UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Ingeniería

https://revistas.udistrital.edu.co/index.php/reving/issue/view/1241

DOI: https://doi.org/10.14483/23448393.22049

Research

Comparative Analysis of the Julia and AMPL Computational Tools Used in the Radial Distribution Network Optimization Problem

Análisis comparativo de las herramientas computacionales Julia y AMPL utilizadas en el problema de optimización de redes de distribución radiales

Juan Camilo Hoyos Vallejo[1]* and Jaime Quintero Restrepo[1]

[1]Universidad Autónoma de Occidente, Santiago de Cali, Colombia.

Abstract

Context: Research on the development of mathematical models to optimize electric power distribution systems has become increasingly important in recent years. Choosing the right optimization tools and solvers to address optimization problems in these systems has therefore become fundamental.
Method: Nonlinear and mixed-integer nonlinear mathematical models addressing optimal capacitor placement and the allocation of distributed generation were implemented in the AMPL and Julia platforms. These models together with the Ipopt, Knitro, and Bonmin solvers, were tested and compared using 33-, 69-, and 83-bus test systems.
Results: The comparative analysis shows that AMPL allows for a more direct and adequate implementation of this type of optimization problem, while Julia requires more elaborate constructions. The experimental results show significant reductions in system losses through optimal capacitor and distributed generation placement.
Conclusions: AMPL offers a faster learning curve and a syntax that is more suitable for mathematical modeling. On the other hand, Julia provides superior versatility and access to a wider diversity of solvers. Although the evaluated nonlinear solvers proved to be suitable for the non-convex models and reached equivalent solutions, Knitro, a commercial solver, exhibited shorter processing times. In this sense, choosing between free or commercial alternatives involves a compromise between processing times and the available budget. Furthermore, solving the aforementioned optimization problems effectively minimized losses in the test systems. These models are basic versions that can be extended to more complex optimization problems.
Keywords: mathematical optimization, power distribution networks, solvers, optimal capacitor placement, optimal allocation of distributed generation

Article history

Received: 10th/Apr/2024
Modified: 8th/Jul/2024
Accepted: 3rd/Oct/2024

Ing, vol. 29, no. 3, 2024, e22049

*Correspondence: juan_camilo.hoyos@uao.edu.co

Resumen

Contexto: La investigación sobre el desarrollo de modelos matemáticos para optimizar los sistemas de distribución de energía eléctrica ha cobrado cada vez más importancia en los últimos años. La selección de las herramientas de optimización y *solvers* adecuados para resolver los problemas de optimización en estos sistemas se ha vuelto fundamental.
Método: Se implementaron modelos matemáticos no lineales y no lineales de enteros mixtos para abordar la ubicación óptima de capacitores y generación distribuida en las plataformas AMPL y Julia. Estos modelos, junto con los solvers Ipop, Knitro y Bonmin, fueron evaluados y comparados utilizando sistemas de prueba de 33, 69 y 83 barras.
Resultados: El análisis comparativo evidencia que AMPL permite una implementación más directa y adecuada para este tipo de problemas de optimización, mientras que Julia requiere construcciones más elaboradas. Los resultados experimentales evidencian reducciones significativas en las pérdidas del sistema mediante la ubicación optima de capacitores y generación distribuida.
Conclusiones: AMPL ofrece una curva de aprendizaje más rápida y una sintaxis más adecuada para el modelado matemático. Por otro lado, Julia proporciona una versatilidad superior y acceso a una diversidad más amplia de *solvers*. Aunque los *solvers* no lineales evaluados resultaron adecuados para los modelos no convexos y alcanzaron soluciones equivalentes, Knitro, un *solver* comercial, presentó tiempos de procesamiento más cortos. En este sentido, elegir entre alternativas gratuitas o comerciales implica un compromiso entre los tiempos de procesamiento y el presupuesto disponible. Además, la solución de los problemas de optimización mencionados permitió minimizar de manera efectiva las perdidas en los sistemas de prueba. Estos modelos son versiones básicas que pueden ampliarse a problemas de optimización más complejos.

Palabras clave: optimización matemática, redes de distribución eléctrica, solucionadores, colocación óptima de condensadores, asignación óptima de generación distribuida

Table of contents

1. Introduction

In recent years, research on the development of mathematical models for power systems optimization has gained relevance, leading to a significant increase in publications in the field. These publications have focused on the optimization of electrical power systems, using and solving increasingly sophisticated, complex, and realistic mathematical models through commercial or open-source solvers (1).

Within the scope of electrical distribution networks, modeling and solving optimization problems is fundamental to improving efficiency and reliability. Two areas of interest in this field are optimal capacitor placement and the allocation of distributed generation, with the goal of reducing system losses, improving voltage profiles, correcting power factors, and increasing circuit capacity (2). However, there are issues associated with both the modeling and the optimization tools used in these areas.

