3) + 1j*Lineas(i,4);
89
90          %ecuacion de Ybus
91 -        Yb(Ni,Nj) = -1/ZL; Yb(Nj,Ni) = -1/ZL;
92          % Diagonal
93 -        Yb(Ni,Ni) = Yb(Ni,Ni) + 1/ZL;
94 -        Yb(Nj,Nj) = Yb(Nj,Nj) + 1/ZL;
95 -    end

```

Figura 30. Formación de la  $Y_{bus}$

- **Paso 2**

Se aplica una tolerancia de  $1 \times 10^{-6}$  e incluyen los valores iniciales de los voltajes que están representados por la variable  $V0$  y  $Vg$  que es el voltaje en el nodo Slack, estos se encuentran ubicados en la columna 3 de la matriz de nodos, se representa en el código las demandas  $Sd$  ubicadas en las columnas 4 y 5, donde se tiene una parte real y una imaginaria, se extrae  $Ydd$  de la matriz de admitancia  $Ybus$ , se encuentra ubicada en la fila 2 hasta el final desde las filas y en columna 2 hasta el final de las columnas, así mismo se representa  $Dg$  que es el vector desde la fila 2 hasta el final de la columna 1, se declara una variable llamada  $IYdd$  que es la inversa de  $Ydd$ , antes de aplicar la fórmula de aproximaciones sucesivas se agrega una interacción máxima de 100 y se inicia un ciclo *for* de  $t = 1$  hasta las iteraciones máximas. Una vez realizado estos pasos se incluye la fórmula general del voltaje que se presenta a continuación.

$$V = -IYdd * ((inv(diag(conj(V0)))) * conj(Sd) + Ydg * Vg) \quad (61)$$

Para asegurar el resultado del cálculo en el código, se tiene que cumplir que la diferencia de los voltajes absolutos entre  $V$  y  $V_0$  sea menor a la tolerancia, dando como resultado los voltajes del sistema dejando en la primera posición el voltaje del Slack y en la segunda posición el voltaje de los nodos de demanda, terminando así el proceso iterativo, si no cumple esta condición se sigue iterando y el nuevo voltaje inicial  $V_0$ , es el voltaje en el ciclo calculado  $V$  actualizado. Para conocer las pérdidas se aplica la fórmula  $S_{loss}$ . Una vez implementado el código las pérdidas de energía son de  $210,987kW$ .

```

95  %%Método de aproximaciones sucesivas
96  tol= 1e-6;
97  V0 = Nodos(2:end,3); Vg = Nodos(1,3);
98  Sd = Nodos(2:end,4) + 1j*Nodos(2:end,5);
99  Ydd= Yb(2:end,2:end) ; Ydg=Yb(2:end,1);
100 intermax = 100;
101 IYdd = inv(Ydd);
102 for t=1:intermax
103     V = -IYdd*(inv(diag(conj(V0))))*conj(Sd) + Ydg*Vg ;
104     if max(abs(abs(V)-abs(V0)))<tol
105         Vr= [Vg;V];
106         break
107     else
108         V0 = V;
109     end
110 end
111 Sloss = sum(Vr.'*(conj(Yb)*conj(Vr)))*Sbase;

```

Figura 31. Programación del método de aproximaciones sucesivas

En la figura 32, se presenta el diagrama de flujo donde se describe el proceso de los ciclos de la programación del método aproximaciones sucesivas.

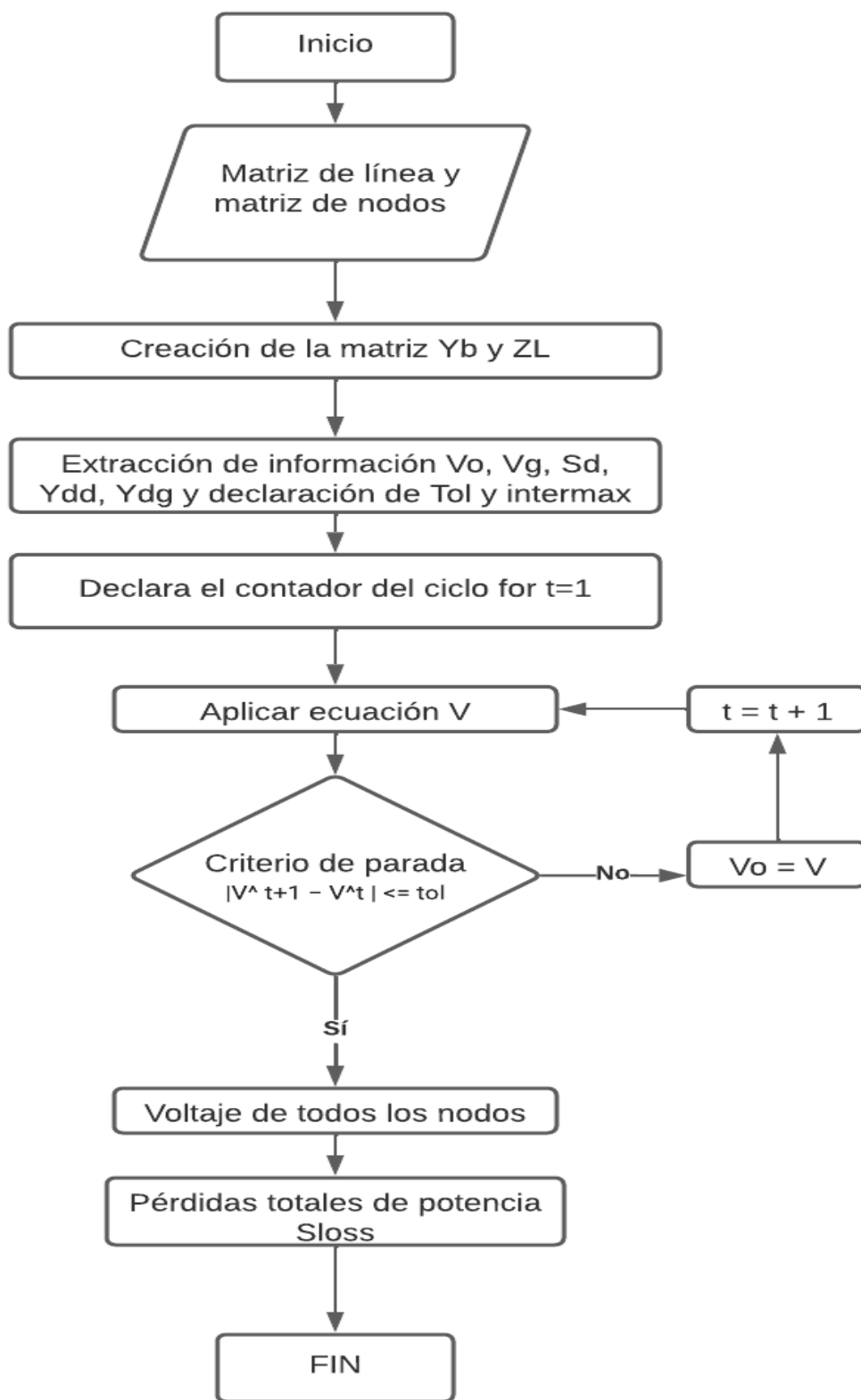


Figura 32. Diagrama de flujo para el método aproximaciones sucesivas

### 3.2.3 Programación de los Flujos de Carga en Matlab 24 Horas con y sin GD

#### 3.2.3.1 Programación del Método Aproximaciones Sucesivas 24 Horas sin GD

Partiendo de la tabla 1 del sistema de 33 nodos, se observa que las dos últimas columnas son las potencias máximas de demanda del sistema, en el código de aproximaciones sucesivas, sus resultados son los voltajes y las pérdidas de energía totales al 100% corriendo un flujo de carga, como los sistemas eléctricos tienen un comportamiento hora a hora que varían se hace necesario la modificación de este.

Tabla 2. Datos de radiación y demanda

Fuente: (Historicos de Demanda XM, n.d.) : “la información base para el presente trabajo fue tomada de xm compañía de expertos en mercados s.a. e.s.p., bajo su autorización y no constituye información reservada de sus clientes o de la empresa.”

Fuente: Semillero de investigación Gestión de Energía y Mantenimiento en el Instituto Tecnológico Metropolitano de Medellín.

Hora	Demanda en Colombia	Radiación Medellín
1	0.3188405797	0
2	0.231884058	0
3	0.2173913043	0
4	0.1739130435	0
5	0.1884057971	0
6	0.2463768116	0
7	0.3188405797	5.46E-05
8	0.4637681159	0.05645695398
9	0.6666666667	0.2643001339
10	0.7826086957	0.5400050784
11	0.884057971	0.7632408035
12	0.9420289855	0.9351836054
13	0.9855072464	0.9528991203
14	0.8985507246	0.954615842

15	0.9130434783	0.8339141554
16	0.9275362319	0.6934887906
17	0.9275362319	0.3888506077
18	0.9275362319	0.1810858073
19	0.884057971	0.02447819646
20	1	0
21	1	0
22	0.8985507246	0
23	0.7391304348	0
24	0.5652173913	0

Según las figuras 33 y 34, se observa el comportamiento de la demanda en Colombia y la radiación en Medellín, de esta manera se puede hacer un análisis más profundo de las potencias inyectadas hora a hora, como nuestros generadores tiene un máximo de potencia de  $1.2MW$ , para saber cuánto es posible inyectar potencia al sistema, por medio de los paneles solares hora a hora, se multiplica la potencia máxima del generador por el porcentaje de radiación hora a hora, a medida que la generación solar aumenta, el generador va permitir inyectar más potencia.

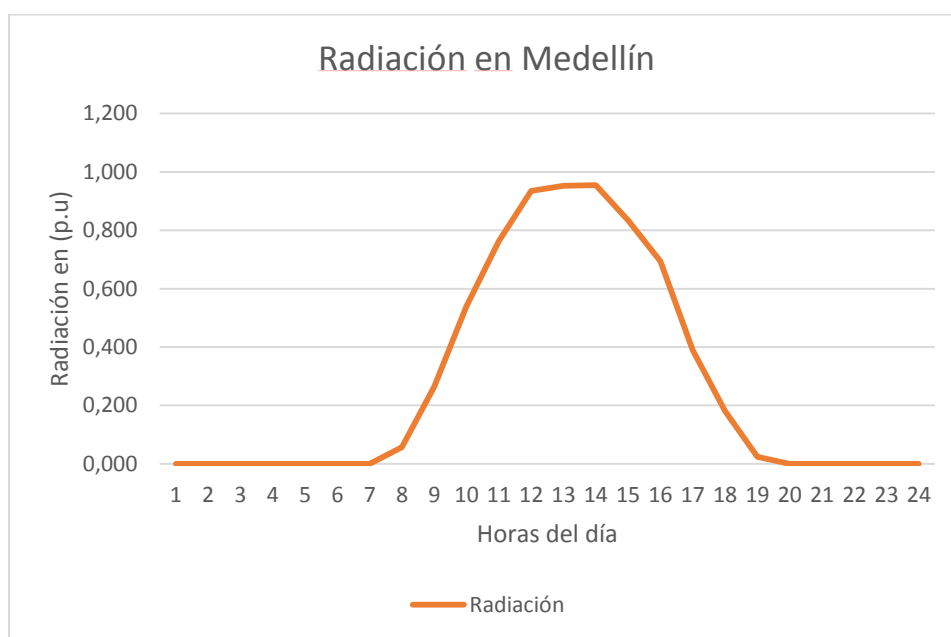


Figura 33. Radiación en Medellín

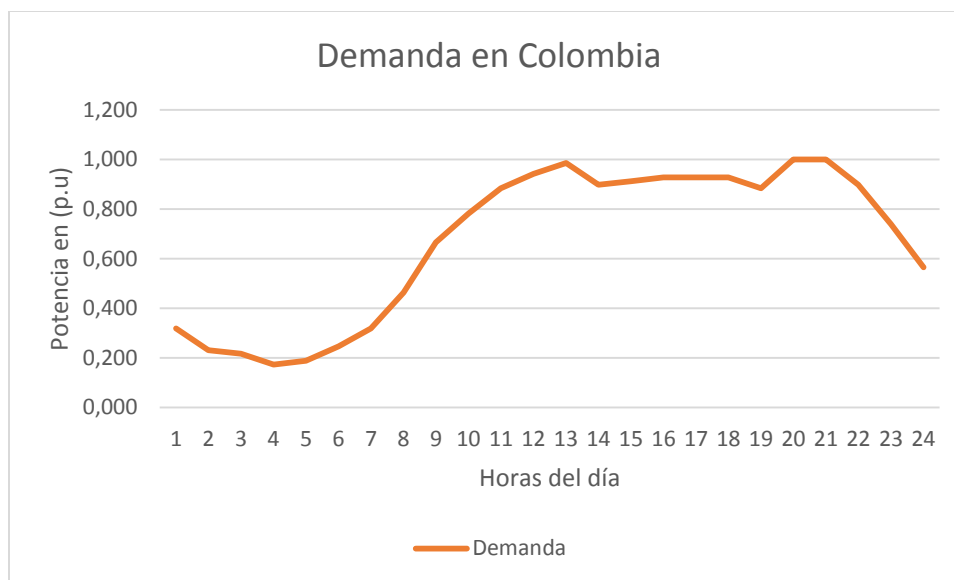


Figura 34. Demanda en Colombia

Se muestra el escenario de máxima potencia por cada generador en la tabla 3, donde se observa cuanta debe ser la potencia inyectada por cada generador en su respectivo nodo, se observa que solo existe un efecto en la red en el rango de 7am a 7pm, generalmente existe un máximo de penetración de generación distribuida que se puede inyectar al sistema.

Tabla 3. Datos de potencia inyectada por los generadores

Hora	G1	G2	G3
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	0	0	0
6	0	0	0
7	0.06549708	0.06549708	0.06549708
8	67.7483448	67.7483448	67.7483448
9	317.160161	317.160161	317.160161
10	648.006094	648.006094	648.006094
11	915.888964	915.888964	915.888964



12	1122.22033	1122.22033	1122.22033
13	1143.47894	1143.47894	1143.47894
14	1145.53901	1145.53901	1145.53901
15	1000.69699	1000.69699	1000.69699
16	832.186549	832.186549	832.186549
17	466.620729	466.620729	466.620729
18	217.302969	217.302969	217.302969
19	29.3738357	29.3738357	29.3738357
20	0	0	0
21	0	0	0
22	0	0	0
23	0	0	0
24	0	0	0

Según la literatura se analiza el efecto de las 24 horas del día y se procede a modificar el código para analizar el comportamiento en cada generador dentro del sistema y de esta manera tomar la decisión más adecuada.

Partiendo del código de aproximaciones sucesivas para lograr correr el flujo de 24 horas, primero se agrega un *for* y se anexa una variable *iii* que permita recorrer de 1 a 24 ciclos, con la finalidad de correr el efecto en la variación en la demanda hora a hora. Seguido al *for* se declara  $factor\_de\_demanda = parametros(iii, 1)$ , se cargan los datos de demanda y radiación en la matriz  $parametros(iii, 1)$ , esto quiere decir que en la hora 1 se obtiene un factor de demanda y calculara un factor cada vez que el *for* genere un ciclo, hasta que finalice los 24 flujos. Los parámetros que se encuentran en la tabla 2, corriendo el primer ciclo del código se observa que en la hora 1 la carga opera al 31,88% en todos los nodos del sistema, entonces se procede a realizar un ajuste en el código, en la matriz de sistema y matriz *node* que contienen la potencia activa y reactiva de la red, a partir de este ajuste se permite calcular la potencia que debe inyectar el generador en cada ciclo como se muestra en la tabla 3, se multiplica el factor de

demanda en Colombia por la carga en todos los nodos de sistema como se presenta en la figura 35.

```

47      %ajuste de demanda
48      %potencia activa
49 -     Sistema(:,5)=factor_de_demanda*Sistema(:,5);
50      %potencia reactiva
51 -     Sistema(:,6)=factor_de_demanda*Sistema(:,6);
87      %ajuste de demanda
88      %potencia activa
89 -     Node(:,4)=factor_de_demanda* Node(:,4);
90      %potencia reactiva
91 -     Node(:,5)=factor_de_demanda* Node(:,5);

```

Figura 35. Ajuste de demanda de la matriz sistema y Node

De esta manera se puede conocer cómo se comportan los usuarios, en porcentajes las 24 horas del día y analizar profundamente las horas en que cae la demanda y también sus horas pico.

Tomando como referencia la programación del método de aproximaciones sucesivas, se procede a incluir las 24 horas del día, con el fin de conocer el flujo de carga que transcurre cada hora, para hallar la pérdida de potencia total y gracias a esta aplicación se lleva al sistema a tener menos saturación en la curva de demanda.

Corriendo el código se observa que las pérdidas en el sistema en la hora 1 disminuyó de 210 *kW* a ser de 19, 33 *kW* y la magnitud de los voltajes se extrae por medio de  $voltajes\_horario = abs(V)$ , el voltaje tiene que estar entre el 0.95 y el 1.05 de la tensión nominal en *p.u.*, se analiza que en la hora uno como baja la demanda y mejoran las tensiones. Se observó que en las horas 20 y 21 es donde más hay demanda en el sistema y se confirma las pérdidas de 210*kW* en esas horas, esto coincide con la gráfica de la demanda en Colombia.

- **Cálculo de Pérdidas Horarias.**

Para hallar las pérdidas del sistema en un día de operación sin generación distribuida lo único que se debe hacer es sumar las pérdidas hora a hora.

Corriendo el código se obtiene las pérdidas totales del sistema en un día y son de  $2691.988479 \text{ kW}$ , se puede observar que las pérdidas aumentan por lo que es la sumatoria de las pérdidas hora a hora en un día, mientras que en el código de aproximaciones sucesivas de un solo flujo eran de  $210 \text{ kW}$ . Se procede a analizar los voltajes para encontrar una forma en la cual el sistema no supere los límites de operación, se tiene en cuenta los límites de corriente, voltaje y el balance de potencia que son las restricciones del sistema.

Se analiza que el balance de potencia cumpla con la restricción, por medio de la ecuación de balance de potencia  $pg - pd - \text{pérdidas} = 0$ , esta formula se cumple debido a que el nodo Slack inyecta al sistema solo lo necesario para garantizar este balance. Se implementa una manera de hallar los límites máximos de voltajes y corrientes hora a hora, esto se hace con el fin de garantizar que se cumplan las restricciones dadas previamente ya que al analizar los valores más altos de corriente y voltaje se garantiza que tanto este como los demás datos estén en las condiciones deseadas.

- **Cálculo voltajes sin GD**

Se anexa la condición para extraer los datos de las tensiones y se analiza el resultado teniendo en cuenta, que la mínima tensión va a ser la que esté más lejos de 1, esto se muestra en la figura 36.

