Solving a Classical Optimization ProblemUsing GAMS Optimizer Package:Economic Dispatch ProblemImplementation

1 author:

Oscar Danilo Montoya Giraldo
Universidad Tecnológica de Bolívar
258 PUBLICATIONS   **2,060** CITATIONS

Some of the authors of this publication are also working on these related projects:

Diseño y Operación Óptima de Redes de Distribución Inteligentes View project

Special Issue "Power System Simulation with Renewable Power: Protection, Optimization and Control" Electronics Journal (Ed. MDPI): IF-WOS 2.412, Q2 View project

Solving a Classical Optimization Problem Using GAMS Optimizer Package: Economic Dispatch Problem Implementation

Oscar Danilo Montoya Giraldo[1]

Abstract

This paper presents an effectiveness strategy to model and solve nonlinear mathematical optimization problems in electrical engineering using General Algebraic Modeling System (GAMS) for undergraduate students. A classical problem known as economic dispatch has been selected to show the need of using mathematical tools to solve a large scale optimization problem related with engineering. The economic dispatch is a classical optimization problem in operation of thermal electric systems, being the main idea to find an economical operation for thermal generators. This operation is based on a quadratic cost curve with some operating constraints, i.e., power balance and generation capacities. A numerical simulation is implemented using GAMS optimization package in demo version. This research has been developed with the support of 36 students of the course of Regulation and Operation of Electrical Systems (ROES) in the program of Electrical Engineering at Universidad Tecnológica de Pereira (UTP).

[1] Universidad Tecnológica de Pereira, o.d.montoyagiraldo@ieee.org, `http://orcid.org/0000-0001-6051-4925`, Pereira, Colombia.

Solución de un problema de optimización clásico usando el paquete optimizador GAMS: implementación del problema de despacho económico

Resumen

En este artículo se presenta una estrategia para modelar y resolver problemas no lineales de optimización con estudiantes de pregrado en ingeniería eléctrica empleando el Sistema de Modelado Algebraico General (GAMS). El problema clásico conocido como despacho económico ha sido seleccionado para mostrar la necesidad de usar herramientas matemáticas para resolver problemas de gran tamaño relacionados con ingeniería. El despacho económico es un problema clásico de optimización en operación de sistemas termoeléctricos, cuya idea principal es encontrar la operación más económica para los generadores térmicos. Esta operación está basada en una curva cuadrática de costos con algunas restricciones operativas, como por ejemplo, balance de potencia y capacidades de generación. La simulación numérica es implementada en GAMS usando una versión de prueba. Esta investigación ha sido desarrollada con la ayuda de 36 estudiantes del curso de Regulación y Operación de Sistemas Eléctricos (ROES) del programa de Ingeniería Eléctrica de la Universidad Tecnológica de Pereira (UTP).

Palabras clave: Pensamiento computacional; despacho económico; modelado matemático; paquete de optimización GAMS; problema de optimización no lineal.

1 Introduction

The progress of the technology is always related with the mathematical advances that allows to understand in a better way how the things work [1], [2]. For this reason, in case of undergraduate courses one of the most important focus is the chance to provide to the students the largest amount of tools available to solve engineering and science problems, specifically in areas of applied mathematics. In this way, they can better understand how to analyze real phenomena through simulation scenarios [3], [4].

The simulation tools have become one of the most important components in the integral formation of the future engineers and scientists, because these tools allow to improve their abilities to understand a particular phenomena, and also explore different forms to propose effective solutions for the problem under study. In this context, in recent years many research groups have contributed to develop multiple toolboxes to simulate different areas of the knowledge. In case of electrical engineering it is possible to find specialized software for electrical power system simulation such as DIGSILENT [5], [6], PSCAD [7], NEPLAN [8] or SIMULINK [9], among others. In case of optimization packages, commercial software applications are found, i.e., GAMS [10], [11], [12], AMPL [13], LINGO [14], Python or [15] MATLAB packages [16], [17], [18], etc.

In regards with simulation toolboxes, it is not an easy task to introduce undergraduate students in this type of tools, because it is strictly necessary that the undergraduate student had developed mathematical analysis skills [19], [20], [21], [19], [22]. In addition to this, it is also necessary that they understand whether the results obtained from simulation are correct. For aforementioned reasons, teaching the way on how to use a certain simulation package, it is required that the instructor evaluates the previous students' knowledge, in order to detect failures in their previous formation. If there are failures, then it is necessary to correct them making the undergraduate students ready to understand the basic concepts to use computational tools.

This paper presents basic ideas to introduce undergraduate students in optimization problems and the available tools for their solutions. A GAMS optimization package was selected as optimizer interface, since a demo version for academic purposes is available [23]. The rest of this document is organized as follows: In section 2 basics concepts about computational thinking skills are presented. In section 3 the GAMS optimization package is presented including a numerical example. In section 4 the mathematical formulation of classical economic dispatch problem is shown. In section 5 the GAMS implementation of classical economic dispatch problem is presented. In section 6 the main conclusions derived from this work are presented, following by the acknowledges and references list.