Regarding modeling, it is crucial to have accurate mathematical formulations that adequately represent the characteristics and constraints of electrical distribution networks. These formulations must consider the relevant input data, as well as the assumptions and limitations of the optimal capacitor and distributed generation placement problem. The lack of an adequate mathematical formulation can lead to suboptimal or unfeasible solutions in practice (1).

Secondly, the optimization tools used to solve the aforementioned mathematical models are a determining factor in the efficiency and effectiveness of the optimization process (3). In this sense, to evaluate and solve optimization problems in this field, it is critical to analyze the characteristics of the available computational tools, *e.g.*, AMPL and Julia. These tools may offer different characteristics, such as ease of use, documentation availability, support, and flexibility.

In this work, Julia was selected as a high-performance open-source alternative to commercial systems, as well as for its JuMP library, which offers significant advances with regard to modeling and extensibility by taking advantage of several features of Julia that are unique among the programming languages used for scientific computing (4). On the other hand, AMPL was chosen because it facilitates experimenting with formulations and simplifies the use of suitable solvers in addressing optimization problems. Moreover, the notation used in AMPL is close to the typical mathematical notation of state variables, objectives, constraints, sets, and parameters (5).

The commercial and open-source solvers used to tackle optimization problems in distribution networks offer different performance characteristics concerning runtime, documentation and support, solution accuracy, and associated costs (6). Comparing and evaluating these solvers will allow determining the most efficient and suitable ones for solving the optimization problems associated with capacitor and generation placement in distribution grids.

Considering the above, this research conducted a detailed analysis of the free computational tools used to solve the studied problem, comparing the AMPL and Julia platforms with regard to mathematical modeling and optimization, as well as different commercial and open-source solvers. This comparison identified the tools' advantages and limitations, as well as their applicability in specific situations.

2. Mathematical model

This section presents the mathematical models implemented in this article: 1) the power flow model adapted for distribution networks, 2) the model for the optimal placement of capacitors, and 3) the model for the optimal placement of distributed generation. The objective function of each model is presented, as well as the assumptions and constraints involved.

2.1. Power flow

To address the optimal capacitor placement and distributed generation allocation problem, a mathematical power flow model must first be implemented, which constitutes the basis for the subsequent models. This model has the following considerations and is based on the one presented in (7), where some modifications are made to the constraints and parameters presented in (8):

1. The electric power distribution system is represented by a single-phase diagram and features a radial topology.
2. The lines' active and reactive power losses are concentrated at the transmitting bus.
3. The loads are modeled as constant powers.
4. The power flow is unidirectional.
5. The PI model's capacitive reactance of the transmission lines is not considered.
6. There is only one energy supply source (*i.e.*, the substation).

The power flow optimization problem is formulated as the following nonlinear problem (7):

$$\text{minimize } f = K_c \sum_{\forall ij \in \Omega_l} R_{ij} I_{ij}^2 \tag{1}$$

The objective function contains a sum representing the energy losses of the system multiplied by a cost constant K_c. Thus, it is possible to determine the total cost associated with the system's energy losses over a period of one year.

Constraints (2) to (7) are technical and operational equations that seek to replicate the limitations and behavior of the analyzed electrical system.

$$\sum_{\forall ki \in \Omega_l} P_{ki} - \sum_{\forall ij \in \Omega_l} \left(P_{ij} + R_{ij} I_{ij}^2\right) + P_i^s = P_i^d, \quad \forall i \in \Omega_b \tag{2}$$

$$\sum_{\forall ki \in \Omega_l} Q_{ki} - \sum_{\forall ij \in \Omega_l} \left(Q_{ij} + X_{ij} I_{ij}^2\right) + Q_i^s = Q_i^d, \quad \forall i \in \Omega_b \tag{3}$$

$$V_i^2 - 2\left(R_{ij}P_{ij} + X_{ij}Q_{ij}\right) - I_{ij}^2\left(R_{ij}^2 + X_{ij}^2\right) - V_j^2 = 0, \quad \forall ij \in \Omega_l \tag{4}$$

$$I_{ij}^2 V_j^2 = P_{ij}^2 + Q_{ij}^2, \quad \forall ij \in \Omega_l \tag{5}$$

$$I_{\text{min}} \leq I \leq I_{\text{max}}, \quad \forall ij \in \Omega_l \tag{6}$$

$$V_{\text{min}} \leq V \leq V_{\text{max}}, \quad \forall i \in \Omega_b \tag{7}$$

Eq. (2) represents the active power flow balance. In it, the first term denotes the active power injected by all the previous k buses that are connected to node i. The second term denotes the active power extracted towards all the subsequent buses j that are connected to node i and the active power losses of the lines ij through which bus i transmits power. The third term is the active power generated by the substation, where $P_i^s = 0$ for all buses $i \neq 1$, and the fourth term is the active power demand at node i.