```

121 | %cálculo de tensiones %%
122 - | voltajes_horario=abs(V);
123 - | errores_de_voltaje= abs(voltajes_horario-1);
124 - | peor_error=max(errores_de_voltaje);
125 - | pos=find(errores_de_voltaje==peor_error);
126 - | peor_tension(iii,1)=voltajes_horario(pos,1);

```

Figura 36. Cálculo de tensiones

Los errores de voltaje en la línea 123 se hacen por medio de una resta de los voltajes horarios menos uno, el *peor\_error* es el error máximo en la variable *errores\_de\_voltaje*, se crea una variable *pos* y se declara la función *find* que permita encontrar en las posiciones, cuál de los nodos tiene menos voltaje, por último se declara la variable *peor\_tension* que va a ser igual al voltaje horario, donde permita extraer de ese vector el valor del nodo que tenga la mínima tensión.

En la tabla 4 se observan los datos de los valores de tensión hora a hora arrojados sin la generación distribuida. La figura 37 muestra el comportamiento del voltaje en un día.

Tabla 4. Datos de tensión sin GD

Hora	Tensión sin GD
1	0.9711
2	0.9791
3	0.9804
4	0.9844
5	0.9831
6	0.9778
7	0.9711
8	0.9574
9	0.9378
10	0.9262
11	0.9159
12	0.9098
13	0.9053
14	0.9144
15	0.9129
16	0.9114
17	0.9114
18	0.9114
19	0.9159
20	0.9038
21	0.9038
22	0.9144
23	0.9306
24	0.9477

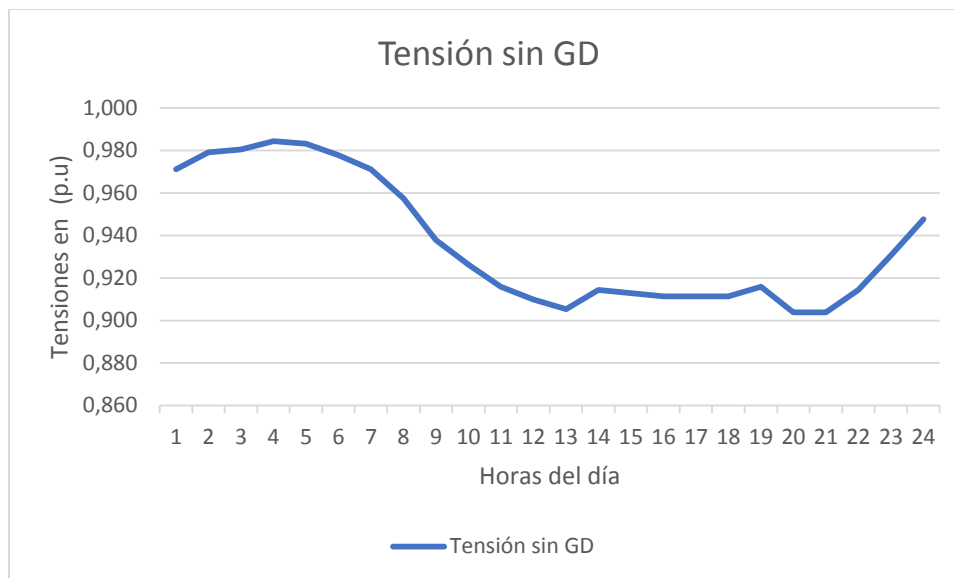


Figura 37. Tensión sin Gd

- **Cálculo de Corrientes**

Se halla la corriente máxima del caso base del sistema, de esta manera se puede seleccionar el conductor necesario de las líneas, para llevar a cabo este proceso se partirá de la ecuación 62, que expresa la corriente por las líneas de un sistema de distribución.

$$i_{12} = \frac{v_1 - v_2}{z_{12}} \quad (62)$$

Partiendo de la ecuación 62, se anexa en el código de flujo de potencia hora a hora en Matlab, las características para el cálculo de la corriente de todas las líneas y de esta manera tomar la mejor decisión para la selección del conductor en la tabla 31016 de la NTC 2050(Instituto Colombiano de Normas Técnicas y Certificación (ICONTEC), 1998).

Lo primero que se define en el código es el número de líneas, como el sistema cuenta con 33 nodos y 32 líneas, para que Matlab reconozca el número de líneas, se declara una constante  $NL = 32$  y se crea una variable  $eee$  que establece el rango de líneas de 1 hasta  $NL$  para el cálculo de cada corriente por medio de un ciclo *for*, se evalúa en cada ciclo de la columna 1 y 2

donde está la información de los nodos en la tabla 1, dentro del *for* se establecen los nodos de envío *node\_e* y los nodos de recibo *node\_r*, también se extrae la información de las resistencias y reactancias de la tabla 1, para poder calcular la impedancia en cada línea y se declara la resistencia *R\_1*, reactancia *X\_1* e impedancia *Z\_1*.

Se declaran las variables de los voltajes en los nodos de envío *V\_e* y de recibo *V\_r*, extrayendo la información del vector de tensión en cada barra del sistema, cabe aclarar que todos los datos en este ciclo están en *p.u.* Para calcular la corriente *I\_L* en cada línea solo basta aplicar la ecuación 62 en el código, dando como resultado la corriente en forma de fasor, para hallar su valor en *p.u* solo se saca la magnitud de esta variable y se calcula su resultado real multiplicando por la corriente base del sistema que es de 78.9889 Amperios, donde el resultado de *I\_L\_horario* son los valores de corriente de todas las líneas.

```

129 - for eee=1:NL
130 -     node_e=Sistema(eee,1);
131 -     V_e=V(node_e,1);
132 -     node_r=Sistema(eee,2);
133 -     V_r=V(node_r,1);
134 -     R_1=Sistema(eee,3);
135 -     X_1=Sistema(eee,4);
136 -     Z_1=R_1+X_1*i;
137 -     I_L=(V_e-V_r)/Z_1;
138 -     I_L=abs(I_L);
139 -     I_L_horario(eee,1)=I_L*78.9889;

```

Figura 38. Parámetros del sistema

Para calcular la máxima corriente del sistema se debe declarar la variable *peor\_corriente* que permita guardar las corrientes hora a hora en el día y con la función *max* se obtendrá la corriente máxima en la hora pico en un día de operación del sistema, de esta manera se procede a indicar el conductor a seleccionar. Se selecciona el calibre de 400 KCMIL en la

tabla 31016 de la NTC 2050 y se comprueba que el calibre soporta la corriente máxima del sistema (Instituto Colombiano de Normas Técnicas y Certificación (ICONTEC), 1998).

En la tabla 5 se observan los datos de los valores de corriente hora a hora arrojados sin la generación distribuida. La figura 39 muestra el comportamiento de la corriente en un día.

Tabla 5. Datos de corriente sin GD

Hora	Corriente sin Gd
1	111.8841
2	80.9931
3	75.8729
4	60.5600
5	65.6564
6	86.1212
7	111.8841
8	164.0403
9	238.5638
10	281.9965
11	320.5423
12	342.8068
13	359.6232
14	326.0919
15	331.6524
16	337.2240
17	337.2240
18	337.2240
19	320.5423
20	365.2516
21	365.2516
22	326.0919
23	265.6337
24	201.0742



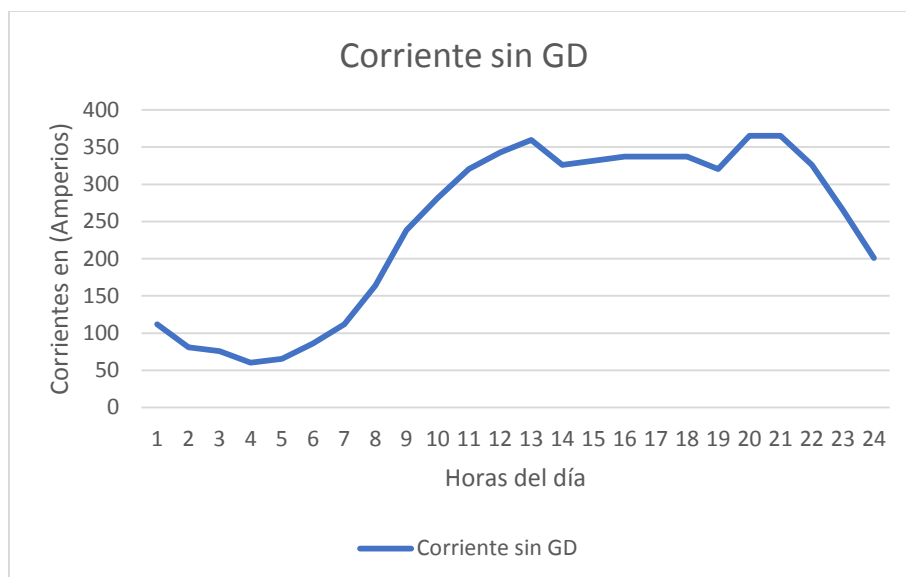


Figura 39. Grafica de la corriente sin GD

En la tabla 6, se observan los valores de pérdidas de energía hora a hora arrojados sin generación distribuida. La figura 40 muestra el comportamiento de las pérdidas de energía en un día de operación.

Tabla 6. Datos de pérdidas de energía sin GD

Tiempo	Pérdidas de energía sin GD
1	19.331
2	10.103
3	8.862
4	5.638
5	6.630
6	11.428
7	19.331
8	41.750
9	88.911
10	124.744
11	161.777
12	185.434
13	204.412
14	167.518
15	173.374
16	179.346

17	179.346
18	179.346
19	161.777
20	210.979
21	210.979
22	167.518
23	110.515
24	62.943

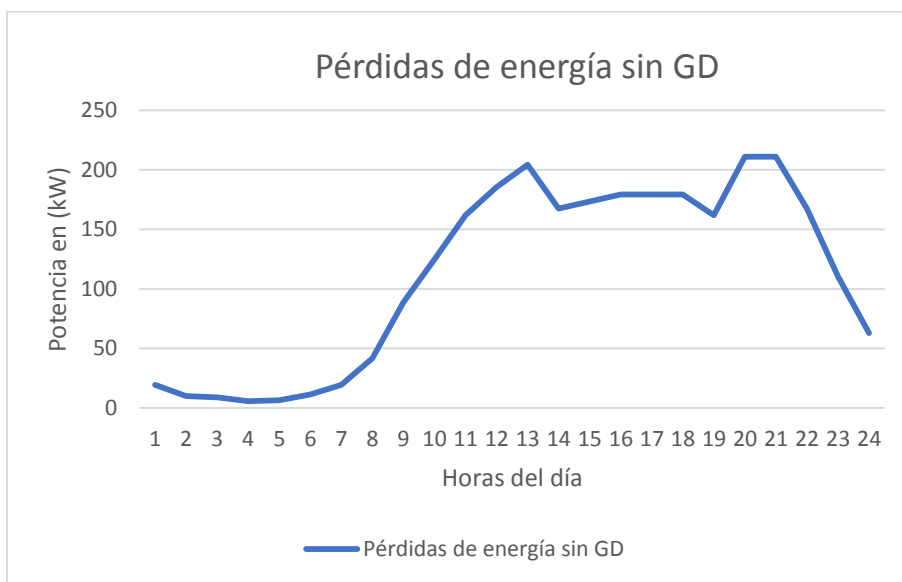


Figura 40. Grafica de pérdidas de energía hora a hora

Cabe resaltar que la corriente de base se obtiene con los datos base del sistema, con la fórmula de potencia general. En la figura 41, se presenta el diagrama de flujo donde se describe el proceso para el desarrollo de la programación de los flujos de carga hora a hora sin generación distribuida.

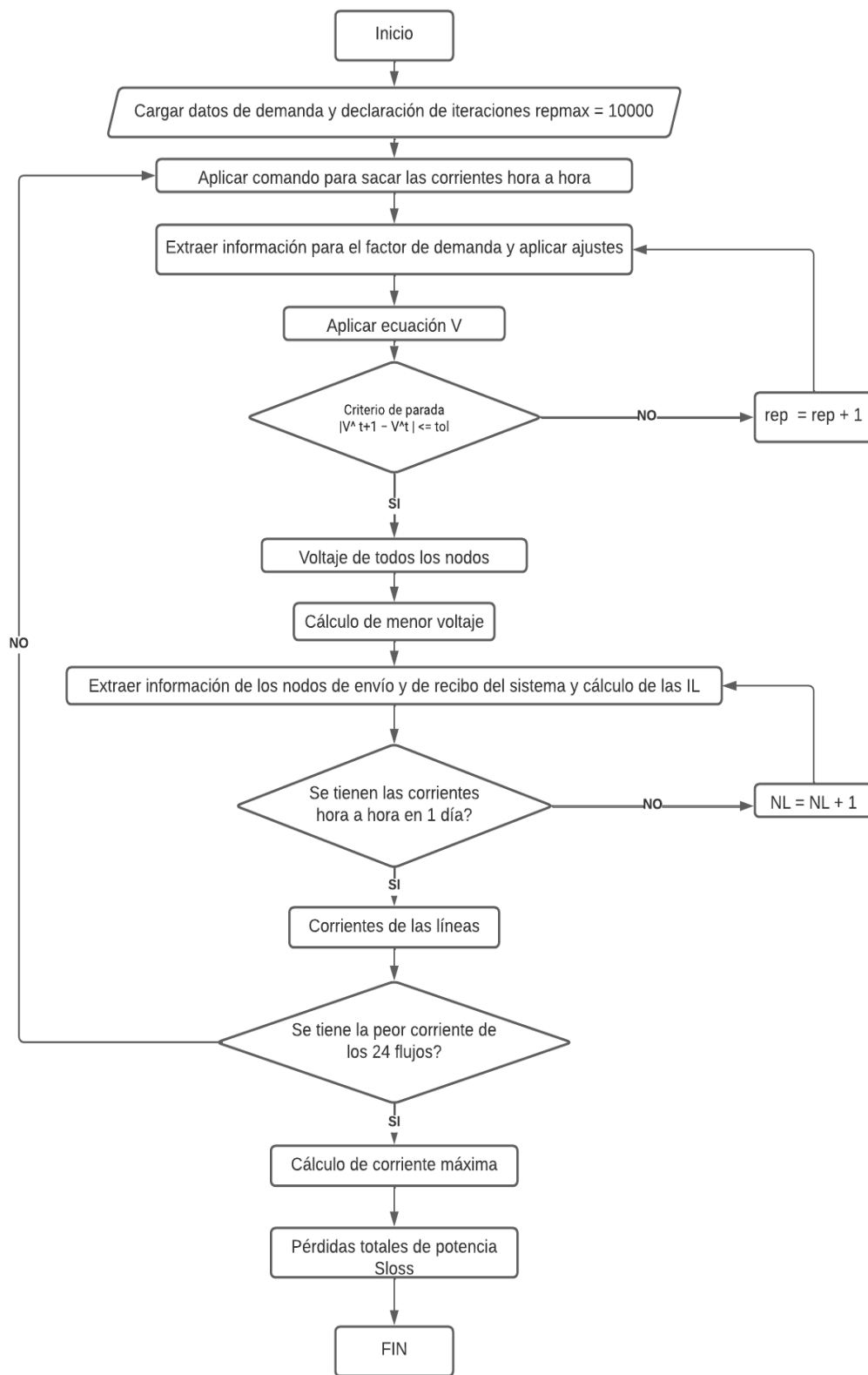


Figura 41. Diagrama de flujo para el flujo de carga hora a hora sin GD

### 3.2.3.2 Programación del Método Aproximaciones Sucesivas 24 Horas con GD

Siguiendo con el código planteado se procede a incluir los generadores, teniendo en cuenta que la información de la ubicación y el rango de inyección de los generadores, se extrae de (Grisales-Noreña et al., 2018a). Para empezar, se declara un vector  $GD$ , donde se permita darle posición a los generadores que están localizados en los nodos 12, 15 y 31, inicialmente se plantea el escenario máximo del rango de potencia que inyecta los generadores el cual es de  $1.2MW$ .

Se agrega al ciclo de 24 horas, compuesto en el código por el *for* en la línea 9 y se crea la variable *factor\_de\_generacionsolar*, se extrae los datos de la columna de radiación, en la ciudad de Medellín ubicados en la segunda columna de la tabla 2, se declara la inyección variable en el código, que es la inyección de los generadores con respecto a la variación de la radiación solar en 24 horas, para lograr que Matlab entienda este procedimiento se declara la variable *inyeccion* con los valores de potencia máxima de los generadores dentro de un vector y se dividen en  $S_{base}$  para establecerlos en *p.u* y la multiplicamos por el factor de generación solar, se verifica que solo existe efecto en las horas transcurridas entre las 7:00 a.m. y 7:00 p.m.

```

9 - Inyeccion= [1200 1200 1200]/Sbase;%se divide por la potencia base
10 - for iii=1:24
11 -     factor_de_demanda=parametros(iii,1);
12 -     factor_de_generacionsolar=parametros(iii,2);
13 -     Inyeccion_variable=Inyeccion*factor_de_generacionsolar;

```

Figura 42. Inclusión de los generadores

Se declara un ciclo *for* que permita modificar la demanda final, en el momento de inyectar potencia con los generadores hora a hora según el porcentaje de radiación del día, para llegar a esto se crea una variable *gen*, que va a tener un rango de uno a tres valores, para poder simular tres ciclos donde se pueda extraer la información de la ubicación de cada generador

según el orden del vector GD, se extrae la posición de la matriz sistema, igualmente la información de potencia activa con respecto a la posición del nodo GD y la columna 5 de potencia activa, este valor se le resta la inyección generada según la radiación del día, se realiza el mismo procedimiento con la matriz de nodos con esto se finaliza el ciclo.

