Journal of Electrical Engineering & Technology (2020) 15:13–25
https://doi.org/10.1007/s42835-019-00291-x

ORIGINAL ARTICLE

Estimation of Smooth and Non-smooth Fuel Cost Function Parameters Using Improved Symbiotic Organisms Search Algorithm

Yusuf Sönmez[1] · Mesut Unal[2]

Received: 20 February 2019 / Revised: 25 July 2019 / Accepted: 23 September 2019 / Published online: 1 October 2019

Abstract

The improved symbiotic organisms search (R-SOS) Algorithm is proposed to estimate parameters of smooth and non-smooth fuel cost functions for improving the solution accuracy of economic dispatch problems. Determining accurately of fuel cost curve is a crucial task, because they effect directly solution accuracy of economic dispatch and optimal power flow problems. There are two models as smooth and non-smooth forms to describe the input–output characteristics of generators in thermal power plants. This paper presents an implementation of the R-SOS algorithm in order to estimate parameters of these functions. First, second and third order smooth fuel cost functions and non-smooth fuel cost function with valve point effects are used in the study. The estimation problem is described as an optimization one. The R-SOS algorithm is proposed for solving this optimization problem and it minimizes the total error of estimated parameters. The performance of the R-SOS algorithm is tested on four different cases having different fuel types. Results obtained are compared to classical Symbiotic Organisms Search and other meta-heuristic methods and they show that the proposed R-SOS algorithm is favourite model in all test cases for estimating accurately of fuel cost function parameters.

Keywords Improved symbiotic organisms search · Parameter estimation · Fuel cost curve

1 Introduction

There are three main input parameters to the production of electricity energy at cost of production in the power plants. These three parameters are operating cost, ownership cost and construction of power plant. The operating cost is the more crucial of other them. Economic dispatch (ED) and optimal power flow (OPF) are main problems that aim to minimize the operating costs [1–5]. These mathematical formulations could be as smooth or non-smooth forms and they can be described as linear, quadratic and cubic functions for solving fuel cost problems. On the formulation of optimization, there are many parameters such as environmental operating temperature, plant aging and fuel type etc. An accurate estimation of the fuel cost curve parameters of thermal units is the most crucial situation. A powerful convergence of the fuel cost function to the real cost curve by estimating the cost function parameters periodically is important due to improve the last solution accuracy of ED and OPF problems [6].

Many researchers continued their studies on the estimation parameters of fuel cost curve. They used many different methods such as traditional, AI based and meta-heuristic methods. These methods can be listed as least square error (LSE), Gauss–Newton algorithm, Brad algorithms, Marquardt algorithms, Powell algorithm etc. [7, 8]. These estimation techniques can be classified static as least absolute value (LAV) and least square error (LSE) or dynamic as Kalman Filter (KF) [9] and square root filter (SRF) [10]. Although, all these methods have been used for estimation of function parameters accurately and stable, they have failed to estimate the parameters of non-smooth functions [11].

After the development of modern machine learning algorithms such as Particle Swarm Optimization (PSO) [12], Artificial Bee Colony (ABC) [13] and Cuckoo Search [14] (CS), Gravitational Search Algorithm (GSA) [15], Differential Evolution (DE) [16] have been applied for parameter

✉ Yusuf Sönmez
ysonmez@gazi.edu.tr

Mesut Unal
mesut.unal@gazi.edu.tr

[1] Technical Sciences Vocational College, Gazi University, Ankara, Turkey

[2] Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Turkey

estimation of functions. Although, all of these algorithms have produced successful results, and each new method recommended has improved the results of reported previous ones generally, they also have problems in convergence to real values, especially in non-convex function models. In order to achieve better convergence strategy, authors have introduced hybrid meta-heuristic search algorithms like Improved Differential Evolution (IDE) [17]. In that study, the authors were able to successfully predict the parameters of smooth cubic and non-smooth functions by finding the global optimum point.

SOS is a new meta-heuristic optimization method inspired by the symbiotic relationships of living beings in nature [18]. SOS algorithm that simulates the strategies of mutualism, commensalism and parasitism is robust and easy to implement. Moreover it doesn't require tuning parameters.

In this study, Improved Symbiotic Organisms Search (R-SOS) algorithm is presented to estimate parameters of fuel cost curve in different forms. The aim of this study is to investigate the effect of roulette selection method on neighbour search and diversity performance of SOS algorithm and improve search performance in fuel cost function parameter estimation. The estimation problem of parameters of fuel cost function is defined as an optimization one in this study. Aim of this problem is minimizing the total error. Fuel cost functions are calculated in four different cases in both smooth and non-smooth way. The R-SOS algorithm is used to estimate optimally the parameters of these functions. In order to investigate the effectiveness of the proposed R-SOS algorithm, results have been compared with original SOS, Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Cuckoo Search (CS), Gravitational Search Algorithm (GSA), Differential Evolution (DE), Improved Differential Evolution (IDE) and Least Square Error (LSE) algorithms in estimation of parameters of fuel cost function in thermal power systems. The comparison shows that the proposed R-SOS algorithm improves the solution quality in solving parameters estimation of different fuel cost functions.

2 Mathematical Model of Fuel Cost Curve

2.1 Smooth Model

The fuel cost function can be determined as a smooth function for optimizing the ED and OPF problems. This smooth fuel cost curve can be explained with polynomial functions mathematically. This function type can be described as follow

$$FC_j\left(P_{g_j}\right) = a_{0j} + \sum_{j=1}^{N} a_{ij} + P_{g_j} + r_j, \quad j = 1, 2, 3, \ldots M_g \tag{1}$$

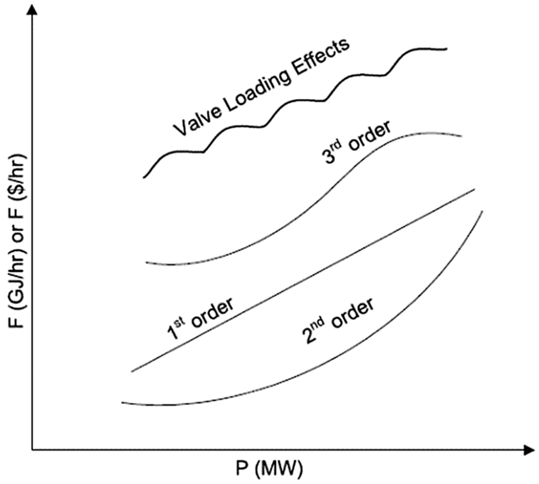

Fig. 1 Smooth and non-smooth fuel cost function curves [13]

where FC_j is the fuel cost function, P_{g_j} is the generating power output in MW, a_{0j} and a_{ij} are cost parameters, r_j is the error value, N is the equation order and Mg is the total number of thermal generators in the power plant.

There are three model as smooth functions, which are first, second and third order. These are called as linear form, quadratic form and cubic form, respectively. The shapes of each form are illustrated in Fig. 1 [6] and these can be formulated as follows.