Eq. (3) represents the reactive power flow balance. Here, the first term denotes the reactive power injected by all the previous k buses that are connected to node i. The second term denotes the reactive power extracted towards all subsequent buses j connected to node i and the reactive power losses of the lines ij through which bus i transmits power. The third term is the reactive power generated by the substation, where $Q_i^s = 0$ for all buses $i \neq 1$, and the fourth term is the reactive power demand at node i.

Eq. (4) represents the voltage drop for each line ij of the network.

The apparent power flow constraint imposes a technical limit on the apparent power that can be transported by each conductor of the distribution network. Eq. (5) represents this constraint for each line ij.

The inequality constraints represent the permissible technical operating limits of the conductors and equipment in the distribution grid. Eq. (6) models the current constraint in each line ij of the system, aiming to prevent the conductors from overheating above their nominal capacities. Similarly, Eq. (7) defines the permissible voltage standards (9) at each node i, within which the connected equipment can operate safely and reliably.

2.2. Optimal capacitor placement

Capacitor placement optimization can be formulated as the following mixed-integer nonlinear

$$\text{minimize } f = f_1 + f_2 + f_3 \tag{8}$$

problem (2):
where f_1 is Eq. (1) and

$$f_2 = D \sum_{\forall i \in \Omega_b} K_i W_i^{ca} \tag{9}$$

$$f_3 = D \sum_{\forall i \in \Omega_b} K_r Q_i^{ca} \tag{10}$$

The expression in (9) accounts for the installation cost of the capacitor banks K_i. Here, the binary variable W_i^{ca} indicates whether or not the capacitor bank is located at a node i. Eq. (10) describes the cost of each capacitor bank K_r. The variable Q_i^{ca} indicates the reactive power injected by the capacitor at node i, thus providing the acquisition cost of the capacitor banks as a function of the reactive power injected. These two sums are multiplied by a depreciation factor D applied to the installation and acquisition costs of the capacitor banks for a period of one year, respectively.

Considering the reactive power introduced by the fixed capacitor banks in the system, the reactive power balance equation in (3) may be modified as follows:

$$\sum_{\forall ki \in \Omega_l} Q_{ki} - \sum_{\forall ij \in \Omega_l} \left(Q_{ij} + X_{ij} I_{ij}^2\right) + Q_i^s + Q_i^{ca} = Q_i^d, \quad \forall i \in \Omega_b \tag{11}$$

The term Q_i^{ca} in Eq. (11) denotes the reactive power injected by the fixed capacitor banks at each node i of the network.
Finally, constraints are added to the model for optimal capacitor placement.

$$\sum_{\forall i \in \Omega_b} W_i^{ca} \leq N_{\text{max}}^{ca} \tag{12}$$

$$0 \leq N_i^{ca} \leq N_{\text{maxunit}}^{ca} W_i^{ca}, \quad \forall i \in \Omega_b \tag{13}$$

$$Q_i^{ca} = N_i^{ca} Q_{\text{base}}^{ca}, \quad \forall i \in \Omega_b \tag{14}$$

Here, Eq. (12) limits the maximum number of capacitor banks allowed in the distribution network, Eq. (13) stipulates the maximum number of capacitive units that can make up each capacitor bank at a specific node i, and Eq. (14) quantifies the reactive power injected by each capacitor bank at each specific node i.

2.3. Optimal allocation of distributed generation

The problem regarding the optimal allocation of distributed generation can be formulated as the following mixed-integer nonlinear model (2):

$$f = \sum_{\forall ij \in \Omega_l} K_c R_{ij} I_{ij}^2 + D \sum_{\forall i \in \Omega_b} \text{Cost}^{gd} P_i^{gd} \tag{15}$$

The objective function of this model contains two sums: Eq. (1) and the investment cost of distributed generation $Cost^{gf}$ as a function of the active power injected into the system Pi^{gd}. An annual depreciation factor of D is applied to the investment costs, with the purpose of conducting a financial analysis in annualized terms over an evaluation period of ten years.

The following modification is made to the active and reactive power balance of Eqs. (2) and (3), respectively:

$$\sum_{\forall ki \in \Omega_l} P_{ki} - \sum_{\forall ij \in \Omega_l} \left(P_{ij} + R_{ij} I_{ij}^2\right) + P_i^s + P_i^{gd} = P_i^d, \quad \forall i \in \Omega_b \tag{16}$$

$$\sum_{\forall ki \in \Omega_l} Q_{ki} - \sum_{\forall ij \in \Omega_l} \left(Q_{ij} + X_{ij} I_{ij}^2\right) + Q_i^s + Q_i^{gd} = Q_i^d, \quad \forall i \in \Omega_b \tag{17}$$

The terms Pi^{gd} in Eq. (16) and Q_i^{gd} in Eq. (17) denote the active and reactive power injected by the distributed generators at each node i of the network. This modification expands the previous mathematical model by incorporating a new source of active and reactive power into the system. Additionally, constraints are added to the model for the optimal allocation of distributed generation.

$$\sum_{\forall i \in \Omega_b} W_i^{gd} \leq N_{\text{max}}^{gd} \tag{18}$$

$$P_i^{gd} \leq P_{\text{max}}^{gd} W_i^{gd}, \quad \forall i \in \Omega_b \tag{19}$$

$$|Q_i^{gd}| = P_i^{gd} \tan\left(\cos^{-1}(fp)\right), \quad \forall i \in \Omega_b \tag{20}$$

Eq. (18) limits the maximum number of distributed generation units in the network, Eq. (19) quantifies the active power injected by each distributed generation unit into each node i, and Eq. (20) quantifies the reactive power injected by each distributed generation unit into each node i.