```

109 - for gen=1:3
110 -     nodo_gd=GD(1,gen);
111 -     pos=find(Sistema(:,2)==nodo_gd);
112 -
113 -     Sistema(pos,5)=Sistema(pos,5)-Inyeccion_variable(1,gen);
114 -
115 -     pos=find(Node(:,1)==nodo_gd);
116 -     Node(pos,4)=Node(pos,4)-Inyeccion_variable(1,gen);
117 - end

```

Figura 43. Ciclo de inyección de potencia de los generadores

Se observa una vez terminado el código de flujo de carga 24 horas con variación en los generadores distribuidos, se observa que en las horas doce y trece los generadores inyectan más potencia, cabe destacar que desde la hora 7:00 a.m. hasta las 7 p.m. los generadores inyectan potencia. Esto se corrobora dentro del código observando la matriz sistema hora a hora y mirando que en los nodos donde están los generadores, pasan a tener valores negativos, esto quiere decir que pasan de ser cargas y se convierte en nodos de inyección (PV), los cuales controlan potencia activa y voltaje en el nodo.

Una vez realizado el código en Matlab, se procede a encontrar la mejor combinación de potencia entre los tres generadores en el cual disminuya las pérdidas al máximo, partiendo de la potencia que tienen los tres generadores con un valor de 0W, se evidencia un resultado de pérdidas de 2691.988479 kW, siguiendo la recomendación de (Grisales-Noreña et al., 2018a), donde la potencia máxima que inyecta cada generador es de 1200W, se observa que existe una reducción de 2156.537338 kW. El procedimiento se realizó de forma manual buscando iteración por iteración, en el cual consistía en hallar la menor disminución en las pérdidas de energía, se

realizaron alrededor de 217 iteraciones dando como resultado que la mejor combinación para este sistema en sus generadores de  $G1=1000kW$ ,  $G2=300kW$  y  $G3=1000kW$ , cuyas pérdidas fueron de  $1832.970354kW$ . Así mismo se halló el valor del voltaje en  $p.u$  y el valor de la corriente, el sistema presentó una reducción aproximada del 31.91%. Debido al número extenso de iteraciones necesarias para encontrar la mejor solución del sistema, se hace necesario aplicar un método de optimización ya que al realizar esto de forma manual, conlleva un gran número de datos y de iteraciones, además esto no garantiza que sea la mejor solución, en cambio con un método de optimización garantiza que su resultado sea una aproximación la mejor solución para el sistema.

En la tabla 13 se observan los datos de los valores de tensión hora a hora arrojados de la posible mejor solución. La figura 44 muestra el comportamiento del voltaje en un día.

Tabla 7. Tabla de voltajes hora a hora con generación manual

Tiempo	Voltaje
1	0.971
2	0.979
3	0.980
4	0.984
5	0.983
6	0.978
7	0.971
8	0.962
9	0.959
10	0.968
11	0.972
12	0.976
13	0.973
14	0.980
15	0.973
16	0.965
17	0.943

18	0.927
19	0.918
20	0.904
21	0.904
22	0.914
23	0.931
24	0.948

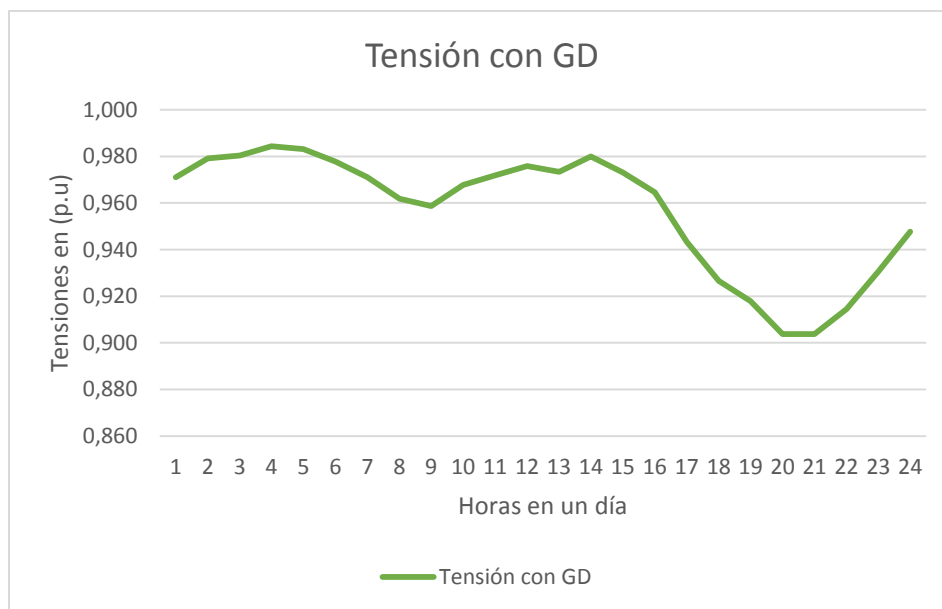


Figura 44. Grafica de voltajes hora a hora con generación manual

En la tabla 8 se observan los datos de los valores de corriente hora a hora arrojados de la posible mejor solución. La figura 45 muestra el comportamiento de la corriente en un día.

Tabla 8. Tabla de corrientes hora a hora con generación manual

Tiempo	Corriente
1	111.884
2	80.993
3	75.873
4	60.560
5	65.656
6	86.121

7	111.875
8	154.795
9	195.828
10	198.765
11	207.029
12	208.567
13	220.996
14	192.983
15	209.140
16	230.049
17	272.139
18	305.514
19	316.161
20	365.252
21	365.252
22	326.092
23	265.634
24	201.074

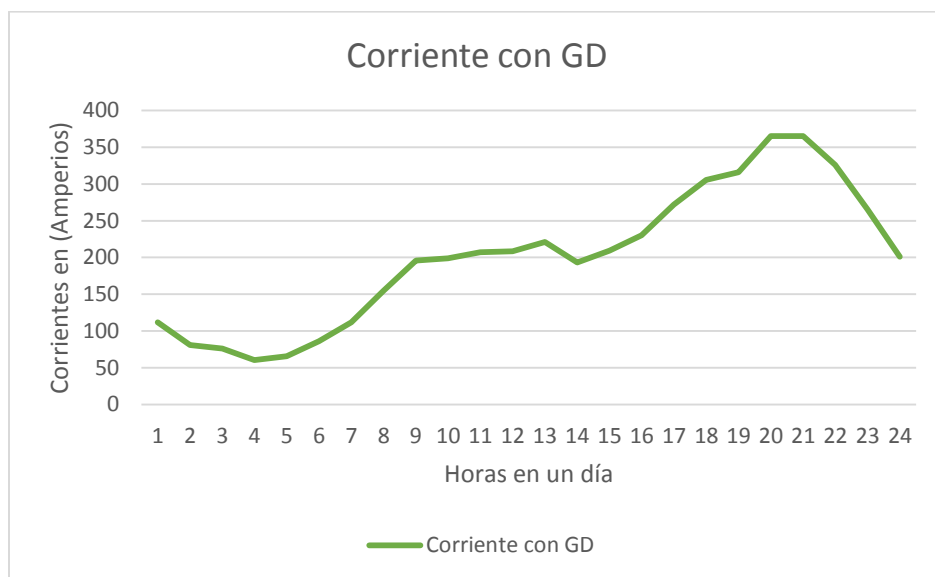


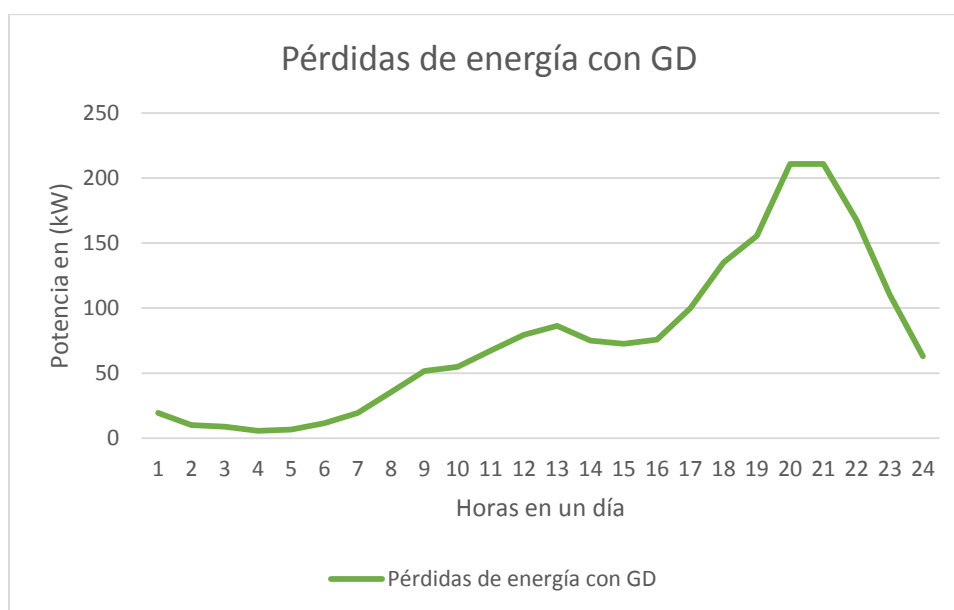
Figura 45. Grafica de corrientes hora a hora con generación manual



En la tabla 9 se observan los datos de los valores de las pérdidas de energía hora a hora arrojados de la posible mejor solución. La figura 46 muestra el comportamiento de las pérdidas de energía en un día.

Tabla 9. Tabla de pérdidas de energía hora a hora con generación manual

Tiempo	Pérdidas
1	19.331
2	10.103
3	8.862
4	5.638
5	6.630
6	11.428
7	19.327
8	35.265
9	51.722
10	54.831
11	67.437
12	79.426
13	86.224
14	74.947
15	72.523
16	75.874
17	99.995
18	135.073
19	155.404
20	210.979
21	210.979
22	167.518
23	110.515
24	62.943



*Figura 46. Grafica de pérdidas de energía hora a hora con generación manual*

En la figura 47, se presenta el diagrama de flujo donde se describe el proceso, los parámetros y las variables para el desarrollo de los flujos de carga hora a hora, incluyendo generación distribuida de forma manual.

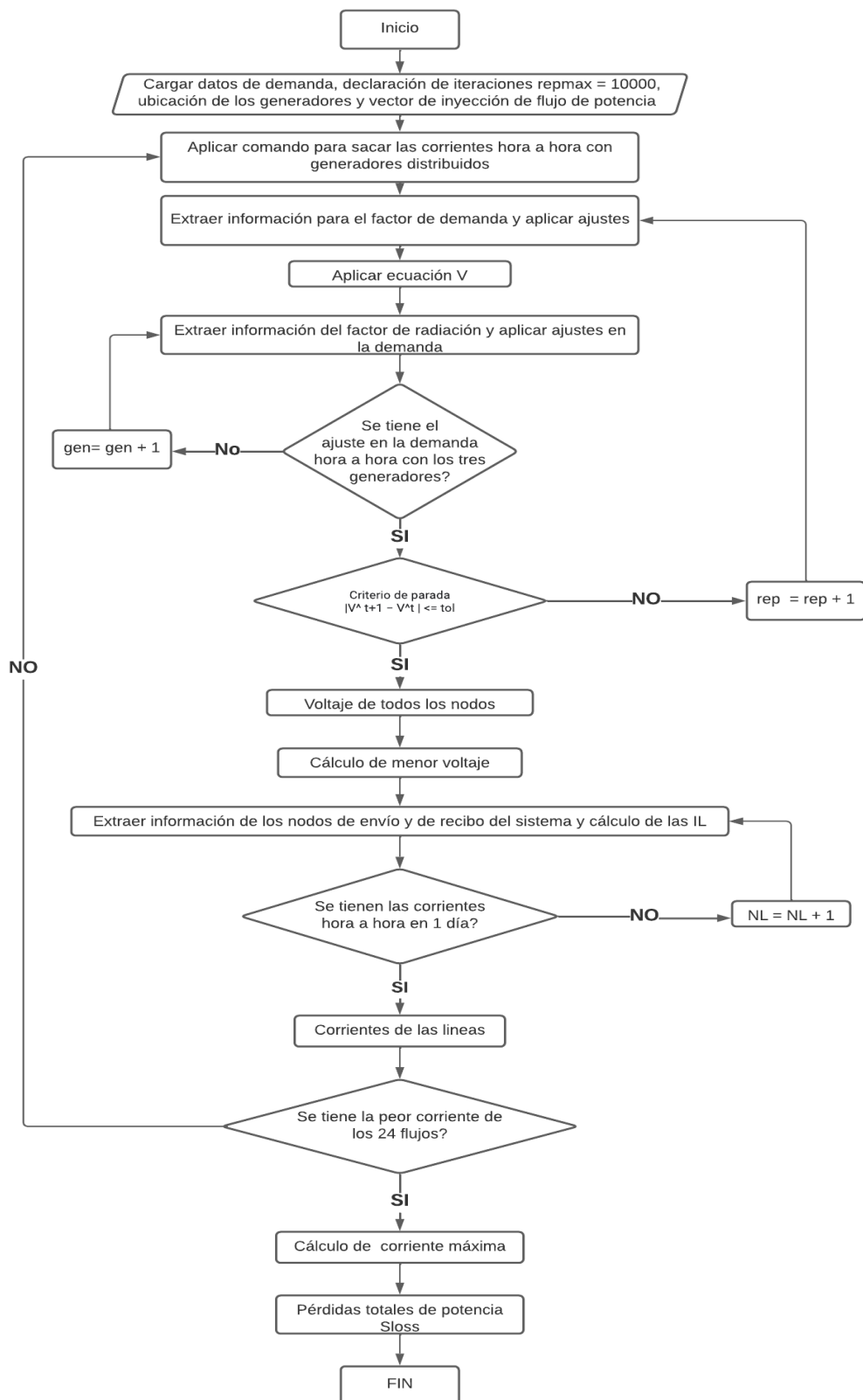


Figura 47. Diagrama de flujo para los flujos de potencia con generación manual

## 4 Flujos Óptimos de Potencia

Este capítulo se basa en hallar el flujo de potencia óptimo para el sistema en el cual se incluyen tres generadores distribuidos, analizando la situación se observa que el problema es de operación y no de localización, ya que la localización se extrajo del artículo (Grisales-Noreña et al., 2018a), donde indica que la mejor ubicación para los generadores se encuentra en los nodos 12, 15 y 31. Se analiza cual es la potencia inyectada hora a hora para alcanzar la función objetivo que se plantea en el trabajo, teniendo en cuenta dos algoritmos de optimización el PSO y SCA.

Para el sistema de 33 nodos, la ubicación de los generadores fueron localizados usando el método PPBIL, con un rango de potencia que no exceda los 1.2 MW, los valores de voltajes en *p.u* no pueden exceder de 0.95 y 1.05 de operación, además la corriente no debe ser superior a los 380 A.

En el caso de este tipo de problema se disminuyen las pérdidas de energía por medio de una función objetivo. Para minimizar las pérdidas por medio del flujo optimo existen dos opciones:

Tabla 10. Opciones para minimizar pérdidas de energía

	Dimensionamiento	Operación
Pregunta	¿Qué tamaño de instalación de los paneles es necesario?	¿Cuánta potencia se debe inyectar hora a hora?
Alcance	Minimizar las pérdidas de energía.	Máxima reducción de pérdidas de energía.

### 4.1 Algoritmo PSO (Optimización por Enjambre de Partículas)

Teniendo en cuenta el código de flujo de potencia 24 horas con generación distribuida manual, este representa nuestra función objetivo y se incluye por medio del comando *function*, este procedimiento permite separar el algoritmo de optimización de la función objetivo.

```
1 function [total_power_loss]=integracion_gd(Xii,Yii,Zii)
```

Figura 48. Función objetivo

La única modificación que se le realizó a este código se encuentra en la parte de inyección, en este caso no van a ver valores concretos como el código de la GD manual, sino que vamos a declarar  $X_{ii}, Y_{ii}, Z_{ii}$ , estos son los datos de inyección a evaluar dentro de la función objetivo, donde *total\_power\_loss* representa la salida de pérdida de potencia en el sistema y *integración\_gd* los datos de entrada.

- **Pseudocódigo PSO**

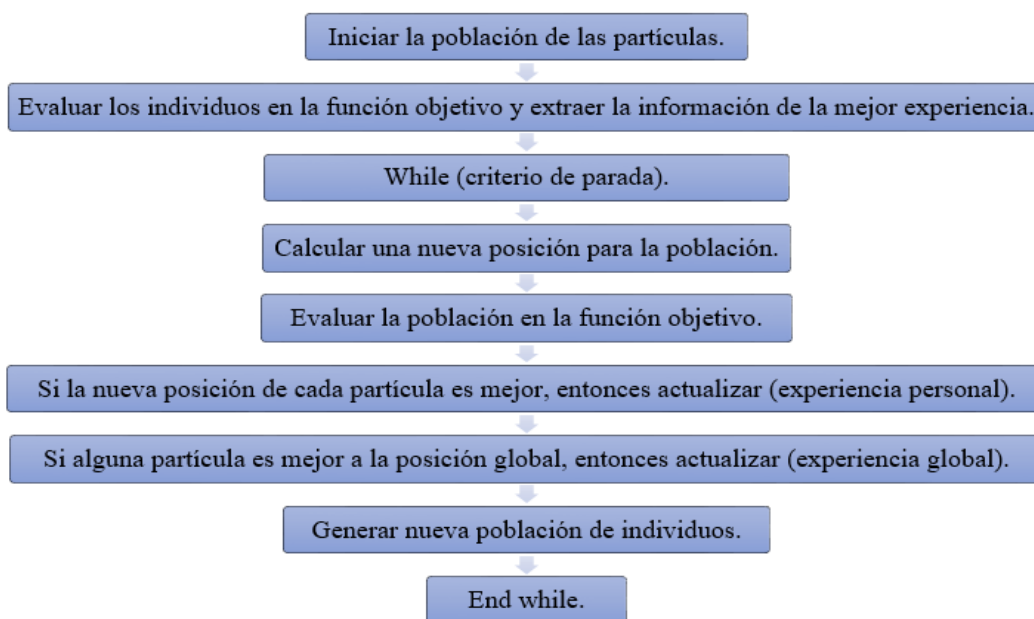


Figura 49. Pseudocódigo PSO

fuelle: (Grisales-Noreña et al., 2018b)

- **Datos Básicos**

Para la realizar este código es importante tener en cuenta el espacio de búsqueda, que permita encontrar el mejor valor que representa la inyección de flujo de potencia de un generador en un sistema de distribución, el espacio de búsqueda va a tener 3 dimensiones, las cuales dependen de los generadores y está representado con los vectores dentro del rango de 0 a 1200 kW, como lo propone el artículo(Grisales-Noreña et al., 2018a).

Se crea la constante “P” la cual indica el número de partículas del sistema, en este caso se selecciona 15 partículas, se declaran  $D_{min}$  y  $D_{max}$ , las cuales representan los límites mínimos y máximos del espacio de búsqueda.

Se declaran las dimensiones por medio de la variable  $Dim$ , con el comando *size* se extraen las variables del sistema. Se procede a crear una inercia máxima y una inercia mínima establecidas por las variables  $inermax$  y  $inermin$  que están dentro del rango de 1 a 0, siendo  $inermax$  una muy grande en este caso de 0.7 y  $inermin$  muy pequeña de 0.001, estos valores permiten que la técnica tenga un desplazamiento, se define a  $w$  como la inercia máxima. Se presenta el número de iteraciones y la tolerancia  $E$  con un valor de  $1 * 10^{-6}$ , por último, se incluyen los coeficientes de aprendizaje cognitivo personal  $C1 = 1.494$  y grupal  $C2 = 1.494$  como lo muestra la figura.

```

5 - P=15;% cantidad de partículas %%
6 - Dmax=[ 1200 1200 1200];
7 - Dmin=[0 0 0];
8 - Dim=size(Dmax,2);
9 - inermax=0.7;
10 - inermin=0.001;
11 - w=inermax;
12 - E=1e-6;
13 - itermax=100;
14 - C1=1.494; %coeficiente cognitivo individual
15 - C2=1.494; %coeficiente cognitivo grupal
16

```

Figura 50. Parámetros de inicio del método PSO

- **Velocidades del Algoritmo**

Partiendo de que el problema tiene tres dimensiones en el sistema, se crean tres velocidades máximas distintas para cada grupo de partículas ( $V_{max\_p1}$ ,  $V_{max\_p2}$ ,  $V_{max\_p3}$ ), como tienen el mismo límite se definen como 0.01, lo que garantiza que inicialmente se desplacen de forma muy corta, cada vez que el algoritmo vaya avanzando aumenta su valor, se define un contador que toma como referencia el valor en el cual inicia la iteración, se crea la velocidad de las partículas en las posiciones de la siguiente manera ( $Vel\_p1$ ,  $Vel\_p2$ ,  $Vel\_p3$ ): Un valor aleatorio (*rand*) que multiplica la cantidad de partículas en una dimensión para cada individuo, se hace el producto de dos veces la velocidad máxima de cada grupo de partículas y restada por la velocidad máxima de cada grupo de partículas, de esta manera se asegura de que se encuentren dentro de su rango de velocidades sin desfasarse de ella misma. Una vez creadas todas las velocidades, se almacenan en un vector llamado *Vel*, donde se observa que se crean velocidades tanto positivas como negativas, cabe resaltar que este ítem en la literatura se le denomina una velocidad, pero realmente a la posición se le suma una velocidad, en esta parte de programación resulta siendo un desplazamiento de las partículas, estas son las velocidades

representadas por nuestros individuos ya que dependiendo de ellos van a avanzar o retroceder según su posición.