2   Basic knowledge for the future engineers

Considering the high advantages of the technological developments, nowadays it is practically impossible to imagine a world without the presence of the computers [24], [25]. In general a computer can be considered as any electronic device that can be operated using basic commands with the capacity to interact with other electronic devices such as cellphones, tablets or smart-TVs, among others [26], [27]. In this sense, many people around the world have basic knowledge about how to use technological devices. Nevertheless, they do not have the possibility to improve their skills, due to the lack of effective teaching courses to learn how to use these devices efficiently and responsibly, in order to prevent possible health problems.

The case of undergraduate students is not the exception, because the students lack of skills to explode the big computation capacities of modern technological devices such as smart-cellphones, personal computers, among others.

Due to reasons mentioned above, professors in all education centers are called to teach the main advantages of using free software to solve real life problems [28], [29]. In case of undergraduate students of engineering program, the focus is to know the correct way to use computers for solving multiple engineering problems.

To improve the computation abilities in undergraduate students, it is strictly necessary that the professors can find the best way to teach computational thinking [30], [31], [32]. A basic definition of computational thinking would be formulating things with enough clarity, and in a systematic enough way, that one can tell a computer how to do them.

Figure 1 presents the basic ideas to develop computational thinking skills for undergraduate students using an ordered sequence of steps.

The block diagram in Figure 1 presents the main aspects of development computational thinking skills [31]. For this reason, the next items show the most relevant aspects related with each one.

- *Decomposition*: In this step it is very important to split the problem up in small pieces using the available information.

- *Pattern recognition*: This step classifies the information by its parti-

cular characteristics, in other words, here all information is organized through the identification of similar patterns for each component of the problem.

- *Algorithm design*: This step proposes possible solutions of the problem using a set of ordered steps, which have been selected based on the information obtained in the above steps.

- *Review and feedback*: This step is essential in the learning process of the computational thinking skills. Here a full process review is done in order to identify possible mistakes in the aforementioned steps. Then, this information is used to do a feedback in each of them to correct the possible problems.

- *Results analysis*: In this point of the learning process, all students need to have critical abilities to analyze the final solution obtained for the particular problem. In this step the final solution needs to fulfill all problem requirements and be completely feasible.

There will be mistakes in any computational problem implementation, as long as the correct sequence to do the particular process is understood. For this reason, it is really important that professors continuously correct their students during the learning process.

3   GAMS optimization package

General Algebraic Modeling System (GAMS) is one of the most powerful optimizer package available for industrial applications or academic research in areas of applied sciences or mathematics [33]. In case of academical applications, full version can be obtained after paying the license, otherwise the demo version is available with some restrictions of number of variables and solvers. [23].

3.1   Main characteristics of GAMS

The students can easily install demo version from [23] by selecting a operative system. The latest demo version 24.8.3 takes less than 5s to complete
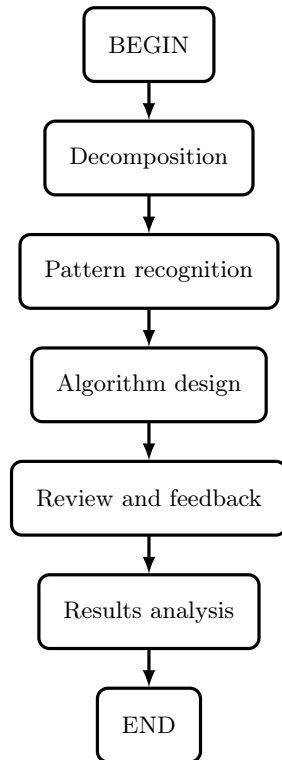
Figure 1: Basic block diagram to understand the correct sequence to teach computational thinking skills

the installation process. The main advantages of using GAMS to implement mathematical optimization models are:

- High computer capacity specifications are not required

- It is easy to download and to install.

- It has good manuals and tutorial, and many books with multiple examples are available.

- The demo version has more than 50 different specialized solvers.

- It is possible to solve, linear or nonlinear models, considering continuous, discrete or binary variables, among other combinations.

These characteristics easily allow to install GAMS optimizer toolbox on personal computers and let the undergraduate students can explore by themselves the main advantages of using structure languages to solve mathematical problems. In the case of the undergraduate students of the course of ROES from electrical engineering program at UTP, the following references have been selected: the GAMS official manual [23] and the book *Building and Solving Mathematical Programming Models in Engineering and Science* [33], to understand the basic details for implementing problems with GAMS optimizer.

3.2 Commands needed to solve an optimization problem in GAMS

The implementation of any optimization problem in GAMS requires that the undergraduate students have basic skills using programming languages like C, C++ or Python, among others [20], [21], [19], [22]. The programming windows of GAMS just receive plain text. In this window, the necessary commands are written to solve any optimization problem [23], [33].

In order to implement a mathematical model using GAMS optimization package the following steps are needed:

1. Define the decision variables using some of these definitions: *variables*, *integer variables*, *positive variables* or *binary variables*. The selection of the type of variables depends on the nature of the optimization problem i.e. binary programming model or mixed-integer nonlinear programming model, among others.

2. Define the set of necessary equations using the reserved word *equations*. First, the names of the equations are defined and then their mathematical expressions are written.