The following values, obtained from (8), were considered in 33-, 69-, and 83-bus test systems: K_c was equal to 168 $USD/kW-year, K_i was 1600 $USD/bus, K_r was 25 $USD/kvar, D was 10 %, N_{maxunit}^{ca} was 5, Q_{base}^{ca} was 50 kvar, N_{max}^{ca} and N_{max}^{gd} were 3 for the 33- and 69-bus systems and 5 and 6 for the 83-bus feeder, Cost^{gd} was 1105 $USD/kW, P_i^{gd} was 100 kW, and fp was 0.92.

3. Computational Implementation

To computationally solve the proposed mathematical models, AMPL and Julia were used. These platforms allow translating the nonlinear equations and constraints of the models into a language that can be processed by a computer in order to find the optimal solution. The implementation process consisted of translating the mathematical formulas of each model into the specific syntax of AMPL and Julia. Due to their characteristics, and although the procedure was similar in both software platforms, some notable differences were observed for each model. The code may be accessed at (10).

3.1. Available documentation

Documentation is essential to guide the user during the implementation of a model. Julia and AMPL provide documentation but differ in approach and content. In Julia, the relevant documentation is in JuMP, a specific library for mathematical modeling, whereas, in AMPL, it is integrated. Therefore, Julia modeling requires consulting JuMP at (11) rather than the general documentation. JuMP's documentation has a collaborative focus that encourages community participation but may cause

inconsistencies between versions. AMPL, on the other hand, provides formal documentation developed by its creators. This ensures complete and reliable information, although it limits user feedback. The content focuses on using the platform (12). Moreover, JuMP focuses on language syntax and application, while AMPL also delves into mathematical optimization concepts.

3.2. Data management

Data management in AMPL and Julia coincides in the use of sets to organize information. However, there is a relevant difference in code structuring. While Julia programming is done in a single file, AMPL uses three separate files: one for data, one for the model, and one for program execution. Both software platforms define line and bus sets with associated parameters to manage the input data of the modeled power system (Figs. 1 and 2). This helps to simplify programming in both AMPL and Julia.

```
# Sets
# Barras
0b = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
    24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
    46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69]

0l = [(1,2), (2,3), (3,4), (4,5), (5,6), (6,7), (7,8), (8,9), (9,10), (10,11), (11,12), (12,13),
    (13,14), (14,15), (15,16), (16,17), (17,18), (18,19), (19,20), (20,21), (21,22), (22,23), (23,24),
    (24,25), (25,26), (26,27), (3,28), (28,29), (29,30), (30,31), (31,32), (32,33), (33,34), (34,35), (3,36), (36,37),
    (37,38), (38,39), (39,40), (40,41), (41,42), (42,43), (43,44), (44,45), (45,46), (4,47), (47,48), (48,49), (49,50),
    (8,51), (51,52), (9,53), (53,54), (54,55), (55,56), (56,57), (57,58), (58,59), (59,60), (60,61), (61,62),
    (62,63), (63,64), (64,65), (11,66), (66,67), (12,68), (68,69)]

# Datos
PD = Dict(1 => 0.0, 2 => 0.0, 3 => 0.0, 4 => 0.0, 5 => 0.0, 6 => 2.6, 7 => 40.4, 8 => 75.0, 9 => 30.0, 10 => 28.0,
    11 => 145.0, 12 => 145.0, 13 => 8.0, 14 => 8.0, 15 => 0.0, 16 => 45.5, 17 => 60.0, 18 => 60.0, 19 => 0.0, 20 => 1.0,
    21 => 114.0, 22 => 5.3, 23 => 0.0, 24 => 28.0, 25 => 0.0, 26 => 14.0, 27 => 14.0, 28 => 26.0, 29 => 26.0, 30 => 0.0,
    31 => 0.0, 32 => 0.0, 33 => 14.0, 34 => 19.5, 35 => 6.0, 36 => 26.0, 37 => 26.0, 38 => 0.0, 39 => 24.0, 40 => 24.0,
    41 => 1.2, 42 => 0.0, 43 => 6.0, 44 => 0.0, 45 => 39.22, 46 => 39.22, 47 => 0.0, 48 => 79.0, 49 => 384.7, 50 => 384.7,
    51 => 40.5, 52 => 3.6, 53 => 4.35, 54 => 26.4, 55 => 24.0, 56 => 0.0, 57 => 0.0, 58 => 0.0, 59 => 100.0, 60 => 0.0,
    61 => 1244.0, 62 => 32.0, 63 => 0.0, 64 => 227.0, 65 => 59.0, 66 => 18.0, 67 => 18.0, 68 => 28.0, 69 => 28.0)

QD = Dict(1 => 0.0, 2 => 0.0, 3 => 0.0, 4 => 0.0, 5 => 0.0, 6 => 2.2, 7 => 30.0, 8 => 54.0, 9 => 22.0, 10 => 19.0, 11 => 104.0,
    12 => 104.0, 13 => 5.5, 14 => 5.5, 15 => 0.0, 16 => 30.0, 17 => 35.0, 18 => 35.0, 19 => 0.0, 20 => 0.6, 21 => 81.0, 22 => 3.5,
    23 => 0.0, 24 => 20.0, 25 => 0.0, 26 => 10.0, 27 => 10.0, 28 => 18.6, 29 => 18.6, 30 => 0.0, 31 => 0.0, 32 => 0.0, 33 => 10.0,
    34 => 14.0, 35 => 4.0, 36 => 18.55, 37 => 18.55, 38 => 0.0, 39 => 17.0, 40 => 17.0, 41 => 1.0, 42 => 0.0, 43 => 4.3, 44 => 0.0,
    45 => 26.3, 46 => 26.3, 47 => 0.0, 48 => 56.4, 49 => 274.5, 50 => 274.5, 51 => 28.3, 52 => 2.7, 53 => 3.5, 54 => 19.0, 55 => 17.2,
    56 => 0.0, 57 => 0.0, 58 => 0.0, 59 => 72.0, 60 => 0.0, 61 => 888.0, 62 => 23.0, 63 => 0.0, 64 => 162.0, 65 => 42.0, 66 => 13.0,
    67 => 13.0, 68 => 20.0, 69 => 20.0)
```

