



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO **MODELO DE FLUJO DE POTENCIA ÓPTIMO EN CORRIENTE ALTERNA**

Autor: Adriana Delgado Romera

Director: Javier García González

Co-Director: Erik Francisco Álvarez Quispe

Madrid

Julio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Modelo de flujo de potencia óptimo en corriente alterna

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.

Fdo.: Adriana Delgado Romera

Fecha: 15/ 07/ 2025

Adriana Delgado Romera

Autorizada la entrega del proyecto

Javier García González

Erik Francisco Álvarez Quispe

Fdo.: Javier García González & Erik Francisco Álvarez Quispe

Fecha: 15/ 07/ 2025

Modelo de flujo de potencia óptimo en corriente alterna

Autora: Delgado Romera, Adriana

Director: García González, Javier

Codirector: Álvarez Quispe, Erik Francisco

Entidad colaboradora: ICAI- Universidad Pontificia Comillas & Instituto de Investigación Tecnológica (IIT)

Resumen del proyecto

Este trabajo presenta un estudio detallado del Flujo Óptimo de Potencia en Corriente Alterna (AC-OPF), enfocándose en la problemática generada por la creciente complejidad de los sistemas eléctricos y la alta integración de generación renovable en redes eléctricas malladas. Se propone utilizar uno de los métodos estudiados en la literatura técnica: la Programación por Cónica de Segundo Orden (SOCP), incluyendo además un conjunto de restricciones cíclicas derivadas de la Ley de Tensiones de Kirchhoff (KVL) que permiten obtener una solución realista del problema relajado resultante. El modelo propuesto se valida en sistemas estándar del IEEE (Institute of Electrical and Electronics Engineers), mostrando una considerable mejora en eficiencia computacional y precisión por lo que se concluye que este método puede ser útil para aplicarse en sistemas de mayor tamaño como solución práctica y escalable.

Palabras clave: flujo óptimo de potencia, ac-opf, ac-pf, programación no lineal, no convexo, socp, redes eléctricas malladas, restricciones cíclicas, Python.

1. Introducción

Durante los últimos años, el rápido crecimiento de fuentes renovables intermitentes, la evolución hacia patrones de consumo dinámicos, y la demanda de sistemas adaptativos han incrementado la complejidad operativa de las redes eléctricas (transición y distribución). En este contexto, el Flujo Óptimo de Potencia en Corriente Alterna (AC-OPF) surge como una herramienta fundamental para optimizar el funcionamiento de los sistemas eléctricos. El objetivo es minimizar el coste de operación del sistema del sin violar la capacidad máxima de las líneas y respetando las restricciones técnicas del equipo de generación. Sin embargo, el problema AC-OPF es intrínsecamente complejo debido a su naturaleza no lineal y no convexa, lo que dificulta la convergencia y estabilidad de los métodos de resolución tradicionales. La relevancia del AC-OPF reside en que las redes eléctricas actuales requieren operar con márgenes reducidos respecto a su capacidad máxima para maximizar la integración eficiente de la generación renovable. Por ello resulta crucial disponer de modelos precisos y robustos que minimicen errores operativos y garanticen que no se incurra en energía no suministrada (ENS).

2. Estado de la cuestión

El AC-OPF se ha abordado históricamente mediante numerosos y diferentes enfoques debido a su complejidad intrínseca y a su naturaleza altamente no lineal y no convexa.

Tradicionalmente, se han utilizado métodos basados en Programación No Lineal (NLP), tales como Newton-Raphson o métodos de punto interior, que permiten una precisión notable en la

resolución del problema, aunque presentan desventajas claras relacionadas con la sensibilidad a las condiciones iniciales y altos costes computacionales en redes de gran escala.

Otra alternativa ampliamente empleada en la literatura técnica es el Flujo Óptimo en Corriente Continua (DC-OPF), una aproximación lineal del problema original que simplifica significativamente el cálculo reduciendo las variables involucradas. Esta aproximación no considera los flujos de potencia reactiva, y las pérdidas en las líneas pueden ser consideradas, pero también de forma aproximada. Por ello, en algunos sistemas, especialmente en aquellos donde los flujos de reactiva o las pérdidas sean relevantes, este método aproximado puede dar lugar a soluciones no aceptables. Para superar las limitaciones de precisión y computacionales inherentes a estos métodos tradicionales, en las últimas décadas se han investigado intensamente técnicas de relajación convexa, destacando especialmente la Programación Semidefinida (SDP) y la Programación por Cónica de Segundo Orden (SOCP). La relajación SDP proporciona una aproximación muy precisa, pero puede resultar computacionalmente intensiva en sistemas grandes debido a la necesidad de manejar grandes matrices semidefinidas positivas. En contraste, la SOCP presenta una atractiva alternativa al ofrecer un compromiso eficiente entre precisión y esfuerzo computacional.

Sin embargo, las técnicas de relajación convexa, particularmente SOCP, requieren la inclusión explícita de restricciones adicionales para preservar la exactitud física, especialmente en sistemas mallados donde la presencia de ciclos eléctricos obliga a mantener estrictamente la Ley de Tensiones de Kirchhoff (KVL). Recientemente, la literatura especializada ha señalado la importancia crucial de añadir estas restricciones cíclicas explícitas a los modelos SOCP para asegurar resultados físicamente viables y operativamente relevantes. Este proyecto toma como base estos recientes avances para proponer una formulación SOCP robusta y exacta mediante la inclusión explícita de dichas restricciones cíclicas.

3. Definición del Proyecto

Según lo anterior, el objetivo principal de este proyecto es desarrollar y validar un modelo avanzado del problema AC-OPF basado en una relajación convexa SOCP, enriquecido con restricciones cíclicas explícitas que aseguren la coherencia física en redes malladas.

Este enfoque busca superar las limitaciones inherentes a los métodos tradicionales NLP, ofreciendo una metodología capaz de manejar de manera rápida y precisa grandes sistemas eléctricos con alta integración renovable. Para la validación del modelo, se seleccionan sistemas de referencia del IEEE con configuraciones de 3, 14 y 30 nodos. Esta selección permite evaluar sistemáticamente el rigor, la velocidad y la potencial escalabilidad, respectivamente, del método desarrollado.

4. Descripción del modelo y de la metodología

El modelo propuesto utiliza una formulación basada en el Branch Flow Model (BFM), adaptado específicamente a través de una relajación SOCP para convertir restricciones originalmente no convexas en restricciones cónicas convexas.

En primer lugar, se implementa el Análisis del Flujo de Potencia en Corriente Alterna (ACPF) como etapa preliminar, verificando la viabilidad física del sistema eléctrico sin optimización. Posteriormente, se incorpora la optimización clásica AC-OPF con restricciones operativas adicionales (límites de tensión, generación activa y reactiva, y capacidad térmica de líneas), así como penalizaciones por Energía No Suministrada (ENS). Finalmente, se introduce la relajación convexa SOCP junto con restricciones cíclicas explícitas basadas en KVL para preservar la consistencia angular y la coherencia física del modelo.

El desarrollo detallado de cada uno de estos tres modelos, que representan etapas sucesivas y complementarias hacia el modelo final propuesto (AC-OPF+SOCP), se presenta minuciosamente en el capítulo 3 de esta memoria. En este capítulo se abordan de manera exhaustiva las bases teóricas, la formulación matemática precisa de cada etapa, los criterios específicos de selección de las restricciones operativas y la implementación práctica de los modelos de Python-Pyomo.

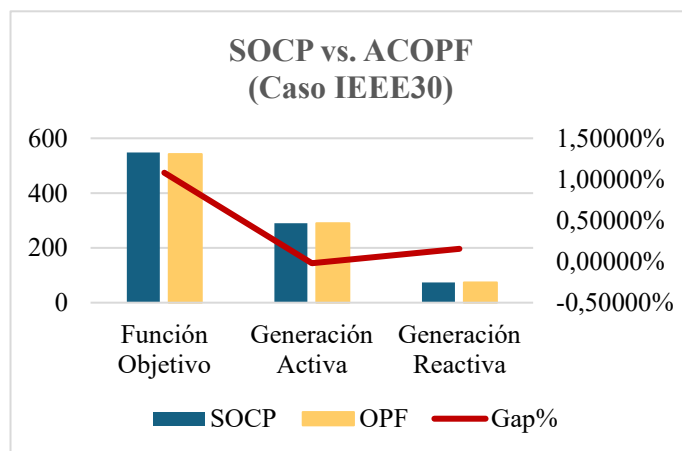
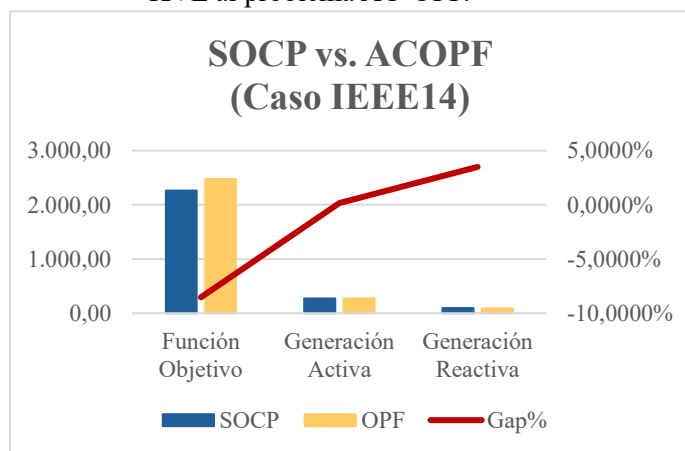
La implementación computacional del modelo utiliza herramientas avanzadas en Python. Pyomo es una biblioteca de modelado algebraico que permite definir problemas de optimización de forma sencilla y eficiente mediante ecuaciones algebraicas simbólicas. Por otro lado, PyPower es una herramienta ampliamente utilizada para el análisis de sistemas eléctricos y simulación de flujos de potencia, proporcionando referencias robustas para la validación del modelo propuesto. Finalmente, IPOPT (Interior Point OPTimizer) es un solver no lineal de punto interior, particularmente eficiente para resolver problemas de optimización no lineales de gran escala, garantizando la obtención de soluciones precisas y factibles en tiempos computacionales razonables. Esta combinación permite lograr soluciones eficientes, escalables y físicamente coherentes para redes eléctricas modernas.

5. Resultados

En este proyecto se evaluó el desempeño del modelo propuesto en tres sistemas de prueba IEEE de 3, 14 y 30 nudos, lo que permitió examinar rigurosamente la precisión y eficiencia del enfoque en redes de distinta escala.

Primero se resuelve el AC Power Flow tradicional (ACPF) y el Optimal Power Flow no convexo (AC-OPF) con Pyomo/IPOPT, que utiliza el método del Punto Interior. Se obtienen resultados prácticamente idénticos a los de herramientas de referencia (PyPower), en todos los casos. Por ejemplo, para el sistema IEEE-30 la solución AC-OPF desarrollada coincide con la referencia con una diferencia absoluta de sólo 0,0007€ en el coste objetivo (error relativo $\sim 0,0000\%$). De igual modo, en el IEEE-14 se alcanza un óptimo de 2472,21€ que difiere menos del 0,5% del valor de referencia. Estas diferencias, presumiblemente originadas por ligeros desajustes en los datos o por diferencias en las metodologías de resolución empleadas (Punto Interior vs. Newton-Raphson), no comprometen la validez ni la fidelidad numérica del modelo AC-OPF implementado. Todas las restricciones operativas (límites de potencia, tensiones, flujos) se satisfacen estrictamente y la energía no suministrada (ENS) resulta nula en las soluciones, garantizando la viabilidad operativa completa del AC-OPF.

A continuación, se le aplica la relajación convexa SOCP enriquecida con restricciones cíclicas de KVL al problema AC-OPF.



Figuras 4 y 7: Comprobación de factibilidad del caso de 14 nudos y 30 nudos

Los resultados demostraron que, gracias a estas relajaciones, el solucionador (solver) encuentra soluciones de costo igual o menor al óptimo tradicional, con mejoras significativas en ciertos casos. En el sistema IEEE-14, la relajación SOCP arrojó un costo objetivo de 2261,41€, un 8,5% inferior al obtenido por el AC-OPF convencional (2472,21 €). Esta notable reducción de coste se logró redistribuyendo la generación: el generador del nudo 1 (el más económico) aumentó su despacho de 254,5 a 269,1 MW, mientras que el generador del nudo 2 redujo su generación activa de 19,7 a 5,6 MW, dedicándose casi exclusivamente al soporte de potencia reactiva (subió ~57%, de 16,4 a 25,9 MVar). Los demás generadores mantuvieron o redujeron ligeramente su generación reactiva para acomodar este nuevo reparto. Globalmente, la demanda se cubrió al 100% y el balance de potencias activas permaneció coherente (274,68 vs 274,22 MW totales), pero la mayor producción de reactiva en el generador 2 hizo que la generación reactiva total aumentase en ~3,17 MVar respecto al AC-OPF.

Tabla 4.18: Comparación optimalidad en el caso de 14 nudos

Parámetro	SOCP IEEE14	OPF IEEE14	Gap	Gap %
Función objetivo [MW]	2.261,41	2.472,21	210,807233	-8,52707%
Generación activa total [MW]	274,6753	274,2222	0,4531	0,1652%
Generación reactiva total [MVar]	94,0079	90,835	3,1729	3,4930%
Demanda activa total [MW]	259	259	0	0,0000%
Demanda reactiva total [MVar]	73,5	73,5	0	0,0000%

Tabla 4.20: Comparación optimalidad en el caso de 30 nudos

Parámetro	SOCP IEEE30	OPF IEEE30	Gap	Gap %
Función objetivo [MW]	548,614562	542,75	5,866462	1,08088%
Generación activa total [MW]	290,3724	290,4307	-0,0583	-0,0201%
Generación reactiva total [MVar]	73,3418	73,2286	0,1132	0,1546%
Demanda activa total [MW]	283,4	283,4	0	0,0000%
Demanda reactiva total [MVar]	126,2	126,2	0	0,0000%

Este resultado evidencia que la relajación convexa “ensancha” la región factible permitiendo un despacho más barato, pero es necesario verificar si dicha solución era físicamente factible en el modelo AC original. Para el sistema IEEE-30, la relajación también encuentra un costo levemente mayor (542,75€ vs 548,61€, un aumento del 1,08%). En este caso de 30 nudos, todas las variables eléctricas obtenidas con el modelo relajado resultan prácticamente indistinguibles de las del AC-OPF exacto, manteniendo un gap relativo inferior al 2% en coste y diferencias despreciables en generación total (variación de -0,02%). La cercanía de estos valores indica que la relajación SOCP propuesta es muy ajustada (tight) para redes de mayor tamaño, logrando reproducir con alta fidelidad las condiciones operativas óptimas del sistema original.

Los flujos de potencia en cada línea y tensiones nodales calculados bajo la relajación coincidieron estrechamente con los del AC-OPF (típicamente con errores menores al 0,5%), lo que podría

sugerir que la incorporación de las restricciones cíclicas de KVL efectivamente preserva la coherencia física en redes malladas grandes.

No obstante, en el caso de 14 nudos la diferencia de 8,5% en el costo alerta sobre una posible violación de las igualdades físicas no convexas en la solución relajada. Al reinyectar los perfiles obtenidos (tensiones, ángulos y potencias) en un flujo de carga AC convencional para validarlos, se detectó que la solución SOCP incumplía ligeramente una restricción: se producía una sobrecarga en una línea, señal de que la relajación había subestimado ciertas pérdidas o límites del modelo original. En otras palabras, la solución hallada por el modelo convexo no era totalmente factible en el sistema real, a pesar de cumplir con los balances de potencia y límites de generación.

Este hallazgo confirma la necesidad de reforzar el modelo relajado en escenarios como IEEE-14, ya sea “apretando” las envolventes (por ejemplo, imponiendo límites angulares más estrictos) o añadiendo cortes válidos adicionales que cierren el pequeño gap restante. De hecho, en pruebas adicionales se comprobó que reduciendo el límite máximo de ángulo entre nodos (de $\pm 60^\circ$ a valores menores) y refinando la restricción cónica de flujo, la solución del IEEE-14 converge a la del AC-OPF original, eliminando el ahorro aparente pero garantizando factibilidad completa.

Desde el punto de vista computacional, los resultados apoyan la eficacia del enfoque propuesto:

Tabla 4.27: Comparación velocidades de cómputo de los modelos

Caso	Modelo	Tiempo [s]	Iteraciones	Mejora en la velocidad (%)
				[Iteraciones/s]
IEEE-14	ACOPF	0,042	21	8,11%
IEEE-14	SOCP–ciclos	0,037	20	
IEEE-30	ACOPF	0,218	78	11,79%
IEEE-30	SOCP–ciclos	0,065	26	

Aunque el modelo AC-OPF+SOCP introduce más variables y restricciones, su naturaleza convexa favorece tiempos de resolución menores en redes grandes. En el IEEE-14, la relajación SOCP necesitó 20 iteraciones en 0,037 s, frente a 21 iteraciones en 0,042 s del AC-OPF clásico; esto equivale a una mejora del ~8,1% en la rapidez de iteración a favor del modelo SOCP. De forma similar, en IEEE-30 el modelo SOCP logró casi un 12% de aceleración, con 26 iteraciones en 0,065 s. Estas comparativas indican que cada iteración del solver es más eficiente con la formulación convexa, compensando con creces el ligero aumento en el número de iteraciones requeridas.

En consecuencia, para tamaños de red crecientes la ventaja computacional del método SOCP se hace más pronunciada, demostrando una excelente escalabilidad. Los experimentos confirman lo reportado en la bibliografía especializada: las relajaciones de segundo orden con refuerzo cíclico ofrecen una alternativa robusta al AC-OPF tradicional, obteniendo soluciones cercanas al óptimo global con un desempeño numérico muy competitivo. La convexidad del modelo no sólo garantiza convergencia global, sino que también proporciona mayor estabilidad frente a distintas condiciones iniciales y escenarios operativos, evitando problemas de divergencia o óptimos locales propios de métodos no convexas. En suma, el modelo AC-OPF propuesto, apoyado en relajación SOCP, cumple con éxito los objetivos de reducir el coste operativo respetando todas las restricciones técnicas, a la vez que mejora significativamente los tiempos de cómputo en comparación con enfoques convencionales. Esto posiciona a la metodología desarrollada como una solución práctica y escalable para la optimización de flujos de potencia en redes eléctricas modernas, facilitando la integración eficiente de generación renovable y garantizando la seguridad del sistema.

6. Conclusiones

Los resultados anteriores permiten extraer una serie de conclusiones técnicas de gran relevancia. En primer lugar, se demostró la viabilidad de relajar convexamente el problema AC-OPF conservando la exactitud física de sus soluciones. La incorporación de restricciones cíclicas basadas en las leyes de Kirchhoff resultó fundamental para que la relajación SOCP reproduzca fielmente los comportamientos de un sistema mallado real, manteniendo los balances de potencia y límites operativos sin desviaciones significativas. En los casos analizados, el modelo propuesto logró gaps de optimalidad muy estrechos, con un esfuerzo computacional muy inferior al modelo no convexo. Esto es un avance notable, ya que se alcanzan prácticamente los mismos óptimos que el AC-OPF no convexo, evitando caer en óptimos locales, pero con las ventajas de rapidez y robustez de un problema convexo global.

Conviene resaltar que el éxito de la relajación depende indudablemente de mantener las principales restricciones físicas. Si se relajaran excesivamente ciertos límites (por ejemplo, no incluir las ecuaciones de flujo cíclicas), el modelo podría entregar soluciones no factibles en la realidad. En nuestro enfoque se confirmó que las restricciones de KVL añadidas eliminan prácticamente este riesgo, al forzar el cumplimiento de las igualdades de tensiones en los ciclos. Aun así, ante la detección de pequeñas discrepancias (caso IEEE-14), se sugiere complementar el modelo con técnicas de “post-proceso”: por ejemplo, utilizar la solución SOCP como punto de partida (warm start) de un flujo de carga AC final, introducir penalizaciones suaves que desincentiven configuraciones no físicas en la relajación, o introducir relaciones específicas entre variables (corriente y límites térmicos). Estas medidas asegurarían que el resultado final sea completamente factible, y lo más cercano a un óptimo global sin sacrificar la convexidad del problema.

Asimismo, quedó patente que la metodología escalonada adoptada (ACPF $\rightarrow$ AC-OPF $\rightarrow$ AC-OPF relajado) fue adecuada. Cada etapa sirvió para validar la anterior y asentar gradualmente la confianza en el modelo final. En particular, la etapa inicial de flujo de carga AC brindó una base sólida al clarificar las ecuaciones y variables del sistema, facilitando la implementación posterior de la función objetivo y restricciones en el modelo de optimización. Por su parte, la solución óptima obtenida con AC-OPF proporcionó un referente de máxima fidelidad contra el cual medir el desempeño de la relajación convexa. De esta manera, fue posible cuantificar con precisión el trade-off entre fidelidad y eficiencia: el AC-OPF tradicional garantiza la solución físicamente exacta (cero errores) a costa de un mayor tiempo de cómputo, mientras que la relajación AC-OPF+SOCP produce una solución casi óptima y factible (error $<2\%$ en el peor caso estudiado) con grandes ahorros en tiempo de cálculo. En contextos operativos donde se valoren los resultados en tiempo real y la convergencia robusta, este pequeño margen de error resulta aceptable, máxime considerando que las soluciones convexas evitan los problemas de inestabilidad y dependencia de condiciones iniciales propios de los algoritmos no convexas.

Otra conclusión importante es la validez generalizada del enfoque para redes de mayor escala. Si bien las pruebas se realizaron en sistemas estándar relativamente pequeños, la literatura y las pruebas realizadas sugieren que el modelo podría escalar a cientos o miles de nudos. Estudios previos ya han aplicado con éxito relajaciones SOCP reforzadas en sistemas de hasta 3000 nudos. En nuestro caso, la excelente escalabilidad observada de 14 a 30 nudos indica que no se prevén cuellos de botella computacionales al incrementar el tamaño: el solver IPOPT mantuvo tiempos de resolución del orden de décimas de segundo, creciendo de forma sub-lineal con el número de variables. Esto abre la puerta a emplear la metodología en casos reales de planificación y operación de red. De hecho, las conclusiones del proyecto señalan que el modelo AC-OPF+SOCP sería aplicable, con las debidas adaptaciones, tanto en estudios de operación en tiempo real (por ejemplo, como herramienta de despacho óptimo en centros de control) como en planificación de la expansión de redes de transporte. La posibilidad de obtener rápidamente soluciones cercanas al óptimo global brinda a los operadores y planificadores una herramienta valiosa para analizar

escenarios complejos con alta penetración renovable, donde los márgenes de operación son reducidos y la rapidez de decisión es crítica.

Finalmente, el proyecto abre varias líneas futuras. Una de las más destacadas es incorporar la gestión de la incertidumbre en la formulación. Dado el auge de las energías renovables, sería valioso extender el modelo a un AC-OPF estocástico o probabilístico, integrando variaciones en la generación eólica/solar y en la demanda. Métodos como la programación robusta o escenarios multi-objetivo podrían ayudar a que la relajación SOCP considere contingencias e incertidumbres, manteniendo su eficiencia. Asimismo, se podría explorar la combinación con técnicas metaheurísticas o de aprendizaje automático (p.ej. redes neuronales entrenadas para aproximar el despacho óptimo), buscando aproximaciones híbridas que ofrezcan soluciones aún más rápidas con errores acotados. Otra extensión natural es aplicar el modelo a redes de transporte malladas de gran tamaño, ya que la literatura reporta buenos resultados de SOCP en sistemas reales de hasta cientos de nodos. Implementar el algoritmo en entornos de software industriales (PSS®E, DlgSILENT) y compararlo contra algoritmos de benchmark actuales (por ejemplo, interior-point robustos), sería un paso lógico para validar su aplicabilidad a nivel industrial.

En resumen, las conclusiones del trabajo confirman que la relajación convexa propuesta es técnica y operativamente sólida, brindando soluciones óptimas, posibles puntos de partido de estas, cercanas con alta eficiencia, y apuntan a un notable potencial para mejorar la gestión óptima de sistemas eléctricos futuros, confiando en que modelos depurados como el desarrollado puedan integrarse como herramientas de uso real en la operación de redes inteligentes.

7. Referencias

A continuación, se presentan las referencias más representativas y utilizadas a lo largo de la memoria:

[CARP62] Carpentier, J., “*Contribution à l’étude du dispatching économique*”, Bulletin de la Société Française des Électriciens (SFE), 1962

[MOMO99] Momoh J.A., El-Hawary M.E., Adapa R., “*A Review of Selected Optimal Power Flow Literature to 1993. Part I: Nonlinear and Quadratic Programming Approaches & Part II: Newton, Linear Programming and Interior Point Methods*”, IEEE Transactions on Power Systems, vol. 14, no. 1, pp. 96–111, 1999.

[FRAN12] Frank S.M., Steponavice I., Rebennack S., “*Optimal Power Flow: A Bibliographic Survey I*”, Energy Systems, vol. 3, no. 3, pp. 221–258, 2012.

[HUAN16] Huang S., Wu Q., Wang J., Zhao H., “*A Sufficient Condition on Convex Relaxation of AC Optimal Power Flow in Distribution Networks*”, IEEE Transactions on Power Systems, vol. 32, no. 2, pp. 1359–1368, 2016.

[YUAN19] Yuan Z., Hesamzadeh M.R., “*Second-order cone AC optimal power flow: convex relaxations and feasible solutions*”, Journal of Modern Power Systems and Clean Energy, vol. 7, no. 2, pp. 268–280, 2019.

[KOCU14] Kocuk B., Dey S.S., Sun X.A., “*Cycle-based decomposition for solving optimal power flow with SOCP relaxation*”, Informe técnico, Georgia Institute of Technology, 2014.

[ALVA19] Álvarez Quispe, E. F., “*Semidefinite Relaxation for the Optimal Operation and Expansion Planning of Power Transmission Systems*”, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, São Paulo, Brasil, 2019.

[ALVA24] Álvarez Quispe E.F., "*Improving Modelling for Optimal Expansion Planning of Power Transmission Systems*", Tesis Doctoral, Universidad Pontificia Comillas, ICAI, 2024.

[CHOW24] Chowdhury, M.M.U.T., Kamalasan S., Paudyal s., "*A Second-Order Cone Programming (SOCP) Based Optimal Power Flow (OPF) Model With Cyclic Constraints for Power Transmission Systems*", IEEE Transactions on Power Systems, vol. 39, n.º 1, pp. 1032–1044. Enero 2024.

Optimal power flow model in alternating current

Author: Delgado Romera, Adriana

Supervisor: García González, Javier

Co-supervisor: Álvarez Quispe, Erik Francisco

Collaborating Entity: ICAI- Universidad Pontificia Comillas & Instituto de Investigación Tecnológica (IIT)

Abstract

This project presents a detailed study of the Alternating Current Optimal Power Flow (AC-OPF), focusing on challenges posed by the increasing complexity of electrical systems and the high integration of renewable generation in meshed electrical grids. An advanced methodology based on Second Order Cone Programming (SOCP), incorporating explicit cyclic constraints derived from Kirchhoff's Voltage Law (KVL), is proposed. The model is validated using standard IEEE systems, demonstrating significant improvements in computational efficiency and accuracy, thus becoming a practical and scalable solution.

Keywords: optimal power flow, AC-OPF, AC power flow, nonlinear programming, nonconvex, SOCP, meshed power networks, cyclic constraints, Python.

1. Introduction

In recent years, the rapid growth of intermittent renewable energy sources, the shift towards dynamic consumption patterns, and the demand for adaptive systems have significantly increased the operational complexity of electrical grids (both transmission and distribution). In this context, Alternating Current Optimal Power Flow (AC-OPF) emerges as a fundamental tool for optimizing the performance of electrical systems by minimizing operational costs and technical losses, while ensuring secure and reliable operating parameters.

However, the AC-OPF problem is inherently complex due to its nonlinear and non-convex nature, complicating the convergence and stability of traditional solution methods. The relevance of AC-OPF is underscored by the necessity for modern electrical grids to operate with narrow margins, maximizing efficient integration of renewable generation. Therefore, precise and robust models capable of minimizing operational errors and preventing unsupplied energy (Energy Not Supplied - ENS) are crucial.

2. State of the art

Historically, the AC-OPF problem has been addressed through numerous approaches due to its intrinsic complexity and highly nonlinear, non-convex nature.

Traditionally, Nonlinear Programming (NLP) methods, such as Newton-Raphson or interior point methods, have been employed, offering significant accuracy. However, these methods present clear disadvantages, such as sensitivity to initial conditions and high computational costs in large-scale networks.

Another widely used alternative in academic literature is Direct Current Optimal Power Flow (DC-OPF), a linear approximation simplifying calculations by reducing involved variables. Nonetheless, this approach sacrifices precision, especially in operational scenarios involving significant reactive flows or substantial energy losses.

To overcome the accuracy and computational limitations of these traditional methods, convex relaxation techniques, especially Semidefinite Programming (SDP) and Second Order Cone Programming (SOCP), have been intensively investigated in recent decades. While SDP provides a highly accurate approximation, it can become computationally intensive for large systems due to handling extensive positive semidefinite matrices. In contrast, SOCP offers an attractive balance between accuracy and computational effort.

However, convex relaxation techniques, particularly SOCP, require explicitly including additional constraints to preserve physical accuracy, particularly in meshed systems where electrical loops strictly demand adherence to Kirchhoff's Voltage Law (KVL). Recent specialized literature highlights the critical importance of explicitly incorporating these cyclic constraints into SOCP models to ensure physically viable and operationally relevant results. This project leverages these recent developments to propose robust and precise SOCP formulation explicitly incorporating cyclic constraints.

3. Project Definition

Considering all the above, the main objective is to develop and validate an advanced AC-OPF model based on SOCP convex relaxation, enriched with explicit cyclic constraints to ensure physical coherence in meshed grids.

This approach aims to overcome inherent limitations in traditional NLP methods, offering a methodology capable of efficiently and accurately handling large electrical systems with significant renewable integration. For validation purposes, standard IEEE reference systems (3, 14, and 30-node configurations) have been selected, enabling systematic evaluation of the robustness, computational speed, and scalability, respectively, of the developed method.

4. Description of the model

The proposed model utilizes a formulation based on the Branch Flow Model (BFM), specifically adapted through SOCP relaxation to convert originally non-convex constraints into convex conical constraints.

Initially, Alternating Current Power Flow (ACPF) analysis is implemented as a preliminary stage, verifying the physical feasibility of the electrical system without optimization. Subsequently, classical AC-OPF optimization is incorporated, adding operational constraints (voltage limits, active and reactive generation limits, and line thermal capacities) as well as penalties for Energy Not Supplied (ENS). Finally, SOCP convex relaxation is introduced along with explicit cyclic constraints based on KVL to preserve angular consistency and the physical coherence of the model.

The detailed development of each of these three models, representing successive and complementary stages towards the final proposed model (AC-OPF+SOCP), is thoroughly presented in Chapter 3 of this report. This chapter covers theoretical foundations, precise mathematical formulation of each stage, specific criteria for selecting operational constraints, and the computational methodologies employed in practical implementation.

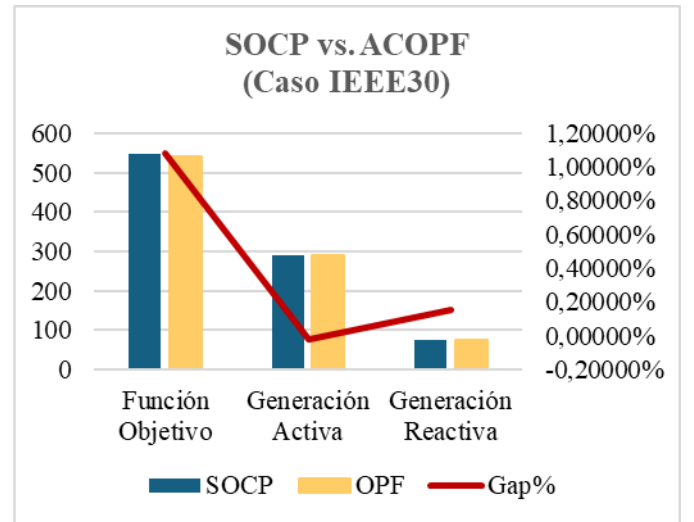
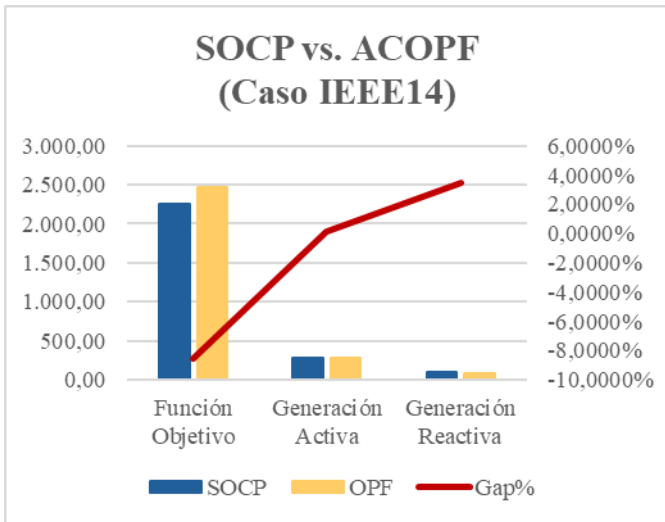
Computational implementation uses advanced Python tools. Pyomo is an algebraic modeling library allowing straightforward and efficient definition of optimization problems through symbolic algebraic equations. PyPower is widely employed for analyzing electrical systems and simulating power flows, providing robust references for validating the proposed model. Finally, IPOPT (Interior Point OPTimizer) is a robust nonlinear interior-point solver particularly efficient in solving large-scale nonlinear optimization problems, ensuring accurate and feasible solutions within reasonable computational times. This combination allows us to achieve efficient, scalable, and physically coherent solutions for modern electrical grids.

5. Results

In this project, the performance of the proposed model was evaluated on three IEEE test systems with 3, 14, and 30 buses, allowing for a rigorous examination of the accuracy and efficiency of the approach across networks of varying scale.

First, the traditional AC Power Flow (ACPF) and the non-convex Optimal Power Flow (AC-OPF) were solved using Pyomo/IPOPT, which employs the Interior Point Method. In all cases, the results obtained were practically identical to those of reference tools (such as PyPower). For instance, in the IEEE-30 system, the developed AC-OPF solution matched reference with an absolute difference of only €0.0007 in the objective cost (relative error ~0.0000%). Similarly, in the IEEE-14 case, an optimum of €2472.21 was reached, differing by less than 0.5% from the benchmark value. These discrepancies, presumably caused by slight data inconsistencies or differences in solution methodologies (Interior Point vs. Newton-Raphson), do not compromise the validity or numerical fidelity of the implemented AC-OPF model. All operational constraints (power limits, voltages, flows) were strictly satisfied, and the energy not supplied (ENS) was zero in the solutions, ensuring full operational feasibility of the AC-OPF.

Next, a convex SOCP relaxation enriched with KVL-based cycle constraints was applied to the AC-OPF problem.



Figures 4 and 7: Feasibility check of case 14 nodes and 30 nodes

The results showed that, thanks to these relaxations, the solver found solutions with equal or lower cost than the traditional optimum, with significant improvements in certain cases. In the IEEE-14 system, the SOCP relaxation yielded an objective cost of €2261.41, which is 8.5% lower than that obtained by the conventional AC-OPF (€2472.21). This notable cost reduction was achieved by redistributing generation: the generator at bus 1 (the most economical) increased its dispatch from 254.5 to 269.1 MW, while the generator at bus 2 reduced its active power output from 19.7 to

5.6 MW, focusing almost exclusively on reactive power support (increasing by ~57%, from 16.4 to 25.9 MVar). The other generators maintained or slightly reduced their reactive generation to accommodate this new distribution. Overall, the demand was fully met, and the active power balance remained consistent (274.68 vs 274.22 MW total), but the higher reactive output from generator 2 caused total reactive generation to increase by ~3.17 MVar compared to the AC-OPF.

Table 4.18: Optimality comparison in the 14 nodes case

Parameter	SOCP IEEE14	OPF IEEE14	Gap	Gap %
Objective Function [MW]	2.261,41	2.472,21	210,807233	-8,52707%
Total Active Generation [MW]	274,6753	274,2222	0,4531	0,1652%
Total Reactive Generation [MVar]	94,0079	90,835	3,1729	3,4930%
Total Active Demand [MW]	259	259	0	0,0000%
Total Reactive Demand [MVar]	73,5	73,5	0	0,0000%

Table 4.20: Optimality comparison in the 30 nodes case

Parameter	SOCP IEEE14	OPF IEEE30	Gap	Gap %
Objective Function [MW]	548,614562	542,75	5,866462	1,08088%
Total Active Generation [MW]	290,3724	290,4307	-0,0583	-0,0201%
Total Reactive Generation [MVar]	73,3418	73,2286	0,1132	0,1546%
Total Active Demand [MW]	283,4	283,4	0	0,0000%
Total Reactive Demand [MVar]	126,2	126,2	0	0,0000%

This result indicates that the convex relaxation "widens" the feasible region, enabling a cheaper dispatch, but it is necessary to verify whether the solution is physically feasible in the original AC model. For the IEEE-30 system, the relaxation also found a slightly lower cost (€542.75 vs €548.61, a reduction of 1.08%). In this 30-bus case, all electrical variables obtained from the relaxed model were virtually indistinguishable from those of the exact AC-OPF, with a relative cost gap of less than 2% and negligible differences in total generation (variation of -0.02%). The closeness of these values indicates that the proposed SOCP relaxation is very tight for larger networks, successfully reproducing the system's optimal operating conditions with high fidelity.

The power flows in each line and nodal voltages calculated under the relaxation closely matched those from the AC-OPF (typically with errors below 0.5%), suggesting that the inclusion of KVL-based cycle constraints effectively preserves physical coherence in large, meshed networks.

However, in the 14-bus case, the 8.5% cost difference raises concerns about a possible violation of non-convex physical equalities in the relaxed solution. When the obtained profiles (voltages, angles, and power) were reinjected into a conventional AC power flow for validation, it was found that the SOCP solution slightly violated one constraint: an overload occurred on one line, indicating that the relaxation had underestimated certain losses or limits in the original model. In other words, the solution found by the convex model was not fully feasible in the real system, despite complying with power balance and generation limits.

This finding confirms the need to reinforce the relaxed model in scenarios like IEEE-14, either by tightening the envelopes (e.g., imposing stricter angle difference limits) or by adding additional valid cuts to close the remaining small gap. In fact, additional tests showed that by reducing the maximum angle limit between nodes (from $\pm 60^\circ$ to smaller values) and refining the conic flow constraint, the IEEE-14 solution converged to that of the original AC-OPF, eliminating the apparent savings but ensuring full feasibility.

From a computational standpoint, the results support the effectiveness of the proposed approach:

Table 4.27: Comparison of Computational Speeds Between Models

Case	Model	Time [s]	Iterations	Speed improvement(%) [Iteraciones/s]
IEEE-14	ACOPF	0,042	21	8,11%
IEEE-14	SOCP–ciclos	0,037	20	
IEEE-30	ACOPF	0,218	78	11,79%
IEEE-30	SOCP–ciclos	0,065	26	

Although the AC-OPF+SOCP model introduces more variables and constraints, its convex nature leads to shorter solution times in larger networks. In the IEEE-14 case, the SOCP relaxation required 20 iterations in 0.037 seconds, compared to 21 iterations in 0.042 seconds for the classic AC-OPF; this corresponds to an approximate 8% improvement in iteration speed in favor of the SOCP model. Similarly, in the IEEE-30 case, the SOCP model achieved a 12% speedup, completing 26 iterations in 0.065 seconds. These comparisons indicate that each solver iteration is more efficient with the convex formulation, more than offsetting the slight increase in the number of iterations required.

As a result, for growing network sizes, the computational advantage of the SOCP method becomes more pronounced, demonstrating excellent scalability. The experiments confirm what has been reported in the specialized literature: second-order cone relaxations with cycle reinforcement offer a robust alternative to traditional AC-OPF, producing solutions close to the global optimum with highly competitive numerical performance. The convexity of the model not only ensures global convergence but also provides greater stability under different initial conditions and operational scenarios, avoiding divergence issues or local optima commonly found in non-convex methods. To sum up, the proposed AC-OPF model, supported by SOCP relaxation, successfully meets the objectives of reducing operating cost while respecting all technical constraints, and significantly improves computation times compared to conventional approaches. This positions the developed methodology as a practical and scalable solution for power flow optimization in modern electric networks, facilitating the efficient integration of renewable generation and ensuring system security.

6. Conclusions

The above results lead to several highly relevant technical conclusions. First, the feasibility of applying a convex relaxation to the AC-OPF problem while preserving the physical accuracy of its solutions was clearly demonstrated. The incorporation of cyclic constraints based on Kirchhoff's laws proved essential for the SOCP relaxation to faithfully reproduce the behavior of a real meshed system, maintaining power balances and operational limits without significant deviations. In the cases analyzed, the proposed model achieved very narrow optimality gaps with substantially lower computational effort than the non-convex model. This represents a notable advancement, as it achieves virtually the same optima as the non-convex AC-OPF while avoiding local minima, and it does so with the speed and robustness advantages of a globally convex problem.

It is important to emphasize that the success of relaxation depends critically on preserving the main physical constraints. If certain limits are overly relaxed (e.g., omitting cyclic flow equations), the model could yield solutions that are infeasible in practice. In our approach, the inclusion of KVL-based constraints effectively eliminated this risk by enforcing voltage equalities around cycles. Still, in cases where small discrepancies were detected (as in IEEE-14), we suggest complementing the model with post-processing techniques — for example, using the SOCP solution as a warm start for a final AC power flow run, introducing soft penalties to discourage non-physical configurations, or including specific relationships between variables (such as current magnitudes and thermal limits). These measures would ensure that the final solution is fully feasible and as close as possible to the global optimum without sacrificing convexity.

Moreover, it became clear that the staged methodology adopted (ACPF $\rightarrow$ AC-OPF $\rightarrow$ relaxed AC-OPF) was well-suited. Each phase validated the previous one and progressively built confidence in the final model. In particular, the initial AC power flow stage provided a solid foundation by clarifying the system's equations and variables, which facilitated the subsequent implementation of the objective function and constraints in the optimization model. The optimal solution obtained with the AC-OPF then served as a high-fidelity reference against which the performance of the convex relaxation could be rigorously evaluated. This made it possible to quantify the trade-off between fidelity and efficiency with precision: the traditional AC-OPF guarantees a physically exact solution (zero modeling error) at the cost of higher computation time, while the relaxed AC-OPF+SOCP delivers an almost optimal and feasible solution (error $<2\%$ in the worst case studied) with significant time savings. In operational contexts where results and robust convergence are essential; such a small margin of error is acceptable—especially considering that convex solutions avoid the instability and sensitivity to initial conditions inherent to non-convex algorithms.

Another key conclusion is the general applicability of the approach to larger-scale networks. Although the tests were conducted on relatively small benchmark systems, both the literature and the project's experiments suggest that the model could scale to hundreds or even thousands of buses. Prior studies have successfully applied reinforced SOCP relaxations to systems with up to 3000 buses. In our case, the excellent scalability observed from 14 to 30 buses indicates that no computational bottlenecks are expected as system size increases: the IPOPT solver maintained solve times on the order of tenths of a second, growing sub-linearly with the number of variables. This opens the door to using the methodology in real-world grid operation and planning scenarios. In fact, the conclusions of this project suggest that the AC-OPF+SOCP model could be applied, with suitable adaptations, both in real-time operation studies (e.g., as an optimal dispatch tool in control centers) and in transmission expansion planning. The ability to quickly obtain near-globally optimal solutions provides system operators and planners with a valuable tool for analyzing complex scenarios with high renewable penetration, where operating margins are tight and rapid decision-making is critical.

Finally, the project outlines several promising future research directions. One of the most notable is the incorporation of uncertainty management into the formulation. Given the rise of renewable energy sources, extending the model to a stochastic or probabilistic AC-OPF framework would be highly valuable, integrating variability in wind/solar generation and load demand. Methods such as robust optimization or multi-scenario approaches could help the SOCP relaxation account for contingencies and uncertainties while preserving its efficiency. Additionally, combining the model with metaheuristic or machine learning techniques (e.g., neural networks trained to approximate the optimal dispatch) could lead to hybrid approaches that deliver even faster solutions with bounded errors. Another natural extension would be to apply the model to large-scale meshed transmission networks, as the literature reports positive SOCP performance in real systems with hundreds of nodes. Implementing the algorithm in industrial software environments

(such as PSS®E or DlgSILENT) and benchmarking it against current state-of-the-art solvers (e.g., robust interior-point methods) would be a logical step to validate its industrial applicability. In summary, the project’s conclusions confirm that the proposed convex relaxation is technically and operationally sound, providing near-optimal solutions with high efficiency, and pointing to significant potential for improving the optimal management of future power systems. There is strong confidence that models like the one developed in this project can be integrated as practical tools for smart grid operation.

7. References

The following are the most representative and frequently cited references throughout the thesis:

[CARP62] Carpentier, J., “Contribution à l’étude du dispatching économique,” *Bulletin of the French Society of Electricians (SFE)*, 1962.

[MOMO99] Momoh, J.A., El-Hawary, M.E., Adapa, R., “A Review of Selected Optimal Power Flow Literature to 1993. Part I: Nonlinear and Quadratic Programming Approaches & Part II: Newton, Linear Programming and Interior Point Methods,” *IEEE Transactions on Power Systems*, vol. 14, no. 1, pp. 96–111, 1999.

[FRAN12] Frank, S.M., Steponavice, I., Rebennack, S., “Optimal Power Flow: A Bibliographic Survey I,” *Energy Systems*, vol. 3, no. 3, pp. 221–258, 2012.

[HUAN16] Huang, S., Wu, Q., Wang, J., Zhao, H., “A Sufficient Condition on Convex Relaxation of AC Optimal Power Flow in Distribution Networks,” *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1359–1368, 2016.

[YUAN19] Yuan, Z., Hesamzadeh, M.R., “Second-Order Cone AC Optimal Power Flow: Convex Relaxations and Feasible Solutions,” *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 2, pp. 268–280, 2019.

[KOCU14] Kocuk, B., Dey, S.S., Sun, X.A., “Cycle-Based Decomposition for Solving Optimal Power Flow with SOCP Relaxation,” Technical Report, Georgia Institute of Technology, 2014.

[ALVA19] Álvarez Quispe, E.F., “Semidefinite Relaxation for the Optimal Operation and Expansion Planning of Power Transmission Systems,” Universidade Estadual de Campinas, Faculty of Electrical and Computer Engineering, São Paulo, Brazil, 2019.

[ALVA24] Álvarez Quispe, E.F., “Improving Modelling for Optimal Expansion Planning of Power Transmission Systems,” Doctoral Thesis, Universidad Pontificia Comillas, ICAI, 2024.

[CHOW24] Chowdhury, M.M.U.T., Kamalasadan, S., Paudyal, S., “A Second-Order Cone Programming (SOCP) Based Optimal Power Flow (OPF) Model With Cyclic Constraints for Power Transmission Systems,” *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 1032–1044, January 2024.

Listado de nomenclaturas

Variables

- $P_{g,i}$ Potencia activa generada en el nudo i.
- $Q_{g,i}$ Potencia reactiva generada en el nudo i.
- $fP_{ij,c}$ Flujo de potencia activa desde nudo i al j, (a través de la línea c).
- $fQ_{ij,c}$ Flujo de potencia reactiva desde nudo i al j, (a través de la línea c).
- V_i Módulo de tensión en el nudo i.
- $I_{ij,c}$ Corriente que fluye por la línea que une los nudos i y j, (a través de la línea c)
- θ_i Ángulo de tensión en el nudo i.
- θ_{ij} Diferencia de ángulos entre nudos i y j.
- P_{ENS} y Q_{ENS} Energía no suministrada en el nudo i: activa y reactiva, respectivamente.

Parámetros

- $P_{d,i}$ Potencia activa demandada en el nudo n.
- $Q_{d,i}$ Potencia reactiva demandada en el nudo n.
- $z_{ij,c}$ Impedancia de la línea (c) que une los nodos i y j.
- $r_{ij,c}$ Resistencia de la línea (c) que une los nodos i y j.
- $x_{ij,c}$ Reactancia de la línea (c) que une los nodos i y j.
- $S_{ij,c}$ Capacidad de transmisión máxima de la línea.
- $g_{ij,c}$ Conductancia de derivación de la línea (c) que une los nodos i y j ($g=0$, línea de transporte).
- $b_{ij,c}$ Susceptancia de derivación de la línea (c) que une los nodos i y j.
- $b_{sh,i}$ Susceptancia Shunt (en paralelo) de derivación del nudo i.
- $g_{sh,i}$ Conductancia Shunt (en paralelo) de derivación del nudo i.
- a_{ij} Tap ratio de transformador entre puntos.
- $P_{gt}^{max}, P_{gt}^{min}, Q_{gt}^{max}, Q_{gt}^{min}, V_t^{min}, V_t^{max}, \theta_t^{min}, \theta_t^{max}$ Límites de producción.

$$z_{ij,c} = r_{ij,c} + x_{ij,c}j$$

$$\frac{1}{z_{ij,c}} = g_{ij,c} + b_{ij,c}j$$

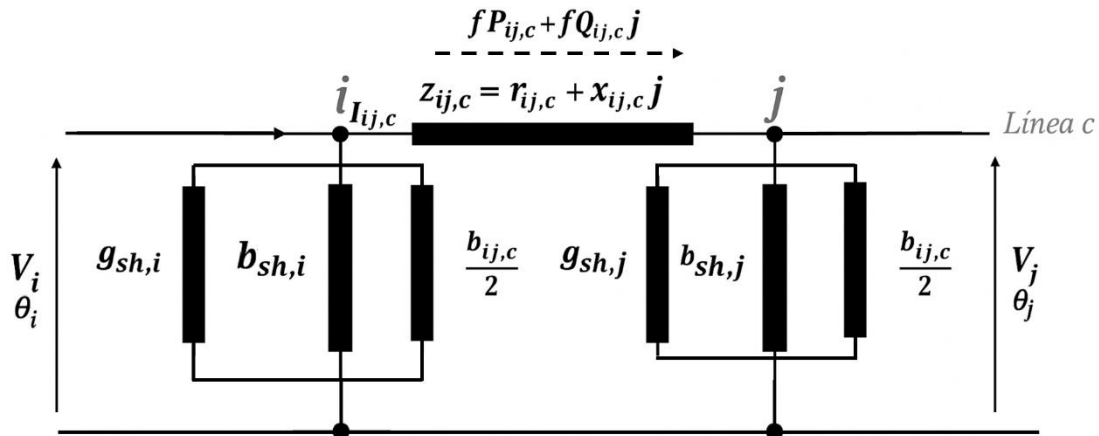


Figura 1: Esquema de referencia en PI de línea c

Elementos de la línea

En esta sección, se ofrece una breve descripción de los principales elementos de la red, los cuales se mencionan de forma recurrente a lo largo de esta memoria.

- **Nudo/Nodo:** Punto de la red en el que se conectan tres o más líneas de transporte. Es esencial para el análisis del sistema, ya que en él se aplica la ley de Kirchhoff de corrientes y se calculan las tensiones. Los nudos representan los lugares donde se concentran y distribuyen las potencias eléctricas.
- **Línea:** Elemento de la red que conecta directamente dos nudos. Puede representar una línea de transporte, un transformador, una carga, una impedancia u otro componente eléctrico. Transporta corriente y tiene caídas de tensión asociadas.
- **Ciclo:** Camino cerrado que comienza y termina en el mismo nudo, pasa por al menos tres nudos diferentes, y no repite ningún nodo ni línea, excepto el de inicio y final.
- **Tomas de un transformador:** También conocidos como *taps*. Puntos accesibles a lo largo del devanado de un transformador (normalmente del primario) que permiten variar su relación de transformación, facilitando el control del nivel de tensión en la red.

Índice de contenido

1. Introducción y planteamiento del proyecto	23
1.1 Introducción	23
1.2 Motivación	24
1.3 Objetivos del proyecto	25
1.4 Metodología de trabajo	25
2. Metodologías actuales de formulación y optimización del flujo de potencia	27
Flujo óptimo de potencia en corriente alterna (AC-OPF)	27
2.1 Métodos tradicionales de resolución.....	29
Aproximación por flujo de potencia óptimo en corriente continua (DC-OPF).....	29
2.1.1 Métodos de programación no lineal (NLP)	30
2.2 Relaxaciones convexas del AC-OPF	31
2.2.1 Programación semidefinida (SDP)	31
2.2.2 Programación Cónica de Segundo Orden (SOCP)	33
2.3 Avances recientes.....	37
2.3.1 Algoritmos metaheurísticos e híbridos	37
2.3.2 Técnicas de aprendizaje automático y aprendizaje profundo en AC-OPF	38
2.3.3 Mejoras en las relajaciones.....	40
3. Descripción del modelo desarrollado.....	42
3.1 Flujo de potencia en corriente alterna, ACPF	42
3.1.1 Ecuaciones del modelo	42
3.1.2 Script del modelo.....	44
3.2 Flujo de potencia óptimo en corriente alterna, ACOPF	45
3.2.1 Diferencias entre ACPF y el ACOPF	45
3.2.2 Función objetivo.....	46
3.2.3 Ecuaciones del modelo	47
3.2.4 Script del modelo.....	48
3.3 Flujo óptimo con relajación SOCP	49
3.3.1 Diferencias entre ACOPF y ACOPF+SOCP	49
3.3.2 Ecuaciones del modelo	50
3.3.3 Script del modelo.....	52
3.3.4 Cálculo de ciclos.....	53
4. Análisis de los resultados y validación.....	57
4.1 Herramientas de resolución	57
4.2 Presentación de los casos	58
4.3 Resultados del flujo de potencia en corriente alterna, ACPF	59

4.3.1	Caso de 3 nudos.....	59
4.3.2	Caso de 14 nudos.....	60
4.3.3	Caso de 30 nudos.....	63
4.3.4	Conclusiones del modelo ACPF	64
4.4	Resultados del flujo de potencia óptimo en corriente alterna, ACOPF.....	65
4.4.1	Caso de 3 nudos.....	65
4.4.2	Caso de 14 nudos.....	66
4.4.3	Caso de 30 nudos.....	70
4.4.4	Conclusiones del modelo ACOPF	72
4.5	Resultados de la relajación	72
4.5.1	Optimalidad	73
4.5.2	Factibilidad.....	78
4.5.3	Conclusiones del modelo SOCP.....	86
4.6	Comparación de modelos.....	87
4.6.1	Comparación cualitativa.....	87
4.6.2	Comparación cuantitativa	88
5.	Conclusiones	90
5.1	Conclusiones acerca de la metodología	90
5.1.1	Mejoras para futuros estudios.....	90
5.1.2	Puntos fuertes de la metodología.....	91
5.2	Conclusiones acerca de los resultados y líneas futuras de acción.....	91
5.2.1	Recomendaciones para futuros estudios.....	92
5.2.2	Líneas futuras de acción: Gestión de incertidumbre con técnicas SOCP	92
5.3	El futuro de las redes eléctricas (ACOPF+SOCP+Renovables).....	93
5.3.1	Aplicación en Redes de Transporte (Sistemas mallados).....	93
5.4	Impacto económico.....	94
6.	Bibliografía	96
7.	Anexo I: Alineación con los Objetivos de Desarrollo Sostenible (ODS).....	99
7.1	ODS 7, 11, 13.....	99
7.2	ODS 9	99
7.2.1	Smart Grids.....	100
7.3	ODS 12	100
8.	Anexo II: Descripción detallada de casos de estudio.....	101
8.1	Caso de 3 nudos	101
8.2	Caso de 14 nudos	102
8.3	Caso de 30 nudos	103
9.	Anexo III: Códigos de Python	107

9.1	Script del flujo de potencia en corriente alterna, ACPF.....	107
9.2	Script del flujo de potencia óptimo en corriente alterna, ACOPF	113
9.3	Script del flujo óptimo con relajación SOCP.....	120
9.4	Script del cálculo de ciclos	130
10.	Anexo IV: Resultados.....	133
10.1	Resultados de Cálculo de Ciclos.....	133
10.2	Resultados relajación SOCP: Caso 14 nudos con $\theta_m = \pm 60$	139
10.2.1	Relajación.....	139
10.2.2	Inyección en ACPF	144
10.3	Resultados relajación SOCP: Caso 30 nudos con $\theta_m = \pm 60$	146
10.3.1	Relajación.....	146
10.3.2	Inyección en ACPF	155
10.4	Endurecimiento de envolventes	158
10.4.1	ACPF con 14 nudos con $\theta_m = \pm 30$	158
10.4.2	ACPF con 14 nudos con $\theta_m = \pm 45$	160

1. Introducción y planteamiento del proyecto

1.1 Introducción

Las últimas décadas han supuesto para la humanidad un tiempo de acelerada expansión, digitalización y descarbonización de los sistemas eléctricos. Este proceso se ha visto tanto acompañado como impulsado por la integración de fuentes de energía renovable, la incorporación de tecnologías de respuesta a la demanda variable, y la aparición de nuevos patrones de consumo eléctrico. Como resultado, ha tenido lugar un crecimiento sostenido en la complejidad estructural y funcional de los sistemas eléctricos en los países desarrollados de todo el mundo.

En el caso español, la red de transporte gestionada por Red Eléctrica de España (REE) opera principalmente en 400 kV y 220 kV y constituye un mallado de alta capacidad. Al cierre de 2024, la longitud total de circuitos alcanzaba 45.674 km, tras la puesta en servicio de 487 km ese año, y se añadieron 197 nuevas posiciones de subestación, elevando la capacidad de transformación a 97.216 MVA (REEE24). El sistema se articula mediante varios centenares de nudos de 220 kV o superior y más de 700 subestaciones (REEE25i). En cuanto a las interconexiones, España dispone de una capacidad comercial de 4 GW con Francia y aproximadamente 4,2 GW con Portugal (tras el nuevo enlace de Fontefría), además de enlaces con Andorra y Marruecos, lo que refuerza la seguridad de suministro (REEE25ii) y la integración de renovables.

