Artificial Bee Colony Framework to Non-convex Economic Dispatch Problem with Valve Point Effects: A Case Study

Doğan Aydın
Dumlupinar University
Computer Engineering Dept. 43100
Kutahya, Turkey
dogan.aydin@dpu.edu.tr

Gürcan Yavuz
Dumlupinar University
Computer Engineering Dept. 43100
Kutahya, Turkey
gurcan.yavuz@dpu.edu.tr

Serdar Özyön
Dumlupinar University
Electrical and Electronics Engineering
Department 43100
Kutahya, Turkey
serdar.ozyon@dpu.edu.tr

Celal Yaşar
Dumlupinar University
Electrical and Electronics Engineering
Department 43100
Kutahya, Turkey
celal.yasar@dpu.edu.tr

Thomas Stützle
IRIDIA, CoDE, Universite Libre de
Bruxelles
Brussels, Belgium
stuetzle@ulb.ac.be

ABSTRACT

ABC-X is a generalized, automatically configurable framework for the Artificial Bee Colony (ABC) metaheuristic. ABC-X has recently been proposed and it has initially been tested on different benchmark functions sets, showing very good results when compared to known ABC algorithms. However, it has never been used in an industrial application. In this case study on a real-world problem, the performance of ABC-X for the economic power dispatch problem (EPDP) with valve point effects is investigated. The algorithm's performance is tested on four instances of different size and compared with the performance of other algorithms that have been proposed earlier for this problem. The results obtained show that ABC-X can successfully solve the EPDP with valve point effects.

CCS CONCEPTS

•**Theory of computation → Evolutionary algorithms; Bio-inspired optimization;** *Design and analysis of algorithms;*

KEYWORDS

Artificial Bee Colony, Enery Dispatch, Automatic Design of Algorithm

ACM Reference format:

Doğan Aydın, Gürcan Yavuz, Serdar Özyön, Celal Yaşar, and Thomas Stützle. 2017. Artificial Bee Colony Framework to Non-convex Economic Dispatch Problem with Valve Point Effects: A Case Study. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082486

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3082486

1 INTRODUCTION

The ABC meta-heuristic has been used for more than ten years in solving continuous optimization problems and a large number of variants of the basic ABC have been proposed [14]. Usually, the ABC variants suggest a fixed set of changes in one or several steps of the classical ABC algorithm and explore the extent to which these changes improve performance. Different from other approaches, ABC-X has been designed as a framework that encompasses many different algorithmic components that have been proposed in the context of ABC algorithm variants [3]. It also features a generalized search equation that can be used to instantiate many of the search equations that have already been applied in various ABC algorithms but it can generate also many additional ones. The main idea underlying ABC-X is to allow, in a modular fashion, the generation of not only many existing ABC variants but also many new ABC algorithms that have never been proposed in the literature before. The space of possible instantiations of ABC algorithms and their associated, typically numerical parameters can then be searched using automated algorithm configuration tools such as irace [17], ParamILS [12], or SMAC [11].

In our previous studies [2, 3], the use of ABC-X with automatic algorithm configuration techniques has been tested using classical benchmark functions that are used to test the performance of continuous optimizers. So far, the results obtained have been positive as they showed that we could generate from ABC-X automatically, using irace [17], new ABC algorithms that are superior to manually generated ABC algorithm designs, even if the latter are fine-tuned using a same configuration budget to set numerical ABC algorithm parameters [3].