Figure 1. Power demand dataset of the 33-bus test system in Julia

3.3. Programming the mathematical model

The main difference between AMPL and Julia lies in their model implementation. Julia is a general programming language with various uses, while AMPL belongs to the family of algebraic modeling and is designed for optimization and mathematical models. This is reflected in its syntax, which can directly and easily translate mathematical models. In contrast, as a complete programming language, Julia has a more complex syntax and requires solid programming foundations for effective use. Figs. 3 and 4 show an example of the computational implementation of Eq. (17) in AMPL and Julia.

Note that the translation of the mathematical model in AMPL is closer to the mathematical formulation of the model, whereas, in Julia, the implementation requires adjusting to its syntax.

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

Figure 2. Power demand dataset of the 33-bus test system in AMPL

```
@NLconstraint(m, [i in Ob],
    sum(P[l] for l in Ol if l[2] == i) -
    sum(P[l] + R[l]*I[l]^2 for l in Ol if l[1] == i) +
    Ps[i] == PD[i])
```

Figure 3. Computational implementation of Eq. (17) in Julia

```
subject to BalancePotenciaActiva{ i in Ob }:
sum{ (k,i) in Ol }( P[k,i] ) - sum{ (i,j) in Ol }( P[i,j] + R[i,j] * I[i,j]^2 ) + Ps[i] = PD[i];
```

Figure 4. Computational implementation of Eq. (17) in AMP

3.4. Code length

The code length required for computational implementation is another substantial difference between AMPL and Julia. The structure of AMPL (*i.e.*, one file for data, another for the model, and

another for program execution) makes the code length increase considerably. Furthermore, in AMPL, it is mandatory to declare all parameters in an initial section before being able to use them in the data and model specifications. Julia does not have this limitation; as a programming language, it enables more compact programming, keeping all the logic in a single script, without requiring prior declarations or distribution across multiple files. Nevertheless, of required, the user also has this possibility.

3.5. Solver installation

AMPL and Julia differ in solver management. As shown in Fig. 5, AMPL has a centralized portal to install commercial solvers according to the license type and problem, with integrated descriptions to facilitate selection by the user. Open-source solvers come by default. Julia does not have an analogous portal: commercial solvers are obtained from independent pages and open-source solvers as downloadable libraries. This decentralized diversity implies greater complexity in solver selection, as one must investigate the available alternatives online to evaluate features and limitations on an individual basis. Although the process is more complex, it provides access to a broader diversity of solvers. In AMPL, centralized management facilitates selection but limits the available options.