Linear form (first order model) In this model, N is 1 and Eq. (1) is in the form of:

$$FC_j\left(P_{gj}\right) = a_{0j} + a_{1j} \cdot P_{gj} + r_j \quad j = 1, 2, \ldots, N \tag{2}$$

Quadratic form (second order model) In this model, N is 2 and Eq (1) is in the form of:

$$FC_j\left(P_{gj}\right) = a_{0j} + a_{1j} \cdot P_{gj} + a_{2j} \cdot P_{gj}^2 + r_j \quad j = 1, 2, \ldots, N \tag{3}$$

Cubic form (third order model) In this model, N is 3 and Eq. (1) is in the form of:

$$FC_j\left(P_{gj}\right) = a_{0j} + a_{1j}P_{gj} + a_{2j}P_{gj}^2 + a_{3j}P_{gj}^3 + r_j \quad j = 1, 2, \ldots, N \tag{4}$$

where a_0, a_1, a_2 and a_3 are function parameters, P_{gj} is the output power of generating units r_j is the error value and N is the total number of generating units.

2.2 Non-smooth Model

The input–output curve of steam turbine that generating units tend to have non-smooth. While the input–output curve can be modelled like as heat rate curve that produces have a rippling effect. In this way, the fuel cost curve becomes non-smooth form and consists a sinusoid term in equation [13, 15]. The new cost function becomes as given in Eq. (5):

$$F_i\left(P_{ti}\right) = \left[a_{0i} + \sum_{x}^{L} a_{ji} P_{ti}^{j} + r_i\right] + \left|e_i \sin\left(f_i\left(P_{i,\min} - P_i\right)\right)\right|, \quad i = 1, 2, \ldots, N \tag{5}$$

where e_i and f_i are the fuel cost coefficients of the i th unit with valve point effects.

Surely, the non-smooth function is increasing the accuracy of the economic dispatch (ED) results. At the same time adds more burden on calculation process.

In this study, The proposed R-SOS algorithm has been applied to find the optimal values of smooth and non-smooth function parameters.

In the calculation, the fuel cost function value with estimated parameters has been computed for each cycle and the error value has been found by subtracting this estimated value from real value of fuel cost function [12, 17]

$$r_j = F_{j(actual)} - F_{j(estimated)} \tag{6}$$

The calculation is continued until the absolute summation of error values reaches the smallest acceptable value.

3 Symbiotic Organisms Search Algorithm

3.1 Overview of the SOS Algorithm

The Symbiotic Organisms Search (SOS) algorithm is proposed by Cheng and Prayogo [12]. That proposing provides a simple and powerful metaheuristic algorithm. Generally the Symbiotic Organisms Search (SOS) algorithm works like as communal behaviour between creatures. They do not live alone because that are dependent to other creatures for living in nature. Between both individual species mutual collaboration is called symbiotic. Some of the symbiotic links in nature are mutualism, commensalism and parasitism. In the SOS algorithm the search space is called as ecosystem, and the ecosystem consists organisms. Each organism represents a candidate solution for the problem and has a certain fitness value that indicates the degree of compliance with the desired target. The steps of SOS algorithm are given below.

```
(i) Generate initial ecosystem
(ii)Repeat
    (a)Calculate the fitness value of organisms
    (b)Apply the symbiosis operators (mutualism,
       commensalism and parasitism)
    (c)Update the ecosystem
    (d)If the termination criterion for SOS is satisfied
       then stop process and save the best organism.
       Otherwise, go to step (ii).
```

3.1.1 Generating Ecosystem

At the beginning stage, termination criteria, size of the ecosystem and maximum number of iteration are defined. The organisms are selected randomly to form that ecosystem. Each organism has an attribute vector corresponding a set of the inputs with x_1, x_2, x_3,…, x_n. Moreover, fitness value is represented with a function *f*. At the beginning, the fitness value is calculated with fitness function for each organism. At this step, initial values for each attribute are generating by using random number between lower and upper limits of parameters given as below

$$x_i = a_0^{min} + rand(0,1) \times \left(a_0^{max} - a_0^{min}\right) \tag{7}$$

3.1.2 Calculate Fitness

The fitness value demonstrates the conformity for each organism in ecosystem for the problem. So, the fitness value *f* is calculated from an objective function.

3.1.3 Mutualism Operator

This operator of the SOS algorithm selects two organisms (X_i, X_j) from the ecosystem. Then finds the best organism (X_{best}) and applies the mutualistic relationship between the organisms by using mutual vector and benefit factor given as below [12].

A. The mutual relationship vector (MV) is generated as below

$$MV = (X_i + X_j)/2 \tag{8}$$

B. The best solution (X_{best}) is determined by the fitness values of organisms.

C. Organisms (X_i, X_j) are updated according to Eqs. (9) and (10). BF1 and BF2 are called the “Benefit Factors” and they are used arbitrarily values of 1 or 2

$$X_{inew} = X_i + rand(0,1) \times \left(X_{best} - MV \times BF_1 \right) \tag{9}$$

$$X_{jnew} = X_j + rand(0,1) \times \left(X_{best} - MV \times BF_1 \right) \tag{10}$$

D. The fitness value of the new organisms X_{inew} and X_{jnew} are calculated. Next, if the new values are better than previous values then replace. Otherwise, the new values are not stored.

3.1.4 Commensalism Operator

A. Attribute vector of an organism is randomly selected (X_i) is assigned randomly to X_j, note that $X_i \neq X_j$.
B. Organism X_i is updated by Eq. (11)

$$X_{inew} = X_i + rand(-1,1) \times \left(X_{best} - X_j \right) \tag{11}$$

C. The fitness value of the new organisms X_{inew} is calculated. If the new value is fitter than previous value then replace the value. Otherwise the new value is not stored.

3.1.5 Parasitism Operator

A. Attribute vector of an organism in the ecosystem (X_j) is randomly selected, note that $X_i \neq X_j$.
B. X_j is replaced "Parasite Vector (PV)". The PV is generated by mutation of some attributes of X_j in a range (lower–upper bounds).
C. The fitness value of the new organisms X_j is calculated. If the fitness value (PV) is better than X_j then change organism X_j with PV. If not, keep X_j and remove PV.

3.1.6 Stop

There is termination criteria for stopping the iteration. If the termination criteria meet then X_{best} is saved as the optimum solution. Otherwise, to move Calculate fitness step and the iteration continues.

4 Improved SOS Algorithm (R-SOS)

The exploitation process is carried out during the mutualism phase of the Original SOS algorithm. The j-th solution candidate used in this process is randomly selected from the ecosystem. This random selection is an obstacle to successful execution of the exploitation process. Because the exploitation is a process that requires fine-tuning, this process should be done either around successful solution candidates or around the candidates who can make the most contribution to the search process.

In this study, in the mutualism stage of classical SOS algorithm, SOS algorithm was modified by using probabilistic selection method which is more suitable for exploitation process instead of randomly selected solution candidate. The roulette wheel method is used as the probabilistic selection method. In this way, instead of random selection from the ecosystem, the solution candidate who has a high probability of contributing to the search process is selected. A better exploitation process is realized by using probabilistic selection process depending on the fitness values of the solution candidates. In this way, the algorithm has achieved a better convergence to the global optimum point by getting rid of the local optimum traps.