3. Select the name of the model using the reserved word: *MODEL name*.

4. Solve the mathematical model using the next commands: *SOLVE name USING model type MAXIMIZING* or *MINIMIZING objective function variable*.

5. Use the reserve word *DISPLAY* to see the solution of variables of interest.

For more details to use the reserved words and more commands to implement mathematical models in GAMS it is strictly necessary to refer [23].

In the next section a first heuristic example to solve the classical transportation problem using GAMS optimization package will be shown.

3.3 An heuristic example

The next example has been adapted from [33]. Suppose a factory has m production plants and n stores. A single product is shipped from the production plants to the stores. Each production plant has a given level of supply, and each store has a given level of demand. The transportation costs are also given between every pair of production plant and store, and these costs are assumed to be linear. Explicitly, the assumptions are:

- The total supply of the product from production plant i is a_i, where $i = 1, 2, ..., m$.

- The total demand for the product at store j is b_j, where $j = 1, 2, ..., n$.

- The cost of sending one unit of the product from production plant i to store j is equal to c_{ij}, where $i = 1, 2, ..., m$ and $j = 1, 2, ...n$.

The objective is to solve this linear programming problem by finding the minimum transportation cost meeting with all demands in the stores and production capacities in the production plants.

A numerical example is given as follows: $m = 3$ and $n = 2$; $a_1 = 50$, $a_2 = 65$, and $a_3 = 45$; $b_1 = 85$ and $b_2 = 75$; and finally, $c_{11} = 4$, $c_{12} = 3$, $c_{21} = 2$, $c_{22} = 7$, $c_{31} = 4$, and $c_{32} = 6$. With this information the model given by (1) is obtained.

$$\min z = 4x_{11} + 3x_{12} + 2x_{21} + 7x_{22} + 4x_{31} + 6x_{32}$$
$$s.t.$$
$$x_{11} + x_{12} \leq 50$$
$$x_{21} + x_{22} \leq 65$$
$$x_{31} + x_{32} \leq 45 \tag{1}$$
$$x_{11} + x_{21} + x_{31} \geq 85$$
$$x_{12} + x_{22} + x_{32} \geq 75$$
$$\{x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}\} \in \mathbb{N}$$

By using GAMS to implement this mathematical model the result is presented in Algorithm 1.

```
1  */ Transportation problem */
2  */ Variable definition */
3  VARIABLE
4  z;
5  POSITIVE VARIABLES
6  x11, x12, x21, x22, x31, x32;
7  */ Necessary equations */
8  EQUATIONS
9  OF Objective function
10 Const1  Capacity production in the plant 1
11 Const2  Capacity production in the plant 2
12 Const3  Capacity production in the plant 3
13 Const4  Demand in the store 1
14 Const5  Demand in the store 2
15 ;
16 OF..       z =E= 4*x11 + 3*x12 + 2*x21 + 6*x22 +
17 4*x31 + 6*x32;
18 Const1..   x11 + x12 =L= 50;
19 Const2..   x21 + x22 =L= 65;
20 Const3..   x31 + x32 =L= 45;
21 Const4..   x11 + x21 + x31 =G= 85;
22 Const5..   x12 + x22 + x32 =G= 75;
23 */ Solving the model */
24 MODEL TP /all/;
25 SOLVE TP USING mip MINIMIZING z;
26 DISPLAY z.l,x11.l,x12.l,x21.l,x22.l,x31.l,x32.l;
```

Algorithm 1: Heuristic implementation of the transportation problem

ing.cienc., vol. 13, no. 26, pp. 39–63, julio-diciembre. 2017.

47|

The final solution obtained from GAMS after solving the transportation problem is given by Figure 2.

It is important to mention that, in the general solution presented in Figure 2 all variables are integers; specifically, x_{12}, x_{21}, x_{31} and x_{32} take values different from zero and the global minimum of this example corresponds to $z = 510$. Additionally, for this linear-integer programming model there is a unique solution called global optimal solution.

```
1   VARIABLE z.L      =    510.000      VARIABLE x22.L    =     0.000
2   VARIABLE x11.L    =      0.000      VARIABLE x31.L    =    20.000
3   VARIABLE x12.L    =     50.000      VARIABLE x32.L    =    25.000
4   VARIABLE x21.L    =     65.000
```

Figure 2: Solution returned by GAMS for the first heuristic example

Notice that in the implementation presented in Algorithm 1 using GAMS to solve the classical transportation problem all the equations were written one by one. Nevertheless, when the mathematical model starts to increase, using this type of implementation is not efficient, because it is possible to have typo mistakes that could be difficult to find in large codes.

For this reason, the students need to understand how to write mathematical models in GAMS using compact notation. To use the latter to represent any mathematical optimization problem it is strictly necessary to use sets, mathematical functions and sums or products. In case of classical transportation problem, it can be rewritten using compact formulation as given in (2).

$$
\min z = \sum_{i=1}^{m} \sum_{j=1}^{n} C(i,j) x(i,j)
$$
$$
s.t.
$$
$$
\sum_{j=1}^{n} x(i,j) \leq a(i) \quad i = 1, 2, ..., m
$$
$$
\sum_{i=1}^{m} x(i,j) \leq b(j) \quad j = 1, 2, ..., n \tag{2}
$$
$$
x(i,j) \in \mathbb{N} \quad \forall (i,j)
$$

where $C(i,j)$ is the transportation cost to send quantity x of product from the production plant i to store j. The variable $x(i,j)$ corresponds to the

decision variable of the problem (positive variable), $a(i)$ is the maximum capacity of production in the $i - th$ plant and $b(j)$ represents the total demand in the $j - th$ store.