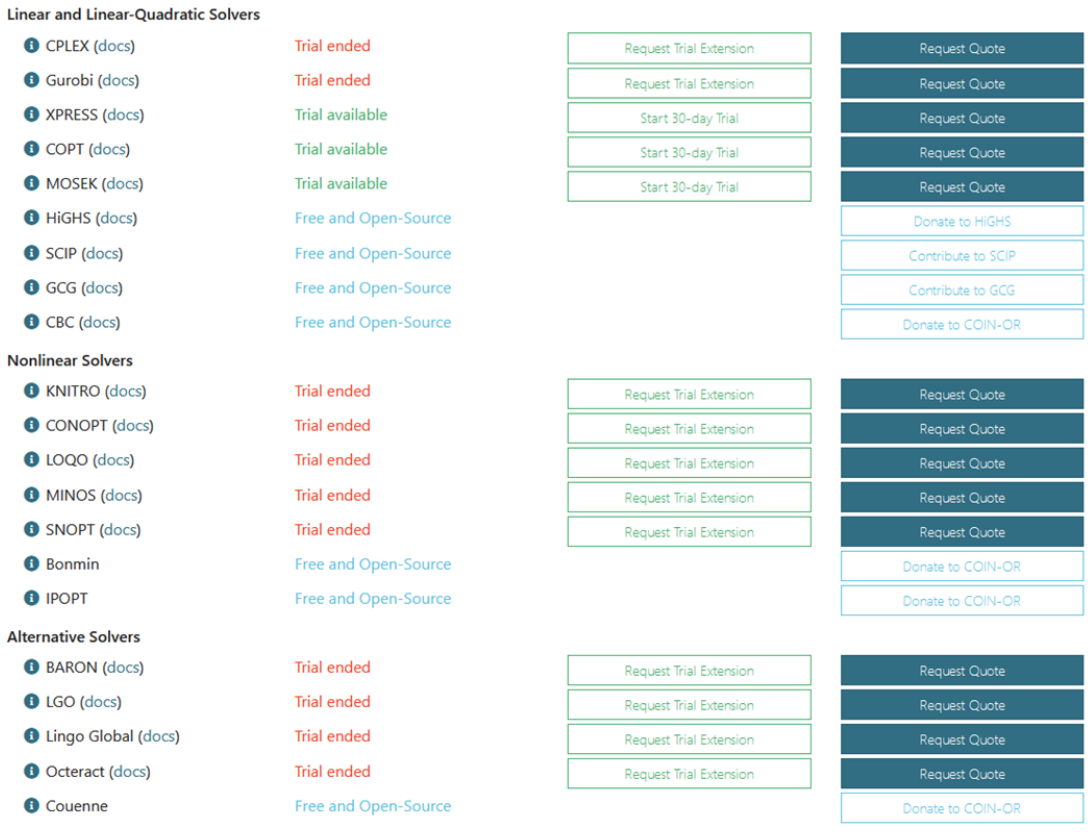

Figure 5. AMPL's centralized portal

4. Testing and results

For computational experimentation, three optimization algorithms were selected: Knitro, which is commercial in nature, and Bonmin and Ipopt, which are open-source. This choice was based on the fact that these three solvers are available in both Julia and AMPL, and they all implement nonlinear optimization techniques. Therefore, they allow for a comparison between platforms in equivalent non-convex problems. The simulations were performed on a personal computer with an Intel Core™ i-3 1215U CPU @ 1.20 Ghz, 8GB RAM, and the default configuration of the solvers.

4.1. Comparison of results

Tables **I**, **II**, and **III** show the results obtained in the test systems with the selected solvers. Differences were observed in the computational time required by the models' execution, being considerably longer in Julia. This occurs in most of the test systems and proposed optimization problems, except for the 83-bus system's optimal capacitor placement. The above denotes the greater computational efficiency of AMPL over Julia when solving the proposed mathematical models.

Table I. Computation times for the power flow model

Test system	Solver	AMPL computation time (s)	Julia computation time (s)
33-bus	Knitro	0,046875	0,82812500
	Bonmin	0,015625	2,10599995
	Ipopt	0,0625	3,96000004
69-bus	Knitro	0,187	0,48437500
	Bonmin	0,265	3,92599988
	Ipopt	0,109	4,21000004
83-bus	Knitro	0,781	1,60937500
	Bonmin	0,11	3,52600023
	Ipopt	1,625	4,31200036

According to Tables **IV**, **V** and **VI**, across all feeders and optimization models tested for optimal capacitor and distributed generation placement, a decrease in active power losses was observed compared to the initial state of the analyzed electrical networks. Likewise, the results indicate economic viability, as reductions were observed in the total annualized costs, which consider both investment and technical losses.

Additionally, the implementation of the proposed mathematical models made it possible to improve the voltage profiles at the system buses, bringing the values closer to the nominal magnitude as well as mitigating voltage drop issues. This was achieved in both AMPL and Julia, confirming the equivalence of the selected solvers regarding operation and accuracy.