In the mutualism stage of the classical SOS algorithm, the j-th organism is randomly selected. In the R-SOS algorithm, the j-th organism is determined using the roulette wheel method. According to the roulette wheel selection method, first the f value given in Eq. (12) is calculated using the fitness values of all organisms in the ecosystem. Then, a randomly determined value in the range [0–1] is compared with the total fitness value. According to this comparison, the j-th organism is determined

$$f = \frac{1}{\sum_{i=1}^{n} f_i}. \tag{12}$$

The pseudo-code of the roulette wheel algorithm is given below by Algorithm 1. Where, n is Number of solution candidates in SOS algorithm, k is size of optimization problem X[n,k] is community of solution candidates, f[n] is fitness value of solution candidates, RW[n] is roulette wheel percentage of solution candidates and L[n] is roulette wheel positions of solution candidates.

Algorithm 1. Pseudo-code of Roulette Wheel selection

```
t=0, L[0]=0;
for j=1:n
    t=t+ f[j]
end
for j=1:n
    RW[j]=f[j]/ f[n]
    L[j]=RW[j] + L[j-1]
end
loc=rand (0,1) // turn the roulette wheel and determine the position where the wheel stops
for j=1:n
    if (L[j-1]<loc<= L[j])
        Selected solution candidate =X[j]
end
```

The general flowchart of the proposed R-SOS algorithm is given in Fig. 2.

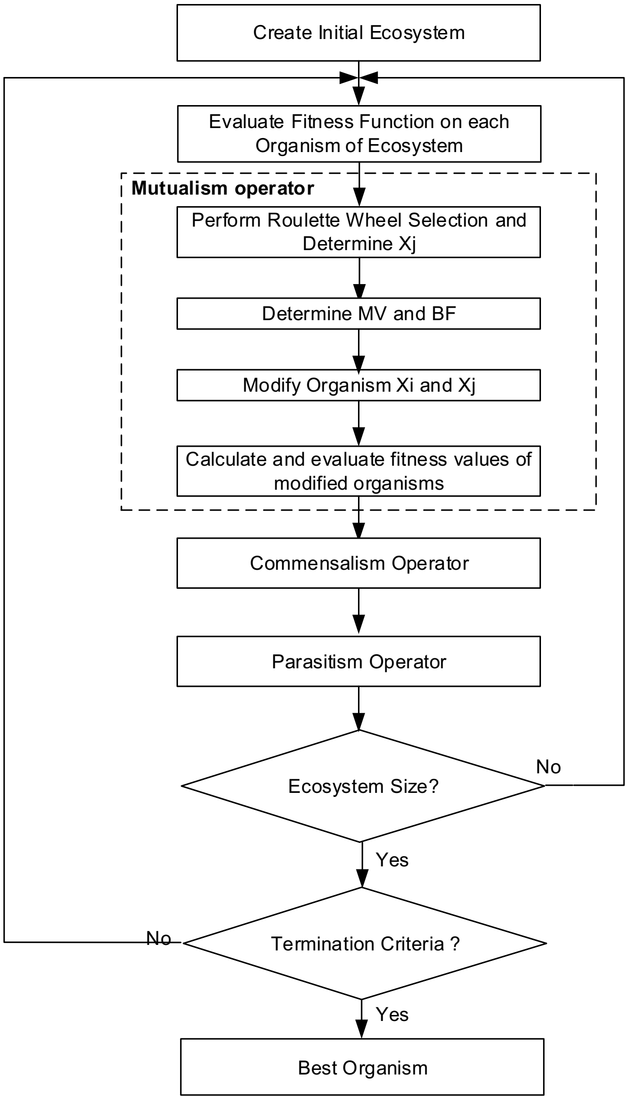
Create Initial Ecosystem
Evaluate Fitness Function on each Organism of Ecosystem
Mutualism operator
Perform Roulette Wheel Selection and Determine Xj
Determine MV and BF
Modify Organism Xi and Xj
Calculate and evaluate fitness values of modified organisms
Commensalism Operator
Parasitism Operator
Ecosystem Size?
No
Yes
Termination Criteria ?
No
Yes
Best Organism

Fig. 2 General flowchart of the R-SOS algorithm

4.1 Implementation of the R-SOS for Parameter Estimation

In this section, implementation of R-SOS algorithm for fuel cost function parameter estimation is given as following steps.

It is aimed in this problem that the most suitable parameters of fuel cost functions are found. Therefore, in the solution, fuel cost function parameters correspond to attributes and the fitness value corresponds to total absolute error value between actual and estimated fuel cost function values as given in Eq. (6) so that it corresponds fitness function for this problem.

In an ecosystem, let the "eco" becomes organism number and the termination criteria is defined to stop the search process. According to these, steps of the proposed R-SOS algorithm can be given as:

```
Operation: SOS
Initialize: Generating ecosystem by using random number between lower and upper limits of parameters by using Eq. (7) (check eco)
While stop conditions are not satisfied do
For $i = 1: eco$
Compute the fitness value of organisms by using Eq. (6) (estimation error)
Obtain the best organism ($X_{best}$)
End for
Implement symbiotic operators by using proposed model given in Sect. 4
Apply Mutualism operator by using Roulette Whell method according to pseudo-code given in Algorithm 1.
Apply Commensalism operator.
Apply Parasitism operator.
End while
Stop the process and save the best organism ($X_{best}$).
```

In this study, the stopping criteria is defined with maximum cycle number (MCN) as 10,000 × d. Where, d corresponds to number of parameters optimized (number of fuel cost function parameters).

5 Results and Analysis

In the experiment, The R-SOS algorithm has been applied to find optimal parameters of fuel cost of a power plant. The power plant has five power generating unit for smooth fuel cost function. In the study, three different fuel types (coal, oil, gas) has been considered and parameters have been found for each. There are four cases that investigated in the experiment. These are linear, quadratic and cubic form of fuel cost smooth function and the non-smooth model has been used.

The implementation of the SOS algorithm has been done by using Matlab Mathworks on an Intel Core-i5 processor personal computer. For each test cases, results have been compared with Least Square Error (LSE) [12], Particle Swarm Optimization (PSO) [12], Artificial Bee Colony (ABC) [13], Cuckoo Search (CS) [14], Gravitational Search Algorithm (GSA) [15], Differential Evolution (DE) [16] and Improved Differential Evolution (DE) [17] methods reported before in order to evaluate the effectiveness of proposed algorithm.

5.1 Smooth Fuel Cost Function Results

5.1.1 Test Case 1: Linear Function

The coefficients for linear equation is given in Table 1.

The first order fuel cost function given in Eq. (2) is used for estimating the parameters for three power plants having coal fuel type, oil fuel type, and gas fuel type. Moreover, all power plants have five generators having power outputs from 10 to 50 MW.