To implement the mathematical model defined by (2) it is necessary to use two new reserved words in the script. The word *SETS* defines the necessary indices in the corresponding decision variables. The word *TABLE* is used to include all numerical information of the problem.

In Algorithm 2 the compact formulation for classical transportation problem is presented .

After solving the compact model presented in Algorithm 2 in GAMS, it is obtained the same solution presented in Figure 2, which obviously implies that both mathematical implementations are equivalents.

The main advantage of a compact formulation is that if the problem size is increased, it is only necessary to increase the numerical information of the problem, due to GAMS automatically includes the number of equations to solve it. This does not happen with the explicit formulation, as it is necessary to expand all the model and is not computationally efficient.

For more information about SETS definition or mathematical functions available in GAMS, refer to [23] and [33].

3.4 General comments

GAMS is a powerful optimization package to solve optimization problems in engineering and science, nevertheless, it is strictly necessary that the undergraduate students account with programming and mathematical skills to represent any optimization problem in equations.

In this context, the following strategy was employed with the students of electrical engineering program at UTP in operation and regulation of electrical power systems course:

1. Explain the basis about optimization problems, questions like: What is optimization?, why is necessary to optimize?, which are the most classical methodologies to solve optimization problems?, among others.

```
1  */ Transportation problem */
2  */ Definition of the Sets */
3  SETS
4  i   set corresponding to the production plants /1*3/
5  j   set corresponding to the stores           /1*2/;
6  */ Definition of the parameters */
7  TABLE C(i,j)
8  1   2
9  1     4   3
10 2     2   6
11 3     4   6;
12 TABLE a(i,*)
13 Prd
14 1    50
15 2    65
16 3    45;
17 TABLE b(j,*)
18 Dmd
19 1    85
20 2    75;
21 */ Variable definition */
22 VARIABLE
23 z;
24 POSITIVE VARIABLES
25 x(i,j);
26 */ Necessary equations */
27 EQUATIONS
28 OF Objective function
29 Const1(i)  Maximum capacity of production in each plant i
30 Const2(j)  Total demanda of products in store j;
31 OF..         z =E= SUM(i,SUM(j,C(i,j)*x(i,j)));
32 Const1(i)..  SUM(j,x(i,j)) =L= a(i,'Prd');
33 Const2(j)..  SUM(i,x(i,j)) =G= b(j,'Dmd');
34 */ Solving the model */
35 MODEL TP /all/;
36 SOLVE TP USING mip MINIMIZING z;
37 DISPLAY z.l,x.l;
```

Algorithm 2: Compact formulation for classical transportation problem

2. Explain how to convert optimization problems in the format of plain information in equations using logic thinking.

3. Show the basic structure of GAMS focused on the syntax.

4. Implement a first mathematical model in GAMS using explicit formulation.

5. Allow the students to explore GAMS and implement their optimization problems using explicit formulations.

6. Explain how to solve the most common errors in GAMS.

7. Show the student how to use compact formulation to reduce codes and possible mistakes.

The most important things in the learning process mentioned above are: the students should not be afraid and to teach the students how to do a good interpretation of the results. It is important to mention that in this teaching-learning process there will have many errors on the way, but it is not possible to forget that it is the way to improve any skill.

4 Mathematical modeling of classical economic dispatch problem

The mathematical model for classical economic dispatch problem considers to obtain all active power generation dispatch for all thermal generators for an specific period of time guaranteeing energy balance on the electrical systems and physical capacities on electric machines. The optimization problem reduces the operation costs of the systems based on a classical quadratic curve to represent generation cost in each thermal generator [34].

In specialized literature there are multiple formulations for classical economic dispatch problem such as secure economic dispatch [35], economic dispatch with exact loss formulation [36], economic dispatch considering greenhouse emission [37] and economic dispatch using optimal power flow representation [38], among others. Nevertheless, to understand the essential analysis for solving nonlinear problems using GAMS optimization

package, the most classical representation of economic dispatch formulation
using unique node representation was selected [34].

For a graphical interpretation on the problem the schematic represen-
tation shown in Figure 3 was considered. In this representation all thermal
generator plants were connected on a equivalent node and all demands
have been aggregated in the same node, in other words, the behavior of the
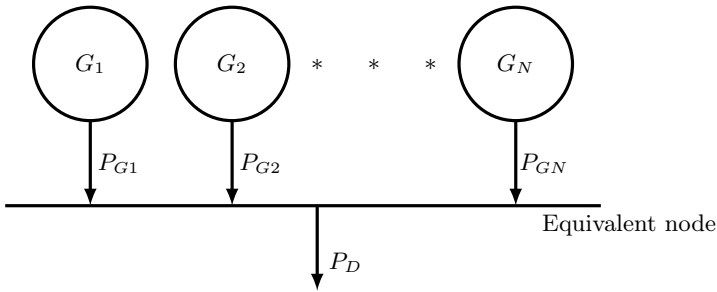electrical network has been removed.