En esta línea, España ha experimentado un crecimiento notable en la integración de energías renovables en los últimos años. En 2024, el parque generador incorporó 7,3 GW adicionales, alrededor de 6GW de fotovoltaica y 1,3 GW de eólica, alcanzando un total de aproximadamente 129 GW de potencia instalada, de los cuales 66% son renovables (REEE24). Como consecuencia, las fuentes renovables representan el 56,8% de la generación eléctrica nacional (REEE25iii).

Para optimizar el rendimiento de esta red eléctrica, es necesario utilizar de forma eficiente sus recursos disponibles, lo que permite obtener un flujo de potencia óptimo y, al mismo tiempo, minimizar tanto las pérdidas energéticas como los costes operativos asociados.

Dado que las ecuaciones que modelan los flujos de potencia por las líneas cuando se considera la naturaleza de la corriente alterna (AC- por sus siglas en inglés) son no lineales ni convexas por naturaleza, el problema de optimización del flujo de potencia a través de dicha red (AC-OPF) resulta complejo. Sin embargo, garantizar una asignación eficiente de los recursos disponibles es fundamental para dar el siguiente paso hacia la existencia de redes inteligentes, innovadoras, eficientes, seguras y económicas. Ante tal situación, la mejora en la eficiencia de la gestión se ha convertido en una prioridad, no solo para los operadores; sino también para los investigadores del ámbito técnico, computacional y académico (MOMO99); (FRAN12); (ONE12).

Como se menciona anteriormente, el problema AC-OPF radica en la distribución óptima de las variables operativas dentro de un sistema eléctrico. La formulación persigue minimizar el coste de operación del sistema mediante la optimización conjunta de las potencias generadas, los módulos de tensión y los flujos de potencia en las líneas. Todo ello respetando las restricciones físicas y técnicas de la red, y asegurando que los parámetros del sistema se mantengan dentro de límites seguros, evitando violaciones de capacidad y sobrecargas.

La cuestión se fundamenta en ecuaciones no lineales, con términos cuadráticos, que representan la relación entre tensión y potencia. Dicho sistema deriva de la formulación de las leyes de Kirchhoff. Todo ello, otorga al problema un carácter no convexo que dificulta su resolución por los métodos tradicionales de optimización (HUAN16).

Tradicionalmente, con el objetivo de solucionar este problema se ha recurrido a las simplificaciones. Principalmente, la más usada es la del flujo de potencia en corriente continua

(DC-OPF). Dicha formulación asume ángulos de tensión pequeños, ignora las pérdidas de transporte por las líneas y excluye la potencia reactiva, elemento clave de la aproximación a red continua. Si bien este enfoque es preciso bajo estos supuestos mencionados, reduce notablemente la complejidad matemática y computacional, también suprime elementos clave del funcionamiento real de la red, lo cual puede poner en duda la fiabilidad de los resultados obtenidos en situaciones de elevada carga o penetración de renovables (FRAN12).

Una alternativa que otorga mayor precisión a los resultados consiste en emplear algoritmos de Programación No Lineal (NLP), como los métodos de punto interior o Newton-Raphson, con capacidad para solucionar el AC-OPF directamente. Sin embargo, estos métodos son sensibles a las condiciones iniciales, y pueden tanto llegar a converger en óptimos locales como a requerir un alto coste computacional a la hora de escalar el problema a instancias de gran tamaño.

En consecuencia, ante las mencionadas limitaciones, se han desarrollado métodos de relajación convexa. Así se obtiene una aproximación al óptimo global con una carga computacional significativamente reducida. Las relajaciones más destacables son la Programación Semidefinida (SDP) y, en especial, la Programación Cónica de Segundo Orden (SOCP), técnica consolidada, eficiente y escalable.

La segunda convierte las restricciones no convexas en restricciones cónicas tratables mediante optimización convexa. Esta reformulación mantiene una alta fidelidad con el comportamiento físico del sistema, (YUAN19), (HUAN16).

Sin embargo, al aplicarse a las redes de transporte, véase las redes malladas, se introduce un desafío adicional: la presencia de ciclos en la red. Esto requiere la incorporación explícita de restricciones cíclicas, derivadas de la Ley de Voltajes de Kirchhoff (KVL), para evitar posibles inconsistencias físicas y mantener la validez del modelo (MARL17). Estas restricciones cíclicas imponen que la suma de los ángulos de tensión de todos los nodos que conforman y bucle, sea nula.

Con todo, este trabajo se basa en una formulación del AC-OPF unida a una relajación SOCP, ampliamente desarrollada en la literatura especializada citada en esta memoria. Dicha formulación incluye las restricciones cíclicas, con el objetivo de mejorar la precisión del modelo en redes malladas.

A partir de unos casos de estudio sobre redes de 3, 14 y 30 nudos, se comparará el rendimiento del método SOCP, con los métodos NLP y con soluciones de referencia obtenidas mediante PyPower, con el fin de validar su aplicabilidad práctica en entornos reales.

1.2 Motivación

Este Trabajo de Fin de Grado (TFG) surge de la intersección entre la formación académica en ingeniería y el creciente protagonismo de la optimización energética en el contexto de la transición eléctrica. La necesidad de operar redes más eficientes, resilientes y adaptadas a una generación cada vez más distribuida y renovable exige herramientas matemáticas robustas. En este sentido, el presente trabajo se centra en el desarrollo y evaluación de un modelo de flujo óptimo de potencia en corriente alterna (AC-OPF) formulado como un problema convexo mediante Programación de Segundo Orden (SOCP). Esta técnica permite garantizar soluciones factibles y eficientes sin incurrir en los costes computacionales de métodos no convexos tradicionales.

Una de las principales motivaciones de este estudio es profundizar en las técnicas matemáticas avanzadas que permiten resolver el AC-OPF, y familiarizarse con los lenguajes y entornos de programación que abordan el problema. Para ello, se utilizarán herramientas de modelado en Python con Pyomo, así como simulaciones de referencia con PYPOWER. Se implementará y

validará el modelo sobre sistemas de prueba de diferente escala (IEEE de 3, 14 y 30 nudos), evaluando su capacidad para minimizar costes y pérdidas sin comprometer la factibilidad física del sistema. Gracias a estos solvers se podrá realizar un análisis profundo y una comparación de los resultados con soluciones clásicas y cuantificar el rendimiento del enfoque SOCP.

Esta investigación no solo aporta valor académico, sino que adquiere especial relevancia en el ámbito profesional, particularmente en el contexto de futuras prácticas obtenidas en consultoría energética. En dichas prácticas se analizará la creciente descentralización del sistema eléctrico y la variabilidad de fuentes como la solar y la eólica requieren enfoques computacionalmente eficientes para abordar problemas como el OPF con garantías físicas. En este marco, el proyecto constituye una contribución técnica relevante con aplicabilidad directa en la operación y planificación moderna de redes eléctricas.

1.3 Objetivos del proyecto

El presente trabajo tiene como objetivo principal el desarrollo de un modelo relajado de optimización del flujo de potencia en redes de corriente alterna (AC-OPF), orientado en primera instancia a su aplicación en redes malladas de tamaño mediano, con la esperanza de poder ser escalado a redes de mayor tamaño.

Este modelo tiene la capacidad de determinar la distribución más eficiente de los flujos potencia activa ($fP_{ij,c}$) y de potencia reactiva ($fQ_{ij,c}$) a través de las diferentes líneas de la red: cada posible conductor c entre la pareja de nudos i y j . Todo ello garantizando el cumplimiento de las restricciones operativas y físicas del sistema. Para conseguirlo, se han centrado los esfuerzos de la siguiente manera:

- Desarrollar un modelo matemático AC-OPF con relajación SOCP: formular un modelo basándose en los últimos desarrollos identificados tras estudiar el estado del arte que, respetando las restricciones operativas de la línea y sustentándose en el Branch Flow Model, explora las potenciales aplicaciones de la relajación de segundo orden convexa (SOCP).
- Implementación y validación en sistemas de prueba: Aplicar la formulación desarrollada en sistemas eléctricos de referencia para comprobar su adecuación. Para ello se utilizarán herramientas como Python/Pyomo para la modelización, la librería PyPower para obtener soluciones de referencia, y el solver no lineal IPOPT para resolver los modelos.
- Validación y análisis de anomalías: Estudiar posibles anomalías en los resultados (por ejemplo, energía no suministrada – ENS –, discrepancias entre el modelo y la referencia, etc.) y analizar cómo la incertidumbre en la generación renovable podría incorporarse en futuros modelos de operación a gran escala.

1.4 Metodología de trabajo

Con el objetivo de alcanzar la meta final del proyecto, se establece una estrategia metodológica progresiva.

- Modelado de flujo de potencia base (ACPF sin optimización): En primer lugar, se desarrolla un modelo de flujo de potencia (ACPF, AC Power Flow) para validar que las ecuaciones utilizadas para modelar los flujos de potencia activa y reactiva sean correctas. Este sistema de ecuaciones es subdeterminado (tiene más variables que ecuaciones), por lo que es necesario fijar ciertas variables en cada nudo según su tipo (PV, PQ o slack) para obtener una solución físicamente coherente. Este modelo preliminar no incluye ninguna función objetivo que optimizar ni todas las restricciones operativas de los generadores; su finalidad es encontrar un estado operativo viable que satisfaga las leyes de Kirchhoff utilizando como variables los módulos y los ángulos de las tensiones de cada uno.

- Extensión al modelo AC-OPF clásico: una vez validada la consistencia de las ecuaciones de potencia y asegurada la convergencia del ACPF, se procederá a completar el modelo AC-OPF. Para ello, se eliminarán las fijaciones iniciales de variables y se introducirán progresivamente las restricciones físicas (por ejemplo, límites de generación, tensiones de nudo, límites térmicos de líneas) junto con la función objetivo que representa los costes operativos. Así se configura un problema de optimización completo que busca el mínimo coste de operación respetando todas las limitaciones técnicas.
- Introducción de la relajación convexa SOCP: finalmente, una vez desarrollado el AC-OPF no convexo, se implementa una relajación convexa del modelo mediante programación cónica de segundo orden (SOCP). Esta versión relajada del problema se resuelve y sus resultados se comparan tanto con los del modelo AC-OPF original no convexo, como con las soluciones de referencia obtenidas con PyPower. De este modo, es posible evaluar cuánto se aproxima la relajación convexa al óptimo real y cuánta reducción de carga computacional se logra.

Siguiendo esta metodología escalonada, primero se comprueba la capacidad del solver para manejar el problema AC original, y luego se evalúa el desempeño del enfoque relajado. Cada etapa sirve para verificar la robustez de la siguiente: el ACPF verifica datos y ecuaciones, el AC-OPF asegura la correcta incorporación de restricciones y costos, y la relajación SOCP evalúa la eficiencia computacional sin perder demasiada precisión respecto al modelo original.

2. Metodologías actuales de formulación y optimización del flujo de potencia

Flujo óptimo de potencia en corriente alterna (AC-OPF)

Planificar, operar o expandir una red eléctrica requiere tomar decisiones complejas que respeten las leyes físicas del sistema, incluyendo tensiones, flujos de potencia activa y reactiva, así como pérdidas energéticas. Para abordar este desafío, se emplea el modelo de Flujo Óptimo de Potencia en corriente alterna (ACOPF), que permite representar con precisión el comportamiento real de la red. Esta herramienta identifica el punto de operación más eficiente posible, minimizando costos, pérdidas en las líneas y el vertido de energía renovable, todo ello cumpliendo con las restricciones técnicas y físicas del sistema.

El problema de optimización, formulado por primera vez en 1962 por el ingeniero J. Carpentier, tenía como objetivo encontrar puntos de operación del sistema eléctrico que no solo fueran técnicamente viables, sino también económicamente óptimos (CARP62).

En este trabajo se plantea con un objetivo principal similar. Minimizar el coste total de operación del sistema eléctrico mientras se satisface el balance de demanda en cada nudo del sistema sin violar los límites de operación de los generadores ni de las líneas.

El coste total se encuentra directamente vinculado a la potencia activa generada por cada uno de los generadores conectados a la red. Matemáticamente, esta función objetivo puede expresarse como:

$$\min \sum_{i \in \delta} (C_{g2,i} P_{g,i}^2 + C_{g1,i} P_{g,i} + C_{g0,i})$$

- δ : representa el conjunto de todos los nodos de la red que cuentan con unidades de generación activa.
- $C_{g,i}$ es el coste de generación (en €/MW o unidad monetaria equivalente) asociado al generador ubicado en el nodo i . $C_{g0,i}$, $C_{g1,i}$ y $C_{g2,i}$ son los coeficientes cuadráticos de la curva de coste: un término independiente, un término lineal y un término cuadrático, respectivamente.
- $P_{g,i}$ indica la potencia activa generada (en MW) por dicho generador g en el nodo i .

La minimización del coste de generación eléctrica, definida en la función objetivo anterior, debe realizarse respetando un conjunto de restricciones físicas y operativas fundamentales para garantizar la viabilidad técnica del sistema, como se detalla en *Monticelli, 1983* (MONT83). Estas restricciones modelan el comportamiento real de la red eléctrica y aseguran que la solución obtenida sea factible, es decir, cumpla las relaciones físicas que existen entre las magnitudes involucradas.

En primer lugar, las variables de decisión deben satisfacer las ecuaciones de balance nodal tanto para (1) potencia activa como para (2) potencia reactiva, que aseguran la conservación de energía en cada nodo de la red. Dichas ecuaciones planteadas del Branch Flow Model (BFM) se derivan de las ecuaciones clásicas del flujo de potencia activa (P_i) y reactiva (Q_i) en corriente alterna. Estas ecuaciones de origen, que relacionan las inyecciones de potencia con las tensiones y los ángulos en los nodos, se expresan de forma general como:

$$P_i = V_i \sum_{j \in N_i} V_j (G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j))$$

$$Q_i = V_i \sum_{j \in N_i} V_j (G_{ij} \sin(\theta_i - \theta_j) - B_{ij} \cos(\theta_i - \theta_j))$$

Las ecuaciones, que trasladan el foco de la inyección nodal hacia el flujo por línea, utilizadas siguen el mismo planteamiento por nudo: Generación – Demanda + Shunt local = Flujo que sale – Flujo que entra. Este se expresa mediante las siguientes ecuaciones:

$$P_{g,i} - P_{d,i} - g_{sh,i} V_i^2 - \sum_{j:(i,j) \in \mathcal{E}} (f P_{ij,c} + r_{ij,c} I_{ij,c}^2) + \sum_{j:(j,i) \in \mathcal{E}} f P_{ji,c} = 0 \quad (1)$$

$$Q_{g,i} - Q_{d,i} + b_{sh,i} V_i^2 - \sum_{j:(i,j) \in \mathcal{E}} (f Q_{ij,c} + x_{ij,c} I_{ij,c}^2 - \frac{b_{sh,i}}{2} V_i^2) + \sum_{j:(j,i) \in \mathcal{E}} (f Q_{ji,c} + \frac{b_{sh,i,j,c}}{2} V_i^2) = 0 \quad (2)$$

Asimismo, es necesario que las variables cumplan con un conjunto de restricciones a nivel de línea, que incluyen: (3) caída de tensión, (4) diferencias de ángulo, (5) corriente.

$$a_{ij}^2 V_i^2 - V_j^2 - Z_{ij,c}^2 I_{ij,c}^2 - 2(r_{ij,c} f P_{ij,c} + x_{ij,c} f Q_{ij,c}) = 0 \quad (3)$$

$$a_{ij}^2 V_i V_j \sin(\theta_i - \theta_j) = x_{ij,c} f P_{ij,c} - r_{ij,c} f Q_{ij,c} \quad (4)$$

$$I_{ij,c}^2 = \frac{f P_{ij,c}^2 + f Q_{ij,c}^2}{V_{ij,c}^2} \quad (5)$$

Finalmente, deben cumplirse las restricciones de seguridad y límites físicos de los elementos del sistema, tales como (6) las restricciones térmicas, las tensiones en nudo y los márgenes operativos de generación.

$$\sqrt{f P_{ij,c}^2 + f Q_{ij,c}^2} \leq S_{ij,c}^{max} \quad (6)$$

$$V^{min} \leq V \leq V^{max} \quad (7)$$

$$P_{g,n}^{min} \leq P_{g,n} \leq P_{g,n}^{max} \quad (8)$$

$$Q_{g,n}^{min} \leq Q_{g,n} \leq Q_{g,n}^{max} \quad (9)$$

Este sistema es altamente no convexo a causa de las restricciones de balance nodal (1) y (2), las restricciones físicas de línea (3)-(5) y la térmica (6). Las primeras, contienen productos bilineales (V_i^2 , $I_{ij,c}^2$) y términos no lineales y no convexos ($f P_{ij,c}$, $f Q_{ij,c}$). De la misma manera, la no convexidad se debe a productos de variables y funciones trigonométricas como: $\sin(\theta_i - \theta_j)$. A su vez, la restricción térmica, aunque puede transformarse en una desigualdad cuadrática, no define un conjunto convexo en todos los espacios de variables, lo que refuerza el carácter no convexo global del problema AC-OPF.

En esencia, al ser no convexo, este problema presenta múltiples desafíos desde un punto de vista físico y computacional. Entre ellos, la admisión de múltiples soluciones locales, lo que implica que algoritmos tradicionales como los métodos de Newton-Raphson o los solvers de programación no lineal (NLP) pueden converger en un óptimo local que no siempre es la solución global. Además, la calidad de resolución y la viabilidad de la solución dependen del punto inicial seleccionado (dependencia de inicialización). A esto se suma el hecho de que no existen garantías formales de alcanzar la solución óptima global ni de unicidad, lo que limita la fiabilidad del enfoque clásico.

En definitiva, la naturaleza no convexa del problema dificulta su resolución mediante métodos convencionales, por lo que resulta fundamental investigar y desarrollar formulaciones alternativas y técnicas de relajación que permitan abordarlo de manera exacta o cuasi-exacta, garantizando soluciones globalmente óptimas y físicamente viables.

2.1 Métodos tradicionales de resolución

Históricamente, se han desarrollado diversas formulaciones con el objetivo de guiar a los solvers hacia soluciones globalmente óptimas, evitando la convergencia a óptimos locales característicos de la naturaleza no convexa del problema. Entre las más destacadas se encuentran:

Aproximación por flujo de potencia óptimo en corriente continua (DC-OPF)

El DC-OPF (Direct Current Optimal Power Flow) es esencialmente una aproximación linealizada del AC-OPF. Es ampliamente utilizada en aquellos contextos operativos en los que se requiere una solución adecuadamente precisa y ante todo rápida.

La simplificación del AC-OPF se basa en un conjunto de hipótesis restrictivas que permiten reducir y linealizar el problema original. Estas aproximaciones transforman el modelo completo en una versión más sencilla y convexa, que, aunque presenta limitaciones en términos de fidelidad física a la ley de Kirchhoff, resulta útil como referencia inicial o punto de partida para metodologías más avanzadas.

Las simplificaciones físicas previamente mencionadas centran la hipótesis del DC-OPF, según se describe en (MARL17), de la siguiente forma:

- Ángulos de fase pequeños asumidos: Esto concluye que $\theta_i - \theta_j$ también es insignificante. En consecuencia se puede asumir: $\sin(\theta_i - \theta_j) \approx \theta_i - \theta_j$.
- No se considera ni se optimiza la potencia reactiva Q_i
- Se ignoran las pérdidas ohmicas en las resistencias de las líneas y solo se estudian las partes imaginarias de las impedancias (reactancia x_{ij}) de las líneas.
- Se normaliza y se toma constante el módulo de la tensión por los nudos $|V_i| = 1 \forall i$, es decir que, al igual que la potencia reactiva, ni se optimiza ni modela las magnitudes de la tensión.

Con todo, la formulación de los flujos de potencia se reduce a lo siguiente:

$$f_{P_{ij,c}} = \frac{\theta_i - \theta_j}{x_{ij,c}}$$

A pesar de la simplicidad y eficiencia computacional que aporta a las restricciones del problema de optimización de flujos, este enfoque presenta limitaciones técnicas significativas cuando se aplica más allá del análisis preliminar o de estudios de mercado. La omisión de la potencia reactiva, la no consideración de la variabilidad en la magnitud de las tensiones y la ausencia de pérdidas ohmicas evidencian que no refleja con precisión el comportamiento real de un sistema eléctrico.

Estas simplificaciones, como se menciona anteriormente, solo son aplicables en casos muy concretos. En particular, pierden validez en escenarios donde la red está altamente cargada, contiene líneas con alto cociente $\frac{r}{x}$, o está sujeta a restricciones de tensión estrictas, como ocurre en redes de distribución o configuraciones con generación distribuida (MARL17). A pesar su utilidad en ciertos casos, diversos estudios han demostrado que la solución obtenida por DC-OPF

puede diferir significativamente de la solución óptima del modelo AC, comprometiendo tanto la precisión económica como la seguridad operativa del sistema (PAN20).

2.1.1 Métodos de programación no lineal (NLP)

Este método es el más extendido en el ámbito académico y profesional, ya que es capaz de mantener el modelo completo sin limitaciones estructurales ni simplificaciones excesivas. Entre los más destacables se encuentran:

- IPOPT (Interior Point OPTimizer), que es el empleado en este trabajo, es un software robusto y de código abierto. Es altamente compatible con entornos como Pyomo, AMPL o JuMP. Su principal ventaja se encuentra en su versatilidad y en su capacidad de trabajar con sistemas de tamaño medio y grande.
A la hora de resolver la optimización, utiliza una estrategia primal-dual. Es decir, calcula simultáneamente las variables a optimizar (primales) y las variables asociadas a las restricciones (duales). En cada iteración, decide si avanzar o no mediante una estrategia flexible llamada “filtro”, que permite avanzar siempre que se mejore el coste o se acerque a una solución factible. Esta flexibilidad mejora notablemente la estabilidad del algoritmo, sobre todo en problemas mal condicionados o con puntos iniciales alejados de la solución.
- Otros solvers no lineales que se utilizan en la actualidad son:

KNITRO destaca por su eficacia en problemas muy restringidos, en otras palabras, aquellos sistemas de topología compleja (nudos muy acoplados o generadores casi en sus límites) o de alta sensibilidad (variaciones grandes por pequeños cambios en la demanda). Esto suele dificultar la convergencia de los algoritmos numéricos y puede generar soluciones inestables o erróneas. Su rendimiento supera normalmente al de IPOPT en la velocidad y estabilidad de resolución del AC-OPF. Sin embargo, siendo un solver comercial, su licencia es de pago.

CONOPT, un solver maduro que utiliza métodos de factorización secuencial y aproximaciones cuadráticas sucesivas para encontrar soluciones viables. Se destaca por su buena tolerancia numérica, lo que significa que puede manejar bien errores de redondeo e inexactitudes en los datos, logrando converger incluso en condiciones poco ideales. No obstante, su desempeño puede verse limitado en problemas no convexos o con múltiples óptimos locales, como el ACOPF, donde corre el riesgo de converger a soluciones subóptimas si no se le proporciona un buen punto de partida.

BELTISTOS, un solver comercial cerrado de alto rendimiento orientado a aplicaciones industriales. Si bien destaca por su eficiencia en la resolución de problemas de optimización no lineales a gran escala, presenta limitaciones importantes en términos de flexibilidad. Al tratarse de una herramienta de código cerrado, no permite la modificación interna de la formulación ni la incorporación de restricciones especializadas, como las cíclicas. Asimismo, carece de compatibilidad con entornos de modelado ampliamente utilizados en el ámbito académico, como Pyomo, JuMP, MATPOWER o PowerModels, lo que restringe su aplicabilidad en procesos de investigación y desarrollo personalizados (KARD22).

En vista de las propiedades no lineales y no convexas del problema del AC-OPF, el uso de solvers de Programación No Lineal demuestra ser un buen punto de partida para resolver la cuestión, sin necesidad de adaptaciones estructurales.

Cabe enfatizar que estos métodos constituyen únicamente una solución de referencia o implementación primaria, ya que, en la práctica, requieren una inicialización viable para evitar converger hacia soluciones localmente óptimas. Además, presentan limitaciones de escalabilidad, ya que la mayoría de estos enfoques pueden experimentar problemas de convergencia o tiempos

computacionales excesivos cuando se aplican a redes eléctricas de gran tamaño o con topologías complejas.

2.2 Relajaciones convexas del AC-OPF

Dado que el objetivo es alcanzar soluciones globalmente óptimas con garantías de estabilidad numérica, se hace imprescindible el desarrollo de técnicas que permitan su reformulación o relajación en términos convexas, habilitando así la aplicación de solvers eficientes y escalables dentro del marco de la optimización convexa.

Entre las distintas técnicas de relajación que permiten abordar la no convexidad del problema, destacan especialmente la Programación Semidefinida (SDP) y la Programación Cónica de Segundo Orden (SOCP), por su capacidad para aproximar el AC-OPF mediante formulaciones convexas resolubles de forma eficiente.

2.2.1 Programación semidefinida (SDP)

Según se expone en el trabajo de Erik Francisco Álvarez Quispe, desarrollado en la Universidad Estadual de Campinas, la Programación Semidefinida (SDP) constituye una herramienta prometedora para aproximar soluciones a problemas de optimización no convexas en sistemas eléctricos, mediante su transformación en problemas convexas de tipo SDP (ALVA19).

Cuando se cumplen ciertas condiciones de convexidad, estas reformulaciones permiten, dependiendo del tipo de relajación, establecer cotas inferiores sólidas o alcanzar soluciones cuasi-óptimas globales del problema original, lo que ha motivado un creciente interés en la investigación aplicada en este ámbito.

La relajación semidefinida es una extensión de la programación lineal en la que las variables de decisión conforman matrices simétricas, y una de las restricciones consiste en que las matrices sean semidefinidas positivas.

➤ Matrices semidefinidas positivas

Una matriz semidefinida es una matriz simétrica (o hermitica, si es compleja) que no tiene autovalores negativos; es decir, todos sus autovalores deben ser mayores o iguales a cero. Formalmente, una matriz $X \in R^{n \times n}$ es considerada semidefinida positiva (denotado como $X \geq 0$) si satisface la condición (GART12):

$$x^T A x \geq 0 \text{ para todo } x \in R^n, x \neq 0$$

$$X = x x^T$$

Esta propiedad es esencial para mantener la convexidad en contextos de optimización porque implica que la matriz no “invierte” el espacio en ninguna dirección.

Otro aspecto relevante en el análisis de las relajaciones semidefinidas (SDP) es el rango de la matriz de decisión A , ya que este parámetro determina la exactitud de la relajación y, por consiguiente, la viabilidad de la solución respecto al problema original no convexo.

En particular, si la matriz resultante cumple: $X \geq 0$ y $\text{rango}(X) = 1$, entonces se puede concluir que la relajación es exacta. En ese caso, la solución obtenida a través de la SDP no solo es óptima para el problema relajado, sino también factible y óptima para el problema original.

En cambio, si la matriz A obtenida tiene rango mayor que 1, ($\text{rango}(X) > 1$), la relajación se considera inexacta. Aunque la solución sigue siendo convexa y computacionalmente eficiente, solo representa una cota inferior del valor óptimo del problema original. En tales situaciones, se requiere aplicar métodos de posprocesamiento, como técnicas de extracción espectral, randomización o penalizaciones adicionales, con el fin de generar una solución aproximada que respete las restricciones físicas y sea viable desde el punto de vista del modelo original.

Por ello, utilizando el siguiente problema de optimización a modo de ejemplo:

$$\begin{aligned} \min_{x \in R^n} \quad & x^T Q x \\ \text{sujeto a} \quad & \underline{S}_{ij,c} \leq x^T Y x \leq \overline{S}_{ij,c} \\ & \underline{v} \leq |x|^2 \leq \overline{v} \end{aligned}$$

Donde:

- El vector x designa las variables restringidas a optimizar, en este caso las tensiones.
- La matriz Q , una matriz cuadrada ($n \times n$) no semidefinida positiva, hace que el problema sea no convexo.
- La matriz Y , representa la matriz de admitancias de la red, la cual no es hermítica ni semidefinida positiva, lo que implica que no posee simetría compleja ni garantiza propiedades de convexidad.

Para transformar las expresiones cuadráticas no convexas en expresiones lineales en matrices, es necesario hacer uso de la siguiente matriz:

$$X = x x^T$$

Puesto que $x^T Q x$ es un escalar, éste equivale a su traza:

$$x^T Q x = \text{tr}(x^T Q x) = \text{tr}(Q x x^T) = \text{tr}(Q X)$$

Por lo tanto el problema original de optimización termina siendo equivalente al siguiente problema SDP (linear y convexo):

$$\begin{aligned} \min_{x \in R^n} \quad & \text{tr}(Q X) \\ \text{sujeto a} \quad & \underline{S}_{ij,c} \leq \text{tr}(Y X) \leq \overline{S}_{ij,c} \\ & \underline{v} \leq \text{tr}(A X) \leq \overline{v} \\ & X = x x^T ; X \geq 0 \end{aligned}$$

Con todo, la programación semidefinida ha demostrado ser una herramienta versátil para problemas no convexas en sistemas eléctricos, especialmente si se trata de la formulación del AC-OPF. Principalmente porque es capaz de convertir productos bilineales y relaciones cuadráticas en expresiones lineales, en matrices. Ello, facilita significativamente la obtención de soluciones globalmente óptimas bajo ciertas condiciones previamente mencionadas (VAND96). También destaca por su flexibilidad, la cual ha permitido su uso en áreas de diversa índole como la

planificación de expansión (GHAD19) (ALVA24), el despacho económico (MADR99), o el análisis de estabilidad (NGUY19).

No obstante, este método de relajación presenta limitaciones significativas. En primer lugar, la complejidad computacional que conlleva operar con matrices de gran tamaño, ya que el número de variables aumenta cuadráticamente. Ello conlleva que no sea un método escalable o aplicable a redes de gran tamaño. Asimismo, si la matriz de decisión resultante en la relajación semidefinida positiva (SDP) tiene un rango mayor que uno, la solución obtenida no corresponde directamente a una solución factible del problema original, lo que indica que la relajación no es exacta. Por último, en comparación con otros métodos como la SOCP, la SDP presenta una mayor dificultad de interpretación directa de los resultados en términos eléctricos.

2.2.2 Programación Cónica de Segundo Orden (SOCP)

La Programación Cónica de Segundo Orden (SOCP) se ha consolidado como una técnica de relajación convexa muy eficiente para abordar el problema de flujo óptimo de potencia en corriente alterna (AC-OPF). En términos formales, un problema es apto para formulación SOCP cuando su región factible puede expresarse mediante la intersección de un conjunto afín (ecuaciones y desigualdades lineales) con el producto cartesiano de conos de segundo orden (también conocidos como conos de Lorentz). En otras palabras, las restricciones no lineales del AC-OPF deben reformularse de modo que tomen la forma de normas euclídeas acotadas por expresiones lineales.

Afortunadamente, varias de las ecuaciones no convexas del AC-OPF admiten esta reformulación, lo que ha permitido desarrollar modelos convexos basados en SOCP que abordan con gran fidelidad el comportamiento de los flujos de potencia en redes AC reales.

Este es un punto clave que derivan en que este enfoque de relajación mediante SOCP se complementa de forma natural con el Branch Flow Model (BFM). Al usar el BFM, las restricciones originalmente no convexas del AC-OPF (como las relaciones cuadráticas entre potencias, tensiones e intensidades) pueden reescribirse en términos que encajan directamente en conos de segundo orden. Por ejemplo, considérese la restricción térmica de una línea (límite de corriente o de potencia aparente).

$$fP_{ij,c}^2 + fQ_{ij,c}^2 \leq V_i^2 I_{ij,c}^2$$

$$\sqrt{fP_{ij,c}^2 + fQ_{ij,c}^2} \leq V_i I_{ij,c}$$

Para imponer la convexidad se hace uso del cono Lorentziano estándar:

$$\sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \leq x_{n+1}$$

Esta restricción impone que la norma de un grupo de variables no exceda otra variable linealmente dependiente de ellas (convierte igualdades en desigualdades), en este caso, el valor máximo de corriente permitido por cada línea.

➤ Aplicaciones prácticas y ventajas de la formulación SOCP

Gracias a estas reformulaciones, este tipo de relajación abarca problemas más generales que incluyen restricciones cuadráticas convexas. Gracias a esta flexibilidad, la SOCP se ha aplicado con éxito en numerosas y variadas disciplinas de la actualidad. Por ejemplo, el cono de Lorentz generaliza muchas estructuras matemáticas comunes en ingeniería y finanzas

(ALIZ03). Además, engloba un conjunto más amplio de problemas no incluidos en estas categorías, como los que surgen en el diseño de filtros FIR, optimización de portafolios, planificación robusta y control de sistemas eléctricos.

Desde el punto de vista computacional, la principal ventaja de SOCP sobre otras relajaciones convexas (en particular sobre la Programación Semidefinida, SDP) reside en su menor complejidad y mejor escalabilidad para redes de gran tamaño. Los algoritmos de punto interior pueden explotar la estructura dispersa de las restricciones cónicas (normas) de forma mucho más eficiente que en el caso de restricciones semidefinidas (resolución de sistemas lineales con matrices semidefinidas positivas), permitiendo resolver problemas con miles de nudos en tiempos compatibles con la operación en línea. Además, las soluciones de la relajación SOCP, al estar más estructuradas (por ejemplo, al proveer directamente tensiones y flujos), suelen ser más fáciles de interpretar físicamente que las de una relajación SDP. Todo ello favorece su escalabilidad a sistemas de gran tamaño.

Desde el punto de vista teórico, SOCP también presenta una relación jerárquica interesante: puede verse como un caso particular de SDP, pero también como una extensión de los programas lineales. Esta dualidad permite conservar la generalidad de modelos robustos y no lineales, sin incurrir en los costos computacionales de trabajar con matrices semidefinidas grandes (VAND96). Además, los problemas formulados como SOCP admiten métodos de resolución altamente estructurados, como los métodos primal-dual de punto interior desarrollados por Nesterov y Todd (NEST97), que garantizan convergencia polinomial y permiten aprovechar la estructura dispersa del problema (ALIZ03).

Esto hace de SOCP una herramienta atractiva para ingenieros, ya que el resultado conserva un significado claro en términos de variables de red (potencias, tensiones, corrientes), facilitando su validación y ajuste. Ha resultado especialmente ventajoso en contextos como: (i) modelado de redes de distribución con alta penetración de renovables, (ii) control de tensión y potencia reactiva, y (iii) planificación conjunta de redes de transporte y distribución (HUAN16). Con todo, demuestra ser una herramienta de resolución factible u óptima incluso en redes malladas.

➤ Limitaciones y consideraciones críticas

No obstante, a pesar de sus ventajas, es importante destacar limitaciones y aspectos críticos del enfoque SOCP para AC-OPF.

Principalmente, la exactitud de la relajación SOCP puede depender fuertemente de las condiciones operativas del sistema y de componentes de la red (topología, parámetros de los generadores y otros componentes). Se ha demostrado que ciertas suposiciones (como acotar muy estrechamente los niveles de tensión o evitar inyecciones reactivas elevadas) facilitan que la relajación sea exacta (HUAN16). Fuera de esas condiciones, especialmente en escenarios de cargas muy elevadas o generación distribuida con altos flujos reactivos, el modelo puede requerir un reforzamiento adicional (añadiendo, por ejemplo, límites de ángulo entre nodos adyacentes o términos de penalización en la función objetivo que incentiven) para obtener soluciones físicamente válidas

A su vez, la relajación no siempre garantiza una solución factible para el problema original. En concreto, si la condición térmica $fP_{ij,c}^2 + fQ_{ij,c}^2 \leq V_i^2 I_{ij,c}^2$ no se satisface en el óptimo para alguna línea, la solución obtenida no corresponderá exactamente a un flujo de potencia realizable.

➤ Desafíos en redes malladas

En una red mallada, un ciclo es un camino cerrado que comienza y termina en el mismo nudo, pasa por al menos tres nudos o líneas diferentes, y no repite ningún nodo ni línea, excepto el de inicio y final.

La solución del SOCP podría violar la ley de KVL (sumatorio de ángulos) alrededor de algún ciclo (lazo cerrado en una red mallada), produciendo un flujo que matemáticamente minimiza el costo pero que físicamente no puede existir (KOCU14).

En redes malladas (aquellas en las que existe más de un camino entre dos nudos), no alcanzar una solución factible puede suceder si no se agregan suficientes restricciones que refuercen la relación entre las variables de los nodos que se encuentran en una malla, dichas condiciones se conocen como: restricciones cíclicas.

La Ley de KVL previamente mencionada se refiere a la Ley de Tensiones de Kirchhoff. Dicha ley establece que la suma de tensiones debe ser cero, en consecuencia, las diferencias de ángulos de tensión a lo largo del recorrido del ciclo debe ser cero:

$$\sum_{(i,j) \in C} \theta_{ij} = \sum_{(i,j) \in C} \theta_i - \theta_j = 0 \text{ (o múltiplo de } 2\pi)$$

Siendo $C \in K_e$ (el conjunto de ciclos existentes dentro del sistema de nudos, N). Esta restricción se debe imponer sobre dicha base fundamental de ciclos para que se cumpla en cualquier otro recorrido de la red.

El número de ciclos sobre los que imponer dicha restricción se obtiene de la siguiente manera: $K_e = x_{líneas} - n_{nudos} + 1$. Por ejemplo, en los casos de estudio, se ha de imponer sobre:

$$\text{caso 3 nudos: } 3 - 3 + 1 = 1 \text{ ciclo}$$

$$\text{caso 14 nudos: } 20 - 14 + 1 = 7 \text{ ciclos}$$

$$\text{caso 30 nudos: } 41 - 30 + 1 = 12 \text{ ciclos}$$

La consecuencia de no introducir dichas restricciones en los ciclos es un gap de dualidad: el valor óptimo de la relajación es inferior al del AC-OPF real (porque ha explorado un espacio factible más amplio al no imponer todas las no convexidades), reflejando una brecha entre el óptimo primal (no convexo) y dual (convexo). En la práctica, este gap indica que la solución SOCP es optimista pero debe corregirse.

Una estrategia común para lidiar con esta situación es utilizar la solución SOCP como punto de partida (*warm start*) de un solver no lineal local (por ejemplo, IPSO o Newton-Raphson) para obtener una solución factible cercana, aprovechando que la relajación ofrece ya una buena aproximación. Otra estrategia explorada es desarrollar algoritmos heurísticos de recuperación de factibilidad que, dados los resultados del SOCP, ajustan gradualmente las variables (por ejemplo, inyectando pequeñas pérdidas adicionales o modificando ángulos) hasta cumplir todas las ecuaciones AC exactas (YUAN19).

Estos métodos han tenido éxito en convertir soluciones relajadas en soluciones utilizables, aunque a costa de un esfuerzo computacional adicional. Sin embargo, aun no siendo exacta, la relajación SOCP produce costos objetivos muy cercanos al óptimo real lo cual la hace valiosa para propósitos prácticos donde una pequeña suboptimalidad puede ser aceptable a cambio de una reducción drástica del tiempo de cómputo, ya que a veces es mejor dicha solución a no poder resolver el problema NLP en tiempos razonables y necesarios.

Ejemplo ilustrativo

Para ilustrar con mayor claridad el funcionamiento y la identificación de los ciclos en una red eléctrica, se propone el siguiente ejemplo basado en una red simplificada de siete nudos, cuya topología se presenta en la imagen siguiente. Aplicando la fórmula mencionada anteriormente, esta red específica contendría tres ciclos independientes.

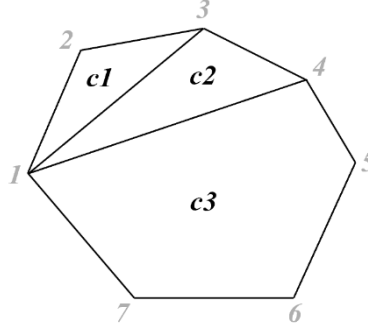


Figura 1: Estructura de 7 nudos (ejemplo desarrollado por la autora)

Al implementar las restricciones cíclicas derivadas de la Ley de Tensiones de Kirchhoff, es necesario establecer un sentido (o dirección) para recorrer cada ciclo, decisión que influye directamente en el cálculo de las diferencias angulares. Debido a que diferentes ciclos pueden compartir líneas comunes, algunas diferencias angulares deberán considerarse en sentidos opuestos según el ciclo que se analice. Dado que cada variable de diferencia angular ($\theta_{ij} = \theta_i - \theta_j$) se define siempre en un único sentido (sentido horario), se utiliza un signo negativo para indicar cuando dicha diferencia se evalúa en dirección contraria a la definida originalmente ($\theta_{ji} = \theta_j - \theta_i = -\theta_{ij}$).

Todo lo anterior conduce al procedimiento detallado a continuación para el caso específico analizado:

$$\text{ciclo 1: } \theta_{21} + \theta_{32} + \theta_{13} = 0$$

$$\text{ciclo 2: } \theta_{31} + \theta_{43} + \theta_{14} = -\theta_{13} + \theta_{43} + \theta_{14} = 0$$

$$\text{ciclo 3: } \theta_{41} + \theta_{54} + \theta_{65} + \theta_{76} = -\theta_{14} + \theta_{54} + \theta_{65} + \theta_{76} = 0$$

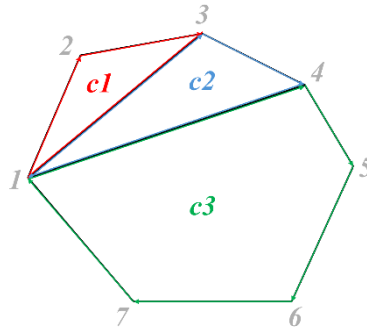


Figura 2: Estructura de 7 nudos dividida en ciclos (ejemplo desarrollado por la autora)

Este procedimiento garantiza coherencia en la definición y evaluación de las restricciones cíclicas, facilitando así una implementación robusta y libre de ambigüedades en el modelo matemático propuesto.

En conclusión, la Programación Convexa de Segundo Orden aplicada al OPF proporciona un marco potente y escalable para resolver problemas de flujo óptimo de potencia con gran eficiencia, manteniendo un alto grado de fidelidad al modelo AC original.

Su integración con el Branch Flow Model permite aprovechar estructuras cónicas naturales en las ecuaciones de potencia, y con las precauciones necesarias (restricciones cíclicas en redes malladas, técnicas de recuperación de factibilidad y validación de solución) se ha consolidado como una alternativa viable para la optimización operativa de redes eléctricas modernas.

2.3 Avances recientes

A medida que los sistemas eléctricos avanzan hacia configuraciones más distribuidas, con mayor penetración renovable y exigencias de respuesta en tiempo real, los modelos se han ido distanciando de los enfoques clásicos.

Los métodos tradicionales (e.g. métodos de Newton, programación lineal o cuadrática secuencial, puntos interiores) pueden converger a óptimos locales y son sensibles a la inicialización. En sistemas grandes, incluso las formulaciones simplificadas (como DC-OPF lineal) se emplean por razones de tiempo; por ejemplo, el operador MISO (Midcontinent ISO, ~45k nudos) resuelve a diario un OPF reducido debido a que el modelo completo (miles de nudos, líneas y escenarios de contingencia) no puede resolverse dentro de los límites de tiempo de mercado usando solvers deterministas convencionales.

Esto ha motivado en los últimos años (2018–2025) una intensa investigación en métodos avanzados para AC-OPF que logren mayor fidelidad física, escalabilidad computacional y óptimos más globales. Existen soluciones e innovaciones que han resultado ser exitosas en su propósito de obtener puntos de operación óptimos.

2.3.1 Algoritmos metaheurísticos e híbridos

Durante los últimos años, y en paralelo a las relajaciones convexas, la comunidad investigadora ha explorado una amplia gama de algoritmos metaheurísticos para resolver el AC-OPF de forma aproximada pero potencialmente global (POP22) (DIAB24).

Estas técnicas se basan en mecanismos de búsqueda aleatoria guiada que evitan caer en óptimos locales, aprovechando principios de inteligencia colectiva o evolución natural. Destacan:

- Particle Swarm Optimization (PSO, optimización por enjambre de partículas) modela el comportamiento cooperativo de un enjambre: múltiples partículas actualizan iterativamente sus posiciones en el espacio de búsqueda en función de su mejor experiencia individual y de la mejor global del grupo, logrando una exploración eficiente del óptimo global.
- Algoritmos Genéticos (GA) imitan la selección natural operando sobre una población de soluciones codificadas, aplicando operadores de cruce y mutación para recombinar características y converger hacia mejores individuos en cada generación (HERB16).
- Recocido Simulado (SA, simulated annealing), emplean un proceso análogo al enfriamiento de metales, donde se aceptan temporalmente soluciones peores para evitar caer en óptimos locales. Al disminuir progresivamente esa tolerancia, el algoritmo explora de forma amplia al inicio y se afina hacia el final, mejorando la probabilidad de hallar una solución global.
- Evolución Diferencial (DE), algoritmo evolutivo que optimiza problemas buscando soluciones mediante la combinación lineal de individuos dentro de una población. En cada iteración, genera nuevos candidatos restando y sumando vectores aleatoriamente, y selecciona aquellos con mejor desempeño para la siguiente generación.

Gracias a estos enfoques, métodos metaheurísticos como PSO, GA o SA han logrado soluciones viables de buena calidad para el OPF en redes de tamaño medio, sin necesidad de convexificar el problema (POP22).

Adicionalmente, se han desarrollado algoritmos híbridos y de nueva generación que combinan fortalezas de diferentes metaheurísticas para mejorar la convergencia global. Ejemplos destacados incluyen ESDE-MC (Enhanced Self-Adaptive Differential Evolution with Mixed Crossover), una variante de evolución diferencial que ajusta dinámicamente sus parámetros y mezcla estrategias de cruce para evitar la convergencia prematura; IGSA (Improved Gravitational Search Algorithm), que refina el algoritmo de búsqueda gravitatoria inspirándose en la interacción de masas para acelerar su convergencia; o híbridos como PSO-DE, que fusiona PSO con Evolución Diferencial para equilibrar exploración y explotación (DIAB24). Más recientemente han

emergido algoritmos como el GBO (Gradient-Based Optimizer), un metaheurístico inspirado en métodos de gradiente que incorpora operadores aleatorios de escape local.

Estos enfoques avanzados aprovechan múltiples metáforas de búsqueda simultáneamente, logrando en tests reportados una mayor probabilidad de alcanzar el óptimo global del AC-OPF y mejores valores objetivo en problemas multiobjetivo. De hecho, trabajos recientes muestran que combinaciones híbridas como ESDE-MC, IGSA o PSO-DE pueden producir soluciones de calidad superior en comparación con métodos individuales, especialmente al integrar fuentes renovables y considerar múltiples objetivos operativos (POP22) (SHAR25). Gracias a que evalúan directamente las ecuaciones originales de flujo de potencia, todos estos algoritmos metaheurísticos garantizan además que las soluciones obtenidas respeten las restricciones físicas de la red (leyes de Kirchhoff de tensiones y corrientes, KVL/KCL) sin necesidad de ajustes posteriores, a diferencia de ciertas relajaciones convexas que pueden violar dichas leyes al optimizar en un espacio expandido (SHAR25).

Ahora bien, a pesar de sus fortalezas en calidad de solución, los métodos metaheurísticos presentan limitaciones importantes que han dificultado su adopción en entornos industriales. En primer lugar, suelen conllevar un coste computacional elevado: requieren evaluar el flujo de potencia muchas veces (miles de iteraciones o población de soluciones) y, en sistemas de gran tamaño o tiempo real, pueden resultar demasiado lentos comparados con solvers matemáticos optimizados. Además, su desempeño depende de parámetros de calibración (tamaño de población, ratios de mutación, temperatura inicial, etc.) que deben ajustarse cuidadosamente para cada caso; una mala parametrización puede degradar la calidad de la solución o la velocidad de convergencia. Tampoco ofrecen una garantía estricta de optimalidad global ni de reproducibilidad exacta, dado su carácter aleatorio: dos ejecuciones pueden arrojar resultados ligeramente distintos, lo cual genera incertidumbre. Por estas razones, su aplicación en la industria todavía es limitada. Los operadores de red tienden a preferir métodos deterministas o convexas, cuyos tiempos de cómputo y comportamiento son más predecibles bajo estrés operativo.

En resumen, las metaheurísticas constituyen un valioso complemento en la investigación del AC-OPF –aportando soluciones factibles cercanas al óptimo global y sirviendo como referencia para evaluar otros enfoques– pero en la práctica actual suelen emplearse principalmente en estudios offline o de planificación, más que en el control en tiempo real de los sistemas eléctricos

2.3.2 Técnicas de aprendizaje automático y aprendizaje profundo en AC-OPF

Más recientemente, ha cobrado fuerza el enfoque de aplicar aprendizaje automático (ML) y aprendizaje profundo (DL) al AC-OPF (JIAN24). La motivación principal es aprovechar la capacidad de las redes neuronales y otros modelos de aproximar funciones complejas para predecir directamente (o asistir en el cálculo de) la solución óptima del OPF, con tiempos de cómputo muy inferiores una vez entrenados. Dado que las operaciones de red neuronal son muy rápidas en hardware moderno, un sistema bien entrenado podría generar setpoints óptimos casi en tiempo real, facilitando aplicaciones como control en intervalos cortos, simulaciones Monte Carlo masivas (estimaciones de comportamiento de un sistema complejo e incierto mediante la generación repetida de escenarios aleatorios) o, despacho predictivo con renovables.

No obstante, reemplazar un solver de optimización por un modelo de ML conlleva varios retos: (i) asegurar la factibilidad física (que las soluciones propuestas respeten KCL, KVL y restricciones), (ii) garantizar la generalización a condiciones no vistas durante la etapa de entrenamiento, y (iii) manejar la naturaleza no convexa de la región factible.

A pesar de ello, durante los últimos años (2018-2025) se ha observado la aparición de numerosos trabajos pioneros que buscan solucionar estas cuestiones.

Un enfoque inicial fue entrenar modelos de regresión supervisada que aprendan el “mapping” no lineal entre las cargas/estado de la red y la solución óptima (generaciones, tensiones). Por ejemplo, Baker (2019) utilizó un Random Forest entrenado sobre datos históricos de OPF para predecir

directamente los valores óptimos de generación y tensiones a partir del vector de demandas. Esta “predicción directa” logró soluciones cercanas al óptimo en fracciones de segundo, sirviendo como punto de arranque (warm start) para refinar con un solver convencional.

De forma análoga, se han utilizado redes neuronales totalmente conectadas (Fully Connected Neural Networks, FCNN) para aprender la relación entre los perfiles de carga y las soluciones óptimas de despacho. Estas redes funcionan tratando el sistema como una función matemática que, dada una entrada (por ejemplo, la demanda en cada nodo), devuelve directamente la salida (como las potencias generadas o tensiones nodales óptimas). Sin embargo, las FCNN convencionales no aprovechan la topología del sistema eléctrico, es decir, no consideran explícitamente cómo están conectados los nodos ni cómo se propagan los flujos de potencia entre ellos. Esto hace que funcionen como una “caja negra”: aprenden correlaciones estadísticas entre entradas y salidas sin entender la estructura física del sistema. Por ello, han sido utilizadas como aproximaciones de soluciones del AC-OPF.

Como consecuencia, estas redes requieren grandes volúmenes de datos de entrenamiento para cubrir la enorme cantidad de posibles configuraciones del sistema, incluyendo combinaciones diversas de estados operativos, restricciones activas (por ejemplo, líneas saturadas, límites de tensión alcanzados) y condiciones de red. Sin este volumen de datos, el modelo puede generalizar mal o fallar en escenarios no vistos previamente.

Para superar estas limitaciones, se han desarrollado modelos que integran conocimiento físico y estructural del sistema eléctrico. Entre ellos destacan las Redes Neuronales Gráficas (GNN), que modelan explícitamente nudos y líneas en su arquitectura, logrando así representar mejor la interacción entre nodos y capturar fenómenos como la Ley de Kirchhoff (KVL). Falconer et al. (2022) demostraron que las GNN ofrecen una mayor precisión y escalabilidad que las FCNN o CNN, especialmente en redes de gran tamaño.

En paralelo, se han propuesto enfoques physics-informed como el Aprendizaje por Refuerzo con Restricciones Convexas (CC-SAC), en el que el agente aprende técnicas de generación óptimas respetando un conjunto factible convexo y las restricciones físicas. También se ha investigado el uso de imitation learning para predecir directamente los subconjuntos de restricciones activas, acelerando la resolución del problema al reducir su dimensión efectiva. A su vez, se han implementado arquitecturas que combinan predicción rápida con una segunda etapa de cálculo de flujo de carga para garantizar la viabilidad AC exacta.

Un hito reciente es el desarrollo de CANOS (Neural AC-OPF Solver) por parte de DeepMind (2024), que utiliza una GNN profunda entrenada de forma end-to-end para predecir soluciones del AC-OPF en tiempos del orden de 33–65 ms, alcanzando costos dentro del 1 % del óptimo incluso en redes de hasta 10.000 nodos. El modelo escala linealmente con el tamaño de la red y ha mostrado robustez ante contingencias N–1, lo que lo convierte en un candidato viable para aplicaciones en control operativo (PILO24).

Aunque las redes neuronales aún presentan un pequeño gap en coste frente a los métodos clásicos, alcanzan errores del 1–3 % con tiempos de cómputo hasta 100 veces menores. En sistemas como IEEE-118, se han logrado aceleraciones de 6× a 22× manteniendo la factibilidad en la mayoría de los casos.

En síntesis, las técnicas de aprendizaje automático aplicadas al AC-OPF han pasado de ser una curiosidad a una prometedora área de desarrollo. Su fortaleza radica en la velocidad de cómputo una vez entrenados y en la posibilidad de integrar información histórica o pronósticos directamente en la toma de decisión óptima. Además, algunos enfoques, como el aprendizaje por refuerzo, ofrecen una vía para incluir incertidumbre y dinamismo (por ejemplo, resolución de OPF en tiempo real bajo fluctuaciones) de forma más natural que los métodos tradicionales. No obstante, asegurar confiabilidad y seguridad sigue siendo crucial: en entornos reales, cualquier solución propuesta por ML debe verificarse rigurosamente. A este fin, la tendencia es hacia mejorar la capacidad de generalización ante cambios topológicos y condiciones fuera del rango

entrenado y desarrollar métodos híbridos Machine Learning + físicos, donde la red neuronal hace gran parte del trabajo pesado de cómputo y un módulo de optimización o simulación se encarga de pulir la solución y certificar su viabilidad física antes de implementarla.

2.3.3 Mejoras en las relajaciones

Dado que este trabajo se centra en la aplicación de métodos de relajación del AC-OPF para que pueda ser aplicado casos de estudio no sólo académicos sino entornos industriales y escalables, es importante mencionar los avances realizados en estos métodos en los últimos años.

- En primer lugar, la relajación SDP, que eleva el problema AC-OPF a un espacio de matrices semidefinidas positivas que representan productos de tensiones complejas (GHAS18), proporciona un límite inferior muy ajustado al óptimo global. De hecho, se ha demostrado que en redes radiales o bajo ciertas condiciones (e.g. cargas moderadas), la solución de la SDP coincide con el óptimo físico (matriz de rango 1) (GHAS18).

Sin embargo, en redes malladas generales la relajación SDP puede ser no-exacta, produciendo soluciones no implantables físicamente (matriz de rango >1). Aunque sigue brindando una cota inferior útil, es necesario un paso adicional para recuperar una solución factible AC a partir de la matriz óptima. Algunas estrategias para esto incluyen “rank reduction” mediante perturbaciones aleatorias o penalizaciones en la función objetivo para forzar la matriz hacia rango 1. Otro límite que se halla es su alto coste computacional, como se ha mencionado con anterioridad, el tamaño crece cuadráticamente, por lo que su escala pobremente.

Para mejorar su escalabilidad, muchos estudios proponen descomponer el problema en submatrices (cliques) más pequeñas correspondientes a subgrafos (SLIW19). No obstante, elegir la descomposición óptima es crucial. Un estudio de 2019 evidenció que buscar la extensión chordal con el menor número de aristas no siempre conduce al mejor desempeño computacional. En muchos casos, resulta más eficaz construir una extensión que, aunque añada más conexiones ficticias, optimice la resolución del problema semidefinido. Este hallazgo supone un cambio de paradigma en la formulación de relajaciones SDP eficientes para redes eléctricas complejas.

Se ha de aclarar que: un grafo chordal se define como aquel en el que todo ciclo de cuatro o más nodos contiene al menos una cuerda, es decir, una arista que conecta dos nodos no consecutivos y rompe el ciclo. Esta propiedad permite descomponer el problema en submatrices (cliques) más manejables, mejorando la escalabilidad del algoritmo.

Con estas mejoras, SDP puede abordar sistemas de tamaño mediano (cientos de nodos) con tiempos aceptables, y sirve como módulo en algoritmos de optimización global: por ejemplo, métodos de branch-and-bound que usan SDP para acotar el óptimo global en subdivisiones del espacio, para certificar globalidad en casos pequeños (SLIW19).

- Por su parte, la SOCP es menos costosa que SDP (las restricciones de cono se resuelven con puntos interiores de manera similar a programaciones cuadráticas), y ha mostrado ser exacta en redes radiales (distribución) bajo supuestos razonables (GHAS18). Sin embargo, en redes malladas puede continuar siendo inexacta sin las restricciones cíclicas previamente explicadas. Estos refuerzos se han añadido recientemente. Por ejemplo, Kocuk et al. (2015) introdujeron cortes lineales válidos y formulaciones fuertes para asegurar soluciones factibles en presencia de ciclos (denominadas Strong SOCP relaxations). En esta formulación alternativa, también aplicada en este trabajo, se agregan ecuaciones que garantizan la consistencia de ángulos de tensión alrededor de cada ciclo fundamental de la red, evitando las soluciones “no-físicas” permitidas por la relajación estándar.

Un avance notable en 2020 fue el método de ciclos de 3 nodos propuesto en (MA20). La idea es descomponer cada ciclo mallado en lazos triangulares equivalentes añadiendo líneas

virtuales, e imponer que ciertas submatrices asociadas a esos triángulos sean semidefinidas positivas. Esto combina la fortaleza de SDP en pequeños subciclos con la rapidez de SOCP. El impacto práctico es significativo: la relajación convexa resultante logra gaps tan estrechos como los de una SDP completa, pero con mucho menor costo computacional. En pruebas con redes de referencia, en la mayoría de los casos, el gap se reduce al nivel de SDP (es decir, prácticamente exacto), y se escaló a sistemas de hasta miles de nudos con éxito. Estos resultados posicionan a las relajaciones SOCP reforzadas como alternativas viables para redes reales a gran escala.