According to the results obtained, estimated parameters obtained from R-SOS, SOS, GSA, CS, ABC PSO and LSE algorithms are given in Table 1. Moreover, the actual and estimated fuel cost values for each unit and for all fuel types obtained from the SOS, GSA, CS, ABC PSO and LSE algorithms, error values are presented in the Table 2. Also, error values for gas fuel type are shown graphically in Fig. 3.

Table 1 Parameters estimated for test case 1

	Algorithms					
	R-SOS/SOS	GSA [15]	CS [14]	ABC [13]	PSO [12]	LSE [12]
C						
a0	45.2	45.1981	43.566	45.212	60.006	63.236
a1	10.56	10.5601	10.597	10.560	10.190	10.170
O						
a0	47.6	47.6359	62.559	47.652	66.001	66.160
a1	11.03	11.0287	10.655	11.031	10.570	10.631
G						
a0	48.4	48.3967	62.899	48.399	66.002	66.700
a1	11.22	11.2204	10.860	11.221	10.780	10.830

Table 2 Results obtained for test case 1

P (MW)	F_{actual} (GJ/H)	$F_{\text{estimated}}$ (GJ/H)	|Error ($F_{\text{actual}} - F_{\text{estimated}}$) (GJ/H)|					
		R-SOS/SOS	R-SOS/SOS	GSA [15]	CS [14]	ABC [13]	PSO [12]	LSE [12]
Unit 1								
Coal								
10	176.62	150.8	25.4	25.820	27.088	25.808	14.715	11.684
20	256.40	256.4	0	0.0004	0.902	0.012	7.403	10.236
30	361.50	362	0.5	0.502	0.036	0.512	4.202	6.836
40	467.60	467.6	0	0.0027	0.170	0.012	0.000	2.436
50	579.50	573.2	6.3	6.296	6.104	6.288	10.002	7.764
$\lvert\sum \text{error}\rvert$			32.62	32.621	34.301	32.632	36.322	38.956
Unit 2								
Oil								
10	184.750	157.9	26.85	26.8266	15.641	26.788	13.049	12.280
20	268.200	268.2	0	0.0107	7.459	0.072	9.200	10.580
30	377.700	378.5	0.8	0.7981	4.509	0.882	5.400	7.390
40	488.800	488.8	0	0.0145	0.041	0.092	0.000	2.600
50	606.000	599.1	6.9	6.9271	10.691	6.798	11.501	8.290
$\lvert\sum \text{error}\rvert$			34.55	34.577	38.341	34.632	39.151	41.140
Unit 3								
Gas								
10	187.200	160.6	26.6	26.5987	15.702	26.591	13.398	12.200
20	272.800	272.8	0	0.0058	7.297	0.019	8.801	10.500
30	384.300	385	0.7	0.7103	4.396	0.729	5.101	7.300
40	497.200	497.2	0	0.0148	0.095	0.039	0.000	2.700
50	616.500	609.4	7.1	7.0807	10.606	7.051	11.501	8.300
$\lvert\sum \text{error}\rvert$			34.4	34.4103	38.096	34.429	35.130	40.928

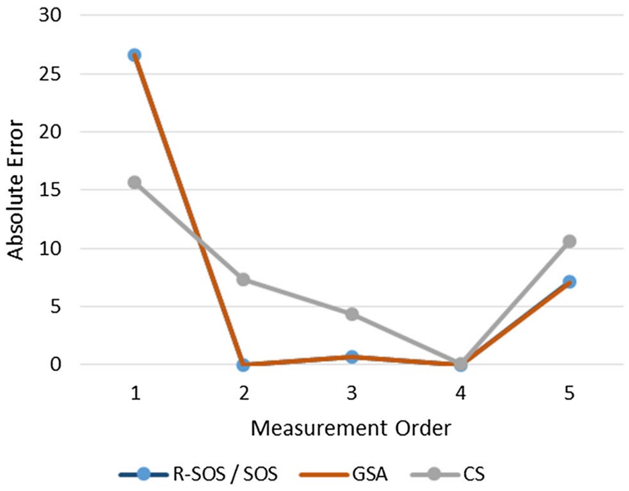

Fig. 3 The linear model error values for gas

As can be seen from Table 2, R-SOS algorithm and SOS algorithm produces same results. The main reason for this is the small number of parameters of the linear function to be estimated. Therefore, the improvement in the R-SOS algorithm could not further reduce the total error value. R-SOS and SOS algorithm reduces the total error value for coal fuel type as 0.421 GJ/H when compared with GSA, as 2.101 GJ/H when compared with the CS, as 0.432 GJ/H when compared with the ABC, as 4.122 GJ/H when compared with the PSO and as 6.756 GJ/H when compared with the LSE. For oil fuel type the reduction is 0.027 GJ/H when compared with GSA, 3.791 GJ/H when compared with the CS, 0.082 GJ/H when compared with the ABC, 4.601 when compared with the PSO and 6.59 GJ/H when compared with the LSE. The reduction is 0.0103 GJ/H for gas fuel type when compared with GSA, 3.696 GJ/H when compared with the CS, 0.029 GJ/H when compared with the ABC, 0.73 GJ/H when compared with the PSO and 6.528 GJ/H when compared with the LSE.

As can be seen that R-SOS algorithm column is our proposal values. The R-SOS algorithm provides the close values to real for all power plants having different fuel types. It is totally clear that R-SOS and SOS algorithm approximates to the actual values closer than other algorithms.

5.1.2 Test Case 2: Quadratic Function

The coefficients for quadratic equation is given in Table 3.

The second order fuel cost function given in Eq. (3) is used for estimating the parameters for three power plants having coal fuel type, oil fuel type, and gas fuel type. Moreover, all power plants have five generators having power outputs from 10 to 50 MW.

The estimated coefficients of the cost function obtained with the R-SOS and SOS algorithm, DE, GSA, CS, ABC, PSO and LSE algorithms are shown in Table 3. Estimated and actual fuel cost values for each unit obtained from proposed R-SOS, SOS, DE, GSA, CS, ABC, PSO and LSE algorithms, and error values are given in Table 4 for all fuel types. Moreover, error values for gas fuel type obtained from SOS, GSA and CS are shown and compare graphically in Fig. 4.