Table II. Computation times for the optimal capacitor placement model

Test system	Solver	Capacitor bank (kvar)	Nodal location	AMPL computation time (s)	Julia computation time (s)
33-bus	Knitro	250	14, 30, 32	1,67188	1,375
	Bonmin	250	14, 30, 32	4,9375	8,77300002
69-bus	Knitro	250	61, 62, 64	0,25	1,078125
	Bonmin	250	61, 62, 64	3,422	3,98300004
83-bus	Knitro	250	8, 9, 10, 11, 72	2,39	1,828125
	Bonmin	250	8, 9, 10, 11, 72	26,172	15,53499985

Table III. Computation times for the model concerning the optimal allocation of distributed generation

Test system	Solver	Distributed generation capacity		Nodal location	AMPL computation time (s)	Julia computation time (s)
		P (kW)	Q (kvar)			
33-bus	Knitro	100	42,5	18, 32, 33	1,64	2,245678
	Bonmin	100	42,5	18, 32, 33	5,098	6,359
69-bus	Knitro	100	42,5	63, 64, 65	0,171	0,5
	Bonmin	100	42,5	63, 64, 65	2,359	4,83699989
83-bus	Knitro	100	42,5	5, 6, 7, 8, 9, 10	0,187	0,53125
	Bonmin	100	42,5	5, 6, 7, 8, 9, 10	2,156	5,33300018

Table IV. Power losses and investment costs for the power flow model

Case	Active power losses (kW)	Total cost per year $USD	V_{min}(p.u)
33-bus	202,5994151	34.036,7017	0,91339845
69-bus	224,828	37.771,1	0,90985
83-bus	423,609	71.166,3	0,943849

Based on the results, it should be noted that the Ipopt solver cannot solve optimization problems with binary variables, since it is designed exclusively for continuous nonlinear programming, where decision variables are continuous in nature. Given that the problems regarding optimal capacitor and distributed generation placement involve binary variables related to the installation of capacitor banks and generators, Ipopt is unable to find a feasible solution.

5. Conclusions

- AMPL, as an algebraic modeling language, exhibits a faster learning curve compared to Julia, a general programming language, since its syntax is specifically designed for a direct and straightforward computational representation of mathematical models.
- It is essential to determine the properties of mathematical models before their computational implementation, including characteristics such as linearity and convexity, among others, in order to select solvers that provide optimal and efficient solutions.
- As the size of the test system increased, the evaluated open-source solvers (Bonmin and Ipopt) exhibited longer execution times compared to the Knitro commercial solver. This indicates the superior scalability of the latter. However, since Bonmin and Ipopt are open source, they do not incur licensing costs, unlike Knitro. This makes them viable alternatives based on the available budget, since solution accuracy was not affected. Thereupon, choosing between free or commercial solvers involves a compromise between processing times and the available economic resources.
- The active power losses observed in the analyzed electrical systems were effectively minimized by implementing the proposed mathematical models for optimal capacitor bank placement and distributed generation allocation. It should be noted that these models can be extended to advanced complexity levels as required. In any case, the accuracy of the solutions largely depends on correctly adjusting the mathematical models according to the particularities of the analyzed electrical system.

Table V. Power losses and investment costs for the optimal capacitor placement model

Case	Active power losses (kW)	Total cost per year $USD	V_{min}(p.u)	Reduction in active power losses (%)
33-bus	151,713	27.842,8	0,929413	25,1001618
69-bus	165,187	30.106,4	0,923717	26,52738983
83-bus	392,184	69.811,8841	0,959071	7,42796257

Table VI. Power losses and investment costs for the optimal distributed generation placement model

Case	Active power losses (kW)	Total cost per year $USD	V_{min}(p.u)	Reduction in active power losses (%)
33-bus	155,529	26.162,1	0,928202	23,2626297
69-bus	166,535	28.011	0,926195	25,9278204
83-bus	391,631	65.860,3	0,952438	7,55905944

Nomenclature

Table VII. Sets and indices of the models

Ω_l	Sets of lines
Ω_b	Sets of nodes
i	Index of current node
ij	Index of lines
k	Index of the previous node
j	Index of the next node

Table VIII. Model parameters

R_{ij}	Resistance of the network branch ij
X_{ij}	Reactance of the network branch ij
I_{ij}	Current flowing through the network branch ij
I_{max}	Maximum current flowing through the network branch ij
V_{min}	Minimum voltage of node i in the network
V_{max}	Maximum voltage of node i in the network
P_i^d	Active power demanded at node i of the network
Q_i^d	Reactive power demanded at node i of the network
Q_{base}	Base power of the capacitive units
N_{max}^{ca}	Maximum number of capacitor banks in the system
N_{maxunit}^{ca}	Maximum number of capacitive units at each bus
P_{max}^{gd}	Maximum power of the distributed generation unit
fp	Power factor of the distributed generation unit
N_{max}^{gd}	Maximum number of distributed generation units
Cost^{gd}	Investment cost of a distributed generation unit in \$US/kW
K_c	Cost in \$US per kW-year
K_i	Capacitor bank installation cost in \$US
K_r	Cost of each capacitor bank in \$US per kVAr
D	Depreciation factor