Figure 3: Equivalent node representation of the economic dispatch problem

The mathematical model of the system presented in Figure 3 is de-
fined for one period of time (typically one hour of operation) as shown
in equations (3) to (5). The equation (3) represents the minimum cost
possible to operate the generation electrical system thermal-based and its
formulation is made using quadratic function to represent the cost of gene-
rate active power in thermal plants. This formulation for operating cost is
the most common formulation in economic dispatch problem [34], because
this objective function is a convex function and guarantees a global opti-
mal solution for this problem, if the solution space defines an affine space.
Equation (4) represents an energy power balance in the thermal system, i.e
the first Kirchhoff law applied as an active power function. Finally, in (5)
the minimum and maximum generation limits for each thermal generator
are defined.

$$\min\ C\left(P_G\right) = \sum_{i=1}^{N}\left(\begin{array}{c} aP_G^2\left(i\right) + b\left(i\right)P_G\left(i\right) \\ +c\left(i\right)\end{array}\right) \tag{3}$$

$$\sum_{i=1}^{N} P_G(i) = P_D \tag{4}$$

$$\begin{array}{c} P_G^{\min}(i) \leq P_G(i) \leq P_G^{\max}(i) \\ i = 1, 2, ..., N \end{array} \tag{5}$$

where $C(P_G(i))$ corresponds to the objective function, in this context, it is the total operation cost of all generators to attend the power demand, $P_G(i)$ is the active power generated by $i-th$ thermal generator, $a(i)$, $b(i)$ and $c(i)$ are the coefficients of the quadratic costs representation for the $i-th$ thermal generator, P_D is the total active power demand in the system and finally $P_G^{\min}(i)$ and $P_G^{\max}(i)$ represent the minimum and maximum limits of capacity to generate active power in the $i-th$ thermal generator, respectively.

In order to complement the classical model of economic dispatch problem, in this paper a common way to consider the electrical network effect in this problem is also included. The effect of the electrical network in this problem includes electrical losses in the economic dispatch model. For this reason, to add the loss effect in the model presented, it is just necessary to include it on balance equation (see eq. (4)) as given in (6).

$$\begin{array}{c} \sum_{i=1}^{N} P_G(i) = P_D + P_L(P_G) \\ P_L(P_G) = \sum_{i=1}^{N} \sum_{j=1}^{N} P_G(i) B_L(i,j) P_G(j) \end{array} \tag{6}$$

where $B_L(i,j)$ corresponds to power loss relation between two different generators. Notice that when $i = j$ this parameter represents a quadratic component of the losses caused by the generation of active power with the $i-th$ thermal generator.

To guarantee that all undergraduate students understand the mathematical development of the economic dispatch model, it is strictly necessary that they have some previous skills in mathematical optimization problem formulation and know what an optimization problem is. To ensure that all students overcome this requirement in the first part of the course, an optimization basis were explored using the basic concepts in reference [33],

ing.cienc., vol. 13, no. 26, pp. 39–63, julio-diciembre. 2017.

53|

where there are optimization problems widely explained with their corresponding implementation using GAMS.

5 GAMS implementation

The classical EDP was implemented using GAMS free demo version 24.8.3 available in [23]. To solve the mathematical nonlinear programming model, a large scale nonlinear solver known as KNITRO was used.

Both classical mathematical models were implemented in ROES course at UTP. In case of first mathematical model defined from (3) to (5), the implementation was done during an explanation of how to use GAMS optimizer package. In case of second nonlinear model (mathematical model defined by (3) to (5), (4) was replaced by (6)), the implementation was performed as a group homework.

The parameters of the test system were taken from [34]. This is the reference book for the course. The generators costs data and power limits are shown in Table 1, and the parameters of the power loss relation matrices are defined in Table 2.

Table 2: Parameters associated with power losses

Generator	G_1	G_2	G_3
G_1	0.0001363	0.0000175	0.0001839
G_2	0.0000175	0.0001545	0.0002828
G_3	0.0001839	0.0002828	0.0016147
Units	[1/MW]	[1/MW]	[1/MW]

Table 1: Generator parameters

Generator	a	b	c	$P_G^{\min}$	$P_G^{\max}$
G_1	0.006085	10.04025	136.9125	5	150
G_2	0.005915	9.760576	59.15500	15	100
G_3	0.005250	8.662500	328.1250	50	250
Units	[US\$/MW^{2}h]	[US\$/MWh]	[US\$]	[MW]	[MW]