In this paper, the performance of the ABC-X algorithm is tested on an industrial application. For this task, the problem of economic power dispatch problem (EPDP) with valve point effects has been chosen. This problem is well known in the field of electrical energy generation and its solution has been studied for many years. According to Happ [10], it is necessary to go back to the beginning of the 1920ies to examine the work of engineers on the problem of

Algorithm 1 The ABC-X Framework

ApplyInitializationStrategy( ) ▷ Comp. 1
while termination condition is not met **do**
  for each employed bee, i, in swarm **do** ▷ Comp. 2
    ApplyEmployedBeeSearchEquationStrategy(i)
  CalculateSelectionProbabilies( ) ▷ Comp. 3
  for each onlooker bee, j, in swarm **do** ▷ Comp. 4
    ApplyOnlookerBeeSearchEquationStrategy(j)
  ApplyScoutBeeStrategy( ) ▷ Comp 5
  ApplyLocalSearch( ) ▷ Comp. 6
  UpdatePopulationSize( ) ▷ Comp. 7

Algorithm 2 The general form of the search equation

% D is the dimension of problem space
for $t = 1$ to m **do**
  select random dimension j ($1 \leq j \leq D$)
  $x_{i,j} = term_1 + term_2 + term_3 + term_4$

economically dividing energy generation and the burden between existing generators. The aim of the EPDP is to minimize the fuel cost of generating units for a given operating time, typically one hour, to achieve optimum production delivery between operating units, and to satisfy the system load demand and the constraints of the generating units. Moreover, the methods used to solve the problem can be easily adapted to the real-world problem instances. As a matter of fact, EPDP was chosen as one of the problems in the CEC 2011 competition for testing evolutionary algorithms applied to real world optimization problems [7].

Using the automatic algorithm configuration tool irace [17], an ABC variant is generated from ABC-X in accordance with the EPDP instances to be tackled, and its performance is tested. In addition, the results obtained with ABC-X are compared with many algorithms that have been proposed before in the literature for the same problem.

The paper is structured as follows. In Section 2, we present the ABC-X framework. In Section 3, we define the economic dispatch problem with valve point effects and in Section 4 we detail how ABC-X was applied to tackle it. Experimental results on four real-world problem instances and a comparison to the literature are given in Section 5. Finally, we conclude the article in 6.

2 ABC-X: A GENERALIZED AND FULLY CONFIGURABLE FRAMEWORK

ABC-X was created by combining different ABC components in the literature into one algorithm framework, which is outlined in Algorithm 1. ABC-X consists of seven components and the different alternatives for each component are described below.

2.1 First Component: Initialization

For the initialization of the population, there are four alternative strategies called *default*, *opposition-based*, *chaotic* and *mix*:

- ***default:*** This is the standard initialization strategy used in the canonical ABC algorithm. The details of this strategy are stated in [2].
- ***opposition-based:*** In this strategy, SN (SN is the population size) candidate solutions are selected randomly in the search space and their opposite positions are determined. Then, the population is initialized by deleting the worst SN solutions from the $2 * SN$ candidate solutions [8].

Table 1: Alternative options for each component in the generalized search equation. $x_{G,j}$ and $x_{GD,j}$ are the best-so-far and the best-distance candidate solutions at dimension j, respectively. X_{r1} and X_{r2} are two randomly selected candidate solutions. ϕ_l can take two possible ranges: $[-1, -1]$ and $[-SF, SF]$ where SF is a randomly selected positive real value. These ranges are calculated randomly for each term of the general search equation.

m	$term1$	$term2 \mid\mid terms3 \mid\mid terms4$
1	$x_{i,j}$	$\phi_l(x_{i,j} - x_{G,j})$
k ($1 \leq k \leq D$)	$x_{G,j}$	$\phi_l(x_{i,j} - x_{r1,j})$
$[t, k]$ ($1 \leq t < k \leq D$)	$x_{r1,j}$	$\phi_l(x_{G,j} - x_{r1,j})$
		$\phi_l(x_{r1,j} - x_{r2,j})$
		$\phi_l(x_{i,j} - x_{GD,j})$
		$DoNotUse$

- ***chaotic:*** In the *default* strategy, food sources are randomly positioned according to a uniform distribution in search space. Here, instead, chaotic maps are used as random number generators. Within ABC-X, random numbers can be generated from seven different chaotic maps.
- ***mix:*** In this method, *chaotic* and *opposition-based* are used together [9].