En definitiva, no existe un único método que domine en todos los criterios, sino un conjunto de herramientas complementarias. Las relajaciones convexas (SDP/SOCP) brindan confianza en la optimalidad y bounds, siendo ideales para análisis offline, validación de soluciones y obtención de cotas —con mejoras recientes haciéndolas más prácticas en tamaño y garantizando factibilidad (KVL) en mallados—. Las técnicas metaheurísticas aportan flexibilidad y búsqueda global: son útiles cuando la función objetivo o restricciones salen de lo convencional (donde derivar gradientes es complejo) y en problemas donde un pequeño porcentaje de mejora en costo justifica esfuerzos computacionales significativos. Con algoritmos híbridos y nuevos metaheurísticos se ha incrementado la eficacia, aunque su lugar probablemente siga siendo como apoyo para refinar soluciones o explorar escenarios, más que reemplazo del despacho determinista. Finalmente, las soluciones basadas en machine learning abren una vía completamente nueva: prometen hacer posibles operaciones en tiempo real con modelos AC completos, incorporando además pronósticos y adaptándose a condiciones cambiantes. Aún requieren superar retos de confiabilidad y extrapolación, por lo que en el corto plazo es de esperar que se combinen con enfoques clásicos.

3. Descripción del modelo desarrollado

Este trabajo emplea el Branch Flow Model (BFM) como base de formulación, al ser el más compatible en la relajación para transformar el problema en SOCP y por centrarse explícitamente en las variables de flujo de potencia por línea.

La secuencia de modelos desarrollados sigue un proceso escalonado. En primer lugar, se verifica la viabilidad física del sistema mediante la resolución del flujo de potencia en corriente alterna (ACPF, por sus siglas en inglés: AC Power Flow), sin optimización. A continuación, se plantea el modelo de AC-OPF clásico (no lineal y no convexo), con el objetivo de minimizar los costes operativos y optimizar los flujos de potencia. Finalmente, se implementan relajaciones convexas mediante programación cónica de segundo orden (SOCP) junto con restricciones cíclicas, con el fin de obtener soluciones factibles que además sean exactas, evitando resultados subóptimos. La validez, escalabilidad y potencial de los modelos se ha evaluado en sistemas de prueba IEEE de 3, 14 y 30 nodos, respectivamente.

Para la implementación y resolución de estos modelos, se ha utilizado Pyomo, un entorno de modelado algebraico en Python que facilita la formulación de problemas de optimización complejos. Además, se ha empleado PYPOWER, una herramienta de análisis de sistemas de potencia en Python, para la validación y corroboración de los resultados obtenidos.

3.1 Flujo de potencia en corriente alterna, ACPF

Como punto de partida, se formula el modelo de flujo de potencia en corriente alterna (ACPF, por sus siglas en inglés: AC Power Flow), cuya finalidad es verificar la viabilidad del sistema eléctrico sin realizar una optimización. Este modelo resuelve el sistema de ecuaciones no lineales que describen el equilibrio de potencias activas y reactivas en cada nodo de la red. Al tratarse de un sistema inicialmente subdeterminado —es decir, con más variables que ecuaciones—, es necesario fijar ciertas magnitudes en cada nudo en función de su tipo (PV, PQ o slack) para obtener una solución físicamente coherente.

A diferencia del AC-OPF, el modelo ACPF no incorpora una función objetivo, ya que no busca minimizar costes económicos ni pérdidas técnicas: su propósito es exclusivamente encontrar un estado operativo viable, consistente con las leyes de Kirchhoff y las relaciones de flujo de potencia en régimen estacionario.

Desarrollar primero el ACPF ofrece varios beneficios antes de abordar el problema completo de optimización (AC-OPF). En primer lugar, permite verificar la consistencia de los datos del sistema (demanda, generación, topología) y detectar posibles configuraciones físicamente inviables o mal condicionadas. Además, proporciona una estimación inicial del estado del sistema (tensiones, potencias) que puede utilizarse como punto de partida (“warm start”) en métodos iterativos de resolución del OPF, mejorando la velocidad de convergencia y la estabilidad numérica del solver. También es útil como base de comparación para validar si las soluciones óptimas obtenidas posteriormente respetan las restricciones físicas del sistema.

3.1.1 Ecuaciones del modelo

Este modelo se fundamenta en las siguientes ecuaciones, cuya nomenclatura puede consultarse al inicio del documento junto con un esquema representativo de la línea de transporte de energía eléctrica utilizada como referencia.

Para formular el problema de flujo óptimo de potencia, es esencial distinguir entre las variables que el modelo ajusta para alcanzar la solución óptima y los parámetros que se mantienen fijos en

el código. A continuación, se detallan las principales variables de decisión y los parámetros utilizados en el modelo:

- Las variables de decisión son (i) el cuadrado de la magnitud de tensión en cada nudo (V_i^2), (ii) el ángulo de fase (Θ_i), (iii) la potencia activa y reactiva generadas en cada nudo (P_g y Q_g), (iv) los flujos de potencia activa y reactiva en cada línea ($fP_{ij,c}$ y $fQ_{ij,c}$) y (v) el cuadrado de la corriente de línea ($I_{ij,c}^2$).
- Los parámetros incluyen los datos del sistema base—potencia base (S_{base}), número de nudos y líneas—; los atributos por nudo: demanda ($P_{d,i}$ y $Q_{d,i}$), conductancia y susceptancia de shunt, tensión y ángulo iniciales (V_0 y Θ_0), generación inicial y tipo de nudo (P_{g0} y Q_{g0}); los atributos por línea de transporte (resistencia, $r_{ij,c}$, reactancia, $x_{ij,c}$, tap-ratio, desfase, impedancia al cuadrado y límite térmico normalizado, S_{max}); y la estructura de conectividad, definida mediante listas de índices de líneas entrantes y salientes para cada nudo.

A continuación, se presentan las ecuaciones implementadas en el código, que describen los balances de potencia activa y reactiva, las relaciones de flujo en cada línea y las restricciones de corriente y tensión empleadas en el modelo.

Las variables de decisión deben satisfacer las ecuaciones de balance nodal tanto para (10) potencia activa como para (11) potencia reactiva, que aseguran la conservación de energía en cada nudo de la red. Estas se expresan mediante las siguientes ecuaciones:

$$P_{g,i} - P_{d,i} - g_{sh,i}V_i^2 - \sum_{j:(i,j) \in \mathcal{E}} (fP_{ij,c} + r_{ij,c}I_{ij,c}^2) + \sum_{j:(j,i) \in \mathcal{E}} fP_{ji,c} = 0 \quad (10)$$

$$Q_{g,i} - Q_{d,i} + b_{sh,i}V_i^2 - \sum_{j:(i,j) \in \mathcal{E}} (fQ_{ij,c} + x_{ij,c}I_{ij,c}^2 - \frac{b_{ij,c}}{2}V_i^2) + \sum_{j:(j,i) \in \mathcal{E}} (fQ_{ji,c} + \frac{b_{ij,c}}{2}V_i^2) = 0 \quad (11)$$

En primer lugar, se encuentran las pérdidas por el efecto Joule (pérdidas que se deben a la resistencia de las líneas de transporte, y representan calor disipado cuando una corriente eléctrica fluye por un conductor):

$$P_{ij,c}^{loss} = r_{ij,c}I_{ij,c}^2 \quad (12)$$

Le siguen las pérdidas por circulación de potencia reactiva, que si bien la reactiva no disipa energía directamente, su transporte incrementa las pérdidas resistivas:

$$Q_{ij,c}^{loss} = x_{ij,c}I_{ij,c}^2 - b_{sh,ij,c}(V_i^2 - V_j^2) \quad (13)$$

A su vez, las variables deben satisfacer un conjunto de restricciones específicas asociadas a cada línea del sistema eléctrico, que incluyen: (14) caída de tensión, (15) diferencias de ángulo y, (16) corriente.

$$a_{ij}^2V_i^2 - V_j^2 - Z_{ij,c}^2I_{ij,c}^2 - 2(r_{ij,c}fP_{ij,c} + x_{ij,c}fQ_{ij,c}) = 0 \quad (14)$$

$$a_{ij}V_iV_j\sin(\Theta_i - \Theta_j) = x_{ij,c}fP_{ij,c} - r_{ij,c}fQ_{ij,c} \quad (15)$$

$$I_{ij,c}^2 = \frac{fP_{ij,c}^2 + fQ_{ij,c}^2}{V_{ij,c}^2} \quad (16)$$

Como se ha mencionado previamente, el sistema de ecuaciones del flujo de potencia presenta un número excesivo de grados de libertad, lo que lo convierte en un sistema subdeterminado si no se imponen ciertas condiciones adicionales. Para garantizar la viabilidad física del sistema y asegurar su resolubilidad, es necesario fijar determinadas variables en función del tipo de nudo considerado. Esta asignación responde tanto a la estructura matemática del problema como al comportamiento físico esperado en cada tipo de nodo.

- En un nudo PQ (establecidos de “tipo 0” en el código), se fijan las potencias activa y reactiva generadas o demandadas, P_g y Q_g , respectivamente, ya que estos nudos representan típicamente demandas puras o generadores sin control de tensión. Las incógnitas por resolver son el módulo de la tensión ($|V_i|$) y el ángulo de fase (θ_i).
- En un nudo PV (conocidos como “tipo 2”), se imponen P_g y $|V_i|$, al tratarse de generadores que regulan la magnitud de la tensión, manteniéndola constante. En este caso, las incógnitas son la potencia reactiva (Q_g) y el ángulo (θ_i).
- En un nudo Slack o de referencia (obtienen el título de “tipo 3”), se fijan directamente el módulo de la tensión ($|V_i|$) y el ángulo de referencia (θ_i), ya que este nudo equilibra el sistema actuando como punto de referencia para las demás. Las variables calculadas son P_g y Q_g , las cuales garantizan el balance global de potencia en la red.

3.1.2 Script del modelo

Estas ecuaciones se implementan mediante un script en Python (ver *Anexo II: Códigos Python en Script del flujo de potencia en corriente alterna, ACPF*). El programa se organiza en los siguientes bloques:

- Imports y configuración: El script comienza importando los módulos esenciales para la modelización y resolución del ACPF. Se emplea `pyomo.environ` para acceder a las clases de modelado (`ConcreteModel`, `Var`, `ConstraintList`, `Objective`) y al entorno de resolución (`SolverFactory`). Para manejar funciones no lineales dentro de Pyomo, se renombran `sin` y `sqrt` como `pyosin` y `pyosqrt`, respectivamente. Asimismo, se importa `pi` desde el módulo `math` para convertir ángulos de grados a radianes, y finalmente se carga el caso de prueba que aporta todos los datos de la red eléctrica a analizar.
- Obtención y estructuración de datos: A partir de la llamada `case()`, se extraen las estructuras originales: potencia base (`baseMVA`), lista de nudos (`bus`), lista de generadores (`gen`) y lista de línea (`branch`). Con esa información se calculan el número de nudos (`nbus`) y de líneas (`nbranch`) y se asigna `Sbase = baseMVA` como factor de escalado. A continuación, los datos se reorganizan en dos diccionarios: `Bus`, que almacena demandas activa y reactiva, admitancias shunt, condiciones iniciales de tensión y generación; y `Branch`, que contiene parámetros de línea (resistencia, reactancia, susceptancia, tap-ratio, desfase y límites térmicos). Se construyen además dos listas de adyacencia (`in_lines` y `out_lines`) que recogen, para cada nudo, los índices de las líneas que entran y salen, facilitando la formulación de los balances nodales.
- Declaración del modelo Pyomo: Con los datos preparados, se instancia un `ConcreteModel` de Pyomo que representará el problema de optimización. Se definen dos conjuntos de índices: `bus_idx = range(nbus)` para recorrer los nudos y `branch_idx = range(nbranch)` para las líneas. Estos rangos se usarán para declarar variables y restricciones de manera estructurada.
- Inicialización: Se introducen las variables de decisión del modelo: el cuadrado de la tensión en cada nudo (`Vsqr[i]`), los ángulos de fase (`th[i]`), las potencias activa y reactiva generadas (`Pg[i]`, `Qg[i]`), los flujos de potencia por línea (`P[i]`, `Q[i]`) y el cuadrado de la corriente en cada línea (`Isqr[i]`). Para asegurar una convergencia numérica estable, cada variable se inicializa con los valores base extraídos del diccionario `Bus`: `Vsqr` con V_0^2 , `th` con el ángulo inicial en radianes, y `Pg`, `Qg` con las inyecciones de potencia iniciales.
- Función objetivo: En este caso como no busca minimizar los costes, se define un único objetivo de minimización: la potencia activa inyectada en el nudo de referencia (Slack). Esto se logra sumando `model.Pg[i]` únicamente para aquellos índices `i` cuyo tipo de nudo (`Bus["bustype"][i]`) sea 3. De este modo, el solver buscará el despacho que equilibre la red gastando la mínima potencia activa en el Slack. Este es el método más cercano para encontrar un flujo de potencia equilibrado.
- Formulación de ecuaciones: El modelado incluye tres bloques principales de ecuaciones: Balance activo y reactivo, el segundo es análogo al activo pero con términos de susceptancia

shunt y pérdidas inductivas. A continuación, se encuentran las ecuaciones de flujo de potencia, que comprenden la relación de tensión y pérdidas, la ecuación de ángulo con desfase por tap-ratio y la definición de corriente en función de los flujos P y Q. Cada ecuación se añade dinámicamente a través de ConstraintList.

- Fijación de variables según el tipo de nudo: Para resolver el sistema subdeterminado, se incorporan restricciones de fijación según el tipo de nudo como se ha especificado con anterioridad.
- Llamada y operación del solver: Una vez definido el modelo, se graba su estructura a un fichero para inspección. A continuación, se invoca IPOPT mediante SolverFactory, especificando la ruta al ejecutable. La resolución del modelo se lleva a cabo sin mostrar explícitamente el proceso de cálculo en pantalla, es decir, se ejecuta en modo silencioso estableciendo la opción tee=False. De este modo, únicamente se recupera en la variable result la información relativa al estado final del proceso de optimización, facilitando el análisis posterior de la calidad y validez de la solución obtenida. Sin embargo, si se desea observar el proceso detallado de resolución en tiempo real, basta con cambiar dicha opción a tee=True.
- Presentación de resultados: Tras la ejecución, el script imprime el estado del solver y el valor de la función objetivo. A continuación, genera tablas formateadas en consola: primero los datos originales de nudos y líneas, luego los resultados optimizados, incluyendo tensiones, ángulos, potencias generadas, flujos, pérdidas y factores de carga. Este despliegue completo facilita la validación del ACPF.

3.2 Flujo de potencia óptimo en corriente alterna, ACOPF

Una vez verificada la viabilidad del sistema de ecuaciones del modelo ACPF, se procedió a su validación mediante la comparación de los resultados obtenidos con aquellos generados a través del modelo de resolución de flujos de potencia de PyPower, utilizado como referencia por uno de los tutores académicos de este trabajo. Esta comprobación permitió garantizar que las ecuaciones implementadas reprodujeran adecuadamente un estado físico factible del sistema eléctrico. El proceso de validación completo, incluyendo tablas comparativas y análisis de desviaciones, se detalla en el siguiente capítulo: *Resultados y Validación*.

Esta validación inicial resulta esencial, ya que constituye el punto de partida para extender el modelo ACPF hacia un modelo de Flujo Óptimo de Potencia en Corriente Alterna (AC-OPF), donde, además de garantizar la viabilidad física, se introduce una función objetivo de optimización económica y se incorporan restricciones adicionales sobre variables operativas.

3.2.1 Diferencias entre ACPF y el ACOPF

Para desarrollar el AC-OPF, se parte de la formulación validada del ACPF y se realizan los siguientes ajustes clave:

- Se eliminan las fijaciones de variables por tipo de nudo, manteniéndose únicamente en el nudo slack, donde se conserva la fijación del ángulo de tensión como referencia.
- Se redefine la función objetivo, centrada ahora en la minimización del coste de generación activa. Para facilitar la resolución numérica, se omite el término cuadrático $P_{g,i}^2$, adoptando una forma lineal.
- Se añaden restricciones sobre el módulo de la tensión nodal, imponiendo límites típicos del rango operativo: $V_i \in [0.95, 1.05] \quad \forall i \in N$.
- Se imponen límites a la potencia activa y reactiva generada, a través de las cotas: $P_{max}, P_{min}, Q_{max}, Q_{min}$, directamente extraídas de los datos del sistema.
- Se incorporan restricciones sobre el ángulo de tensión de la forma: $\theta_i \in [\theta_{min}, \theta_{max}] = [-60^\circ, 60^\circ] \quad \forall i \in N$.
- De la misma manera se añaden restricciones térmicas de cada línea: $fP_{ij,c}^2 + fQ_{ij,c}^2 \leq S_{maxij,c}^2$

- Finalmente, se incorpora el concepto de Energía No Suministrada (ENS), tanto activa (P_{ENS}) como reactiva (Q_{ENS}), como indicador para detectar posibles errores o incumplimientos en la solución obtenida. Esta variable, expresada en porcentaje, acompaña a las demandas de cada nudo y se incluye en la función objetivo con una penalización elevada (del orden de 10^5), asegurando así que el modelo minimice drásticamente la energía no suministrada y garantice la atención completa de todas las demandas del sistema.

En el modelo ACPF (Análisis de Flujo de Potencia en Corriente Alterna), el objetivo principal es verificar la viabilidad física de un estado operativo propuesto. Esto implica partir de un conjunto conocido de valores —como generación, demanda, tensiones y topología de red— y comprobar si dicho estado cumple las ecuaciones de equilibrio de potencia, derivadas de las leyes de Kirchhoff. Este enfoque no busca optimizar ningún criterio económico ni operativo, sino simplemente verificar que un punto sea físicamente realizable en el sistema eléctrico.

Dado que el ACPF no implica optimización, no se exploran múltiples combinaciones de generación ni de flujos de energía; únicamente se evalúa si un punto dado es consistente con el modelo físico del sistema. Por esta razón, no es necesario imponer restricciones en las líneas de transporte, ya que no se está realizando un despacho alternativo que pudiera, eventualmente, sobrepasar la capacidad de las líneas.

En contraste, el modelo ACOPF (Flujo Óptimo de Potencia en Corriente Alterna) tiene como objetivo determinar el estado operativo más eficiente económicamente, usualmente minimizando el coste de generación activa. Este modelo sí requiere optimización y, por tanto, el algoritmo explora distintos valores posibles de generación y flujo para encontrar la configuración óptima, siempre cumpliendo con las restricciones físicas del sistema.

En este contexto, si no se impusiera restricciones, el optimizador podría proponer soluciones en las que los flujos superen la capacidad térmica admisible de las líneas de transporte, lo cual resultaría físicamente inviable y pondría en riesgo la seguridad del sistema eléctrico. Esta restricción asegura que la solución del ACOPF sea no solo óptima en términos de coste, sino también implementable y segura desde el punto de vista operativo.

3.2.2 Función objetivo

La función objetivo empleada en el modelo AC-OPF es de carácter cuadrático y busca minimizar los costes totales asociados a la generación activa de potencia. Se expresa como:

$$\min \sum_{i \in N} (a_i P_{g,i}^2 + b_i P_{g,i} + c_i)$$

Donde los subíndices de los sumatorios tienen un significado operativo clave. El conjunto N representa el conjunto de nudos con generación del sistema eléctrico. Así, la doble suma recorre todos los generadores del sistema, agrupados por su ubicación nodal, acumulando el coste de generación asociado a cada uno. Esta estructura refleja la realidad física de la red eléctrica, en la que múltiples generadores pueden estar instalados en una mismo nudo, como se observa también en los datos utilizados en el script ACOPF.py (ver *Anexo III: Códigos Python en Script del flujo de potencia óptimo en corriente alterna, ACOPF*).

En dicho script los coeficientes a_i , b_i y c_i definen la curva de costes de cada generación en el punto i , siendo a_i el coeficiente cuadrático que modela el coste marginal creciente, b_i el coeficiente lineal que representa el coste proporcional, y c_i el término constante que puede incluir costes fijos o administrativos asociados a la operación. Esta forma funcional responde a la realidad

técnica de los generadores térmicos, cuyos costes de combustible tienden a incrementarse de manera no lineal con la potencia generada.

Como se argumenta en (ALVA19), el uso de funciones cuadráticas es común debido a su capacidad para representar de forma realista el comportamiento del coste incremental y, al mismo tiempo, mantener una estructura computacionalmente manejable dentro de problemas de optimización no convexa. Además, estos polinomios son compatibles con solvers como IPOPT, que resuelven problemas de tipo NLP (programación no lineal) con funciones objetivo suaves y derivables.

Cabe señalar que este tipo de formulación es un estándar en la literatura académica y ha sido adoptado tanto en estudios de mercado como en herramientas prácticas de operación de redes (ONEI12), dado que proporciona una aproximación precisa de las curvas reales de generación en la mayoría de las tecnologías térmicas convencionales. En el caso concreto de este estudio, se contempla inicialmente la inclusión de costes asociados a la generación de potencia reactiva. No obstante, dichos costes resultan ser muy reducidos o cercanos a cero en la mayoría de los escenarios analizados. Por esta razón, es común que se omitan en muchas formulaciones prácticas, incluida la presente, sin que ello comprometa la validez económica ni operativa del modelo.

3.2.3 Ecuaciones del modelo

El modelo AC-OPF comparte las mismas variables de decisión y parámetros físicos fundamentales que el modelo ACPF, ya que ambos describen el comportamiento eléctrico del sistema en régimen estacionario. No obstante, el AC-OPF introduce además una función objetivo de optimización económica, centrada en minimizar los costes de generación. En ella se encuentra también la nueva variable de Energía No Suministrada (ENS).

Este modelo se compone, por tanto, de dos bloques principales: (i) la función objetivo que representa el coste total del despacho, y (ii) un conjunto de restricciones, que incluye tanto las ecuaciones de flujo de potencia como las limitaciones operativas del sistema (tensión, generación, ángulos y capacidad de línea).

En primer lugar, la función objetivo (17) busca minimizar el coste asociado a la generación de potencia activa. Esta incluye tanto los costes variables como los términos cuadráticos asociados al rendimiento del generador. Aunque algunos modelos incorporan también los costes de arranque y parada (startup/shutdown), en esta formulación dichos efectos se consideran despreciables o absorbidos en el término constante c_i . La función objetivo permite penalizar directamente cualquier incumplimiento en la satisfacción de la demanda.

$$\min(\sum_{i \in N}(a_i P_{g,i}^2 + b_i P_{g,i} + c_i) + 10^5(Q_{ENS} + P_{ENS})) \quad (17)$$

Las variables de decisión deben satisfacer las ecuaciones de balance nodal de potencia. La ecuación (18) garantiza el equilibrio de potencia activa en cada nudo, mientras que la ecuación (19) hace lo propio para la potencia reactiva:

$$P_{g,i} - (1 - P_{ENS,i})P_{d,i} - g_{sh,i}V_i^2 - \sum_{j:(i,j) \in \epsilon} (fP_{ij,c} + r_{ij,c}I_{ij,c}^2) + \sum_{j:(j,i) \in \epsilon} fP_{ji,c} = 0 \quad (18)$$

$$Q_{g,i} - (1 - Q_{ENS,i})Q_{d,i} + b_{sh,i}V_i^2 - \sum_{j:(i,j) \in \epsilon} (fQ_{ij,c} + x_{ij,c}I_{ij,c}^2 - \frac{b_{ij,c}}{2}V_i^2) + \sum_{j:(j,i) \in \epsilon} (fQ_{ji,c} + \frac{b_{ij,c}}{2}V_i^2) = 0 \quad (19)$$

Adicionalmente, para cada línea del sistema deben cumplirse las ecuaciones que describen el comportamiento del flujo de potencia. La ecuación (20) impone la caída de tensión entre extremos de la línea como función de las pérdidas y de los flujos circulantes, mientras que la ecuación (21) relaciona la diferencia angular entre nudos con los flujos activo y reactivo a través de las

impedancias de línea. La ecuación (22) establece la relación entre corriente e intensidad de flujo en base al módulo de la tensión. En este caso de flujo óptimo, es necesario añadir la ecuación (23) que impone la restricción térmica de cada línea, limitando la potencia aparente transmitida.

$$a_{ij}^2 V_i^2 - V_j^2 - Z_{ij,c}^2 I_{ij,c}^2 - 2(r_{ij,c} f P_{ij,c} + x_{ij,c} f Q_{ij,c}) = 0 \quad (20)$$

$$a_{ij} V_i V_j \sin(\theta_i - \theta_j) = x_{ij,c} f P_{ij,c} - r_{ij,c} f Q_{ij,c} \quad (21)$$

$$I_{ij,c}^2 = \frac{f P_{ij,c}^2 + f Q_{ij,c}^2}{V_{ij,c}^2} \quad (22)$$

$$f P_{ij,c}^2 + f Q_{ij,c}^2 \leq S_{maxij,c}^2 \quad (23)$$

Por último, y a diferencia del ACPF, se han de respetar los límites físicos y las normativas de seguridad de los elementos de la red. Las ecuaciones (24) y (25) establecen los rangos admisibles para el módulo de la tensión y el ángulo de fase, respectivamente. Por último, las ecuaciones (26) y (27) definen los intervalos válidos para la potencia activa y reactiva generada en cada unidad.

$$V^{min} \leq V \leq V^{max} \quad (24)$$

$$\theta^{min} \leq \theta \leq \theta^{max} \quad (25)$$

$$P_{g,i}^{min} \leq P_{g,i} \leq P_{g,i}^{max} \quad (26)$$

$$Q_{g,i}^{min} \leq Q_{g,i} \leq Q_{g,i}^{max} \quad (27)$$

3.2.4 Script del modelo

La implementación de estas ecuaciones se ha realizado mediante un script en Python, cuya estructura completa se presenta en el Anexo II (véase *Códigos Python: Script del flujo de potencia óptimos en corriente alterna, ACOPF*). El programa se organiza en los siguientes bloques funcionales:

- Imports y configuración: Siguiendo el mismo procedimiento que el código del ACPF, importa los módulos esenciales de construcción y resolución de ACOPF.
- Obtención y estructuración de datos: Sigue el mismo proceso que el código elaborado para el flujo de potencia.
- Declaración del modelo Pyomo: Se crea una instancia del modelo con ConcreteModel() y se definen los índices bus_idx y branch_idx para recorrer nudos y líneas. Luego se declaran las variables de decisión: el cuadrado de la tensión por nodo (Vsqr[i]), los ángulos de tensión (th[i]), las potencias generadas (Pg[i], Qg[i]), los flujos de potencia por línea (P[k], Q[k]), y el cuadrado de la corriente (Isqr[k]).
- Función objetivo: La función objetivo del modelo, previamente explicada en el apartado anterior, extrae los coeficientes del campo gencost de cada generador.
- Formulación de ecuaciones: El modelo incluye varios bloques de restricciones. En primer lugar, los balances de potencia activa y reactiva en cada nudo. Además, se formulan las ecuaciones de flujo por línea: caída de tensión entre extremos (según la impedancia), ecuación de ángulo ajustada por el tap-ratio, y definición de corriente en función de los flujos de carga. Todas estas restricciones se añaden como ConstraintList.
- Restricciones operativas: Se imponen diversas restricciones que garantizan la validez física de la solución. Estas se refieren a los límites físicos especificados en las ecuaciones (23 a 27).

- Condición de slack: El ángulo del nudo slack se fija en su valor inicial, eliminando la ambigüedad de fase global.
- Llamada y operación del solver: Una vez completada la definición del modelo, se especifica la ruta del ejecutable del solver IPOPT y se lanza la optimización mediante SolverFactory. El resultado se almacena en la variable result, desde donde se recupera el estado del solver y el valor mínimo de la función objetivo alcanzado.
- Presentación de resultados: Finalmente, el script imprime el estado de la optimización y el valor de la función objetivo. A continuación, muestra en consola una tabla detallada con los resultados optimizados: tensiones, ángulos, generación activa y reactiva por nodo, flujos de potencia por línea, pérdidas por efecto Joule, y la carga relativa de cada línea expresada como porcentaje del límite térmico. Al igual que en el caso del ACPF, esta visualización final permite verificar tanto la calidad como la factibilidad de la solución obtenida.

3.3 Flujo óptimo con relajación SOCP

Una vez analizado el modelo clásico de flujo óptimo, es evidente que, si bien representa fielmente el comportamiento físico del sistema, también plantea importantes desafíos de cálculo en sistemas grandes, debido a su naturaleza no convexa. En este contexto, surge el interés por explorar enfoques alternativos que permitan una mayor flexibilidad y eficiencia. En particular, como se menciona con anterioridad, se propone ahora el desarrollo de un modelo artificial basado en programación cónica de segundo orden (SOCP), cuyo objetivo principal es evaluar si, al desprenderse parcialmente de ciertas restricciones físicas del sistema original —es decir, al relajarlas—, se puede alcanzar una mejora en la optimalidad de las soluciones.

Este nuevo enfoque busca aprovechar la estructura convexa del SOCP para construir un modelo más flexible, que pueda actuar como una aproximación eficiente o incluso como una herramienta complementaria para resolver el problema de flujo óptimo. Sin embargo, al tratarse de un modelo artificial, la principal preocupación radica en determinar si las soluciones obtenidas son factibles desde el punto de vista del sistema físico real. A continuación, se presenta el desarrollo de este modelo artificial basado en SOCP, así como un análisis de sus ventajas, limitaciones y comportamiento frente al ACPF tradicional.

3.3.1 Diferencias entre ACPF y ACPF+SOCP

El modelo ACPF tradicional se caracteriza por su fidelidad al comportamiento físico del sistema eléctrico, lo cual se refleja en la inclusión de restricciones de igualdad no lineales altamente acopladas, como las ecuaciones de flujo de potencia en corriente alterna. Este nivel de realismo, no obstante, introduce una estructura no convexa que puede llevar a que los algoritmos de resolución —como el método de punto interior utilizado en este proyecto— converjan hacia óptimos locales en lugar del óptimo global.

Por el contrario, el modelo basado en programación cónica de segundo orden (SOCP) emplea una relajación convexa del problema original, lo que modifica el espacio de búsqueda y permite garantizar la obtención del óptimo global dentro del dominio definido por la relajación. Esto se logra, en parte, transformando algunas restricciones del problema: en particular, ciertas ecuaciones de igualdad del ACPF se convierten en desigualdades dentro del modelo SOCP. Esta modificación no solo reduce la complejidad del problema, sino que también amplía el espacio factible, aunque a costa de una representación menos estricta del sistema físico. Como se ha expuesto, dichas restricciones relajadas se muestran en la siguiente sección de esta memoria.

Es importante destacar que, dado el carácter artificial de esta formulación, la solución obtenida a través del SOCP no puede considerarse válida directamente. Es necesario someterla a un análisis posterior para comprobar si satisface las condiciones físicas y operativas del sistema real.

Así, el enfoque SOCP ofrece una promesa de mayor optimalidad y eficiencia computacional, pero su utilidad práctica dependerá de la calidad de dicha solución en términos de factibilidad para el ACOPF original.

En la próxima figura se ilustra la diferencia, previamente mencionada, entre regiones factibles del problema ACOPF y de su relajación SOCP. Se observa que la región factible del ACOPF está contenida dentro de la del SOCP, lo que explica por qué este último puede encontrar soluciones globales, mientras el ACOPF puede quedar atrapado en las locales.

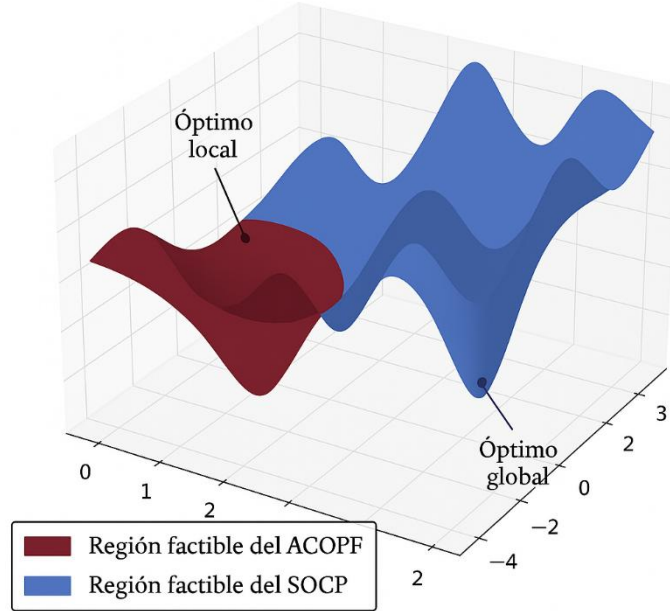


Figura 3: Ilustración de la comparación de regiones factibles ACOPF y SOCP
(ejemplo desarrollado por la autora)

3.3.2 Ecuaciones del modelo

Al partir directamente del modelo ACOPF, sus ecuaciones coinciden en gran medida con las de este.

En primer lugar, se incluye la función objetivo (28). En este caso, no se considera la energía no suministrada, ya que no se permite dejar de satisfacer las demandas. Además, su uso como detector de errores no se considera necesario, puesto que el correcto funcionamiento del sistema se verifica posteriormente mediante las restricciones impuestas por las envolventes.

$$\min(\sum_{i \in N}(a_i P_{g,i}^2 + b_i P_{g,i} + c_i)) \quad (28)$$

A continuación, se mantienen las ecuaciones de balance de potencia activas (29) y reactivas (30):

$$P_{g,i} - P_{d,i} - g_{sh,i} V_i^2 - \sum_{j:(i,j) \in \epsilon} (f P_{ij,c} + r_{ij,c} I_{ij,c}^2) + \sum_{j:(j,i) \in \epsilon} f P_{ji,c} = 0 \quad (29)$$

$$Q_{g,i} - Q_{d,i} + b_{sh,i} V_i^2 - \sum_{j:(i,j) \in \epsilon} (f Q_{ij,c} + x_{ij,c} I_{ij,c}^2 - \frac{b_{ij,c}}{2} V_i^2) + \sum_{j:(j,i) \in \epsilon} (f Q_{ji,c} + \frac{b_{ij,c}}{2} V_i^2) = 0 \quad (30)$$

En cuanto a las restricciones por línea, se introducen ligeras modificaciones. La caída de tensión (31) y el límite térmico (32) se conservan. Sin embargo, la restricción sobre la intensidad se relaja, lo que da lugar a una de las principales ecuaciones (33) de la relajación propuesta.

$$a_{ij}^2 V_i^2 - V_j^2 - Z_{ij,c}^2 I_{sqrt} - 2(r_{ij,c} f P_{ij,c} + x_{ij,c} f Q_{ij,c}) = 0 \quad (31)$$

$$\sqrt{f P_{ij,c}^2 + f Q_{ij,c}^2} \leq S_{maxij,c} \quad (32)$$

$$f P_{ij,c}^2 + f Q_{ij,c}^2 \leq V_{ij,c} I_{sqrt} ; V_{ij,c} = V_i V_j ; I_{sqrt} = I_{ij,c}^2 \quad (33)$$

Dado que en esta formulación se incorporan restricciones cíclicas —derivadas de la Ley de Voltajes de Kirchhoff (KVL)—, resulta necesario prestar especial atención a las diferencias de ángulo entre nudos. Para ello, se introduce una nueva variable que representa dichas diferencias, las cuales son fundamentales en las ecuaciones de tensión que rigen el comportamiento de la red.

La correcta aplicación de estas restricciones requiere identificar los ciclos presentes en la topología de la red. A partir de estos ciclos, se definen las variables de ángulo asociadas a cada una de las líneas involucradas, considerando ambos sentidos de circulación. El proceso de detección de ciclos, así como la asignación y lectura de estas variables, se detalla en la sección 3.3.4 Cálculo de ciclos. Una vez identificados los ciclos, debe cumplirse la ley de tensiones de Kirchhoff (34), ley que afecta a la variable θ_{ij} (diferencia angular), una de las más relevantes en la relajación.

$$\sum_{(i,j) \in C} \theta_{ij} = \sum_{(i,j) \in C} \theta_i - \theta_j = 0 \text{ (o múltiplo de } 2\pi) \quad (34)$$

Las diferencias angulares deben, asimismo, sujetarse a las restricciones de las envolventes (35), una condición adicional, necesaria para garantizar que no superen el límite que anteriormente se imponía mediante la ecuación (21). La efectividad de las envolventes depende, en parte, de la topología de la red analizada. En configuraciones con un mayor número de ciclos, se generan más envolventes y, por tanto, un mayor número de restricciones, lo que puede favorecer la convexidad y la convergencia del modelo. Dicha envolvente se calcula haciendo uso del parámetro θ_m , y la variable $Mvar_{ij,c}$, dichas deducciones se pueden encontrar en (CHOW24):

$$Mvar_{ij,c} = \frac{B_{ij} f P_{ij,c} - G_{ij} f Q_{ij,c}}{B_{ij}^2 + G_{ij}^2} ; \text{ siendo } B_{ij} \text{ y } G_{ij} \text{ elementos de la matriz de impedancias: } Y.$$

$$\begin{cases} \theta_{ij} \geq \frac{Mvar_{ij,c}}{V_{ij,c}^{max} \cos(\theta_m/2)} - \tan(\theta_m/2) + \theta_m/2 \\ \theta_{ij} \leq \frac{Mvar_{ij,c}}{V_{ij,c}^{min} \cos(\theta_m/2)} + \tan(\theta_m/2) - \theta_m/2 \end{cases} \quad (35)$$

$$V_{ij,c} = V_i V_j$$

Finalmente, se mantienen los límites físicos de cada máquina, como en el modelo original:

$$V^{min} \leq V \leq V^{max} \quad (36)$$

$$\theta^{min} \leq \theta \leq \theta^{max} \quad (37)$$

$$P_{g,i}^{min} \leq P_{g,i} \leq P_{g,i}^{max} \quad (38)$$

$$Q_{g,i}^{min} \leq Q_{g,i} \leq Q_{g,i}^{max} \quad (39)$$

3.3.3 Script del modelo

La unión de la relajación con el ACOPF se ha realizado mediante un script en Python, cuya estructura completa se presenta en el Anexo II (véase *Códigos Python: Script del flujo óptimo con relajación SOCP*). El programa se organiza en los siguientes bloques funcionales:

- Imports y configuración: El programa arranca exactamente igual que los códigos anteriores, importando de `pyomo.environ` las funciones necesarias para definir el modelo de optimización (variables, restricciones, función objetivo y solver). Como se ha explicado anteriormente, se ha de incorporar el módulo `math` para poder hacer operaciones trigonométricas y constantes, y usa `networkx` para poder detectar los ciclos en la red. Finalmente, se carga el caso de prueba, que aporta los datos de nudos, generadores y costes.
- Detección y procesado de ciclos: Antes de construir el modelo, se define una función que extrae del diccionario MATPOWER la lista de aristas y construye un grafo, de donde se obtiene la base de ciclos simples, dicha función se puede consultar en el script *calculodeciclos*, explicado en la sección 3.4.4 *Cálculo de ciclos*. Luego, otra función toma esos ciclos y genera una representación orientada de cada uno, identificando las variables ángulo asociadas y la forma simbólica de la Ley de Voltajes de Kirchhoff (KVL) para cada lazo. Tras haber procesado los ciclos simples, se construye el conjunto completo de variables de diferencia de ángulo para todas las posibles direcciones de cada línea, incluyendo aquellas surgidas de los ciclos. Estas variables quedan acotadas a $\pm 60^\circ$ para posteriormente imponer la “envolvente” angular.
- Carga y estructuración de datos: Se invoca el caso para extraer las matrices de datos (bus, branch, gen, gencost) y la base de potencia. A partir de ahí se calcula el número de nudos y líneas, y se normalizan potencias y corrientes respecto a la base. Con ello, se construyen dos diccionarios:
 - Bus: tipo de nudo, demanda activa y reactiva, admitancias de shunt y condiciones iniciales de tensión y ángulo.
 - Branch: índices de origen y destino, parámetros eléctricos (resistencia, reactancia, admitancia de shunt, ratio de transformación y desfase), límites térmicos e impedancias al cuadrado.
- Límites y costes de generación: A partir de la matriz de generadores se obtienen los mínimos y máximos de generación activa y reactiva en cada nudo, y de la información de costes (gencost) se extraen los coeficientes del coste cuadrático, lineal y constante, que luego servirán para formar la función objetivo. A su vez se crean dos listas de adyacencia: líneas entrantes y salientes por cada nudo, lo que facilita la construcción de los balances de potencia activa y reactiva al recorrer la red.
- Declaración del modelo Pyomo: Se instancia un modelo concreto y se definen índices para nudos y ramos. Se declaran las variables de estado (tensión al cuadrado, ángulo de tensión, generación activa/reactiva, flujos en cada línea y corriente al cuadrado) y las variables auxiliares (parte imaginaria del flujo, producto de tensiones y diferencias de ángulo). Todas ellas se inicializan con valores planos (tensión unidad, ángulo cero, flujos y auxiliares a cero o uno) para facilitar la convergencia del solver.
- Función objetivo: El modelo minimiza el coste agregado de generación, formado por la suma de los términos cuadráticos, lineales y constantes de cada generador. No se incluye ninguna penalización de carga no servida, pues aquí no hay variables ENS.
- Balance de potencia, restricciones por línea y relajación SOCP: Para cada nudo se añade una restricción que iguala generación, demanda y flujos entrantes/salientes, incorporando también la contribución de los shunt. Se hace por separado para la potencia activa y la reactiva, garantizando que toda demanda se satisfaga sin relajaciones. En cada línea se modeliza primero la caída de tensión tradicional, luego se incorpora la relajación SOCP que reemplaza la igualdad del flujo aparente por una desigualdad cónica (asegurando convexidad), y por último la restricción térmica. Evidentemente, también se han de mantener los límites de seguridad de cada uno de ellos elementos.

- Condición de slack: El ángulo del nudo slack se fija en su valor inicial, eliminando la ambigüedad de fase global.
- Ciclos KVL y envolvente angular: Se vinculan las variables de diferencia de ángulo con las variables de ángulo de cada nudo, luego se imponen las ecuaciones de Kirchhoff en cada ciclo orientado, y finalmente se añaden las cotas convexas (“envolventes”) que limitan las diferencias de ángulo a $\pm 60^\circ$. Esto asegura la coherencia de fases en toda la red dentro de un marco convexificado.
- Resolución y presentación de resultados: Con IPOPT como solver (ruta configurada en el script), se lanza la optimización del problema relajado. Una vez finalizada, el script imprime tablas detalladas de nudos, líneas y generadores, además de verificar las sumas de ángulo en los ciclos y las condiciones de las envolventes, facilitando la validación de la factibilidad.

3.3.4 Cálculo de ciclos

Al eliminar los términos trigonométricos $\cos(\theta_{ij})$ y $\sin(\theta_{ij})$ de la formulación principal y sustituirlos por magnitudes cuadradas más la inecuación cónica, se relajan todas las ecuaciones de ángulo de los nudos. Esta relajación convexo-cónica sacrifica exactitud en las mallas de la red a cambio de una formulación totalmente convexa y de gran escalabilidad computacional.

Precisamente por ello, resulta de vital importancia incluir restricciones cíclicas mediante un «envolvente convexa» sobre las diferencias de ángulo. La inclusión de estas restricciones permite que la relajación SOCP evolucione de una mera aproximación a una solución exacta, es decir, cierra la brecha existente entre la solución relajada y la solución real del OPF, manteniendo al mismo tiempo la reconocida eficiencia computacional de los modelos de segundo orden cónico, (CHOW24).

Para introducir correctamente los ciclos en la relajación, se ha seguido la siguiente estrategia en el script *calculodeciclos*, elaborado en Python (véase *Códigos Python: Script del cálculo de ciclos*), posteriormente se ha integrado en la relajación SOCP:

- En primer lugar, a partir de la topología del sistema, se determina el número total de ciclos existentes mediante la relación: $K_e = x_{líneas} - n_{nudos} + 1$. Una vez conocido el número de ciclos, se identifica cada uno de ellos (C) y los nodos (a_n) que lo definen siguiendo un sentido horario. Por cada ciclo, entonces se aplica la Ley de Tensiones de Kirchhoff (KVL):

$$C = \{a_1, a_2, \dots, a_n, a_{n+1}\}$$

$$\sum_{(i,j) \in C} \theta_{a_{i+1}} - \theta_{a_i} = 0$$

- Con el propósito de evitar redundancias y permitir su utilización efectiva en el script de relajación, se definen variables auxiliares correspondientes a las diferencias angulares entre cada par de nodos:

$$var_{ij} = \theta_i - \theta_j$$

Si al recorrer el ciclo se transita en el sentido $a \rightarrow b$ la variable se definirá como $var_{ba} = \theta_b - \theta_a$. En el caso de leer dicha al revés ($b \rightarrow a$), al recorrer el ciclo adyacente en sentido horario, se tiene $var_{ab} = -var_{ba}$.

- Por ello, finalmente se implementa la siguiente suma:

$$\sum_{(i,j) \in C} q \cdot var_{ij} = 0$$

$$q = \begin{cases} 1 & \text{si el recorrido es } i \text{ a } j \\ -1 & \text{si el recorrido es } j \text{ a } i \end{cases}$$

El modelo propuesto se ha aplicado a todos los casos base con el fin de verificar el correcto cumplimiento de las restricciones cíclicas. A continuación, se presenta detalladamente el ejemplo correspondiente al caso IEEE de 14 nudos. Aunque los otros dos casos no se exponen explícitamente aquí, siguen la misma metodología y presentan resultados igualmente consistentes, los cuales pueden consultarse en el *Anexo IV: Resultados en Resultados de Cálculo de Ciclos*.

IEEE 14 nudos: 7 ciclos detectados

Ciclo 1: [6, 12, 13, 6]

Ciclo 2: [9, 10, 11, 6, 13, 14, 9]

Ciclo 3: [2, 4, 5, 2]

Ciclo 4: [9, 4, 5, 6, 13, 14, 9]

Ciclo 5: [9, 7, 4, 9]

Ciclo 6: [2, 3, 4, 2]

Ciclo 7: [1, 2, 5, 1]

Variables theta definidas (en grados, y su equivalente en radianes) para IEEE 14 nudos:

theta_10_9: -0.1600° (-0.0028 rad)
theta_11_10: 0.3100° (0.0054 rad)
theta_11_6: -0.5700° (-0.0099 rad)
theta_12_6: -0.8500° (-0.0148 rad)
theta_13_12: -0.0900° (-0.0016 rad)
theta_13_6: -0.9400° (-0.0164 rad)
theta_14_13: -0.8800° (-0.0154 rad)
theta_14_9: -1.1000° (-0.0192 rad)
theta_2_1: -4.9800° (-0.0869 rad)
theta_3_2: -7.7400° (-0.1351 rad)
theta_4_2: -5.3500° (-0.0934 rad)
theta_4_3: 2.3900° (0.0417 rad)
theta_5_1: -8.7800° (-0.1532 rad)
theta_5_2: -3.8000° (-0.0663 rad)
theta_5_4: 1.5500° (0.0271 rad)
theta_6_5: -5.4400° (-0.0949 rad)

theta_7_4: -3.0400° (-0.0531 rad)

theta_9_4: -4.6100° (-0.0805 rad)

theta_9_7: -1.5700° (-0.0274 rad)

Detalles de diferencias angulares para cada ciclo en IEEE 14 nudos:

Ciclo 1:

+theta_12_6 = -0.8500°

+theta_13_12 = -0.0900°

-theta_13_6 = +0.9400°

Suma numérica = 0.0000e+00° → OK

Ecuación simbólica: theta_12_6 + theta_13_12 - theta_13_6 = 0

Ciclo 2:

+theta_10_9 = -0.1600°

+theta_11_10 = +0.3100°

-theta_11_6 = +0.5700°

+theta_13_6 = -0.9400°

+theta_14_13 = -0.8800°

-theta_14_9 = +1.1000°

Suma numérica = 0.0000e+00° → OK

Ecuación simbólica: theta_10_9 + theta_11_10 - theta_11_6 + theta_13_6 + theta_14_13 - theta_14_9 = 0

Ciclo 3:

+theta_4_2 = -5.3500°

+theta_5_4 = +1.5500°

-theta_5_2 = +3.8000°

Suma numérica = 0.0000e+00° → OK

Ecuación simbólica: theta_4_2 + theta_5_4 - theta_5_2 = 0

Ciclo 4:

-theta_9_4 = +4.6100°

+theta_5_4 = +1.5500°

+theta_6_5 = -5.4400°

+theta_13_6 = -0.9400°

+theta_14_13 = -0.8800°

-theta_14_9 = +1.1000°

Suma numérica = $0.0000e+00^\circ$ → OK

Ecuación simbólica: $-\text{theta}_{9_4} + \text{theta}_{5_4} + \text{theta}_{6_5} + \text{theta}_{13_6} + \text{theta}_{14_13} - \text{theta}_{14_9} = 0$

Ciclo 5:

$-\text{theta}_{9_7} = +1.5700^\circ$

$-\text{theta}_{7_4} = +3.0400^\circ$

$+\text{theta}_{9_4} = -4.6100^\circ$

Suma numérica = $0.0000e+00^\circ$ → OK

Ecuación simbólica: $-\text{theta}_{9_7} - \text{theta}_{7_4} + \text{theta}_{9_4} = 0$

Ciclo 6:

$+\text{theta}_{3_2} = -7.7400^\circ$

$+\text{theta}_{4_3} = +2.3900^\circ$

$-\text{theta}_{4_2} = +5.3500^\circ$

Suma numérica = $0.0000e+00^\circ$ → OK

Ecuación simbólica: $\text{theta}_{3_2} + \text{theta}_{4_3} - \text{theta}_{4_2} = 0$

Ciclo 7:

$+\text{theta}_{2_1} = -4.9800^\circ$

$+\text{theta}_{5_2} = -3.8000^\circ$

$-\text{theta}_{5_1} = +8.7800^\circ$

Suma numérica = $0.0000e+00^\circ$ → OK

Ecuación simbólica: $\text{theta}_{2_1} + \text{theta}_{5_2} - \text{theta}_{5_1} = 0$

4. Análisis de los resultados y validación

4.1 Herramientas de resolución

Para poner a prueba los modelos desarrollados en este trabajo, se adoptó una estrategia escalonada de validación. En primer lugar, se implementó un caso sencillo de 3 nudos (IEEE 3 nudos). Este caso inicial permite validar y asegurar tanto la exactitud matemática como la factibilidad técnica del conjunto de ecuaciones planteadas en los modelos de flujo de potencia alterno convencional (ACPF) y óptimo (ACOPF).

Posteriormente, se aplicaron los modelos a una red eléctrica de tamaño intermedio, específicamente el sistema IEEE de 14 nudos. Esta fase permitió evaluar la capacidad de escalado de los modelos, asegurando que tanto el método de resolución como las estructuras computacionales utilizadas fueran eficientes y robustas al incrementarse la complejidad del problema.

Finalmente, los modelos fueron implementados en un caso aún más complejo, correspondiente al sistema IEEE de 30 nudos, multiplicando así por diez la escala inicial del análisis. Esta etapa permitió simular de manera más realista y exhaustiva las capacidades de todos los modelos estudiados, verificando su rendimiento, precisión y tiempo de solución en escenarios de mayor tamaño y complejidad.

Como se ha mencionado con anterioridad en esta memoria, para la resolución numérica de los modelos ACPF y ACOPF desarrollados, se utilizó el solver IPOPT (Interior Point OPTimizer) - versión 3.14.17, un solver de optimización no lineal especialmente apropiado para grandes problemas de programación matemática, tal y como se confirma en la documentación oficial proporcionada por COIN-OR.

IPOPT resuelve problemas mediante métodos del punto interior, una técnica que opera transformando el problema original con restricciones de desigualdad en una secuencia de problemas más fáciles mediante la inclusión de barreras logarítmicas. Estas barreras penalizan las soluciones cercanas a los límites de las restricciones, evitando que el algoritmo salga prematuramente del conjunto de soluciones factibles. El método inicia la búsqueda de la solución en un punto interior del conjunto factible y avanza iterativamente hacia la frontera, disminuyendo gradualmente la influencia de las barreras logarítmicas hasta que converja a una solución óptima. Este procedimiento garantiza soluciones más estables y rápidas en comparación con métodos tradicionales basados en puntos extremos, siendo particularmente efectivo para resolver sistemas de ecuaciones no lineales complejas típicos en análisis de flujos de potencia.

Este solver maneja restricciones de desigualdad y ecuaciones no lineales, lo que lo convierte en una herramienta especialmente valiosa para aplicaciones en sistemas eléctricos de potencia. De la misma manera, ofrece múltiples estrategias para manejar problemas de escalado, ajuste fino de tolerancias numéricas y preconditionamiento, facilitando su uso práctico y robusto (COIN24).

El código computacional que implementa los modelos ha sido desarrollado utilizando el lenguaje de programación Python en conjunto con Pyomo, una biblioteca de modelado matemático que permite la formulación explícita de problemas de optimización de manera algebraica. Todo ello se realiza mediante ecuaciones algebraicas simbólicas, facilitando considerablemente la definición y el análisis de problemas complejos como los flujos de potencia alternos y óptimos, según se confirma en la documentación oficial de Pyomo. Esta proporciona una interfaz amigable que conecta eficientemente con IPOPT, lo que simplifica tanto la preparación de los datos de entrada como la interpretación de resultados (PYOM24).

La integración de IPOPT y Pyomo resulta especialmente poderosa para el modelado y análisis de sistemas eléctricos de potencia, al proporcionar una plataforma flexible, escalable y precisa para enfrentar tanto análisis académicos como problemas prácticos de ingeniería eléctrica en entornos industriales.

A continuación, se procede al análisis detallado de los resultados obtenidos a partir de los modelos desarrollados, aplicados a los distintos casos previamente descritos. Además, se presenta una comparación cuantitativa y cualitativa con resultados de referencia considerados como estándar. Estos valores de referencia han sido verificados con anterioridad por el co-director del proyecto, Erik Francisco Álvarez Quispe, mediante el uso de la herramienta PyPower, ampliamente reconocida en el ámbito académico para simulaciones de sistemas eléctricos de potencia.

4.2 Presentación de los casos

Los casos de estudio seleccionados son de 3, 14 y 30 nudos, elegidos estratégicamente por su tamaño, ya que cada uno permite evaluar distintos aspectos de los modelos. El caso de 3 nudos es útil para validar con precisión la solidez y robustez de las ecuaciones. Posteriormente, el caso de 14 nudos facilita la evaluación de la escalabilidad del modelo. Finalmente, tras verificar ambos aspectos anteriores, el caso de 30 nudos pone a prueba la capacidad y rendimiento del modelo en sistemas considerablemente mayores, específicamente en una red cuyo tamaño es diez veces superior al del caso base. Si bien no todas las literaturas académicas consideran este caso uno de gran tamaño, es grande en consideración con los primeros modelos.

El sistema de 3 nudos es el más simple y constituye un caso básico fundamental. Este sistema presenta una estructura radial sencilla (una sola malla) sin elementos complejos adicionales tales como shunts o compensadores síncronos. Esta simplicidad es ideal para validar con precisión la robustez matemática y numérica de las ecuaciones fundamentales del ACOPF, incluyendo la convergencia y la estabilidad numérica. La ausencia de elementos complejos asegura que cualquier comportamiento observado es atribuible directamente a las ecuaciones base del modelo.

El sistema de 14 nudos presenta un nivel intermedio de complejidad, incluyendo múltiples buses tipo PV y PQ, así como una estructura reticular más elaborada que introduce ciclos y posibles rutas alternativas para el flujo de potencia. Aunque este sistema no incluye explícitamente compensadores síncronos, sí presenta *shunts* reactivos en tres nudos (3, 6 y 8) que afectan al control de la tensión local y al flujo reactivo (no generan activa, solo generan reactiva). Estas características permiten evaluar aspectos como la escalabilidad del modelo y la sensibilidad del ACOPF a las restricciones operativas y límites térmicos de las líneas. La existencia de ciclos también requiere considerar restricciones adicionales para mantener coherencia angular, lo cual influirá directamente en los resultados de la solución óptima.

Finalmente, el sistema de 30 nudos incrementa significativamente la complejidad, ofreciendo una estructura de red considerablemente más extensa con múltiples ciclos, una gran cantidad de nudos PQ y varios nudos PV. Destaca la presencia explícita de transformadores con tomas (*taps*) variables y *shunts* reactivos en diversos nudos, elementos clave para evaluar cómo la presencia de control de tensión adicional y la regulación de reactivos influyen en la solución del ACOPF. La presencia de estos elementos implica mayor sensibilidad de los resultados ante perturbaciones y restricciones operativas, permitiendo analizar con precisión la potencia del modelo en sistemas realistas y complejos. Este escenario también permite observar claramente el efecto acumulado de las restricciones angulares de ciclo y cómo estas afectan la distribución de flujos y tensiones dentro del sistema, crucial para asegurar resultados coherentes y físicamente plausibles.

4.3 Resultados del flujo de potencia en corriente alterna, ACPF

4.3.1 Caso de 3 nudos

Los valores obtenidos mediante el código en Pyomo resuelto con IPOPT coinciden prácticamente de forma exacta con los valores de referencia proporcionados en el archivo PF_IEEE3.txt. El valor del objetivo, correspondiente a la generación activa en el nodo slack, es de 148.0637 MW en ambos casos, con una diferencia numéricamente insignificante (del orden de $1e-9$), atribuible únicamente a efectos de redondeo y precisión computacional. Del mismo modo, los valores totales de generación activa (318.0737 MW), generación reactiva (41.0908 MVar), demanda activa (315.0000 MW) y demanda reactiva (130.0000 MVar) son idénticos en ambos resultados. Además, los flujos de potencia activa por línea (P_{ij}), como los -13.2446 MW en la línea 1, 51.3084 MW en la línea 2 y -45.8533 MW en la línea 3, coinciden exactamente.

A continuación, se presentan los resultados obtenidos mediante los modelos desarrollados, junto con los valores de referencia extraídos de simulaciones previas. Para facilitar su interpretación y análisis, se incluye tanto la diferencia absoluta como el gap porcentual, el cual permite evaluar la precisión relativa de los modelos respecto a los datos de referencia.