5.1 First model implementation

In Algorithm 3, GAMS implementation of the classical EDP using compact formulation via set definitions is presented. Notice that this strategy of implementation allows to easily modify GAMS code in regards with the parameters. This also allows to include more generation plants without increasing the number of equations.

```
1  /* Classical EDP */
2  /* Definition of the Sets */
3  SETS
4  G Set of generators /G1*G3/;
5  /* Definition of the Scalars and Parameters */
6  SCALAR
7  PD Active power demand [MW] /210/;
8  TABLE GDATA(G,*)
9  a          b          c          PMIN     PMAX
10 *   [$/MW^2h] [$/MWh]   [$]          [MW]     [MW]
11 G1  0.006085  10.04025  136.9125    5        150
12 G2  0.005915  9.760576  59.15500    15       100
13 G3  0.005250  8.662500  328.1250    50       250;
14 /* Definition of the variables */
15 VARIABLES
16 CPgi Total operation costs [$]
17 P(G) Active power delivered by each generator [MW];
18 /* Operating limits of the variables */
19 P.lo(G) = GDATA(G,'PMIN');
20 P.up(G) = GDATA(G,'PMAX');
21 /* Necessary equations */
22 EQUATIONS
23 OF      Objective function
24 Balance Active power balance;
25 OF..      CPgi =E= SUM(G,GDATA(G,'a')*sqr(P(G)) +      GDATA(
      G,'b')*(P(G)) + GDATA(G,'c'));
26 Balance.. SUM(G,P(G)) =E= PD;
27 MODEL EDP /all/;
28 SOLVE EDp USING nlp MINIMIZING CPgi;
29 DISPLAY CPgi.l, P.l;
```

Algorithm 3: GAMS implementation of classical EDP without considers transmission losses effect

After solving the model for classical EDP presented in Algorithm 3

```
1  VARIABLE CPgi.L = 2544.754 Total operation costs [$]
2  VARIABLE P.L  Active power delivered by each generator [MW]
3         G1 17.299, G2 41.437, G3 151.264
```

Figure 4: GAMS solution of EDP without transmission losses

using GAMS, the results given in Figure 4 are obtained.

When the total power generation given by Figure 4 is analyzed, the third generator supplies the 72.03% of the total demand of the system, followed by the second generator with the 19.64%, while the first generator just attends the 8.33%. This results implies that the global optimal solution of this problem is not restricted by the maximum or minimum capacities of each generator. This means that when the solutions space is analyzed, none capacity constraint has been activated, and the global solution is located inside of the solution space.

5.2   Second model implementation

In case of classical EDP considering transmission losses effect, the same compact formulation presented in the first case is used; including the expression (6) related with the transmission losses effect in the active power balance.

When it is observed the optimal solution of the classical EDP considering transmission power losses effect (Figure 5), it is clear that the total power generation produced by each generator changes drastically in comparison to the ideal case. In this context, the first generator supplies around the 35.86%, while the second plant generates 41.07% and the third generator produces 23.07% of the total generation. It is also possible to observe that the total active power losses caused by the transmission lines are around 4.74%. Additionally, by examining the total power generated by the third generator, the power generation is located at its minimum value, which implies that this generator has the most effect in total operative cost of the system, since this generator has the worst relation between its power generation and technical losses as shown in *TABLE* BL(G,G) in Algorithm 4.

```
1  */ Classical EDP */
2  */ Definition of the Sets */
3  SETS
4  G Set of generators /G1*G3/;
5  */ Duplication of indeces of G set */
6  ALIAS(G,Gd)
7  * Definition of the Scalars and Parameters
8  SCALAR
9  PD Active power demand [MW] /210/;
10 TABLE GDATA(G,*)
11 a         b          c              PMIN    PMAX
12 *   [$/MW^2h] [$/MWh]     [$]             [MW]    [MW]
13 G1  0.006085  10.04025   136.9125    5        150
14 G2  0.005915  9.760576   59.15500    15       100
15 G3  0.005250  8.662500   328.1250    50       250;
16 TABLE BL(G,G)
17 G1          G2          G3
18 *   [1/MW]      [1/MW]      [1/MW]
19 G1  0.0001363  0.0000175  0.0001839
20 G2  0.0000175  0.0001545  0.0002828
21 G3  0.0001839  0.0002828  0.0016147;
22 * Definition of the variables
23 VARIABLES
24 CPgi   Total operation costs [$]
25 P(G)   Active power delivered by each generator [MW];
26 * Operating limits of the variables
27 P.lo(G) = GDATA(G,'PMIN');
28 P.up(G) = GDATA(G,'PMAX');
29 * Necessary equations
30 EQUATIONS
31 OF       Objetive function
32 Balance Active power balance;
33 OF..       CPgi =E= SUM(G,GDATA(G,'a')*sqr(P(G)) +       GDATA(
      G,'b')*(P(G)) + GDATA(G,'c'));
34 Balance.. SUM(G,P(G)) =E= PD + SUM(G,SUM(Gd,P(G)*       BL(G,
      Gd)*P(Gd)));
35 MODEL EDP /all/;
36 SOLVE EDp USING nlp MINIMIZING CPgi;
37 DISPLAY CPgi.l, P.l;
```

Algorithm 4: GAMS implementation of classical EDP considering transmission losses effect

```
1    VARIABLE CPgi.L = 2743.905 Total operation costs [$]
2    VARIABLE P.L Active power delivered by each generator [MW]
3           G1 79.906, G2 90.551, G3 50.000
4    VARIABLE PL.L = 10.457 Active power losses [MW]
```

Figure 5: GAMS solution of EDP considering transmission losses effect

Finally, when the total operative costs of the thermal system are com-
pared, the effect of transmission losses increases up to 7.83%.