According to Table 4, the proposed R-SOS and SOS algorithm produces same result just same as in case 1. Again, because of the small number of parameters, for the quadratic function with three parameters, R-SOS did not reduce the total error value compared to SOS and DE. However, The R-SOS algorithm can reduce the total error for coal fuel type as 0.0262 GJ/H when compared with GSA, as 0.34 GJ/H when compared with the CS, as 0.05 GJ/H when compared with the ABC, as 0.357 GJ/H when compared with the PSO and as 4.448 GJ/H when compared with the LSE. For oil fuel type, the reduction is 0.1188 GJ/H when compared with GSA, 0.6938 GJ/H when compared with the CS, 0.1578 GJ/H when compared with the ABC, 1.8748 GJ/H when compared with the PSO and 4.4888 GJ/H when

Table 3 Parameters estimated for test case 2

Unit number	Params	Algorithms						
		R-SOS/SOS	DE [16]	GSA [15]	CS [14]	ABC [13]	PSO [12]	LSE [12]
Coal	a0	96.6	96.6	96.5023	96.540	96.6046	96.279	95.856
	a1	7.588	7.588	7.5932	7.575	7.5874	7.592	7.374
	a2	0.0414	0.041400	0.0413	0.042	0.0414	0.042	0.047
Oil	a0	101.5312	101.531250	101.5021	100.887	101.5360	101.000	100.710
	a1	7.88	7.88	7.8751	7.890	7.8779	7.800	7.670
	a2	0.0441875	0.04418	0.0443	0.045	0.0442	0.046	0.049
Gas	a0	101.8125	101.8125	101.02	99.239	101.8179	102.000	101.100
	a1	8.1	8.1	8.1433	8.138	8.0991	7.900	7.881
	a2	0.043875	0.043875	0.0433	0.045	0.0439	0.048	0.049

Table 4 Results obtained for test case 2 (Quadratic Model)

P (MW)	F_{actual} (GJ/H)	$F_{estimated}$ (GJ/H)		Error ($F_{actual} - F_{estimated}$) (GJ/H)					
		R-SOS/SOS	R-SOS/SOS	DE [16]	GSA [15]	CS [14]	ABC [13]	PSO [12]	LSE [12]
Unit 1									
Coal									
10	176.62	176.62	0	0	0.0520	0.140	0.001	0.262	2.368
20	256.40	264.92	8.52	8.52	8.5006	8.400	8.513	8.365	5.568
30	361.50	361.5	0	0	0.0001	0.000	0.013	0.000	2.496
40	467.60	466.36	1.24	1.24	1.2334	1.020	1.259	1.038	2.240
50	579.50	579.5	0	0	0.0001	0.540	0.025	0.452	1.536
$\lvert\sum \text{error}\rvert$			9.76	9.76	9.7862	10.100	9.81	10.117	14.208
Unit 2									
Oil									
10	184.750	184.75	0	0	0.0645	0.502	0.015	1.150	2.404
20	268.200	276.8062	8.6062	8.606	8.5337	8.325	8.574	7.200	5.662
30	377.700	377.7	0	0	0.0534	0.018	0.047	1.300	2.442
40	488.800	487.4312	1.3688	1.368	1.3757	0.973	1.428	2.200	2.266
50	606.000	606	0	0.001	0.0667	0.851	0.069	0.000	1.690
$\lvert\sum \text{error}\rvert$			9.9752	9.975	10.094	10.669	10.133	11.850	14.464
Unit 3									
Gas									
10	187.200	187.2	0	0	0.4145	2.055	0.599	1.420	2.376
20	272.800	281.3625	8.5625	8.563	8.4150	7.311	8.560	6.321	5.568
30	384.300	384.3	0	0	0.0086	0.163	0.001	2.278	2.568
40	497.200	496.0125	1.1875	1.187	1.1339	0.023	1.178	2.716	2.284
50	616.500	616.5	0	0	0.0124	2.869	0.023	0.007	1.420
$\lvert\sum \text{error}\rvert$			9.75	9.75	9.9844	12.421	10.361	12.741	14.216

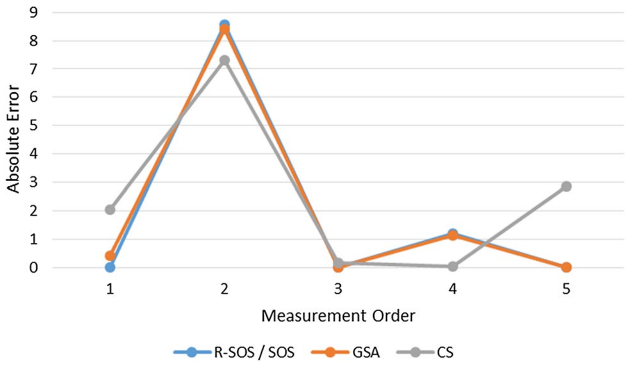

Fig. 4 The quadratic model error values for gas

compared with the LSE. For gas fuel type, the reduction is 0.2344 GJ/H when compared with GSA, 2.671 GJ/H when compared with the CS, 0.611 GJ/H when compared with the ABC, 2.991 GJ/H when compared with the PSO and 4.466 GJ/H when compared with the LSE. The R-SOS and SOS algorithms provides the closed values to real cost of fuel for coal, oil and gas power plants. It is totally clear that R-SOS and SOS algorithms produces close values to the actual when compared other algorithms reported.

5.1.3 Test Case 3: Cubic Function

The third order fuel cost function given in Eq. (4) is used for estimating the parameters for three power plants having coal fuel type, oil fuel type, and gas fuel type. Moreover, all power plants have five generators having power outputs from 10 to 50 MW.

Results of the SOS are compared to classical SOS, IDE, DE, GSA, ABC, PSO and LSE with tables and graphics. The estimated parameters of the cubic cost function obtained from SOS algorithm, and values reported before for GSA, ABC PSO and LSE algorithms are shown in Table 5. Results consisting actual and estimated fuel cost values for each unit; obtained from proposed algorithm and classical SOS algorithm are given in Table 6. Error values are also given in same table. Moreover, error values are shown in Fig. 5.

As can be seen from Table 6, when results are compared obtained from proposed R-SOS and classical SOS algorithm, it can be seen that the R-SOS algorithm reduces the total error value as 0.0429 GJ/H for coal fuel type, as 0.0076

Table 5 Parameters estimated for test case 3

Unit number	Parameters	Algorithms							
		R-SOS	IDE [17]	DE [16]	SOS	GSA [15]	ABC [13]	PSO [12]	LSE [12]
1	a0	127.066845813399	127.0666667	127.0666667	126.953423	122.589143	124.5362	120.241	123.180
	a1	3.11863847927159	3.11866666	3.1186667	3.1322176	3.76799765	3.4859	3.979	3.535
	a2	0.19993460753054	0.19993337	0.1999333	0.19947435	0.17754168	0.1872	0.184	0.193
	a3	− 0.0016266823088267	−0.00162667	−0.0016267	−0.00162185	−0.001402744	−0.0015	−0.002	−0.002
2	a0	132.499951039167	132.5000000	132.500	132.479226	129.879253	129.2351	130.278	128.640
	a1	3.33250685174167	3.3325000	3.3325000	3.33517466	3.67100800	3.4859	3.542	3.746
	a2	0.20587478822323	0.2058750	0.2058750	0.20579572	0.19461388	0.1872	0.200	0.199
	a3	−0.00166249810749064	−0.0016625	−0.00166625	−0.00166186	−0.00155171	−0.0015	−0.002	−0.002
3	a0	132.333239572659	132.3333295	132.3333295	132.412454	129.532636	126.0143	128.376	128.400
	a1	3.62501332596882	3.6250006	3.6250006	3.61208247	3.96083217	3.8044	4.146	4.046
	a2	0.202416221063094	0.2024166	0.2024166	0.20301881	0.19311198	0.1896	0.188	0.195
	a3	−0.00162499568513628	−0.0016250	−0.0016250	−0.00163252	−0.00155213	−0.0015	−0.002	−0.002