```

17 - Vmax_p1=0.01;
18 - Vmax_p2=0.01;
19 - Vmax_p3=0.01;
20 - % Vmax_p3=
21 - cont=1;
22 - Vel_p1=rand(P,1)*(2*Vmax_p1)-Vmax_p1; %velocidad de las partículas en las posiciones %%
23 - Vel_p2=rand(P,1)*(2*Vmax_p2)-Vmax_p2;
24 - Vel_p3=rand(P,1)*(2*Vmax_p3)-Vmax_p3;
25 - Vel=[Vel_p1 Vel_p2 Vel_p3];

```

Figura 51. Velocidades del algoritmo

- **Creación de la posición inicial  $X_i$**

Para crear la partícula inicial se aplica la ecuación 63 y tiene que cumplir con los siguientes pasos:

**Paso1:** Garantizar que la partícula esté de forma aleatoria.

**Paso2:** Garantizar que la partícula no se salga del espacio de búsqueda o dentro del espacio de soluciones.

$$X_i = (D_{max} - D_{min}) \cdot \text{rand}(P, Dim) + D_{min} \quad (63)$$

Se procede a crear un vector de ceros que almacene las soluciones dentro del problema y su dimensión está establecida por el número de partículas, este vector es igual a *Aptitud\_Xi* como se presenta en la figura 52.

```

27 - Xi=(Dmax-Dmin) .*rand(P,Dim)+Dmin; %posición inicial
28 - Aptitud_Xi=zeros(P,1);%almacena las soluciones dentro del problema

```

Figura 52. Programación de la posición inicial  $X_i$



- **Evaluación de los Individuos en la Función Objetivo y Evaluación de la Mejor Experiencia.**

Para evaluar la función objetivo que es el flujo de 24 horas con generación distribuida manual, se procede a crear un ciclo *for* dentro del código del algoritmo de optimización, donde permita que el número de ciclos sea proporcional a las partículas y dentro de esta se crea una variable llamada *resultado*, que va a ser igual a la función objetivo evaluada en las posiciones iniciales creadas en los 3 ejes, por cada solución creada se va a almacenar el resultado en el vector de *Aptitud\_Xi* como lo presenta la figura 53.

```

30 - for i=1:P %evaluación de las funciones por medio del ciclo for %%
31 -     Resultado=integracion_gd(Xi(i,1),Xi(i,2),Xi(i,3));
32 -     Aptitud_Xi(i,1)=Resultado%evaluación de la funcion objetivo %%
33 - end

```

Figura 53. Evaluación de la función objetivo

Nuevamente se almacenan las soluciones de la evaluación en la función objetivo en un nuevo vector llamado *Aptitud\_mejorposi*, se crea el vector *MejorXi* que va a ser igual a las posiciones iniciales creadas, se hace de esta manera para volver a reutilizar las variables creadas y no tener conflictos con la programación, ya que en el algoritmo por cada ciclo se van actualizando y almacenando las soluciones y posiciones de los nuevos *Xi*. Usando el comando *Aptitud\_mejorpos\_g Ind*, extrae la mejor solución de las soluciones creadas que es el menor valor de los *resultados* almacenados y es la mejor solución global hasta el momento, cada vez que se vayan creando más partículas, se van actualizando y almacenando estos resultados, de esta manera se va encontrando una reducción de potencia a la hora de evaluar la función objetivo, la función *Aptitud\_mejorpos\_g Ind* también indica cual partícula genera el resultado más pequeño dentro del espacio de búsqueda. El vector *mejorposXig* va a almacenar las

coordenadas de la partícula que dio el menor resultado hasta ahora, que representa la mejor solución global, después se crea el avance de la posición  $X_i$ .

```

34 - Aptitud_mejorposi=Aptitud_Xi; % almacenamos las soluciones
35 - MejorXi=Xi;
36 - [Aptitud_mejorpos_g Ind]=min(Aptitud_mejorposi);
37 - mejorposXig=MejorXi(Ind,:);
38 - Xi=Xi+Vel;

```

Figura 54. Actualización y almacenamiento de datos

- **Restricción del Espacio de Búsqueda**

Cada vez que se crea un avance de las partículas, se tiene que declarar una restricción para que no se desborden y se salgan de la región factible, esto se logra creando un ciclo *for* donde permita recorrer el numero de ciclos proporcional a las partículas del problema y evaluando que cada una cumpla con las condiciones de los límites máximos y mínimos, si incumple con esta condición se crea una nueva partícula de forma aleatoria hasta que cumplan las condiciones todas las partículas.

```

40 %restricción del espacio de busqueda
41 - for i=1:P
42 -     for j=1:Dim
43 -
44 -         if ((Xi(i,j)<Dmin(1,j)) || (Xi(i,j)>Dmax(1,j)))
45 -             Xi(i,j)=Dmin(1,j)+(Dmax(1,j)-Dmin(1,j))*rand;
46 -         end
47 -     end
48 - end

```

Figura 55. Restricción del espacio de búsqueda

- **Proceso Total iterativo y Método de Aprendizaje de la Técnica de Optimización**

Se declara un *while* incluyendo la condición la cual es que cuando la mejor posición global *Aptitud\_mejorpos\_g* sea mayor que el error *E* cierre el ciclo, además se define un vector de ceros que va a ser el encargado de reiniciar las soluciones almacenadas anteriormente en

*Aptitud\_Xi*, y dentro de este *while* se define un *for* que evalúa nuevamente la función objetivo, para poder comparar el resultado actual con el anterior y de esta manera encontrar un avance, se crea un condicional *if* donde se compare la solución actual *Aptitud\_Xi* con el almacenamiento de las soluciones anteriores *Aptitud\_mejorposi*, si esta nueva solución encuentra un mejor resultado que las anteriores compara y actualiza, si no está mejorando no hace ningún cambio, lo mismo se hace para actualizar la posición de las partículas *MejorXi*.

```

53 - while (Aptitud_mejorpos_g>E)
54 -
55 -     Aptitud_Xi=zeros(P,1);%almacena las soluciones dentro del problema
56 -
57 -     % Método de aprendizaje de las tecnicas de optimizacion %%
58 -     for i=1:P
59 -         Resultado=integracion_gd(Xi(i,1),Xi(i,2),Xi(i,3));
60 -         Aptitud_Xi(i,1)=Resultado;%evaluacion de la funcion objetivo
61 -         if (Aptitud_Xi(i,1)<Aptitud_mejorposi(i,1))%%compara y actualiza los resultados
62 -             Aptitud_mejorposi(i,1)=Aptitud_Xi(i,1);
63 -             MejorXi(i,:)=Xi(i,:);
64 -         end

```

Figura 56. Segunda evaluación de la función objetivo

De esta manera el algoritmo va aprendiendo constantemente que resultado es el mejor en cada ciclo, se crea un nuevo condicional *if* para comparar y actualizar con la condición del método de aprendizaje de la técnica de optimización  $Aptitud\_mejorposi < Aptitud\_mejorpos\_g$ , de esta manera asegura que las mejores posiciones se pueden extraer del vector *Aptitud\_mejorposi* y tener el mejor resultado global con su coordenada *mejorposXig*.

```

65 -         if (Aptitud_mejorposi(i,1)<Aptitud_mejorpos_g)
66 -             Aptitud_mejorpos_g=Aptitud_mejorposi(i,1);
67 -             Ind=i;
68 -             mejorposXig=MejorXi(Ind,:);
69 -         end
70 -     end

```

Figura 57. Aprendizaje de la técnica de optimización

#### 4.1.1 Método de Avance de la Técnica de Optimización

- **Generar la nueva población de individuos**

Continuando en el ciclo *while* se crea una condición, declarando que, si el proceso iterativo no ha terminado de mejorar y el contador es menor que el número de iteraciones se empieza a crear el avance de la técnica de optimización o el método de aprendizaje, se crean dos *for* que vayan de  $i = 1:P$  y de  $j = 1:Dim$  y se empieza a crear las nuevas partículas. Para aplicar la ecuación general de la actualización de la velocidad se declaran  $R1$  y  $R2$  el cual son dos valores aleatorios de 0 a 1 dentro en la ecuación, entonces el vector nuevo de velocidades de las partículas se expresa de la siguiente manera:

$$Vel(i,j) = w * Vel(i,j) + C1 * R1 * (MejorXi(i,j) - Xi(i,j)) + C2 * R2 * (mejorposXig(1,j) - Xi(i,j)) \quad (64)$$

Esta ecuación se obtiene del artículo (D. Wang et al., 2017), una vez aplicada esta ecuación se actualiza la posición de la partícula.

```

71      % método de avance de la técnica de optimización
72      %este método es el que siempre cambia en cada técnica de optimización
73      %%%
74
75      if (Aptitud_mejorpos_g>E && cont<itermax)
76          for i=1:P
77              for j=1:Dim
78
79                  R1=rand();
80                  R2=rand();
81                  Vel(i,j)=w*Vel(i,j)+C1*R1*(MejorXi(i,j)-Xi(i,j))+C2*R2*(mejorposXig(1,j)-Xi(i,j));
82                  Xi(i,j)=Xi(i,j)+Vel(i,j);
83              end
84          end
85
86      end

```

Figura 58. Método de avance de la técnica de optimización

Teniendo actualizada la nueva velocidad y posición de la partícula se procede a aplicarle la restricción, para que no se desborde dentro de los límites máximos y mínimos del espacio de búsqueda en las tres dimensiones, igualmente como se realizó en la sección anterior.

```

85 -   for i=1:P
86 -       for j=1:Dim
87 -
88 -           if ((Xi(i,j)<Dmin(1,j)) || (Xi(i,j)>Dmax(1,j)))
89 -               Xi(i,j)=Dmin(1,j)+(Dmax(1,j)-Dmin(1,j))*rand;
90 -           end
91 -       end
92 -   end
--

```

Figura 59. Restricción del espacio de búsqueda actualizado

Finalmente se anexa un *cont* igual a *cont* + 1 y se establece un criterio de parada dentro de un condicional, donde establezca que el contador sea mayor que las interacciones máximas, si esto es así, finalizan las iteraciones del código y aplica un *break*, así mismo se agrega al inicio del código el comando *tic* y al finalizar el comando *toc*, para conocer el tiempo en que Matlab procesa este código hasta que encuentre una respuesta, en la figura 60 se muestra el diagrama de flujo de la programación completa de los flujos óptimos con el método de optimización PSO.

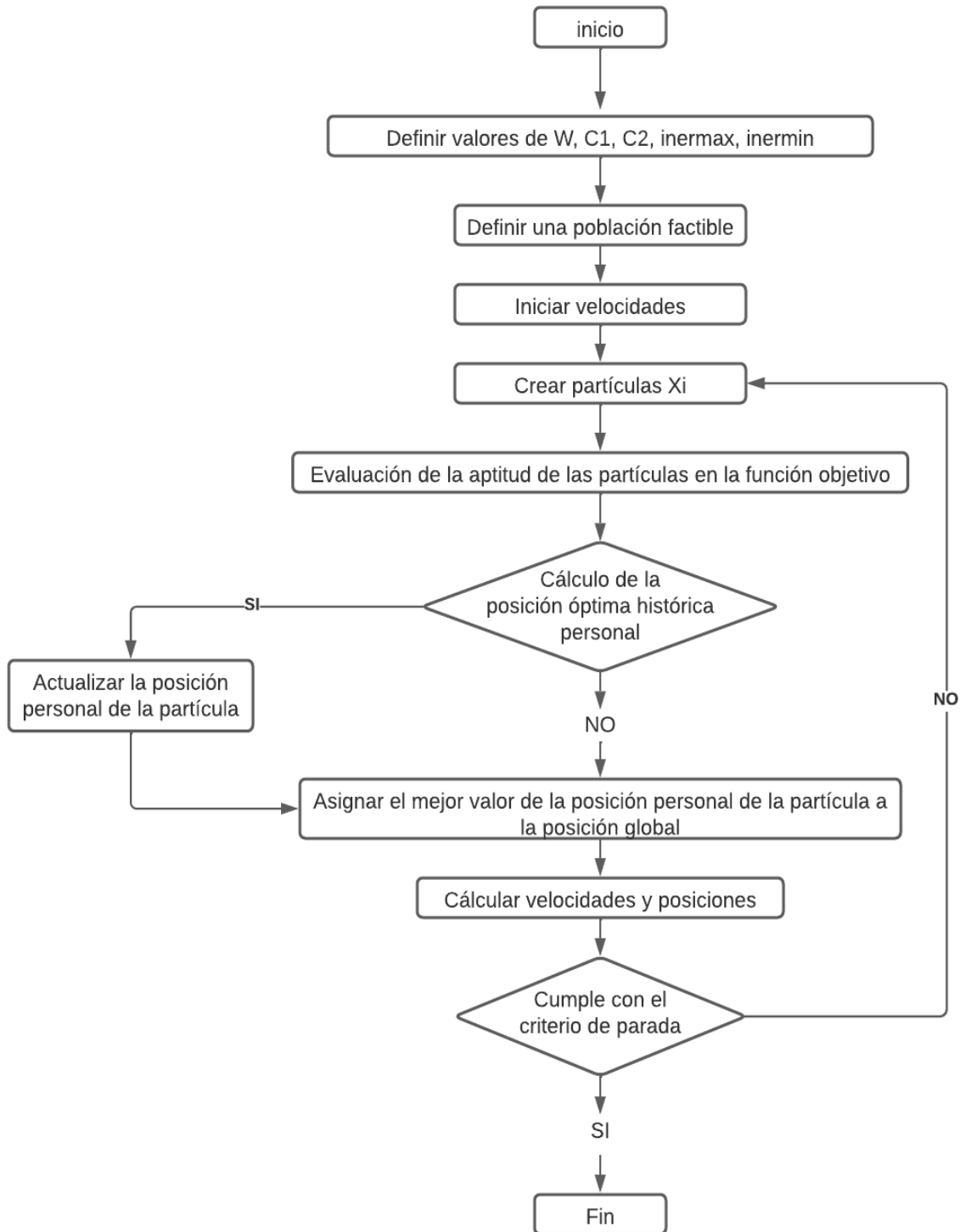


Figura 60. Diagrama de flujo de la programación de los flujos óptimos con PSO

## 4.2 Algoritmo SCA (Senos y Cosenos)

Para el procedimiento de este método de optimización se toma como referencia algunos pasos explicados en el PSO, tales como la creación de la *FUCTION* para almacenar la función objetivo, la creación de la posición inicial de las partículas  $X_i$ , la evaluación de la función objetivo, las restricciones, el proceso iterativo y método de aprendizaje de las técnicas de optimización, ya que para aplicar este método solo tiene una variación en los datos básicos iniciales y el método de avance de la técnica de optimización SCA.

- **Seudocódigo SCA**

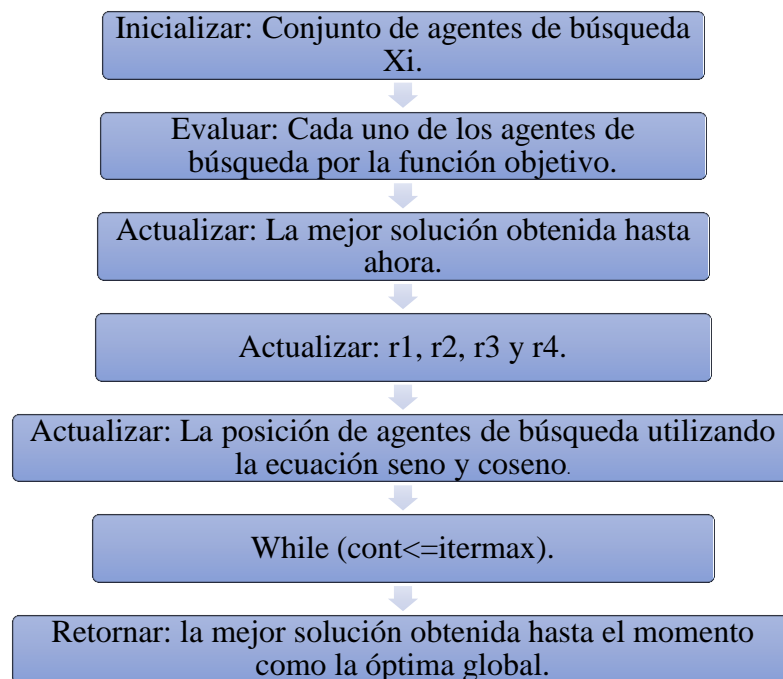


Figura 61. Pseudocódigo del SCA

fuelle: (Mirjalili, 2016)

- **Datos Básicos**

En comparación al método PSO en el proceso del SCA no se tiene en cuenta las velocidades, ni la inercia por que maneja un aprendizaje diferente, de igual forma se emplean

iguales parámetros tales como  $P$ ,  $D_{max}$ ,  $D_{min}$ ,  $Dim$ ,  $E$ ,  $itermax$  y  $cont$ , la nueva constante incluida es Alpha, que tiene un valor constante de 2 según el artículo (Mirjalili, 2016).

```

6 - P=15;% cantidad de partículas
7
8 - Dmax=[ 1200 1200 1200];
9 - Dmin=[0 0 0];
10 - Dim=size(Dmax,2);
11 - alpha= 2;
12 - E=1e-6;
13 - itermax=15;
14 - cont=1;

```

Figura 62. Parámetros de inicio SCA

- **Creación de Posición Inicial Xi o Conjunto de Agentes de Búsqueda Xi**

Para la realización de este ítem se toma en cuenta el mismo proceso realizado en el código del PSO en la figura 52.

- **Evaluación de los individuos en la función objetivo y extraer la información de la mejor experiencia**

De igual forma se toma en cuenta el mismo proceso realizado en el código del PSO en la figura 53 y 54.

- **Proceso Total Iterativo y Método de Aprendizaje de la Técnica de Optimización**

Según el seudocódigo indica que ahora el ciclo *while* tiene como condición  $cont \leq itermax$  por cada ciclo, esta es la única variación en esta parte del código, de igual forma el método de aprendizaje es el mismo.



```

45 - while (cont<=itermax)
46 -     Aptitud_Xi=zeros(P,1);%almacena las soluciones dentro del problema
47 -     % Método de aprendizaje de las técnicas de optimización %%
48 -     for i=1:P
49 -         Resultado=integracion_gd(Xi(i,1),Xi(i,2),Xi(i,3));
50 -         Aptitud_Xi(i,1)=Resultado;%evaluación de la función objetivo %%
51 -         if (Aptitud_Xi(i,1)<Aptitud_mejorposi(i,1))%compara y actualiza los resultados
52 -             Aptitud_mejorposi(i,1)=Aptitud_Xi(i,1);
53 -             MejorXi(i,:)=Xi(i,:);
54 -         end
55 -         if (Aptitud_mejorposi(i,1)<Aptitud_mejorpos_g)
56 -             Aptitud_mejorpos_g=Aptitud_mejorposi(i,1);
57 -             Ind=i;
58 -             mejorposXig=MejorXi(Ind,:);
59 -         end

```

Figura 63. Método de aprendizaje SCA

#### 4.2.1 Método de Avance de la Técnica de Optimización SCA

Este método ofrece dos ecuaciones para la actualización de las posiciones de las partículas por cada iteración, se evalúa por medio del seno y coseno donde definen un espacio entre dos soluciones, en el método de avance del código se declara dos ciclos *for*, el primero que va de  $i = 1:P$  y el segundo  $j = 1:Dim$ , después se declaran cuatro parámetros para el correcto funcionamiento del SCA los cuales son  $r1$ ,  $r2$ ,  $r3$ ,  $r4$ , donde  $r1$  establece la dirección del movimiento y se define por la fórmula  $alpha - cont * (alpha / itermax)$ ,  $r2$  es el desplazamiento de la posición y está definida por  $(2 * pi) * rand()$ ,  $r3$  es un valor que da peso mayor a 1 con el fin de recalcar que no es predecible en la evaluación de la actualización de la posición  $2 * rand()$ ,  $r4$  es un valor aleatorio y se define una condición donde se establece que si  $r4$  es menor a 0.5, entonces se aplica la ecuación del seno, si  $r4$  es mayor a 0.5 se aplica la ecuación del coseno y finaliza actualizando la partícula  $Xi = X$  evaluada.