Table IX. Model variables

I_{ij}	Minimum current flowing through branch ij
P_{ij}	Active power in branch ij
Q_{ij}	Reactive power in branch ij
V_i	Voltage at node i of the network
P_i^s	Active power generated at the substation
Q_i^s	Reactive power generated at the substation
Q_i^{ca}	Reactive power injected by the capacitor at node i
W_i^{ca}	Binary variable indicating whether a capacitor is installed at node i
N_i^{ca}	Integer variable indicating how many units can be installed at node i
P_i^{gd}	Active power injected by a distributed generation unit at node i
Q_i^{gd}	Reactive power injected by a distributed generation unit at node i
W_i^{gd}	Binary variable indicating whether distributed generation units are installed at node i

6. CRediT author statement

Jaime Quintero Restrepo: conceptualization, investigation, supervision, writing (review and editing)

Juan Camilo Hoyos Vallejo: investigation, software, writing (original draft)

References

[1] R. Dos Reis Gonçalves, "Modelos de programação linear inteira Mista para resolver problemas de Otimização de sistemas de distribuição de Energia elétrica radiais," PhD thesis, Dept. Electrical Eng., Universidade Estadual Paulista, Ilha Solteira, BRA, 2013. [Online]. Available: https://repositorio.unesp.br/items/bd089af3-8e0b-4ee7-a33b-d7f7a42a4170 ↑3

[2] L. A. Gallego Pareja, J. M. Lopez Lezama, and O. Gomez Carmona, "Optimal placement of capacitors, voltage regulators, and distributed generators in electric power distribution systems," *Ingeniería*, vol. 25, no. 3, pp. 334-354, Oct. 2020. https://doi.org/10.14483/23448393.16925 ↑3, 5, 6

[3] A. Karbowski and K. Wyskiel, "Comparative study of AMPL, Pyomo and JuMP optimization modeling languages on a flood control problem example," *Pomiary Automatyka Robotyka*, vol. 25, no. 4, pp. 19-24, Dec. 2021. https://doi.org/10.14313/par_242/19 ↑3

[4] I. Dunning, J. Huchette, and M. Lubin, "JuMP: A Modeling Language for Mathematical Optimization," *SIAM Rev.*, vol. 59, no. 2, pp-295-320, Sep. 2017. https://doi.org/10.1137/15M1020575 ↑3

[5] M.-B. Lucioograndinetti and A. Editors, "Springer proceedings in mathematics & statistics," 2014. [Online]. Available: http://www.springer.com/series/10533 ↑3

[6] R. Anand, D. Aggarwal, and V. Kumar, "A comparative analysis of optimization solvers," *J. Stat. Manag. Syst.*, vol. 20, no. 4, pp. 623-635, Jul. 2017. https://doi.org/10.1080/09720510.2017.1395182 ↑3

[7] L. A. Gallego, J. M. López-Lezama, and O. G. Carmona, "A mixed-integer linear programming model for simultaneous optimal reconfiguration and optimal placement of capacitor banks in distribution networks," *IEEE Access*, vol. 10, pp. 52655-52673, 2022. https://doi.org/10.1109/ACCESS.2022.3175189 ↑4

[8] J. C. Hoyos Vallejo, "Análisis de herramientas computacionales libres utilizadas en el modelado y en la solución del problema de optimización en la ubicación de capacitores y de la generación en redes de distribución radiales," Undergraduate Thesis, Dept. Engineering, Universidad Autónoma de Occidente, Cali, Colombia, 2023. [Online]. Available: https://red.uao.edu.co/handle/10614/15108 ↑4, 7

[9] *Norma Técnica: Calidad de la potencia de redes de distribución*, Empresas Públicas de Medellín, Colombia, 2019, pp.11-12. ↑5

[10] J. C. Hoyos, "Repositorio código AMPL - JuMP." [Online]. Available: https://github.com/camilohoyos0499/Code ↑7

[11] "Documentation for JuMP," JuMP. [Online]. Available: https://jump.dev/JuMP.jl/stable/ ↑7

[12] R. Fourer, D. M. Gay, and B. W. Kernighan, "*AMPL: A modeling language for mathematical programming, second edition*," 2003. [Online]. Available: https://vanderbei.princeton.edu/307/textbook/AMPLbook.pdf ↑8

Juan Camilo Hoyos Vallejo

Born in Cali (Valle del Cauca, Colombia). Electrical engineer from Universidad Autónoma de Occidente. His research interests include power systems analysis, optimization, and operation.

Email: juan_camilo.hoyos@uao.edu.co

Jaime Quintero Restrepo

PhD in Electrical Engineering from Washington State University (2005). He is currently a full professor at Universidad Autónoma de Occidente (Cali, Colombia). His research interests include the integration of renewable electricity and power systems stability, security, operation, planning, dynamics, and modeling.

Email: jquintero@uao.edu.co