Table 6 Results obtained for test case 3 (cubic model)

P (MW)	F_{actual} (GJ/H)	R-SOS	SOS	Error ($F_{actual} - F_{estimated}$) (GJ/H)							
				R-SOS	IDE [17]	DE [16]	SOS	GSA [15]	ABC [13]	PSO [12]	LSE [12]
Unit 1											
Coal											
10	176.62	176.62	176.6012	0	0	0	0.0188	0.0005	0.0048	0.186	0.393
20	256.40	256.40	256.4127	0	0	0	0.0127	1.3438	0.7342	4.157	1.874
30	361.50	356.6467	356.6569	4.8533	4.8533	4.854	4.8431	3.9575	4.4068	0.451	1.779
40	467.60	467.60	467.6027	0	0	0.002	0.0027	0.0000	0.1078	3.846	3.368
50	579.50	579.50	579.5189	0	0	0.004	0.0189	0.0001	0.1688	0.000	2.915
$\lvert\sum$ error$\rvert$				4.8533	4.8533	4.860	4.8962	5.302	5.4224	8.641	10.329
Unit 2											
Oil											
10	184.750	184.750	184.7487	0	0	0	0.0013	0.2510	0.0109	0.674	0.449
20	268.200	268.2000	268.2061	0	0	0	0.0061	0.5312	0.9631	0.000	1.362
30	377.700	372.8750	372.8804	4.8250	4.8249	4.825	4.8196	4.4344	4.1929	4.690	3.477
40	488.800	488.800	488.8003	0	0	0	0.0003	0.0081	0.0289	0.063	0.716
50	606.000	606.000	605.9948	0	0	0	0.0052	0.0001	0.0449	0.119	5.055
$\lvert\sum$ error$\rvert$				4.8250	4.8249	4.825	4.8326	5.2248	5.2407	5.547	11.059
Unit 3											
Gas											
10	187.200	187.200	187.2026	0	0	0	0.0026	0.3000	0.0167	0.099	0.396
20	272.800	272.800	272.8015	0	0	0	0.0015	0.7770	1.8323	1.526	1.888
30	384.300	379.3834	379.4138	4.9166	4.9167	4.917	4.8862	4.0492	3.7387	3.300	1.848
40	497.200	497.200	497.2446	0	0	0	0.0446	0.4085	0.0297	0.874	3.296
50	616.500	616.500	616.4986	0	0	0	0.0014	0.1626	0.1593	0.000	2.720
$\lvert\sum$ error$\rvert$				4.9166	4.9167	4.917	4.9363	5.6973	5.7767	5.799	10.148

GJ/H for oil fuel type and 0.0197 GJ/H for gas fuel type by comparing with classical SOS. The cubic test function has four parameters. This result indicates that the improved R-SOS algorithm can perform better estimation in functions with a large number of parameters providing more effective convergence. Moreover, when the Table 6 is investigated, it is seen that, it is seen that the R-SOS algorithm produces almost the same results with the IDE algorithm and reaches the same total error value. While both algorithm produce same result for coal fuel type, IDE produces better result for oil fuel type as 0.0001 GJ/H and R-SOS better result for gas fuel type as 0.0001 GJ/H.

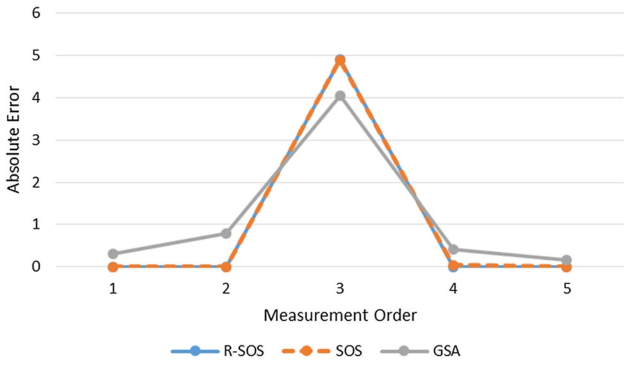

Fig. 5 The cubic model error values for gas

When the rest of results are investigated from Table 6, the R-SOS algorithm can reduce the total error for coal fuel type as 0.4487 GJ/H by comparison with GSA, as 0.5691 GJ/H by comparison with the ABC, as 3.7877 GJ/H by comparison with the PSO and as 5.4328 GJ/H by comparison with the LSE. For oil fuel type, reduction is 0.3998 GJ/H by comparison with GSA, 0.4157 GJ/by comparison with the ABC, 0.722 GJ/H by comparison with the PSO and 6.234 GJ/H by comparison with the LSE method. The reduction is 0.7807 GJ/H for gas fuel type by comparison with GSA, 0.8601 GJ/H by comparison with the ABC, as 0.8824 GJ/H by comparison with the PSO and as 5.2314 GJ/H by comparison with the LSE.

Moreover, the proposed R-SOS algorithm runs for 100 times in order to evaluate the robustness of it. Thus, minimum error, maximum error, mean error and standard deviation values are obtained. These values for cubic function form are given in Table 7. When the Table 7 is investigated, it can be clearly seen that the proposed R-SOS algorithm have less minimum, maximum, mean error and standard deviation values by producing almost same reults for all runs. Thus, it produces more efficient results than classical SOS.

5.2 Non-smooth Fuel Cost Function Results

In this case, parameters of this function type given in Eq. (5) is estimated. In order to evaluate the proposed algorithm, two thermal units are tested. The Unit 1 consists 21 generators having power output from 0 to 500 MW and The Unit 2 consists same number generators having power output from 0 to 360 MW.

The results obtained from the proposed R-SOS algorithm are compared to classical SOS, IDE, DE, CS and PSO algorithms in this case. The estimated parameters of the non-smooth fuel cost function obtained with the R-SOS algorithm, are shown in Table 8 by comparing others. Actual and

Table 7 Statistical results obtained for test case 3 (cubic model)

Algorithm	Fuel type	Minimum error	Maximum error	Mean error	Standard deviation
R-SOS	Coal	4.8533	4.8533	4.8533	1.8127×10^{-6}
SOS		4.8962	4.9003	4.8975	1.526×10^{-3}
R-SOS	Oil	4.8250	4.8250	4.8250	1.6215×10^{-6}
SOS		4.8326	4.8397	4.8348	1.453×10^{-3}
R-SOS	Gas	4.9166	4.9167	4.9166	9.293×10^{-7}
SOS		4.9363	4.9411	4.9384	1.841×10^{-3}