```

67 - for i=1:P
68 -     for j=1:Dim
69 -         r1 = alpha - cont * (alpha / itermax);
70 -         r2 = (2 * pi) * rand();
71 -         r3 = 2 * rand();
72 -         r4 = rand();
73 -         if r4 < 0.5
74 -             X(i,j) = Xi(i,j) + (r1 * sin(r2) * abs(r3 * mejorposXig(1,j)-Xi(i,j)));
75 -         else
76 -             X(i,j) = Xi(i,j) + (r1 * cos(r2) * abs(r3 * mejorposXig(1,j)-Xi(i,j)));
77 -         end
78 -     end
79 - end
80 - Xi=X;
81 -

```

Figura 64. Método de avance SCA

- **Restricción del Espacio de Búsqueda de la Nueva Posición Creada**

Como ya se mencionó la función de la restricción, esta sirve para no dejar desbordar las nuevas partículas creadas dentro de los límites del espacio de búsqueda.

```

83 - for i=1:P
84 -     for j=1:Dim
85 -         if ((Xi(i,j)<Dmin(1,j)) || (Xi(i,j)>Dmax(1,j)))
86 -             Xi(i,j)=Dmin(1,j)+(Dmax(1,j)-Dmin(1,j))*rand;
87 -         end
88 -     end
89 - end

```

Figura 65. Restricción del espacio de búsqueda SCA

Finalmente, el código termina con un  $cont + 1$  y se aplica un criterio de parada donde la condición sea  $cont > itermax$ , cuando cumpla este condicional se cierra el ciclo *while*, en la figura 66 se presenta el diagrama de flujo del método SCA.

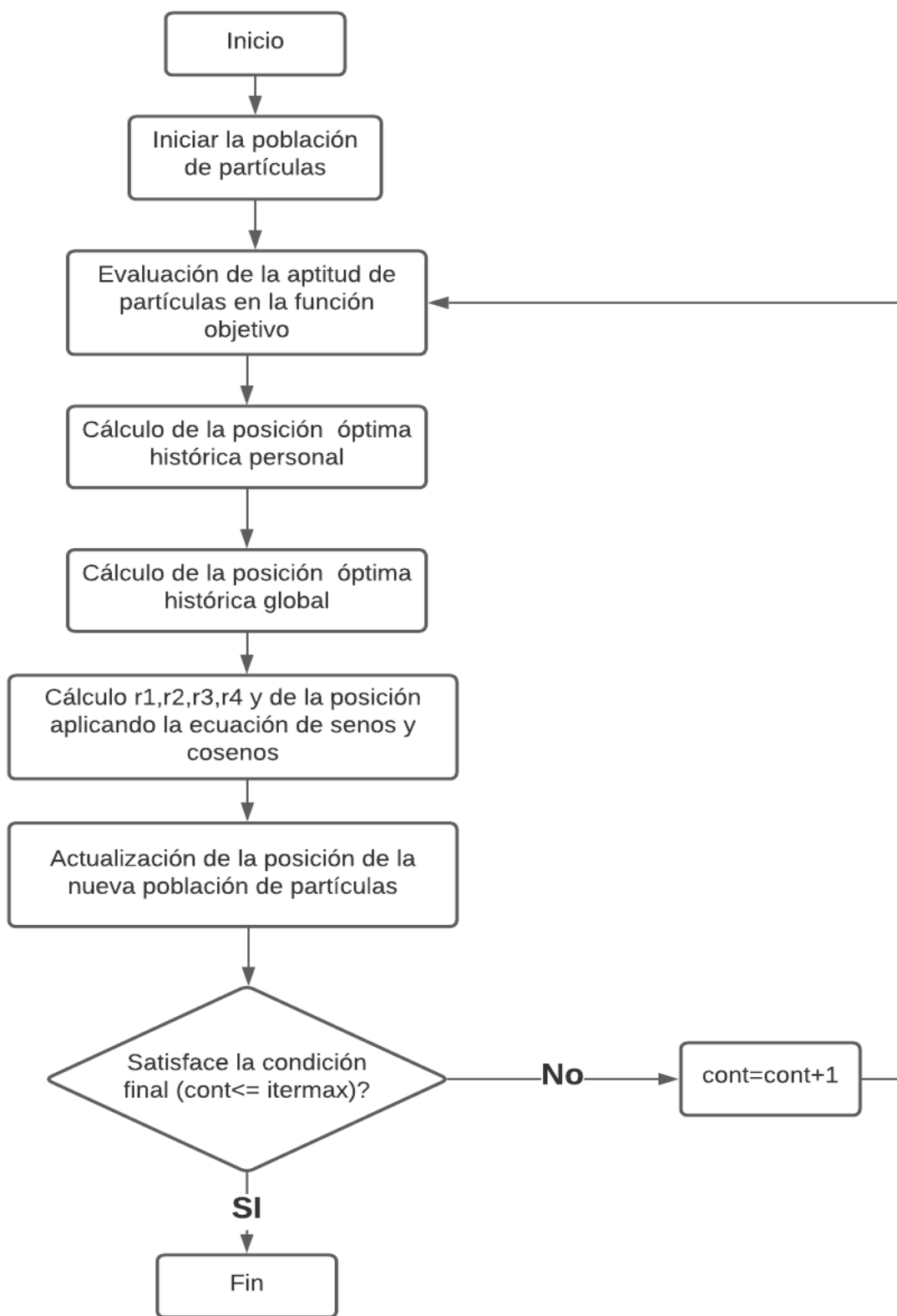


Figura 66. Diagrama de flujo del SCA

Se realizaron diversas pruebas en los métodos de optimización PSO y SCA, para hallar una aproximación a la mejor solución, además observar si el procedimiento de optimización disminuía las pérdidas en comparación con el procedimiento realizado en el capítulo 3. Al realizar varias veces el procedimiento de estos dos métodos, se observa que emplean un tiempo considerable dependiendo del ingreso de las partículas de búsqueda y el número de iteraciones, entre mayor sea la cantidad de estos dos factores, mayor es el tiempo de respuesta. Debido a que estas técnicas no especifican el valor exacto a ingresar de los individuos e iteraciones, se establecieron los valores las partículas en 30 y el número de iteraciones en 100. Al encontrar la mejor aproximación a la solución del problema, se evidencia en la tabla 11 el tiempo empleado en que Matlab arroja el resultado, en la tabla 12 se presenta el tamaño y combinación ideal de la potencia que los generadores tienen que inyectar al sistema.

Tabla 11. Tiempo de respuesta del PSO

Segundos	Minutos	Horas	Días
203633.020	3393.883	56.564	2.356

Tabla 12. Configuración de potencia optima

Generador 1	Generador 2	Generador 3
701k W	490 kW	1173.9k W

En la tabla 13 se observan los datos de los valores de tensión hora a hora arrojados de la mejor solución optimizada. La figura 67 muestra el comportamiento del voltaje en un día.

Tabla 13. Datos de tensión con GD optimizada

Tensión con GD Optimizada	
Hora	Tensión
1	0.971
2	0.979
3	0.980
4	0.984

5	0.983
6	0.978
7	0.971
8	0.962
9	0.959
10	0.969
11	0.975
12	0.979
13	0.977
14	0.981
15	0.976
16	0.967
17	0.944
18	0.927
19	0.918
20	0.904
21	0.904
22	0.914
23	0.931
24	0.948

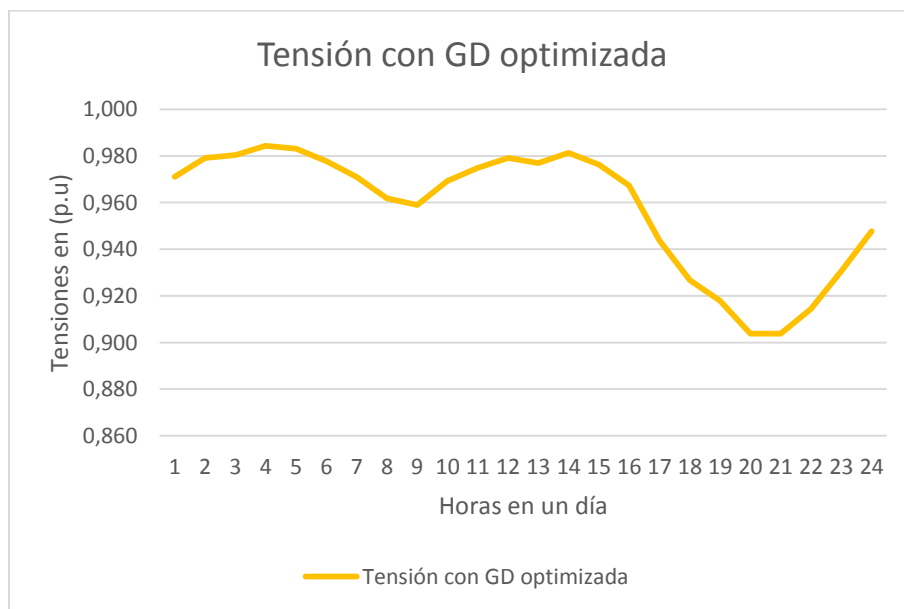


Figura 67. Grafica del voltaje con GD optimizada

En la tabla 14 se observan los datos de los valores de corriente hora a hora arrojados de la mejor solución optimizada. La figura 68 muestra el comportamiento de la corriente en un día.

Tabla 14. Datos de corriente con GD optimizada

Corriente con GD Optimizado	
Hora	Corriente
1	111.884
2	80.993
3	75.873
4	60.560
5	65.656
6	86.121
7	111.875
8	154.535
9	194.692
10	196.787
11	204.620
12	206.045
13	218.330
14	190.669
15	206.649
16	227.597
17	270.425
18	304.629
19	316.035
20	365.252
21	365.252
22	326.092
23	265.634
24	201.074

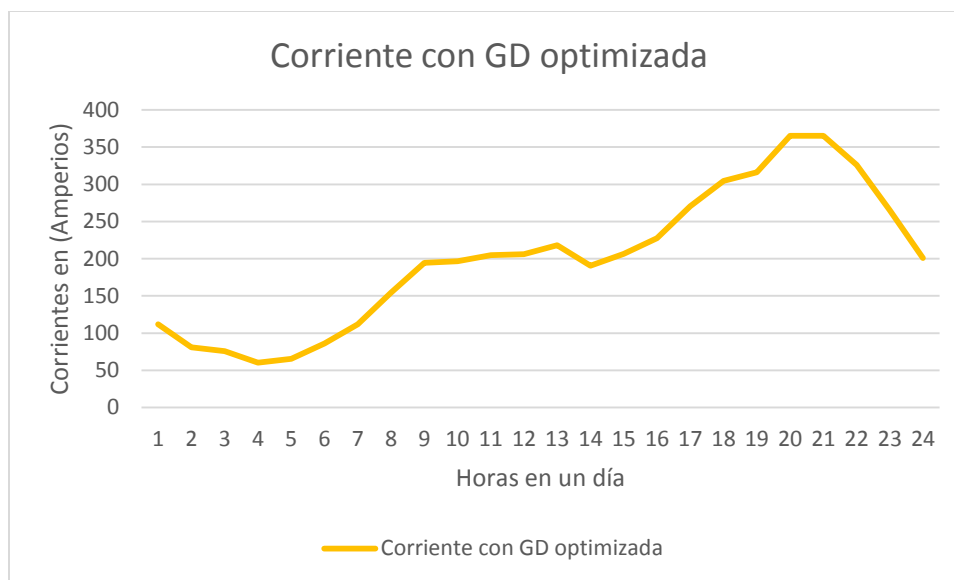


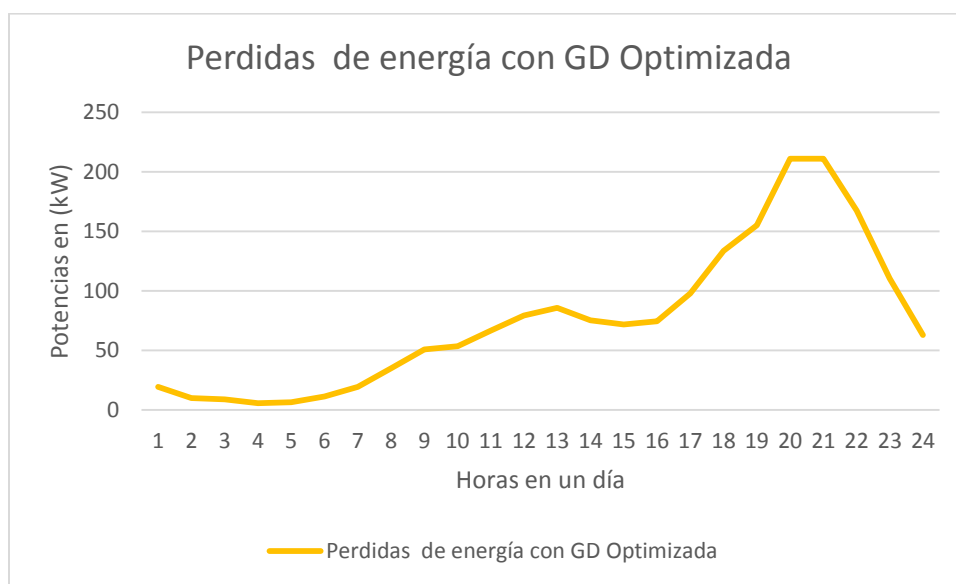
Figura 68. Grafica de la corriente con GD optimizada

En la tabla 15 se observan los datos de los valores de pérdidas de energía hora a hora arrojados de la mejor solución optimizada. La figura 69 muestra el comportamiento de las pérdidas de energía en un día.

Tabla 15. Datos de Energía de Energía con GD Optimizada

Tiempo	Pérdidas de energía con GD Optimizada
1	19.331
2	10.103
3	8.862
4	5.638
5	6.630
6	11.428
7	19.327
8	35.065
9	50.781
10	53.688
11	66.624
12	79.327
13	85.909
14	75.347
15	71.929
16	74.431

17	98.043
18	133.759
19	155.195
20	210.979
21	210.979
22	167.518
23	110.515
24	62.943



*Figura 69. Grafica de las pérdidas de energía con GD optimizada*



## 5. Análisis e Interpretación de los Resultados

En el presente capítulo se hace un análisis de la respuesta obtenida de los flujos de potencia con y sin generadores distribuidos por medio del programa especializado Matlab, con la aplicación de métodos de optimización. Se procede a realizar la comparación de las gráficas tanto de tensiones, corrientes y pérdidas hora a hora, así como las pérdidas totales del sistema. Así mismo, se realiza una comparación con los métodos de optimización PSO y SCA, verificando cuál de ellos presenta un tiempo de respuesta menor y comparando el más preciso (López-Lezama et al., 2015).

Para realizar dicha verificación de los algoritmos de optimización, se procede a ejecutar el código de ambos métodos treinta veces con tres partículas y tres iteraciones, promediando el resultado de pérdidas de energía y el tiempo de ejecución de este, con el fin de conocer cuál de los algoritmos se adapta mejor al presente trabajo de investigación. Estos resultados se muestran en la tabla 16.

Tabla 16. Promedio del tiempo de respuesta y pérdidas de energía

Algoritmos	PSO		SCA	
	Tiempo en Segundos	Pérdidas de energía	Tiempo en Segundos	Pérdidas de energía
1	791.214	1892.5	600.244	1948.2
2	576.477	1839.3	599.062	1851.1
3	574.489	1832.0	614.880	1851.1
4	575.908	1833.0	594.209	1872.9
5	568.201	1828.3	610.697	1824.7
6	570.110	1891.6	604.779	1876.1
7	575.736	1861.2	597.808	1947.0
8	572.626	1827.7	560.295	1827.9
9	641.472	1832.6	601.805	1833.2
10	687.717	1828.0	597.418	1842.5
11	807.785	1831.2	603.904	1902.9
12	985.747	1834.1	614.344	1841.8
13	1009.331	1854.3	574894	1837.5
14	574.030	1834.7	617.835	1839.5
15	585.617	2045.9	623.614	1829.3

16	576.106	1831.0	606.216	1830.4
17	576.353	1826.2	618.851	1880.4
18	572.076	1826.2	607.696	1826.7
19	598.576	1892.5	609.935	1847.3
20	754.458	1839.3	601.315	1856.4
21	665.536	1832.0	595.323	1847.1
22	605.724	1833.5	595.185	1824.7
23	521.320	1828.3	595.391	1855.0
24	581.903	1891.6	600.099	1850.7
25	591.207	1861.2	602.927	1860.0
26	579.616	1827.7	619.547	1857.2
27	594.946	1832.6	623.209	1962.3
28	586.899	1828.0	621.943	1841.4
29	592.498	1831.2	622.914	1854.0
30	585.593	1834.1	619.569	1995.5
Promedio	635.976	1849.4	605.197	1863.8

Una vez realizadas las iteraciones, a pesar de que no existen diferencias significativas a la hora de comprar las respuestas y el tiempo de procesamiento, se observa que el PSO se adapta mejor al problema, arrojando respuestas cercanas a la óptima hallada en el capítulo 4.

De acuerdo con lo mencionando en el capítulo 4, se puede comprobar que al ingresar menos partículas al código, el tiempo de respuesta es mucho menor y comparando los algoritmos de optimización, el SCA tiene menor tiempo de respuesta.

Teniendo en cuenta los valores de la tabla 16, se realizó una comparación entre el valor obtenido de las pérdidas totales del sistema tanto el realizado manualmente, así como el optimizado. El objetivo de la tabla 17, fue conocer cuántos de los valores obtenidos de los dos métodos de optimización fue inferior al de pérdida de GD manual, de igual forma con el menor valor obtenido. De lo anterior se determinó que el algoritmo PSO tuvo un mayor número de iteraciones que fueron inferiores al GD manual en comparación con el SCA, a su vez ninguno tuvo un valor inferior a las pérdidas totales con GD optimizado.

Tabla 17. Comparación de los métodos PSO y SCA

Comparación de las pérdidas totales de	Resultados Menores que las Pérdidas Totales con GD Manual	Resultados Menores que las Pérdidas Totales con GD Optimizado
--	---	---

potencia GD Manual con Optimizada	PSO	SCA	PSO	SCA
	FALSO	FALSO	FALSO	FALSO
FALSO	FALSO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
VERDADERO	VERDADERO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
VERDADERO	VERDADERO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
FALSO	VERDADERO	FALSO	FALSO	
VERDADERO	VERDADERO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
VERDADERO	VERDADERO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
FALSO	VERDADERO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
VERDADERO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
FALSO	FALSO	FALSO	FALSO	
Cuantos cumplen	15	6	0	0
Porcentaje	50%	20%	0%	0%

### 5.1 Resultados de las Corrientes con GD Optimizada y sin GD.

A continuación, se muestra el efecto que tiene la corriente al incluir GD optimizada y se compara sin GD, se observa que a medida que incrementa la radiación solar, se genera un

cambio en el resultado de la corriente de la curva de demanda con flujos óptimos de potencia, se evidencia que en los momentos que se inyectan los flujos óptimos, existe un descongestionamiento en las líneas de transporte de energía, de esta manera se reduce la saturación de la red en los periodos donde exigen más las cargas y mejora la calidad del sistema eléctrico para los usuarios.

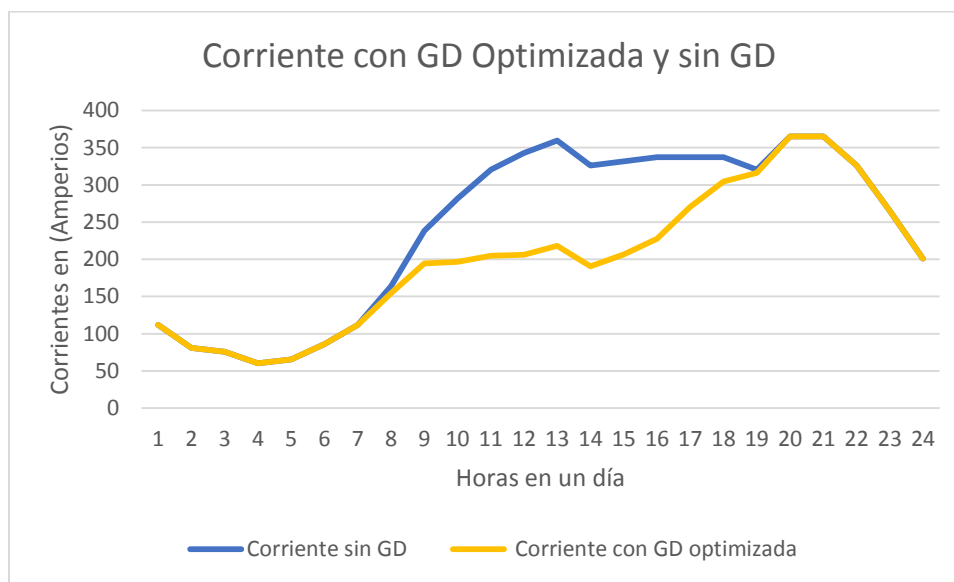


Figura 70. Grafica la corriente con GD optimizada y sin GD

Tabla 18. Comparación de la corriente sin GD y aplicando flujos óptimos

Hora	Corrientes sin GD	Corrientes con GD optimizada	Resultados de corrientes con GD menores a sin GD	Resultados de corrientes con GD mayores a sin GD
1	111.884	111.884	FALSO	FALSO
2	80.993	80.993	FALSO	FALSO
3	75.873	75.873	FALSO	FALSO
4	60.560	60.560	FALSO	FALSO
5	65.656	65.656	FALSO	FALSO
6	86.121	86.121	FALSO	FALSO
7	111.884	111.875	VERDADERO	FALSO
8	164.040	154.535	VERDADERO	FALSO

9	238.564	194.692	VERDADERO	FALSO
10	281.997	196.787	VERDADERO	FALSO
11	320.542	204.620	VERDADERO	FALSO
12	342.807	206.045	VERDADERO	FALSO
13	359.623	218.330	VERDADERO	FALSO
14	326.092	190.669	VERDADERO	FALSO
15	331.652	206.649	VERDADERO	FALSO
16	337.224	227.597	VERDADERO	FALSO
17	337.224	270.425	VERDADERO	FALSO
18	337.224	304.629	VERDADERO	FALSO
19	320.542	316.035	VERDADERO	FALSO
20	365.252	365.252	FALSO	FALSO
21	365.252	365.252	FALSO	FALSO
22	326.092	326.092	FALSO	FALSO
23	265.634	265.634	FALSO	FALSO
24	201.074	201.074	FALSO	FALSO
Promedio	242.242	200.303	Disminución	17,313%

## 5.2 Resultados de Voltajes con GD Optimizado y sin GD

Al comparar los resultados de los voltajes en *p.u* con GD optimizada y sin GD de la figura 71, muestra que en las horas 7 a.m. y 7p.m existe un aumento en el voltaje con la aplicación de flujos óptimos de potencia. Esto se debe a que en las redes de distribución que no tienen generación distribuida, normalmente los operadores de red utilizan reguladores para compensar los efectos que generan las caídas de tensiones y al incluir los generadores en las horas de operación, estos entran a compensar los efectos de los niveles de tensión baja y de esta manera mejora el perfil de voltaje manteniéndose dentro de los límites permitidos por la norma.

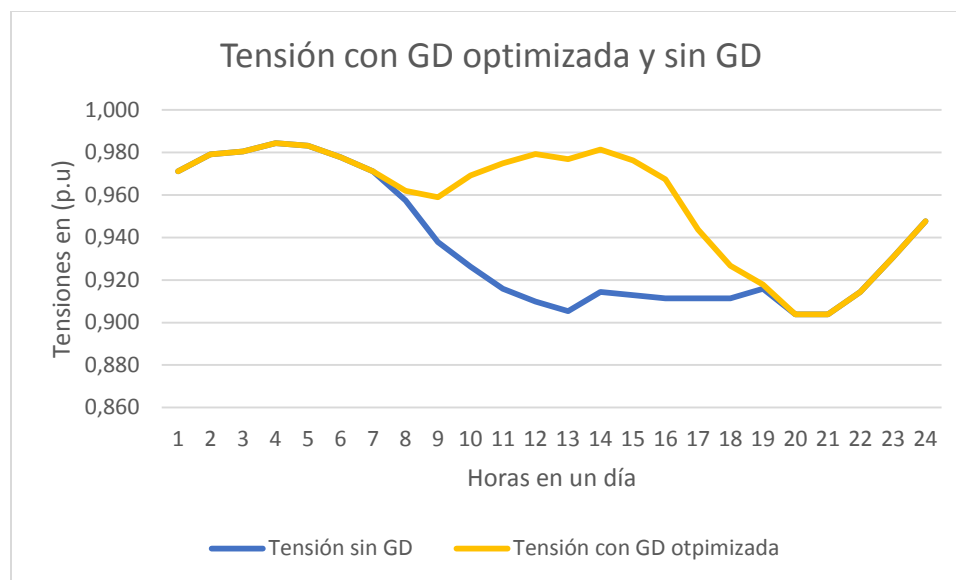


Figura 71. Grafica del voltaje con GD optimizada y sin GD

Tabla 19. Comparación de los voltajes sin GD y aplicando flujos óptimos

Hora	Voltajes sin GD	Voltajes con GD optimizada	Resultados de voltajes con GD menores a sin GD	Resultados de voltajes con GD mayores a sin GD
1	0.971	0.971	FALSO	FALSO
2	0.979	0.979	FALSO	FALSO
3	0.980	0.980	FALSO	FALSO
4	0.984	0.984	FALSO	FALSO
5	0.983	0.983	FALSO	FALSO
6	0.978	0.978	FALSO	FALSO
7	0.971	0.971	FALSO	FALSO
8	0.957	0.962	FALSO	VERDADERO
9	0.938	0.959	FALSO	VERDADERO
10	0.926	0.969	FALSO	VERDADERO
11	0.916	0.975	FALSO	VERDADERO
12	0.910	0.979	FALSO	VERDADERO
13	0.905	0.977	FALSO	VERDADERO
14	0.914	0.981	FALSO	VERDADERO
15	0.913	0.976	FALSO	VERDADERO
16	0.911	0.967	FALSO	VERDADERO
17	0.911	0.944	FALSO	VERDADERO
18	0.911	0.927	FALSO	VERDADERO
19	0.916	0.918	FALSO	VERDADERO

20	0.904	0.904	FALSO	FALSO
21	0.904	0.904	FALSO	FALSO
22	0.914	0.914	FALSO	FALSO