La expresión utilizada para calcular el gap porcentual es la siguiente: $gap(\%) = \frac{x_{modelo} - x_{ref}}{x_{ref}} * 100$.

Esta métrica resulta especialmente útil para cuantificar desviaciones incluso cuando los valores absolutos son similares, ofreciendo una perspectiva más clara del rendimiento numérico del modelo.

Tabla 4.1: Resultados globales del flujo de potencia del caso de 3 nudos

Parámetro	ACPF IEEE3	Referencia IEEE3	Gap	Gap %
Función objetivo [MW]	148,0638	148,0638	0,0000	0,0000
Generación activa total [MW]	318,0737	318,0737	0,0000	0,0000
Generación reactiva total [MVar]	41,0908	41,0908	0,0000	0,0000
Demanda activa total [MW]	315,0000	315,0000	0,0000	0,0000
Demanda reactiva total [MVar]	130,0000	130,0000	0,0000	0,0000

Tabla 4.2: Resultados de generación del caso de 3 nudos

Nudo	P_g (ACPF)	P_g (Ref)	Gap	Gap %	Q_g (Modelo)	Q_g (Ref)	Gap	Gap %
1	148,0637	148,0637	0,0000	0,0000	54,7194	54,7194	0,0000	0,0000
2	170,0100	170,0100	0,0000	0,0000	-8,8087	-8,8087	0,0000	0,0000
3	0,0000	0,0000	0,0000	0,0000	-4,8199	-4,8199	0,0000	0,0000

Tabla 4.3: Resultados de flujo de potencia del caso de 3 nudos

Línea	Parámetro	ACPF	Ref.	Gap	Gap %
1 al 2	P_{ij}	-13,2446	-13,2446	0,0000	0,0000
	P_{ji}	13,4858	13,4858	0,0000	0,0000
	Q_{ij}	4,6428	4,6428	0,0000	0,0000
	Q_{ji}	-30,4861	-30,4861	0,0000	0,0000
	P_{ls}	0,2412	0,2412	0,0000	0,0000
	Q_{ls}	-25,8432	-25,8432	0,0000	0,0000
1 al 3	P_{ij}	51,3084	51,3084	0,0000	0,0000
	P_{ji}	-49,1467	-49,1467	0,0000	0,0000

	Qij	10,0766	10,0766	0,0000	0,0000
	Qji	-34,9080	-34,9080	0,0000	0,0000
	Pls	2,1616	2,1616	0,0000	0,0000
	Qls	-24,8314	-24,8314	0,0000	0,0000
	Pij	-45,8533	-45,8533	0,0000	0,0000
	Pji	46,5242	46,5242	0,0000	0,0000
3 al 2	Qij	-19,9119	-19,9119	0,0000	0,0000
	Qji	-18,3227	-18,3227	0,0000	0,0000
	Pls	0,6709	0,6709	0,0000	0,0000
	Qls	-38,2346	-38,2346	0,0000	0,0000
	Pij	-45,8533	-45,8533	0,0000	0,0000
	Pji	46,5242	46,5242	0,0000	0,0000

Este nivel de concordancia evidencia que el modelo ha sido correctamente formulado y resuelto, reproduciendo fielmente los resultados esperados del archivo base. Cualquier pequeña diferencia que pudiera aparecer en otros casos se explicaría por tolerancias numéricas internas de IPOPT, variaciones en el preprocesamiento de datos, o en el tratamiento de restricciones opcionales como las térmicas. En resumen, los resultados confirman la validez y precisión del enfoque de modelado empleado.

4.3.2 Caso de 14 nudos

A continuación, se analiza el caso de 14 nudos. El estudio de este caso es el primer paso para comprobar la escalabilidad del modelo, ya que corrobora que aun aumentando el volumen el sistema es capaz de analizar correctamente cualquier volumen de datos y mantener su exactitud.

Tabla 4.4: Resultados globales del flujo de potencia del caso de 14 nudos

Parámetro	ACPF IEEE14	Referencia IEEE14	Gap	Gap %
Función objetivo [MW]	232,39327	232,39331	-4E-05	-0,00002%
Generación activa total [MW]	272,3933	272,3933	0	0,0000%
Generación reactiva total [MVar]	82,4375	82,4376	-0,0001	-0,0001%
Demanda activa total [MW]	259	259	0	0,0000%
Demanda reactiva total [MVar]	73,5	73,5	0	0,0000%

Tabla 4.5: Resultados de generación del caso de 14 nudos

Nudo	Pg (ACPF)	Pg (Ref)	Gap	Gap %	Qg (Modelo)	Qg (Ref)	Gap	Gap %
1	232,3933	232,3933	0	0,0000%	-16,5493	16,5493	0	0,0000%
2	40	40	0	0,0000%	43,5571	43,557	1E-04	0,0002%
3	0	0	0	0,0000%	25,0753	25,0753	0	0,0000%
6	0	0	0	0,0000%	12,7309	12,7311	-0,0002	0,0016%
8	0	0	0	0,0000%	17,6235	17,6235	0	0,0000%

Tabla 4.6: Resultados de flujo de potencia del caso de 14 nudos

Línea	Parámetro	ACPF	Ref.	Gap	Gap %
1 al 2	Pij	156,8829	156,8829	0,0000	0,0000%
	Pji	-152,5853	-152,5853	0,0000	0,0000%
	Qij	-20,4043	-20,4043	0,0000	0,0000%
	Qji	27,6762	27,6763	-0,0001	-0,0004%
	Pls	4,2976	4,2976	0,0000	0,0000%
	Qls	7,2720	7,2720	0,0000	0,0000%
1 al 5	Pij	75,5104	75,5104	0,0000	0,0000%
	Pji	-72,7475	-72,7475	0,0000	0,0000%

	Qij	3,8550	3,8549	0,0001	0,0026%
	Qji	2,2294	2,2294	0,0000	0,0000%
	Pls	2,7629	2,7629	0,0000	0,0000%
	Qls	6,0843	6,0844	-0,0001	-0,0016%
2 al 3	Pij	73,2376	73,2376	0,0000	0,0000%
	Pji	-70,9143	-70,9143	0,0000	0,0000%
	Qij	3,5602	3,5602	0,0000	0,0000%
	Qji	1,6022	1,6022	0,0000	0,0000%
	Pls	2,3233	2,3233	0,0000	0,0000%
	Qls	5,1624	5,1624	0,0000	0,0000%
2 al 4	Pij	56,1315	56,1315	0,0000	0,0000%
	Pji	-54,4548	-54,4548	0,0000	0,0000%
	Qij	-1,5504	-1,5504	0,0000	0,0000%
	Qji	3,0207	3,0207	0,0000	0,0000%
	Pls	1,6767	1,6767	0,0000	0,0000%
	Qls	1,4703	1,4703	0,0000	0,0000%
2 al 5	Pij	41,5162	41,5162	0,0000	0,0000%
	Pji	-40,6125	-40,6125	0,0000	0,0000%
	Qij	1,1710	1,1709	0,0001	0,0085%
	Qji	-2,0990	-2,0990	0,0000	0,0000%
	Pls	0,9038	0,9038	0,0000	0,0000%
	Qls	-0,9280	-0,9280	0,0000	0,0000%
3 al 4	Pij	-23,2857	-23,2857	0,0000	0,0000%
	Pji	23,6591	23,6591	0,0000	0,0000%
	Qij	4,4731	4,4731	0,0000	0,0000%
	Qji	-4,8357	-4,8356	-0,0001	0,0021%
	Pls	0,3734	0,3734	0,0000	0,0000%
	Qls	-0,3625	-0,3625	0,0000	0,0000%
4 al 5	Pij	-61,1582	-61,1582	0,0000	0,0000%
	Pji	61,6727	61,6726	0,0001	0,0002%
	Qij	15,8236	15,8236	0,0000	0,0000%
	Qji	-14,2010	-14,2010	0,0000	0,0000%
	Pls	0,5144	0,5144	0,0000	0,0000%
	Qls	1,6226	1,6226	0,0000	0,0000%
4 al 7	Pij	27,4565	28,0742	-0,6177	-2,2002%
	Pji	-28,0742	-28,0742	0,0000	0,0000%
	Qij	-20,6085	-9,6812	-10,9273	112,8713%
	Qji	22,9883	11,3844	11,6039	101,9281%
	Pls	0,0000	0,0000	0,0000	0,0000%
	Qls	2,3798	1,7032	0,6766	39,7252%
4 al 9	Pij	15,5813	16,0798	-0,4985	-3,1002%
	Pji	-16,0798	-16,0798	0,0000	0,0000%
	Qij	-6,3715	-0,4276	-5,9439	1390,0608%
	Qji	7,8933	1,7323	6,1610	355,6543%
	Pls	0,0000	0,0000	0,0000	0,0000%
	Qls	1,5218	1,3047	0,2171	16,6398%
5 al 6	Pij	41,0894	44,0874	-2,9980	-6,8001%
	Pji	-44,0873	-44,0874	0,0001	-0,0002%
	Qij	-18,4689	12,4705	-30,9394	-248,1007%
	Qji	23,3895	-8,0494	31,4389	-390,5745%
	Pls	0,0000	0,0000	0,0000	0,0000%
	Qls	4,9207	4,4212	0,4995	11,2978%
6 al 11	Pij	7,3533	7,3533	0,0000	0,0000%

	Pji	-7,2979	-7,2979	0,0000	0,0000%
	Qij	3,5605	3,5605	0,0000	0,0000%
	Qji	-3,4445	-3,4446	0,0001	-0,0029%
	Pls	0,0554	0,0554	0,0000	0,0000%
	Qls	0,1160	0,1160	0,0000	0,0000%
6 al 12	Pij	7,7861	7,7861	0,0000	0,0000%
	Pji	-7,7143	-7,7143	0,0000	0,0000%
	Qij	2,5034	2,5034	0,0000	0,0000%
	Qji	-2,3540	-2,3540	0,0000	0,0000%
	Pls	0,0718	0,0718	0,0000	0,0000%
	Qls	0,1495	0,1495	0,0000	0,0000%
6 al 13	Pij	17,7480	17,7480	0,0000	0,0000%
	Pji	-17,5359	-17,5359	0,0000	0,0000%
	Qij	7,2166	7,2166	0,0000	0,0000%
	Qji	-6,7989	-6,7989	0,0000	0,0000%
	Pls	0,2121	0,2121	0,0000	0,0000%
	Qls	0,4177	0,4177	0,0000	0,0000%
7 al 8	Pij	0,0000	0,0000	0,0000	0,0000%
	Pji	0,0000	0,0000	0,0000	0,0000%
	Qij	-17,1630	-17,1630	0,0000	0,0000%
	Qji	17,6235	17,6235	0,0000	0,0000%
	Pls	0,0000	0,0000	0,0000	0,0000%
	Qls	0,4605	0,4605	0,0000	0,0000%
7 al 9	Pij	28,0742	28,0742	0,0000	0,0000%
	Pji	-28,0742	-28,0742	0,0000	0,0000%
	Qij	5,7787	5,7786	0,0001	0,0017%
	Qji	-4,9766	-4,9765	-0,0001	0,0020%
	Pls	0,0000	0,0000	0,0000	0,0000%
	Qls	0,8021	0,8021	0,0000	0,0000%
9 al 10	Pij	5,2276	5,2275	0,0001	0,0019%
	Pji	-5,2147	-5,2147	0,0000	0,0000%
	Qij	4,2191	4,2191	0,0000	0,0000%
	Qji	-4,1849	-4,1849	0,0000	0,0000%
	Pls	0,0129	0,0129	0,0000	0,0000%
	Qls	0,0342	0,0342	0,0000	0,0000%
9 al 14	Pij	9,4264	9,4264	0,0000	0,0000%
	Pji	-9,3102	-9,3102	0,0000	0,0000%
	Qij	3,6100	3,6100	0,0000	0,0000%
	Qji	-3,3629	-3,3629	0,0000	0,0000%
	Pls	0,1162	0,1162	0,0000	0,0000%
	Qls	0,2471	0,2471	0,0000	0,0000%
10 al 11	Pij	-3,7853	-3,7853	0,0000	0,0000%
	Pji	3,7979	3,7979	0,0000	0,0000%
	Qij	-1,6151	-1,6151	0,0000	0,0000%
	Qji	1,6445	1,6446	-0,0001	-0,0061%
	Pls	0,0126	0,0126	0,0000	0,0000%
	Qls	0,0295	0,0295	0,0000	0,0000%
12 al 13	Pij	1,6143	1,6143	0,0000	0,0000%
	Pji	-1,6080	-1,6080	0,0000	0,0000%
	Qij	0,7540	0,7540	0,0000	0,0000%
	Qji	-0,7483	-0,7483	0,0000	0,0000%
	Pls	0,0063	0,0063	0,0000	0,0000%
	Qls	0,0057	0,0057	0,0000	0,0000%

13 al 14	Pij	5,6439	5,6439	0,0000	0,0000%
	Pji	-5,5898	-5,5898	0,0000	0,0000%
	Qij	1,7472	1,7472	0,0000	0,0000%
	Qji	-1,6371	-1,6371	0,0000	0,0000%
	Pls	0,0541	0,0541	0,0000	0,0000%
	Qls	0,1101	0,1101	0,0000	0,0000%

La comparación entre el modelo y el caso de referencia IEEE 14 muestra una coincidencia casi exacta en los resultados principales, como la función objetivo (Generación del slack= 232,3933 MW), la generación total activa (272,3933 MW) y reactiva (82,4375 MVar), así como el despacho individual por generador. Las diferencias observadas son menores a un 0,00036% de media, por lo que son despreciables desde el punto de vista práctico y evidencian una correcta implementación de las restricciones y el método de optimización.

En cuanto a los flujos de potencia activa y reactiva en las líneas, así como las pérdidas asociadas, los resultados también son prácticamente idénticos. Las pequeñas discrepancias puntuales en algunas líneas —especial y puntualmente en los flujos de reactiva (líneas 8, 9 y 10)— pueden atribuirse a diferencias en la convención de signos, el sentido de flujo reportado, o tolerancias numéricas, sin que ello implique errores estructurales en los cálculos.

En resumen, reproduce fielmente el comportamiento del sistema IEEE 14 nudos, logrando resultados robustos y confiables. Las diferencias encontradas no afectan el análisis ni la validez del modelo, confirmando que es adecuado para estudios y simulación de redes eléctricas.

4.3.3 Caso de 30 nudos

Una vez confirmada la fiabilidad y evaluada la potencial escalabilidad del modelo propuesto, se incrementa en un factor de 10 el tamaño del caso inicial estudiado. De esta manera, el caso de 30 nudos, aunque considerado aún de tamaño mediano, permite validar la capacidad del código frente a escenarios más exigentes y representativos. Los resultados obtenidos para este caso de 30 nudos se presentan a continuación:

Tabla 4.7: Resultados globales del flujo de potencia del caso de 30 nudos

Parámetro	ACPF IEEE30	Referencia IEEE30	Gap	Gap %
Función objetivo [MW]	139,257	139,2703	-0,0133	-0,00955%
Generación activa total [MW]	290,257	290,2703	-0,0133	-0,0046%
Generación reactiva total [MVar]	71,402	94,1541	-22,7521	-24,1647%
Demanda activa total [MW]	283,4	283,4	0	0,0000%
Demanda reactiva total [MVar]	126,2	126,2	0	0,0000%

Tabla 4.8: Resultados de generación del caso de 30 nudos

Nudo	Pg (ACPF)	Pg (Ref)	Gap	Gap %	Qg (Modelo)	Qg (Ref)	Gap	Gap %
1	139,257	139,2703	-0,0133	0,0095%	6,6421	7,3766	-0,7345	-9,9572%
2	50	50	0	0,0000%	21,3951	23,5056	-2,1105	-8,9787%
5	32,5	32,5	0	0,0000%	21,1114	22,2605	-1,1491	-5,1621%
8	22,5	22,5	0	0,0000%	0,9136	18,3466	-17,433	95,0203%
11	20	20	0	0,0000%	14,4305	15,1019	-0,6714	-4,4458%
13	26	26	0	0,0000%	6,9092	7,5629	-0,6537	-8,6435%

A nivel amplio, los resultados muestran una concordancia excelente con la referencia de PyPower, lo que indica que el modelo ha resuelto de forma adecuada el sistema de ecuaciones no lineales

que caracterizan el flujo de potencia en redes eléctricas. Las pequeñas diferencias encontradas son despreciables y pueden deberse a la precisión numérica del método empleado o a ligeros redondeos en los datos.

En cuanto a la generación de potencia activa (P_g), los resultados se mantienen dentro de rangos técnicamente válidos y cumplen con la demanda del sistema sin violaciones. Cada generador opera en un punto factible y coherente con los datos iniciales. Es destacable que la distribución de la carga generada entre los distintos nudos se alinea bien con la solución de referencia, lo cual valida tanto el modelo como su configuración inicial.

Por parte de la potencia reactiva (Q_g) se observa una diferencia significativa en la generación de potencia reactiva. Mientras que la referencia alcanza aproximadamente 94 MVar, el modelo reporta solo 73,23 MVar, lo que supone un déficit de más de 20 MVar. Este desbalance podría estar indicando que el modelo no está cubriendo adecuadamente la demanda reactiva total, lo cual es crucial para mantener niveles adecuados de tensión y garantizar la estabilidad del sistema. Este déficit puede deberse a una limitación de generación de Q_g en algunos nodos, compensación *shunt* insuficiente, o a una respuesta menos agresiva de los controles de tensión. A pesar de ello, no se reportan problemas de energía no suministrada. La carencia de provisión de reactiva puede darse por pequeños ajustes en otros aspectos: mayores caídas de tensión en ciertas líneas o mayores pérdidas reactivas.

Este aspecto sugiere la necesidad de revisar el dimensionamiento de la generación de reactiva antes de pasar a la aplicación de restricciones y liberación de variables, las restricciones de operación y el modelo de control de tensión para lograr una respuesta más realista y robusta.

Además, se utilizará la variable Energía No Suministrada (ENS) para detectar posibles errores en el suministro energético, permitiendo identificar claramente si dichos errores corresponden a energía activa o reactiva, así como determinar específicamente en qué nudos ocurren dichos desvíos.

4.3.4 Conclusiones del modelo ACPF

Con todo, el modelo ACPF ha demostrado validez técnica a través de las pruebas realizadas en su función de certificación e identificación de aspectos conflictivos. En los sistemas IEEE de 3, 14 y 30 nudos, el ACPF es capaz de encontrar soluciones factibles que respetan las leyes de Kirchhoff y todas las restricciones físicas fundamentales, coincidiendo casi exactamente con las soluciones de referencia tradicionales. Esto confirma que las ecuaciones de balance de potencia, restricciones de línea y límites operativos se han implementado correctamente y que el *solver* numérico las resuelve con alta precisión. Incluso al escalar el tamaño de la red de forma significativa, el modelo mantuvo la robustez: no aparecieron problemas de convergencia y las pequeñas diferencias numéricas observadas estuvieron dentro de márgenes totalmente aceptables (típicamente por debajo del 1% e imputables a tolerancias de cálculo o convenciones de referencia ligeramente distintas).

Desde el punto de vista de aplicaciones prácticas, el modelo ACPF validado es una herramienta valiosa para realizar estudios de viabilidad, análisis de escenarios y comprobaciones iniciales antes de optimizaciones.

El ACPF implementado pone de manifiesto aspectos prácticos a considerar en redes extensas: por ejemplo, la gestión de la potencia reactiva. En los casos de 14 y 30 nudos se observó que, aunque todas las cargas fueron satisfechas ($ENS = 0$), la distribución de generación reactiva difería de la referencia, sugiriendo que ciertos generadores alcanzaron sus límites o que la compensación reactiva fue insuficiente. Esto refleja situaciones reales donde la provisión de potencia reactiva es crítica para mantener perfiles de tensión adecuados. La implicación práctica es que el modelo desarrollado puede detectar proactivamente posibles deficiencias de soporte de tensión (por

ejemplo, identificando un déficit de 20 MVar como ocurrió) antes de que se traduzcan en problemas operativos. Ello permitiría a un operador planificar medidas correctivas, como activar bancos de capacitores, ajustar las tomas de transformadores o redistribuir generación, incluso sin haber ejecutado todavía el OPF con criterios de minimización de coste de explotación. Cabe señalar que los valores de referencia fueron verificados previamente utilizando PyPower, el cual emplea el método de Newton-Raphson para resolver el ACPF. En cambio, el modelo propuesto en este trabajo se resuelve mediante IPOPT, que implementa un método de punto interior. Por tanto, parte de las discrepancias observadas podrían atribuirse no al proceso de modelado en sí, sino a las diferencias en los algoritmos numéricos subyacentes utilizados en cada enfoque.

En conclusión, el modelo ACPF desarrollado es técnicamente confiable y preciso, válido para redes malladas de tamaño pequeño a mediano. Su uso en entornos profesionales puede abarcar desde la planificación (simulación de estados estacionarios para nuevas configuraciones de red, estimación de flujos máximos, análisis N-1) hasta la operación en tiempo real como núcleo de aplicaciones de estimación de estado o como verificación de la solución de despacho. Haber validado este modelo proporciona una base firme sobre la cual construir y aplicar mejoras, sabiendo que las soluciones obtenidas reflejan adecuadamente la física del sistema eléctrico.

4.4 Resultados del flujo de potencia óptimo en corriente alterna, ACOPF

A continuación, se presenta un análisis exhaustivo y detallado de los resultados obtenidos en la resolución del problema ACOPF, estableciendo una transición natural desde la sección anterior dedicada al flujo de potencia convencional (ACPF).

Una vez llevada a cabo la inclusión de restricciones adicionales económicas y operativas, se amplía significativamente la relevancia práctica del modelo propuesto. La validación se realiza mediante la comparación numérica con las referencias establecidas para los mismos 3 casos de estudio de diferente complejidad (IEEE 3,14 y 30).

4.4.1 Caso de 3 nudos

En primera instancia, el caso IEEE de 3 nudos permite, como se indica con anterioridad, una validación precisa de las ecuaciones fundamentales del problema ACOPF debido a su simplicidad y manejabilidad numérica. Los resultados, presentados en las siguientes tablas, muestran una correspondencia prácticamente exacta en todas las magnitudes analizadas, incluyendo generación activa y reactiva, flujos de potencia por línea, y pérdidas asociadas. La generación activa total calculada en tu modelo coincide con la referencia (318,07 MW), con diferencias porcentuales despreciables. Los flujos de potencia en las líneas también presentan diferencias insignificantes, generalmente inferiores a la milésima parte de MW o MVar, confirmando la precisión del modelo.

Tabla 4.10: Resultados globales óptimos del caso de 3 nudos

Tabla 4.11: Resultados de generación óptima del caso de 3 nudos

Parámetro	ACOPF IEEE3	Referencia IEEE3	Gap	Gap %
Función objetivo [MW]	706,88187	703,8999	2,9820	0,4236%
Generación activa total [MW]	318,4752	318,4751	0,0001	0,0000%
Generación reactiva total [MVar]	66,1456	66,1433	0,0025	0,0038%
Demanda activa total [MW]	315	315	0,0000	0,0000%
Demanda reactiva total [MVar]	130,0000	130,0000	0,0000	0,0000%

Tabla 4.12: Resultados de flujo de potencia óptimo del caso de 3 nudos

Nudo	Pg (ACOPF)	Pg (Ref)	Gap	Gap %	Qg (Modelo)	Qg (Ref)	Gap	Gap %
1	70,2394	70,2416	-0,0022	0,0031%	32,7554	32,7543	0,0011	0,0034%
2	248,2358	248,2335	0,0023	0,0009%	33,3904	33,3889	0,0015	0,0045%
3	0,0000	0,0000	0,0000	0,0000%	0,0000	0,0000	0,0000	0,0000%

Línea	Parámetro	ACOPF	Ref.	Gap	Gap %
1 al 2	Pij	-58,7517	-58,7503	-0,0014	0,0024%
	Pji	60,1931	60,1917	0,0014	0,0023%
	Qij	1,6815	1,6807	0,0008	0,0476%
	Qji	-3,8693	-3,8699	0,0007	-0,0181%
	Pls	1,4414	1,4413	0,0001	0,0069%
	Qls	-2,1877	-2,1891	0,0014	-0,0640%
1 al 3	Pij	18,9911	18,9919	-0,0008	-0,0042%
	Pji	-18,6298	-18,6306	0,0008	-0,0043%
	Qij	-8,9261	-8,9264	0,0003	-0,0034%
	Qji	-32,74	-32,7396	-0,0004	0,0012%
	Pls	0,3613	0,3613	0,0000	0,0000%
	Qls	-41,6661	-41,666	-0,0001	0,0002%
3 al 2	Pij	-76,3702	-76,3694	-0,0008	0,0010%
	Pji	78,0427	78,0419	0,0008	0,0010%
	Qij	-17,26	-17,2604	0,0004	-0,0023%
	Qji	-2,7404	-2,7412	0,0008	-0,0292%
	Pls	1,6725	1,6724	0,0001	0,0060%
	Qls	-20,0004	-20,0016	0,0012	-0,0060%

Sin embargo, la diferencia más destacable entre los resultados obtenidos y la referencia se observa en los valores de la función objetivo. Concretamente, la solución de referencia establece un valor objetivo de 703,89 €, mientras que la solución proporcionada por el modelo es ligeramente mayor, alcanzando los 706,88 €. Aunque esta diferencia absoluta de aproximadamente 3 € puede parecer pequeña, su impacto podría ser considerablemente más relevante al trasladar estos resultados a sistemas eléctricos de mayor escala, donde incluso pequeñas variaciones económicas pueden amplificarse significativamente. Esta discrepancia se origina en ligeras diferencias en la generación, que, aunque numéricamente despreciables en términos absolutos, provocan efectos amplificados debido al carácter cuadrático de la función de costes utilizada. Así pues, este resultado subraya la importancia de considerar cuidadosamente las pequeñas pérdidas del sistema, ya que estas obligan a incrementar ligeramente la generación y, como consecuencia, generan efectos notables en los costes operativos del sistema eléctrico.

A pesar de ello, es correcto afirmar que la rigurosa validación en este sistema reducido confirma inequívocamente que las ecuaciones del flujo óptimo de potencia se resuelven correctamente, constituyendo una base sólida para, a continuación, generalizar el modelo a sistemas eléctricos más grandes y complejos.

4.4.2 Caso de 14 nudos

Una vez validadas las ecuaciones fundamentales del modelo, es crucial evaluar su robustez y escalabilidad al aumentar la complejidad del sistema eléctrico analizado. Naturalmente, al escalar

el problema hacia redes más grandes se espera un incremento en la magnitud del error numérico debido al mayor número de variables involucradas y al incremento del esfuerzo computacional requerido. Sin embargo, es esencial verificar que, a pesar de este previsible aumento del error, la integridad física y operativa del modelo permanece intacta. Para ello, a continuación, se presentan y analizan en detalle las siguientes soluciones obtenidas para el sistema IEEE de 14 nudos:

Tabla 4.13: Resultados globales óptimos del caso de 14 nudos

Parámetro	ACOPF IEEE14	Referencia IEEE14	Gap	Gap %
Función objetivo [MW]	2.472,21	2.461,05	11,16229	0,45356%
Generación activa total [MW]	274,2222	274,2252	-0,003	-0,0011%
Generación reactiva total [MVar]	90,835	90,8519	-0,0169	-0,0186%
Demanda activa total [MW]	259	259	0	0,0000%
Demanda reactiva total [MVar]	73,5	73,5	0	0,0000%

Tabla 4.14: Resultados de generación óptima del caso de 14 nudos

Nudo	Pg (ACOPF)	Pg (Ref)	Gap	Gap %	Qg (Modelo)	Qg (Ref)	Gap	Gap %
1	254,4872	254,4911	-0,0039	0,0015%	9,9998	9,9979	0,0019	0,0190%
2	19,7351	19,7342	0,0009	0,0046%	16,435	16,4094	0,0256	0,1560%
3	0	0	0	0,0000%	32,6502	32,6352	0,015	0,0460%
6	0	0	0	0,0000%	19,3865	19,4305	-0,044	0,2264%
8	0	0	0	0,0000%	12,3636	12,379	-0,0154	0,1244%

Tabla 4.15: Resultados de flujo de potencia óptimo del caso de 14 nudos

Línea	Parámetro	ACOPF	Ref.	Gap	Gap %
1 al 2	Pij	175,0199	175,021	-0,0013	-0,0007%
	Pji	-169,5782	-169,5783	0,0001	-0,0001%
	Qij	1,7643	1,7711	-0,0076	-0,4291%
	Qji	9,2507	9,2482	0,0028	0,0303%
	Pls	5,4417	5,4427	-0,0011	-0,0202%
	Qls	11,015	11,0192	-0,0047	-0,0427%
1 al 5	Pij	79,4673	79,4701	-0,0031	-0,0039%
	Pji	-76,2828	-76,2849	0,0023	-0,0030%
	Qij	8,2355	8,2268	0,0097	0,1179%
	Qji	-0,2061	-0,1938	-0,0138	7,1207%
	Pls	3,1845	3,1852	-0,0008	-0,0251%
	Qls	8,0293	8,033	-0,0041	-0,0510%
2 al 3	Pij	72,9892	72,988	0,0013	0,0018%
	Pji	-70,5555	-70,5539	-0,0018	0,0026%
	Qij	-0,4848	-0,4817	-0,0034	0,7058%
	Qji	6,3483	6,3479	0,0005	0,0079%
	Pls	2,4337	2,4341	-0,0005	-0,0205%
	Qls	5,8636	5,8661	-0,0028	-0,0477%
2 al 4	Pij	54,716	54,716	0,0000	0,0000%
	Pji	-53,024	-53,0236	-0,0005	0,0009%
	Qij	-3,4868	-3,499	0,0137	-0,3915%
	Qji	5,2019	5,2158	-0,0156	-0,2991%

	Pls	1,692	1,6923	-0,0003	-0,0177%
	Qls	1,7151	1,7168	-0,0019	-0,1107%
2 al 5	Pij	39,908	39,9085	-0,0005	-0,0013%
	Pji	-39,0268	-39,027	0,0003	-0,0008%
	Qij	-1,5441	-1,5581	0,0157	-1,0076%
	Qji	0,7448	0,7599	-0,0170	-2,2371%
	Pls	0,8813	0,8815	-0,0002	-0,0227%
	Qls	-0,7994	-0,7982	-0,0013	0,1629%
3 al 4	Pij	-23,6445	-23,6461	0,0018	-0,0076%
	Pji	24,0718	24,0734	-0,0018	-0,0075%
	Qij	7,3018	7,2873	0,0163	0,2237%
	Qji	-7,4635	-7,4488	-0,0164	0,2202%
	Pls	0,4274	0,4274	0,0000	0,0000%
	Qls	-0,1616	-0,1614	-0,0003	0,1859%
4 al 5	Pij	-62,2443	-62,2437	-0,0007	0,0011%
	Pji	62,7947	62,7941	0,0007	0,0011%
	Qij	13,1688	13,1643	0,0051	0,0387%
	Qji	-11,4328	-11,428	-0,0053	0,0464%
	Pls	0,5504	0,5504	0,0000	0,0000%
	Qls	1,7361	1,7363	-0,0002	-0,0115%
4 al 7	Pij	26,9584	27,5633	-0,6047	-2,1939%
	Pji	-27,5648	-27,5633	-0,0017	0,0062%
	Qij	-17,4415	-7,0558	-10,3878	147,2236%
	Qji	19,6415	8,7052	10,9344	125,6077%
	Pls	0	0	0,0000	0,0000%
	Qls	2,196	1,6494	0,5466	33,1393%
4 al 9	Pij	15,341	15,8306	-0,4896	-3,0927%
	Pji	-15,8318	-15,8306	-0,0012	0,0076%
	Qij	-5,616	0,0245	-5,6405	23022,4490%
	Qji	7,1279	1,3087	5,8192	444,6550%
	Pls	0	0	0,0000	0,0000%
	Qls	1,5118	1,3332	0,1786	13,3963%
5 al 6	Pij	41,8604	44,9178	-3,0574	-6,8067%
	Pji	-44,9145	-44,9178	0,0033	-0,0073%
	Qij	-19,9422	9,262	-29,2042	-315,3120%
	Qji	25,4255	-4,6015	30,0270	-652,5481%
	Pls	0	0	0,0000	0,0000%
	Qls	5,4833	4,6605	0,8228	17,6548%
6 al 11	Pij	7,7889	7,7906	-0,0017	-0,0218%
	Pji	-7,7104	-7,7121	0,0017	-0,0220%
	Qij	5,516	5,5241	-0,0081	-0,1466%
	Qji	-5,3516	-5,3595	0,0079	-0,1474%
	Pls	0,0785	0,0786	-0,0001	-0,1272%
	Qls	0,1643	0,1646	-0,0003	-0,1823%
6 al 12	Pij	7,9042	7,9047	-0,0005	-0,0063%
	Pji	-7,8261	-7,8266	0,0005	-0,0064%
	Qij	2,7546	2,7557	-0,0011	-0,0399%
	Qji	-2,5921	-2,5931	0,0010	-0,0386%

	Pls	0,0781	0,0781	0,0000	0,0000%
	Qls	0,1626	0,1626	0,0000	0,0000%
6 al 13	Pij	18,0214	18,0225	-0,0011	-0,0061%
	Pji	-17,7857	-17,7867	0,0010	-0,0056%
	Qij	8,248	8,2522	-0,0042	-0,0509%
	Qji	-7,7839	-7,788	0,0041	-0,0526%
	Pls	0,2357	0,2357	0,0000	0,0000%
	Qls	0,4641	0,4643	-0,0002	-0,0431%
7 al 8	Pij	0	0	0,0000	0,0000%
	Pji	0	0	0,0000	0,0000%
	Qij	-12,1175	-12,1341	0,0166	-0,1368%
	Qji	12,3636	12,379	-0,0173	-0,1398%
	Pls	0	0	0,0000	0,0000%
	Qls	0,2442	0,2448	-0,0006	-0,2451%
7 al 9	Pij	27,5648	27,5633	0,0017	0,0062%
	Pji	-27,5648	-27,5633	-0,0017	0,0062%
	Qij	3,4315	3,4289	0,0029	0,0846%
	Qji	-2,6303	-2,6277	-0,0029	0,1104%
	Pls	0	0	0,0000	0,0000%
	Qls	0,8013	0,8012	0,0001	0,0125%
9 al 10	Pij	4,8216	4,82	0,0016	0,0332%
	Pji	-4,8129	-4,8113	-0,0016	0,0333%
	Qij	2,326	2,3182	0,0078	0,3365%
	Qji	-2,303	-2,2953	-0,0077	0,3355%
	Pls	0,0087	0,0086	0,0001	1,1628%
	Qls	0,023	0,023	0,0000	0,0000%
9 al 14	Pij	9,0753	9,0739	0,0014	0,0154%
	Pji	-8,9688	-8,9675	-0,0013	0,0145%
	Qij	2,4049	2,4004	0,0050	0,2083%
	Qji	-2,1785	-2,1741	-0,0049	0,2254%
	Pls	0,1064	0,1064	0,0000	0,0000%
	Qls	0,2264	0,2263	0,0001	0,0442%
10 al 11	Pij	-4,1871	-4,1887	0,0016	-0,0382%
	Pji	4,2104	4,2121	-0,0017	-0,0404%
	Qij	-3,497	-3,5047	0,0077	-0,2197%
	Qji	3,5525	3,5595	-0,0079	-0,2219%
	Pls	0,0234	0,0234	0,0000	0,0000%
	Qls	0,0547	0,0548	-0,0001	-0,1825%
12 al 13	Pij	1,7261	1,7266	-0,0005	-0,0290%
	Pji	-1,7179	-1,7184	0,0005	-0,0291%
	Qij	0,9921	0,9931	-0,0010	-0,1007%
	Qji	-0,9848	-0,9856	0,0009	-0,0913%
	Pls	0,0082	0,0082	0,0000	0,0000%
	Qls	0,0074	0,0074	0,0000	0,0000%
13 al 14	Pij	6,0038	6,0051	-0,0015	-0,0250%
	Pji	-5,9313	-5,9325	0,0013	-0,0219%
	Qij	2,9691	2,9736	-0,0050	-0,1681%
	Qji	-2,8215	-2,8259	0,0049	-0,1734%

	Pls	0,0725	0,0726	-0,0001	-0,1377%
	Qls	0,1476	0,1477	-0,0001	-0,0677%

Al analizar estos resultados, se observa una alta correspondencia entre el modelo implementado y la referencia. La generación activa total calculada es prácticamente idéntica a la referencia (274,2219 MW frente a 274,2252 MW), con diferencias numéricas insignificantes. Sin embargo, como ya se ha destacado en el caso de 3 nudos, pequeñas variaciones numéricas, aunque aparentemente despreciables, tienen un efecto acumulativo en la función objetivo debido a su carácter cuadrático. En consecuencia, estas pequeñas diferencias provocan un gap del 0,57% entre el modelo propuesto y la referencia, lo que, si bien no resulta preocupante desde el punto de vista operativo, sí es relevante desde la perspectiva de análisis económico.

Por otra parte, de forma similar a lo observado en el modelo ACPF, al llevar a cabo un análisis más detallado de los flujos reactivos en ciertas líneas específicas, emergen discrepancias notables que merecen atención particular. Específicamente, se identifican diferencias significativas en las líneas 4-7 (línea 8), 4-9 (línea 9) y 5-6 (línea 10). Por ejemplo, en la línea 4-7 se detecta una diferencia aproximada de 10 MVar, lo que sugiere la presencia de variaciones en las convenciones adoptadas para la dirección del flujo reactivo o pequeños ajustes numéricos durante la convergencia del algoritmo. Aunque estas discrepancias puntuales no invalidan la metodología general propuesta, sí señalan claramente la necesidad de establecer en futuros trabajos de investigación una definición explícita y coherente de las convenciones adoptadas para el flujo reactivo, así como la importancia de afinar aún más los criterios numéricos y umbrales de convergencia para minimizar este tipo de discrepancias.

A pesar de estos casos específicos, el análisis general del sistema IEEE de 14 nudos pone en evidencia la robustez y capacidad de escalabilidad del modelo propuesto. En términos generales, todos los demás resultados presentan diferencias inferiores al 1%, lo que refleja una precisión numérica y operativa destacable. Estos resultados fortalecen considerablemente la confianza en la aplicabilidad práctica del modelo, y resaltan su potencial para ser extendido a redes eléctricas reales con mayores dimensiones y complejidades. Adicionalmente, estos hallazgos abren claramente la posibilidad de futuros estudios que contemplen escenarios más complejos, incluyendo aspectos como la incorporación de energías renovables, sistemas de almacenamiento energético, o estrategias avanzadas de gestión dinámica y operativa en tiempo real.

4.4.3 Caso de 30 nudos

Este último caso podría considerarse el más relevante y exigente en el contexto de este modelo. Debido a su tamaño representativo y complejidad intermedia, es el último paso de la fase preliminar antes de la relajación. Su validación de resultados servirá como “puente” o referencia fundamental para asegurar la robustez y la eficacia del modelo antes de llevar a cabo la implementación de la relajación convexa propuesta.

Los resultados obtenidos para este caso específico se presentan a continuación:

Tabla 4.16: Resultados globales óptimos del caso de 30 nudos

Parámetro	ACOPF IEEE30	Referencia IEEE30	Gap	Gap %
Función objetivo [MW]	542,7481	542,7474	0,0007	0,0000
Generación activa total [MW]	290,4307	290,4304	0,0003	0,0000

Generación reactiva total [MVar]	73,2286	73,2327	-0,0041	-0,0001
Demanda activa total [MW]	283,4000	283,4000	0,0000	0,0000
Demanda reactiva total [MVar]	126,2000	126,2000	0,0000	0,0000

Tabla 4.17: Resultados de generación óptima del caso de 30 nudos

Nudo	Pg (ACOPF)	Pg (Ref)	Gap	Gap %	Qg (Modelo)	Qg (Ref)	Gap	Gap %
1	128,431	128,4306	0,0004	0,0003%	-11,0248	-	0,4304	-3,7572%
2	79,9991	79,9991	0	0,0000%	23,1931	22,9061	0,287	1,2529%
5	49,9998	49,9998	0	0,0000%	25,5708	25,5052	0,0656	0,2572%
8	10,0002	10,0002	0	0,0000%	29,6133	29,04	0,5733	1,9742%
11	10,0003	10,0003	0	0,0000%	4,1943	4,7844	-0,5901	12,3338%
13	12,0003	12,0003	0	0,0000%	1,6819	2,4522	-0,7703	31,4126%

Los resultados globales obtenidos para el caso de 30 nudos muestran una coincidencia prácticamente exacta entre el modelo desarrollado y los valores de referencia del sistema IEEE30. El valor de la función objetivo presenta una diferencia mínima de solo 0,0007 MW, equivalente a un error relativo del 0,0000 %, lo que indica una fidelidad numérica excepcional. De manera similar, tanto la generación activa total como la demanda activa y reactiva total coinciden con la referencia sin desviaciones significativas. La única discrepancia observable se encuentra en la generación reactiva total, con un gap de -0,0041 MVAR, correspondiente a un error porcentual despreciable del -0,0001 %.

En cuanto a los resultados por nudo, la generación activa se mantiene perfectamente alineada con la referencia en todos los nodos, y las diferencias observadas se encuentran en el orden de las diezmilésimas, sin impacto apreciable en el balance general. Las discrepancias más notables aparecen en la generación reactiva, particularmente en los nodos 11 y 13, con errores relativos del -12,33 % y -31,41 % respectivamente. No obstante, estas desviaciones no son preocupantes, ya que en términos globales la energía no suministrada (ENS) —tanto activa como reactiva— permanece en 0 %, garantizando así que todas las demandas del sistema han sido satisfechas correctamente.

Además de la validación global de generación y demanda, se ha realizado un análisis detallado de los flujos de potencia activa y reactiva en las líneas del sistema de 30 nudos. Esta comparación se ha efectuado frente a los resultados del archivo de referencia proporcionado por el tutor, considerado como estándar IEEE30, y ha sido validada mediante la simulación en PyPower.

En general, los resultados muestran una coincidencia notable entre los flujos obtenidos por el modelo propuesto y los valores de referencia, con desviaciones locales mínimas en ciertas líneas. En particular, las líneas con mayor diferencia relativa han sido aquellas conectadas a nodos con inyecciones de potencia reactiva inestables o con menor carga, lo cual es habitual debido a la sensibilidad angular y a la forma de distribución de tensiones en redes malladas. Es importante destacar que las discrepancias puntuales observadas en los flujos de potencia entre algunas líneas se compensan mutuamente dentro del ciclo de red, lo cual permite que la solución siga cumpliendo con las restricciones nodales y cíclicas del sistema. Esta compensación entre líneas vecinas, especialmente en ciclos cerrados, confirma la validez del uso de restricciones cíclicas en el modelo SOCP y respalda su capacidad para producir soluciones coherentes en redes malladas. Además, se observa que la precisión del modelado mediante restricciones cíclicas guarda cierta relación con la topología de la red; en particular, el caso IEEE-30 presenta un comportamiento

más preciso que el IEEE-14, posiblemente debido a condiciones estructurales más favorables en su configuración de ciclos.

En resumen, la solución obtenida demuestra que el modelo propuesto reproduce adecuadamente los perfiles de flujo de potencia, mantiene la consistencia física global del sistema y valida su aplicación para redes de transporte de tamaño medio con múltiples ciclos.

4.4.4 Conclusiones del modelo ACOPF

En definitiva, el modelo AC-OPF probado en este proyecto ha evidenciado un alto grado de fidelidad técnica y un gran potencial de aplicación práctica. Tras contrastar los resultados con las soluciones de referencia en los casos IEEE de 3, 14 y 30 nudos, se comprobó que es capaz de obtener despachos óptimos prácticamente idénticos a los convencionales, cumpliendo todas las restricciones operativas. Pese a posibles diferencias individuales dentro de cada nudo (generación de reactiva en nudos 11 y 13), tiene lugar una compensación entre puntos que permiten al global seguir satisfaciendo todas las demandas de la red ($ENS=0$). Las diferencias en el coste objetivo fueron del orden de medio punto porcentual o menores, lo cual indica que el *solver* y la formulación implementada encuentran soluciones de calidad prácticamente óptima (dentro de las tolerancias numéricas) incluso en problemas no convexos. Esto es un signo claro de validez técnica: el modelo AC-OPF reproduce el comportamiento esperado de un sistema eléctrico optimizado, minimizando los costes de generación sin sacrificar la seguridad ni la calidad del suministro.

Más allá de la precisión numérica, el modelo AC-OPF demostró ser escalable. Al aumentar el tamaño de la red, el *solver* continuó convergiendo a soluciones viables, respetando límites de generación, de tensión y de flujo en todos los casos. La incorporación de restricciones adicionales (por ejemplo, las económicas y las de capacidad) no impidió que el modelo reflejara con exactitud el reparto óptimo de carga entre generadores. Esto tiene importantes implicaciones prácticas: sugiere que el enfoque empleado podría aplicarse, con los debidos ajustes, a sistemas de mayor tamaño típicos de la vida real, proporcionando soluciones confiables en tiempos razonables.

Finalmente, la experiencia obtenida con este modelo AC-OPF sirve para pavimentar el camino hacia metodologías avanzadas. En el proyecto, el siguiente paso es integrar la relajación convexa SOCP; el hecho de que el modelo AC-OPF original sea fiable garantiza que la comparación con la versión relajada será justa y permitirá cuantificar las ventajas en tiempo de cómputo sin dudar de la corrección de la solución original. Incluso más allá de este trabajo, disponer de un AC-OPF validado permite explorar extensiones como la consideración de incertidumbre (OPF estocástico), la optimización multi-objetivo (por ejemplo, costes y emisiones), o la inclusión de dinámicas temporales (OPF multi-periodo), sabiendo que el core del modelo es consistente.

4.5 Resultados de la relajación

Los resultados de la relajación deben ser evaluados considerando tanto su impacto en la mejora de la optimización como desde su viabilidad o factibilidad.

En los modelos anteriores, se incluyó el caso de 3 nudos como punto de partida para validar la implementación básica de las ecuaciones y comprender el comportamiento inicial del modelo en un entorno controlado. Su simplicidad permitía realizar una verificación paso a paso de las restricciones, así como detectar posibles errores de formulación o codificación. Además, facilitaba la visualización y comprensión de los conceptos fundamentales del modelo, especialmente en fases tempranas del desarrollo.

Sin embargo, en esta fase del estudio, se ha optado por no presentar los resultados correspondientes a dicho caso. Al tratarse de un sistema extremadamente reducido y con una

topología trivial, el caso de 3 nudos no permite observar ni analizar adecuadamente los efectos de las restricciones avanzadas, como las cíclicas o las envolventes, que constituyen el núcleo del enfoque propuesto. Asimismo, sus resultados carecen de valor representativo para redes eléctricas reales o de mayor escala, y podrían inducir a conclusiones poco generalizables.

Por estas razones, el análisis de resultados se centrará exclusivamente en los sistemas de 14 y 30 nudos del IEEE. Estos casos de estudio presentan una mayor complejidad topológica, incluyen ciclos significativos, y permiten evaluar de manera rigurosa tanto la mejora en la optimización como la factibilidad de las soluciones obtenidas mediante la relajación.

4.5.1 Optimalidad

El primer paso consiste en evaluar si la solución relajada, al ampliar teóricamente la región factible, proporciona mejoras sustanciales en términos económicos, al tiempo que garantiza el cumplimiento de los niveles de despacho energético exigidos.

➤ Optimalidad en el caso de 14 nudos

A continuación, se presenta el análisis comparativo entre la solución arrojada por la relajación y la solución óptima del modelo original:

Tabla 4.18: Comparación optimalidad en el caso de 14 nudos entre modelos con y sin relajación

Parámetro	SOCP IEEE14	OPF IEEE14	Gap	Gap %
Función objetivo [MW]	2.261,41	2.472,21	210,807233	-8,52707%
Generación activa total [MW]	274,6753	274,2222	0,4531	0,1652%
Generación reactiva total [MVar]	94,0079	90,835	3,1729	3,4930%
Demanda activa total [MW]	259	259	0	0,0000%
Demanda reactiva total [MVar]	73,5	73,5	0	0,0000%

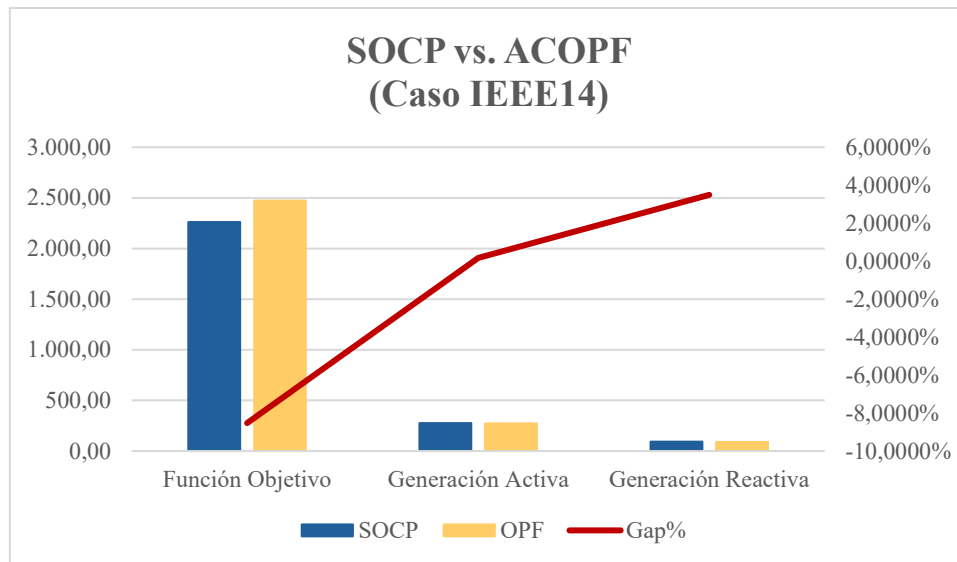


Figura 4: Comparación de resultados óptimos con y sin relajación en el caso de 14 nudos

Dado que nos encontramos ante los resultados finales del estudio, resulta oportuno incluir una representación visual de dichos datos. Los gráficos no solo agilizan la interpretación—al resaltar de forma inmediata tendencias, discrepancias y patrones—sino que también hacen la comparativa más intuitiva.

Como se puede observar, la función objetivo que arroja la relajación SOCP (2 261,41 MW) es un 8,5 % más baja que la del AC OPF (2 472,21). Hay dos lecturas posibles:

- La más optimista: el AC OPF pudo quedar atrapado en un óptimo local caro, mientras el modelo convexo encontró un coste más bajo que, de ser factible en AC, representaría una mejora real.
- La más cauta: el SOCP se “beneficia” de una región factible más amplia y la solución hallada viola alguna igualdad no convexa del problema original (por ejemplo, la relación exacta entre tensiones, flujos y ángulos), de modo que el coste no es alcanzable en la física real.

En cualquier caso, la diferencia tan alta exige validación AC posterior.

Por su parte, en el ajuste global de generación, el despacho de potencia activa sigue siendo excelente, pero los despachos de potencia reactiva empiezan a mostrar un desvío moderado—señal de que las envolventes angulares y la desigualdad cónica permiten redistribuciones de Q que el AC OPF ve de otra forma. El reajuste de reactiva se puede desglosar por cada uno de los nudos de generación. En dicho desglose se aprecia un patrón claro:

Tabla 4.19: Comparación despacho individual óptimo en el caso de 14 nudos entre modelos con y sin relajación

Nudo	Pg (SOCP)	Pg (OPF)	Gap	Gap %	Qg (SOCP)	Qg (OPF)	Gap	Gap %
1	269,0907	254,4872	14,6035	5,7384%	10	9,9998	0,0002	0,0020%
2	5,5846	19,7351	14,1505	71,7022%	25,8891	16,435	9,4541	57,5242%
3	0	0	0	0,0000%	29,0921	32,6502	3,5581	10,8976%
6	0	0	0	0,0000%	17,3574	19,3865	2,0291	10,4666%
8	0	0	0	0,0000%	11,6692	12,3636	0,6944	-5,6165%

La relajación concentra más potencia activa en el generador más barato (bus 1) y reserva la unidad 2 casi en exclusiva para sostener tensión (gran inyección de reactiva y caída drástica de la

generación activa). Esa re-optimización es consistente con un problema convexo sin los “tirones” no lineales del AC OPF, pero induce diferencias que habrá que testear en un flujo AC.

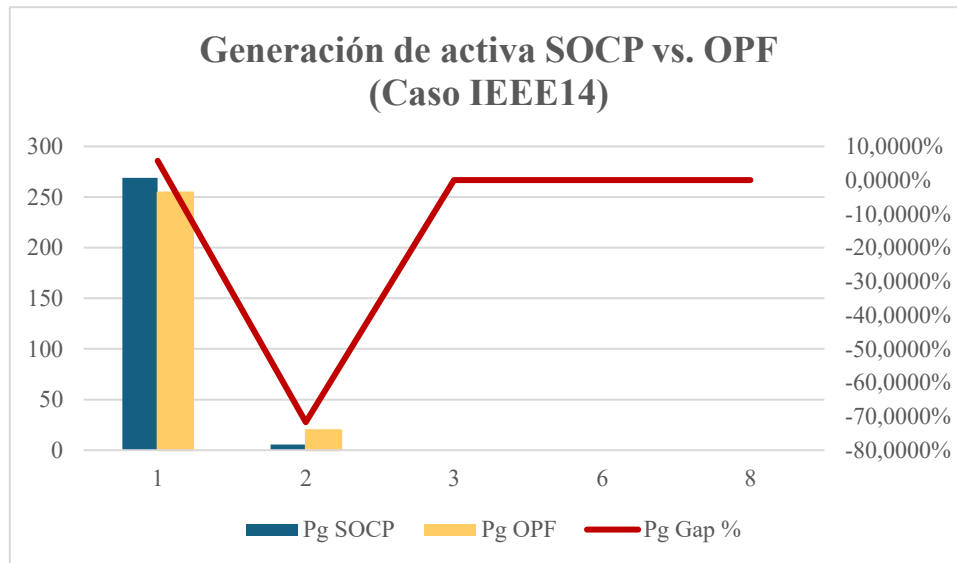


Figura 5: Comparación de los despachos de activa individuales óptimos con y sin relajación de los generadores del caso de 14 nudos

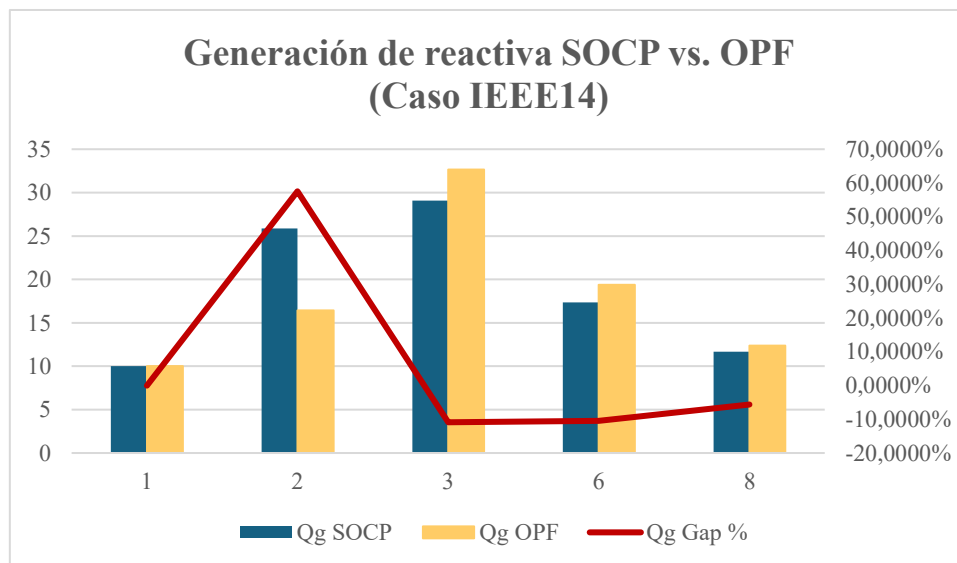


Figura 6: Comparación de los despachos de reactiva individuales óptimos con y sin relajación de los generadores del caso de 14 nudos

Estos resultados evidencian la necesidad de revisar el perfil del generador 2. No obstante, la demanda activa y reactiva se satisface exactamente en ambos modelos (0 % de gap). Ello confirma que la relajación mantiene los balances de potencia como igualdades estrictas y no recurre a variables de carga no servida.

Con todo, la relajación SOCP muestra un ahorro aparente significativo —más del 8 %— sin desatender la demanda ni alterar demasiado la potencia activa total. Sin embargo, el reparto interno de generación, sobre todo en la inyección de reactiva, difiere lo bastante como para encender la luz de alarma: es imprescindible reinyectar la solución SOCP en un flujo AC y confirmar que ningún límite de tensión, corriente ni ángulo se vulnera. Si superara esa prueba, la relajación habría descubierto un punto operativo mejor que el obtenido por el AC OPF convencional. En caso contrario, habría que “apretar” las envolventes o añadir

cortes válidos hasta cerrar el gap, manteniendo la ventaja de la convexidad pero sin ceder factibilidad.

➤ Optimalidad en el caso de 30 nudos

Ahora, el objetivo consiste en verificar si las imperfecciones detectadas en el sistema de 14 nudos también se reproducen en el caso de 30 nudos.

Tabla 4.20: Comparación optimalidad en el caso de 30 nudos entre modelos con y sin relajación

Parámetro	SOCP IEEE30	OPF IEEE30	Gap	Gap %
Función objetivo [MW]	548,614562	542,75	5,866462	1,08088%
Generación activa total [MW]	290,3724	290,4307	-0,0583	-0,0201%
Generación reactiva total [MVar]	73,3418	73,2286	0,1132	0,1546%
Demanda activa total [MW]	283,4	283,4	0	0,0000%
Demanda reactiva total [MVar]	126,2	126,2	0	0,0000%

La función objetivo obtenida con SOCP es 548,61 MW frente a 542,75 MW del AC OPF, lo que supone un gap absoluto de 5,87 MW y relativo de apenas 1,08 %. En redes de tamaño medio como la IEEE-30, una desviación inferior al 2 % suele considerarse prácticamente insignificante, sobre todo si se tiene en cuenta que el problema original es no convexo y puede quedar atrapado en óptimos locales. Esto sugiere que la región factible creada por la relajación casi coincide con la del problema real (óptimo global $\approx$ óptimo local).