Table 8 Estimated parameters of the non-smooth function

Unit number	Parameters	Algorithms				
		R-SOS	IDE [17]	SOS	CS [14]	PSO [12]
1	a_0	550	550.000000004	550	551.129	548.921
	a_1	8.09999937916374	8.099999462	8.100	8.100	8.096
	a_2	0.000279999042814207	0.000279999	0.00028	0.000	0.000
	a_3	300.000200131087	300.000184360	300.002	298.990	301.424
	a_4	74.00535	637.07990147	81.8421	99.561	0.035
2	a_0	308.999625863596	308.999625883	308.99962586	309.0015	308.410
	a_1	8.10000301071587	8.100003011	8.1000030107	8.09	8.107
	a_2	0.000559989876851128	0.000559990	0.00055998987679	0.000586	0.001
	a_3	200.000481972466	200.000481965	200.00048196	200.7088	200.611
	a_4	267.593577429857	421.189948503	236.53411364	382.6182	0.042

Table 9 Results obtained for Unit 1 in test case 4 (Thermal Unit 1 with power output ranging from 0 to 500 MW)

Unit 1	F_{actual} (GJ/H)	$F_{estimated}$ (GJ/H)		Error ($F_{actual} - F_{estimated}$) (GJ/H)				
P (MW)		R-SOS	SOS	R-SOS	IDE [17]	SOS	CS [14]	PSO [12]
0	550.000	550	550	0	0.0000	0	0.205	0.196
25	982.938	982.9382	982.93818	0.00019	0.0002	0.00018	0.036	0.007
50	1250.896	1250.89594	1250.89593	0	0.0001	0	0.009	0.010
75	1307.251	1307.251113	1307.25111	0.00011	0.0001	0.00011	0.043	0.055
100	1468.035	1468.03499	1468.03499	0	0	0	0.051	0.055
125	1849.962	1849.961767	1849.96176	0.00023	0.0002	0.00024	0.005	0.004
150	2028.980	2028.980377	2028.98037	0.00037	0.0004	0.00037	0.003	0.019
175	2023.333	2023.33327	2023.33277	0.00023	0.0002	0.00023	0.033	0.075
200	2378.296	2378.29600	2378.29600	0	0	0	0.011	0.020
225	2686.609	2686.60874	2686.60874	0.000253	0.0003	0.00026	0.009	0.006
250	2779.917	2779.91702	2779.91703	0	0	0	0.001	0.033
275	2858.341	2858.34097	2858.34097	0	0	0	0.019	0.035
300	3269.109	3269.108681	3269.10867	0.000318	0.0003	0.00033	0.010	0.004
325	3490.738	3490.738089	3490.73808	0	0.0001	0	0.017	0.009
350	3512.636	3512.635417	3512.63542	0.00058	0.0006	0.00,058	0.006	0.040
375	3785.882	3785.88218	3785.88218	0.00018	0.0002	0.00018	0.006	0.008
400	4131.982	4131.98202	4131.98201	0	0	0	0.023	0.003
425	4265.053	4265.0530	4265.05299	0	0	0	0.024	0.012
450	4264.307	4264.30700	4264.30700	0	0	0	0.002	0.015
475	4698.816	4698.8156	4698.81561	0.00037	0.0004	0.00039	0.024	0.008
500	4962.688	4962.6874	4962.68739	0.00059	0.0006	0.00061	0.034	0.006
$\lvert\sum \text{error}\rvert$				0.003681	0.0037	0.00372	0.572	0.619

Table 10 Results obtained for Unit 2 in test case 4 (Thermal Unit 2 with power output ranging from 0 to 360 MW)

Unit 2	F_{actual} (GJ/H)	Error ($F_{actual} - F_{estimated}$) (GJ/H)						
P (MW)		R-SOS	SOS	R-SOS	IDE [17]	SOS	CS [14]	PSO [12]
0	309.000	308.99961	308.99962	0.00038	0.0004	0.00038	0.000	0.191
18	592.185	592.184752	592.18475	0.00024	0.0002	0.00025	0.056	0.007
36	800.980	800.980354	800.980358	0.00035	0.0004	0.000358	0.048	0.030
54	901.361	901.36123	901.36125	0.000232	0.0002	0.00025	0.004	0.015
72	918.567	918.567290	918.56732	0.00029	0.0003	0.00032	0.064	0.026
90	1161.719	1161.71950	1161.71947	0.00050	0.0005	0.00047	0.015	0.022
108	1387.229	1387.228885	1387.22887	0.000114	0.0001	0.00013	0.004	0.033
126	1505.826	1505.825998	1505.82603	0	0	0.00003	0.022	0.019
144	1533.617	1533.616841	1533.61690	0.000158	0.0002	0.0001	0.058	0.011
162	1735.414	1735.414039	1735.41397	0.00003	0	0.00003	0.023	0.016
180	1976.564	1976.563815	1976.56379	0.000184	0.0002	0.00021	0.010	0.021
198	2113.785	2113.785473	2113.78551	0.000473	0.0005	0.00051	0.020	0.007
216	2153.829	2153.828654	2153.82874	0.000345	0.0003	0.00026	0.043	0.018
234	2313.540	2313.540000	2313.53990	0	0	0.0001	0.016	0.000
252	2569.061	2569.060796	2569.06075	0.0002	0.0002	0.00025	0.003	0.003
270	2725.078	2725.0781128	2725.07815	0.000112	0.0001	0.00015	0.008	0.010
288	2778.892	2778.8919995	2778.89211	0	0	0.00011	0.024	0.032
306	2896.388	2896.387999	2896.38786	0	0	0.00014	0.001	0.019
324	3164.832	3164.83200	3164.83192	0	0	0.00008	0.011	0.017
342	3339.577	3339.576526	3339.57656	0.000473	0.0005	0.00044	0.008	0.028
360	3408.509	3408.509333	3408.50947	0.000333	0.0003	0.00047	0.004	0.049
$\lvert\sum \text{error}\rvert$				0.004448	0.0044	0.005038	0.444	0.575

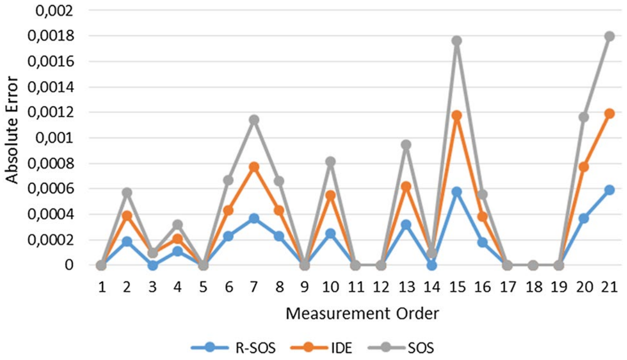

Fig. 6 Non-smooth model error values for Unit 1

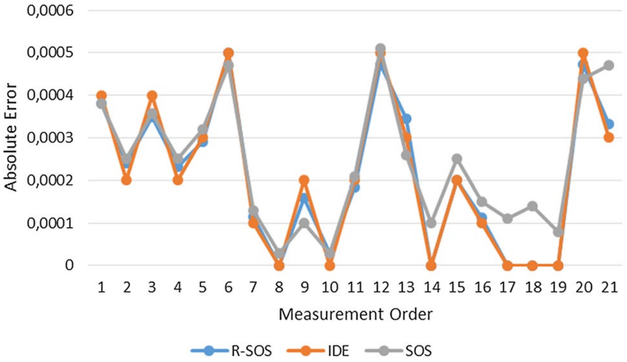

Fig. 7 Non-smooth model error values for Unit 2

estimated fuel cost values, and error values obtained from the R-SOS, SOS, IDE, DE, CS and PSO for two different plant have been given in Tables 9 and 10.