5.3   Undergraduate students experience

During the ROES course, there were multiple experiences while the stu-
dents had basic ideas to implement optimization models in GAMS. Some
of them are listed below:

Many students have expressed that GAMS can be employed in different
contexts to analyze real optimization problems that involve high invest-
ments and optimization of natural resources as renewable energies or pro-
duction process in the industry; nevertheless, all of them also expressed that
the most important thing to learn when using GAMS is understand how
to develop mathematical models, which represents adequately the problem
under study.

The students think that the implementation and solution of classical
optimization problems such as optimal power flow or economical dispatch in
GAMS, allows to understand in better way the general concepts of electrical
engineering, specifically, concepts related with planning and operation of
electrical power systems.

Finally, some of them mentioned that thanks to Internet documenta-
tion, tutorial and videos it was possible to understand the structure and
syntax employed by GAMS, to implement large scale optimization pro-
blems.

6   Conclusions

In this paper a GAMS optimization package to solve single-objective optimization problems was presented. GAMS is presented as a toolbox to engage multiple classical engineering problems to introduce undergraduate students in the learning process of computational skills. Additionally, to understand how GAMS receives the information of the problem, the basic commands to implement a mathematical optimization problem was presented, showing the main advantages to use compact mathematical formulations to implement multiple optimization problems.

The approach proposed allows to know basic ideas to use optimization toolboxes like GAMS to solve classical engineering optimization problems using free demo versions for academical purposes. The improvement of the computational skills of the undergraduate engineering students, can carry out the development of practical solutions in their professional lives, specially in the age of information and communication technologies.

Acknowledgments

References

[1] M. D. Shepherd and C. C. van de Sande, "Reading mathematics for understanding-from novice to expert," *The Journal of Mathematical Behavior*, vol. 35, pp. 74 – 86, 2014. 40

[2] G. L. Donohue, "Undergraduate use of complex simulation tools for airspace design," *Computing in Science Engineering*, vol. 5, no. 5, pp. 72–75, Sept 2003. 40

[3] J. H. Chow and K. W. Cheung, "A toolbox for power system dynamics and control engineering education and research," *IEEE Trans. Power Syst.*, vol. 7, no. 4, pp. 1559–1564, Nov 1992. 40

[4] F. Milano, L. Vanfretti, and J. C. Morataya, "An open source power system virtual laboratory: The psat case and experience," *IEEE Trans. Educ.*, vol. 51, no. 1, pp. 17–23, Feb 2008. 40

[5] T. Boonseng and N. Sarasiri, "Dynamic railway simulation using digsilent programming language," in *2016 19th International Conference on Electrical Machines and Systems (ICEMS)*, Nov 2016, pp. 1–4. 41

[6] P. Pradhan, T. Deki, P. Wangmo, D. Dorji, D. Phuntsho, and C. Dorji, "Simulation and optimization of blackstart restoration plan in bhutan using digsilent," in *2015 2nd International Conference on Recent Advances in Engineering Computational Sciences (RAECS)*, Dec 2015, pp. 1–6. 41

[7] M. Brignone, F. Delfino, R. Procopio, M. Rossi, and F. Rachidi, "Evaluation of power system lightning performance, part i: Model and numerical solution using the pscad-emtdc platform," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 1, pp. 137–145, Feb 2017. 41

[8] D. Bica, C. Moldovan, and M. Muji, "Power engineering education using neplan software," in *2008 43rd International Universities Power Engineering Conference*, Sept 2008, pp. 1–3. 41

[9] V. F. Pires and J. F. A. Silva, "Teaching nonlinear modeling, simulation, and control of electronic power converters using matlab/simulink," *IEEE Trans. Educ.*, vol. 45, no. 3, pp. 253–261, Aug 2002. 41

[10] F. Benhamida, I. Ziane, S. Souag, Y. Salhi, and B. Dehiba, "A quadratic programming optimization for dynamic economic load dispatch: Comparison with GAMS," in *3rd International Conference on Systems and Control*, Oct 2013, pp. 625–630. 41

[11] L. Tartibu, B. Sun, and M. Kaunda, "Multi-objective optimization of the stack of a thermoacoustic engine using GAMS," *Appl. Soft Comput.*, vol. 28, pp. 30 – 43, 2015. 41

[12] M. K. Amosa and T. Majozi, "GAMS supported optimization and predictability study of a multi-objective adsorption process with conflicting regions of optimal operating conditions," *Comput. Chem. Eng.*, vol. 94, pp. 354 – 361, 2016. 41

[13] J. A. Gonzalez, "Probabilistic production costing modeled with ampl," *IEEE Trans. Power Syst.*, vol. 17, no. 2, pp. 277–282, May 2002. 41

[14] B. Liu and N. Jin, "An application of lingo software to solve dynamic programming problem in the field of environmental protection," in *2015 IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Dec 2015, pp. 577–580. 41

[15] A. F. J. López, M. C. P. Pelayo, and . R. Forero, "Teaching image processing in engineering using python," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 11, no. 3, pp. 129–136, Aug 2016. 41