23	0.931	0.931	FALSO	FALSO
24	0.948	0.948	FALSO	FALSO
Promedio	0.937	0.958	Incremento	2.245%

### 5.3 Resultados de Pérdidas de Energía hora a hora con GD optimizado y sin GD

De acuerdo con los resultados obtenidos en Matlab, la figura 72 ilustra la gráfica de la variación de pérdidas de energía hora a hora con GD optimizada y sin GD en un día de operación, al presentarse la GD optimizada ubicada en los nodos 12, 15 y 31, se aprecia que existe una disminución considerable de pérdidas de energía del 32.23%, la cual regula los perfiles de tensión de la red y permite librar la líneas sobrecargadas ocasionando una reducción del efecto Joule, esto se debe a la variación de potencia inyectada por los generadores o flujos óptimos, el cual su función es alivianar la red hora a hora cada vez que aumente la demanda dentro del rango de radiación solar.

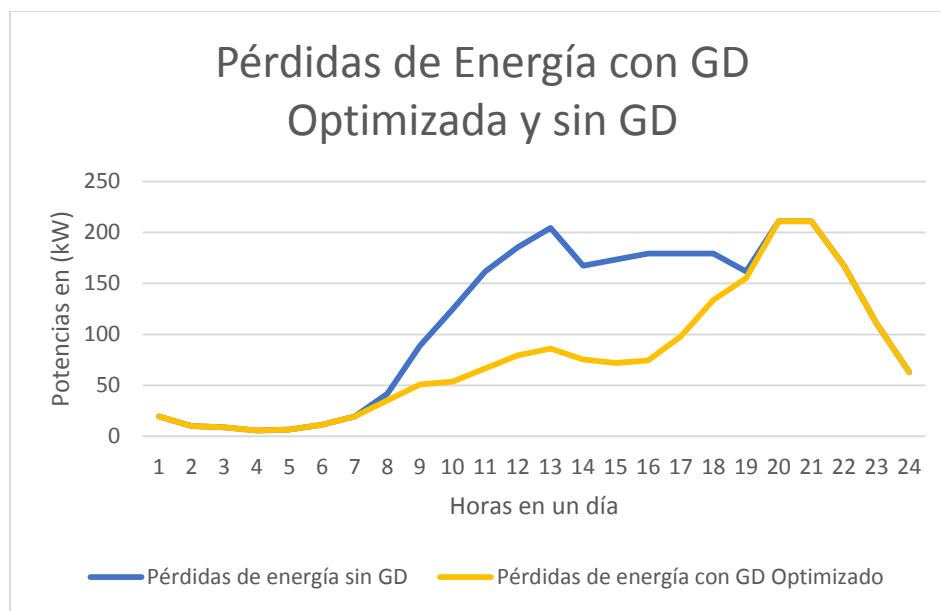


Figura 72. Grafica de las paridas de potencia con GD optimizada y sin GD

Tabla 20. Comparación de las pérdidas de enetgía sin GD y aplicando flujos óptimos

Tiempo	Pérdidas sin GD	Pérdidas con GD Optimizado	Resultados de las Pérdidas con GD menores a sin GD	Resultados de las Pérdidas con GD mayores a sin GD
1	19.331	19.331	FALSO	FALSO
2	10.103	10.103	FALSO	FALSO
3	8.862	8.862	FALSO	FALSO
4	5.638	5.638	FALSO	FALSO
5	6.630	6.630	FALSO	FALSO
6	11.428	11.428	FALSO	FALSO
7	19.331	19.327	VERDADERO	FALSO
8	41.750	35.065	VERDADERO	FALSO
9	88.911	50.781	VERDADERO	FALSO
10	124.744	53.688	VERDADERO	FALSO
11	161.777	66.624	VERDADERO	FALSO
12	185.434	79.327	VERDADERO	FALSO
13	204.412	85.909	VERDADERO	FALSO
14	167.518	75.347	VERDADERO	FALSO
15	173.374	71.929	VERDADERO	FALSO
16	179.346	74.431	VERDADERO	FALSO
17	179.346	98.043	VERDADERO	FALSO
18	179.346	133.759	VERDADERO	FALSO
19	161.777	155.195	VERDADERO	FALSO



20	210.979	210.979	FALSO	FALSO
21	210.979	210.979	FALSO	FALSO
22	167.518	167.518	FALSO	FALSO
23	110.515	110.515	FALSO	FALSO
24	62.943	62.943	FALSO	FALSO
Promedio	112.166	76.014	Disminuyó	32.230%

#### 5.4 Resultados de las pérdidas totales con GD Optimizado y sin GD

Como se mencionó en el documento, este proyecto tiene como finalidad reducir las pérdidas de energía con la aplicación de flujos óptimos programados en Matlab, en la tabla 18 se ilustran los valores de los resultados sin GD y con GD optimizada y en la figura 73, se presenta el diagrama de barras donde se evidencia notablemente la reducción de pérdidas de energía.

Tabla 21. Pérdidas totales con GD optimizada y sin GD

Pérdidas totales	Pérdidas sin GD	Pérdidas con GD optimizada
	2691.988kW	1824.348kW

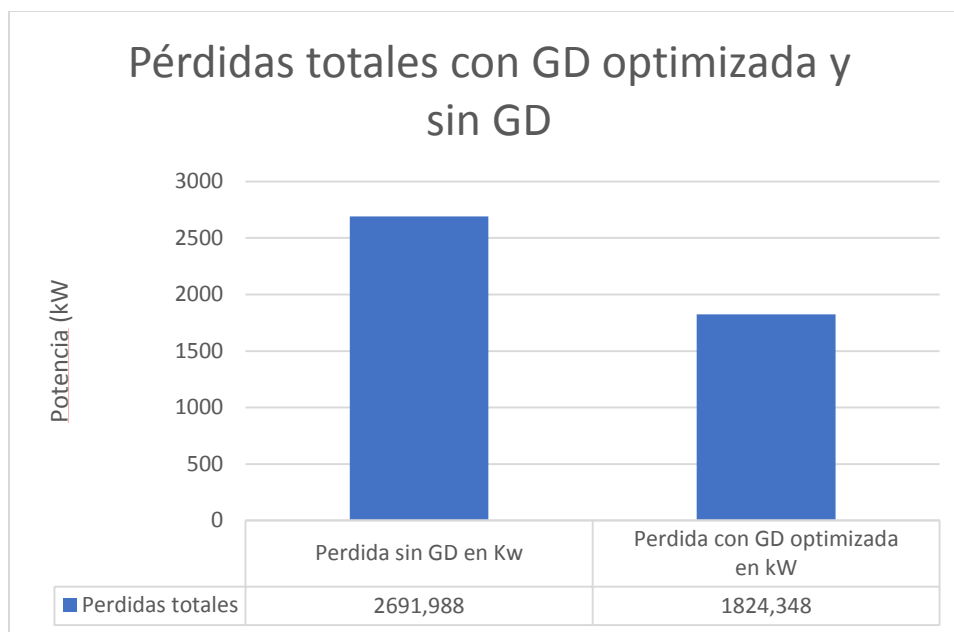


Figura 73. Diagrama de barras de las pérdidas totales de potencia con GD optimizada y sin GD.

## 6. Conclusiones

Este proyecto permitió adquirir nuevos conocimientos en el área de análisis de sistemas eléctricos de potencia, su enfoque estuvo en las redes de distribución. Para el análisis, es necesario tener una amplia información sobre los parámetros de la red, donde los más relevantes son los voltajes en los nodos, corrientes en las líneas, potencias individuales y totales, con estos datos se pueden tomar diversas decisiones de mejora en el sistema eléctrico. Un fragmento en esta tesis fue conocer los flujos de carga, donde se estudiaron diversos tipos de métodos, en comparación con los métodos clásicos se evidencio que aproximaciones sucesivas y barrido iterativo son los que presentan mejor precisión en su respuesta. Una vez adquiridos todos estos conocimientos se determinó que estos métodos sirven para cualquier topología que pueda presentar la red, sin importar el número de nodos, de esta manera se ejecutó la evaluación de las pérdidas de energía en un sistema de distribución en corriente alterna.

Los algoritmos de optimización PSO y SCA se basan en crear partículas o individuos aleatoriamente dentro de un espacio de búsqueda para asegurar la exploración del mismo, las posiciones de cada partícula se actualizan en relación con la mejor solución “solución global” hasta llegar a un punto de convergencia, se demostró el rendimiento del método PSO en la resolución de un problema en un sistema de prueba de 33 nodos con espacios de búsqueda restringidos, se observó que la mejor configuración para inyectar flujos óptimos al sistema con generadores fotovoltaicos es de  $G1 = 701\text{kW}$ ,  $G2 = 490\text{kW}$ ,  $G3 = 1173.9\text{kW}$  con un tiempo de respuesta de 203633.020 segundos.

Al estudiar el comportamiento que tiene la inclusión de generadores fotovoltaicos en una red de distribución aplicando métodos de optimización, se presentan los flujos óptimos en el sistema. La función que cumple este nuevo parámetro es alivianar la saturación en la curva de carga en las horas pico donde la demanda es más exigente, de esta forma se puede lograr una reducción significativa de las pérdidas de energía. A la hora de realizar la simulación de estos métodos, el tiempo de respuesta es muy elevado dependiendo de la cantidad de número de partículas e iteraciones que se ingresen en estos algoritmos, por lo tanto, es indispensable tener un buen equipo de cómputo que se mantenga conectado a la red eléctrica, de esta forma se asegura corriendo el código el cual puede llegar a durar días en dar una respuesta, garantizando una buena solución al reducir las pérdidas de potencia de cualquier sistema de distribución radial.

Mediante los resultados de los algoritmos de optimización, se evidencio una disminución en la corriente de 17.313% y una mejora en el nivel de tensión de 2.245% en un día de operación, la reducción en las pérdidas totales de potencia aplicando flujos óptimos fueron del 32.23% en comparación a las pérdidas de energía sin generación distribuida, por lo tanto, se demuestra que el modelo matemático propuesto y programado en Matlab, cumple con el objetivo planteado de minimizar las pérdidas de energía.

Al comparar el método PSO y SCA, se comprobó por medio de iteraciones el tiempo de respuesta de cada algoritmo y se evidencio que el método SCA tiene un porcentaje menor de 4.8% en el tiempo de respuesta a comparación con el PSO, el tiempo promedio de estos algoritmos fue superior a los diez minutos. En el caso del PSO, el 50% de las respuestas de la

tabla 17 fueron inferiores a la pérdida de potencia total obtenida en la GD manual y en el SCA solo el 20%, sin embargo, ninguna respuesta fue menor a la reducción de pérdidas de energía óptima global.

## **7. Recomendaciones**

Este proyecto se realizó con datos de la ciudad de Medellín proporcionados por el ITM, se sugiere que en un futuro el proyecto sea aplicado con los datos de la ciudad de Cúcuta, así como la implementación de los diferentes métodos de optimización que existen en la actualidad, con el fin de encontrar cuál de estos se adapta mejor al problema de flujos óptimos de potencia.

Para futuros proyectos se debe analizar las condiciones de una red eléctrica en Norte de Santander, para aplicar la ubicación de los generadores en un sistema real y no uno de prueba.

## 8. Referencias

- Clerc, M., & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58–73. <https://doi.org/10.1109/4235.985692>
- Cordero, F., Del Valle, T., & Morales, A. (2019). USOS DE LA OPTIMIZACIÓN DE INGENIEROS EN FORMACIÓN: EL ROL DE LA INGENIERÍA MECATRÓNICA Y DE LA OBRA DE LAGRANGE. *Revista Latinoamericana de Investigación En Matemática Educativa*.
- Eberhart, R., & Kennedy, J. (1995). A new optimizer using particle swarm theory. In *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science* (pp. 39–43). <https://doi.org/10.1109/MHS.1995.494215>
- ELECTRONICA R2M, S. A. D. C. V. (n.d.). *Lineas de Distribución Eléctrica Primaria*. <https://www.electronicar2m.com/mantenimiento-y-construccion-de-lineas-de-distribucion-electrica-primaria/>
- Gönen, T. (1988). *ELECTRIC POWER TRANSMISSION SYSTEM ENGINEERING*. <https://www.dropbox.com/sh/1sst84x6ky7wg29/w6pH6G9rDR>
- Grainger, J. J., & Stevenson Jr, W. D. (1996). *Analisis de sistemas de potencia*. <https://docs.google.com/file/d/0ByA6qgvaKBUNbWNxR28tN25Zd0k/edit?resourcekey=0-Vsmf-ufbbogh2KEO1G-7Q>
- Grisales-Noreña, L. F., González Montoya, D., & Ramos-Paja, C. A. (2018a). Optimal Sizing and Location of DistributedGenerators Based on PBIL and PSO Techniques. *Energies*. <https://doi.org/10.3390>
- Grisales-Noreña, L. F., González Montoya, D., & Ramos-Paja, C. A. (2018b). Optimal Sizing and Location of DistributedGenerators Based on PBIL and PSO Techniques. *Energies*. <https://doi.org/10.3390>
- Grisales Noreña, L. F., & Giraldo Montoya, O. D. (2020). An exact MINLP model for optimal location and sizing of DGs in distribution networks: A general algebraic modeling system approach. *Ain Shams Engineering Journal*, 11(2), 409–418.
- Hafez, A. I., Zawbaa, H. M., Emary, E., & Hassanien, A. E. (2016). Sine cosine optimization algorithm for feature selection. In *2016 International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)* (pp. 1–5). <https://doi.org/10.1109/INISTA.2016.7571853>
- Herrera Briñez, M. C. (2021). *Formulacion y comparacion de cinco metodos para el analisis de flujo de potencia en sistemas de distribucion AC en el software MATLAB*. Universidad Distrital Francisco Jose de Caldas.
- Historicos de demanda XM*. (n.d.). <https://www.xm.com.co/consumo/historicos-de-demanda#:~:text=El sábadomantiene un consumo,consumo de energía eléctrica similar.>
- Instituto Colombiano de Normas Técnicas y Certificación (ICONTEC). (1998). *CÓDIGO ELÉCTRICO COLOMBIANO*.

[https://www.armada.mil.co/sites/default/files/normograma\\_arc/mantenimiento1/NTC2050.pdf](https://www.armada.mil.co/sites/default/files/normograma_arc/mantenimiento1/NTC2050.pdf)

- Jatmiko, W., Sekiyama, K., & Fukuda, T. (2007). A pso-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement. *IEEE Computational Intelligence Magazine*, 2(2), 37–51. <https://doi.org/10.1109/MCI.2007.353419>
- López-Lezama, J. M., Villada, F., & Muñoz, N. (2015). Metaheurísticas Aplicadas a la Ubicación y Dimensionamiento Óptimo de Generación Distribuida en Sistemas de Distribución de Energía Eléctrica. *Información Tecnológica*, 145–158. <https://doi.org/10.4067/S0718-07642015000400018>
- López Lezama, J. M., & Gallego Pareja, L. A. (2008). Flujo de potencia óptimo usando el método del gradiente para reducción de pérdidas en sistemas de potencia,”. *Ingeniería y Ciencias*, 4(7), 15.
- Maynard Smith, J. (1978). OPTIMIZATION THEORY IN EVOLUTION. *Annual Reviews*, 26.
- Meneu Gaya, R., Pérez Salamero González, J. M., & Ventura Marco, M. (1999). *Fundamentos de Optimización Matemática en Economía*. <https://doi.org/10.13140/RG.2.1.2605.8402>
- Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowledge-Based Systems*, 96, 120–133. <https://doi.org/https://doi.org/10.1016/j.knsys.2015.12.022>
- Mosquera, F., & Carrión, D. (2019). Localización óptima de plantas virtuales de generación en sistemas eléctricos de potencia basados en flujos óptimos de potencia. *UTP*, 13.
- Pérez Peña, R. (2019). *Introducción A LOS MODELOS DE OPTIMIZACIÓN*.
- Ramos, A., Sanchez, P., Ferrer, J. M., & Linares, P. (2010). *Modelos Matemáticos de Optimización*.
- Schmieg, M. (1985). *Digsilent*.
- Schweickardt, G., & Miranda, V. (2010). Metaheurística FEPSO aplicada a problemas de Optimización Combinatoria: Balance de Fases en Sistemas de Distribución Eléctrica. *Redalyc.Org*, 32.
- Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)* (pp. 69–73). <https://doi.org/10.1109/ICEC.1998.699146>
- Velasquez Rodríguez, O. S. (2019). *Flujo de potencia óptimo en redes de corriente continua empleando un algoritmo de optimización basado en agujeros negros*.
- Wang, D., Tan, D., & Liu, L. (2017). Particle swarm optimization algorithm: an overview. *Soft Computing SP*.
- Wang, G.-G., Gandomi, A. H., & Alavi, A. H. (2014). An effective krill herd algorithm with migration operator in biogeography-based optimization. *Elsevier*, 38(9–10), 9.



<https://doi.org/https://doi.org/10.1016/j.apm.2013.10.052>

## 9. Anexos

### Código Barrido Iterativo