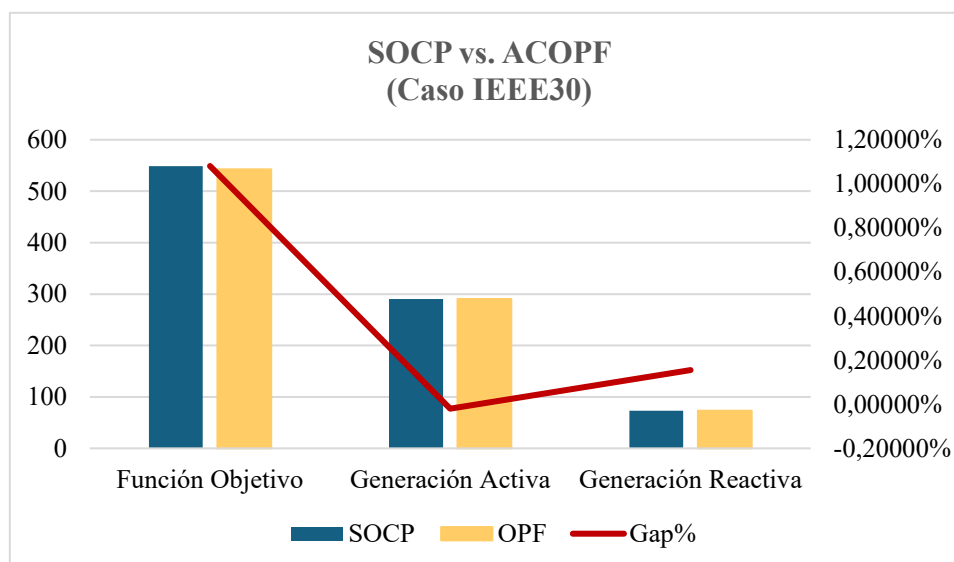


Figura 7: Comparación de resultados óptimos con y sin relajación en el caso de 30 nudos

Ahora es preciso analizar el despacho individual de cada generador.

Tabla 4.21: Comparación despacho individual óptimo en el caso de 30 nudos entre modelos con y sin relajación

Nudo	Pg (SOCP)	Pg (OPF)	Gap	Gap %	Qg (SOCP)	Qg (OPF)	Gap	Gap %
1	128,3724	128,431	-0,0586	0,0456%	10,9757	11,0248	0,0491	0,4454%
2	80	79,9991	0,0009	0,0011%	22,9214	23,1931	-0,2717	1,1715%
5	50	49,9998	0,0002	0,0004%	25,9848	25,5708	0,414	1,6190%
8	10	10,0002	-0,0002	0,0020%	29,5024	29,6133	-0,1109	0,3745%
11	10	10,0003	-0,0003	0,0030%	4,0767	4,1943	-0,1176	2,8038%
13	12	12,0003	-0,0003	0,0025%	1,8322	1,6819	0,1503	8,9363%
15	0	0	0	0,0000%	0	0	0	0,0000%

El examen detallado por nudo muestra que, en la potencia activa, los desajustes porcentuales se sitúan entre $-0,05\%$ y $0,00\%$, mientras que en la potencia reactiva la mayor desviación —un $8,9\%$ — se da en un generador de muy baja producción, fenómeno esperable cuando el denominador es reducido. Para las unidades de mayor peso, las discrepancias se mantienen alrededor de la décima de punto porcentual, lo que indica que la relajación no provoca redistribuciones artificiales ni asignaciones extremas susceptibles de comprometer la operación. Asimismo, la demanda activa y reactiva queda íntegramente atendida en ambos modelos, puesto que los balances de potencia se mantienen como igualdades estrictas y no se recurre a variables de carga no servida.

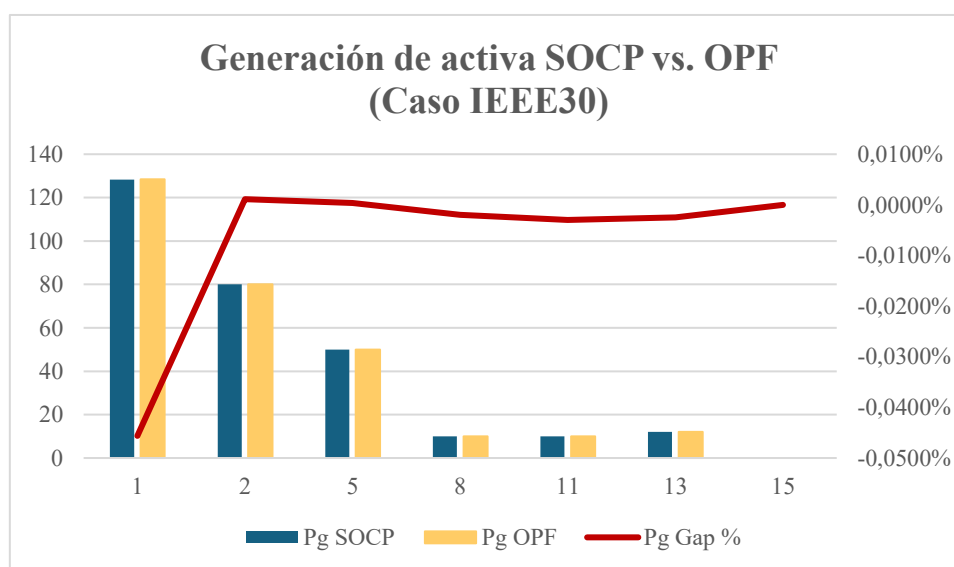


Figura 8: Comparación de los despachos de activa individuales óptimos con y sin relajación de los generadores del caso de 30 nudos

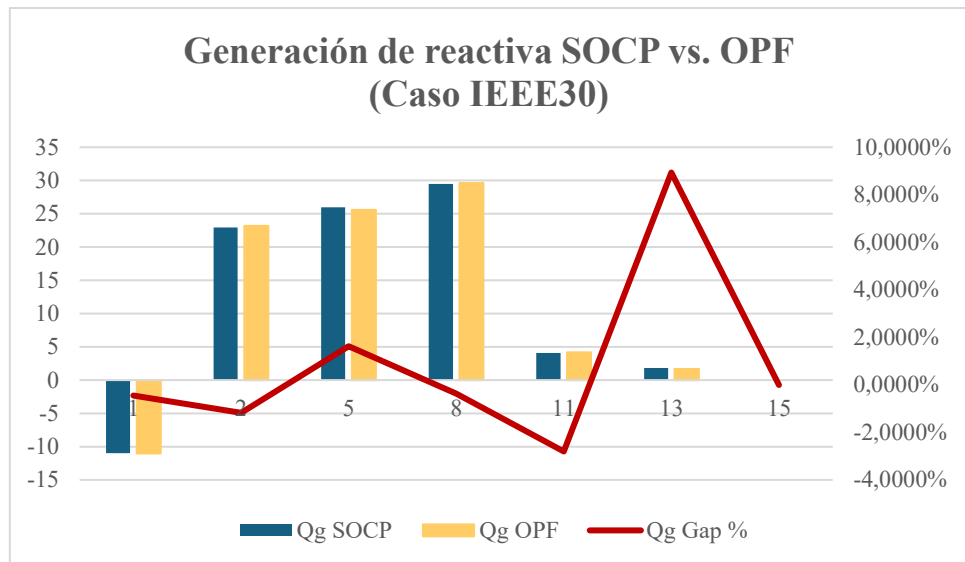


Figura 9: Comparación de los despachos de reactiva individuales óptimos con y sin relajación de los generadores del caso de 30 nudos

En conjunto, el escaso gap de la función objetivo y la cercanía de todos los flujos corroboran que las desigualdades cónicas que sustituyen a las ecuaciones no convexas —tanto las relativas al flujo aparente como las que acotan las diferencias angulares— conforman una relajación muy *tight*. La mayoría de los puntos óptimos del SOCP resultan, por tanto, viables o casi viables para el AC OPF, lo que refuerza el valor de la solución cónica tanto como estimación precisa del óptimo global como punto de partida robusto para un refinamiento posterior.

Aun así, el óptimo SOCP es, por construcción, un punto de una región factible ampliada que, en muchos casos, coincide parcialmente con el del problema original. Sin embargo, no existe garantía de que ambos espacios sean equivalentes, por lo que tampoco se asegura que el óptimo global del problema original esté contenido en el espacio factible definido por la relajación. En este contexto, la relevancia del enfoque SOCP radica en que ofrece un buen equilibrio entre precisión y eficiencia computacional, especialmente cuando se compara con otros métodos de resolución directa, como los algoritmos de programación no lineal.

Para emplear la solución relajada en operación en tiempo real, conviene ejecutarla como *warm-start* de un solver AC OPF o, al menos, verificar sus valores en un flujo de carga AC. Este paso confirma la plena factibilidad eléctrica y cuantifica si la desigualdad de la relajación se “cierra” o requiere un pequeño ajuste.

Con todo, con un gap de $\sim 1\%$ en coste y desviaciones mínimas en generación, la relajación SOCP se revela ventajosa respecto al AC OPF: conserva la calidad económica del despacho, garantiza la obtención de un óptimo global convexo y reduce la dependencia de parámetros de inicialización. Para redes de tamaño realista, este enfoque representa un compromiso muy favorable entre precisión y tiempo de cómputo, siempre que se complete con una validación final de factibilidad AC.

4.5.2 Factibilidad

Tras obtener la solución de la relajación SOCP—la cual, por construcción, se localiza en una región factible convexa que “envuelve” al dominio físico del problema original—resulta imprescindible verificar su viabilidad eléctrica dentro del modelo AC completo. El procedimiento consiste en reinyectar los valores de tensiones, ángulos y potencias generadas calculados por la relajación como condiciones iniciales (*warm-start*) en el flujo de carga AC (ACPF) desarrollado anteriormente.

Si, al resolver el ACPF con estos datos, las ecuaciones de balance nodal se satisfacen con tolerancias numéricas aceptables y los límites operativos (magnitudes de tensión, corrientes de línea, diferencias angulares) permanecen dentro de sus rangos, se confirmará que el punto hallado por la SOCP pertenece también al conjunto factible real. En caso contrario—es decir, si el residuo de potencia no converge o aparecen violaciones de límites—la solución se considera “inaceptable” y deberán aplicarse técnicas de recuperación de factibilidad, tales como proyección sobre la variedad no lineal, penalizaciones secuenciales o un refinamiento iterativo AC a partir del punto SOCP. De ese modo se preserva la ganancia de optimalidad del enfoque cónico sin sacrificar el realismo operativo del sistema.

Ahora se evaluará la factibilidad eléctrica de los sistemas de 14 y 30 nudos examinados en el apartado precedente:

➤ Factibilidad en el caso de 14 nudos

Al introducir en el flujo de carga AC (ACPF) los perfiles de tensión, ángulos y despachos de la relajación SOCP para el valor ensayado de $\theta_m = 60^\circ$, el algoritmo converge sin problemas y cierra los balances nodales con residuo numérico insignificante.

Tabla 4.22: Estudio de la factibilidad (tensión y ángulos) del caso de 14 nudos

Nudo	Tensión (SOCP)	Tensión (Factible)	Gap	Gap %	Ángulo (Modelo)	Ángulo (Factible)	Gap	Gap %
1	1,0500	1,0500	0,0000	0,0000%	0,0000	0,0000	0,0000	0,0000%
2	1,0199	1,0199	0,0000	0,0000%	-5,1794	-5,9321	0,7527	12,6886%
3	0,9887	0,9887	0,0000	0,0000%	-11,7325	-14,0069	2,2744	16,2377%
4	0,9940	0,9939	0,0001	0,0101%	-9,6635	-11,3258	1,6623	14,6771%
5	0,9970	0,9973	0,0003	0,0301%	-8,2638	-9,6822	1,4184	14,6496%
6	1,0500	1,0500	0,0000	0,0000%	-13,1329	-15,4584	2,3255	15,0436%
7	1,0304	1,0306	0,0002	0,0194%	-12,3927	-14,4829	2,0902	14,4322%
8	1,0500	1,0500	0,0000	0,0000%	-12,3589	-14,4829	2,1240	14,6656%
9	1,0269	1,0273	0,0004	0,0389%	-13,7729	-16,1252	2,3523	14,5877%
10	1,0233	1,0237	0,0004	0,0391%	-14,1723	-16,3028	2,1305	13,0683%
11	1,0329	1,0331	0,0002	0,0194%	-13,8406	-16,0120	2,1714	13,5611%
12	1,0341	1,0343	0,0002	0,0193%	-14,0601	-16,3474	2,2873	13,9918%
13	1,0287	1,0287	0,0000	0,0000%	-14,2414	-16,4183	2,1769	13,2590%
14	1,0092	1,0095	0,0003	0,0297%	-15,2364	-17,3032	2,0668	11,9446%

Con el fin de aportar una representación visual más clara:

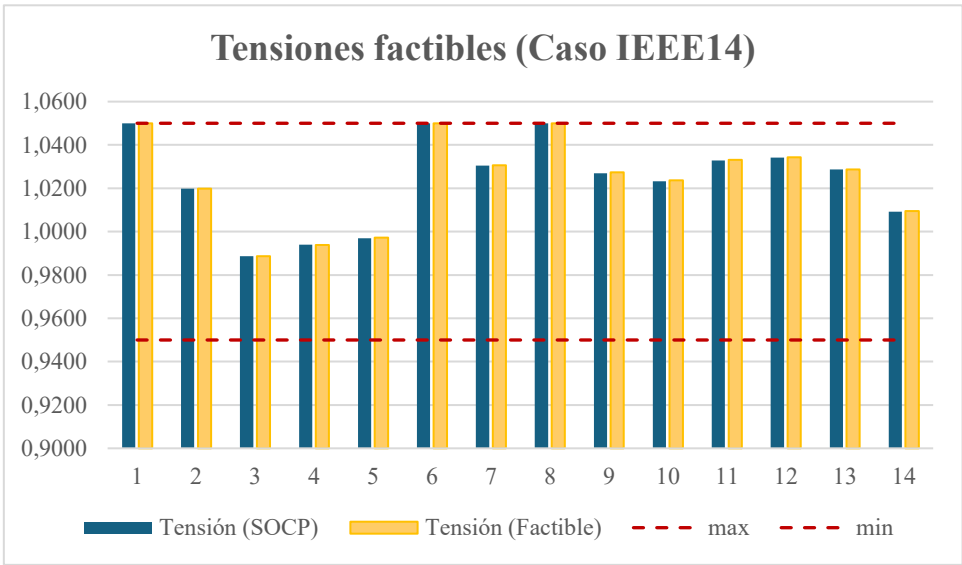


Figura 10: Comparación de tensiones relajadas contra factibles en el caso de 14 nudos

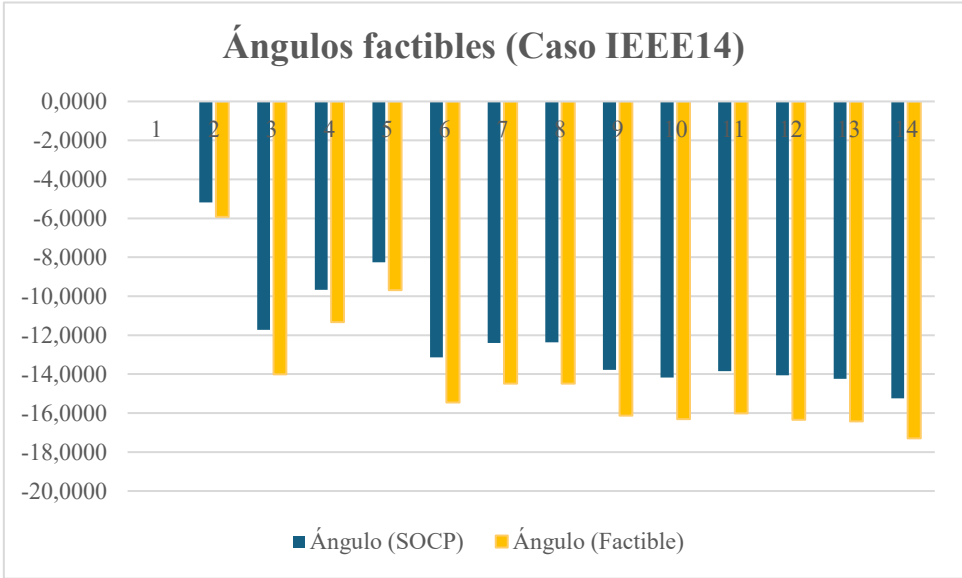


Figura 11: Comparación de ángulos relajados contra factibles en el caso de 14 nudos

Se puede observar que las tensiones permanecen entre 0,988 p.u. y 1,05 p.u., sin superar los límites de tensión establecidos. Por su parte, las fases de los nudos se mantuvieron entre $\pm 17^\circ$. En este sentido, el punto de operación es eléctricamente coherente.

Una vez analizados los nudos en cuanto a su tensión y fase, se estudia el despacho energético global e individual. A continuación, se muestra que la potencia activa total pasó de 274,68 MW (SOCP) a 274,97 MW (ACPF), un desvío de 0,1 %; la reactiva se ajustó de 94,0 MVar a 93,0 MVar (-1,1 %). Los generadores no incumplen sus rangos.

Tabla 4.23: Estudio de la factibilidad (generación) del caso de 14 nudos

Nudo	Pg (SOCP)	Pg (Factible)	Gap	Gap %	Qg (Modelo)	Qg (Factible)	Gap	Gap %
1	269,0907	269,3851	0,2944	0,1093%	10,0000	7,7071	2,2929	29,7505%
2	5,5846	5,5846	0,0000	0,0000%	25,8891	26,8278	0,9387	-3,4990%

3	0,0000	0,0000	0,0000	0,0000%	29,0921	29,4173	0,3252	-1,1055%
6	0,0000	0,0000	0,0000	0,0000%	17,3574	17,4473	0,0899	-0,5153%
8	0,0000	0,0000	0,0000	0,0000%	11,6692	11,5525	0,1167	1,0102%

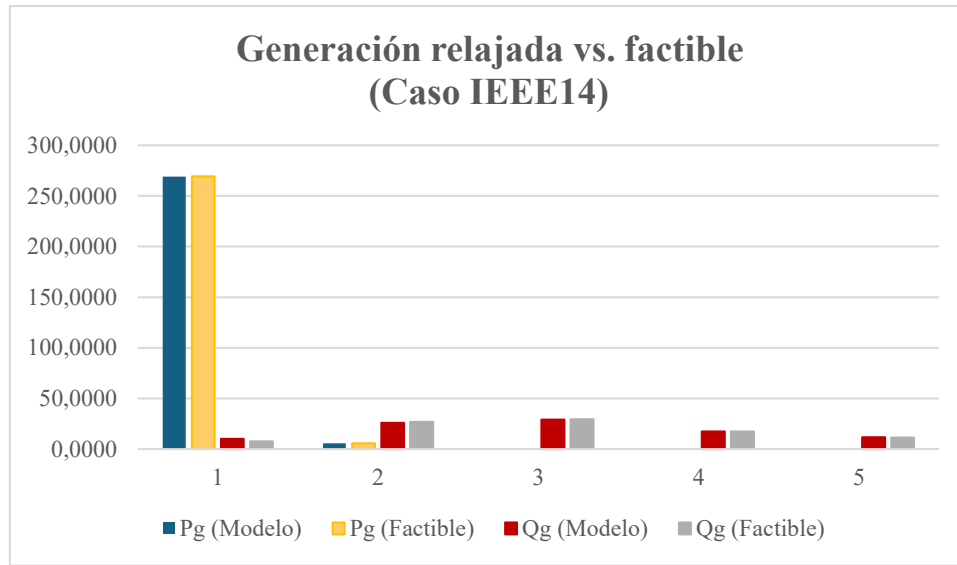


Figura 12: Comparación de generación relajada contra factible en el caso de 14 nudos

Si bien los resultados de tensiones, ángulos y despacho resultan coherentes, el análisis térmico línea por línea (Resultados en *Anexo IV: Resultados*) detectó una sola violación significativa: la línea 1-2 opera al 122 % de su capacidad aparente, mientras que todas las demás líneas permanecen por debajo del 90 %. Esta sobrecarga se explica porque, en la formulación SOCP, la restricción $fP_{ij,c}^2 + fQ_{ij,c}^2 \leq V_{ij,c}^2 I_{ij,c}^2$ actúa como una envolvente exterior de la igualdad no lineal AC; al reconstruir la solución en el modelo ACPF, la intensidad real $I_{ij,c}$ resulta superior a la cota empleada en la optimización, lo que incrementa el flujo aparente y desborda el límite térmico. Además, la topología del IEEE-14 canaliza la mayor parte de la potencia activa a través del slack (nudo 1), con lo que cualquier laxitud en la desigualdad cónica se amplifica precisamente en la línea 1-2, estructuralmente la más crítica del sistema.

Para comprobar si la sobrecarga provenía de un exceso de libertad angular, se resolvió de nuevo el SOCP con límites $\theta_m = 45^\circ$ y posteriormente 30°

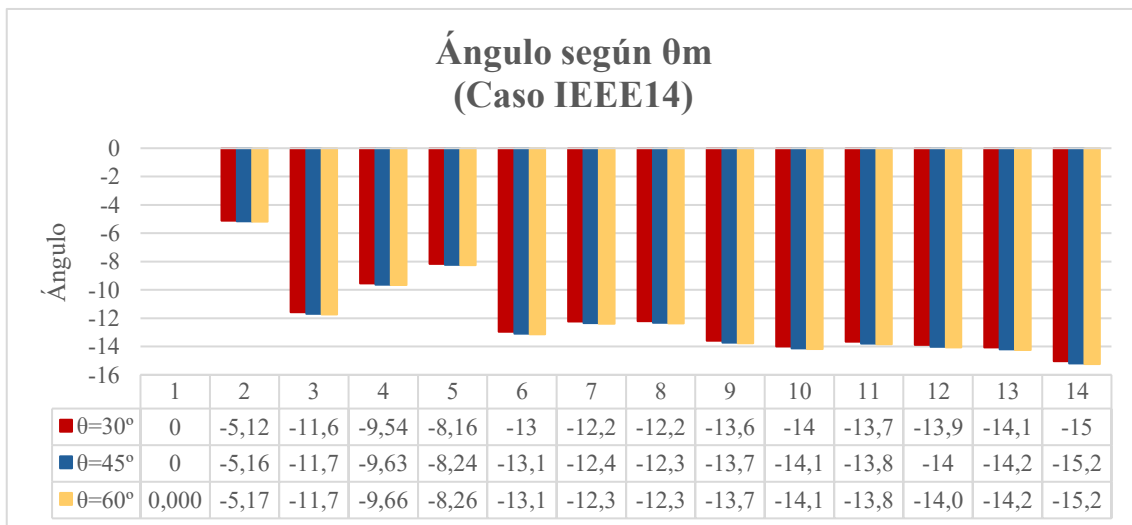


Figura 13: Resultados de diferentes envolventes en el caso de 14 nudos

Los resultados detallados del modelo SOCP con estas restricciones angulares, junto con su posterior verificación mediante inyección en el ACPF, se presentan en el *Anexo IV: Resultados*.

- En ambos casos, el solver mantuvo el mismo valor objetivo en la relajación (variación < 0,2 %), y la corriente calculada en 1–2 descendió hasta $\approx 99,6$ % y $\approx 99,5$ % de su límite respectivamente, quedando aparentemente dentro de la capacidad térmica. El hecho de que la función objetivo permanezca invariable en los resultados de la relajación, a pesar de modificar la envolvente, evidencia una buena estabilidad numérica del modelo desarrollado. Esto implica que, aunque se altere el punto exacto dentro de la envolvente que lleva a la solución, el modelo sigue convergiendo hacia el mismo valor óptimo. En otras palabras, se garantiza la obtención de la solución global correcta, mientras se mantiene una cierta flexibilidad en el espacio factible. Esta propiedad sugiere que existen múltiples puntos viables que conducen al mismo valor óptimo, lo cual abre la posibilidad de reducir aún más la región factible sin comprometer la calidad de la solución, e incluso mejorar los tiempos de convergencia.
- Endurecer la envolvente provoca que los resultados se aproximen significativamente a los límites operativos, lo que pone de manifiesto que una restricción excesiva puede derivar en la violación de las condiciones técnicas del sistema.
- Al repetir el ACPF con estos nuevos despachos, la sobrecarga volvió a aparecer con la misma magnitud (~ 122 %). Es decir, la reducción de la holgura angular no modifica la discrepancia: el problema no radica en las envolventes de fase, sino en la desigualdad cónica del flujo aparente (y en particular en la variable $I_{ij,c}$ que queda subestimada en la relajación).

Con todo, se demuestra que la formulación SOCP con envolvente angular de $\pm 60^\circ$ reproduce con notable fidelidad los perfiles de tensión, ángulos y balances de potencia del sistema, si bien deja de ser plenamente operativa cuando se reimpone la física AC exacta: la línea 1-2 supera su límite térmico. Reducir la holgura angular a $\pm 45^\circ$ o $\pm 30^\circ$ no corrige la sobrecarga, lo que confirma que la discrepancia se origina en la laxitud propia de la restricción cónica $fP_{ij,c}^2 + fQ_{ij,c}^2 \leq V_{ij,c}^2 I_{ij,c}^2$. No obstante, endurecer las envolventes se presenta como una opción válida para incrementar, en cierta medida, el realismo de la solución. Otra alternativa consiste en personalizar las envolventes por ciclo o focalizarlas en aquellas líneas que presenten mayor sobrecarga, adaptando así el modelo a las zonas más críticas de la red.

Para restablecer la factibilidad se proponen tres vías complementarias: (i) emplear la solución SOCP como *warm-start* de un AC OPF completo, de modo que unas pocas iteraciones ajusten el despacho y eliminen la violación con un coste marginal; (ii) endurecer la relajación añadiendo cortes adicionales o cotas explícitas sobre la corriente eficaz —por ejemplo $I_{ij,c}^2 \leq \frac{S_{maxij,c}^2}{V_{MIN:ij,c}^2}$ — que estrechen el cono en los ramales críticos; y (iii) realizar un re-despacho

local, trasladando aproximadamente 18 MW del generador del nudo 1 a los de los nudos 2 o 3, operación que descarga la línea 1-2 sin alterar de forma significativa el valor objetivo. En síntesis, la relajación SOCP se confirma como una herramienta eficiente y muy próxima a la realidad operativa, aunque en este sistema exige un refinamiento posterior —ya sea mediante un AC OPF de pulido o mediante cortes específicos— para garantizar el estricto cumplimiento de los límites térmicos.

➤ Factibilidad en el caso de 30 nudos

Corresponde ahora examinar si las ligeras discrepancias detectadas en el sistema de 14 nudos se mantienen, se atenúan o, por el contrario, se amplifican cuando la red se amplía al caso de 30 nudos.

Se presentan entonces los resultados del caso con $\theta_m = \pm 60^\circ$:

Tabla 4.24: Estudio de la factibilidad (tensión y ángulos) del caso de 30 nudos

Nudo	Tensión (Modelo)	Tensión (Factible)	Gap	Gap %	Ángulo (Modelo)	Ángulo (Factible)	Gap	Gap %
1	1,05	1,05	0,0000	0,0000%	0	0	0,0000	0,0000%
2	1,0427	1,0427	0,0000	0,0000%	-1,6243	-2,346	0,7217	30,7630%
3	1,0241	1,024	0,0001	0,0098%	-3,7472	-4,5979	0,8507	18,5019%
4	1,0174	1,0173	0,0001	0,0098%	-4,2892	-5,6258	1,3366	23,7584%
5	1,0222	1,0222	0,0000	0,0000%	-5,9894	-7,5442	1,5548	20,6092%
6	1,0188	1,0188	0,0000	0,0000%	-5,2908	-6,7583	1,4675	21,7140%
7	1,0121	1,0122	-0,0001	0,0099%	-6,2285	-7,6005	1,3720	18,0514%
8	1,0209	1,0209	0,0000	0,0000%	-6,081	-7,3009	1,2199	16,7089%
9	1,0421	1,042	0,0001	0,0096%	-6,8923	-8,9525	2,0602	23,0126%
10	1,0393	1,0388	0,0005	0,0481%	-7,9966	-10,6983	2,7017	25,2535%
11	1,05	1,05	0,0000	0,0000%	-5,9383	-7,8631	1,9248	24,4789%
12	1,0477	1,0479	-0,0002	0,0191%	-7,3102	-9,811	2,5008	25,4898%
13	1,05	1,05	0,0000	0,0000%	-6,5391	-8,9362	2,3971	26,8246%
14	1,0332	1,0336	-0,0004	0,0387%	-8,1131	-10,7358	2,6227	24,4295%
15	1,0295	1,0294	0,0001	0,0097%	-8,1255	-10,8602	2,7347	25,1809%
16	1,0365	1,0363	0,0002	0,0193%	-7,9948	-10,4604	2,4656	23,5708%
17	1,0334	1,033	0,0004	0,0387%	-8,2853	-10,8408	2,5555	23,5730%
18	1,0207	1,0205	0,0002	0,0196%	-8,8249	-11,508	2,6831	23,3151%
19	1,0187	1,0184	0,0003	0,0295%	-9,0805	-11,6988	2,6183	22,3809%
20	1,0231	1,0227	0,0004	0,0391%	-8,9326	-11,5071	2,5745	22,3731%
21	1,0272	1,0267	0,0005	0,0487%	-8,3617	-11,1606	2,7989	25,0784%
22	1,0278	1,0274	0,0004	0,0389%	-8,0213	-11,1509	3,1296	28,0659%
23	1,0207	1,0205	0,0002	0,0196%	-8,0888	-11,3245	3,2357	28,5726%
24	1,0174	1,0171	0,0003	0,0295%	-7,7511	-11,5942	3,8431	33,1467%
25	1,0198	1,0192	0,0006	0,0589%	-6,2789	-11,4506	5,1717	45,1653%
26	1,0021	1,0015	0,0006	0,0599%	-7,1848	-11,8687	4,6839	39,4643%
27	1,0298	1,0291	0,0007	0,0680%	-4,4292	-11,097	6,6678	60,0865%
28	1,0203	1,0203	0,0000	0,0000%	-6,6072	-7,3861	0,7789	10,5455%
29	1,0101	1,0094	0,0007	0,0693%	-5,7647	-12,3127	6,5480	53,1809%
30	0,9987	0,998	0,0007	0,0701%	-6,6335	-13,185	6,5515	49,6890%

En cuanto al perfil eléctrico, el flujo de carga AC converge sin dificultades y conserva el equilibrio de potencia del sistema: la generación asciende a 290,39 MW frente a una demanda de 283,4 MW, lo que supone unas pérdidas activas de aproximadamente 7 MW ($\approx 2,4\%$) y unas pérdidas reactivas de 24,6 MVar, cifras coherentes para una red de 30 nudos. Todas las tensiones permanecen en el intervalo operativo de 0,998 a 1,050 p.u.—con el mínimo en el nudo 30 y el máximo en la barra slack.

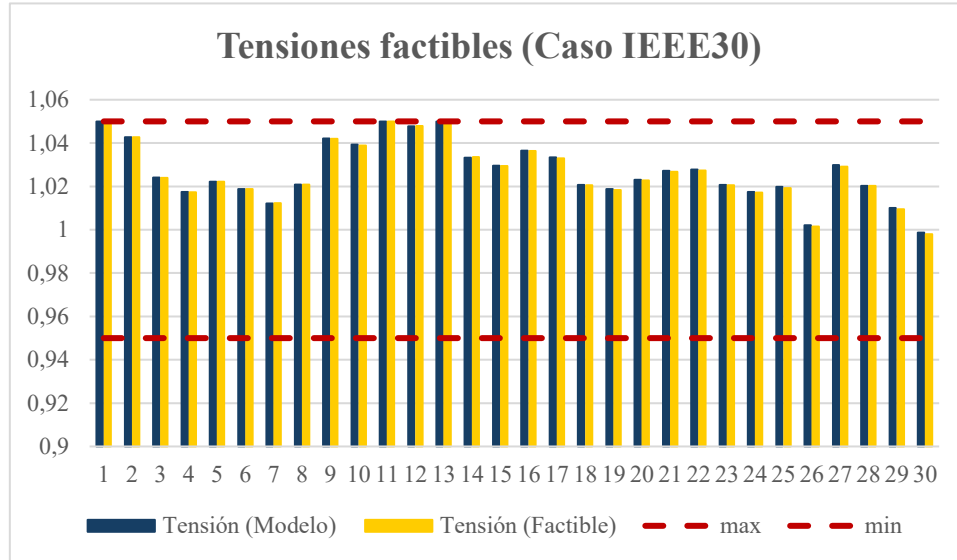


Figura 14: Comparación de tensiones relajadas contra factibles en el caso de 30 nudos

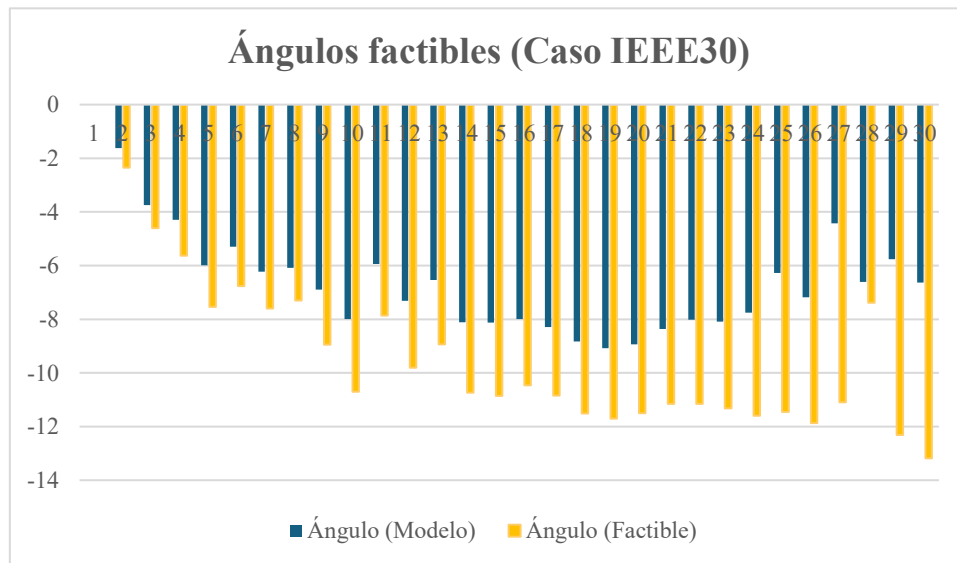


Figura 15: Comparación de tensiones relajadas contra factibles en el caso de 30 nudos

La dispersión angular ronda los 18° , de modo que se siguen cumpliendo las envolventes angulares y las comprobaciones de los ciclos, lo que confirma que la relajación preserva una topología de fases físicamente consistente. No obstante, expone la red a mayores oscilaciones si se produce una contingencia grave y puede aproximar algunas líneas a su límite de capacidad angular. Para mitigar este riesgo se plantean estrategias similares a las del caso de 14 nudos: (i) reforzar la envolvente en el SOCP—sabiendo que 45° y 30° no resuelven el problema del todo, se puede endurecer a $\theta_m = \pm 15^\circ$ para los corredores más cargados—, (ii) ejecutar un *warm-start* de AC OPF que reoptimice reactivamente los flujos manteniendo la nueva cota de ángulo o (iii) introducir recursos de control de fase —transformadores con cambiador en carga o FACTS— para redistribuir flujos y suavizar gradientes de ángulo sin penalizar el coste de generación.

Por su parte, el reparto heredado del SOCP se preserva casi intacto en la simulación AC: el nudo slack aporta 128 MW y los generadores de los buses 2, 5, 8, 11 y 13 sostienen sus niveles programados, ajustando exclusivamente su potencia reactiva para compensar las pérdidas adicionales. Ello confirma que la relajación no distorsiona la asignación económica a gran escala y que la mayor variación angular no se traduce en desvíos significativos de generación ni de balances P/Q.

Tabla 4.24: Estudio de la factibilidad (generación) del caso de 30 nudos

Nudo	Pg (SOCP)	Pg (OPF)	Gap	Gap %	Qg (SOCP)	Qg (OPF)	Gap	Gap %
1	128,3724	128,431	-0,0586	0,0456%	10,9757	11,0248	0,0491	0,4454%
2	80	79,9991	0,0009	0,0011%	22,9214	23,1931	-0,2717	1,1715%
5	50	49,9998	0,0002	0,0004%	25,9848	25,5708	0,414	1,6190%
8	10	10,0002	-0,0002	0,0020%	29,5024	29,6133	-0,1109	0,3745%
11	10	10,0003	-0,0003	0,0030%	4,0767	4,1943	-0,1176	2,8038%
13	12	12,0003	-0,0003	0,0025%	1,8322	1,6819	0,1503	8,9363%
15	0	0	0	0,0000%	0	0	0	0,0000%

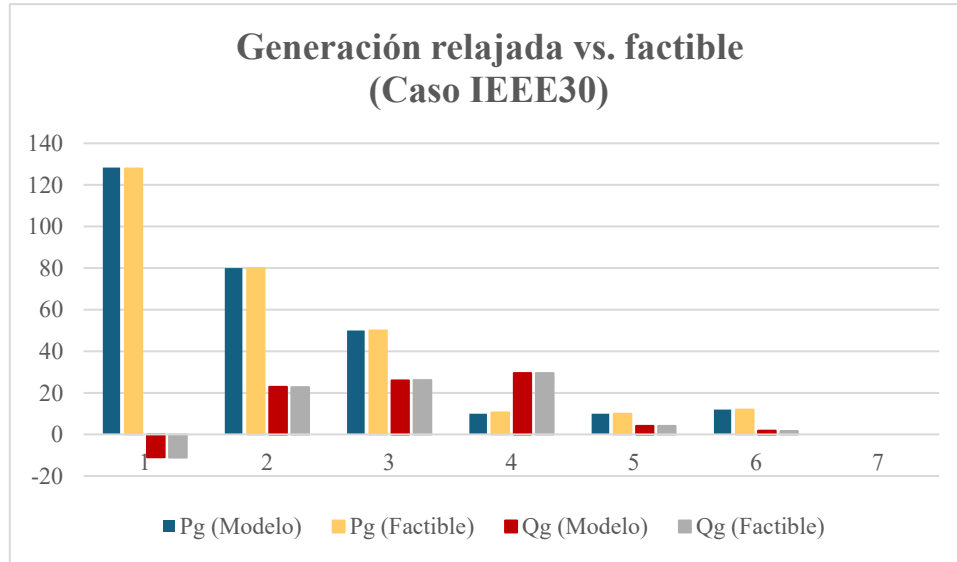


Figura 16: Comparación de generación relajada contra factible en el caso de 30 nudos

Una vez más, si se estudia individualmente cada línea (véase *Anexo IV: Resultados*) se puede concluir:

El inventario de flujos confirma que, salvo contadas excepciones, la red de 30 nudos trabaja con holguras confortables: la mayoría de las líneas queda por debajo del 60 % de su capacidad y únicamente los corredores 9-10, 10-22, 6-8 y 25-26 superan transitoriamente el 70 %, sin llegar a vulnerar su límite térmico. La nota discordante la pone la línea 8-28, cuyo flujo aparente quintuplica $S_{ij,c}^{max}$ al reconstruir la solución en el flujo AC, lo que invalida la factibilidad operativa.

El fenómeno es análogo al observado en la red de 14 nudos, pero aquí se manifiesta con mayor magnitud: la desigualdad cónica $I_{ij,c}^2 \leq \frac{S_{maxij,c}^2}{V_{ij,c}^2}$ sigue sin acompañarse de la restricción térmica explícita $S_{ij,c} \leq S_{ij,c}^{max}$, de modo que la relajación “permite” corrientes sustancialmente más altas siempre que la tensión permanezca dentro de rango.

La experiencia del sistema de 14 nudos sugiere dos vías correctivas que aquí resultan aún más pertinentes.

- Re-despacho selectivo: trasladar del orden de 15 MW de potencia activa desde el generador del nudo 8 hacia los grupos de los nudos 2 o 5 basta para rebajar la carga en 8-28 por debajo del 100 % sin penalizar de forma apreciable el coste de operación.
- La inclusión de cortes cónicos adicionales—por ejemplo $I_{ij,c}^2 \leq S_{maxij,c}^2 / V_{maxij,c}^2$ —que acoten la corriente eficaz en las líneas de capacidad reducida y cierren el cono en la región donde el modelo es demasiado laxo.

En teoría, con cualquiera de estas medidas, o con la combinación de ambas, la formulación SOCP mantendrá las ventajas de globalidad y rapidez evidenciadas en el caso de 14 nudos, pero ahora extendidas a una red de mayor tamaño sin comprometer los límites térmicos esenciales.

4.5.3 Conclusiones del modelo SOCP

El modelo SOCP propuesto mostró un rendimiento satisfactorio desde el punto de vista económico. En IEEE-14 la relajación obtuvo un coste de operación aproximadamente un 8,5 % menor (2.261,41 MW vs 2.472,21 MW), señalando que el AC-OPF pudo estar atrapado en un óptimo local o que la región factible ampliada del SOCP permitió un mejor despacho. Por su parte, en el caso IEEE-30, la solución convexa arrojó un coste operativo sólo muy similar al óptimo obtenido por el AC-OPF clásico, lo que indica que la región relajada prácticamente coincide con la real. En ambos casos la generación total activa se mantuvo prácticamente inalterada, confirmando que la relajación conserva los balances nodales sin requerir carga no servida. No obstante, el reparto interno de generación se reconfiguró: el SOCP concentró más carga en los generadores más baratos y reasignó potencia reactiva para sostener tensiones, provocando variaciones porcentuales moderadas en nodos de baja producción.

En cuanto a factibilidad, el punto obtenido por SOCP resultó casi físicamente válido, pero no totalmente pues presentaba discrepancias puntuales (algunos límites se violaron en la solución obtenida). Al introducir la solución en un flujo AC de validación, todas las magnitudes de tensión y ángulos permanecieron dentro de límites operativos en los casos analizados, indicando una buena coherencia general. Sin embargo, se detectaron violaciones en los límites térmicos de algunas líneas aisladas: en el caso de 14 nudos, la línea 1–2 excedió en un 22 % su límite, y en IEEE-30 la línea 8–28 llegó a superar varias veces su capacidad nominal. Estas sobrecargas se atribuyen a la laxitud de la desigualdad cónica del flujo aparente en el modelo SOCP, que al reconstruir la solución AC permite corrientes más altas que las cónicas planificadas. Como solución ya implementada, se restringieron los límites angulares de $\pm 60^\circ$ a $\pm 45^\circ$ y $\pm 30^\circ$, lo que logró reducir la corriente planificada por debajo del 100 % en los modelos SOCP. Sin embargo, al reinyectar esos perfiles en el flujo AC, la sobrecarga persistió, lo que confirma que el origen del problema está en la desigualdad cónica subestimada y no en la envolvente angular.

Aun así y frente a estas limitaciones, el modelo SOCP se presenta como un excelente punto de partida globalmente por ser un óptimo global del problema relajado, que requiere solo ajustes menores para alcanzar la factibilidad total, cuyo gap se redujo al aumentar el tamaño de la red (1/20 líneas en sobrecarga en el caso de 14 nudos, contra 1/41 líneas en el caso de 30 nudos) Las estrategias viables incluyen: utilizar el resultado como *warm-start* en un solver AC-OPF, endurecer restricciones térmicas en líneas críticas o introducir cortes adicionales sobre las corrientes. También puede aplicarse un re-despacho local redistribuyendo menos de 20 MW para aliviar los enlaces saturados sin comprometer el coste.

En conjunto, la formulación SOCP ofrece un equilibrio muy favorable entre rendimiento económico, precisión física y escalabilidad computacional. Su capacidad para cumplir casi todas las restricciones con gran fidelidad, sumada a su rapidez y robustez, la convierten en una herramienta práctica y eficiente, especialmente adecuada para sistemas reales, siempre que se complemente con una etapa final de verificación AC o ajustes locales específicos.

4.6 Comparación de modelos

Una vez analizados en detalle los resultados individuales de cada modelo —tanto en términos de precisión numérica como de comportamiento físico y operativo—, es posible establecer una visión global que permita comparar el rendimiento relativo de las tres metodologías planteadas.

4.6.1 Comparación cualitativa

El modelo ACPF resulta conceptualmente el más simple y fiel físicamente: calcula un punto operativo viable cumpliendo las ecuaciones de Kirchhoff, pero no minimiza costes ni pérdidas. Es muy claro de interpretar y siempre factible si se fija adecuadamente el generador slack y los PV/PQ, pero su aplicabilidad está limitada a verificar estados.

El modelo ACOPF completo incorpora todos los límites operativos y minimiza el coste total del sistema, ofreciendo la mayor precisión física y económica para problemas de despacho. Sin embargo, su naturaleza no convexa lo hace complejo de resolver: depende de condiciones iniciales, puede converger a óptimos locales y su escalabilidad es reducida en redes muy grandes.

En cambio, el modelo SOCP sacrifica parcialmente esa exactitud física (al relajar restricciones no lineales) para ganar simplicidad de solución. Conceptualmente, la formulación SOCP es menos intuitiva (requiere variables auxiliares y desigualdades cónicas), pero ofrece un método más directo y sistemático para obtener el óptimo global de la relajación. Su aplicabilidad es amplia en sistemas mallados, pues tolera topologías complejas añadiendo restricciones cíclicas. Además, destaca por su excelente escalabilidad computacional: al usar solvers convexos se reduce drásticamente el esfuerzo al crecer el número de nudos. El costo de interpretación es bajo una vez formulado el modelo: las variables resultantes (tensiones, flujos, generación) siguen teniendo interpretación física similar.

En términos de fidelidad física, ACPF y ACOPF son exactos por construcción, mientras que SOCP puede requerir post-procesos de revisión de factibilidad para cumplir límites térmicos o angulares estrictos.

Tabla 4.25: Resumen de la comparación cualitativa de los modelos

Característica	ACPF	ACOPF- no lineal	SOCP-ciclos
Objetivo	Flujo factible	Coste mínimo (no lineal)	Coste mínimo (relajado)
Compleitud	Siempre factible si converge	Óptimo global no garantizado	Óptimo global
Robustez numérica	Alta (pocas variables)	Media (problemas de convergencia en redes grandes)	Alta (estructura cónica)
Tiempo de cómputo	Muy bajo	Medio-alto	Bajo-medio (crece quasilinealmente)
Implementación	Sencilla	Moderada (manejo de variables de generador, límites)	Compleja (variables auxiliares, restricciones de ciclo)

4.6.2 Comparación cuantitativa

Una vez estudiados a nivel general los modelos ACPF, ACOPF y SOCP, se procede a una comparación cuantitativa entre ellos con el objetivo de evaluar su escalabilidad, tamaño estructural y potencial impacto computacional.

A continuación, la tabla y el gráfico siguientes permiten observar claramente el aumento progresivo de complejidad estructural a medida que se incrementa el tamaño de la red (de 3 a 30 nodos) y según la formulación empleada.

Tabla 4.26: Comparación tamaño y complejidad de los modelos

Caso	IEEE-3			IEEE-14			IEEE-30		
Modelo	ACPF	ACOPF	SOCP	ACPF	ACOPF	SOCP	ACPF	ACOPF	SOCP
Nº Variables	21	25	31	116	123	175	243	255	359
Nº Restricciones	21	19	35	116	109	216	243	225	442

De manera más visual, se evidencia lo siguiente:

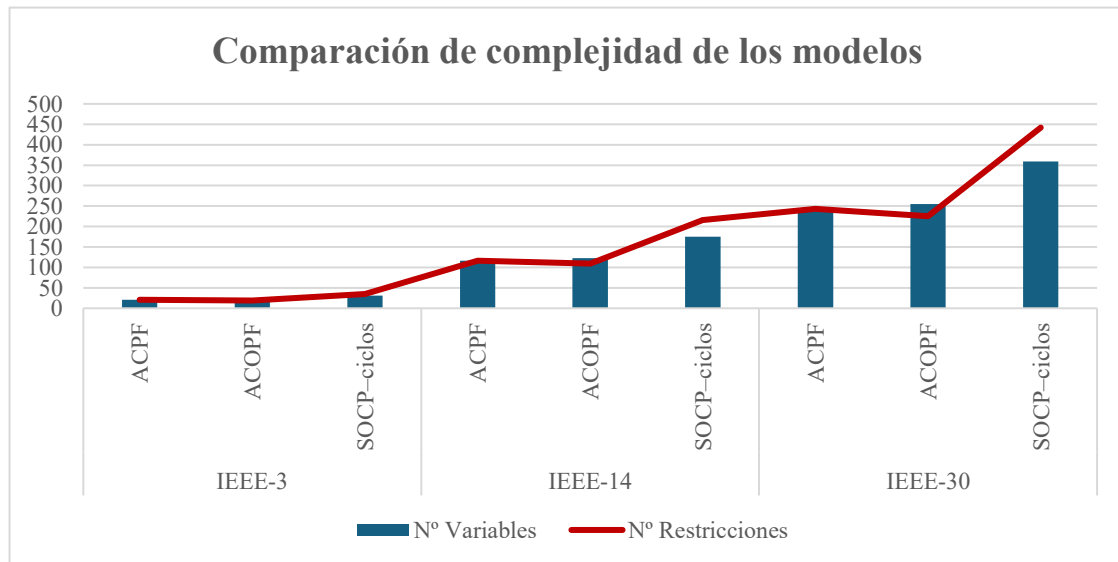


Figura 17: Complejidad de los diferentes modelos en los casos de estudio

En todos los casos, el modelo ACPF presenta la menor cantidad de variables y restricciones, lo que refleja su simplicidad al tratarse de una resolución directa del flujo de potencia sin optimización. En contraste, ACOPF introduce nuevas variables y restricciones relacionadas con la optimización del despacho, lo que incrementa moderadamente el tamaño del problema.

Finalmente, el modelo SOCP con ciclos es el que mayor dimensión alcanza en todas las redes analizadas, debido a la inclusión de variables auxiliares, restricciones cónicas y condiciones de ciclo para reforzar la viabilidad física. Este crecimiento estructural es especialmente marcado en el caso IEEE-30, donde supera las 400 variables y restricciones, duplicando prácticamente el tamaño del ACPF.

Cabe destacar que estos valores no han sido estimados teóricamente, sino que han sido obtenidos directamente durante la ejecución de los modelos en Python. Gracias al parámetro True utilizado en las funciones `display()` y a la impresión de estructuras internas del modelo (como `model.component_data_objects()`), se pueden contabilizar automáticamente tanto las variables como las restricciones activas de cada caso.

Ahora se ha de comprobar si este tamaño afecta a la rapidez y rendimiento de la resolución del modelo. Para este análisis se compararon los modelos ACOPF y SOCP–ciclos en términos de tiempo de ejecución y eficiencia computacional, medida como número de iteraciones por segundo. Cabe señalar que el caso IEEE-3 no se incluye en esta evaluación, ya que su reducido tamaño no representa una carga computacional significativa ni permite observar patrones de escalabilidad. En sistemas tan pequeños, las diferencias entre modelos son mínimas y dominadas por tiempos de inicialización, no por la estructura matemática del problema.

Los casos IEEE-14 e IEEE-30 permiten, en cambio, identificar diferencias más representativas. Aunque el modelo SOCP con ciclos incorpora una mayor cantidad de variables y restricciones debido a su formulación convexa con envolventes angulares y condiciones de ciclo, demuestra un comportamiento computacionalmente eficiente.

Tabla 4.27: Comparación velocidades de cómputo de los modelos

Caso	Modelo	Tiempo [s]	Iteraciones	Mejora en la velocidad (%) [Iteraciones/s]
IEEE-14	ACOPF	0,042	21	8,11%
IEEE-14	SOCP–ciclos	0,037	20	
IEEE-30	ACOPF	0,218	78	11,79%
IEEE-30	SOCP–ciclos	0,065	26	

Estos resultados muestran que, aunque el modelo SOCP con ciclos presenta una estructura más compleja en términos de número de variables y restricciones, su formulación convexa permite una mayor eficiencia computacional por iteración. Esta ventaja no solo se mantiene, sino que incluso se acentúa con respecto al tamaño de la red, lo que pone de manifiesto su buena escalabilidad. Así, el modelo SOCP logra resolver problemas más grandes sin comprometer tiempos de cómputo de forma significativa.

Además, los valores obtenidos en este estudio empírico coinciden con lo reportado en la literatura: el enfoque SOCP ofrece una alternativa robusta al ACOPF clásico, alcanzando soluciones cercanas al óptimo con un desempeño numérico notablemente competitivo. La convexidad del modelo favorece tanto la rapidez como la estabilidad del solver, lo que lo convierte en una opción especialmente atractiva en entornos donde se prioriza eficiencia y viabilidad.

En conjunto, puede concluirse que tanto el tamaño de la red como la formulación empleada tienen un impacto directo sobre la complejidad estructural y el comportamiento computacional. Por ello, la elección del modelo más adecuado dependerá del equilibrio deseado entre fidelidad física, calidad óptima y rendimiento.

En resumen, las tres metodologías presentan los siguientes trade-offs: ACPF ofrece máxima exactitud física y simplicidad (ideal para checks de sistema), ACOPF brinda óptimos reales económicos con total fidelidad (pero a costa de mayor complejidad de solución), y SOCP logra óptimos globales casi factibles con alta eficiencia computacional y buena escalabilidad, admitiendo un mínimo de ajuste posterior.

5. Conclusiones

En esta última sección de la memoria se presentan las conclusiones generales del trabajo, una vez analizados en profundidad los métodos implementados, los resultados obtenidos y las fortalezas y limitaciones del enfoque propuesto.

El objetivo es sintetizar los aprendizajes clave sobre la formulación y resolución del problema de Flujo Óptimo de Potencia en Corriente Alterna (AC-OPF) con relajación SOCP, las posibles líneas de mejora metodológica, y su escalabilidad a sistemas de mayor complejidad. Asimismo, se reflexiona brevemente sobre el potencial de aplicación de este tipo de modelos en entornos reales de operación y planificación de redes eléctricas. Esta síntesis final busca cerrar de forma estructurada el hilo argumental desarrollado a lo largo del proyecto.

5.1 Conclusiones acerca de la metodología

La metodología empleada en este trabajo ha resultado eficaz y coherente con los objetivos planteados. El enfoque escalonado tanto en la complejidad del modelo como en el tamaño de los sistemas analizados ha permitido consolidar progresivamente cada etapa del desarrollo, facilitando tanto la validación teórica como computacional.

Como se ha ido mostrando a lo largo de esta memoria, este proceso comenzó con la implementación del flujo de potencia en corriente alterna (ACPF) – que fue de gran ayuda para aclarar las ecuaciones del BFM –, continuó con la formulación y resolución del flujo óptimo de potencia AC (ACOPF) y culminó con la introducción de la relajación convexa basada en programación cónica de segundo orden (SOCP). De forma paralela, se estudiaron tres sistemas eléctricos de referencia con 3, 14 y 30 nudos, lo que permitió evaluar el rendimiento del modelo bajo diferentes grados de complejidad estructural.

5.1.1 Mejoras para futuros estudios

En conjunto, la metodología aplicada ha resultado adecuada para esta circunstancia de estudio concreta. No obstante, durante el progreso del proyecto se ha demostrado que todavía existen márgenes de mejora para futuros estudios.

Primero, si bien gracias a que el proceso adoptado ha sido manual y detallado, se ha podido comprender a fondo la mecánica de la relajación SOCP; se ha observado que, cuando la formulación mediante el Branch Flow Model (BFM) está bien asentada, no es estrictamente necesario realizar una simulación inicial de ACPF. Esto permitiría una implementación más directa y rápida del ACOPF.

En segundo lugar, la implementación escalonada se ha realizado íntegramente en Python, lo cual ha ofrecido una plataforma flexible, intuitiva y ampliamente personalizable. No obstante, se ha detectado cierta sensibilidad del solver IPOPT (versión 3.14.17) frente a las condiciones iniciales. En algunos casos, el algoritmo detenía su ejecución en las primeras iteraciones al no encontrar una dirección descendente válida. Esta dependencia de la inicialización ha condicionado ligeramente la estabilidad numérica del método, lo cual se ha hecho especialmente evidente en redes malladas, como la IEEE14, y de grandes y más complejas, como la IEEE30, donde pequeños desajustes en la generación podían generar discrepancias relevantes en los flujos por línea. Ante esta situación, existen solvers comerciales especializados en optimización cónica, como GUROBI o MOSEK, los cuales ofrecen un rendimiento superior en la resolución de problemas del tipo SOCP.

Como mejora metodológica, si lo que se busca es una resolución diligente y no un análisis exhaustivo, se podría considerar la incorporación de *solvers* menos sensibles a las condiciones iniciales, como KNITRO, comercializado por Artelys, o incluso aplicaciones como PSS®E o DlgSILENT, que integran procedimientos de inicialización robusta basados en el flujo de carga previo. De la misma manera, herramientas como PyPower, con las que se ha verificado la solución del codirector Erik Francisco, permiten resolver el ACOPF de forma automática. PyPower utiliza una formulación matricial clásica basada en el modelo de admitancias nodales y algoritmos de Newton-Raphson, lo que permite resolver el ACOPF mediante técnicas estándar y comprobar rápidamente el cumplimiento de las restricciones físicas.

5.1.2 Puntos fuertes de la metodología

En cualquier caso, el motivo por el cual se optó por un enfoque más orgánico fue para facilitar el análisis detallado de la relajación convexa. Al ejecutar primero el ACOPF original y luego la versión relajada con SOCP, observando las mejoras en rendimiento del sistema de forma directa, se alcanzó un entendimiento profundo del problema y de las mejoras acontecidas.