[16] W. Chuan, Y. Lei, and Z. Jianguo, "Study on optimization of radiological worker allocation problem based on nonlinear programming function-fmincon," in *2014 IEEE International Conference on Mechatronics and Automation*, Aug 2014, pp. 1073–1078. 41

[17] M. S. Castro, J. T. Saraiva, and J. C. Sousa, "Application of the matlab linprog function to plan the short term operation of hydro stations considered as price makers," in *2016 13th International Conference on the European Energy Market (EEM)*, June 2016, pp. 1–5. 41

[18] Y. Hua and R. V. Iyer, "On the solution of operator equation problems with application to preisach density estimation," in *2016 American Control Conference (ACC)*, July 2016, pp. 2427–2432. 41

[19] S. Nieto and H. Ramos, "Use of a symbolic computation program to reinforce the spatial abilities of engineering students," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, vol. 12, no. 1, pp. 37–44, Feb 2017. 41, 45

[20] J. Holvikivi, "Logical reasoning ability in engineering students: A case study," *IEEE Trans. Educ.*, vol. 50, no. 4, pp. 367–372, Nov 2007. 41, 45

[21] G. Silva-Maceda, P. D. A.-V. na, and F. E. Castillo-Barrera, "More time or better tools? a large-scale retrospective comparison of pedagogical approaches to teach programming," *IEEE Trans. Educ.*, vol. 59, no. 4, pp. 274–281, Nov 2016. 41, 45

[22] A. Gomes, F. B. Correia, and P. H. Abreu, "Types of assessing student programming knowledge," in *2016 IEEE Frontiers in Education Conference (FIE)*, Oct 2016, pp. 1–8. 41, 45

[23] GAMS Development Corp. GAMS free demo version. [Online]. Available: https://www.gams.com/download/ 41, 43, 45, 46, 49, 54

[24] M. Resnick and N. Rusk, "The computer clubhouse: Preparing for life in a digital world," *IBM Syst. J.*, vol. 35, no. 3.4, pp. 431–439, 1996. 42

[25] J. W. Cortada, *How Computers Spread Around the World So Fast.* Wiley-IEEE Press, 2009, pp. 27–70. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6381994 42

[26] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, March 2017. 42

[27] Y. Guo, H. Zhu, and L. Yang, "Smart service system(sss): A novel architecture enabling coordination of heterogeneous networking technologies and devices for internet of things," *China Commun*, vol. 14, no. 3, pp. 130–144, March 2017. 42

[28] G. H. Gaynor, "Solving problems," *IEEE Eng. Manage. Rev.*, vol. 44, no. 4, pp. 6–8, Fourth 2016. 42

[29] P. S. Steif, L. Fu, and L. B. Kara, "Computer tutors can address students learning to solve complex engineering problems," in *2014 IEEE Frontiers in Education Conference (FIE) Proceedings*, Oct 2014, pp. 1–8. 42

[30] K. Yevseyeva and M. Towhidnejad, "Work in progress: Teaching computational thinking in middle and high school," in *2012 Frontiers in Education Conference Proceedings*, Oct 2012, pp. 1–2. 42

[31] P. Silapachote and A. Srisuphab, "Teaching and learning computational thinking through solving problems in artificial intelligence: On designing introductory engineering and computing courses," in *2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, Dec 2016, pp. 50–54. 42

[32] C. Enriquez and O. Aguilar, "Using robot to motivate computational thinking in high school students," *IEEE Lat. Am. Trans.*, vol. 14, no. 11, pp. 4620–4625, Nov 2016. 42

[33] E. Castillo, A. Conejo, P. Pedregal, R. García, and N. Alguacil, *Building and Solving Mathematical Programming Models in Engineering and Science*, ser. Pure and Applied Mathematics: A Wiley Series of Texts, Monographs and Tracts. Wiley, 2001. 43, 45, 46, 49, 53

[34] D. Kothari and J. Dhillon, *Power System Optimization*. Prentice-Hall of India, 2004. 51, 52, 54

[35] C. Shao, X. Wang, M. Shahidehpour, X. Wang, and B. Wang, "Power system economic dispatch considering steady-state secure region for wind power," *IEEE Trans. Sustainable Energy*, vol. 8, no. 1, pp. 268–278, Jan 2017. 51

[36] A. K. Zadeh, K. M. Nor, and H. Zeynal, "Multi-thread security constraint economic dispatch with exact loss formulation," in *2010 IEEE International Conference on Power and Energy*, Nov 2010, pp. 864–869. 51

[37] A. Gholami, J. Ansari, M. Jamei, and A. Kazemi, "Environmental/economic dispatch incorporating renewable energy sources and plug-in vehicles," *IET Generation, Transmission Distribution*, vol. 8, no. 12, pp. 2183–2198, 2014. 51

[38] W. Uturbey and A. S. Costa, "Dynamic optimal power flow approach to account for consumer response in short term hydrothermal coordination studies," *IET Generation, Transmission Distribution*, vol. 1, no. 3, pp. 414–421, May 2007. 51

ing.cienc., vol. 13, no. 26, pp. 39–63, julio-diciembre. 2017.

63|