```

Sbase= 1000;%kVA
Vbase= 12.66; %kV
%% Lineas [ i j R[ohm] X[ohm]]
Lineas= [ 1 2 0.0922 0.0477;
          2 3 0.4930 0.2511;
          3 4 0.3660 0.1864;
          4 5 0.3811 0.1941;
          5 6 0.8190 0.7070;
          6 7 0.1872 0.6188;
          7 8 1.7114 1.2351;
          8 9 1.0300 0.7400;
          9 10 1.0400 0.7400;
          10 11 0.1966 0.0650;
          11 12 0.3744 0.1238;
          12 13 1.4680 1.1550;
          13 14 0.5416 0.7129;
          14 15 0.5910 0.5260;
          15 16 0.7463 0.5450;
          16 17 1.2890 1.7210;
          17 18 0.7320 0.5740;
          2 19 0.1640 0.1565;
          19 20 1.5042 1.3554;
          20 21 0.4095 0.4784;
          21 22 0.7089 0.9373;
          3 23 0.4512 0.3083;
          23 24 0.8980 0.7091;
          24 25 0.8960 0.7011;
          6 26 0.2030 0.1034;
          26 27 0.2842 0.1447;
          27 28 1.0590 0.9337;
          28 29 0.8042 0.7006;
          29 30 0.5075 0.2585;
          30 31 0.9744 0.9630;
          31 32 0.3105 0.3619;
          32 33 0.3410 0.5302];
%% Nodo [ i Tipo V P[kW] Q[kW]]
Nodos= [ 1 0 1 0 0
          2 1 1 100 60;
          3 1 1 90 40;
          4 1 1 120 80;

```

```

5 1 1 60 30;
6 1 1 60 20;
7 1 1 200 100;
8 1 1 200 100;
9 1 1 60 20;
10 1 1 60 20;
11 1 1 45 30;
12 1 1 60 35;
13 1 1 60 35;
14 1 1 120 80;
15 1 1 60 10;
16 1 1 60 20;
17 1 1 60 20;
18 1 1 90 40;
19 1 1 90 40;
20 1 1 90 40;
21 1 1 90 40;
22 1 1 90 40;
23 1 1 90 50;
24 1 1 420 200;
25 1 1 420 200;
26 1 1 60 25;
27 1 1 60 25;
28 1 1 60 20;
29 1 1 120 70;
30 1 1 200 600;
31 1 1 150 70;
32 1 1 210 100;
33 1 1 60 40];

%% Cambio a por Unidad
Zbase= ((1000*Vbase)^2)/(1000*Sbase);
Lineas(:,3:4)=Lineas(:,3:4)/Zbase;
Nodos(:,4:5)=Nodos(:,4:5)/Sbase;
%% Formación Matriz de Incidencia (Matriz A)
NN=size(Nodos,1);
NL=size(Lineas,1);
A=zeros(NN,NL);
Zr=zeros(NL,NL);
for i=1:NL
    Ni=Lineas(i,1); Nj=Lineas(i,2);
    A(Ni,i)=1;
    A(Nj,i)=-1;
    Zr(i,i)=Lineas(i,3)+1j*Lineas(i,4);
end
%% Barrido iterativo
tol=1e-6;

```

```

Ao=A(1,:);
Ad=A(2:end,:);
Ydd=Ad*(inv(Zr))*Ad.';
IYdd=inv(Ydd);
Ydg=Ad*(inv(Zr))*Ao.';
Vdo=Nodos(2:end,3);
Vo=Nodos(1,3);
Sd=Nodos(2:end,4)+1j*Nodos(2:end,5);
itermax=100;
for t=1:itermax
    Vd=-IYdd*((inv(diag(conj(Vdo))))*conj(Sd)+Ydg*Vo);
    if max(abs((abs(Vd)-abs(Vdo))))<=tol
        V=[Vo;Vd];
        break
    else
        Vdo=Vd;
    end
end
%%Perdidas Sloss
Vr=Ao.*Vo+Ad.*Vd;
Ir=(inv(Zr))*Vr;
Sloss=(sum(Zr*(abs(Ir)).^2))*Sbase;
repmax=10000;
tic
ti = toc;

```

### Código Aproximaciones Sucesivas

```

Sbase = 1000; %KVA
Vbase = 12.66; %KV
%% Lineas [ i j R[ohm] X[ohm]]
Lineas= [ 1 2 0.0922 0.0477;
          2 3 0.4930 0.2511;
          3 4 0.3660 0.1864;
          4 5 0.3811 0.1941;
          5 6 0.8190 0.7070;
          6 7 0.1872 0.6188;
          7 8 1.7114 1.2351;
          8 9 1.0300 0.7400;
          9 10 1.0400 0.7400;
          10 11 0.1966 0.0650;
          11 12 0.3744 0.1238;
          12 13 1.4680 1.1550;
          13 14 0.5416 0.7129;
          14 15 0.5910 0.5260;

```

```

15 16 0.7463 0.5450;
16 17 1.2890 1.7210;
17 18 0.7320 0.5740;
2 19 0.1640 0.1565;
19 20 1.5042 1.3554;
20 21 0.4095 0.4784;
21 22 0.7089 0.9373;
3 23 0.4512 0.3083;
23 24 0.8980 0.7091;
24 25 0.8960 0.7011;
6 26 0.2030 0.1034;
26 27 0.2842 0.1447;
27 28 1.0590 0.9337;
28 29 0.8042 0.7006;
29 30 0.5075 0.2585;
30 31 0.9744 0.9630;
31 32 0.3105 0.3619;
32 33 0.3410 0.5302];

```

```
%% Nodo [ i Tipo V P[kW] Q[kW]]
```

```

Nodos= [ 1 0 1 0 0
2 1 1 100 60;
3 1 1 90 40;
4 1 1 120 80;
5 1 1 60 30;
6 1 1 60 20;
7 1 1 200 100;
8 1 1 200 100;
9 1 1 60 20;
10 1 1 60 20;
11 1 1 45 30;
12 1 1 60 35;
13 1 1 60 35;
14 1 1 120 80;
15 1 1 60 10;
16 1 1 60 20;
17 1 1 60 20;
18 1 1 90 40;
19 1 1 90 40;
20 1 1 90 40;
21 1 1 90 40;
22 1 1 90 40;
23 1 1 90 50;
24 1 1 420 200;
25 1 1 420 200;
26 1 1 60 25;

```

```

27 1 1 60 25;
28 1 1 60 20;
29 1 1 120 70;
30 1 1 200 600;
31 1 1 150 70;
32 1 1 210 100;
33 1 1 60 40];

%%pasar a pu
Zbase = ((1000*Vbase)^2)/(1000*Sbase);
Lineas(:,3:4) = Lineas(:,3:4)/Zbase;
Nodos(:,4:5) = Nodos(:,4:5)/Sbase;

%Formacion de la Ybus
NN = size(Nodos,1);
NL = size(Lineas,1);

%tamaño de la matriz ybus
Yb = zeros(NN,NN);

for i=1:NL
    Ni = Lineas(i,1); Nj = Lineas(i,2);
    ZL = Lineas(i,3) + 1j*Lineas(i,4);

    %ecuacion de Ybus
    Yb(Ni,Nj) = -1/ZL; Yb(Nj,Ni) = -1/ZL;
    % Diagonal
    Yb(Ni,Ni) = Yb(Ni,Ni) + 1/ZL;
    Yb(Nj,Nj) = Yb(Nj,Nj) + 1/ZL;
end

%%Metodo de aproximaciones sucesivas
tol= 1e-6;
V0 = Nodos(2:end,3); Vg = Nodos(1,3);
Sd = Nodos(2:end,4) + 1j*Nodos(2:end,5);
Ydd= Yb(2:end,2:end) ; Ydg=Yb(2:end,1);
intermax = 100;
IYdd = inv(Ydd);
for t=1:intermax
    V = -IYdd*((inv(diag(conj(V0))))*conj(Sd) + Ydg*Vg );
    if max(abs(abs(V)-abs(V0)))<tol
        Vr= [Vg;V];
        break
    else
        V0 = V;
    end
end
end

```

```

Sloss = sum(Vr.*(conj(Yb)*conj(Vr)))*Sbase;
F=1;
repmax=10000;
tic
ti = toc;

```

### Código Flujo 24 Horas sin Generadores Distribuidos

```

tic
repmax = 10000;
load datos;
for iii=1:24
    factor_de_demanda=parametros(iii,1);
for rep = 1:repmax
    Sbase = 1000;%kVA
    Vbase = 12.66;%kV
    Slack = 1; NB = 33; NL = 32;
    %% Datos [i j R[ohm] X[ohm] P[kW] Q[kVA]]
    Sistema = [1 2 0.0922 0.0477 100 60
                2 3 0.4930 0.2511 90 40
                3 4 0.3660 0.1864 120 80
                4 5 0.3811 0.1941 60 30
                5 6 0.8190 0.7070 60 20
                6 7 0.1872 0.6188 200 100
                7 8 1.7114 1.2351 200 100
                8 9 1.0300 0.7400 60 20
                9 10 1.0400 0.7400 60 20
                10 11 0.1966 0.0650 45 30
                11 12 0.3744 0.1238 60 35
                12 13 1.4680 1.1550 60 35
                13 14 0.5416 0.7129 120 80
                14 15 0.5910 0.5260 60 10
                15 16 0.7463 0.5450 60 20
                16 17 1.2890 1.7210 60 20
                17 18 0.7320 0.5740 90 40
                2 19 0.1640 0.1565 90 40
                19 20 1.5042 1.3554 90 40
                20 21 0.4095 0.4784 90 40
                21 22 0.7089 0.9373 90 40
                3 23 0.4512 0.3083 90 50
                23 24 0.8980 0.7091 420 200
                24 25 0.8900 0.7011 420 200
                6 26 0.2030 0.1034 60 25
                26 27 0.2842 0.1447 60 25
                27 28 1.0590 0.9337 60 20

```

```

28 29 0.8042 0.7006 120 70
29 30 0.5075 0.2585 200 600
30 31 0.9744 0.9630 150 70
31 32 0.3105 0.3619 210 100
32 33 0.3410 0.5302 60 40];

```

```

%ajuste de demanda
%potencia activa
Sistema(:,5)=factor_de_demanda*Sistema(:,5);
%potencia reactiva
Sistema(:,6)=factor_de_demanda*Sistema(:,6);

```

```

Node = [1 0 1 0 0;
2 1 1 100 60;
3 1 1 90 40;
4 1 1 120 80;
5 1 1 60 30;
6 1 1 60 20;
7 1 1 200 100;
8 1 1 200 100;
9 1 1 60 20;
10 1 1 60 20;
11 1 1 45 30;
12 1 1 60 35;
13 1 1 60 35;
14 1 1 120 80;
15 1 1 60 10;
16 1 1 60 20;
17 1 1 60 20;
18 1 1 90 40;
19 1 1 90 40;
20 1 1 90 40;
21 1 1 90 40;
22 1 1 90 40;
23 1 1 90 50;
24 1 1 420 200;
25 1 1 420 200;
26 1 1 60 25;
27 1 1 60 25;
28 1 1 60 20;
29 1 1 120 70;
30 1 1 200 600;
31 1 1 150 70;
32 1 1 210 100;
33 1 1 60 40];

```



```

%ajuste de demanda
%potencia activa
Node(:,4)=factor_de_demanda* Node(:,4);
%potencia reactiva
Node(:,5)=factor_de_demanda* Node(:,5);

%% Pasando a por unidad
Zbase = (1000*Vbase)^2/(1000*Sbase);
Sistema(:,3:4) = Sistema(:,3:4)/Zbase;
Node(:,4:5) = Node(:,4:5)/Sbase;
%% Formación de la matriz GBUS
[Ybus] = formacionYbus(Sistema,NB,NL);
m = 1000;
Ydd = Ybus(2:end,2:end);
S = (Node(2:end,4)+1i*Node(2:end,5));
Ydg = Ybus(2:end,1);
Vg = 1; %% Voltaje asociado al nodo slack
idd = inv(Ydd);
V0 = ones(size(Ydd,2),1);
for k = 1:m
    %% Extracción de los datos para el flujo lineal:
    x = -idd*((diag(conj(V0)))\conj(S) + Ydg*Vg);
    if abs(abs(x) -abs(V0)) <= 1e-10
        V = [Vg;x];
        break
    else
        V0 = x;
    end
end
end

%calculo de pérdidas horarias
Losses(iii,1)= real(V.*conj(Ybus*V))*1000;
%calculo de peores tensiones
voltajes_horario=abs(V);
errores_de_voltaje= abs(voltajes_horario-1);
peor_error=max(errores_de_voltaje);
pos=find(errores_de_voltaje==peor_error);
peor_tension(iii,1)=voltajes_horario(pos,1);
%calculo de peor corriente
I_L=zeros(NL,1);
for eee=1:NL
    node_e=Sistema(eee,1);
    V_e=V(node_e,1);
    node_r=Sistema(eee,2);

```

```

V_r=V(node_r,1);
R_l=Sistema(eee,3);
X_l=Sistema(eee,4);
Z_l=R_l+X_l*1i;
I_L=(V_e-V_r)/Z_l;
I_L=abs(I_L);
I_L_horario(eee,1)=I_L*78.9889;

end
peor_corriente(iii,1)=max(I_L_horario);

end
total_power_loss=sum(Losses(:,1));

```

### Código Flujo 24 Horas con Generación Distribuida sin Optimizar