De la misma manera, gracias al uso de Python, se pudo habilitar la opción `disp=True` en las funciones de resolución de Pyomo. Por consiguiente, se obtuvo información detallada del proceso iterativo del *solver* (tiempo de ejecución, número de iteraciones, convergencia, cumplimiento de restricciones, etc.), lo que facilitó el análisis comparativo entre ambas formulaciones. Y, tal y como se ha querido demostrar a lo largo del trabajo, ese es el propósito fundamental de este estudio: no solo obtener una solución óptima para el flujo de potencia en los casos analizados, sino también explorar posibles mejoras que permitan alcanzar una solución más eficiente en términos de tiempo de ejecución (una característica de gran relevancia en contextos operativos reales), sin renunciar al realismo físico del sistema. La naturaleza relajada del modelo propuesto, basada en técnicas convexas, no solo mejora la escalabilidad, sino que además facilita la incorporación de ajustes posteriores y el tratamiento de la incertidumbre inherente a los sistemas eléctricos reales con los que operamos en el mundo actual.

Otro aspecto destacable, fue el uso de IPOPT como solucionador, un solver de código abierto, lo que contribuye a que el trabajo desarrollado sea altamente replicable y accesible para otros investigadores o desarrolladores.

Por ello, se puede concluir que el uso Python+IPOPT como entorno de trabajo ha sido una elección acertada: a diferencia de otros entornos más cerrados, ofrece un control total sobre el código, facilita la trazabilidad de errores y permite modificar libremente tanto la formulación matemática como los parámetros del solver. Esto ha permitido adaptar la metodología a los objetivos del proyecto de manera precisa y eficiente.

5.2 Conclusiones acerca de los resultados y líneas futuras de acción

Los tres casos de estudio analizados (IEEE de 3, 14 y 30 nudos) evidenciaron la alta fidelidad y eficacia del modelo propuesto, comparando el Flujo Óptimo de Potencia clásico (ACOPF) con su versión relajada mediante Programación Cónica de Segundo Orden (SOCP). En los sistemas IEEE-14 e IEEE-30, el ACOPF tradicional obtuvo despachos óptimos prácticamente idénticos a las soluciones de referencia, cumpliendo todas las restricciones operativas y manteniendo la energía no suministrada en cero. La versión relajada con SOCP, enriquecida con restricciones cíclicas de Kirchhoff, alcanzó soluciones igualmente consistentes: en IEEE-30 la coincidencia con el óptimo original fue prácticamente exacta, mientras que en IEEE-14 el modelo SOCP logró un costo operativo aproximadamente un 8,5% menor que el ACOPF. Este ahorro sugiere que la relajación pudo hallar un óptimo más económico, si bien requiere cautela al analizar su factibilidad física.

En cuanto a precisión y factibilidad, los resultados confirman que la formulación SOCP es muy ajustada (tight) respecto al modelo no convexo. Las diferencias numéricas fueron marginales en la mayoría de las variables: los perfiles de tensiones, flujos de potencia y repartos de generación del SOCP se alinean estrechamente con los del ACOPT, sin introducir violaciones significativas de las leyes físicas del sistema. La demanda activa y reactiva se satisfizo por completo en todos los casos ($ENS = 0$), lo que indica que la relajación no recurre a artificios como cortes de carga. Si bien la solución SOCP pertenece a una región factible ampliada, se verificó su viabilidad reinyectándola en un flujo de carga AC: el algoritmo convergió sin inconvenientes y cerró los balances nodales con residuos numéricos insignificantes. Esto confirma que los óptimos hallados por el SOCP, tras incorporar las restricciones cíclicas, son casi factibles para el problema original, requiriendo ajustes particulares en caso de discrepancias.

5.2.1 Recomendaciones para futuros estudios

No obstante, en algunos casos se detectaron violaciones puntuales de restricciones de los límites térmicos de las líneas, lo que pone de manifiesto que la relajación, aun siendo una herramienta potente, no es completamente exacta y que, si bien ofrece un punto de partida excelente, necesita ser refinada para su uso en entornos reales. Por ello, de cara a futuros estudios se ha de tener en cuenta lo siguiente:

Se observó que el modelo responde favorablemente al endurecimiento y refinamiento de la envolvente angular —bien de forma global, como en este trabajo, o de manera selectiva sobre las líneas más críticas. De la misma manera, se puede implementar un refuerzo de la relación entre corriente y límite térmico. Ambas son vías efectivas para cerrar la brecha entre el modelo relajado y la factibilidad física. En conjunto, los resultados respaldan el potencial del enfoque SOCP como herramienta avanzada para mejorar el estado del arte, pero también evidencian que debe complementarse con técnicas adicionales de ajuste: cortes automáticos, penalizaciones específicas o un re-despacho local que redistribuya la generación, sobre todo en redes con alta penetración de renovables. Si la reforma del modelo resultara demasiado compleja, puede adoptarse una alternativa más sencilla: usar la solución obtenida tras la reinserción AC como *warm start* en sucesivas reinserciones, acercándose gradualmente al óptimo sin modificar la formulación original.

Desde el punto de vista computacional, la relajación SOCP demostró ser más eficiente que la formulación no convexa. Al convexificar el problema se evita la dependencia de condiciones iniciales críticas y óptimos locales pobres; los algoritmos de punto interior explotan la estructura cónica resolviendo sistemas de gran tamaño en tiempos compatibles con la operación en línea. En suma, el enfoque SOCP ofrece un compromiso muy favorable entre precisión y tiempo de cómputo, garantizando soluciones cercanas al óptimo global con un coste computacional reducido, siempre que se valide posteriormente la factibilidad AC.

De la misma manera, una posible mejora en cuanto al realismo del modelo para futuros estudios, consistiría en tener en cuenta la naturaleza multiperiodo del SOCP, incluyendo las dinámicas temporales, a diferencia del modelo monofásico estático desarrollado en este proyecto.

5.2.2 Líneas futuras de acción: Gestión de incertidumbre con técnicas SOCP

En perspectiva futura, la integración efectiva de la relajación SOCP en entornos reales resulta técnicamente viable, siempre que se implementen mecanismos de corrección para subsanar las discrepancias observadas frente al modelo físico original, especialmente en lo relativo al cumplimiento estricto de las restricciones térmicas y angulares. Adicionalmente, su aplicabilidad práctica dependerá de la capacidad del sistema para adaptarse a entornos operativos con alta variabilidad, lo que requiere desarrollar estrategias de gestión robusta de la incertidumbre asociada a la generación renovable.

Bajo estas condiciones, el modelo ACOPF+SOCP se perfila como una herramienta operativa con gran potencial para la planificación y optimización de redes eléctricas modernas. Afortunadamente, la capacidad de la relajación SOCP para manejar conjuntos amplios de condiciones operativas y permanecer estable ante cambios súbitos refuerza su utilidad en entornos reales: por ejemplo, el operador del sistema puede recalcular puntos óptimos de operación frente a variaciones bruscas de generación renovable manteniendo la seguridad y calidad del suministro. De igual modo, en redes de distribución de baja tensión con alta penetración de DERs (recursos energéticos distribuidos), las técnicas basadas en SOCP ofrecen una herramienta potente para el despacho óptimo en presencia de incertidumbre, minimizando pérdidas y gestionando congestiones con garantías de optimalidad física y económica. En conjunto, estas características alinean la metodología con las necesidades de las futuras redes inteligentes, donde la resiliencia y sostenibilidad dependen de aprovechar al máximo la flexibilidad operativa ante la intermitencia renovable.

5.3 El futuro de las redes eléctricas (ACOPF+SOCP+Renovables)

Como se ha señalado anteriormente en la memoria, el mundo del mañana se enfrenta a una creciente penetración de energías renovables. Esto plantea considerables desafíos para los sistemas de red eléctrica. En España (líder en energía eólica y solar), según se recoge en los informes de Red Eléctrica (REEE25iii), citados en la memoria, el 66 % de la potencia instalada es renovable, el 98 % de la generación renovable disponible se integra instantáneamente en la red, y las fuentes limpias aportaron el 56,8 % del mix eléctrico nacional en 2024. Esto implica que operadores deben gestionar continuamente flujos variables con alta incertidumbre.

El Flujo Óptimo de Potencia en AC (ACOPF) tradicional, no convexo y de elevada complejidad, se vuelve difícil de resolver eficientemente en tiempo real en un escenario con alta ER.

Tal y como se ha expuesto repetidamente en distintos apartados de este trabajo, y poniendo el foco en la eficiencia computacional al momento, la SOCP multiperiodo resulta atractiva para estas aplicaciones en tiempo real, de gran escala y cambiantes. Si bien son muchos los estudios que demuestran que el modelo SOCP ofrece alguna pérdida de exactitud frente al SDP, esta metodología presenta muy alta eficiencia computacional ante numerosas restricciones y constantes cambios. Dicha pérdida, como indica la literatura académica (YUAN19), puede subsanarse a través de envolventes y restricciones lineales dinámicas (ciclos) sobre las funciones trigonométricas, algo que se ha implementado en este estudio.

5.3.1 Aplicación en Redes de Transporte (Sistemas mallados)

En las redes de transporte de alta tensión, típicamente malladas y de topología compleja, el problema AC-OPF incluye numerosas variables (ángulos de fase, tensiones, potencias en líneas) acopladas a través de las leyes de Kirchhoff. En estos sistemas, es posible que la solución de la relajación SOCP no satisfaga exactamente todas las ecuaciones originales (e.g., podría violar ligeramente la ley de tensiones en algún ciclo), dando lugar a soluciones no factibles para el problema original. Sin embargo, la experiencia ha mostrado que, en muchos casos prácticos, la solución relajada sí coincide con un óptimo factible o bien puede refinarse fácilmente (KOCU16).

Un enfoque común para subsanar este error es emplear la solución de la relajación SOCP como base para un ajuste posterior que ayude a ese refinamiento. La alta calidad de la solución cónica la hace un excelente punto de arranque (*warm start*) para algoritmos no lineales tradicionales, acelerando su convergencia hacia un óptimo factible cercano al global (KOCU16).

Desde la perspectiva operativa, la capacidad de resolver de forma rápida y casi óptima el OPF en el sistema de transporte es muy valiosa. Permite, por un lado, evaluar límites operativos y cuellos de botella en tiempo próximo al real, y por otro, habilita la realización de análisis de contingencias o de seguridad. En el contexto español, donde la red de transporte está interconectada con la

europea, pero tiene ciertas limitaciones de interconexión, disponer de soluciones óptimas globales ayuda a maximizar la utilización segura de las líneas existentes y facilita la integración de nueva generación renovable lejana a centros de consumo. El operador del sistema puede beneficiarse de la rapidez del SOCP para recalcular puntos de operación óptimos ante cambios súbitos (como variaciones bruscas de producción eólica o fotovoltaica) manteniendo la seguridad del sistema. Es importante resaltar que, al ser la formulación SOCP mucho más escalable, permite abordar modelos de red de transporte muy detallados o de gran tamaño sin incurrir en tiempos de cómputo prohibitivos (KOCU16). Esto la posiciona como una herramienta práctica de operación en tiempo real o en procesos iterativos de despacho económico, complementando o sustituyendo métodos heurísticos tradicionales.

Además, la red de transporte en España enfrenta importantes retos debido a la creciente penetración de generación renovable variable, lo que requiere no solo una optimización efectiva de la operación, sino también una planificación adecuada de la expansión futura del sistema. En este contexto, investigaciones recientes como la de Alvarez Quispe (ALVA24) han introducido metodologías avanzadas y robustas para la planificación óptima de expansión de sistemas eléctricos, destacando especialmente la integración explícita de recursos flexibles centralizados y distribuidos, recursos energéticos distribuidos (DER, por sus siglas en inglés), tales como generación fotovoltaica y eólica distribuida, generación combinada de calor y energía, vehículos eléctricos y sistemas de respuesta de demanda, además de almacenamiento energético de gran escala, como baterías electroquímicas y sistemas de almacenamiento hidráulico por bombeo. Dichas metodologías avanzadas permiten un análisis integral y exhaustivo de las complejas interacciones entre generación, transporte y distribución, contribuyendo significativamente a la identificación precisa de necesidades futuras de expansión y localización estratégica de nuevas infraestructuras.

Este enfoque sofisticado unido a las relajaciones fortalece la robustez y la flexibilidad operativa frente a escenarios altamente inciertos derivados de la intermitencia renovable, alineándose eficazmente con los objetivos nacionales e internacionales de transición energética hacia sistemas eléctricos más sostenibles, seguros y resilientes

5.4 Impacto económico

Si bien este trabajo no se centra en el análisis económico ni incorpora métricas financieras específicas, es importante destacar la relevancia que tiene el problema del AC-OPF en la economía actual y futura, puesto que se centra -como se muestra en los resultados-, en buscar el menor coste posible al mayor rendimiento del sistema. Por lo tanto, es evidente que implementar este enfoque en el futuro tiene un impacto directo en la eficiencia operativa y en la optimización de costes del sistema eléctrico.

Este breve inciso resulta relevante debido a que a lo largo del presente año se ha puesto claramente de manifiesto la importancia crucial de operar una red dentro de sus límites establecidos. Por ejemplo, el apagón ocurrido en España en 2025 resaltó las consecuencias graves que pueden derivarse cuando, buscando la mayor generación, no se garantiza una operación segura y dentro de límites preestablecidos. Asimismo, las fluctuaciones recientes en los precios de la electricidad, que han alcanzado niveles históricamente elevados, subrayan la importancia de implementar soluciones precisas y eficientes, como las proporcionadas por el ACOPF.

El Flujo Óptimo de Potencia en CA no solo tiene importancia técnica, sino también un impacto económico directo en la operación de los sistemas eléctricos. En los mercados eléctricos modernos, el ACOPF constituye el núcleo del despacho económico: las decisiones de qué generadores operan y a qué nivel se toman resolviendo este problema de optimización bajo restricciones físicas (FARI22). De hecho, operadores independientes del sistema (ISOs) y otros agentes resuelven variantes del OPF de manera rutinaria (diaria e incluso cada 5 minutos en

tiempo real) para asignar la generación al menor coste posible cumpliendo con la seguridad de red (ONEI12). Un ACOPF bien resuelto asegura que la energía se genere al coste mínimo necesario manteniendo las tensiones, flujos y demás parámetros dentro de límites seguros.

Históricamente, debido a la complejidad del AC-OPF completo (no lineal y no convexo), los operadores han utilizado aproximaciones como el DCOPF para calcular el despacho. Si bien, como se menciona con anterioridad, el DCOPF simplifica el problema, ignorar la naturaleza AC (por ejemplo, las pérdidas y la potencia reactiva) conlleva soluciones subóptimas desde el punto de vista económico. Estudios recientes han cuantificado esta brecha: se estima que las aproximaciones del ACOPF implican sobrecostos significativos en la operación. Por ejemplo, de acuerdo con los autores del artículo de la revista *IEEE Smart Grid*, la no optimalidad de las soluciones actuales podría estar costando del orden de 6 a 19 mil millones de dólares al año en Estados Unidos (FARI22). En otras palabras, el uso de modelos aproximados en lugar del ACOPF exacto se traduce en un despacho menos eficiente, obligando a redispatch (reajustes posteriores) y pagos compensatorios que encarecen el funcionamiento del sistema.

Mejorar la fidelidad del modelo de OPF a través de técnicas avanzadas (p. ej., relajaciones no lineales más ajustadas o métodos convexos) puede revertir esos sobrecostos. Bichler y Knörr (BICH23) demuestran que al emplear modelos AC más precisos o relajaciones convexas del AC-OPF, aumenta el bienestar económico del mercado eléctrico y se reducen los costes de redispatch y pagos fuera de mercado (side-payments) necesarios para garantizar la viabilidad de la solución. En particular, los precios locacionales resultantes reflejan mejor las condiciones reales de la red, evitando señales de precios distorsionadas que puedan llevar a decisiones ineficientes de consumo o inversión (BICH23). Este hallazgo subraya que una solución AC-OPF más exacta no solo reduce el coste operativo inmediato, sino que mejora la eficiencia a largo plazo al proporcionar señales económicas más correctas.

En resumen, el impacto económico del AC-OPF se manifiesta en dos ámbitos principales: operativo (minimizando los costos diarios de generación y evitando medidas correctivas costosas) y estratégico (permitiendo precios y señales de inversión adecuados que favorecen un desarrollo eficiente del sistema a futuro). Implementar metodologías que resuelvan o aproximen mejor el AC-OPF (por ejemplo, mediante relajaciones convexas como SOCP o SDP, o algoritmos globalmente óptimos) puede traducirse en ahorros millonarios para los sistemas eléctricos y en una operación más segura y sostenible, al reducir también las pérdidas y emisiones asociadas a ineficiencias (ONEI12). Por todo ello, la búsqueda de soluciones AC-OPF más precisas no es solo un desafío teórico, sino una necesidad económica palpable en la transición hacia redes eléctricas más inteligentes y limpias.

6. Bibliografía

[REEE24] Red Eléctrica de España (REE), “Informe del Sistema Eléctrico Español 2024”, marzo 2025.

[REEE25i] Red Eléctrica de España (REE), “Publicación de solicitudes de acceso de demanda. Marzo 2025”, marzo 2025.

[REEE25ii] Red Eléctrica de España (REE), “Capacidad de intercambio internacional”, 2025.

[REEE25iii] Red Eléctrica de España (REE), “Integración de energías renovables”. Disponible en: <https://www.ree.es/es/operacion/integracion-renovables>, consultado en junio de 2025.

[MOMO99] Momoh J.A., El-Hawary M.E., Adapa R., “A Review of Selected Optimal Power Flow Literature to 1993. Part I: Nonlinear and Quadratic Programming Approaches & Part II: Newton, Linear Programming and Interior Point Methods”, IEEE Transactions on Power Systems, vol. 14, no. 1, pp. 96–111, 1999.

[FRAN12] Frank S.M., Steponavice I., Rebennack S., “Optimal Power Flow: A Bibliographic Survey I”, Energy Systems, vol. 3, no. 3, pp. 221–258, 2012.

[ONEI12] O’Neill R.P., Castillo A., Cain M.B., “The computational testing of AC Optimal Power Flow using the IV formulation”, Federal Energy Regulatory Commission, 2012.

[HUAN16] Huang S., Wu Q., Wang J., Zhao H., “A Sufficient Condition on Convex Relaxation of AC Optimal Power Flow in Distribution Networks”, IEEE Transactions on Power Systems, vol. 32, no. 2, pp. 1359–1368, 2016.

[YUAN19] Yuan Z., Hesamzadeh M.R., “Second-order cone AC optimal power flow: convex relaxations and feasible solutions”, Journal of Modern Power Systems and Clean Energy, vol. 7, no. 2, pp. 268–280, 2019.

[CARP62] Carpentier, J., “Contribution à l’étude du dispatching économique”, Bulletin de la Société Française des Électriciens (SFE), 1962

[MONT83] Monticelli A., Liv T., “Fluxo de Carga em Sistemas de Energia Elétrica”, Universidade Estadual de Campinas, 1983.

[MARL17] Marley J.F., Molzahn D.K., Hiskens I.A., “Solving multiperiod OPF problems using an AC-QP algorithm initialized with an SOCP relaxation”, IEEE Transactions on Power Systems, vol. 32, no. 5, 2017.

[KOCU14] Kocuk B., Dey S.S., Sun X.A., “Cycle-based decomposition for solving optimal power flow with SOCP relaxation”, Informe técnico, Georgia Institute of Technology, 2014.

[ALVA19] Álvarez Quispe, E. F., “Semidefinite Relaxation for the Optimal Operation and Expansion Planning of Power Transmission Systems”, Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, São Paulo, Brasil, 2019.

[PAN20] Pan X., Zhao T., Chen M., “DeepOPF: Deep neural network for DC optimal power flow”, The Chinese University of Hong Kong, 2020

- [KARD22] Kardoš J., Kourounis D., Schenk O., Zimmerman R., "*BELTISTOS: A robust interior point method for large-scale optimal power flow problems*", Electric Power Systems Research, vol. 212, no. 108613, 2022.
- [GART12] Gärtner B., Matoušek J., "*Approximation Algorithms and Semidefinite Programming*", Springer, 2012.
- [VAND96] Vandenberghe L., Boyd S., "*Semidefinite Programming*", SIAM Review, vol. 38, no. 1, pp. 49–95, 1996.
- [GHAD19] Ghaddar B., Jabr R.A., "*Power transmission network expansion planning: A semidefinite programming branch-and-bound approach*", European Journal of Operational Research, vol. 274, no. 3, pp. 837–844, 2019.
- [ALVA24] Álvarez Quispe E.F., "*Improving Modelling for Optimal Expansion Planning of Power Transmission Systems*", Tesis Doctoral, Universidad Pontificia Comillas, ICAI, 2024.
- [MADR99] Madrigal M., Quintana V., "*An interior-point method for optimal power flow with a discontinuous objective function*", IEEE Transactions on Power Systems, vol. 14, no. 4, pp. 1469–1475, 1999.
- [NGUY19] Nguyen H. D., Dvijotham K., Turitsyn K., "*Constructing convex inner approximations of steady-state security regions*", IEEE Transactions on Power Systems, vol. 34, no. 1, pp. 257–267, 2019.
- [ALIZ03] Alizadeh F., Goldfarb D., "*Second-order cone programming*", Mathematical Programming, vol. 95, no. 1, pp. 3–51, 2003.
- [NEST97] Nesterov Y., Todd M., "*Self-scaled barriers and interior-point methods for convex programming*", Mathematics of Operations Research, vol. 22, no. 1, pp. 1–42, 1997.
- [POP22] Pop C.B., Cioara T., Anghel I., Antal M., Chifu V.R., Antal C., Salomie I., "*Review of bio-inspired optimization applications in renewable-powered smart grids: Emerging population-based metaheuristics*" Energy Reports, vol. 8, pp. 11769–11798, 2022.
- [DIAB24] Diab, A.A.Z., Abdelhamid, A.M., Sultan, H.M., "*Comprehensive analysis of optimal power flow using recent metaheuristic algorithms*", Scientific Reports, vol. 14, art. 13422, 2024.
- [HERB16] Herbadji, O., Slimani, L., Bouktir, T., "*Solving Bi-Objective Optimal Power Flow using Hybrid method of Biogeography-Based Optimization and Differential Evolution Algorithm: A case study of the Algerian Electrical Network*", Journal of Electrical Systems, vol. 12, no. 1, pp. 197–215, 2016.
- [SHAR25] Sharma V., Naresh R., Verma P., "*Hybrid metaheuristic approach for multi-objective AC optimal power flow with renewable integration*" International Journal of Electrical Power & Energy Systems, vol. 134, 107336, 2025.
- [JIAN24] Jiang B., Wang Q., Wu S., Wang Y., Lu G., "*Advancements and Future Directions in the Application of Machine Learning to AC Optimal Power Flow: A Critical Review*", Energies, vol. 17, n° 6, art. 1381, 2024.

- [PILO24] Piloto L., Liguori S., Madjiheurem S., Zgubič M., Lovett S., Tomlinson H., Elster S., Apps C., Witherspoon S., “*CANOS: A Fast and Scalable Neural AC-OPF Solver Robust to N-1 Perturbations*”, 2024.
- [GHAS18] Ghassempour Aghamolki H., Miao Z., Fan L., “*SOCP Convex Relaxation-Based Simultaneous State Estimation and Bad Data Identification*” Department of Electrical Engineering, University of South Florida, 2018.
- [SLIW19] Sliwak J., Anjos M.F., Létocart L., Maeght J., Traversi E., “*Improving Clique Decompositions of Semidefinite Relaxations for Optimal Power Flow Problems*”, 2019.
- [MA20] Ma W., Fan L., Miao Z., Zeng B., Ghassempour Aghamolki H., “*A Sparse Convex AC OPF Solver and Convex Iteration Implementation Based on 3-Node Cycles*”, Electric Power Systems Research, vol.180, art. 106169, 2020.
- [COIN24] COIN-OR, “*Ipopt Documentation*”. Disponible en: <https://coin-or.github.io/Ipopt/>, consultado en junio de 2025.
- [PYOM24] Pyomo Contributors, “*Pyomo Documentation (v6.7.1)*”. Disponible en: <https://pyomo.readthedocs.io/en/stable/>, consultado en junio de 2025.
- [CHOW24] Chowdhury, M.M.U.T., Kamalasadan S., Paudyal S., “*A Second-Order Cone Programming (SOCP) Based Optimal Power Flow (OPF) Model With Cyclic Constraints for Power Transmission Systems*”, IEEE Transactions on Power Systems, vol. 39, n.º 1, pp. 1032–1044. Enero 2024.
- [KOCU16] Kocuk B., Dey S. S., Sun X. A., “*Strong SOCP Relaxations for the Optimal Power Flow Problem*” Operations Research, vol. 64, no. 6, pp. 1177-1196, 2016.
- [GAN13] Gan L., Li N., Low S.H., Topcu U., “*Exact convex relaxation for optimal power flow in distribution networks*” ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 1, pp. 351–352, 2013.
- [CHEN18] Chen Y., Xiang J., Li Y., “*SOCP Relaxations of Optimal Power Flow Problem Considering Current Margins in Radial Networks*” Energies, vol. 11, no. 11, art. 3164, 2018.
- [FARI22] Farid, A. M., “*A Profit-Maximizing Security-Constrained IV-AC Optimal Power Flow Model & Global Solution*”, IEEE Transactions on Smart Grid, vol. 10, pp. 2842–2859, 2022.
- [BICH23] Bichler, M. y Knörr, J., “*Getting Prices Right on Electricity Spot Markets: On the Economic Impact of Advanced Power Flow Models*”, Energy Economics, vol. 126, art. 106968, 2023.

7. Anexo I: Alineación con los Objetivos de Desarrollo Sostenible (ODS)

Este proyecto se encuentra claramente alineado con varios de los Objetivos de Desarrollo Sostenible (ODS), establecidos por la Organización de Naciones Unidas (ONU) en la Agenda 2030.

La optimización del flujo de potencia en redes eléctricas gracias al uso de métodos matemáticos avanzados contribuye directamente a la transformación del modelo energético. No solo es una herramienta técnica que consigue mejorar el rendimiento operativo, sino que también genera un impacto positivo a nivel económico, social y medioambiental.

En el desarrollo de un modelo energético más eficiente, sostenible, resiliente e inclusivo, el proyecto guarda una relación directa con los siguientes Objetivos de Desarrollo Sostenible de la Agenda 2030:

7.1 ODS 7, 11, 13

Una de las principales aportaciones de este estudio al marco del desarrollo sostenible se articula en torno a los *ODS 7: Energía Asequible y no contaminante*, *ODS 11: Ciudades y comunidades sostenibles* y *ODS 13: Acción por el clima*. Todos ellos, están estrechamente ligados al contexto de la transición energética.

Para garantizar el acceso a una electricidad más asequible y fiable, en línea con la meta del ODS 7, se han de reducir los costes operativos del sistema. Por ello, la optimización del flujo de potencia en las redes resulta de gran ayuda. Al mejorar la eficiencia energética, se minimizan las pérdidas en el transporte y distribución, y se reduce el coste operativo del sistema.

A su vez, al verse mejorada la eficiencia operativa, y con ella la flexibilidad del sistema, se facilita la integración de fuentes de energía renovable, las cuales sufren continuas fluctuaciones. Esto abre las puertas a la descarbonización de la red y a la oportunidad de avanzar en el desarrollo de comunidades más sostenibles, tal y como pretende el ODS 11. Inequívocamente, las redes optimizadas contribuyen a una planificación urbana mejorada y perfeccionan sectores clave como la movilidad eléctrica, el alumbrado o la gestión de residuos.

Todo ello, repercute directamente en la lucha contra el cambio climático, pilar del ODS 13. Una producción eficiente de energía permite generar menos, lo cual reduce la emisión de gases de efecto invernadero, para cubrir la misma demanda. De hecho, según la ONU, mejorar la eficiencia energética es una de las vías más rentables e inmediatas para reducir las emisiones globales y limitar el calentamiento global a 1.5 °C (ONU, s.f.). La aplicación de técnicas como la Programación Convexa de Segundo Orden (SOCP) permite avanzar hacia sistemas eléctricos inteligentes y sostenibles, capaces de responder de forma eficiente a los desafíos ambientales y sociales actuales.

7.2 ODS 9

El consejo de las Naciones Unidas ha enfatizado que la construcción de infraestructuras resilientes, desde transporte hasta el regadío, es crucial para el desarrollo económico y el bienestar humano futuro.

Al hacer uso de las técnicas matemáticas avanzadas de optimización y promover el uso de tecnologías emergentes aplicadas al sector eléctrico, en este caso: modelos de optimización convexa, se contribuye al desarrollo e infraestructuras eléctricas más inteligentes, sostenibles, inclusivas y robustas.

El ODS 9 busca: *Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación*. Al mejorar de esta manera el rendimiento operativo de las redes, se cimienta la implementación de las Smart Grids.

7.2.1 Smart Grids

También conocidas como redes eléctricas inteligentes, las Smart Grids son sistemas que integran tecnologías digitales avanzadas para supervisar, gestionar y optimizar en tiempo real todos los niveles del sistema eléctrico. Estas condiciones son indispensables para la implementación del AC-OPF en entornos reales complejos.

La supervisión es llevada a cabo gracias a los sistemas SCADA, sensores IoT (Internet of Things) y PMUs (Phasor Measurements Units) que permiten obtener mediciones eléctricas sincronizadas y altamente precisas. Estos datos, que reflejan el estado instantáneo del sistema, son fundamentales a la hora de formular de manera dinámica el AC-OPF y ajustar a tiempo real las variables de control (módulo de tensión, ángulo de tensión, potencias y flujos de línea). La implementación y resolución del AC-OPF está basada precisamente en la disponibilidad permanente de datos.

La gestión y control es asistida por algoritmos de aprendizaje automático o incluso inteligencia artificial. Esta es llevada a cabo por plataformas operativas como EMS (Energy Management Systems) y DMS (Distribution Management Systems) que realizan predicciones de demanda, estimaciones de estado y respuestas automáticas ante desviaciones. Además, gracias a los controladores automáticos distribuidos, es posible implementar versiones descentralizadas del AC-OPF que optimicen el funcionamiento de subredes, microrredes o zonas con alta penetración renovable, respetando las restricciones de potencia, tensión y seguridad en todo momento.

Por último, las Smart Grids permiten aplicar el AC-OPF en escenarios de alta variabilidad, como aquellos en los que se aplica energía eólica y solar. Tecnologías como los sistemas de respuesta a la demanda, la gestión de almacenamiento distribuido, o el control activo de generación renovable interactúan directamente con los modelos AC-OPF para maximizar la eficiencia de la red. Esta interacción se ve reforzada por la infraestructura digital de comunicaciones (fibra óptica, 5G, LPWAN) y plataformas de procesamiento como cloud computing y edge computing, que facilitan la resolución rápida de problemas AC-OPF en redes de gran escala, promoviendo así la evolución hacia una infraestructura energética automatizada, optimizada y resiliente, en línea con los principios del ODS 9.

7.3 ODS 12

De la misma manera, este proyecto guarda relación con el ODS 12: *Producción y consumo responsables*. Al promover una producción de potencia menos derrochadora y así minimizar las pérdidas de energía, se alcanza un transporte y distribución más racional. Al no tener que contabilizar un nivel elevado de pérdidas, la necesidad de producir energía en exceso mengua. En consecuencia, se reduce la presión sobre los recursos naturales (carbón, uranio, gas natural...) y se potencia el desarrollo de un modelo de consumo más eficiente y responsable.

Este enfoque promueve la implementación de buenas prácticas en la operación de los sistemas eléctricos. Lo cual, como la ONU destaca, garantizará la subsistencia de las generaciones actuales y futuras.

8. Anexo II: Descripción detallada de casos de estudio

8.1 Caso de 3 nudos

```
from numpy import array

def case3ierik():
    ppc = {"version": '2'}

    ## system MVA base
    ppc["baseMVA"] = 100.0

    ## bus data
    # bus_i, type, Pd, Qd, Gs, Bs, area, Vm, Th0, baseKV, zone, Vmax, Vmin
    ppc["bus"] = array([
        [1, 3, 110, 40, 0.00, 0.00, 1, 1.100, 0.00, 240.0, 1, 1.100, 0.900],
        [2, 2, 110, 40, 0.00, 0.00, 1, 0.92600, 7.26, 240.0, 1, 1.100, 0.900],
        [3, 2, 95, 50, 0.00, 0.00, 1, 0.900, -17.27, 240.0, 1, 1.100, 0.900]
    ])

    ## generator data
    # bus, Pg, Qg, Qmax, Qmin, Vg, mBase, status, Pmax, Pmin, Pc1, ..., apf
    ppc["gen"] = array([
        [1, 148.07, 54.7, 1000, -1000.00, 1.06, 100.0, 1, 2000, 0.00, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0],
        [2, 170.01, -8.79, 1000, -1000, 1.045, 100.0, 1, 2000, 0.00, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0, 0],
        [3, 0, -4.84, 1000, -1000.00, 1.045, 100.0, 1, 0.00, 0.00, 0, 0, 0, 0, 0, 0,
         0, 0, 0, 0, 0],
        [3, 0, 0, 0, 0, 1.06, 100.0, 0, 180, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    ])

    ## branch data
    # fbus, tbus, r, x, b (total), rateA, rateB, rateC, ratio, angle, status, angmin,
    angmax
    ppc["branch"] = array([
        [1, 2, 0.042, 0.9, 0.3, 88, 88, 88, 0.00, 0.0, 1, -30.0, 30.0],
        [1, 3, 0.065, 0.62, 0.45, 88, 88, 88, 0.00, 0.0, 1, -30.0, 30.0],
        [3, 2, 0.025, 0.75, 0.70, 88, 88, 88, 0.00, 0.0, 1, -30.0, 30.0]
    ])

    ## generator cost data
    # model, startup, shutdown, n, c(n-1), ..., c0
    ppc["gencost"] = array([
        [2, 0.0, 0.0, 3, 0.11, 5.000000, 0.000000],
        [2, 0.0, 0.0, 3, 0.085, 1.200000, 0.000000],
        [2, 0.0, 0.0, 3, 0.000000, 0.000000, 0.000000],
        [2, 0.0, 0.0, 3, 0.000000, 3000.000000, 0.000000]
```

```
)
return ppc
```

8.2 Caso de 14 nudos

```
from numpy import array

def case142():
    ppc = {"version": '2'}

    ## system MVA base
    ppc["baseMVA"] = 100.0

    ## bus data
    # bus_i, type, Pd, Qd, Gs, Bs, area, Vm, Th0, baseKV, zone, Vmax, Vmin
    ppc["bus"] = array([
        [1, 3, 0.00, 0.00, 0.00, 0.00, 1, 1.0600, 0.00, 1.0, 1, 1.0500, 0.9500],
        [2, 2, 21.7, 12.7, 0.00, 0.00, 1, 1.0450, -4.98, 1.0, 1, 1.0500, 0.9500],
        [3, 2, 94.2, 19, 0.00, 0.00, 1, 1.0100, -12.72, 1.0, 1, 1.0500, 0.9500],
        [4, 0, 47.8, -3.9, 0.00, 0.00, 1, 1.0190, -10.33, 1.0, 1, 1.0500, 0.9500],
        [5, 0, 7.6, 1.6, 0.00, 0.00, 1, 1.0200, -8.78, 1.0, 1, 1.0500, 0.9500],
        [6, 2, 11.2, 7.5, 0.00, 0.00, 1, 1.0700, -14.22, 1.0, 1, 1.0500, 0.9500],
        [7, 0, 0.00, 0.00, 0.00, 0.00, 1, 1.0620, -13.37, 1.0, 1, 1.0500, 0.9500],
        [8, 2, 0.00, 0.00, 0.00, 0.00, 1, 1.0900, -13.36, 1.0, 1, 1.0500, 0.9500],
        [9, 0, 29.5, 16.6, 0.00, 0.19, 1, 1.0560, -14.94, 1.0, 1, 1.0500, 0.9500],
        [10, 0, 9, 5.8, 0.00, 0.00, 1, 1.0510, -15.10, 1.0, 1, 1.0500, 0.9500],
        [11, 0, 3.5, 1.8, 0.00, 0.00, 1, 1.0570, -14.79, 1.0, 1, 1.0500, 0.9500],
        [12, 0, 6.1, 1.6, 0.00, 0.00, 1, 1.0550, -15.07, 1.0, 1, 1.0500, 0.9500],
        [13, 0, 13.5, 5.8, 0.00, 0.00, 1, 1.0500, -15.16, 1.0, 1, 1.0500, 0.9500],
        [14, 0, 14.9, 5, 0.00, 0.00, 1, 1.0360, -16.04, 1.0, 1, 1.0500, 0.9500]
    ])

    ## generator data
    # bus, Pg, Qg, Qmax, Qmin, Vg, mBase, status, Pmax, Pmin, Pc1, ..., apf
    ppc["gen"] = array([
        [1, 232, -17, 10, 0.00, 1.06, 100.0, 1, 340, 0.00, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [2, 40, 42, 30, -30, 1.045, 100.0, 1, 59, 0.00, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [3, 0.00, 23, 40, 0.00, 1.01, 100.0, 1, 0, 0.00, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [6, 0.00, 12, 24, -6, 1.07, 100.0, 1, 0, 0.00, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        [8, 0.00, 17, 24, -6, 1.09, 100.0, 1, 0.00, 0.00, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
    ])

    ## branch data
```

```

# fbus, tbus, r, x, b(total), rateA, rateB, rateC, ratio, angle, status, angmin,
angmax
ppc["branch"] = array([
    [1, 2, 0.01938, 0.05917, 0.0528, 170,170,170, 0.000, 0.0, 1,-30.0,30.0],
    [1, 5, 0.05403, 0.22304, 0.0492, 90, 90, 90, 0.000, 0.0, 1,-30.0,30.0],
    [2, 3, 0.04699, 0.19797, 0.0438, 90, 90, 90, 0.000, 0.0, 1,-30.0,30.0],
    [2, 4, 0.05811, 0.17632, 0.0340, 70, 70, 70, 0.000, 0.0, 1,-30.0,30.0],
    [2, 5, 0.05695, 0.17388, 0.0346, 50, 50, 50, 0.000, 0.0, 1,-30.0,30.0],
    [3, 4, 0.06701, 0.17103, 0.0128, 30, 30, 30, 0.000, 0.0, 1,-30.0,30.0],
    [4, 5, 0.01335, 0.04211, 0.0000, 70, 70, 70, 0.000, 0.0, 1,-30.0,30.0],
    [4, 7, 0.00000, 0.20912, 0.0000, 50, 50, 50, 0.978, 0.0, 1,-30.0,30.0],
    [4, 9, 0.00000, 0.55618, 0.0000, 30, 30, 30, 0.969, 0.0, 1,-30.0,30.0],
    [5, 6, 0.00000, 0.25202, 0.0000, 70, 70, 70, 0.932, 0.0, 1,-30.0,30.0],
    [6,11, 0.09498, 0.19890, 0.0000, 20, 20, 20, 0.000, 0.0, 1,-30.0,30.0],
    [6,12, 0.12291, 0.25581, 0.0000, 20, 20, 20, 0.000, 0.0, 1,-30.0,30.0],
    [6,13, 0.06615, 0.13027, 0.0000, 30, 30, 30, 0.000, 0.0, 1,-30.0,30.0],
    [7, 8, 0.00000, 0.17615, 0.0000, 30, 30, 30, 0.000, 0.0, 1,-30.0,30.0],
    [7, 9, 0.00000, 0.11001, 0.0000, 40, 40, 40, 0.000, 0.0, 1,-30.0,30.0],
    [9,10, 0.03181, 0.08450, 0.0000, 20, 20, 20, 0.000, 0.0, 1,-30.0,30.0],
    [9,14, 0.12711, 0.27038, 0.0000, 20, 20, 20, 0.000, 0.0, 1,-30.0,30.0],
    [10,11, 0.08205, 0.19207, 0.0000, 20, 20, 20, 0.000, 0.0, 1,-30.0,30.0],
    [12,13, 0.22092, 0.19988, 0.0000, 20, 20, 20, 0.000, 0.0, 1,-30.0,30.0],
    [13,14, 0.17093, 0.34802, 0.0000, 20, 20, 20, 0.000, 0.0, 1,-30.0,30.0]
])

## generator cost data
# model, startup, shutdown, n, c(n-1), ..., c0
# model, coste arranque, coste parada, numero de terminos, a, b, c
ppc["gencost"] = array([
    [2, 0.0, 0.0, 3, 0.000000, 7.920951, 0.000000],
    [2, 0.0, 0.0, 3, 0.000000, 23.269494, 0.000000],
    [2, 0.0, 0.0, 3, 0.000000, 0.000000, 0.000000],
    [2, 0.0, 0.0, 3, 0.000000, 0.000000, 0.000000],
    [2, 0.0, 0.0, 3, 0.000000, 0.000000, 0.000000]
])

return ppc

```

8.3 Caso de 30 nudos

```

from numpy import array

def case31erik():
    """Power flow data for IEEE 30 bus system adapted to match PF_IEEE30.txt."""
    ppc = {"version": '2'}

    ## system MVA base
    ppc["baseMVA"] = 100

```

```

## bus data
# bus_i, type, Pd, Qd, Gs, Bs, area, Vm, Th0, baseKV, zone, Vmax, Vmin
ppc["bus"] = array([
    [1, 3, 0.00, 0.00, 0.00, 0.00, 1, 1.06, 0.0, 1, 1.0, 1.05, 0.95],
    [2, 2, 21.7, 12.7, 0.00, 0.00, 1, 1.043, -5.48, 1.0, 1.0, 1.05, 0.95],
    [3, 0, 2.4, 1.2, 0.00, 0.00, 1, 1.021, -7.96, 1.0, 1.0, 1.05, 0.95],
    [4, 0, 7.6, 1.6, 0.00, 0.00, 1, 1.012, -9.62, 1.0, 1.0, 1.05, 0.95],
    [5, 2, 94.2, 19, 0.00, 0.00, 1, 1.01, -14.37, 1.0, 1.0, 1.05, 0.95],
    [6, 0, 0.00, 0.00, 0.00, 0.00, 1, 1.01, -11.34, 1.0, 1.0, 1.05, 0.95],
    [7, 0, 22.8, 10.9, 0.00, 0.00, 1, 1.002, -13.12, 1.0, 1.0, 1.05, 0.95],
    [8, 2, 30, 30, 0.00, 0.00, 1, 1.01, -12.10, 1.0, 1.0, 1.05, 0.95],
    [9, 0, 0.00, 0.00, 0.00, 0.00, 1, 1.051, -14.38, 1.0, 1.0, 1.05, 0.95],
    [10, 0, 5.8, 2, 0.00, 0.19, 1, 1.045, -15.97, 1.0, 1.0, 1.05, 0.95],
    [11, 2, 0.00, 0.00, 0.00, 0.00, 1, 1.082, -14.39, 1.0, 1.0, 1.05, 0.95],
    [12, 0, 11.2, 7.5, 0.00, 0.00, 1, 1.057, -15.24, 1.0, 1.0, 1.05, 0.95],
    [13, 2, 0.00, 0.00, 0.00, 0.00, 1, 1.071, -15.24, 1.0, 1.0, 1.05, 0.95],
    [14, 0, 6.2, 1.6, 0.00, 0.00, 1, 1.042, -16.13, 1.0, 1.0, 1.05, 0.95],
    [15, 0, 8.2, 2.5, 0.00, 0.00, 1, 1.038, -16.22, 1.0, 1.0, 1.05, 0.95],
    [16, 0, 3.5, 1.8, 0.00, 0.00, 1, 1.045, -15.83, 1.0, 1.0, 1.05, 0.95],
    [17, 0, 9, 5.8, 0.00, 0.00, 1, 1.04, -16.14, 1.0, 1.0, 1.05, 0.95],
    [18, 0, 3.2, 0.9, 0.00, 0.00, 1, 1.028, -16.82, 1.0, 1.0, 1.05, 0.95],
    [19, 0, 9.5, 3.4, 0.00, 0.00, 1, 1.026, -17.00, 1.0, 1.0, 1.05, 0.95],
    [20, 0, 2.2, 0.7, 0.00, 0.00, 1, 1.03, -16.80, 1.0, 1.0, 1.05, 0.95],
    [21, 0, 17.5, 11.2, 0.00, 0.00, 1, 1.033, -16.42, 1.0, 1.0, 1.05, 0.95],
    [22, 0, 0.00, 0.00, 0.00, 0.00, 1, 1.033, -16.41, 1.0, 1.0, 1.05, 0.95],
    [23, 0, 3.2, 1.6, 0.00, 0.00, 1, 1.027, -16.61, 1.0, 1.0, 1.05, 0.95],
    [24, 0, 8.7, 6.7, 0.00, 0.04, 1, 1.021, -16.78, 1.0, 1.0, 1.05, 0.95],
    [25, 0, 0.00, 0.00, 0.00, 0.00, 1, 1.017, -16.35, 1.0, 1.0, 1.05, 0.95],
    [26, 0, 3.5, 2.3, 0.00, 0.00, 1, 1.00, -16.77, 1.0, 1.0, 1.05, 0.95],
    [27, 0, 0.00, 0.00, 0.00, 0.00, 1, 1.023, -15.82, 1.0, 1.0, 1.05, 0.95],
    [28, 0, 0.00, 0.00, 0.00, 0.00, 1, 1.007, -11.97, 1.0, 1.0, 1.05, 0.95],
    [29, 0, 2.4, 0.9, 0.00, 0.00, 1, 1.003, -17.06, 1.0, 1.0, 1.05, 0.95],
    [30, 0, 10.6, 1.9, 0.00, 0.00, 1, 0.992, -17.94, 1.0, 1.0, 1.05, 0.95]
])

## generator data
## bus, Pg, Qg 115 y 50 y 37 y 37 y 16 y 11, Qmax, Qmin, Vg, mBase, status, Pmax,
Pmin, Pc1,
ppc["gen"] = array([
    [1, 125, 115, 250, -20, 1.06, 1.06, 1, 200, 50],
    [2, 50, 40, 100, -20, 1.043, 1.025, 1, 80, 20],
    [5, 32.5, 32.5, 80, -15, 1.0, 1, 1, 50, 15],
    [8, 22.5, 22.5, 60, -15, 1.0, 1, 1, 35, 10.0],
    [11, 20, 20, 50, -10, 1.025, 1, 1, 30, 10],
    [13, 26, 22.5, 60, -15, 1.025, 1, 1, 40, 12],
    [15, 0, 0, 0, 0, 1.025, 0, 0, 0, 0]
])

```

```

## branch data
# fbus tbus r x b total rateA rateB rateC ratio angle status angmin angmax
ppc["branch"] = array([
[1, 2, 0.0192, 0.0575, 0.0528, 220.0, 220.0, 220.0, 0.000, 0.0, 1, -360, 360],
[1, 3, 0.0452, 0.1652, 0.0408, 100.0, 100.0, 100.0, 0.000, 0.0, 1, -360, 360],
[2, 4, 0.0570, 0.1737, 0.0368, 55.0, 55.0, 55.0, 0.000, 0.0, 1, -360, 360],
[3, 4, 0.0132, 0.0379, 0.0084, 95.0, 95.0, 95.0, 0.000, 0.0, 1, -360, 360],
[2, 5, 0.0472, 0.1983, 0.0418, 100.0, 100.0, 100.0, 0.000, 0.0, 1, -360, 360],
[2, 6, 0.0581, 0.1763, 0.0364, 75.0, 75.0, 75.0, 0.000, 0.0, 1, -360, 360],
[4, 6, 0.0119, 0.0414, 0.0090, 90.0, 90.0, 90.0, 0.000, 0.0, 1, -360, 360],
[5, 7, 0.0460, 0.1160, 0.0204, 25.0, 25.0, 25.0, 0.000, 0.0, 1, -360, 360],
[6, 7, 0.0267, 0.0820, 0.0170, 45.0, 45.0, 45.0, 0.000, 0.0, 1, -360, 360],
[6, 8, 0.0120, 0.0420, 0.0090, 40.0, 40.0, 40.0, 0.000, 0.0, 1, -360, 360],
[6, 9, 0.0000, 0.2080, 0.0000, 35.0, 35.0, 35.0, 0.978, 0.0, 1, -360, 360],
[6, 10, 0.0000, 0.5560, 0.0000, 20.0, 20.0, 20.0, 0.969, 0.0, 1, -360, 360],
[9, 11, 0.0000, 0.2080, 0.0000, 20.0, 20.0, 20.0, 0.000, 0.0, 1, -360, 360],
[9, 10, 0.0000, 0.1100, 0.0000, 35.0, 35.0, 35.0, 0.000, 0.0, 1, -360, 360],
[4, 12, 0.0000, 0.2560, 0.0000, 60.0, 60.0, 60.0, 0.932, 0.0, 1, -360, 360],
[12, 13, 0.0000, 0.1400, 0.0000, 15.0, 15.0, 15.0, 0.000, 0.0, 1, -360, 360],
[12, 14, 0.1231, 0.2559, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[12, 15, 0.0662, 0.1304, 0.0000, 25.0, 25.0, 25.0, 0.000, 0.0, 1, -360, 360],
[12, 16, 0.0945, 0.1987, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[14, 15, 0.2210, 0.1997, 0.0000, 5.0, 5.0, 5.0, 0.000, 0.0, 1, -360, 360],
[16, 17, 0.0524, 0.1923, 0.0000, 5.0, 5.0, 5.0, 0.000, 0.0, 1, -360, 360],
[15, 18, 0.1073, 0.2185, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[18, 19, 0.0639, 0.1292, 0.0000, 5.0, 5.0, 5.0, 0.000, 0.0, 1, -360, 360],
[19, 20, 0.0340, 0.0680, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[10, 20, 0.0936, 0.2090, 0.0000, 15.0, 15.0, 15.0, 0.000, 0.0, 1, -360, 360],
[10, 17, 0.0324, 0.0845, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[10, 21, 0.0348, 0.0749, 0.0000, 25.0, 25.0, 25.0, 0.000, 0.0, 1, -360, 360],
[10, 22, 0.0727, 0.1499, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[21, 22, 0.0116, 0.0236, 0.0000, 5.0, 5.0, 5.0, 0.000, 0.0, 1, -360, 360],
[15, 23, 0.1000, 0.2020, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[22, 24, 0.1150, 0.1790, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[23, 24, 0.1320, 0.2700, 0.0000, 5.0, 5.0, 5.0, 0.000, 0.0, 1, -360, 360],
[24, 25, 0.1885, 0.3292, 0.0000, 5.0, 5.0, 5.0, 0.000, 0.0, 1, -360, 360],
[25, 26, 0.2544, 0.3800, 0.0000, 5.0, 5.0, 5.0, 0.000, 0.0, 1, -360, 360],
[25, 27, 0.1093, 0.2087, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[28, 27, 0.0000, 0.3960, 0.0000, 25.0, 25.0, 25.0, 0.968, 0.0, 1, -360, 360],
[27, 29, 0.2198, 0.4153, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[27, 30, 0.3202, 0.6027, 0.0000, 10.0, 10.0, 10.0, 0.000, 0.0, 1, -360, 360],
[29, 30, 0.2399, 0.4533, 0.0000, 5.0, 5.0, 5.0, 0.000, 0.0, 1, -360, 360],
[8, 28, 0.0636, 0.2000, 0.2140, 5.0, 5.0, 5.0, 0.000, 0.0, 1, -360, 360],
[6, 28, 0.0169, 0.0599, 0.0650, 25.0, 25.0, 25.0, 0.000, 0.0, 1, -360, 360]
])

```

```

## area data
ppc["areas"] = array([

```

```

        [1, 8],
        [2, 23],
        [3, 26]
    ])

    ## generator cost data
    ppc["gencost"] = array([
        [2, 0, 0, 3, 0.00375, 2, 0],
        [2, 0, 0, 3, 0.0175, 1.75, 0],
        [2, 0, 0, 3, 0.0625, 1, 0],
        [2, 0, 0, 3, 0.00834, 3.25, 0],
        [2, 0, 0, 3, 0.025, 3, 0],
        [2, 0, 0, 3, 0.025, 3, 0],
        [2, 0, 0, 3, 0, 0, 0]
    ])

    return ppc

```

9. Anexo III: Códigos de Python

9.1 Script del flujo de potencia en corriente alterna, ACPF

```
from pyomo.environ import *
from pyomo.environ import sin as pyosin, sqrt as pyosqrt
from math import pi
from case31erik import case31erik

# -----
# Obtención de datos
# -----

ppc      = case31erik()
baseMVA  = ppc["baseMVA"]
bus      = ppc["bus"]
gen      = ppc["gen"]
branch   = ppc["branch"]

nbus     = len(bus)
nbranch  = len(branch)
Sbase    = baseMVA

# -----
# Estructuración de datos
# -----

Bus = {
    "busnum": list(range(nbus)),
    "bustype": [int(b[1]) for b in bus],
    "Pd":      [b[2]/Sbase for b in bus],
    "Qd":      [b[3]/Sbase for b in bus],
    "gshb":    [b[4] for b in bus],
    "bshb":    [b[5] for b in bus],
    "V0":      [b[7] for b in bus],
    "Th0":     [b[8] for b in bus],
    "Pg0":     [0.0 for _ in range(nbus)],
    "Qg0":     [0.0 for _ in range(nbus)]
}

# Acumular aportaciones de cada generador
for g in gen:
    idx = int(g[0]) - 1
    Bus["Pg0"][idx] += g[1] / Sbase
    Bus["Qg0"][idx] += g[2] / Sbase

Branch = {
    "from": [int(b[0]) - 1 for b in branch],
    "to":   [int(b[1]) - 1 for b in branch],
    "r":    [b[2] for b in branch],
    "x":    [b[3] for b in branch],

```

```

    "c":      [b[4]/2      for b in branch], # b/2
    "a":      [1/b[8] if b[8] != 0 else 1.0 for b in branch],
    "fi":     [b[9]       for b in branch],
    "z2":     [b[2]**2 + b[3]**2 for b in branch],
    "smax":   [b[5]/Sbase   for b in branch]
}

# Listas de adyacencia
in_lines = [[] for _ in range(nbus)]
out_lines = [[] for _ in range(nbus)]
for i in range(nbranch):
    in_lines[ Branch["to"][i] ].append(i)
    out_lines[Branch["from"][i] ].append(i)

# print in_lines and out_lines
print(f'In_lines: {in_lines}')
print(f'Out_lines: {out_lines}')

# -----
# Definición del modelo
# -----
model = ConcreteModel()
bus_idx = range(nbus)
branch_idx = range(nbranch)

# Variables
model.Vsqr = Var(bus_idx, domain=NonNegativeReals) # V_i^2
model.th = Var(bus_idx) #  $\theta_i$ 
model.Pg = Var(bus_idx) # P_gi
model.Qg = Var(bus_idx) # Q_gi
model.P = Var(branch_idx) # P_ij
model.Q = Var(branch_idx) # Q_ij
model.Isqr = Var(branch_idx, domain=NonNegativeReals) # I_ij^2

# Inicialización
for i in bus_idx:
    model.Vsqr[i].value = Bus["V0"][i]**2
    model.th[i].value = Bus["Th0"][i] * pi/180
    model.Pg[i].value = Bus["Pg0"][i]
    model.Qg[i].value = Bus["Qg0"][i]

# Función objetivo: minimizar Pg en slack => *min P_g_slack*
model.obj = Objective(
    expr=sum(model.Pg[i] for i in bus_idx if Bus["bustype"][i] == 3),
    sense=minimize
)

# Balance de potencia activa: *P_gi - P_di + V_i^2·G_sh + Σ_in(P_ij) - Σ_out(P_ij +
R_ij·I_ij^2) = 0*

```

```

model.p_balance = ConstraintList()
for k in bus_idx:
    model.p_balance.add(
        model.Pg[k] - Bus["Pd"][k] + model.Vsqr[k]*Bus["gshb"][k]
        + sum(model.P[i] for i in in_lines[k])
        - sum(model.P[i] + Branch["r"][i]*model.Isqr[i] for i in out_lines[k])
        == 0
    )

# Balance de potencia reactiva:  $*Q_{gi} - Q_{di} + V_i^2 \cdot B_{sh} + \sum_{in}(Q_{ij} + b/2 \cdot V_i^2) - \sum_{out}(Q_{ij} - b/2 \cdot V_i^2 + X_{ij} \cdot I_{ij}^2) = 0$ 
model.q_balance = ConstraintList()
for k in bus_idx:
    model.q_balance.add(
        model.Qg[k] - Bus["Qd"][k] + model.Vsqr[k]*Bus["bshb"][k]
        + sum(model.Q[i] + Branch["c"][i]*model.Vsqr[k] for i in in_lines[k])
        - sum(model.Q[i] - Branch["c"][i]*model.Vsqr[k] +
        Branch["x"][i]*model.Isqr[i] for i in out_lines[k])
        == 0
    )

# Ecuaciones del flujo de rama
model.branch_constraints = ConstraintList()
for i in branch_idx:
    a = Branch["from"][i]
    b = Branch["to"][i]
    # Magnitud de tensión y pérdidas:  $*V_a^2 \cdot a^2 - 2 \cdot (R \cdot P + X \cdot Q) - Z^2 \cdot I^2 - V_b^2 = 0$ 
    model.branch_constraints.add(
        model.Vsqr[a]*Branch["a"][i]**2
        - 2*(Branch["r"][i]*model.P[i] + Branch["x"][i]*model.Q[i])
        - Branch["z2"][i]*model.Isqr[i]
        - model.Vsqr[b] == 0
    )
    # Ecuación de ángulo:  $*a \cdot \sqrt{V_a^2} \cdot \sqrt{V_b^2} \cdot \sin(\theta_a - \theta_b + \varphi) = X \cdot P - R \cdot Q$ 
    model.branch_constraints.add(
        Branch["a"][i]*pyosqrt(model.Vsqr[a])*pyosqrt(model.Vsqr[b])*
        pyosin(model.th[a] - model.th[b] + Branch["fi"][i])
        == Branch["x"][i]*model.P[i] - Branch["r"][i]*model.Q[i]
    )
    # Definición de corriente:  $*V_b^2 \cdot I^2 = P^2 + Q^2$ 
    model.branch_constraints.add(
        model.Vsqr[b]*model.Isqr[i] == model.P[i]**2 + model.Q[i]**2
    )

# Restricción térmica:  $*P^2 + Q^2 \leq S_{max}^2$ 
#model.thermal_constraints = ConstraintList()
#for i in branch_idx:
#    Smax_sq = Branch["smax"][i]**2