As can be seen from Table 9, when results are compared obtained from proposed R-SOS, and other algorithms, it can be seen that the R-SOS algorithm produces better results than others and it reduces the total error value for unit 1 as 0.000019 comparing with IDE, as 0.000039 GJ/H comparing with classical SOS, 0.56831 GJ/H comparing with CS, and as 0.615319 GJ/H comparing with the PSO. Moreover, Error values are compared as graphically given in Fig. 6.

When the Table 10 is investigated, it is seen that the R-SOS algorithm produces almost the same results with the IDE algorithm and reaches the same total error value for unit 2. Moreover, the R-SOS algorithm can reduce the total error for unit 2 as 0.000638 GJ/H comparing with classical SOS, as 0.4395 GJ/H comparing with CS, and as 0.57052 GJ/H comparing with the PSO. Moreover, Error values are compared as graphically given in Fig. 7.

As can be clearly seen that the R-SOS algorithm produces values close to real for thermal unit 2. It is totally clear that proposed R-SOS algorithm produces almost same results with IDE algorithm and provides better results when comparing other algorithms for this test case. This case also shows that the R-SOS algorithm can converge in a powerful way especially for non-smooth cost functions.

Moreover, the proposed R-SOS algorithm runs for 100 times in order to evaluate the robustness of it as same in previous test case. Minimum error, maximum error, mean error and standard deviation values for non-smooth test function are given in Table 11. It can be clearly seen from Table 11 that the proposed R-SOS algorithm have less minimum, maximum, mean error and standard deviation values by producing almost same results for all runs. Thus, it produces more efficient results than classical SOS.

6 Conclusion

In this study, the improved R-SOS Algorithm has been proposed for estimation of parameters of fuel cost function, which are used for solving optimal power flow and economic dispatch problems. The smooth fuel cost function forms such as linear, quadratic cubic and non-smooth have been considered. In the experiments three different plants have been used and each plant consists of five generating units for smooth cost function. Moreover, two different unit has been considered to test the non-smooth fuel cost function type. Obtained results show that the proposed R-SOS algorithm produces better results and decrease the error between estimated and actual fuel cost values for all test cases and for all plants with different fuel types. Especially for non-smooth fuel type, error value obtained from the proposed algorithm is very close to zero. This result shows that the improved R-SOS algorithm can show a good convergence by ensuring a better exploitation process in solving optimization problems having complex, non-linear and non-smooth cost functions.

Table 11 Statistical results obtained for test case 4 (Non-smooth Model)

Algorithm	Unit	Minimum error	Maximum error	Mean error	Standard deviation
R-SOS	1	0.0036812	0.0036820	0.0036815	1.8127×10^{-6}
SOS		0.00372	0.00394	0.00381	1.266×10^{-4}
R-SOS	2	0.004480	0.004485	0.004482	1.5741×10^{-6}
SOS		0.005038	0.005051	0.005042	1.987×10^{-4}

References

1. Abou El Ela AA, Abido MA, Spea SR (2010) Optimal power flow using differential evolution algorithm. Electr Power Syst Res 80(7):878–885
2. Sayah S, Zehar K (2008) Modifed differential evolution algorithm for optimal power ow with non-smooth cost functions. Energy Convers Manage 49(11):3036–3042
3. Pao-La-Or P, Oonsivilai A, Kulworawanichpong T (2010) Combined economic and emission dispatch using particle swarm optimization. WSEAS Trans Environ Dev 6:296–305
4. Basu M (2008) Dynamic economic emission dispatch using nondominated sorting genetic algorithm-II. Electr Power Energy Syst 30(2):140–149
5. Duman S, Güvenc U, Sönmez Y, Yörükeren N (2012) Optimal power ow using gravitational search algorithm. Energy Convers Manage 59:86–95
6. El-Naggar KM, AlRashidi MR, Al-Othman AK (2009) Estimating the input-output parameters of thermal power plants using PSO. Energy Conv Manag 50(7):1767–1772
7. Taylor FJ, Huang CH (1977) Recursive estimation of incremental cost curves. Comput Electr Eng 4(4):297–307
8. El-Hawary ME, Mansour SY (1982) Performance evaluation of parameter estimation algorithms for economic operation of power systems. IEEE Trans Power Appar Syst 3:574–582
9. Soliman SA, Al-Kandari AM (1996) Kalman filtering algorithm for on-line parameter identification of input–output curves for thermal units. In: Proc. 8th Mediterranean Electrotechnical Conference, pp 1588–1593
10. Ferreira IM, Maciel Barbosa FP (1996) A square root filter algorithm for dynamic state estimation of electric power systems. Proc 7th Mediterr Electrotech Conf 3:877–880
11. Shivakumar NR, Jain A (2008) A review of power system dynamic state estimation techniques. In: Proc. Power System Technology and IEEE Power India Conference, pp 1–6
12. Alrashidi MR, El-Naggar KM, Al-Othman AK (2009) Particle Swarm Optimization based approach for estimating the fuel-cost function parameters of thermal power plants with valve loading effects. Electr Power Compon Syst 37(11):1219–1230
13. Sonmez Y (2013) Estimation of fuel cost curve parameters for thermal power plants using the ABC algorithm. Turk J Electr Eng Comput Sci 21(1):1827–1841
14. AlRashidi MR, El-Naggar KM, AlHajri MF (2015) Convex and non-convex heat curve parameters estimation using cuckoo search. Arab J Sci Eng 40(3):873–882
15. Sonmez Y, Güvenç , Yılmaz C, Kahraman HT (2018) Fuel cost function parameter optimization by using gravitational search algorithm. In: Proc. 7th International Conference on Advanced Technologies, pp 226–231
16. Sayah S, Hamouda A (2015) Novel application of differential evolution algorithm for estimating fuel cost function of thermal generating units. In: Proc. Third World Conference on Complex Systems (WCCS), IEEE
17. Sayah S, Hamouda A (2018) Efficient method for estimation of smooth and nonsmooth fuel cost curves for thermal power plants. Int Trans Electr Energy Syst 28(3):e2498
18. Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112

Yusuf Sönmez received the B.S., M.S., and Ph.D. degrees in electrical education from Gazi University (GU), Ankara, Turkey, in 2002, 2005, and 2008, respectively. Currently, he is an Associate Professor with GU. His research interests include power systems, artificial intelligence, and metaheuristic search algorithms.

Mesut Unal received the B.S. degree in computer engineering from East Mediterranean University, Famagusta, North Cyprus, in 2008, and M.S. degree in Gazi University in 2019. Currently, he is working as a computer engineer in Ministry of National Defense of Turkey. His research interests include power systems and meta-heuristic search algorithms.