

## RESEARCH ARTICLE

# Particle swarm grammatical evolution for energy demand estimation

David Martínez-Rodríguez<sup>1</sup> | J. Manuel Colmenar<sup>2</sup> | J. Ignacio Hidalgo<sup>3</sup> |  
Rafael-J. Villanueva Micó<sup>1</sup> | Sancho Salcedo-Sanz<sup>4</sup>

<sup>1</sup>Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Valencia, Spain

<sup>2</sup>Department of Computer Science, Universidad Rey Juan Carlos, Móstoles, Spain

<sup>3</sup>Department of Computer Architecture, Universidad Complutense de Madrid, Madrid, Spain

<sup>4</sup>Department of Signal Processing and Communications, Universidad de Alcalá, Madrid, Spain

## Correspondence

David Martínez-Rodríguez, Instituto Universitario de Matemática Multidisciplinar, Universitat Politècnica de València, Valencia, Spain.  
Email: damarro3@upv.com

## Funding information

Spanish Ministerial Commission of Science and Technology (MICYT), Grant/Award Number: TIN2017-85887-C2-2-P; Ministerio de Ciencia, Innovación y Universidades, Grant/Award Number: PGC2018-095322-B-C22 and RTI2018-095180-B-I00; Comunidad de Madrid y Fondos Estructurales de la Unión Europea, Grant/Award Number: S2018/TCS-4566 and Y2018/NMT-4668; GenObIA-CM, Grant/Award Number: S2017/BMD-3773; Ministerio de Economía, Industria y Competitividad, Grant/Award Number: MTM2017-89664-P

## Abstract

Grammatical Swarm is a search and optimization algorithm that belongs to the more general *Grammatical Evolution* family, which works with a set of solutions called individuals or particles. It uses the Particle Swarm Optimization algorithm as the search engine in the evolution of solutions. In this paper, we present a Grammatical Swarm algorithm for total energy demand estimation in a country from macroeconomic variables. Each particle in the Grammatical Swarm encodes a different model for energy demand estimation, which will be decoded by a predefined grammar. The parameters of the model are also optimized by the proposed algorithm, in such a way that the model is adjusted to a training set of real energy demand data, selecting the more appropriate variables to appear in the model. We analyze the performance of the Grammatical Swarm evolution in two real problems of one-year ahead energy demand estimation in Spain and France. The proposal is compared with previous approaches with competitive results.

## KEY WORDS

energy prediction models, grammatical swarm evolution, macroeconomic variables, total energy demand

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2019 The Authors. *Energy Science & Engineering* published by the Society of Chemical Industry and John Wiley & Sons Ltd.

## 1 | INTRODUCTION

Energy demand estimation from macroeconomic variables is one of the most important indicators of a country's economic growth.<sup>1</sup> Financial crises are known to deeply affect energy demand, producing sudden reductions in the demand of the total energy consumed in a country,<sup>2</sup> as recently occurred during the hard worldwide crisis of 2008.<sup>3,4</sup> In periods of economic bonanza, on the contrary, energy demand usually soars, mainly pushed by the most energy-demanding sectors such as industry and construction.<sup>5</sup> Energy demand is also in close connection with social and environmental issues: Currently, about 80% of the global energy demand is covered by nonrenewable sources such as coal or petroleum. This dependence on nonrenewable resources produces important environmental issues and contributes to global warming and climate change, problems which are currently affecting millions of people, mainly in developing and poor countries.<sup>6</sup> The Paris Agreement, signed in 2015, is a global agreement on the reduction in climate change, which will try to limit the global temperature increase to less than 2 Celsius.<sup>7</sup> Moreover, the effective application of the Paris Agreement should achieve net zero emissions in the second half of this century. This goal will require for sure to deeply reduce emissions by signing countries. Nonetheless, the countries in the Paris Agreement publicly outlined what post-2020 climate actions they intended to take under it, known as intended nationally determined contributions (INDCs). It is easy to understand that some of these INDCs will be political decisions affecting key macroeconomic sectors such as industry and transport, among many others. The idea is to make compatible a correct economic development of a country with the reduction in emissions to fulfill the commitments signed in the Paris Agreement. The correct management of these balance between economic growth and emissions reduction will be, therefore, the guiding line of policy decisions over the next years. In this context, it is clear that obtaining an accurate prediction of medium and long-term energy demand of a country from its macroeconomic indicators will be basic for policy and decision makers.<sup>8</sup>

Previous works on total energy demand forecasting from macroeconomic variables have mainly applied machine learning algorithms, including meta-heuristics techniques and neural computation approaches. The first work dealing with this problem was,<sup>5</sup> where a Genetic Algorithm (GA) was proposed to obtain the parameters of an exponential prediction model for energy demand in Turkey. The model proposed in that work was based on four input macroeconomic variables; Gross Domestic Product or GDP, population, import size, and export size. These four predictive variables have been also used in,<sup>9</sup> where a multilayer neural network was applied to solve a problem of total energy demand estimation in Korea. Other approaches such as Particle Swarm Optimization (PSO) have been proposed for

problems of total energy demand estimation. In,<sup>10</sup> a Particle Swarm approach (PSO) is proposed for this problem in Turkey, including two prediction models, a linear and a quadratic one. Five predictive variables are considered in that work, the ones in<sup>5</sup> plus the country's growth rate in the years of analysis. In,<sup>11</sup> a PSO algorithm is also applied, in this case, to a problem of electricity demand in Turkey, considering linear and quadratic models. Ant Colony Optimization (ACO) has been hybridized with PSO in,<sup>12</sup> also for a problem of electricity demand estimation in Turkey. Other hybrids based on PSO and GA have been proposed in<sup>13-15</sup> applied to problems of energy demand estimation in China. Specifically, in<sup>15</sup> a PSO-GA has been applied to a problem of energy demand forecasting. Three different prediction models, linear, exponential, and quadratic are considered, and the predictive variables are economic growth, population, economic structure, urbanization rate, energy structure, and energy price. Moreover, there are other techniques in,<sup>16-18</sup> where different mixtures between clustering and neural networks in order to predict energy consumption have really good performance applied to industry problems. In,<sup>14</sup> a problem of energy demand future projection has been considered, and in,<sup>13</sup> the energy demand forecasting in the primary sector has been considered. More recently, in<sup>3</sup> a Harmony Search approach with feature selection has been proposed for a problem of energy demand estimation in Spain, and in,<sup>4</sup> a Variable Neighborhood Search (VNS) approach was employed for solving this problem, also comparing the results with that of an Extreme Learning Machine. In,<sup>19</sup> new models for energy demand estimation were proposed based on a Grammatical Evolution approach combined with a Differential Evolution algorithm.