```

```

#     model.thermal_constraints.add(
#         model.P[i]**2 + model.Q[i]**2 <= Smax_sq
#     )

# Fijación de variables según tipo de bus
for i in bus_idx:
    t = Bus["bustype"][i]
    if t == 0: # PQ
        model.add_component(f"Pg_fixed_{i}", Constraint(expr=model.Pg[i] ==
Bus["Pg0"][i]))
        model.add_component(f"Qg_fixed_{i}", Constraint(expr=model.Qg[i] ==
Bus["Qg0"][i]))
    elif t == 2: # PV
        model.add_component(f"Pg_fixed_{i}", Constraint(expr=model.Pg[i] ==
Bus["Pg0"][i]))
        model.add_component(f"Vsqr_fixed_{i}", Constraint(expr=model.Vsqr[i] ==
Bus["V0"][i]**2))
    elif t == 3: # Slack
        model.add_component(f"Vsqr_fixed_{i}", Constraint(expr=model.Vsqr[i] ==
Bus["V0"][i]**2))
        model.add_component(f"th_fixed_{i}", Constraint(expr=model.th[i] ==
Bus["Th0"][i]*pi/180))

with open("C:/Users/adria/OneDrive/Documentos/APUNTES/Cuarto -
Boston/TFG/python/PyPower/PYPOWER-master/pypower/ACPF6_modelo.txt", 'w') as f:
    model.pprint(ostream=f)

# -----
# Resolución con IPOPT
# -----
solver = SolverFactory(
    "ipopt",
    executable="C:/Users/adria/OneDrive/Documentos/APUNTES/Cuarto -
Boston/TFG/python/Ipopt-3.14.17-win64-msvs2022-md/bin/ipopt.exe"
)
result = solver.solve(model, tee=False)

# -----
# Impresión de la función objetivo
# -----
print("-----")
print("Status of the Optimization:", result.solver.termination_condition)
print("Objective value:", value(model.obj) * Sbase)

# -----
# Presentación de datos (formato original)
# -----

```

```

print("-----")
print("-----")
print("-" * 97)
print("{:^97}".format("BUS-DATA"))
print("-" * 97)
print("{:>5} {:>5} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>4} {:>5}".format(
    "Bus", "Type", "V0", "Th0", "Pd", "Qd", "Pg0", "Qg0", "bshb", "n0", "scost"))
print("-" * 97)
for i in bus_idx:
    print("{:5d} {:5d} {:8.4f} {:8.2f} {:8.2f} {:8.2f} {:8.2f} {:8.2f} {:8.2f} {:4d}
{:5d}".format(
        i+1,
        Bus["bustype"][i],
        Bus["V0"][i],
        Bus["Th0"][i],
        Bus["Pd"][i]*Sbase,
        Bus["Qd"][i]*Sbase,
        Bus["Pg0"][i]*Sbase,
        Bus["Qg0"][i]*Sbase,
        Bus["bshb"][i]*Sbase,
        0, 80
    ))

print("-" * 105)
print("{:^105}".format("BRANCH-DATA"))
print("-" * 105)
print("{:>6} {:>5} {:>5} {:>9} {:>9} {:>8} {:>9} {:>8} {:>6} {:>7} {:>4}".format(
    "Branch", "From", "To", "r", "x", "g", "b", "bshb", "a", "smax", "n0"))
print("-" * 105)
for i in branch_idx:
    a, b = Branch["from"][i], Branch["to"][i]
    r, x = Branch["r"][i], Branch["x"][i]
    z2 = Branch["z2"][i]
    g = r / z2 if z2 != 0 else 0.0
    b_val = -x / z2 if z2 != 0 else 0.0
    print("{:6d} {:5d} {:5d} {:9.4f} {:9.4f} {:8.4f} {:9.4f} {:8.4f} {:6.2f} {:7.2f}
{:4d}".format(
        i+1, a+1, b+1, r, x, g, b_val, Branch["c"][i], Branch["a"][i],
        Branch["smax"][i], 1
    ))

# -----
# Resultados finales
# -----
print("-" * 94)
print("-----")
print("-----")
print("BUS__RESULTS")
print("
")

```

```

print("-----")
print("-----")
print("-" * 94)
print("  Bus  Type    V0      Th0      Pg      Qg      Pd      Qd      gsh
b      bshb")
print("-" * 94)
for i in bus_idx:
    print("{:5d}{:5d}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}"
".format(
        i+1,
        Bus["bustype"][i],
        sqrt(value(model.Vsqr[i])),
        value(model.th[i])*180/pi,
        value(model.Pg[i])*Sbase,
        value(model.Qg[i])*Sbase,
        Bus["Pd"][i]*Sbase,
        Bus["Qd"][i]*Sbase,
        Sbase*Bus["gshb"][i]*value(model.Vsqr[i]),
        Sbase*Bus["bshb"][i]*value(model.Vsqr[i])
    ))
print("-" * 94)
print("TOTAL{:35.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}".format(
    Sbase*sum(value(model.Pg[i]) for i in bus_idx),
    Sbase*sum(value(model.Qg[i]) for i in bus_idx),
    Sbase*sum(Bus["Pd"][i] for i in bus_idx),
    Sbase*sum(Bus["Qd"][i] for i in bus_idx),
    Sbase*sum(Bus["gshb"][i]*value(model.Vsqr[i]) for i in bus_idx),
    Sbase*sum(Bus["bshb"][i]*value(model.Vsqr[i]) for i in bus_idx)
))
print("-----")
print("-----")
print("                                BRANCH__RESULTS
")
print("-----")
print("-----")
print(" Branch
From To Pij[MW]  Pji[MW]  Qij[MW]  Qji[MW]  Pls[MW]  Qls[MVAr]  SLoad")
print("-----")
print("-----")
for i in branch_idx:
    a, b = Branch["from"][i], Branch["to"][i]
    Vi, Vj = sqrt(value(model.Vsqr[a])), sqrt(value(model.Vsqr[b]))
    dth = value(model.th[a]) - value(model.th[b])
    g    = Branch["r"][i]/Branch["z2"][i]
    b_line = -Branch["x"][i]/Branch["z2"][i]
    b_sh   = Branch["c"][i]
    Pij = (Vi**2 * g
        - Vi*Vj*(g*cos(dth) + b_line*sin(dth))) * Sbase
    Pji = -value(model.P[i]) * Sbase

```

```

Qij = (-Vi**2*(b_line + b_sh)
        - Vi*Vj*(g*sin(dth) - b_line*cos(dth))) * Sbase
Qji = (-Vj**2*(b_line + b_sh)
        - Vi*Vj*(g*sin(-dth) - b_line*cos(-dth))) * Sbase
Pls = Branch["r"][i] * value(model.Isqr[i]) * Sbase
Qls = Qij + Qji
Sload = max(Pij**2 + Qij**2, Pji**2 + Qji**2) / (Branch["smax"][i]*Sbase)**2 *
100
print("{:5d}{:6d}{:5d}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.2f}".f
ormat(
    i+1, a+1, b+1, Pij, Pji, Qij, Qji, Pls, Qls, Sload
))
print("-----")
print("-----")

```

9.2 Script del flujo de potencia óptimo en corriente alterna, ACOPF

```

from pyomo.environ import *
from pyomo.environ import sin as pyosin, sqrt as pyosqrt
from math import pi
from case142 import case142

# -----
# Obtención de datos
# -----
ppc = case142() # Cargar el caso de estudio
baseMVA = ppc["baseMVA"]
bus = ppc["bus"]
gen = ppc["gen"]
branch = ppc["branch"]
gencost = ppc.get("gencost", [])

nbus = len(bus)
nbranch = len(branch)
Sbase = baseMVA

# -----
# Parámetros de cada bus
# -----
Bus = {
    "bustype": [int(b[1]) for b in bus],
    "Pd": [b[2]/Sbase for b in bus],
    "Qd": [b[3]/Sbase for b in bus],
    "gsh": [b[4] for b in bus],
    "bsh": [b[5] for b in bus],
    "b": [b[5] for b in bus], # para balance reactivo
    "V0": [b[7] for b in bus],

```

```

        "Th0":      [b[8]          for b in bus],
    }
# Inicialización de Pg0 y Qg0 para impresión
Bus["Pg0"] = [0.0]*nbus
Bus["Qg0"] = [0.0]*nbus
for g in gen:
    idx = int(g[0]) - 1
    Bus["Pg0"][idx] += g[1]/Sbase
    Bus["Qg0"][idx] += g[2]/Sbase
# Alias para impresión de shunts
Bus["gshb"] = Bus["gsh"]
Bus["bshb"] = Bus["bsh"]

# -----
# Límites de generación activa y reactiva
# -----
Pg_min = [0.0]*nbus; Pg_max = [0.0]*nbus
Qg_min = [0.0]*nbus; Qg_max = [0.0]*nbus
for g in gen:
    i = int(g[0]) - 1
    Pg_max[i] = g[8]/Sbase
    Pg_min[i] = g[9]/Sbase
    Qg_max[i] = g[3]/Sbase
    Qg_min[i] = g[4]/Sbase

# -----
# Coeficientes de coste de generación
# -----
a = [0.0]*nbus; b = [0.0]*nbus; c = [0.0]*nbus
for j, cost in enumerate(gencost):
    if int(cost[0]) == 2:
        idx = int(gen[j][0]) - 1
        deg = int(cost[3])
        if deg == 3:
            a[idx], b[idx], c[idx] = cost[4], cost[5], cost[6]
        elif deg == 2:
            b[idx], c[idx] = cost[4], cost[5]

# -----
# Parámetros de líneas y sus límites
# -----
Branch = {
    "f":      [int(br[0]) - 1 for br in branch],
    "t":      [int(br[1]) - 1 for br in branch],
    "r":      [br[2]          for br in branch],
    "x":      [br[3]          for br in branch],
    "c":      [br[4]/2        for br in branch],
    "a":      [1/br[8] if br[8] else 1.0 for br in branch],
    # "a":      [br[8] if br[8] else 1.0 for br in branch],

```

```

    "ang": [br[9]          for br in branch],
    "z2":  [br[2]**2+br[3]**2 for br in branch],
    "smax": [br[5]/Sbase   for br in branch],
}

# print a from branch
print("Branch a values:", Branch["a"])

# -----
# Listas de adyacencia para balances
# -----
en = [[] for _ in range(nbus)]
ex = [[] for _ in range(nbus)]
for k in range(nbranch):
    en[Branch["t"][k]].append(k)
    ex[Branch["f"][k]].append(k)

# -----
# Definición del modelo OPF en Pyomo
# -----
model = ConcreteModel()
bus_idx = range(nbus)
branch_idx = range(nbranch)
model.Vsqr = Var(bus_idx, bounds=lambda m,i: (0.95**2, 1.05**2))
model.PENS = Var(bus_idx, bounds=lambda m,i: (0.0, 1.0)) # Energía no suministrada
model.QENS = Var(bus_idx, bounds=lambda m,i: (0.0, 1.0)) # Energía no suministrada
model.th = Var(bus_idx, bounds=(-60*pi/180, 60*pi/180))
model.Pg = Var(bus_idx, bounds=lambda m,i: (Pg_min[i], Pg_max[i]))
model.Qg = Var(bus_idx, bounds=lambda m,i: (Qg_min[i], Qg_max[i]))
model.P = Var(branch_idx)
model.Q = Var(branch_idx)
model.Isqr = Var(branch_idx, domain=NonNegativeReals)

# Inicialización de variables
for i in bus_idx:
    model.PENS[i].value = 0.0 # Energía no suministrada
    model.QENS[i].value = 0.0 # Energía no suministrada
    model.Vsqr[i].value = Bus["V0"][i]**2
    model.th[i].value = Bus["Th0"][i] * pi/180
    model.Pg[i].value = Pg_min[i]
    model.Qg[i].value = Qg_min[i]

# Función objetivo: coste de generación
model.obj = Objective(
    expr=sum(a[i]*model.Pg[i]**2 + b[i]*model.Pg[i] + c[i] for i in bus_idx) +
    1e5*sum(model.PENS[i] + model.QENS[i] for i in bus_idx),
    sense=minimize
)

```

```

# Balance de potencia activa y reactiva
model.p_balance = ConstraintList()
model.q_balance = ConstraintList()
for i in bus_idx:
    model.p_balance.add(
        model.Pg[i] - Bus["Pd"][i] * (1-model.PENS[i]) + model.Vsqr[i]*Bus["gsh"][i]
        + sum(model.P[k] for k in en[i]) - sum(model.P[k] +
Branch["r"][k]*model.Isqr[k] for k in ex[i])
        == 0
    )
    model.q_balance.add(
        model.Qg[i] - Bus["Qd"][i] * (1-model.QENS[i]) + model.Vsqr[i]*Bus["bsh"][i]
        + sum(model.Q[k] + Branch["c"][k]*model.Vsqr[i] for k in en[i]) -
sum(model.Q[k] - Branch["c"][k]*model.Vsqr[i] + Branch["x"][k]*model.Isqr[k] for k in
ex[i])
        == 0
    )

# Ecuaciones de flujo de rama
model.branch_eqs = ConstraintList()
for k in branch_idx:
    fbus, tbus = Branch["f"][k], Branch["t"][k]
    model.branch_eqs.add(
        model.Vsqr[fbus]*Branch["a"][k]**2
        - 2*(Branch["r"][k]*model.P[k] + Branch["x"][k]*model.Q[k])
        - Branch["z2"][k]*model.Isqr[k]
        - model.Vsqr[tbus] == 0
    )
    ##convertirlo tbn en desigualdad, y añadir ciclos
    model.branch_eqs.add(
        Branch["a"][k]*pyosqrt(model.Vsqr[fbus])*pyosqrt(model.Vsqr[tbus])*
        pyosin(model.th[fbus] - model.th[tbus] + Branch["ang"][k])
        == Branch["x"][k]*model.P[k] - Branch["r"][k]*model.Q[k]
    )
    ##producto de variables, cambiarlo a desigualdad con la relajación
    model.branch_eqs.add(
        model.Vsqr[tbus]*model.Isqr[k] == model.P[k]**2 + model.Q[k]**2
    )

# Restricción térmica de líneas
model.therm = Constraint(branch_idx, rule=lambda m,k: m.P[k]**2 + m.Q[k]**2 <=
Branch["smax"][k]**2)

# Fijación de variables según tipo de bus
for i in bus_idx:
    t = Bus["bustype"][i]
    if t == 3: # Slack
        model.add_component(f"th_fixed_{i}", Constraint(expr=model.th[i] ==
Bus["Th0"][i]*pi/180))

```

```

# -----
# Resolución con IPOPT
# -----
solver = SolverFactory(
    "ipopt",
    executable="C:/Users/adria/OneDrive/Documentos/APUNTES/Cuarto -
Boston/TFG/python/Ipopt-3.14.17-win64-msvs2022-md/bin/ipopt.exe"
)
result = solver.solve(model, tee=True)

# -----
# Presentación de datos (formato original)
# -----
print("-"*97)
print("{:^97}".format("BUS-DATA"))
print("-"*97)
print("{:>5} {:>5} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>4} {:>5}".format(
    "Bus", "Type", "V0", "Th0", "Pd", "Qd", "Pg0", "Qg0", "bshb", "n0", "scost"
))
print("-"*97)
for i in bus_idx:
    print("{:5d} {:5d} {:8.4f} {:8.2f} {:8.2f} {:8.2f} {:8.2f} {:8.2f} {:8.2f} {:4d}
{:5d}".format(
        i+1,
        Bus["bustype"][i],
        Bus["V0"][i],
        Bus["Th0"][i],
        Bus["Pd"][i]*Sbase,
        Bus["Qd"][i]*Sbase,
        Bus["Pg0"][i]*Sbase,
        Bus["Qg0"][i]*Sbase,
        Bus["bshb"][i]*Sbase,
        0, 80
    ))

print("-"*105)
print("{:^105}".format("BRANCH-DATA"))
print("-"*105)
print("{:>6} {:>5} {:>5} {:>9} {:>9} {:>8} {:>9} {:>8} {:>6} {:>7} {:>4}".format(
    "Branch", "From", "To", "r", "x", "g", "b", "bshb", "a", "smax", "n0"
))
print("-"*105)
for k in branch_idx:
    fbus, tbus = Branch["f"][k], Branch["t"][k]
    r, x = Branch["r"][k], Branch["x"][k]
    z2 = Branch["z2"][k]
    g_val = r/z2 if z2 else 0.0
    b_val = -x/z2 if z2 else 0.0

```

```

        print("{:6d} {:5d} {:5d} {:9.4f} {:9.4f} {:8.4f} {:9.4f} {:8.4f} {:6.2f} {:7.2f}
{:4d}".format(
            k+1, fbus+1, tbus+1, r, x, g_val, b_val, Branch["c"][k], Branch["a"][k],
            Branch["smax"][k], 1
        ))

# cgen data related to the generators and costs
print("-"*105)
print("{:^105}".format("GENERATOR-DATA"))
print("-"*105)
print("{:>5} {:>5} {:>10} {:>10} {:>10} {:>10} {:>10} {:>10}".format(
    "Gen", "Bus", "Pg0[MW]", "Qg0[MVAr]", "Pg_min[MW]", "Pg_max[MW]", "Qg_min[MVAr]", "Qg_ma
x[MVAr]"
))
for j, g in enumerate(gen):
    i = int(g[0]) - 1
    print("{:5d} {:5d} {:10.4f} {:10.4f} {:12.4f} {:12.4f} {:14.4f} {:14.4f}".format(
        j+1,          # número de generador
        i+1,          # bus al que pertenece
        Bus["Pg0"][i] * Sbase,
        Bus["Qg0"][i] * Sbase,
        Pg_min[i]     * Sbase,
        Pg_max[i]     * Sbase,
        Qg_min[i]     * Sbase,
        Qg_max[i]     * Sbase
    ))
print("-"*105)

# -----
# Resultados
# -----
print("-----")
print("-----")
print("Status of the Optimization:", result.solver.termination_condition)
print("Objective value:", value(model.obj) * Sbase)
print("-" * 94)

print("-----")
print("{:^86}".format("GENERATOR_RESULTS"))
print("-----")
print("  Gen  Bus   Pg[MW]   Qg[MVAr]   Pg0[MW]   Qg0[MVAr]   Pg_min[MW]   Pg_max[M
W]   Qg_min[MVAr]   Qg_max[MVAr]")
print("-----")
for j, g in enumerate(gen):

```

```

i = int(g[0]) - 1
print("{:5d}{:5d}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:12.4f}{:12.4f}{:14.4f}{:14.4f}"
".format(
    j+1,          # número de generador
    i+1,          # bus al que pertenece
    value(model.Pg[i]) * Sbase,
    value(model.Qg[i]) * Sbase,
    Bus["Pg0"][i] * Sbase,
    Bus["Qg0"][i] * Sbase,
    Pg_min[i] * Sbase,
    Pg_max[i] * Sbase,
    Qg_min[i] * Sbase,
    Qg_max[i] * Sbase
))
print("-----")
print("-----")
print("-----")
print("BUS__RESULTS")
print("-----")
print("-----")
print("-" * 94)
print("  Bus  Type   V0      Th0      Pg      Qg      Pd      Qd      gsh
b      bshb")
print("-" * 94)
for i in bus_idx:
    print("{:5d}{:5d}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}"
".format(
    i+1,
    Bus["bustype"][i],
    sqrt(value(model.Vsqr[i])),
    value(model.th[i])*180/pi,
    value(model.Pg[i])*Sbase,
    value(model.Qg[i])*Sbase,
    Bus["Pd"][i]*Sbase,
    Bus["Qd"][i]*Sbase,
    Sbase*Bus["gshb"][i]*value(model.Vsqr[i]),
    Sbase*Bus["bshb"][i]*value(model.Vsqr[i])
))
print("-" * 94)
print("TOTAL{:35.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}".format(
    Sbase*sum(value(model.Pg[i]) for i in bus_idx),
    Sbase*sum(value(model.Qg[i]) for i in bus_idx),
    Sbase*sum(Bus["Pd"][i] for i in bus_idx),
    Sbase*sum(Bus["Qd"][i] for i in bus_idx),
    Sbase*sum(Bus["gshb"][i]*value(model.Vsqr[i]) for i in bus_idx),
    Sbase*sum(Bus["bshb"][i]*value(model.Vsqr[i]) for i in bus_idx)

```

```

))

print("-"*94)
print("                                BRANCH
RESULTS                                ")
print("-"*94)
print(" Branch From To Pij[MW]   Pji[MW]   Qij[MVAr]   Qji[MVAr]   Pls[MW]   Qls[MVAr]
SLoad[%]")
print("-"*94)
for k in branch_idx:
    fbus, tbus = Branch["f"][k], Branch["t"][k]
    Vi = sqrt(value(model.Vsqr[fbus])); Vj = sqrt(value(model.Vsqr[tbus]))
    dth = value(model.th[fbus]) - value(model.th[tbus])
    g_line = Branch["r"][k]/Branch["z2"][k] if Branch["z2"][k] else 0.0
    b_line = -Branch["x"][k]/Branch["z2"][k] if Branch["z2"][k] else 0.0
    b_sh = Branch["c"][k]
    Pij = (Vi**2*g_line - Vi*Vj*(g_line*cos(dth)+b_line*sin(dth)))*Sbase
    Pji = -value(model.P[k])*Sbase
    Qij = (-Vi**2*(b_line+b_sh) - Vi*Vj*(g_line*sin(dth)-b_line*cos(dth)))*Sbase
    Qji = (-Vj**2*(b_line+b_sh) - Vi*Vj*(g_line*sin(-dth)-b_line*cos(dth)))*Sbase
    Pls = Branch["r"][k]*value(model.Isqr[k])*Sbase
    Qls = Qij + Qji
    Sload = max(Pij**2+Qij**2, Pji**2+Qji**2)/(Branch["smax"][k]*Sbase)**2*100
    print("{:5d}{:6d}{:5d}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.2f}".f
ormat(
        k+1, fbus+1, tbus+1, Pij, Pji, Qij, Qji, Pls, Qls, Sload
    ))
print("-"*94)

# print ENS
print("-----")
print("-----")
print("Energía no suministrada por bus (Activa):")
for i in bus_idx:
    print(f"Bus {i+1}: {value(model.PENS[i]) * Sbase:.4f} %")
print("-----")
print("-----")
print("-----")
print("-----")
print("Energía no suministrada por bus (Reactiva):")
for i in bus_idx:
    print(f"Bus {i+1}: {value(model.QENS[i]) * Sbase:.4f} %")
print("-----")
print("-----")

```

9.3 Script del flujo óptimo con relajación SOCP

```

from pyomo.environ import (

```

```

ConcreteModel, Var, Constraint, ConstraintList, Objective, SolverFactory,
NonNegativeReals, Reals, value, minimize
)
import math
import networkx as nx
from case142 import case142

# -----
# 1) Funciones para detectar y procesar ciclos
# -----

def count_cycles(ppc):
    edges = [(int(r[0]), int(r[1])) for r in ppc["branch"]]
    G = nx.Graph()
    G.add_edges_from(edges)
    return nx.cycle_basis(G)

def process_cycles(ciclos):
    ciclos_orientados, vars_ciclos = [], set()
    for ciclo in ciclos:
        terminos = []
        n = len(ciclo)
        for i in range(n):
            a, b = ciclo[i], ciclo[(i+1)%n]
            if b > a:
                var = f"theta_{b}_{a}"
                terminos.append(("+", var))
            else:
                var = f"theta_{a}_{b}"
                terminos.append("-", var)
            vars_ciclos.add(var)
        # construir ecuación simbólica
        eq = ""
        for k, (sign, var) in enumerate(terminos):
            eq += (f"-{var}" if k==0 and sign=="-" else ("+" + var if k==0 and
sign=="+" else f" {sign} {var}"))
        eq += " = 0"
        ciclos_orientados.append((terminos, eq))
    return ciclos_orientados, vars_ciclos

# -----
# 2) Cargar datos MATPOWER
# -----

ppc = case142()
baseMVA = ppc["baseMVA"]
bus_data = ppc["bus"]

```

```

branch_data = ppc["branch"]
gen_data    = ppc["gen"]
gencost     = ppc.get("gencost", [])

```

```

nbus      = len(bus_data)
nbranch   = len(branch_data)
Sbase     = baseMVA

```

```

# -----
# -----

```

```

# 3) Parámetros por bus

```

```

# -----
# -----

```

```

Bus = {
    "bustype": [int(r[1]) for r in bus_data],
    "Pd":      [r[2]/Sbase for r in bus_data],
    "Qd":      [r[3]/Sbase for r in bus_data],
    "gsh":     [r[4]      for r in bus_data],
    "bsh":     [r[5]      for r in bus_data],
    "V0":      [r[7]      for r in bus_data],
    "Th0":     [r[8]      for r in bus_data],
}

```

```

Bus["Pg0"] = [0]*nbus

```

```

Bus["Qg0"] = [0]*nbus

```

```

for g in gen_data:

```

```

    i = int(g[0]) - 1

```

```

    Bus["Pg0"][i] += g[1]/Sbase

```

```

    Bus["Qg0"][i] += g[2]/Sbase

```

```

Bus["gshb"] = Bus["gsh"]

```

```

Bus["bshb"] = Bus["bsh"]

```

```

# -----
# -----

```

```

# 4) Límites de generación

```

```

# -----
# -----

```

```

Pg_min = [0]*nbus; Pg_max = [0]*nbus

```

```

Qg_min = [0]*nbus; Qg_max = [0]*nbus

```

```

for g in gen_data:

```

```

    i = int(g[0]) - 1

```

```

    Pg_min[i] = g[9]/Sbase

```

```

    Pg_max[i] = g[8]/Sbase

```

```

    Qg_min[i] = g[4]/Sbase

```

```

    Qg_max[i] = g[3]/Sbase

```

```

# -----
# -----

```

```

# 5) Costes de generación

```

```

# -----
# -----
a=[0]*nbus; b=[0]*nbus; c=[0]*nbus
for idx,cost in enumerate(gencost):
    if int(cost[0])==2:
        i=int(gen_data[idx][0])-1
        deg=int(cost[3])
        if deg==3:
            a[i],b[i],c[i]=cost[4],cost[5],cost[6]
        elif deg==2:
            b[i],c[i]=cost[4],cost[5]

# -----
# -----
# 6) Parámetros de líneas
# -----
# -----
Branch = {
    "f": [int(r[0])-1 for r in branch_data],
    "t": [int(r[1])-1 for r in branch_data],
    "r": [r[2] for r in branch_data],
    "x": [r[3] for r in branch_data],
    "c": [r[4]/2 for r in branch_data],
    "a": [1/r[8] if r[8]!=0 else 1.0 for r in branch_data],
    "ang": [r[9] for r in branch_data],
    "smax": [r[5]/Sbase for r in branch_data],
}
Branch["z2"] = [ri**2+xi**2 for ri,xi in zip(Branch["r"],Branch["x"])]

# Gij y Bij
G = [ri/Branch["z2"][k] for k,ri in enumerate(Branch["r"])]
B = [-xi/Branch["z2"][k] for k,xi in enumerate(Branch["x"])]

# -----
# -----
# 7) Adyacencia
# -----
# -----
en=[] for _ in range(nbus)); ex=[] for _ in range(nbus)
for k in range(nbranch):
    en[Branch["t"][k]].append(k)
    ex[Branch["f"][k]].append(k)

# -----
# -----
# 8) Detección de ciclos + construcción de todas_vars para ThetaVar
# -----
# -----
ppc_cycles={"branch":[(int(r[0]),int(r[1])) for r in branch_data]}

```

```

cycles = count_cycles(ppc_cycles)
ciclos_orientados, vars_ciclos = process_cycles(cycles)

# --- ambas direcciones para TODAS las ramas ---
todas_vars = set()
for k in range(nbranch):
    u = Branch["f"][k] + 1
    v = Branch["t"][k] + 1
    todas_vars.add(f"theta_{u}_{v}")
    todas_vars.add(f"theta_{v}_{u}")
# variables que surgieron de los ciclos
todas_vars.update(vars_ciclos)

# -----
# 9) Construcción del modelo
# -----

model=ConcreteModel()
buses=range(nbus); ramos=range(nbranch)

# Variables de estado
model.Vsqr = Var(buses, bounds=lambda m,i:(0.95**2,1.05**2))
model.th   = Var(buses, bounds=(-math.pi/3, math.pi/3))
model.Pg   = Var(buses, bounds=lambda m,i:(Pg_min[i],Pg_max[i]))
model.Qg   = Var(buses, bounds=lambda m,i:(Qg_min[i],Qg_max[i]))
model.P     = Var(ramos, domain=Reals)
model.Q     = Var(ramos, domain=Reals)
model.Isqr = Var(ramos, domain=NonNegativeReals)

# Auxiliares
model.Mvar   = Var(ramos, domain=Reals)
model.Vprod  = Var(ramos, domain=NonNegativeReals)
model.ThetaVar = Var(sorted(todas_vars), domain=Reals, bounds=(-60.0 * math.pi/180.0,
60.0 * math.pi/180.0))

# Init
for i in buses:
    model.Vsqr[i].value = Bus["V0"][i]**2
    model.th[i].value   = Bus["Th0"][i]*math.pi/180
    model.Pg[i].value   = Pg_min[i]
    model.Qg[i].value   = Qg_min[i]
for k in ramos:
    model.P[k].value     = 0
    model.Q[k].value     = 0
    model.Isqr[k].value  = 0
    model.Mvar[k].value  = 0
    model.Vprod[k].value = 1
for var in model.ThetaVar:

```

```

model.ThetaVar[var].value=0

# Objetivo
model.obj=Objective(
    expr=sum(a[i]*model.Pg[i]**2 + b[i]*model.Pg[i] + c[i] for i in buses),
    sense=minimize
)

# Balance
model.p_balance=ConstraintList(); model.q_balance=ConstraintList()
for i in buses:
    model.p_balance.add(
        model.Pg[i]-Bus["Pd"][i]+Bus["gsh"][i]*model.Vsqr[i]
        + sum(model.P[k] for k in en[i])
        - sum(model.P[k]+Branch["r"][k]*model.Isqr[k] for k in ex[i])
        ==0
    )
    model.q_balance.add(
        model.Qg[i]-Bus["Qd"][i]+Bus["bsh"][i]*model.Vsqr[i]
        + sum(model.Q[k]+Branch["c"][k]*model.Vsqr[i] for k in en[i])
        - sum(model.Q[k]-Branch["c"][k]*model.Vsqr[i]+Branch["x"][k]*model.Isqr[k]
        for k in ex[i])
        ==0
    )

# Ramas y definiciones SOCP + envolventes
model.branch_eqs=ConstraintList()
for k in ramos:
    f,t=Branch["f"][k],Branch["t"][k]
    r_k,x_k=Branch["r"][k],Branch["x"][k]
    z2=Branch["z2"][k]; a_k=Branch["a"][k]

# caída de tensión
model.branch_eqs.add(
    a_k**2*model.Vsqr[f]
    -2*(r_k*model.P[k]+x_k*model.Q[k])
    -z2*model.Isqr[k]
    -model.Vsqr[t]==0
)

# SOCP
model.branch_eqs.add(model.P[k]**2+model.Q[k]**2<=model.Vsqr[t]*model.Isqr[k])

# Mvar (para los ciclos)
model.branch_eqs.add(
    model.Mvar[k]*(G[k]**2+B[k]**2)
    - (B[k]*model.P[k] - G[k]*model.Q[k])==0
)

# Vprod
model.branch_eqs.add(
    model.Vprod[k]**2-model.Vsqr[f]*model.Vsqr[t]==0

```

```

)

# Térmica
model.therm=Constraint(ramos,rule=lambda
m,k:m.P[k]**2+m.Q[k]**2<=Branch["smax"][k]**2)

# Slack
for i in buses:
    if Bus["bustype"][i]==3:
        model.add_component(f"th_fix_{i}",Constraint(expr=model.th[i]==Bus["Th0"][i]*
math.pi/180))

# Ciclos KVL
for var in sorted(todas_vars):
    u,v=map(int,var.split("_")[1:])
    model.add_component(
        f"link_{var}",
        Constraint(expr=model.ThetaVar[var]==model.th[u-1]-model.th[v-1])
    )
for idx,(terminos,_) in enumerate(ciclos_orientados):
    expr=0
    for s,var in terminos:
        expr+= (1 if s=="+" else -1)*model.ThetaVar[var]
    model.add_component(f"cycle_{idx}",Constraint(expr=expr==0))

# Envolvente  $\theta_m=60^\circ$ 
theta_m=60*math.pi/180
cosh, tanh = math.cos(theta_m/2), math.tan(theta_m/2)
half=theta_m/2
for k in ramos:
    u,v=Branch["f"][k]+1,Branch["t"][k]+1
    varname=f"theta_{max(u,v)}_{min(u,v)}"
    ThVar=model.ThetaVar[varname]; Mv, Vp = model.Mvar[k], model.Vprod[k]
    model.add_component(f"env_min_{k}",Constraint(expr=ThVar >= (Mv/Vp)*cosh - tanh +
half))
    model.add_component(f"env_max_{k}",Constraint(expr=ThVar <= (Mv/Vp)*cosh + tanh -
half))

# Resolver
solver = SolverFactory(
    "ipopt",
    executable="C:/Users/adria/OneDrive/Documentos/APUNTES/Cuarto -
Boston/TFG/python/Ipopt-3.14.17-win64-msvs2022-md/bin/ipopt.exe"
)
result = solver.solve(model, tee=False)

# ----- Impresión Final -----

# BUS-DATA

```

```

print("-"*97)
print("{:^97}".format("BUS-DATA"))
print("-"*97)
print("{:>5} {:>5} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>8} {:>4} {:>5}".format(
    "Bus", "Type", "V0", "Th0", "Pd", "Qd", "Pg0", "Qg0", "bshb", "n0", "scost"))
print("-"*97)
for i in buses:
    print("{:5d} {:5d} {:8.4f} {:8.2f} {:8.2f} {:8.2f} {:8.2f} {:8.2f} {:8.2f} {:4d}
{:5d}".format(
        i+1, Bus["bustype"][i], Bus["V0"][i], Bus["Th0"][i],
        Bus["Pd"][i]*Sbase, Bus["Qd"][i]*Sbase,
        Bus["Pg0"][i]*Sbase, Bus["Qg0"][i]*Sbase,
        Bus["bshb"][i]*Sbase, 0, 80))

# BRANCH-DATA
print("-"*105)
print("{:^105}".format("BRANCH-DATA"))
print("-"*105)
print("{:>6} {:>5} {:>5} {:>9} {:>9} {:>8} {:>9} {:>8} {:>6} {:>7} {:>4}".format(
    "Branch", "From", "To", "r", "x", "g", "b", "bshb", "a", "smax", "n0"))
print("-"*105)
for k in ramos:
    f,t=Branch["f"][k],Branch["t"][k]
    r_k,x_k=Branch["r"][k],Branch["x"][k]
    z2=Branch["z2"][k]
    g_val = G[k]; b_val = B[k]
    print("{:6d} {:5d} {:5d} {:9.4f} {:9.4f} {:8.4f} {:9.4f} {:8.4f} {:6.2f} {:7.2f}
{:4d}".format(
        k+1, f+1, t+1,
        r_k, x_k, g_val, b_val,
        Branch["c"][k], Branch["a"][k],
        Branch["smax"][k], 1))

print("-"*94)
print("Status of the Optimization:", result.solver.termination_condition)
print("Objective value: %.6f €" % (value(model.obj)*Sbase))
print("-"*94)

# GENERATOR_RESULTS
print("-----")
print("{:^86}".format("GENERATOR_RESULTS"))
print("-----")
print("  Gen  Bus   Pg[MW]   Qg[MVAr]   Pg0[MW]   Qg0[MVAr]   Pg_min[MW]   Pg_max[M
W]   Qg_min[MVAr]   Qg_max[MVAr]")
print("-----")
print("-----")
for j,g in enumerate(gen_data):

```

```

        bi=int(g[0])-1
        print("{:5d}{:5d}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:12.4f}{:12.4f}{:14.4f}{:14.4f}"
            ".format(
                j+1, bi+1,
                value(model.Pg[bi])*Sbase,
                value(model.Qg[bi])*Sbase,
                Bus["Pg0"][bi]*Sbase,
                Bus["Qg0"][bi]*Sbase,
                Pg_min[bi]*Sbase,
                Pg_max[bi]*Sbase,
                Qg_min[bi]*Sbase,
                Qg_max[bi]*Sbase))
    print("-----")
    print("-----")

# BUS__RESULTS
print("-----")
print("-----")
print("
                                BUS__RESULTS
")
print("-----")
print("-----")
print("-"*94)
print("  Bus   Type   V0      Th0      Pg      Qg      Pd      Qd      gsh
b      bshb")
print("-"*94)
for i in buses:
    Vmag = math.sqrt(value(model.Vsqr[i]))
    Th_deg = value(model.th[i])*180/math.pi
    Pg_v = value(model.Pg[i])*Sbase
    Qg_v = value(model.Qg[i])*Sbase
    Pd_v = Bus["Pd"][i]*Sbase
    Qd_v = Bus["Qd"][i]*Sbase
    gsh_mw = Bus["gshb"][i]*value(model.Vsqr[i])*Sbase
    bsh_mv = Bus["bshb"][i]*value(model.Vsqr[i])*Sbase
    print("{:5d}{:5d}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}"
        ".format(
            i+1, Bus["bustype"][i],
            Vmag, Th_deg, Pg_v, Qg_v, Pd_v, Qd_v, gsh_mw, bsh_mv))
print("-"*94)
print("TOTAL{:35.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}".format(
    Sbase*sum(value(model.Pg[i]) for i in buses),
    Sbase*sum(value(model.Qg[i]) for i in buses),
    Sbase*sum(Bus["Pd"][i] for i in buses),
    Sbase*sum(Bus["Qd"][i] for i in buses),
    Sbase*sum(Bus["gshb"][i]*value(model.Vsqr[i]) for i in buses),
    Sbase*sum(Bus["bshb"][i]*value(model.Vsqr[i]) for i in buses)))

# BRANCH RESULTS

```

```

print("-"*94)
print("
RESULTS
")
print("-"*94)
print(" Branch From To Pij[MW] Pji[MW] Qij[MVAr] Qji[MVAr] Pls[MW] Qls[MVAr]
SLoad[%]")
print("-"*94)
for k in ramos:
    f,t=Branch["f"][k],Branch["t"][k]
    Vi, Vj = math.sqrt(value(model.Vsqr[f])), math.sqrt(value(model.Vsqr[t]))
    dth = value(model.th[f])-value(model.th[t])
    g_line, b_line = G[k], B[k]
    b_sh = Branch["c"][k]
    Pij = (Vi**2*g_line - Vi*Vj*(g_line*math.cos(dth)+b_line*math.sin(dth)))*Sbase
    Qij = (-Vi**2*(b_line+b_sh) - Vi*Vj*(g_line*math.sin(dth)-
b_line*math.cos(dth)))*Sbase
    Pji = -value(model.P[k])*Sbase
    Qji = (-Vj**2*(b_line+b_sh) - Vi*Vj*(g_line*math.sin(-dth)-b_line*math.cos(-
dth)))*Sbase
    Pls = Branch["r"][k]*value(model.Isqr[k])*Sbase
    Qls = Qij+Qji
    Sload = max(Pij**2+Qij**2, Pji**2+Qji**2)/(Branch["smax"][k]*Sbase)**2*100
    print("{:5d}{:6d}{:5d}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.4f}{:10.2f}".f
ormat(
        k+1, f+1, t+1, Pij, Pji, Qij, Qji, Pls, Qls, Sload))
print("-"*94)

# CICLOS FINALES
print("\n" + "-"*80)
print("{:^80}".format("CICLOS FINALES Y DIFERENCIAS ANGULARES"))
print("-"*80)
for var in sorted(todas_vars):
    vr = value(model.ThetaVar[var])
    print(f"{var}: {vr*180/math.pi:+.4f}° ({vr:+.4f} rad)")
print("\nVerificación KVL:")
for idx,(terminos,eq) in enumerate(ciclos_orientados, start=1):
    suma=0
    print(f"\n Ciclo {idx}:")
    for s,v in terminos:
        val = value(model.ThetaVar[v])*180/math.pi
        coef = 1 if s=="+" else -1
        suma += coef*val
    print(f" {s}{v}: {coef*val:+.4f}°")
    print(f" Suma= {suma:.4e}° ({'OK' if abs(suma)<1e-6 else 'ERROR'}) {eq}")
print("-"*80)

# --- COMPROBACIÓN DE ENVOLVENTES  $\theta_{ij}$  ( $\theta_m = 60^\circ$ ) ---
tol = 1e-6
theta_m = 60.0 * math.pi/180.0

```

```

cosh    = math.cos(theta_m/2.0)
tanh    = math.tan(theta_m/2.0)
half    = theta_m/2.0

print("\n" + "-"*80)
print("{:^80}".format("VERIFICACIÓN DE LAS ENVOLVENTES  $\theta_{ij}$ "))
print("-"*80)
print(" #    $\theta_k$  [°]    $L_k$  [°]    $U_k$  [°]   cumple_inf   cumple_sup")
for k in ramos:
    # determinar el nombre de la variable ThetaVar para esta rama
    u = Branch["f"][k] + 1
    v = Branch["t"][k] + 1
    varname = f"theta_{max(u,v)}_{min(u,v)}"
    # valores numéricos
    theta_k = value(model.ThetaVar[varname])
    M_k      = value(model.Mvar[k])
    Vp_k     = value(model.Vprod[k])
    # límites inferior y superior
    L_k = (M_k/Vp_k)*cosh - tanh + half
    U_k = (M_k/Vp_k)*cosh + tanh - half
    # pasar a grados para imprimir
    th_deg = theta_k * 180/math.pi
    L_deg  = L_k      * 180/math.pi
    U_deg  = U_k      * 180/math.pi
    ok_inf = (th_deg + tol) >= L_deg
    ok_sup = (th_deg - tol) <= U_deg
    print(f"{k+1:2d}   {th_deg:8.4f}   {L_deg:8.4f}   {U_deg:8.4f}   {str(ok_inf):>5}
    {str(ok_sup):>5}")
print("-"*80)

```

9.4 Script del cálculo de ciclos

```

import networkx as nx
from case3ierik import case3ierik
from case31erik import case31erik
from case142 import case142
import math

def count_cycles(ppc):
    branches = [(int(b[0]), int(b[1])) for b in ppc["branch"]]
    G = nx.Graph()
    G.add_edges_from(branches)
    return nx.cycle_basis(G)

def process_cycles(ciclos):
    ciclos_orientados = [] # Lista de tuplas (lista_de_terminos, ecuacion_str)
    todas_vars = set()    # Conjunto de var_name (sin signo) que aparecen

```

```

for ciclo in ciclos:
    terminos = [] # Lista de (signo_str, var_name) para este ciclo
    n = len(ciclo)
    for i in range(n):
        a = ciclo[i]
        b = ciclo[(i + 1) % n] # siguiente nodo (y vuelve al primero)
        # Deseamos la diferencia theta_b - theta_a
        if b > a:
            var = f"theta_{b}_{a}"
            # signo positivo porque b>a → (θ_b - θ_a) = theta_{b}_{a}
            terminos.append(( "+", var ))
            todas_vars.add(var)
        else:
            # b < a → (θ_b - θ_a) = -(θ_a - θ_b); definimos var = theta_{a}_{b}
            var = f"theta_{a}_{b}"
            terminos.append(( "-", var ))
            todas_vars.add(var)
    # Construir la ecuación KVL del ciclo como cadena
    ecuacion = ""
    for idx, (signo, var) in enumerate(terminos):
        if idx == 0:
            # Primer término no lleva signo '+' explícito; sólo '-' si aplica
            ecuacion += (f"-{var}" if signo == "-" else var)
        else:
            ecuacion += f" {signo} {var}"
    ecuacion += " = 0"
    ciclos_orientados.append((terminos, ecuacion))

return ciclos_orientados, todas_vars

def extract_bus_angles(ppc):
    angle_map = {}
    for row in ppc["bus"]:
        bus_i = int(row[0])
        # Índice 8 corresponde a Th0 en grados (formato Matpower)
        angle_map[bus_i] = float(row[8])
    return angle_map

def compute_theta_values(todas_vars, angle_map):
    valores_deg = {}
    valores_rad = {}
    for var in todas_vars:
        # var tiene formato "theta_u_v"
        parts = var.split('_')
        # parts = ["theta", "u", "v"]
        u = int(parts[1])
        v = int(parts[2])
        va_u = angle_map[u]
        va_v = angle_map[v]

```

```

        diff_deg = va_u - va_v
        diff_rad = diff_deg * math.pi / 180.0
        valores_deg[var] = diff_deg
        valores_rad[var] = diff_rad
    return valores_deg, valores_rad

if __name__ == "__main__":
    # Lista de casos a procesar: nombre descriptivo y función que genera el ppc
    casos = [
        ("IEEE 3 nudos", case3ierik),
        ("IEEE 30 nudos", case31erik),
        ("IEEE 14 nudos", case142),
    ]

    for nombre, constructor in casos:
        ppc = constructor()
        ciclos = count_cycles(ppc)

        print(f"{nombre}: {len(ciclos)} ciclos detectados\n")
        for idx, ciclo in enumerate(ciclos, start=1):
            ciclo_cerrado = ciclo + [ciclo[0]]
            print(f"    Ciclo {idx}: {ciclo_cerrado}")

        # Procesar cada ciclo para determinar sentido y variables theta
        ciclos_orientados, todas_vars = process_cycles(ciclos)

        # Extraer ángulos de cada bus (en grados)
        angle_map = extract_bus_angles(ppc)

        # Calcular valores de cada variable theta (en grados y en radianes)
        valores_deg, valores_rad = compute_theta_values(todas_vars, angle_map)

        print(f"\nVariables theta definidas (en grados, y su equivalente en radianes)
para {nombre}:")
        for var in sorted(todas_vars):
            print(f"    {var}: {valores_deg[var]:.4f}°    ({valores_rad[var]:.4f} rad)")

        print(f"\nDetalles de diferencias angulares para cada ciclo en {nombre}:")
        for idx, (terminos, ecuacion_str) in enumerate(ciclos_orientados, start=1):
            print(f"    Ciclo {idx}:")
            suma_numerica = 0.0
            # Mostrar cada arista del ciclo con su variable, signo y valor numérico
            # corregido
            for signo, var in terminos:
                valor = valores_deg[var]
                coef = 1 if signo == "+" else -1
                term_value = coef * valor
                suma_numerica += term_value
                signo_text = signo # '+' o '-'

```

```

        # Ahora mostramos el valor real de la contribución: coef * valor
        print(f"    {signo_text}{var} = {term_value:+.4f}°")
    # Mostrar la suma de todos los términos para verificar que sea (aprox.) 0
    print(f"    Suma numérica = {suma_numerica:.4e}° → {'OK' if
abs(suma_numerica) < 1e-6 else 'ERROR'}")
    print(f"    Ecuación simbólica: {ecuacion_str}")

    print("\n" + "-"*80 + "\n")

```

10. Anexo IV: Resultados

10.1 Resultados de Cálculo de Ciclos

IEEE 3 nudos: 1 ciclos detectados

Ciclo 1: [1, 2, 3, 1]

Variables theta definidas (en grados, y su equivalente en radianes) para IEEE 3 nudos:

```

theta_2_1: 7.2600°    (0.1267 rad)
theta_3_1: -17.2700°  (-0.3014 rad)
theta_3_2: -24.5300°  (-0.4281 rad)

```

Detalles de diferencias angulares para cada ciclo en IEEE 3 nudos:

```

Ciclo 1:
+theta_2_1 = +7.2600°
+theta_3_2 = -24.5300°
-theta_3_1 = +17.2700°
Suma numérica = -3.5527e-15° → OK
Ecuación simbólica: theta_2_1 + theta_3_2 - theta_3_1 = 0
-----

```

IEEE 30 nudos: 12 ciclos detectados

```

Ciclo 1: [27, 29, 30, 27]
Ciclo 2: [8, 6, 28, 8]
Ciclo 3: [9, 10, 6, 9]
Ciclo 4: [21, 22, 10, 21]
Ciclo 5: [25, 24, 22, 10, 6, 28, 27, 25]
Ciclo 6: [20, 19, 18, 15, 23, 24, 22, 10, 20]
Ciclo 7: [12, 14, 15, 12]
Ciclo 8: [4, 12, 15, 23, 24, 22, 10, 6, 4]
Ciclo 9: [17, 16, 12, 15, 23, 24, 22, 10, 17]
Ciclo 10: [2, 5, 7, 6, 2]
Ciclo 11: [2, 4, 6, 2]
Ciclo 12: [2, 1, 3, 4, 2]

```

Variables theta definidas (en grados, y su equivalente en radianes) para IEEE 30 nudos:

```

theta_10_6: -4.6300° (-0.0808 rad)
theta_10_9: -1.5900° (-0.0278 rad)
theta_12_4: -5.6200° (-0.0981 rad)
theta_14_12: -0.8900° (-0.0155 rad)
theta_15_12: -0.9800° (-0.0171 rad)
theta_15_14: -0.0900° (-0.0016 rad)
theta_16_12: -0.5900° (-0.0103 rad)
theta_17_10: -0.1700° (-0.0030 rad)
theta_17_16: -0.3100° (-0.0054 rad)
theta_18_15: -0.6000° (-0.0105 rad)
theta_19_18: -0.1800° (-0.0031 rad)
theta_20_10: -0.8300° (-0.0145 rad)
theta_20_19: 0.2000° (0.0035 rad)
theta_21_10: -0.4500° (-0.0079 rad)
theta_22_10: -0.4400° (-0.0077 rad)
theta_22_21: 0.0100° (0.0002 rad)
theta_23_15: -0.3900° (-0.0068 rad)
theta_24_22: -0.3700° (-0.0065 rad)
theta_24_23: -0.1700° (-0.0030 rad)
theta_25_24: 0.4300° (0.0075 rad)
theta_27_25: 0.5300° (0.0093 rad)
theta_28_27: 3.8500° (0.0672 rad)
theta_28_6: -0.6300° (-0.0110 rad)
theta_28_8: 0.1300° (0.0023 rad)
theta_29_27: -1.2400° (-0.0216 rad)
theta_2_1: -5.4800° (-0.0956 rad)
theta_30_27: -2.1200° (-0.0370 rad)
theta_30_29: -0.8800° (-0.0154 rad)
theta_3_1: -7.9600° (-0.1389 rad)
theta_4_2: -4.1400° (-0.0723 rad)
theta_4_3: -1.6600° (-0.0290 rad)
theta_5_2: -8.8900° (-0.1552 rad)
theta_6_2: -5.8600° (-0.1023 rad)
theta_6_4: -1.7200° (-0.0300 rad)
theta_7_5: 1.2500° (0.0218 rad)
theta_7_6: -1.7800° (-0.0311 rad)
theta_8_6: -0.7600° (-0.0133 rad)
theta_9_6: -3.0400° (-0.0531 rad)

```

Detalles de diferencias angulares para cada ciclo en IEEE 30 nudos:

Ciclo 1:

```

+theta_29_27 = -1.2400°
+theta_30_29 = -0.8800°
-theta_30_27 = +2.1200°
Suma numérica = 0.0000e+00° → OK

```

$$\text{Ecuación simbólica: } \theta_{29_27} + \theta_{30_29} - \theta_{30_27} = 0$$

Ciclo 2:

$$-\theta_{8_6} = +0.7600^\circ$$

$$+\theta_{28_6} = -0.6300^\circ$$

$$-\theta_{28_8} = -0.1300^\circ$$

$$\text{Suma numérica} = 0.0000\text{e}+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } -\theta_{8_6} + \theta_{28_6} - \theta_{28_8} = 0$$

Ciclo 3:

$$+\theta_{10_9} = -1.5900^\circ$$

$$-\theta_{10_6} = +4.6300^\circ$$

$$+\theta_{9_6} = -3.0400^\circ$$

$$\text{Suma numérica} = 0.0000\text{e}+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } \theta_{10_9} - \theta_{10_6} + \theta_{9_6} = 0$$

Ciclo 4:

$$+\theta_{22_21} = +0.0100^\circ$$

$$-\theta_{22_10} = +0.4400^\circ$$

$$+\theta_{21_10} = -0.4500^\circ$$

$$\text{Suma numérica} = 0.0000\text{e}+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } \theta_{22_21} - \theta_{22_10} + \theta_{21_10} = 0$$

Ciclo 5:

$$-\theta_{25_24} = -0.4300^\circ$$

$$-\theta_{24_22} = +0.3700^\circ$$

$$-\theta_{22_10} = +0.4400^\circ$$

$$-\theta_{10_6} = +4.6300^\circ$$

$$+\theta_{28_6} = -0.6300^\circ$$

$$-\theta_{28_27} = -3.8500^\circ$$

$$-\theta_{27_25} = -0.5300^\circ$$

$$\text{Suma numérica} = 0.0000\text{e}+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } -\theta_{25_24} - \theta_{24_22} - \theta_{22_10} - \theta_{10_6} + \theta_{28_6} - \theta_{28_27} - \theta_{27_25} = 0$$

Ciclo 6:

$$-\theta_{20_19} = -0.2000^\circ$$

$$-\theta_{19_18} = +0.1800^\circ$$

$$-\theta_{18_15} = +0.6000^\circ$$

$$+\theta_{23_15} = -0.3900^\circ$$

$$+\theta_{24_23} = -0.1700^\circ$$

$$-\theta_{24_22} = +0.3700^\circ$$

$$-\theta_{22_10} = +0.4400^\circ$$

$$+\theta_{20_10} = -0.8300^\circ$$

$$\text{Suma numérica} = 0.0000\text{e}+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } -\theta_{20_19} - \theta_{19_18} - \theta_{18_15} + \theta_{23_15} + \theta_{24_23} - \theta_{24_22} - \theta_{22_10} + \theta_{20_10} = 0$$

Ciclo 7:

$+theta_{14_12} = -0.8900^\circ$
 $+theta_{15_14} = -0.0900^\circ$
 $-theta_{15_12} = +0.9800^\circ$
 Suma numérica = $0.0000e+00^\circ \rightarrow OK$
 Ecuación simbólica: $theta_{14_12} + theta_{15_14} - theta_{15_12} = 0$

Ciclo 8:

$+theta_{12_4} = -5.6200^\circ$
 $+theta_{15_12} = -0.9800^\circ$
 $+theta_{23_15} = -0.3900^\circ$
 $+theta_{24_23} = -0.1700^\circ$
 $-theta_{24_22} = +0.3700^\circ$
 $-theta_{22_10} = +0.4400^\circ$
 $-theta_{10_6} = +4.6300^\circ$
 $-theta_{6_4} = +1.7200^\circ$
 Suma numérica = $0.0000e+00^\circ \rightarrow OK$
 Ecuación simbólica: $theta_{12_4} + theta_{15_12} + theta_{23_15} + theta_{24_23} -$
 $theta_{24_22} - theta_{22_10} - theta_{10_6} - theta_{6_4} = 0$

Ciclo 9:

$-theta_{17_16} = +0.3100^\circ$
 $-theta_{16_12} = +0.5900^\circ$
 $+theta_{15_12} = -0.9800^\circ$
 $+theta_{23_15} = -0.3900^\circ$
 $+theta_{24_23} = -0.1700^\circ$
 $-theta_{24_22} = +0.3700^\circ$
 $-theta_{22_10} = +0.4400^\circ$
 $+theta_{17_10} = -0.1700^\circ$
 Suma numérica = $0.0000e+00^\circ \rightarrow OK$
 Ecuación simbólica: $-theta_{17_16} - theta_{16_12} + theta_{15_12} + theta_{23_15} +$
 $theta_{24_23} - theta_{24_22} - theta_{22_10} + theta_{17_10} = 0$

Ciclo 10:

$+theta_{5_2} = -8.8900^\circ$
 $+theta_{7_5} = +1.2500^\circ$
 $-theta_{7_6} = +1.7800^\circ$
 $-theta_{6_2} = +5.8600^\circ$
 Suma numérica = $0.0000e+00^\circ \rightarrow OK$
 Ecuación simbólica: $theta_{5_2} + theta_{7_5} - theta_{7_6} - theta_{6_2} = 0$

Ciclo 11:

$+theta_{4_2} = -4.1400^\circ$
 $+theta_{6_4} = -1.7200^\circ$
 $-theta_{6_2} = +5.8600^\circ$
 Suma numérica = $0.0000e+00^\circ \rightarrow OK$
 Ecuación simbólica: $theta_{4_2} + theta_{6_4} - theta_{6_2} = 0$

Ciclo 12:

$-theta_{2_1} = +5.4800^\circ$

```

+theta_3_1 = -7.9600°
+theta_4_3 = -1.6600°
-theta_4_2 = +4.1400°
Suma numérica = 0.0000e+00° → OK
Ecuación simbólica: -theta_2_1 + theta_3_1 + theta_4_3 - theta_4_2 = 0

```

IEEE 14 nudos: 7 ciclos detectados

```

Ciclo 1: [6, 12, 13, 6]
Ciclo 2: [9, 10, 11, 6, 13, 14, 9]
Ciclo 3: [2, 4, 5, 2]
Ciclo 4: [9, 4, 5, 6, 13, 14, 9]
Ciclo 5: [9, 7, 4, 9]
Ciclo 6: [2, 3, 4, 2]
Ciclo 7: [1, 2, 5, 1]

```

Variables theta definidas (en grados, y su equivalente en radianes) para IEEE 14 nudos:

```

theta_10_9: -0.1600° (-0.0028 rad)
theta_11_10: 0.3100° (0.0054 rad)
theta_11_6: -0.5700° (-0.0099 rad)
theta_12_6: -0.8500° (-0.0148 rad)
theta_13_12: -0.0900° (-0.0016 rad)
theta_13_6: -0.9400° (-0.0164 rad)
theta_14_13: -0.8800° (-0.0154 rad)
theta_14_9: -1.1000° (-0.0192 rad)
theta_2_1: -4.9800° (-0.0869 rad)
theta_3_2: -7.7400° (-0.1351 rad)
theta_4_2: -5.3500° (-0.0934 rad)
theta_4_3: 2.3900° (0.0417 rad)
theta_5_1: -8.7800° (-0.1532 rad)
theta_5_2: -3.8000° (-0.0663 rad)
theta_5_4: 1.5500° (0.0271 rad)
theta_6_5: -5.4400° (-0.0949 rad)
theta_7_4: -3.0400° (-0.0531 rad)
theta_9_4: -4.6100° (-0.0805 rad)
theta_9_7: -1.5700° (-0.0274 rad)