```

tic
repmax = 10000;
load datos;
GD=[12 15 31]

Inyeccion=[734.2049 456.4527 1187.6830]/Sbase;%se divide por la potencia base
for iii=1:24
    factor_de_demanda=parametros(iii,1);
    factor_de_generacionsolar=parametros(iii,2);
    Inyeccion_variable=Inyeccion*factor_de_generacionsolar;
for rep = 1:repmax
    Sbase = 1000;%kVA
    Vbase = 12.66;%kV
    Slack = 1; NB = 33; NL = 32;
    %% Datos [i j R[ohm] X[ohm] P[kW] Q[kVA]]
    Sistema = [1 2 0.0922 0.0477 100 60
                2 3 0.4930 0.2511 90 40
                3 4 0.3660 0.1864 120 80
                4 5 0.3811 0.1941 60 30
                5 6 0.8190 0.7070 60 20
                6 7 0.1872 0.6188 200 100
                7 8 1.7114 1.2351 200 100
                8 9 1.0300 0.7400 60 20
                9 10 1.0400 0.7400 60 20
                10 11 0.1966 0.0650 45 30
                11 12 0.3744 0.1238 60 35
                12 13 1.4680 1.1550 60 35
                13 14 0.5416 0.7129 120 80
                14 15 0.5910 0.5260 60 10

```

```

15 16 0.7463 0.5450 60 20
16 17 1.2890 1.7210 60 20
17 18 0.7320 0.5740 90 40
2 19 0.1640 0.1565 90 40
19 20 1.5042 1.3554 90 40
20 21 0.4095 0.4784 90 40
21 22 0.7089 0.9373 90 40
3 23 0.4512 0.3083 90 50
23 24 0.8980 0.7091 420 200
24 25 0.8900 0.7011 420 200
6 26 0.2030 0.1034 60 25
26 27 0.2842 0.1447 60 25
27 28 1.0590 0.9337 60 20
28 29 0.8042 0.7006 120 70
29 30 0.5075 0.2585 200 600
30 31 0.9744 0.9630 150 70
31 32 0.3105 0.3619 210 100
32 33 0.3410 0.5302 60 40];

```

```
%ajuste de demanda
```

```
%potencia activa
```

```
Sistema(:,5)=factor_de_demanda*Sistema(:,5);
```

```
%potencia reactiva
```

```
Sistema(:,6)=factor_de_demanda*Sistema(:,6);
```

```
Node = [1 0 1 0 0;
```

```
2 1 1 100 60;
```

```
3 1 1 90 40;
```

```
4 1 1 120 80;
```

```
5 1 1 60 30;
```

```
6 1 1 60 20;
```

```
7 1 1 200 100;
```

```
8 1 1 200 100;
```

```
9 1 1 60 20;
```

```
10 1 1 60 20;
```

```
11 1 1 45 30;
```

```
12 1 1 60 35;
```

```
13 1 1 60 35;
```

```
14 1 1 120 80;
```

```
15 1 1 60 10;
```

```
16 1 1 60 20;
```

```
17 1 1 60 20;
```

```
18 1 1 90 40;
```

```
19 1 1 90 40;
```

```
20 1 1 90 40;
```

```
21 1 1 90 40;
```

```

22 1 1 90 40;
23 1 1 90 50;
24 1 1 420 200;
25 1 1 420 200;
26 1 1 60 25;
27 1 1 60 25;
28 1 1 60 20;
29 1 1 120 70;
30 1 1 200 600;
31 1 1 150 70;
32 1 1 210 100;
33 1 1 60 40];

%ajuste de demanda
%potencia activa
Node(:,4)=factor_de_demanda* Node(:,4);
%potencia reactiva
Node(:,5)=factor_de_demanda* Node(:,5);

%% Pasando a por unidad
Zbase = (1000*Vbase)^2/(1000*Sbase);
Sistema(:,3:4) = Sistema(:,3:4)/Zbase;
Sistema(:,5:6) = Sistema(:,5:6)/Sbase;
Node(:,4:5) = Node(:,4:5)/Sbase;

for gen=1:3
    nodo_gd=GD(1,gen);
    pos=find(Sistema(:,2)==nodo_gd);

    Sistema(pos,5)=Sistema(pos,5)-Inyeccion_variable(1,gen);

    pos=find(Node(:,1)==nodo_gd);
    Node(pos,4)=Node(pos,4)-Inyeccion_variable(1,gen);
end

%% Formaci3n de la matriz GBUS
[Ybus] = formacionYbus(Sistema,NB,NL);
m = 1000;
Ydd = Ybus(2:end,2:end);
S = (Node(2:end,4)+1i*Node(2:end,5));
Ydg = Ybus(2:end,1);

```

```

Vg = 1; %% Voltaje asociado al nodo slack
idd = inv(Ydd);
V0 = ones(size(Ydd,2),1);
for k = 1:m
    %% Estracción de los datos para el flujo lineal:
    x = -idd*((diag(conj(V0))\conj(S) + Ydg*Vg);
    if abs(abs(x) -abs(V0)) <= 1e-10
        V = [Vg;x];
        break
    else
        V0 = x;
    end
end
end
end

%calculo de pérdidas horarias
Losses(iii,1)= real(V.*conj(Ybus*V))*1000;
%calculo de peores tensiones
voltajes_horario=abs(V);
errores_de_voltaje= abs(voltajes_horario-1);
peor_error=max(errores_de_voltaje);
pos=find(errores_de_voltaje==peor_error);
peor_tension(iii,1)=voltajes_horario(pos,1);
%calculo de peor corriente
I_L=zeros(NL,1);
for eee=1:NL
    node_e=Sistema(eee,1);
    V_e=V(node_e,1);
    node_r=Sistema(eee,2);
    V_r=V(node_r,1);
    R_l=Sistema(eee,3);
    X_l=Sistema(eee,4);
    Z_l=R_l+X_l*1i;
    I_L=(V_e-V_r)/Z_l;
    I_L=abs(I_L);
    I_L_horario(eee,1)=I_L*78.9889;

end
peor_corriente(iii,1)=max(I_L_horario);

end
total_power_loss=sum(Losses(:,1));

```

**Código Función Objetivo para Inclusión en los Códigos PSO y SCA**

```

function [total_power_loss]=integracion_gd(Xii,Yii,Zii)
repmax = 10000;
load datos

GD=[12 15 31];

Inyeccion= [Xii Yii Zii]./1000;%se divide por la potencia base10

for iii=1:24
    factor_de_demanda=parametros(iii,1);
    factor_de_generacionsolar=parametros(iii,2);
    Inyeccion_variable=Inyeccion*factor_de_generacionsolar;
for rep = 1:repmax
    Sbase = 1000;%kVA
    Vbase = 12.66;%kV
    Slack = 1; NB = 33; NL = 32;
    %% Datos [i j R[ohm] X[ohm] P[kW] Q[kVA]]
    Sistema = [1 2 0.0922 0.0477 100 60
        2 3 0.4930 0.2511 90 40
        3 4 0.3660 0.1864 120 80
        4 5 0.3811 0.1941 60 30
        5 6 0.8190 0.7070 60 20
        6 7 0.1872 0.6188 200 100
        7 8 1.7114 1.2351 200 100
        8 9 1.0300 0.7400 60 20
        9 10 1.0400 0.7400 60 20
        10 11 0.1966 0.0650 45 30
        11 12 0.3744 0.1238 60 35
        12 13 1.4680 1.1550 60 35
        13 14 0.5416 0.7129 120 80
        14 15 0.5910 0.5260 60 10
        15 16 0.7463 0.5450 60 20
        16 17 1.2890 1.7210 60 20
        17 18 0.7320 0.5740 90 40
        2 19 0.1640 0.1565 90 40
        19 20 1.5042 1.3554 90 40
        20 21 0.4095 0.4784 90 40
        21 22 0.7089 0.9373 90 40
        3 23 0.4512 0.3083 90 50
        23 24 0.8980 0.7091 420 200
        24 25 0.8900 0.7011 420 200
        6 26 0.2030 0.1034 60 25
        26 27 0.2842 0.1447 60 25
        27 28 1.0590 0.9337 60 20
        28 29 0.8042 0.7006 120 70

```

```

29 30 0.5075 0.2585 200 600
30 31 0.9744 0.9630 150 70
31 32 0.3105 0.3619 210 100
32 33 0.3410 0.5302 60 40];

```

```
%ajuste de demanda
```

```
%potencia activa
```

```
Sistema(:,5)=factor_de_demanda*Sistema(:,5);
```

```
%potencia reactiva
```

```
Sistema(:,6)=factor_de_demanda*Sistema(:,6);
```

```
Node = [1 0 1 0 0;
```

```
2 1 1 100 60;
```

```
3 1 1 90 40;
```

```
4 1 1 120 80;
```

```
5 1 1 60 30;
```

```
6 1 1 60 20;
```

```
7 1 1 200 100;
```

```
8 1 1 200 100;
```

```
9 1 1 60 20;
```

```
10 1 1 60 20;
```

```
11 1 1 45 30;
```

```
12 1 1 60 35;
```

```
13 1 1 60 35;
```

```
14 1 1 120 80;
```

```
15 1 1 60 10;
```

```
16 1 1 60 20;
```

```
17 1 1 60 20;
```

```
18 1 1 90 40;
```

```
19 1 1 90 40;
```

```
20 1 1 90 40;
```

```
21 1 1 90 40;
```

```
22 1 1 90 40;
```

```
23 1 1 90 50;
```

```
24 1 1 420 200;
```

```
25 1 1 420 200;
```

```
26 1 1 60 25;
```

```
27 1 1 60 25;
```

```
28 1 1 60 20;
```

```
29 1 1 120 70;
```

```
30 1 1 200 600;
```

```
31 1 1 150 70;
```

```
32 1 1 210 100;
```

```
33 1 1 60 40];
```

```
%ajuste de demanda
```

```

%potencia activa
Node(:,4)=factor_de_demanda* Node(:,4);
%potencia reactiva
Node(:,5)=factor_de_demanda* Node(:,5);

%% Pasando a por unidad
Zbase = (1000*Vbase)^2/(1000*Sbase);
Sistema(:,3:4) = Sistema(:,3:4)/Zbase;
Sistema(:,5:6) = Sistema(:,5:6)/Sbase;
Node(:,4:5) = Node(:,4:5)/Sbase;

for gen=1:3
    nodo_gd=GD(1,gen);
    pos=find(Sistema(:,2)==nodo_gd);

    Sistema(pos,5)=Sistema(pos,5)-Inyeccion_variable(1,gen);

    pos=find(Node(:,1)==nodo_gd);
    Node(pos,4)=Node(pos,4)-Inyeccion_variable(1,gen);
end

%% Formaci3n de la matriz GBUS
[Ybus] = formacionYbus(Sistema,NB,NL);
m = 1000;
Ydd = Ybus(2:end,2:end);
S = (Node(2:end,4)+1i*Node(2:end,5));
Ydg = Ybus(2:end,1);
Vg = 1; %% Voltaje asociado al nodo slack
idd = inv(Ydd);
V0 = ones(size(Ydd,2),1);
for k = 1:m
    %% Estracci3n de los datos para el flujo lineal:
    x = -idd*((diag(conj(V0))\conj(S) + Ydg*Vg);
    if abs(abs(x) -abs(V0)) <= 1e-10
        V = [Vg;x];
        break
    else
        V0 = x;
    end
end
end
end

```



```

%calculo de pÃ©rdidas horarias
Losses(iii,1)= real(V.*conj(Ybus*V))*1000;
%calculo de peores tensiones
voltajes_horario=abs(V);
errores_de_voltaje= abs(voltajes_horario-1);
peor_error=max(errores_de_voltaje);
pos=find(errores_de_voltaje==peor_error);
peor_tension(iii,1)=voltajes_horario(pos,1);
%calculo de peor corriente
I_L=zeros(NL,1);
for eee=1:NL
    node_e=Sistema(eee,1);
    V_e=V(node_e,1);
    node_r=Sistema(eee,2);
    V_r=V(node_r,1);
    R_l=Sistema(eee,3);
    X_l=Sistema(eee,4);
    Z_l=R_l+X_l*1i;
    I_L=(V_e-V_r)/Z_l;
    I_L=abs(I_L);
    I_L_horario(eee,1)=I_L*78.9889;
end
peor_corriente(iii,1)=max(I_L_horario);

end
total_power_loss=sum(Losses(:,1));

```

### CÃ³digo PSO

```

P=3;% cantidad de particulas
Dmax=[ 1200 1200 1200];
Dmin=[0 0 0];
Dim=size(Dmax,2);
inermax=0.7;
inermin=0.001;
w=inermax;
E=1e-6;
itermax=3;
C1=1.494; %coeficiente cognitivo individual
C2=1.494; %coeficiente cognitivo grupal

Vmax_p1=0.01;
Vmax_p2=0.01;

```

```

Vmax_p3=0.01;
% Vmax_p3=
cont=1;
Vel_p1=rand(P,1)*(2*Vmax_p1)-Vmax_p1; %velocidad de las particulas en las posiciones
Vel_p2=rand(P,1)*(2*Vmax_p2)-Vmax_p2;
Vel_p3=rand(P,1)*(2*Vmax_p3)-Vmax_p3;
Vel=[Vel_p1 Vel_p2 Vel_p3];

Xi=(Dmax-Dmin).*rand(P,Dim)+Dmin; %posición inicial
Aptitud_Xi=zeros(P,1);%almacena las soluciones dentro del problema

for i=1:P %evaluacion de las funciones por medio del ciclo for
    Resultado=integracion_gd(Xi(i,1),Xi(i,2),Xi(i,3));
    Aptitud_Xi(i,1)=Resultado%evaluacion de la funcion objetivo
end
Aptitud_mejorposi=Aptitud_Xi; % almacenamos las soluciones
MejorXi=Xi;
[Aptitud_mejorpos_g Ind]=min(Aptitud_mejorposi);
mejorposXig=MejorXi(Ind,:);
Xi=Xi+Vel;

%restricción del espacio de busqueda
for i=1:P
    for j=1:Dim

        if ((Xi(i,j)<Dmin(1,j)) || (Xi(i,j)>Dmax(1,j)) )
            Xi(i,j)=Dmin(1,j)+(Dmax(1,j)-Dmin(1,j))*rand;
        end
    end
end

%proceso total iterativo

while (Aptitud_mejorpos_g>E)

    Aptitud_Xi=zeros(P,1);%almacena las soluciones dentro del problema

    % Metodo de aprendizaje de las tecnicas de optimizacion
    for i=1:P
        Resultado=integracion_gd(Xi(i,1),Xi(i,2),Xi(i,3));
        Aptitud_Xi(i,1)=Resultado;%evaluacion de la funcion objetivo
        if (Aptitud_Xi(i,1)<Aptitud_mejorposi(i,1))%%compara y actualiza los resultados
            Aptitud_mejorposi(i,1)=Aptitud_Xi(i,1);
            MejorXi(i,:)=Xi(i,:);
        end
    end
end

```

```

if (Aptitud_mejorposi(i,1)<Aptitud_mejorpos_g)
    Aptitud_mejorpos_g=Aptitud_mejorposi(i,1);
    Ind=i;
    mejorposXig=MejorXi(Ind,:);
end
end
% método de avance de la técnica de optimización
%este metodo es el que siempre cambia en cada tecnica de optimizacion
if (Aptitud_mejorpos_g>E & cont<itermax)
    for i=1:P
        for j=1:Dim

            R1=rand();
            R2=rand();
            Vel(i,j)=w*Vel(i,j)+C1*R1*(MejorXi(i,j)-Xi(i,j))+C2*R2*(mejorposXig(1,j)-Xi(i,j));
            Xi(i,j)=Xi(i,j)+Vel(i,j);
        end
    end

end
for i=1:P
    for j=1:Dim

        if ((Xi(i,j)<Dmin(1,j)) || (Xi(i,j)>Dmax(1,j)))
            Xi(i,j)=Dmin(1,j)+(Dmax(1,j)-Dmin(1,j))*rand;
        end
    end
end

cont=cont+1;
if cont>itermax
    break
end

end
toc

```

### Código SCA

```

tic
P=3;% cantidad de particulas

Dmax=[ 1200 1200 1200];

```

```

Dmin=[0 0 0];
Dim=size(Dmax,2);
alpha= 2;
E=1e-6;
itermax=3;
cont=1;

Xi=(Dmax-Dmin).*rand(P,Dim)+Dmin; %posición inicial

Aptitud_Xi=zeros(P,1);%almacena las soluciones dentro del problema

for i=1:P %evaluacion de las funciones por medio del ciclo for
    Resultado=integracion_gd(Xi(i,1),Xi(i,2),Xi(i,3));
    Aptitud_Xi(i,1)=Resultado;%evaluacion de la funcion objetivo
end

Aptitud_mejorposi=Aptitud_Xi; % almacenamos las soluciones
MejorXi=Xi;
[Aptitud_mejorposi Ind]=min(Aptitud_mejorposi);
mejorposiXig=MejorXi(Ind,:);

%restriccion del espacio de busqueda
for i=1:P
    for j=1:Dim
        if ((Xi(i,j)<Dmin(1,j)) || (Xi(i,j)>Dmax(1,j)))
            Xi(i,j)=Dmin(1,j)+(Dmax(1,j)-Dmin(1,j))*rand;
        end
    end
end

%proceso total iterativo

while (cont<=itermax)
    Aptitud_Xi=zeros(P,1);%almacena las soluciones dentro del problema
    % Metodo de aprendizaje de las tecnicas de optimizacion
    for i=1:P
        Resultado=integracion_gd(Xi(i,1),Xi(i,2),Xi(i,3));
        Aptitud_Xi(i,1)=Resultado;%evaluacion de la funcion objetivo
        if (Aptitud_Xi(i,1)<Aptitud_mejorposi(i,1))%%compara y actualiza los resultados
            Aptitud_mejorposi(i,1)=Aptitud_Xi(i,1);
            MejorXi(i,:)=Xi(i,:);
        end
    end
end

```

```

if (Aptitud_mejorposi(i,1)<Aptitud_mejorpos_g)
    Aptitud_mejorpos_g=Aptitud_mejorposi(i,1);
    Ind=i;
    mejorposXig=MejorXi(Ind,:);
end

end

% metodo de avance de la tecnica de optimizacion
%este metodo es el que siempre cambia en cada tecnica de optimizacion
for i=1:P
    for j=1:Dim
        r1 = alpha - cont * (alpha / itermax);
        r2 = (2 * pi) * rand();
        r3 = 2 * rand();
        r4 = rand();
        if r4 < 0.5
            X(i,j) = Xi(i,j) + (r1 * sin(r2) * abs(r3 * mejorposXig(1,j)-Xi(i,j)));
        else
            X(i,j) = Xi(i,j) + (r1 * cos(r2) * abs(r3 * mejorposXig(1,j)-Xi(i,j)));
        end
    end
end
Xi=X;
%restriccion del espacio de busqueda
for i=1:P
    for j=1:Dim
        if ((Xi(i,j)<Dmin(1,j)) || (Xi(i,j)>Dmax(1,j)))
            Xi(i,j)=Dmin(1,j)+(Dmax(1,j)-Dmin(1,j))*rand;
        end
    end
end
cont=cont+1;
if cont>itermax
    break
end
end
toc

```