In this paper, we explore the performance of a Grammatical Swarm (GS) approach to predict the total energy demand estimation of countries from macroeconomic variables. Grammatical Evolution is a modeling technique based on genetic Programming. It is divided into two main parts. On one side, there is a set of functions which may explain the process to be modeled. On the other side, there is an optimization engine which finds the best combinations of the set of functions in order to minimize the difference between the real data of the studied process and the data obtained by the model provided by Grammatical Evolution. Grammatical Swarm<sup>20</sup> is an implementation of a Grammatical Evolution,<sup>21</sup> where the optimization engine is a Particle Swarm Optimization (PSO)<sup>22</sup> algorithm. The exploration and intensification phases of the PSO are quite different from the classical grammar evolution based on genetic algorithms, which produces a significant effect in the algorithm's performance. This algorithm has shown to be superior to the classical popular Grammatical Evolution in several problems,<sup>20</sup> including neural network training.<sup>23</sup> In our proposal, each particle of the swarm represents one energy prediction model. Together with PSO, some modifications have been added to the original algorithm in order to avoid the stuck of the optimization process in local minima. The PSO engine

searches for the best model according to the training data available. Note that in this algorithm, the PSO not only selects the best prediction model, including the best predictive variables, but also tunes its parameters. Thus, the idea of the paper is to use the GS algorithm to explore new models for energy demand estimation. Previous works use existing models, mainly linear and exponential models, though alternative nonlinear models have also been considered in the literature.<sup>3</sup> The proposed approach goes a step further, by trying to obtain completely new models using a GS algorithm to construct them and tune their parameters, in the same line as<sup>19</sup> did with a classical GE algorithm. In order to show the performance of our proposal, we have tackled two different data sets related to annual energy prediction in Spain and France in the last 30 years. The Grammatical Swarm has shown an excellent performance obtaining good results in terms of mean average relative error for both problems and showing a robust behavior despite the different distribution of the training data in both problems, escaping from overfitted models. The proposed PSO approach was first tested on a set of benchmark problems, giving better results than other PSOs strategies, when the difficulty of the problem to be optimized increases.

The remainder of the paper has been structured as follows: In Section 2, we present the problem definition and in Section 3, we present the Grammatical Swarm algorithm proposed in

this paper, including a description of the grammar that has been applied and the PSO search engine. Section 4 details the results obtained by the Grammatical Swarm approach in two real problems of annual energy demand estimation in Spain and France, with comparison of the results to previous algorithms in the literature. In Section 5, we discuss the obtained results. Section 6 closes the paper by giving some final conclusions and remarks.

## 2 | PROBLEM DEFINITION

Let us consider a time series  $E = \{E(t)\}_{t=1}^n$  of past energy demands for a given country, with  $n$  discrete values corresponding to different years; and a set of  $m$  predictive variables  $X = \{X_1(t), \dots, X_m(t)\}$ , with  $t = 1, \dots, n$ . A model  $\mathcal{M}$  provides an estimation  $\hat{E}$  for  $E$ , at a prediction time-horizon  $t + 1$  respect to the predictive variables. The problem tackled in this paper consists of finding a mathematical model  $\mathcal{M}$  for the total energy demand estimation prediction, and tuning its parameters, in such a way that it estimates, as accurately as possible, the real energy demand  $E$  at time  $t + 1$  (ie, it obtains, from the set  $X = \{X_1(t), \dots, X_m(t)\}$ ).

The quality of a solution  $S$  in the GS (a given prediction model  $\mathcal{M}$  and its parameters) is evaluated using a given