2.2 Second and Forth Component: Employed Bees and Onlooker Bees Steps

In the employed and onlooker bees steps, each bee chooses a candidate solution and generates a new, possibly better candidate solution in the vicinity of the visited one. While employed bees select a candidate solution to visit deterministically, onlookers determine the one to visit by using a probabilistic selection rule based on the candidate solutions' qualities. The new candidate solution is determined by a search equations using the visited candidate solution as a reference solution. The visited candidate solution is replaced by the newly generated candidate solution if the quality of the new one is better than the visited one. Several search equations have been proposed in the literature. Instead of adding all search equations one by one in the ABC-X algorithm, a general search equation, which can produce several search equations including previously explored ones, is proposed. This search equation is given in Algorithm 2 and possible alternatives for each term are defined in Table 1. The details of this generalized search equation can be also found in [2, 3].

2.3 Third Component: Calculation of Selection Probabilities

The onlooker bees visit candidate solutions according to a probabilistic selection rule. In the ABC literature, different approaches have been proposed for the probabilistic selection rules and ABC-X considers the following ones.

- ***default***: A candidate solution is chosen according to a probabilistic selection rule. In the ABC literature, the different approaches proposed for probabilistic selection rules are used in the ABC-X algorithm as follows:

$$p_i = \frac{fitness_{X_i}}{\sum_{n=1}^{SN} fitness_{X_n}}. \tag{1}$$

  In Equation 1, $fitness_{X_i}$ is the *fitness* value of solution X_i. It is defined as

$$fitness_{X_i} = \begin{cases} \frac{1}{1+f_{X_i}}, \ f_{X_i} \geqslant 0, \\ 1 + abs(f_{X_i}), \ f_{X_i} < 0, \end{cases} \tag{2}$$

  where f_{X_i} is the objective function value of the solution X_i.
- ***weightedSum***: In the weighted sum method, the weight of selecting the worst solution is set to at least $1 - w$ and the weight of other solutions may vary from $1 - w$ to 1 using

$$p_i = (1 - w) + (w - (w\frac{f(X_G)}{f(X_i)})) \tag{3}$$

  where $f(X_G)$ is the objective value of the best-so-far solution and w is a constant, which as default is set to 0.9.
- ***rankedBased***: In the Rosenbrock ABC algorithm [13], a ranked based fitness adaptation rule is used when calculating fitness value of candidate solution X_i, $fitness_{X_i}$, instead of using Equation 2 as follows:

$$fitness_{X_i} = 2 - SP + \frac{2(SP - 1)(r_i - 1)}{SN - 1} \tag{4}$$

  where SP is defined as selection pressure set to between 1.0 and 2.0, and r_i is the rank of the candidate solution X_i.

2.4 Fifth Component: Scout Bees Step

When a candidate solution is visited *limit* times, it is assumed that a better candidate solution close to it is difficult or not possible to find. In that case, the candidate solution is abandoned and a new candidate solution to be explored is generated in the search space. (This mechanism is called scout bees step in analogy to the original inspiration of the ABC algorithm.) The original ABC algorithm generates a new candidate solution in the scout bees step in the same way as in the initialization step. This is called *default* in ABC-X. However, there are other strategies proposed for the scout bees step, trying to improve ABC's search capability. In ABC-X, we used two chaotic methods namely *chaotic1* and *chaotic2* and two other methods called *scoutBABC* and *scoutIABC*. The details of these methods can be found in [3].

2.5 Sixth Component: Applying a Local Search Procedure

There are multiple local search procedures within the ABC-X framework. For now, the applicable local search procedures are Powell, Mtsls, IPOP-CMAES, and competitive in which an appropriate procedure may be found by adding a competition phase [2]. The local search procedure is applied only to the best-so-far solution in each iteration if the best-so-far solution is updated.

2.6 Seventh Component: Updating Population Size

In the original ABC algorithm, the population size is kept constant throughout the execution. However, the number of candidate solutions in the population can change dynamically in some ABC variants, and these approaches have been implemented in ABC-X. In the ABC-X framework, the fixed population size approach is called *default*, the approach with the incremental population size is called *incremental*, and the approach that alternatingly reduces and increases the population during the search process is called *dynamic* [3].