```

Detalles de diferencias angulares para cada ciclo en IEEE 14 nudos:

```

Ciclo 1:
+theta_12_6 = -0.8500°
+theta_13_12 = -0.0900°
-theta_13_6 = +0.9400°
Suma numérica = 0.0000e+00° → OK
Ecuación simbólica: theta_12_6 + theta_13_12 - theta_13_6 = 0

```

Ciclo 2:

$$+\theta_{10_9} = -0.1600^\circ$$

$$+\theta_{11_{10}} = +0.3100^\circ$$

$$-\theta_{11_6} = +0.5700^\circ$$

$$+\theta_{13_6} = -0.9400^\circ$$

$$+\theta_{14_{13}} = -0.8800^\circ$$

$$-\theta_{14_9} = +1.1000^\circ$$

$$\text{Suma numérica} = 0.0000e+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } \theta_{10_9} + \theta_{11_{10}} - \theta_{11_6} + \theta_{13_6} + \theta_{14_{13}} - \theta_{14_9} = 0$$

Ciclo 3:

$$+\theta_{4_2} = -5.3500^\circ$$

$$+\theta_{5_4} = +1.5500^\circ$$

$$-\theta_{5_2} = +3.8000^\circ$$

$$\text{Suma numérica} = 0.0000e+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } \theta_{4_2} + \theta_{5_4} - \theta_{5_2} = 0$$

Ciclo 4:

$$-\theta_{9_4} = +4.6100^\circ$$

$$+\theta_{5_4} = +1.5500^\circ$$

$$+\theta_{6_5} = -5.4400^\circ$$

$$+\theta_{13_6} = -0.9400^\circ$$

$$+\theta_{14_{13}} = -0.8800^\circ$$

$$-\theta_{14_9} = +1.1000^\circ$$

$$\text{Suma numérica} = 0.0000e+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } -\theta_{9_4} + \theta_{5_4} + \theta_{6_5} + \theta_{13_6} + \theta_{14_{13}} - \theta_{14_9} = 0$$

Ciclo 5:

$$-\theta_{9_7} = +1.5700^\circ$$

$$-\theta_{7_4} = +3.0400^\circ$$

$$+\theta_{9_4} = -4.6100^\circ$$

$$\text{Suma numérica} = 0.0000e+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } -\theta_{9_7} - \theta_{7_4} + \theta_{9_4} = 0$$

Ciclo 6:

$$+\theta_{3_2} = -7.7400^\circ$$

$$+\theta_{4_3} = +2.3900^\circ$$

$$-\theta_{4_2} = +5.3500^\circ$$

$$\text{Suma numérica} = 0.0000e+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } \theta_{3_2} + \theta_{4_3} - \theta_{4_2} = 0$$

Ciclo 7:

$$+\theta_{2_1} = -4.9800^\circ$$

$$+\theta_{5_2} = -3.8000^\circ$$

$$-\theta_{5_1} = +8.7800^\circ$$

$$\text{Suma numérica} = 0.0000e+00^\circ \rightarrow \text{OK}$$

$$\text{Ecuación simbólica: } \theta_{2_1} + \theta_{5_2} - \theta_{5_1} = 0$$

10.2 Resultados relajación SOCP: Caso 14 nudos con $\theta_m = \pm 60$

10.2.1 Relajación

Status of the Optimization: optimal

Objective value: 2261.405207 €

-

GENERATOR__RESULTS

Gen	Bus	Pg[MW]	Qg[MVar]	Pg0[MW]	Qg0[MVar]	Pg_min[MW]	Pg_max[MW]
Qg_min[MVar]	Qg_max[MVar]						

1	1	269.0907	10.0000	232.0000	-17.0000	0.0000	340.0000
0.0000		10.0000					
2	2	5.5846	25.8891	40.0000	42.0000	0.0000	59.0000
30.0000		30.0000					-
3	3	0.0000	29.0921	0.0000	23.0000	0.0000	0.0000
0.0000		40.0000					
4	6	0.0000	17.3574	0.0000	12.0000	0.0000	0.0000
6.0000		24.0000					-
5	8	0.0000	11.6692	0.0000	17.0000	0.0000	0.0000
6.0000		24.0000					-

BUS__RESULTS

Bus	Type	V0	Th0	Pg	Qg	Pd	Qd	gshb
-----	------	----	-----	----	----	----	----	------

1	3	1.0500	-0.0000	269.0907	10.0000	0.0000	0.0000	0.0000
0.0000								
2	2	1.0199	-5.1794	5.5846	25.8891	21.7000	12.7000	0.0000
0.0000								
3	2	0.9887	-11.7325	0.0000	29.0921	94.2000	19.0000	0.0000
0.0000								

4	0	0.9940	-9.6635	0.0000	0.0000	47.8000	-3.9000	0.0000
0.0000								
5	0	0.9970	-8.2638	0.0000	0.0000	7.6000	1.6000	0.0000
0.0000								
6	2	1.0500	-13.1329	0.0000	17.3574	11.2000	7.5000	0.0000
0.0000								
7	0	1.0304	-12.3927	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000								
8	2	1.0500	-12.3589	0.0000	11.6692	0.0000	0.0000	0.0000
0.0000								
9	0	1.0269	-13.7729	0.0000	0.0000	29.5000	16.6000	0.0000
20.0376								
10	0	1.0233	-14.1723	0.0000	0.0000	9.0000	5.8000	0.0000
0.0000								
11	0	1.0329	-13.8406	0.0000	0.0000	3.5000	1.8000	0.0000
0.0000								
12	0	1.0341	-14.0601	0.0000	0.0000	6.1000	1.6000	0.0000
0.0000								
13	0	1.0287	-14.2414	0.0000	0.0000	13.5000	5.8000	0.0000
0.0000								
14	0	1.0092	-15.2364	0.0000	0.0000	14.9000	5.0000	0.0000
0.0000								

TOTAL				274.6753	94.0079	259.0000	73.5000	0.0000
20.0376								

BRANCH RESULTS

Branch	From	To	Pij[MW]	Pji[MW]	Qij[MVar]	Qji[MVar]	Pls[MW]	Qls[MVar]
SLoad[%]								

1	1	2	165.5548	-169.5805	3.7311	5.3459	5.3848	9.0769
99.61								
2	1	5	70.5441	-89.7230	10.0008	-4.7639	4.4024	5.2369
99.67								
3	2	3	59.3842	-77.3340	3.0232	-0.6762	2.9141	2.3470
73.84								
4	2	4	45.5216	-42.9480	-0.0318	0.1019	1.0857	0.0701
42.29								
5	2	5	32.6273	-28.7059	1.7444	-3.4630	0.4773	-1.7186
42.70								
6	3	4	-18.8956	17.0711	4.0915	-4.6858	0.2051	-0.5942
41.53								
7	4	5	-54.1286	64.8066	10.5923	-9.2957	0.5842	1.2966
87.48								

8	4	7	23.3207	-28.6300	-16.7722	18.5188	0.0000	1.7466
46.50								
9	4	9	13.1517	-13.6693	-5.4212	6.5604	0.0000	1.1392
25.54								
10	5	6	35.2588	-46.0223	-19.4499	23.5606	0.0000	4.1107
54.55								
11	6	11	9.0196	-8.5913	4.7854	-4.5973	0.0859	0.1881
26.06								
12	6	12	8.1460	-7.6335	2.6580	-2.4877	0.0754	0.1704
18.36								
13	6	13	19.7408	-18.1947	7.2857	-6.7625	0.2416	0.5232
49.20								
14	7	8	-0.3626	-0.0000	-11.4515	11.6693	0.0000	0.2178
15.13								
15	7	9	23.1685	-28.6300	3.5395	-2.9704	0.0000	0.5691
51.78								
16	9	10	9.0530	-3.9354	1.0210	-0.9545	0.0073	0.0665
20.75								
17	9	14	10.6537	-8.7538	1.8366	-1.5370	0.1029	0.2996
29.22								
18	10	11	-4.5254	5.0913	-3.1359	3.1915	0.0267	0.0556
9.03								
19	12	13	2.1513	-1.5261	0.4226	-0.4137	0.0074	0.0090
1.20								
20	13	14	6.4684	-6.1462	2.6223	-2.4621	0.0746	0.1602
12.18								

CICLOS FINALES Y DIFERENCIAS ANGULARES

theta_10_11: -0.3317° (-0.0058 rad)
theta_10_9: -0.3994° (-0.0070 rad)
theta_11_10: +0.3317° (+0.0058 rad)
theta_11_6: -0.7077° (-0.0124 rad)
theta_12_13: +0.1813° (+0.0032 rad)
theta_12_6: -0.9272° (-0.0162 rad)
theta_13_12: -0.1813° (-0.0032 rad)
theta_13_14: +0.9950° (+0.0174 rad)
theta_13_6: -1.1085° (-0.0193 rad)
theta_14_13: -0.9950° (-0.0174 rad)
theta_14_9: -1.4635° (-0.0255 rad)
theta_1_2: +5.1794° (+0.0904 rad)
theta_1_5: +8.2638° (+0.1442 rad)
theta_2_1: -5.1794° (-0.0904 rad)
theta_2_3: +6.5531° (+0.1144 rad)
theta_2_4: +4.4841° (+0.0783 rad)
theta_2_5: +3.0844° (+0.0538 rad)

theta_3_2: -6.5531° (-0.1144 rad)
 theta_3_4: -2.0690° (-0.0361 rad)
 theta_4_2: -4.4841° (-0.0783 rad)
 theta_4_3: +2.0690° (+0.0361 rad)
 theta_4_5: -1.3997° (-0.0244 rad)
 theta_4_7: +2.7292° (+0.0476 rad)
 theta_4_9: +4.1094° (+0.0717 rad)
 theta_5_1: -8.2638° (-0.1442 rad)
 theta_5_2: -3.0844° (-0.0538 rad)
 theta_5_4: +1.3997° (+0.0244 rad)
 theta_5_6: +4.8690° (+0.0850 rad)
 theta_6_11: +0.7077° (+0.0124 rad)
 theta_6_12: +0.9272° (+0.0162 rad)
 theta_6_13: +1.1085° (+0.0193 rad)
 theta_6_5: -4.8690° (-0.0850 rad)
 theta_7_4: -2.7292° (-0.0476 rad)
 theta_7_8: -0.0338° (-0.0006 rad)
 theta_7_9: +1.3802° (+0.0241 rad)
 theta_8_7: +0.0338° (+0.0006 rad)
 theta_9_10: +0.3994° (+0.0070 rad)
 theta_9_14: +1.4635° (+0.0255 rad)
 theta_9_4: -4.1094° (-0.0717 rad)
 theta_9_7: -1.3802° (-0.0241 rad)

Verificación KVL:

Ciclo 1:

+theta_12_6: -0.9272°

+theta_13_12: -0.1813°

-theta_13_6: +1.1085°

Suma= 0.0000e+00° (OK) +theta_12_6 + theta_13_12 - theta_13_6 = 0

Ciclo 2:

+theta_10_9: -0.3994°

+theta_11_10: +0.3317°

-theta_11_6: +0.7077°

+theta_13_6: -1.1085°

+theta_14_13: -0.9950°

-theta_14_9: +1.4635°

Suma= 0.0000e+00° (OK) +theta_10_9 + theta_11_10 - theta_11_6 + theta_13_6 + theta_14_13 - theta_14_9 = 0

Ciclo 3:

+theta_4_2: -4.4841°

+theta_5_4: +1.3997°

-theta_5_2: +3.0844°

Suma= -8.8818e-16° (OK) +theta_4_2 + theta_5_4 - theta_5_2 = 0

Ciclo 4:

-theta_9_4: +4.1094°
 +theta_5_4: +1.3997°
 +theta_6_5: -4.8690°
 +theta_13_6: -1.1085°
 +theta_14_13: -0.9950°
 -theta_14_9: +1.4635°
 Suma= 1.3323e-15° (OK) -theta_9_4 + theta_5_4 + theta_6_5 + theta_13_6 +
 theta_14_13 - theta_14_9 = 0

Ciclo 5:

-theta_9_7: +1.3802°
 -theta_7_4: +2.7292°
 +theta_9_4: -4.1094°
 Suma= 0.0000e+00° (OK) -theta_9_7 - theta_7_4 + theta_9_4 = 0

Ciclo 6:

+theta_3_2: -6.5531°
 +theta_4_3: +2.0690°
 -theta_4_2: +4.4841°
 Suma= 8.8818e-16° (OK) +theta_3_2 + theta_4_3 - theta_4_2 = 0

Ciclo 7:

+theta_2_1: -5.1794°
 +theta_5_2: -3.0844°
 -theta_5_1: +8.2638°
 Suma= 0.0000e+00° (OK) +theta_2_1 + theta_5_2 - theta_5_1 = 0

VERIFICACIÓN DE LAS ENVOLVENTES θ_{ij}

#	θ_k [°]	L_k [°]	U_k [°]	cumple_inf	cumple_sup
1	-5.1794	-7.6220	-1.4625	True	True
2	-8.2638	-12.3840	-6.2245	True	True
3	-6.5531	-10.4053	-4.2458	True	True
4	-4.4841	-6.7525	-0.5930	True	True
5	-3.0844	-5.5994	0.5600	True	True
6	2.0690	-1.7178	4.4417	True	True
7	1.3997	-1.7952	4.3642	True	True
8	-2.7292	-5.9803	0.1792	True	True
9	-4.1094	-6.7754	-0.6160	True	True
10	-4.8690	-8.5771	-2.4176	True	True
11	-0.7077	-4.0686	2.0909	True	True
12	-0.9272	-4.1239	2.0356	True	True
13	-1.1085	-4.3947	1.7647	True	True
14	0.0338	-3.0797	3.0797	True	True
15	-1.3802	-4.5566	1.6029	True	True
16	-0.3994	-3.2803	2.8792	True	True
17	-1.4635	-4.3597	1.7997	True	True

18	0.3317	-2.5065	3.6530	True	True
19	-0.1813	-3.3350	2.8245	True	True
20	-0.9950	-4.3133	1.8461	True	True

10.2.2 Inyección en ACPF

```

-----
-----
Status of the Optimization: optimal
Objective value: 269.38509427846964
-----
-----
-----

```

BUS__RESULTS

```

-----
-----
-----
Bus   Type   V0      Th0      Pg      Qg      Pd      Qd      gshb
bshb
-----
-----
1     3     1.0500  -0.0000  269.3851  7.7071  0.0000  0.0000  0.0000
0.0000
2     2     1.0199  -5.9321  5.5846  26.8278  21.7000  12.7000  0.0000
0.0000
3     2     0.9887  -14.0069  0.0000  29.4173  94.2000  19.0000  0.0000
0.0000
4     0     0.9939  -11.3258  -0.0000  0.0000  47.8000  -3.9000  0.0000
0.0000
5     0     0.9973  -9.6822  -0.0000  -0.0000  7.6000  1.6000  0.0000
0.0000
6     2     1.0500  -15.4584  -0.0000  17.4473  11.2000  7.5000  0.0000
0.0000
7     0     1.0306  -14.4829  0.0000  0.0000  0.0000  0.0000  0.0000
0.0000
8     2     1.0500  -14.4829  -0.0000  11.5525  0.0000  0.0000  0.0000
0.0000
9     0     1.0273  -16.1252  0.0000  -0.0000  29.5000  16.6000  0.0000
20.0521
10    0     1.0237  -16.3028  0.0000  -0.0000  9.0000  5.8000  0.0000
0.0000
11    0     1.0331  -16.0120  0.0000  -0.0000  3.5000  1.8000  0.0000
0.0000
12    0     1.0343  -16.3474  -0.0000  -0.0000  6.1000  1.6000  0.0000
0.0000
13    0     1.0287  -16.4183  0.0000  -0.0000  13.5000  5.8000  0.0000
0.0000

```

14	0	1.0095	-17.3032	-0.0000	0.0000	14.9000	5.0000	0.0000
----	---	--------	----------	---------	--------	---------	--------	--------

0.0000

TOTAL			274.9697	92.9520	259.0000	73.5000	0.0000
-------	--	--	----------	---------	----------	---------	--------

20.0521

BRANCH__RESULTS

Branch	From	To	Pij[MW]	Pji[MW]	Qij[MW]	Qji[MW]	Pls[MW]	Qls[MVAr]
--------	------	----	---------	---------	---------	---------	---------	-----------

SLoad

1	1	2	187.5946	-181.4081	-1.2479	14.4797	6.1866	13.2318
---	---	---	----------	-----------	---------	---------	--------	---------

121.78

2	1	5	81.7905	-78.4454	8.9550	-0.3052	3.3451	8.6498
---	---	---	---------	----------	--------	---------	--------	--------

83.58

3	2	3	72.4774	-70.0975	1.6423	3.9656	2.3799	5.6079
---	---	---	---------	----------	--------	--------	--------	--------

64.88

4	2	4	53.9758	-52.3482	-1.9613	3.4522	1.6276	1.4909
---	---	---	---------	----------	---------	--------	--------	--------

59.54

5	2	5	38.8394	-38.0118	-0.0329	-0.9606	0.8276	-0.9935
---	---	---	---------	----------	---------	---------	--------	---------

60.34

6	3	4	-24.1025	24.5351	6.4517	-6.6055	0.4326	-0.1538
---	---	---	----------	---------	--------	---------	--------	---------

71.73

7	4	5	-63.4206	63.9867	12.9196	-11.1338	0.5662	1.7858
---	---	---	----------	---------	---------	----------	--------	--------

86.09

8	4	7	26.9761	-27.5829	-16.7172	18.8495	0.0000	2.1322
---	---	---	---------	----------	----------	---------	--------	--------

44.64

9	4	9	15.3595	-15.8508	-5.3308	6.8192	0.0000	1.4883
---	---	---	---------	----------	---------	--------	--------	--------

33.08

10	5	6	41.8193	-44.8705	-18.7313	24.0513	0.0000	5.3200
----	---	---	---------	----------	----------	---------	--------	--------

52.89

11	6	11	7.7706	-7.6950	5.2367	-5.0783	0.0756	0.1584
----	---	----	--------	---------	--------	---------	--------	--------

21.95

12	6	12	7.8936	-7.8159	2.7189	-2.5571	0.0777	0.1617
----	---	----	--------	---------	--------	---------	--------	--------

17.43

13	6	13	18.0062	-17.7723	8.1037	-7.6430	0.2339	0.4607
----	---	----	---------	----------	--------	---------	--------	--------

43.32

14	7	8	0.0000	0.0000	-11.3392	11.5525	0.0000	0.2132
----	---	---	--------	--------	----------	---------	--------	--------

14.83

15	7	9	27.5829	-27.5829	3.4915	-2.6909	0.0000	0.8006
----	---	---	---------	----------	--------	---------	--------	--------

48.31

16	9	10	4.8359	-4.8268	2.5969	-2.5727	0.0091	0.0241
----	---	----	--------	---------	--------	---------	--------	--------

7.53

17	9	14	9.0978	-8.9901	2.5794	-2.3503	0.1077	0.2291
----	---	----	--------	---------	--------	---------	--------	--------

22.36

18	10	11	-4.1732	4.1950	-3.2273	3.2783	0.0218	0.0510
7.09								
19	12	13	1.7159	-1.7079	0.9571	-0.9499	0.0080	0.0072
0.97								
20	13	14	5.9802	-5.9099	2.7929	-2.6497	0.0704	0.1433
10.89								

10.3 Resultados relajación SOCP: Caso 30 nudos con $\theta_m = \pm 60$

10.3.1 Relajación

Status of the Optimization: optimal

Objective value: 548.614562 €

-

GENERATOR__RESULTS

-

Gen	Bus	Pg[MW]	Qg[MVar]	Pg0[MW]	Qg0[MVar]	Pg_min[MW]	Pg_max[MW]	Qg_min[MVar]	Qg_max[MVar]
1	1	128.3724	-10.9757	125.0000	115.0000	50.0000	200.0000	-	-
20.0000		250.0000							
2	2	80.0000	22.9214	50.0000	40.0000	20.0000	80.0000	-	-
20.0000		100.0000							
3	5	50.0000	25.9848	32.5000	32.5000	15.0000	50.0000	-	-
15.0000		80.0000							
4	8	10.0000	29.5024	22.5000	22.5000	10.0000	35.0000	-	-
15.0000		60.0000							
5	11	10.0000	4.0767	20.0000	20.0000	10.0000	30.0000	-	-
10.0000		50.0000							
6	13	12.0000	1.8322	26.0000	22.5000	12.0000	40.0000	-	-
15.0000		60.0000							
7	15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-	-
0.0000		0.0000							

-

BUS__RESULTS

-

Bus	Type	V0	Th0	Pg	Qg	Pd	Qd	gshb
bshb								

1	3	1.0500	0.0000	128.3724	-10.9757	0.0000	0.0000	0.0000
0.0000								
2	2	1.0427	-1.6243	80.0000	22.9214	21.7000	12.7000	0.0000
0.0000								
3	0	1.0241	-3.7472	0.0000	0.0000	2.4000	1.2000	0.0000
0.0000								
4	0	1.0174	-4.2892	0.0000	0.0000	7.6000	1.6000	0.0000
0.0000								
5	2	1.0222	-5.9894	50.0000	25.9848	94.2000	19.0000	0.0000
0.0000								
6	0	1.0188	-5.2908	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000								
7	0	1.0121	-6.2285	0.0000	0.0000	22.8000	10.9000	0.0000
0.0000								
8	2	1.0209	-6.0810	10.0000	29.5024	30.0000	30.0000	0.0000
0.0000								
9	0	1.0421	-6.8923	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000								
10	0	1.0393	-7.9966	0.0000	0.0000	5.8000	2.0000	0.0000
20.5223								
11	2	1.0500	-5.9383	10.0000	4.0767	0.0000	0.0000	0.0000
0.0000								
12	0	1.0477	-7.3102	0.0000	0.0000	11.2000	7.5000	0.0000
0.0000								
13	2	1.0500	-6.5391	12.0000	1.8322	0.0000	0.0000	0.0000
0.0000								
14	0	1.0332	-8.1131	0.0000	0.0000	6.2000	1.6000	0.0000
0.0000								
15	0	1.0295	-8.1255	0.0000	0.0000	8.2000	2.5000	0.0000
0.0000								
16	0	1.0365	-7.9948	0.0000	0.0000	3.5000	1.8000	0.0000
0.0000								
17	0	1.0334	-8.2853	0.0000	0.0000	9.0000	5.8000	0.0000
0.0000								
18	0	1.0207	-8.8249	0.0000	0.0000	3.2000	0.9000	0.0000
0.0000								
19	0	1.0187	-9.0805	0.0000	0.0000	9.5000	3.4000	0.0000
0.0000								
20	0	1.0231	-8.9326	0.0000	0.0000	2.2000	0.7000	0.0000
0.0000								
21	0	1.0272	-8.3617	0.0000	0.0000	17.5000	11.2000	0.0000
0.0000								
22	0	1.0278	-8.0213	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000								
23	0	1.0207	-8.0888	0.0000	0.0000	3.2000	1.6000	0.0000
0.0000								

24	0	1.0174	-7.7511	0.0000	0.0000	8.7000	6.7000	0.0000
4.1408								
25	0	1.0198	-6.2789	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000								
26	0	1.0021	-7.1848	0.0000	0.0000	3.5000	2.3000	0.0000
0.0000								
27	0	1.0298	-4.4292	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000								
28	0	1.0203	-6.6072	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000								
29	0	1.0101	-5.7647	0.0000	0.0000	2.4000	0.9000	0.0000
0.0000								
30	0	0.9987	-6.6335	0.0000	0.0000	10.6000	1.9000	0.0000
0.0000								

TOTAL				290.3724	73.3418	283.4000	126.2000	0.0000
24.6631								

BRANCH RESULTS

Branch	From	To	Pij[MW]	Pji[MW]	Qij[MVAr]	Qji[MVAr]	Pls[MW]	Qls[MVAr]
SLoad[%]								

1	1	2	52.7894	-71.0974	-6.4555	2.1346	0.9181	-4.3210
10.45								
2	1	3	44.1271	-55.0500	3.5572	-4.9775	1.3070	-1.4202
30.55								
3	2	4	30.3246	-32.5586	3.8735	-6.2545	0.5897	-2.3810
36.34								
4	3	4	28.7801	-52.3010	7.5272	-8.0802	0.3489	-0.5529
31.03								
5	2	5	41.4954	-52.4434	0.1900	-1.4946	1.2519	-1.3046
27.53								
6	2	6	39.3259	-41.5848	0.4256	-1.7764	0.9690	-1.3508
30.80								
7	4	6	39.6365	-46.4411	-14.8422	14.6202	0.2814	-0.2220
29.27								
8	5	7	6.2671	-8.1995	5.3526	-7.3739	0.0440	-2.0213
19.46								
9	6	7	21.1066	-14.6005	0.7428	-2.1418	0.0587	-1.3990
22.03								
10	6	8	30.3162	-20.7282	-13.8952	13.4039	0.0635	-0.4912
69.51								
11	6	9	14.2656	-22.1051	-11.2121	11.8719	0.0000	0.6597
51.39								

12	6	10	8.9901	-12.9318	-3.5394	4.0395	0.0000	0.5000
45.89								
13	9	11	-8.7588	10.0000	-3.8788	4.0545	0.0000	0.1757
29.11								
14	9	10	18.9756	-32.1051	2.8574	-2.4844	0.0000	0.3730
84.65								
15	4	12	21.9445	-30.5373	-11.4375	12.9519	0.0000	1.5144
30.56								
16	12	13	-10.5754	12.0000	-1.6655	1.8117	0.0000	0.1462
65.46								
17	12	14	7.1402	-7.3629	2.5201	-2.3865	0.0683	0.1337
59.91								
18	12	15	15.2850	-16.9811	6.9255	-6.5910	0.1997	0.3345
53.09								
19	12	16	7.6259	-6.6799	2.3056	-2.1907	0.0453	0.1149
63.47								
20	14	15	1.0116	-1.1592	0.8105	-0.8073	0.0037	0.0031
7.98								
21	16	17	3.0587	-3.1746	0.8589	-0.8409	0.0053	0.0181
43.14								
22	15	18	6.3793	-5.1404	1.0357	-0.9496	0.0297	0.0861
41.77								
23	18	19	3.5284	-1.9378	-0.1258	0.1413	0.0026	0.0155
49.86								
24	19	20	-5.7810	7.5834	-3.6500	3.6806	0.0212	0.0306
71.06								
25	10	20	9.9572	-9.7834	3.6769	-3.4588	0.0965	0.2180
50.07								
26	10	17	8.0202	-5.8254	4.2147	-4.1505	0.0179	0.0642
82.09								
27	10	21	13.9057	-15.9070	10.3803	-10.1714	0.1112	0.2088
57.04								
28	10	22	3.3710	-7.4451	6.3188	-6.2476	0.0505	0.0712
94.46								
29	21	22	-22.4910	1.5937	8.3288	-8.2001	0.0007	0.1287
2300.86								
30	15	23	1.5147	-4.7445	3.7315	-3.7006	0.0258	0.0309
36.20								
31	22	24	0.7604	-5.8087	5.4715	-5.4198	0.0426	0.0517
63.12								
32	23	24	-1.3406	-1.5412	1.8959	-1.8819	0.0033	0.0140
23.67								
33	24	25	-6.3619	1.3534	3.0305	-2.8726	0.0033	0.1579
198.63								
34	25	26	5.1391	-3.5000	1.3253	-1.2223	0.0444	0.1029
112.67								
35	25	27	-14.6485	4.9287	3.0429	-2.5936	0.0308	0.4492
223.84								

36	28	27	-10.0834	-18.2070	-2.2550	2.6611	0.0000	0.4061
54.17								
37	27	29	6.6078	-6.1095	1.4557	-1.2764	0.0853	0.1793
45.78								
38	27	30	7.3719	-6.9237	1.5225	-1.2004	0.1598	0.3220
56.66								
39	29	30	3.6961	-3.6763	0.6083	-0.5460	0.0332	0.0623
56.13								
40	8	28	4.4376	-0.7278	-12.2362	-10.0135	0.0003	-22.2498
677.67								
41	6	28	36.3955	-17.4792	-15.6768	9.7720	0.0588	-5.9048
251.26								

CICLOS FINALES Y DIFERENCIAS ANGULARES

theta_10_17:	+0.2887°	(+0.0050 rad)
theta_10_20:	+0.9360°	(+0.0163 rad)
theta_10_21:	+0.3651°	(+0.0064 rad)
theta_10_22:	+0.0246°	(+0.0004 rad)
theta_10_6:	-2.7058°	(-0.0472 rad)
theta_10_9:	-1.1043°	(-0.0193 rad)
theta_11_9:	+0.9540°	(+0.0167 rad)
theta_12_13:	-0.7712°	(-0.0135 rad)
theta_12_14:	+0.8029°	(+0.0140 rad)
theta_12_15:	+0.8153°	(+0.0142 rad)
theta_12_16:	+0.6845°	(+0.0119 rad)
theta_12_4:	-3.0210°	(-0.0527 rad)
theta_13_12:	+0.7712°	(+0.0135 rad)
theta_14_12:	-0.8029°	(-0.0140 rad)
theta_14_15:	+0.0123°	(+0.0002 rad)
theta_15_12:	-0.8153°	(-0.0142 rad)
theta_15_14:	-0.0123°	(-0.0002 rad)
theta_15_18:	+0.6994°	(+0.0122 rad)
theta_15_23:	-0.0366°	(-0.0006 rad)
theta_16_12:	-0.6845°	(-0.0119 rad)
theta_16_17:	+0.2906°	(+0.0051 rad)
theta_17_10:	-0.2887°	(-0.0050 rad)
theta_17_16:	-0.2906°	(-0.0051 rad)
theta_18_15:	-0.6994°	(-0.0122 rad)
theta_18_19:	+0.2556°	(+0.0045 rad)
theta_19_18:	-0.2556°	(-0.0045 rad)
theta_19_20:	-0.1479°	(-0.0026 rad)
theta_1_2:	+1.6243°	(+0.0283 rad)
theta_1_3:	+3.7472°	(+0.0654 rad)
theta_20_10:	-0.9360°	(-0.0163 rad)
theta_20_19:	+0.1479°	(+0.0026 rad)

theta_21_10: -0.3651° (-0.0064 rad)
 theta_21_22: -0.3405° (-0.0059 rad)
 theta_22_10: -0.0246° (-0.0004 rad)
 theta_22_21: +0.3405° (+0.0059 rad)
 theta_22_24: -0.2702° (-0.0047 rad)
 theta_23_15: +0.0366° (+0.0006 rad)
 theta_23_24: -0.3378° (-0.0059 rad)
 theta_24_22: +0.2702° (+0.0047 rad)
 theta_24_23: +0.3378° (+0.0059 rad)
 theta_24_25: -1.4721° (-0.0257 rad)
 theta_25_24: +1.4721° (+0.0257 rad)
 theta_25_26: +0.9059° (+0.0158 rad)
 theta_25_27: -1.8498° (-0.0323 rad)
 theta_26_25: -0.9059° (-0.0158 rad)
 theta_27_25: +1.8498° (+0.0323 rad)
 theta_27_28: +2.1781° (+0.0380 rad)
 theta_27_29: +1.3355° (+0.0233 rad)
 theta_27_30: +2.2043° (+0.0385 rad)
 theta_28_27: -2.1781° (-0.0380 rad)
 theta_28_6: -1.3164° (-0.0230 rad)
 theta_28_8: -0.5262° (-0.0092 rad)
 theta_29_27: -1.3355° (-0.0233 rad)
 theta_29_30: +0.8688° (+0.0152 rad)
 theta_2_1: -1.6243° (-0.0283 rad)
 theta_2_4: +2.6649° (+0.0465 rad)
 theta_2_5: +4.3650° (+0.0762 rad)
 theta_2_6: +3.6665° (+0.0640 rad)
 theta_30_27: -2.2043° (-0.0385 rad)
 theta_30_29: -0.8688° (-0.0152 rad)
 theta_3_1: -3.7472° (-0.0654 rad)
 theta_3_4: +0.5420° (+0.0095 rad)
 theta_4_12: +3.0210° (+0.0527 rad)
 theta_4_2: -2.6649° (-0.0465 rad)
 theta_4_3: -0.5420° (-0.0095 rad)
 theta_4_6: +1.0016° (+0.0175 rad)
 theta_5_2: -4.3650° (-0.0762 rad)
 theta_5_7: +0.2391° (+0.0042 rad)
 theta_6_10: +2.7058° (+0.0472 rad)
 theta_6_2: -3.6665° (-0.0640 rad)
 theta_6_28: +1.3164° (+0.0230 rad)
 theta_6_4: -1.0016° (-0.0175 rad)
 theta_6_7: +0.9376° (+0.0164 rad)
 theta_6_8: +0.7902° (+0.0138 rad)
 theta_6_9: +1.6015° (+0.0280 rad)
 theta_7_5: -0.2391° (-0.0042 rad)
 theta_7_6: -0.9376° (-0.0164 rad)
 theta_8_28: +0.5262° (+0.0092 rad)
 theta_8_6: -0.7902° (-0.0138 rad)
 theta_9_10: +1.1043° (+0.0193 rad)

theta_9_11: -0.9540° (-0.0167 rad)

theta_9_6: -1.6015° (-0.0280 rad)

Verificación KVL:

Ciclo 1:

+theta_29_27: -1.3355°

+theta_30_29: -0.8688°

-theta_30_27: $+2.2043^\circ$

Suma= $0.0000e+00^\circ$ (OK) +theta_29_27 + theta_30_29 - theta_30_27 = 0

Ciclo 2:

-theta_8_6: $+0.7902^\circ$

+theta_28_6: -1.3164°

-theta_28_8: $+0.5262^\circ$

Suma= $-3.3307e-16^\circ$ (OK) -theta_8_6 + theta_28_6 - theta_28_8 = 0

Ciclo 3:

+theta_10_9: -1.1043°

-theta_10_6: $+2.7058^\circ$

+theta_9_6: -1.6015°

Suma= $-2.2204e-16^\circ$ (OK) +theta_10_9 - theta_10_6 + theta_9_6 = 0

Ciclo 4:

+theta_22_21: $+0.3405^\circ$

-theta_22_10: $+0.0246^\circ$

+theta_21_10: -0.3651°

Suma= $2.7756e-16^\circ$ (OK) +theta_22_21 - theta_22_10 + theta_21_10 = 0

Ciclo 5:

-theta_25_24: -1.4721°

-theta_24_22: -0.2702°

-theta_22_10: $+0.0246^\circ$

-theta_10_6: $+2.7058^\circ$

+theta_28_6: -1.3164°

-theta_28_27: $+2.1781^\circ$

-theta_27_25: -1.8498°

Suma= $4.4409e-16^\circ$ (OK) -theta_25_24 - theta_24_22 - theta_22_10 - theta_10_6 +
theta_28_6 - theta_28_27 - theta_27_25 = 0

Ciclo 6:

-theta_20_19: -0.1479°

-theta_19_18: $+0.2556^\circ$

-theta_18_15: $+0.6994^\circ$

+theta_23_15: $+0.0366^\circ$

+theta_24_23: $+0.3378^\circ$

-theta_24_22: -0.2702°

-theta_22_10: $+0.0246^\circ$

+theta_20_10: -0.9360°

Suma= $-7.7716 \times 10^{-16}^\circ$ (OK) $-\theta_{20_19} - \theta_{19_18} - \theta_{18_15} + \theta_{23_15} + \theta_{24_23} - \theta_{24_22} - \theta_{22_10} + \theta_{20_10} = 0$

Ciclo 7:

+ θ_{14_12} : -0.8029°

+ θ_{15_14} : -0.0123°

- θ_{15_12} : $+0.8153^\circ$

Suma= $1.1102 \times 10^{-16}^\circ$ (OK) $+\theta_{14_12} + \theta_{15_14} - \theta_{15_12} = 0$

Ciclo 8:

+ θ_{12_4} : -3.0210°

+ θ_{15_12} : -0.8153°

+ θ_{23_15} : $+0.0366^\circ$

+ θ_{24_23} : $+0.3378^\circ$

- θ_{24_22} : -0.2702°

- θ_{22_10} : $+0.0246^\circ$

- θ_{10_6} : $+2.7058^\circ$

- θ_{6_4} : $+1.0016^\circ$

Suma= $-4.4409 \times 10^{-16}^\circ$ (OK) $+\theta_{12_4} + \theta_{15_12} + \theta_{23_15} + \theta_{24_23} - \theta_{24_22} - \theta_{22_10} - \theta_{10_6} - \theta_{6_4} = 0$

Ciclo 9:

- θ_{17_16} : $+0.2906^\circ$

- θ_{16_12} : $+0.6845^\circ$

+ θ_{15_12} : -0.8153°

+ θ_{23_15} : $+0.0366^\circ$

+ θ_{24_23} : $+0.3378^\circ$

- θ_{24_22} : -0.2702°

- θ_{22_10} : $+0.0246^\circ$

+ θ_{17_10} : -0.2887°

Suma= $-1.3323 \times 10^{-15}^\circ$ (OK) $-\theta_{17_16} - \theta_{16_12} + \theta_{15_12} + \theta_{23_15} + \theta_{24_23} - \theta_{24_22} - \theta_{22_10} + \theta_{17_10} = 0$

Ciclo 10:

+ θ_{5_2} : -4.3650°

+ θ_{7_5} : -0.2391°

- θ_{7_6} : $+0.9376^\circ$

- θ_{6_2} : $+3.6665^\circ$

Suma= 0.0000×10^0 (OK) $+\theta_{5_2} + \theta_{7_5} - \theta_{7_6} - \theta_{6_2} = 0$

Ciclo 11:

+ θ_{4_2} : -2.6649°

+ θ_{6_4} : -1.0016°

- θ_{6_2} : $+3.6665^\circ$

Suma= $-4.4409 \times 10^{-16}^\circ$ (OK) $+\theta_{4_2} + \theta_{6_4} - \theta_{6_2} = 0$

Ciclo 12:

- θ_{2_1} : $+1.6243^\circ$

+ θ_{3_1} : -3.7472°

+theta_4_3: -0.5420°

-theta_4_2: +2.6649°

Suma= -8.8818e-16° (OK) -theta_2_1 + theta_3_1 + theta_4_3 - theta_4_2 = 0

VERIFICACIÓN DE LAS ENVOLVENTES θ_{ij}

#	θ_k [°]	L_k [°]	U_k [°]	cumple_inf	cumple_sup
1	-1.6243	-4.8281	1.3314	True	True
2	-3.7472	-7.2484	-1.0889	True	True
3	-2.6649	-5.8129	0.3466	True	True
4	-0.5420	-4.0176	2.1418	True	True
5	-4.3650	-7.8203	-1.6608	True	True
6	-3.6665	-6.4671	-0.3076	True	True
7	-1.0016	-3.9018	2.2576	True	True
8	-0.2391	-3.6580	2.5014	True	True
9	-0.9376	-3.7002	2.4593	True	True
10	-0.7902	-3.4318	2.7277	True	True
11	-1.6015	-5.2286	0.9309	True	True
12	-2.7058	-6.4492	-0.2897	True	True
13	0.9540	-2.1365	4.0230	True	True
14	-1.1043	-4.6977	1.4618	True	True
15	-3.0210	-6.7187	-0.5593	True	True
16	0.7712	-2.3220	3.8375	True	True
17	-0.8029	-4.0700	2.0895	True	True
18	-0.8153	-4.2692	1.8903	True	True
19	-0.6845	-3.7996	2.3598	True	True
20	-0.0123	-3.2537	2.9058	True	True
21	-0.2906	-3.3821	2.7773	True	True
22	-0.6994	-3.6888	2.4707	True	True
23	-0.2556	-3.2190	2.9405	True	True
24	0.1479	-2.7890	3.3705	True	True
25	-0.9360	-4.1866	1.9729	True	True
26	-0.2887	-3.3819	2.7776	True	True
27	-0.3651	-3.7820	2.3775	True	True
28	-0.0246	-3.7411	2.4183	True	True
29	0.3405	-3.0510	3.1084	True	True
30	0.0366	-3.6305	2.5289	True	True
31	0.2702	-3.6901	2.4693	True	True
32	0.3378	-3.3085	2.8510	True	True
33	1.4721	-2.8715	3.2879	True	True
34	-0.9059	-4.0096	2.1499	True	True
35	1.8498	-2.4712	3.6882	True	True
36	-2.1781	-6.4848	-0.3253	True	True
37	-1.3355	-4.4476	1.7119	True	True
38	-2.2043	-5.3034	0.8561	True	True
39	-0.8688	-3.9630	2.1964	True	True
40	-0.5262	-3.1513	3.0082	True	True

41 -1.3164 -3.5187 2.6408 True True

10.3.2 Inyección en ACPF

BUS__RESULTS								
Bus	Type	V0	Th0	Pg	Qg	Pd	Qd	gshb
bshb								
1	3	1.0500	-0.0000	127.8897	-11.0794	0.0000	0.0000	0.0000
2	2	1.0427	-2.3460	80.0000	22.7548	21.7000	12.7000	0.0000
3	0	1.0240	-4.5979	0.0000	0.0000	2.4000	1.2000	0.0000
4	0	1.0173	-5.6258	0.0000	0.0000	7.6000	1.6000	0.0000
5	2	1.0222	-7.5442	50.0000	26.1072	94.2000	19.0000	0.0000
6	0	1.0188	-6.7583	0.0000	0.0000	0.0000	0.0000	0.0000
7	0	1.0122	-7.6005	0.0000	-0.0000	22.8000	10.9000	0.0000
8	2	1.0209	-7.3009	10.5000	29.4876	30.0000	30.0000	0.0000
9	0	1.0420	-8.9525	0.0000	0.0000	0.0000	0.0000	0.0000
10	0	1.0388	-10.6983	0.0000	-0.0000	5.8000	2.0000	0.0000
11	2	1.0500	-7.8631	10.0000	4.1546	0.0000	0.0000	0.0000
12	0	1.0479	-9.8110	0.0000	0.0000	11.2000	7.5000	0.0000
13	2	1.0500	-8.9362	12.0000	1.6335	0.0000	0.0000	0.0000
14	0	1.0336	-10.7358	0.0000	-0.0000	6.2000	1.6000	0.0000
15	0	1.0294	-10.8602	0.0000	0.0000	8.2000	2.5000	0.0000
16	0	1.0363	-10.4604	0.0000	-0.0000	3.5000	1.8000	0.0000
17	0	1.0330	-10.8408	0.0000	-0.0000	9.0000	5.8000	0.0000

18	0	1.0205	-11.5080	-0.0000	-0.0000	3.2000	0.9000	0.0000
0.0000								
19	0	1.0184	-11.6988	0.0000	0.0000	9.5000	3.4000	0.0000
0.0000								
20	0	1.0227	-11.5071	-0.0000	-0.0000	2.2000	0.7000	0.0000
0.0000								
21	0	1.0267	-11.1606	-0.0000	0.0000	17.5000	11.2000	0.0000
0.0000								
22	0	1.0274	-11.1509	0.0000	-0.0000	0.0000	0.0000	0.0000
0.0000								
23	0	1.0205	-11.3245	0.0000	-0.0000	3.2000	1.6000	0.0000
0.0000								
24	0	1.0171	-11.5942	-0.0000	0.0000	8.7000	6.7000	0.0000
4.1376								
25	0	1.0192	-11.4506	-0.0000	-0.0000	0.0000	0.0000	0.0000
0.0000								
26	0	1.0015	-11.8687	-0.0000	-0.0000	3.5000	2.3000	0.0000
0.0000								
27	0	1.0291	-11.0970	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000								
28	0	1.0203	-7.3861	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000								
29	0	1.0094	-12.3127	-0.0000	-0.0000	2.4000	0.9000	0.0000
0.0000								
30	0	0.9980	-13.1850	0.0000	-0.0000	10.6000	1.9000	0.0000
0.0000								

TOTAL			290.3897	73.0583	283.4000	126.2000	0.0000	24.6416
-------	--	--	----------	---------	----------	----------	--------	---------

BRANCH__RESULTS

Branch	From	To	Pij[MW]	Pji[MW]	Qij[MW]	Qji[MW]	Pls[MW]	Qls[MVar]
SLoad								
1	1	2	74.6060	-73.6193	-12.8962	10.0703	0.9867	-2.8259
11.84								
2	1	3	53.2837	-52.1129	1.8168	-1.9258	1.1708	-0.1090
28.42								
3	2	4	36.3573	-35.6544	2.3257	-4.0886	0.7028	-1.7629
43.88								
4	3	4	49.7129	-49.4016	0.7258	-0.7070	0.3113	0.0188
27.39								
5	2	5	49.0131	-47.9694	-0.9486	0.8772	1.0437	-0.0714
24.03								

6	2	6	46.5490	-45.3909	-1.3926	1.0390	1.1581	-0.3536
38.56								
7	4	6	44.8478	-44.5867	-16.5696	16.5453	0.2611	-0.0244
28.22								
8	5	7	3.7694	-3.7397	6.2300	-8.2660	0.0297	-2.0360
13.17								
9	6	7	19.1558	-19.0603	1.1741	-2.6340	0.0955	-1.4599
18.28								
10	6	8	20.3592	-20.2977	-11.2888	10.5679	0.0615	-0.7210
33.87								
11	6	9	19.5398	-19.9793	-10.9733	11.9797	0.0000	1.0064
44.30								
12	6	10	13.0795	-13.4979	-3.2209	4.1929	0.0000	0.9720
49.94								
13	9	11	-10.0000	10.0000	-3.9333	4.1546	0.0000	0.2212
29.32								
14	9	10	29.9793	-29.9793	3.4256	-2.5031	0.0000	0.9225
74.33								
15	4	12	30.3909	-32.6083	-11.0730	13.6611	0.0000	2.5881
34.72								
16	12	13	-12.0000	12.0000	-1.4473	1.6335	0.0000	0.1862
65.19								
17	12	14	7.8639	-7.7894	2.1458	-1.9910	0.0745	0.1548
66.44								
18	12	15	18.0994	-17.8816	5.8128	-5.3837	0.2178	0.4291
57.82								
19	12	16	7.4450	-7.3914	2.6297	-2.5169	0.0536	0.1128
62.34								
20	14	15	1.5894	-1.5838	0.3910	-0.3860	0.0055	0.0050
10.72								
21	16	17	3.8914	-3.8838	0.7169	-0.6889	0.0076	0.0280
62.63								
22	15	18	6.0602	-6.0213	1.2696	-1.1905	0.0388	0.0790
38.34								
23	18	19	2.8213	-2.8164	0.2905	-0.2805	0.0049	0.0100
32.18								
24	19	20	-6.6836	6.7014	-3.1195	3.1551	0.0178	0.0357
54.86								
25	10	20	8.9856	-8.9014	4.0432	-3.8551	0.0842	0.1880
43.15								
26	10	17	5.1321	-5.1162	5.1525	-5.1111	0.0159	0.0414
52.89								
27	10	21	15.8814	-15.7713	9.4570	-9.2198	0.1102	0.2371
54.66								
28	10	22	7.6780	-7.6262	4.2368	-4.1299	0.0518	0.1068
76.90								
29	21	22	-1.7287	1.7295	-1.9802	1.9817	0.0008	0.0015
27.67								

30	15	23	5.2052	-5.1759	2.0001	-1.9408	0.0293	0.0593
31.10								
31	22	24	5.8967	-5.8538	2.1482	-2.0814	0.0429	0.0668
39.39								
32	23	24	1.9759	-1.9708	0.3408	-0.3304	0.0051	0.0104
16.08								
33	24	25	-0.8754	0.8768	-0.1505	0.1531	0.0014	0.0025
3.17								
34	25	26	3.5445	-3.5000	2.3664	-2.3000	0.0445	0.0664
72.65								
35	25	27	-4.4213	4.4485	-2.5195	2.5715	0.0272	0.0520
26.40								
36	28	27	17.1600	-17.7273	-1.7180	2.8495	0.0000	1.1314
51.58								
37	27	29	6.1885	-6.1033	1.6661	-1.5050	0.0852	0.1611
41.07								
38	27	30	7.0902	-6.9299	1.6594	-1.3577	0.1603	0.3018
53.02								
39	29	30	3.7033	-3.6701	0.6050	-0.5423	0.0332	0.0626
56.32								
40	8	28	0.7977	-0.7973	-11.0803	-11.2085	0.0004	-22.2887
505.06								
41	6	28	16.9853	-16.9299	-10.5663	4.0063	0.0554	-6.5600
64.02								

10.4 Endurecimiento de envolventes

10.4.1 ACPF con 14 nudos con $\theta_m = \pm 30$

Status of the Optimization: optimal

Objective value: 2261.405207 €

GENERATOR__RESULTS								

Gen	Bus	Pg[MW]	Qg[MVAr]	Pg0[MW]	Qg0[MVAr]	Pg_min[MW]	Pg_max[MW]	
Qg_min[MVAr]	Qg_max[MVAr]							

1	1	269.0907	10.0000	232.0000	-17.0000	0.0000	340.0000	
0.0000		10.0000						
2	2	5.5846	25.8891	40.0000	42.0000	0.0000	59.0000	-
30.0000		30.0000						
3	3	0.0000	29.0921	0.0000	23.0000	0.0000	0.0000	
0.0000		40.0000						
4	6	0.0000	17.3574	0.0000	12.0000	0.0000	0.0000	-
6.0000		24.0000						
5	8	0.0000	11.6692	0.0000	17.0000	0.0000	0.0000	-
6.0000		24.0000						

BUS__RESULTS								
Bus	Type	V0	Th0	Pg	Qg	Pd	Qd	gshb
bshb								
1	3	1.0500	0.0000	269.3851	7.7071	0.0000	0.0000	0.0000
2	2	1.0199	-5.9321	5.5846	26.8278	21.7000	12.7000	0.0000
3	2	0.9887	-14.0069	0.0000	29.4173	94.2000	19.0000	0.0000
4	0	0.9939	-11.3258	-0.0000	-0.0000	47.8000	-3.9000	0.0000
5	0	0.9973	-9.6822	0.0000	-0.0000	7.6000	1.6000	0.0000
6	2	1.0500	-15.4584	-0.0000	17.4473	11.2000	7.5000	0.0000
7	0	1.0306	-14.4829	-0.0000	-0.0000	0.0000	0.0000	0.0000
8	2	1.0500	-14.4829	0.0000	11.5525	0.0000	0.0000	0.0000
9	0	1.0273	-16.1252	0.0000	-0.0000	29.5000	16.6000	0.0000
10	0	1.0237	-16.3028	-0.0000	-0.0000	9.0000	5.8000	0.0000
11	0	1.0331	-16.0120	0.0000	0.0000	3.5000	1.8000	0.0000
12	0	1.0343	-16.3474	-0.0000	-0.0000	6.1000	1.6000	0.0000
13	0	1.0287	-16.4183	0.0000	0.0000	13.5000	5.8000	0.0000
14	0	1.0095	-17.3032	-0.0000	-0.0000	14.9000	5.0000	0.0000
TOTAL				274.9697	92.9520	259.0000	73.5000	0.0000

BRANCH__RESULTS								
Branch	From	To	Pij[MW]	Pji[MW]	Qij[MW]	Qji[MW]	Pls[MW]	Qls[MVar]
SLoad								
1	1	2	187.5946	-181.4081	-1.2479	14.4797	6.1866	13.2318
2	1	5	81.7905	-78.4454	8.9550	-0.3052	3.3451	8.6498
3	2	3	72.4774	-70.0975	1.6423	3.9656	2.3799	5.6079

4	2	4	53.9758	-52.3482	-1.9613	3.4522	1.6276	1.4909
59.54								
5	2	5	38.8394	-38.0118	-0.0329	-0.9606	0.8276	-0.9935
60.34								
6	3	4	-24.1025	24.5351	6.4517	-6.6055	0.4326	-0.1538
71.73								
7	4	5	-63.4206	63.9867	12.9196	-11.1338	0.5662	1.7858
86.09								
8	4	7	26.9761	-27.5829	-16.7172	18.8495	0.0000	2.1322
44.64								
9	4	9	15.3595	-15.8508	-5.3308	6.8192	0.0000	1.4883
33.08								
10	5	6	41.8193	-44.8705	-18.7313	24.0513	0.0000	5.3200
52.89								
11	6	11	7.7706	-7.6950	5.2367	-5.0783	0.0756	0.1584
21.95								
12	6	12	7.8936	-7.8159	2.7189	-2.5571	0.0777	0.1617
17.43								
13	6	13	18.0062	-17.7723	8.1037	-7.6430	0.2339	0.4607
43.32								
14	7	8	0.0000	0.0000	-11.3392	11.5525	0.0000	0.2132
14.83								
15	7	9	27.5829	-27.5829	3.4915	-2.6909	0.0000	0.8006
48.31								
16	9	10	4.8359	-4.8268	2.5969	-2.5727	0.0091	0.0241
7.53								
17	9	14	9.0978	-8.9901	2.5794	-2.3503	0.1077	0.2291
22.36								
18	10	11	-4.1732	4.1950	-3.2273	3.2783	0.0218	0.0510
7.09								
19	12	13	1.7159	-1.7079	0.9571	-0.9499	0.0080	0.0072
0.97								
20	13	14	5.9802	-5.9099	2.7929	-2.6497	0.0704	0.1433
10.89								

10.4.2 ACPF con 14 nudos con $\theta_m = \pm 45$

Status of the Optimization: optimal

Objective value: 2261.405207 €

GENERATOR__RESULTS							

Gen	Bus	Pg[MW]	Qg[MVAr]	Pg0[MW]	Qg0[MVAr]	Pg_min[MW]	Pg_max[MW]
Qg_min[MVAr]	Qg_max[MVAr]						

1	1	269.0907	10.0000	232.0000	-17.0000	0.0000	340.0000
0.0000	10.0000						

2	2	5.5846	25.8891	40.0000	42.0000	0.0000	59.0000	-
30.0000		30.0000						
3	3	0.0000	29.0921	0.0000	23.0000	0.0000	0.0000	
0.0000		40.0000						
4	6	0.0000	17.3574	0.0000	12.0000	0.0000	0.0000	-
6.0000		24.0000						
5	8	0.0000	11.6692	0.0000	17.0000	0.0000	0.0000	-
6.0000		24.0000						

BUS__RESULTS

Bus	Type	V0	Th0	Pg	Qg	Pd	Qd	gshb
bshb								
1	3	1.0500	0.0000	269.3851	7.7071	0.0000	0.0000	0.0000
0.0000								
2	2	1.0199	-5.9321	5.5846	26.8278	21.7000	12.7000	0.0000
0.0000								
3	2	0.9887	-14.0069	-0.0000	29.4173	94.2000	19.0000	0.0000
0.0000								
4	0	0.9939	-11.3258	-0.0000	-0.0000	47.8000	-3.9000	0.0000
0.0000								
5	0	0.9973	-9.6822	0.0000	0.0000	7.6000	1.6000	0.0000
0.0000								
6	2	1.0500	-15.4584	0.0000	17.4473	11.2000	7.5000	0.0000
0.0000								
7	0	1.0306	-14.4829	-0.0000	-0.0000	0.0000	0.0000	0.0000
0.0000								
8	2	1.0500	-14.4829	0.0000	11.5525	0.0000	0.0000	0.0000
0.0000								
9	0	1.0273	-16.1252	-0.0000	-0.0000	29.5000	16.6000	0.0000
20.0521								
10	0	1.0237	-16.3028	0.0000	-0.0000	9.0000	5.8000	0.0000
0.0000								
11	0	1.0331	-16.0120	-0.0000	-0.0000	3.5000	1.8000	0.0000
0.0000								
12	0	1.0343	-16.3474	0.0000	-0.0000	6.1000	1.6000	0.0000
0.0000								
13	0	1.0287	-16.4183	0.0000	-0.0000	13.5000	5.8000	0.0000
0.0000								
14	0	1.0095	-17.3032	0.0000	-0.0000	14.9000	5.0000	0.0000
0.0000								
TOTAL				274.9697	92.9520	259.0000	73.5000	0.0000
20.0521								

BRANCH__RESULTS

Branch	From	To	Pij[MW]	Pji[MW]	Qij[MW]	Qji[MW]	Pls[MW]	Qls[MVar]
SLoad								
1	1	2	187.5946	-181.4081	-1.2479	14.4797	6.1866	13.2318
121.78								
2	1	5	81.7905	-78.4454	8.9550	-0.3052	3.3451	8.6498
83.58								
3	2	3	72.4774	-70.0975	1.6423	3.9656	2.3799	5.6079
64.88								
4	2	4	53.9758	-52.3482	-1.9613	3.4522	1.6276	1.4909
59.54								
5	2	5	38.8394	-38.0118	-0.0329	-0.9606	0.8276	-0.9935
60.34								
6	3	4	-24.1025	24.5351	6.4517	-6.6055	0.4326	-0.1538
71.73								
7	4	5	-63.4206	63.9867	12.9196	-11.1338	0.5662	1.7858
86.09								
8	4	7	26.9761	-27.5829	-16.7172	18.8495	0.0000	2.1322
44.64								
9	4	9	15.3595	-15.8508	-5.3308	6.8192	0.0000	1.4883
33.08								
10	5	6	41.8193	-44.8705	-18.7313	24.0513	0.0000	5.3200
52.89								
11	6	11	7.7706	-7.6950	5.2367	-5.0783	0.0756	0.1584
21.95								
12	6	12	7.8936	-7.8159	2.7189	-2.5571	0.0777	0.1617
17.43								
13	6	13	18.0062	-17.7723	8.1037	-7.6430	0.2339	0.4607
43.32								
14	7	8	0.0000	0.0000	-11.3392	11.5525	0.0000	0.2132
14.83								
15	7	9	27.5829	-27.5829	3.4915	-2.6909	0.0000	0.8006
48.31								
16	9	10	4.8359	-4.8268	2.5969	-2.5727	0.0091	0.0241
7.53								
17	9	14	9.0978	-8.9901	2.5794	-2.3503	0.1077	0.2291
22.36								
18	10	11	-4.1732	4.1950	-3.2273	3.2783	0.0218	0.0510
7.09								
19	12	13	1.7159	-1.7079	0.9571	-0.9499	0.0080	0.0072
0.97								
20	13	14	5.9802	-5.9099	2.7929	-2.6497	0.0704	0.1433
10.89								