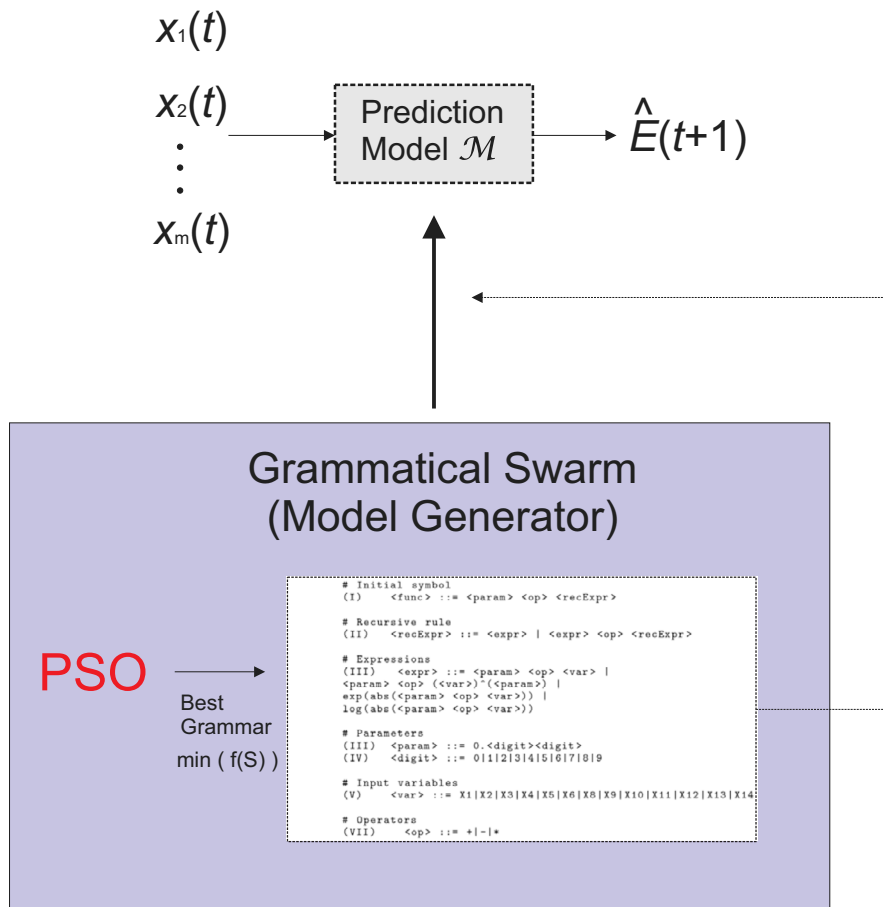


FIGURE 1 Outline of the problem, with GS for evolving the model  $\mathcal{M}$

**FIGURE 2** Grammar applied to the estimation problem for the Spanish data (14 input variables)

```

# Initial symbol
(I) <func> ::= <param> <op> <recExpr>

# Recursive rule
(II) <recExpr> ::= <expr> | <expr> <op> <recExpr>

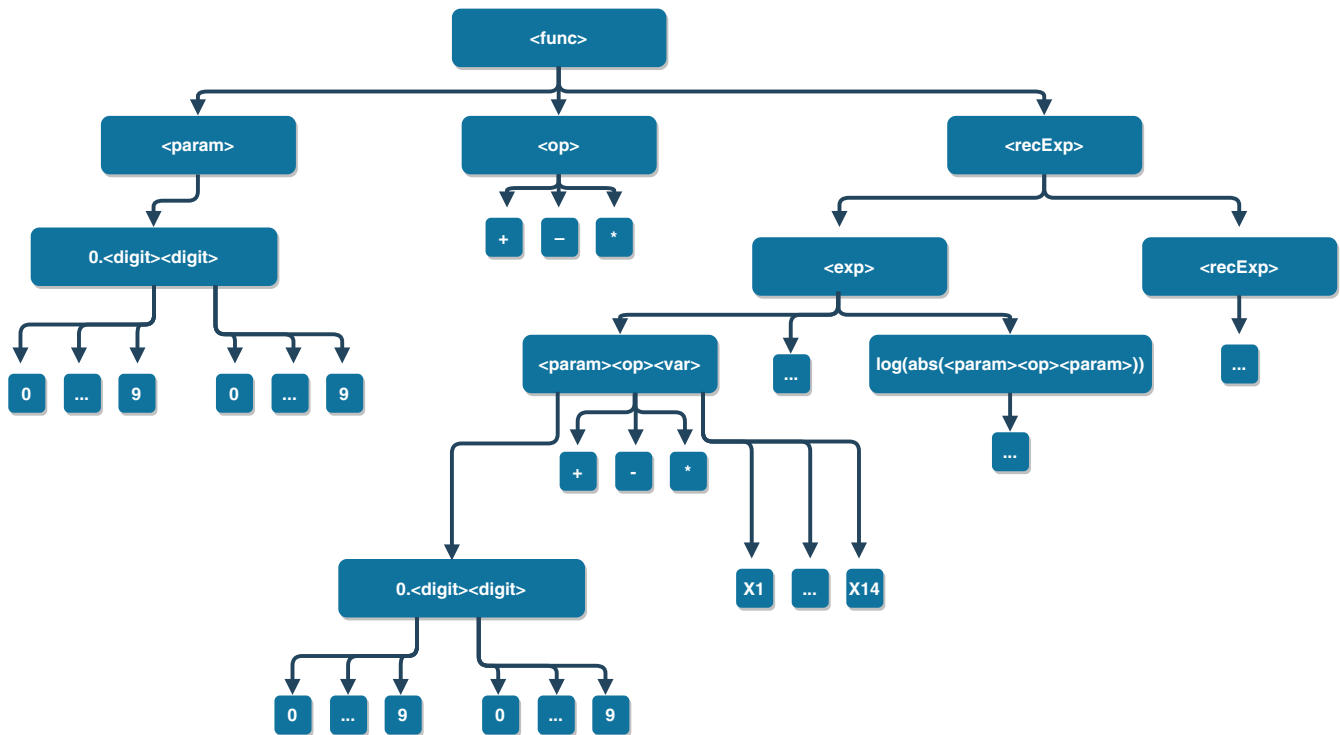
# Expressions
(III) <expr> ::= <param> <op> <var> |
<param> <op> (<var>)^(<param>) |
exp(abs(<param> <op> <var>)) |
log(abs(<param> <op> <var>))

# Parameters
(III) <param> ::= 0.<digit><digit>
(IV) <digit> ::= 0|1|2|3|4|5|6|7|8|9

# Input variables
(V) <var> ::= X1|X2|X3|X4|X5|X6|X8|X9|X10|X11|X12|X13|X14

# Operators
(VII) <op> ::= +|-|*

```



**FIGURE 3** Possibilities of the grammar expansion

objective function, usually related to the similarity of the model output to the real energy demand values. In this case, it is considered that the root-mean-square error between the observed values and the predicted ones, which is to be minimized:

$$f(S) = \sqrt{\frac{1}{n^*} \sum_{j=1}^{n^*} (E(j) - \hat{E}(j))^2}, \quad (1)$$

where  $n^*$  is the size of a reduced training sample ( $n^* < n$ ).

Figure 1 shows an outline of the problem definition, and the role played by the GS in the prediction problem, as generator of the prediction model  $\mathcal{M}$ .

### 3 | PROPOSED GRAMMATICAL SWARM

Swarm intelligence is a computational paradigm that was introduced at the end of the 1980 decade, and published in the early 1990s.<sup>24</sup> It is based on the assumption that artificial intelligence cannot depend solely on individual behavior, but it should take into account the society influence. A swarm is formed by an indeterminate amount of *particles* making elemental actions. All particles interact among them and the environment without a central control, forcing the particles to do any action. Particles are subjected to limited abilities for problem resolution. Nevertheless, the interaction between the different particles which form the swarm allows improving

the individual behavior, obtaining good solutions in many different scenarios.<sup>25</sup>

In this paper, we propose the application of one of the existing swarm algorithms to a real-life problem. Specifically, we apply GS<sup>20</sup> to the problem of annual energy demand estimation from macroeconomic variables. In the rest of this section, we describe in detail the GS algorithm proposed in this paper.

### 3.1 | Grammar production

Grammatical Swarm is a population-based algorithm where each individual of the population is a particle, which represents a solution for a given optimization problem. Specifically, in this case each particle represents a model for energy demand estimation, encoded as a vector of integer numbers that will be decoded by means of a given *grammar*, in the same way as performed in Grammatical Evolution (GE).<sup>21</sup> Figure 2 shows the production rules of the grammar that we have considered in this work, and Figure 3 shows all the possibilities of the grammar expansion. This grammar is taken from,<sup>19</sup> in order to make the results of the GS comparable to those described in that paper.

Table 1 shows an example of a particle decoding, taking into account the grammar shown in Figure 2. Assuming that a particle consists of a vector of integer numbers (named codons) such as (7, 10, 4, 5, 21, 52, 12, 77, 25, 31, 27, 15, 44, 89, 112), shown in the first row of the table, the decoding phase takes the first integer, 7 to decode the `< func >` symbol. The rule in the grammar for that symbol has only one production. Hence, the symbol is translated into `< param > < op > < recExpr >`. The next integer to be processed is 10, which is used to decode the first nonterminal symbol, `< param >`. Again, the rule for this symbol has only one production, `0.< digit >< digit >`, which substitutes the `< param >` symbol. Then, the element 4 is processed to decode the leftmost `< digit >` symbol. The rule for `< digit >` has 10 productions. Given that  $4 \bmod 10 = 4$ , the fifth production of the rule is selected, which corresponds to the value 4. This process continues using the modulus operation till all the nonterminal symbols are decoded. In the example of the table, three remaining integer values are not used for decoding.

In the classical GE implementation, it is very common to consider *wrapping*, which is the feature allowing the return to the first codon if the final one is reached and there are nonterminal symbols not yet decoded. In the example shown above, no wrapping is considered.

### 3.2 | Particle swarm optimization engine

As stated before, GS is a GE implementation where the optimization engine is a Particle Swarm Optimization (PSO) algorithm. PSO is a bio-inspired meta-heuristic algorithm developed

**TABLE 1** Mapping procedure that converts an individual (second column) into an expression. Each row represents a derivation step. The grammar applied is represented in Figure 2

Derivation step	Codons left
<code>&lt; func &gt;</code>	7, 10, 4, 5, 21, 52, 12, 77, 25, 31, 27, 15, 44, 89, 112
<code>&lt; param &gt; &lt; op &gt;</code> <code>&lt; recExpr &gt;</code>	10, 4, 5, 21, 52, 12, 77, 25, 31, 27, 15, 44, 89, 112
0. <code>&lt; digit &gt; &lt; digit &gt;</code> <code>&lt; op &gt;</code> <code>&lt; recExpr &gt;</code>	4, 5, 21, 52, 12, 77, 25, 31, 27, 15, 44, 89, 112
0.4 <code>&lt; digit &gt;</code> <code>&lt; op &gt;</code> <code>&lt; recExpr &gt;</code>	5, 21, 52, 12, 77, 25, 31, 27, 15, 44, 89, 112
0.45 <code>&lt; op &gt;</code> <code>&lt; recExpr &gt;</code>	21, 52, 12, 77, 25, 31, 27, 15, 44, 89, 112
0.45+ <code>&lt; recExpr &gt;</code>	52, 12, 77, 25, 31, 27, 15, 44, 89, 112
0.45+ <code>&lt; expr &gt;</code>	12, 77, 25, 31, 27, 15, 44, 89, 112
0.45+ <code>&lt; param &gt;</code> <code>&lt; op &gt;</code> <code>&lt; var &gt;</code>	77, 25, 31, 27, 15, 44, 89, 112
0.45 + 0. <code>&lt; digit &gt;</code> <code>&lt; digit &gt;</code> <code>&lt; op &gt;</code> <code>&lt; var &gt;</code>	25, 31, 27, 15, 44, 89, 112
0.45 + 0.5 <code>&lt; digit &gt;</code> <code>&lt; op &gt;</code> <code>&lt; var &gt;</code>	31, 27, 15, 44, 89, 112
0.45 + 0.51 <code>&lt; op &gt;</code> <code>&lt; var &gt;</code>	27, 15, 44, 89, 112
0.45 + 0.51 * <code>&lt; var &gt;</code>	15, 44, 89, 112
0.45 + 0.51 * X1	44, 89, 112

by James Kennedy and Russell Eberhart in 1995.<sup>22</sup> It is based on the social behavior of bird swarms, which try to make their movements in the most optimal possible way to find food.

Mathematically, given a target function  $f$ .

$$f: \mathbb{R}^D \rightarrow \mathbb{R}, \quad (2)$$

the optimization problem consists of finding

$$x_{opt} | f(x_{opt}) \geq f(x) \quad \forall x \in \mathbb{R}^D, \quad (3)$$

in the maximization case, and

$$x_{opt} | f(x_{opt}) \leq f(x) \quad \forall x \in \mathbb{R}^D, \quad (4)$$

in the minimization case.

The  $D$ -dimensional domain of function  $f$  in  $\mathbb{R}^D$  is called the search space. Every point in this space is characterized by the  $D$ -dimensional vector,  $x$ , which represents a candidate solution for the problem. These vectors are called particles.<sup>26</sup>

Let  $X$  be a *swarm*, formed by  $N$  particles:

$$X = [x_1, x_2, \dots, x_i, \dots, x_N] \quad (5)$$

where

$$x_i = [x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{iD}] \quad (6)$$

The number  $N$  of particles defined in the swarm depends on the nature of the problem to optimize, but the typical range is between 10 and 60 particles. During the search process, the particles update their position at iteration  $t$ , that is, the values of the elements  $x_i(t)$ , according to the Equation (7):

$$x_i(t+1) = x_i(t) + v_i(t) \quad (7)$$

where  $v_i(t)$  is the velocity of the particle  $i$  at iteration  $t$  and is computed by Equation (8):

$$v_i(t+1) = u(0,1) \cdot w \cdot v_i(t) + u(0,1) \cdot c_1 \cdot (p_i - x_i(t)) + u(0,1) \cdot c_2 \cdot (g - x_i(t)) \quad (8)$$

being  $u(0,1)$  a sample from an uniform distribution between 0 and 1,  $w$  the inertia weight,  $c_1$  the individual weight and  $c_2$  the social weight. Those weights are meta-parameters of the algorithms that we fixed after an experimental setup.  $p_i$  is the the best position reached by the particle  $i$ , and  $g$  is the best position of all the  $N$  particles which form the swarm.

Some modifications have been made to the original GS algorithm in order to adapt its behavior to the problem at hand. In order to avoid the stuck of the particles of the swarm in local minima, two different families of particles are deployed by the algorithm. The first family ( $C$ ) is ruled by Equation (8), and the second family ( $E$ ) is ruled by a modified velocity equation, Equation (2), that repels the particles from the already explored parts of the domain of the function. This modification is based on the calculation of the mass center for the  $t^{\text{th}}$  iteration,  $MC(t)$ , of all the particles that have been deployed in the function, assuming that the behavior of the mass center is equivalent to the behavior of the system and can be computed by Equation (9):

$$MC(t) = \frac{\sum_{i=1}^t \sum_{k=1}^N x_k(i)}{N \times t} \quad (9)$$

---

**Algorithm 1** Implementation of Grammatical Swarm.

---

**Require:**  $w, c_1, c_2, c_3, C, E$

**Ensure:** *Min of Grammar*

```

1: for Every particle  $i = 1, \dots, C$  do
2:   Initialize the position  $x_i$ 
3:   Initialize the best individual position  $p_i = Grammar(x_i(0))$ 
4: end for
5: for Every particle  $i = 1, \dots, E$  do
6:   Initialize the position  $x_i$ 
7:   Initialize the best individual position  $p_i = f(x_i(0))$ 
8: end for
9: Initialize the best global position (shared between both families)  $g = p_i / Grammar(p_i)$  is the minimum.
10: while The stop criteria is not accomplished do
11:   for Every particle  $i = 1, \dots, C$  do
12:     Update velocity according equation (8)
13:     Update position according equation (7)
14:     Evaluate the function  $Grammar(x_i(t+1))$ 
15:     if  $Grammar(x_i(t+1)) \leq Grammar(p_i)$  then
16:        $p_i = x_i(t+1)$ 
17:     end if
18:     if  $Grammar(x_i(t+1)) \leq Grammar(g)$  then
19:        $g = x_i(t+1)$ 
20:     end if
21:   end for
22:   Update the mass centre according to equation (9)
23:   for Every particle  $i = 1, \dots, E$  do
24:     Update velocity according equation (10)
25:     Update position according equation (7)
26:     Evaluate the function  $Grammar(x_i(t+1))$ 
27:     if  $Grammar(x_i(t+1)) \leq Grammar(p_i)$  then
28:        $p_i = x_i(t+1)$ 
29:     end if
30:     if  $Grammar(x_i(t+1)) \leq Grammar(g)$  then
31:        $g = x_i(t+1)$ 
32:     end if
33:   end for
34: end while

```

---

The equation of the velocity that rules the  $E$  family of particles is then the following,

$$v_i^E(t+1) = u(0,1) \cdot w \cdot v_i(t) + u(0,1) \cdot c_1 \cdot (p_i - x_i(t)) + u(0,1) \cdot c_3 \cdot \left( 1 - \left( \frac{x_i(t) - MC(t)}{x_{j,\max} - x_{j,\min}} \right)^2 \right) \cdot (x_i(t) - MC(t)) \quad (10)$$

where  $c_3$  is the weight of the repulsion and  $x_{j,\max}$ ,  $x_{j,\min}$  are the boundaries of the domain for the function  $f$ .

Besides, in order to make possible the integral optimization because of the nature of the problem's encoding (integer values), the result of Equation (7) applied to both families is rounded to the closer integer.

### 3.3 | Final grammatical swarm algorithm proposed

As previously defined, we have a *Grammar Production*, which is the method that allows the automatic creation of mathematical models from the individuals by means of a given grammar, and also a *PSO* algorithm, which allows the optimization of the best grammar by means of swarm intelligence. The combination of both techniques produces *Grammatical Swarm* algorithm, according to.<sup>20</sup> Algorithm 1 shows the pseudo-code for the proposed GS, which includes the modifications explained below. The quality of each particle, that is, the objective function to be minimized, is the root-mean-square error (RMSE) between the real energy demand data and the prediction made by the model generated after the decoding of a particle. Figure 4 shows a diagram of this process.

## 4 | EXPERIMENTS AND RESULTS

This section presents the results obtained applying GS for solving two different problems of energy demand estimation (one-year ahead prediction), Spain and France. We have gathered data from both countries that involve several macro-economic variables. The energy prediction has been carried out separately for each country, so we have tackled two different experiments.

The data sets have been randomly divided into two different sets of values: the training set, used for the obtention of the mathematical model, and the test set, used to check the quality of the model found. This division is made in order to avoid the effect of *overfitting*, where the model is only able to describe the data that has been trained with, instead of showing a good generalizing performance<sup>27</sup> and to test that the forecast is correct. We have considered the same values for the parameters of the

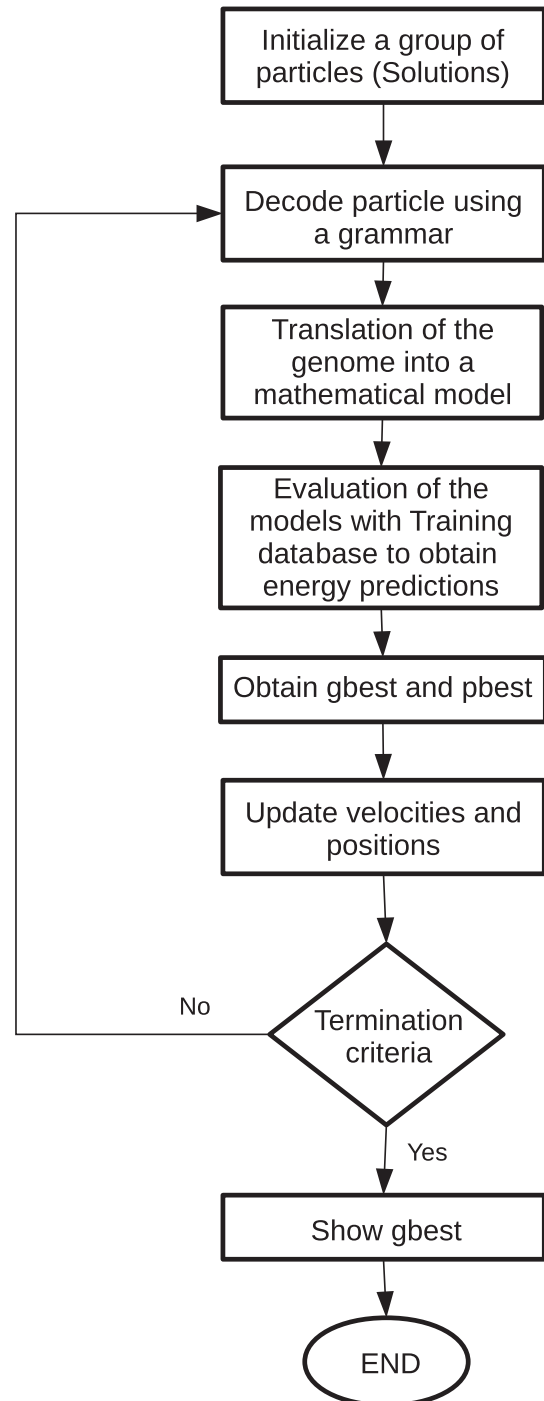


FIGURE 4 Grammatical Swarm flow diagram

PSO in both experiments. These parameters are summarized in Table 2.

In order to perform a comparison with the state-of-the-art techniques, we will run two alternative algorithms on the same data set. These algorithms are a classical implementation of Grammatical Evolution, denoted as *GE*, and a hybridization between *GE* and Differential Evolution (*DE*), denoted as *GE + DE*. Both algorithms follow the implementation described in,<sup>19</sup> where *GE + DE* obtained the best results in the data sets studied in that paper.

**TABLE 2** Parameters of Grammatical Swarm

Parameter	Value
Maximum number of Wraps	1
Chromosome length	300
Number of C particles	60
Number of E particles	30
$w$	0-0.73
$c_1$	0-1.5
$c_2$	0-1.5
$c_3$	0-1.5
Maximum number of iterations	10 000 000

**TABLE 3** Parameters  $GE$ ,  $GE + DE$  and the DE part of  $GE + DE$ 

Parameter	$GE$	$GE + DE$	DE
Population	300	50	75
Generations	1000	40	100
Crossover prob.	0.65	0.65	—
Mutation prob.	0.02	0.02	—
Selection	Tournament	Tournament	—
Mutation Factor $MF$	—	—	0.4717
Recombination Factor $RF$	—	—	0.8803

Table 3 shows the parameter configuration for both  $GE$  and  $GE + DE$  algorithms in the experiments that we have conducted. Notice that the third column shows the values of the parameters that apply in the DE part of  $GE + DE$ .

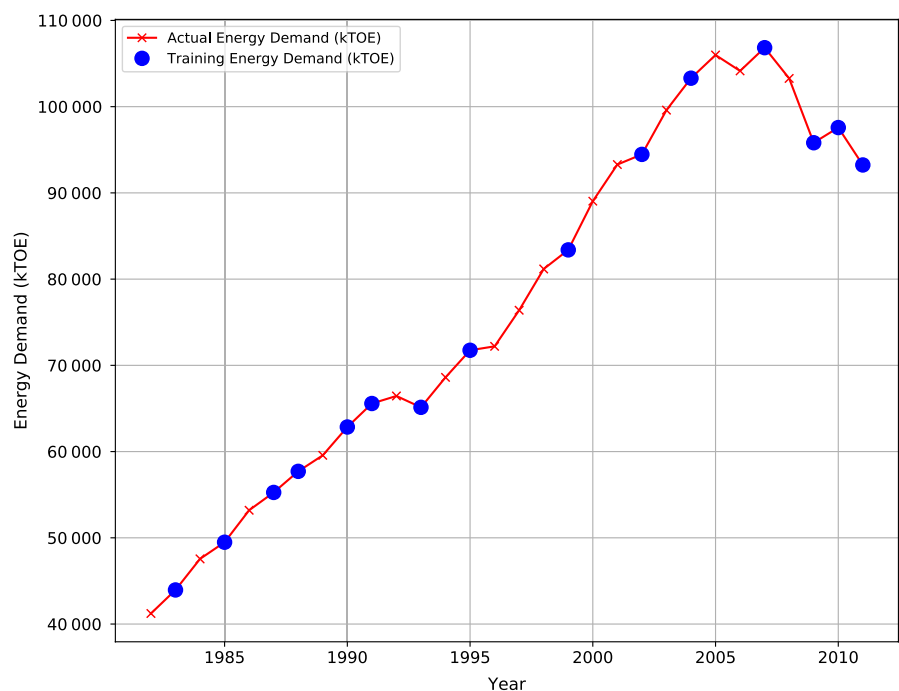
## 4.1 | Data

### 4.1.1 | Energy demand estimation in Spain

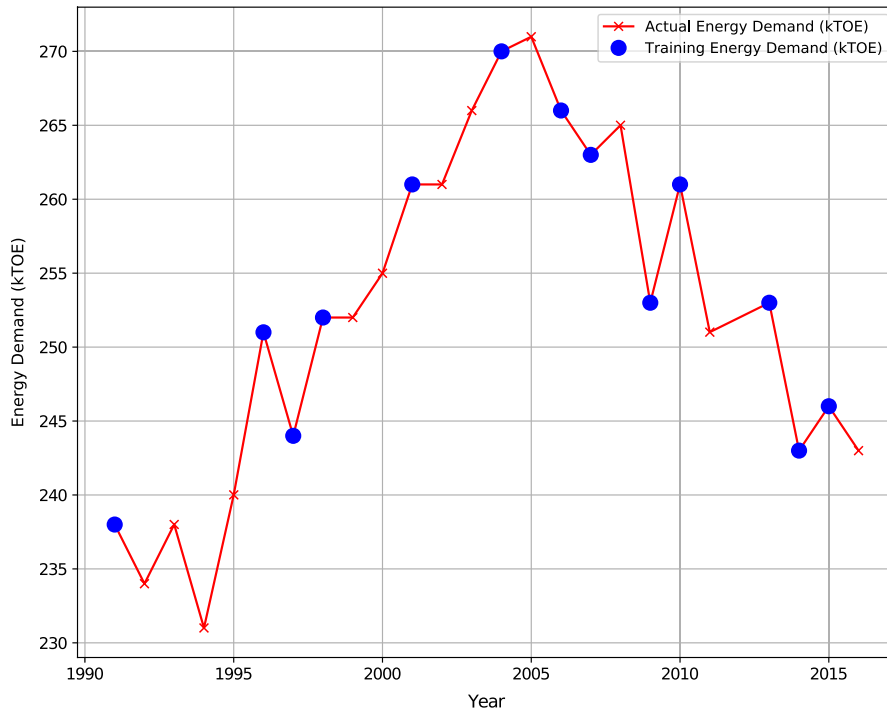
For the estimation of the energy demanded in Spain, we have gathered data of fourteen macroeconomic variables. The full data set ranges from 1980 until 2011. The training set is formed by 15 years chosen with a uniform random sampling. Note that the test set is formed by the remaining 16 years. Figure 5 shows a graphic representation of the data set of actual demand of energy in Spain, highlighting the training data years (blue points).

All the variable values have been scaled to the values of 1980, which present the smallest values. The variables that have been taken into account are the following ones:

1. Gross Domestic Product (€)
2. Population
3. Exports (€)
4. Imports (€)
5. Energy production (kTOE)
6. Electricity power transport (kWh)
7. Electricity production (kWh)
8. GDP per unit of energy use (€ per kTOE)
9. Energy imports net (% use)

**FIGURE 5** Spanish data set. Training data are the blue points



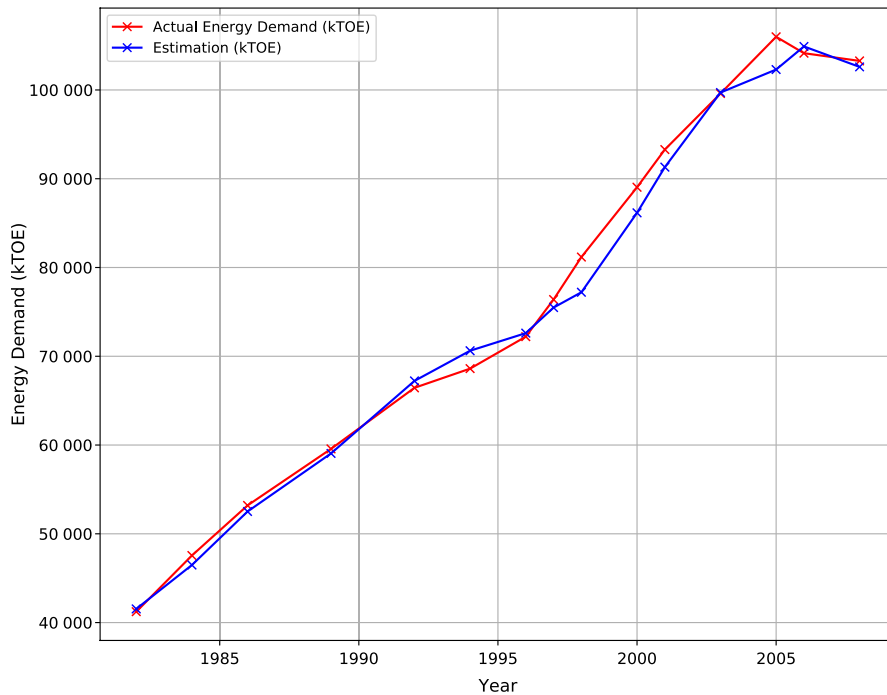


**FIGURE 6** French data set. Training data are the blue points

Country	Spain				France			
	Training		Test		Training		Test	
Algorithm	$\epsilon_R^*$	$\overline{\epsilon}_R$	$\epsilon_R^*$	$\overline{\epsilon}_R$	$\epsilon_R^*$	$\overline{\epsilon}_R$	$\epsilon_R^*$	$\overline{\epsilon}_R$
GE	1.46	3.48	6.53	53.25	0.78	1.44	1.57	3.48
GE + DE	1.66	2.27	11.25	59.39	0.81	1.03	1.66	2.68
GS	1.52	2.37	1.75	3.27	1.33	1.40	2.10	2.74

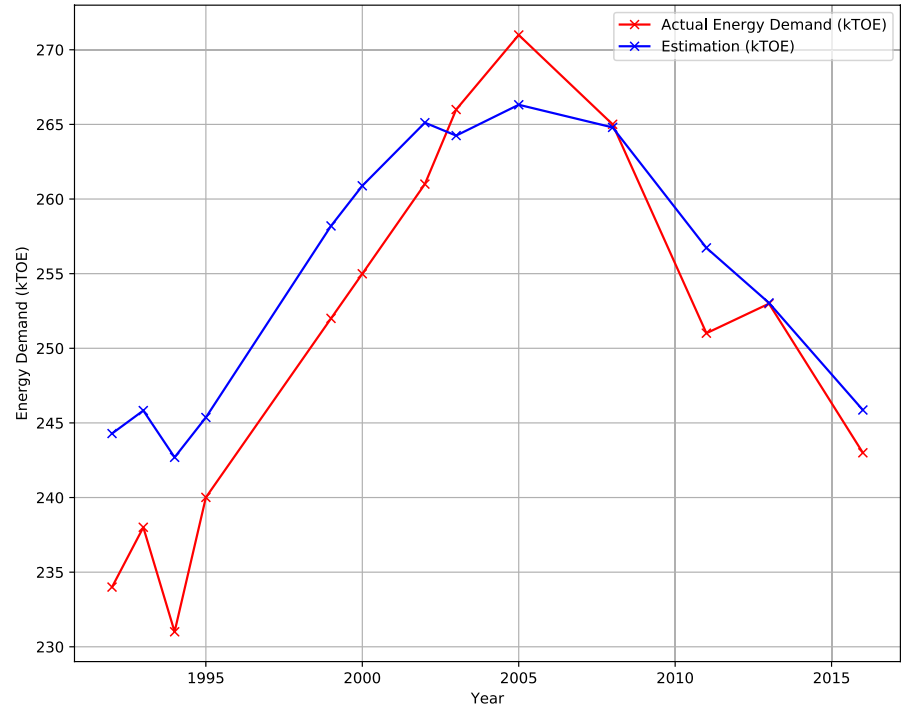
**TABLE 4** Results of energy prediction estimations for training and test data sets in both studied countries

Note: The lower are the error, the better. A high difference between training and test errors means over fitting.



**FIGURE 7** Actual energy demand compared to the prediction made by GS for the test data set of Spain

**FIGURE 8** Actual energy demand compared to the prediction made by GS for the test data set of France



10. Fossil fuel consumption (% total)
11. Electric power consumption (kWh)
12. CO<sub>2</sub> emissions (Mton)
13. Unemployment rate
14. Diesel consumption in road (kTOE)

#### 4.1.2 | Energy demand prediction in France

Similarly as in the previous works in the literature, we have collected a set of values from nine macroeconomic variables from France. In this case, we have data from 1990 until 2016. The training set is formed by 13 years chosen with a uniform random sample. The test set is formed by the remaining 13 years. Figure 6 shows a graphic representation of the data set of actual demand of energy in France, highlighting the training data years.

All the variable values have been scaled to the values of 1990, which present the smallest values. The variables that have been taken into account are the following ones:

1. Energy production (kTOE)
2. Population
3. Gross Domestic Product (€)
4. CO<sub>2</sub> emissions (Mton)
5. Electricity production (TWh)
6. Electricity consumption (TWh)
7. Unemployment rate
8. Usage of renewable energy (% electricity production)
9. Oil demand (Mtons)

The settings of the grammars are the same as in the experiment for the Spanish data. However, given that the data set of France has only nine variables, the grammar has been modified by removing X10 to X14 productions from rule V (see Figure 2).

## 4.2 | Results

We have run 32 executions of the optimization process on the training data set, obtaining a set of 32 models for each one of the algorithms and considering as objective function the minimization of the root-mean-square error (RMSE) between the real data and the data resulting from applying the input parameters to the models obtained during the search process. Each model considers input variables in year  $t$  and produces an estimation of energy in the year  $t + 1$ . Both *GE* and *GS* use the grammar shown in Figure 2, while *GE + DE* uses the same grammar, but including the parameters to be optimized by the DE part, as described in.<sup>19</sup> For the test set, we have calculated the best average relative error ( $\epsilon_R^*$ ) and the mean average relative error ( $\overline{\epsilon_R}$ ). In the calculation of the mean average relative error, we have removed the worst and the best performing models. Table 4 shows the results for the energy demand prediction obtained in the training and test data set by the algorithms under study.

## 5 | DISCUSSION

As can be seen in Table 4, the results obtained by *GS* on the test data of Spain, outperform both *GE* and *GE + DE*

methods, obtaining a reduction of 50% in relation to the mean average relative error of  $GE$ , which is the second best performer. On the other hand, in the test data of France, the behavior of both  $GS$  and  $GE + DE$  is comparable. However, the results of  $GE + DE$  are slightly better.

We observe that the behavior of the swarm intelligence is able to find out good models in both data sets. On the other hand, the  $GE + DE$  and  $GE$  methods are sensitive to the data that are selected for the test. As seen in Figures 5 and 6, the training data from Spain are more consecutively selected, which may have caused an overfitting of the  $GE + DE$  and  $GE$  methods, which obtained good results for the training data. The selected data from France are more sparsely distributed, and  $GE + DE$  is able to capture the behavior of the data.  $GS$ , on the contrary, performs very similarly with both data sets, showing more robustness than the  $GE$  methods and obtaining good results, which are also the best ones in the experiment with the data from Spain. This hypothesis can be confirmed by comparing the training data from the same problem in,<sup>19</sup> where data are more consecutively distributed and both algorithms ( $GE$ ,  $GE + DE$ ) obtain much better results in the test set.

Figure 7 shows a graphic representation of the comparison between the estimation and the actual demand of energy in Spain for the test data set, obtained with the best model from the  $GS$  executions. As it can be seen, the model is able to successfully capture the trends of the actual data.

As in the previous experiment, Figure 8 shows the comparison between the prediction of the best model obtained by  $GS$  and the actual demand of energy in France for the test data set. Again, the behavior of the  $GS$  model is comparable to that of the actual data.

$GS$  technique has shown to be more robust than  $DE$  or  $GE + DE$ . The use of this algorithm is recommended when data provided by the system to be modeled (ie, energy demand) are not uniform.

In future works, different energy fields should be tested. Electricity consumption from the electric network could be an interesting problem to be proposed, since the prediction of future power demand is crucial for a proper system functioning.

## 6 | CONCLUSIONS

Total energy demand estimation from macroeconomic variables is an important problem, deeply related to countries' economic growth measurement, and with consequences in policies for energy efficiency and environment protection. As such, research in developing accurate new models for energy demand estimation has been intense in the last years. In this paper, we have evaluated the performance of a Grammatical Swarm algorithm for generating new energy demand estimation models from macroeconomic predictors.

The proposed algorithm is based on a predefined grammar, able to decode different prediction models evolved with a particle swarm approach. The algorithm is also able to adjust the models to a series of real energy demand data. Once trained, the developed model should be able to reproduce with a good accuracy the series of energy demand from the macroeconomic variables and also estimate new values for the future if needed. We have tested the proposed Grammatical Swarm algorithm in two real problems of one-year ahead energy demand estimation from macroeconomic variables, for the case of Spain and France. In both cases, the proposed approach is able to obtain models with an excellent performance in terms of prediction error, showing robustness with different distributions of training data, and improving the results of previous algorithms in one of the problems.

## ACKNOWLEDGMENTS

This work has been partially supported by the following funding sources: Spanish Ministerial Commission of Science and Technology (MICYT) through project number TIN2017-85887-C2-2-P. Ministerio de Ciencia, Innovación y Universidades under grants ref. PGC2018-095322-B-C22 and RTI2018-095180-B-I00. Comunidad de Madrid y Fondos Estructurales de la Unión Europea with grant ref. S2018/TCS-4566 and Y2018/NMT-4668 (Micro-Stres-MAP-CM) and GenObIA-CM (S2017/BMD- 3773). Ministerio de Economía, Industria y Competitividad MTM2017-89664-P.

## ORCID

David Martínez-Rodríguez  <https://orcid.org/0000-0002-2093-7840>

## REFERENCES

1. Safarzynska K, van den Bergh JCM. Integrated crisisenergy policy: Macro-evolutionary modelling of technology, finance and energy interactions. *Technol Forecast Soc Chang*. 2017;114:119-137.
2. Li F, Song Z, Liu W. China's energy consumption under the global economic crisis: decomposition and sectoral analysis. *Energy Pol*. 2014;64:193-202.
3. Salcedo-Sanz S, Muñoz-Bulnes J, Portilla-Figueras JA, Del Ser J. One-year-ahead energy demand estimation from macroeconomic variables using computational intelligence algorithms. *Energy Convers Manage*. 2015;99:62-71.
4. Sánchez-Oro J, Duarte A, Salcedo-Sanz S. Robust total energy demand estimation with a hybrid Variable Neighborhood Search—Extreme Learning Machine algorithm. *Energy Convers Manage*. 2016;123:445-452.
5. Ceylan H, Ozturk HK. Estimating energy demand of Turkey based on economic indicators using genetic algorithm approach. *Energy Convers Manage*. 2004;45:2525-2537.
6. Shaik S, Yeboah O-A. Does climate influence energy demand? A regional analysis. *Appl Energy*. 2018;212:691-703.
7. United Nations Climate Change Conference. The Paris Agreement. UNCTD XXVII 7.d.

8. Suganthi L, Samuel AA. Energy models for demand forecasting—a review. *Renew Sustain Energy Rev.* 2012;16:1223-1240.
9. Geem ZW, Roper WE. Energy demand estimation of south korea using artificial neural network. *Energy Pol.* 2009;37:4049-4054.
10. Ünler A. Improvement of energy demand forecasts using swarm intelligence: the case of Turkey with projections to 2025. *Energy Pol.* 2008;36:1937-1944.
11. Kiran MS, Özceylan E, Gündüz M, Paksoy T. Swarm intelligence approaches to estimate electricity energy demand in Turkey. *Knowl-Based Syst.* 2012;36:93-103.
12. Kiran MS, Özceylan E, Gündüz M, Paksoy T. A novel hybrid approach based on particle swarm optimization and ant colony optimization to forecast energy demand of Turkey. *Energy Convers Manage.* 2012;53:75-83.
13. Yu S, Wei YM, Wang K. A PSO-GA optimal model to estimate primary energy demand of china. *Energy Pol.* 2012;42:329-340.
14. Yu S, Zhu K, Zhang X. Energy demand projection of China using a path-coefficient analysis and PSO-GA approach. *Energy Convers Manage.* 2012;53(1):142-153.
15. Yu S, Zhu KJ. A hybrid procedure for energy demand forecasting in China. *Energy.* 2012;37:396-404.
16. Geng Z, Zeng R, Han Y, Zhong Y, Hua FU. Energy efficiency evaluation and energy saving based on DEA integrated affinity propagation clustering: case study of complex petrochemical industries. *Energy.* 2019;179:863-875.
17. Han Y, Long C, Geng Z, Zhu Q, Zhong Y. A novel DEACM integrating affinity propagation for performance evaluation and energy optimization modeling: application to complex petrochemical industries. *Energy Convers Manage.* 2019;183:349-359.
18. Han Y, Hao WU, Jia M, Geng Z, Zhong Y. Production capacity analysis and energy optimization of complex petrochemical industries using novel extreme learning machine integrating affinity propagation. *Energy Convers Manage.* 2019;180:240-249.
19. Colmenar JM, Hidalgo JI, Salcedo-Sanz S. Automatic generation of models for energy demand estimation using grammatical evolution. *Energy.* 2018;164:183-193.
20. O'Neill M, Brabazon A. Grammatical swarm: the generation of programs by social programming. *Nat Comput.* 2006;5(4):443-462.
21. O'Neill M, Ryan C. Grammatical evolution. *IEEE Trans Evolutionary Computation.* 2001;5(4):349-358.
22. Kennedy J, Eberhart R. Particle swarm optimization. Vol. 4. Proceedings of ICNN'95 - International Conference on Neural Networks. Perth, WA;1995:1942-1948. <https://doi.org/10.1109/ICNN.1995.488968>
23. Tsoulos IG, Gavrilis D, Glavas E. Neural network construction using grammatical evolution. In *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology*, IEEE: 2005, 827–831.
24. Beni G, Wang J. Swarm intelligence in cellular robotic systems. In Dario P, Sandini G, Aebischer P, eds, *Robots and Biological Systems: Towards a New Bionics?* Berlin, Heidelberg: Springer Berlin Heidelberg, 1993: 703-712.
25. Krause J, Ruxton GD, Krause S. Swarm intelligence in animals and humans. *Trends Ecol Evol.* 2010;25(1):28-34.
26. Marini F, Walczak B. Particle swarm optimization (PSO). A tutorial. *Chemometr Intell Lab Sys.* 2015;149:153-165.
27. Ling CX. Overfitting and generalization in learning discrete patterns. *Neurocomputing.* 1995;8(3):341–347.

**How to cite this article:** Martínez-Rodríguez D, Colmenar JM, Hidalgo JI, Villanueva Micó R-J, Salcedo-Sanz S. Particle swarm grammatical evolution for energy demand estimation. *Energy Sci Eng.* 2020;8:1068–1079. <https://doi.org/10.1002/ese3.568>