3 NON-CONVEX ECONOMIC DISPATCH PROBLEM WITH VALVE POINT EFFECTS

With today's increasing need for electricity, the issue of economic power dispatch has become one of the most important issues in the operation of power systems.The fact that the cost of fuel used in electricity production reaches a significant fraction of the production costs, requires the fuel producing companies to use the fuels more efficiently. In the literature, the problem of economic power dispatch is defined as the adjustment of the active power output of production units to meet the current load in the system (under system constraints and minimum fuel cost). In such problems, the fuel cost function curves of the production units are used to calculate the total fuel cost [1].

3.1 The objective function

The solution of the economic power dispatch problem is found by the minimization of total fuel cost under the limits of the system. The objective function of the standard EPDP is defined as [6, 30]:

$$\min F_{total} = \min \sum_{n=1}^{N} F_n(P_{G,n}) \tag{5}$$

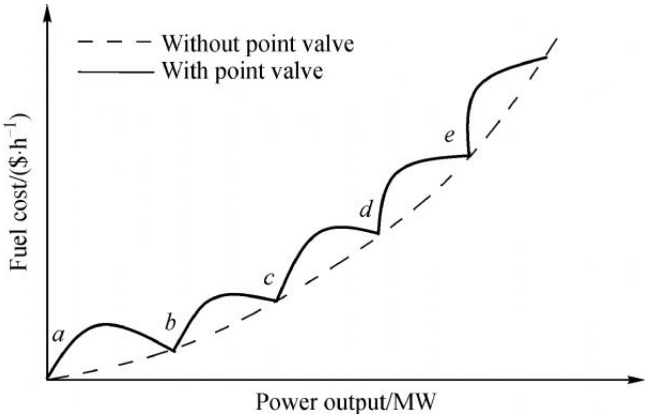
where $F_n(P_{G,n})$ represents the fuel cost function of the n-th generation unit as R/h which is defined as:

$$F_n(P_{G,n}) = a_n + b_n P_{G,n} + c_n P_{G,n}^2, \ (R/h) \tag{6}$$

where $P_{G,n}$ represents the output power of the n-th generation unit in MW, and a_n, b_n, c_n are constants.

The fuel cost function of the generation units is shown in a schematic way in Figure 1. In Figure 1, the convex function shown by the broken line is the fuel cost function when we used Equation 6, which contains no valve point effects. In fact, the input-output curve of multi-valve steam stand generation units is very different when it is compared with the equality in Equation 6. The inclusion of the valve point effect in the fuel cost of the generation unit

Figure 1: The input-output characteristic of valve point effect generation unit



makes the representation of the fuel cost more realistic. As the valve point is finalized with surges, the fuel cost function includes higher non-linear series. For this reason, as for the study aimed at considering the valve point effects, a non-convex function is used in the Equation 7 that is varied following the continuous line in Figure 1.

$$F_n(P_{G,n}) = a_n + b_n P_{G,n} + c_n P_{G,n}^2 + |d_n.sin(e_n(P_{G,n}^{min} - P_{G,n}))|, (R/h). \quad (7)$$

In Equation 7, d_n and e_n represent the valve point effects.

3.2 Problem constraints

The EPDP with valve point effects has two constraints that should be satisfied when tackling the problem, the generating capacity constraints and the power balance constraints.

The generating capacity constraints. The power generated from the nth unit, $P_{G,n}$, should not exceed the minimum and the maximum bounds on the power generated by the nth unit depicted by $P_{G,n}^{min}$ and $P_{G,n}^{min}$ respectively. This constraint is defined as

$$P_{G,n}^{min} \leq P_{G,n} \leq P_{G,n}^{max}, \ (n \in N). \quad (8)$$

The power balance constraint. While planning for an optimal operation of energy production systems, AC power flow analysis is needed to be done first. In the AC power flow analysis of the energy system, the net active and reactive powers of all generators should be determined and amplitudes and phase angles of all generators are found. With this information, the system losses are calculated by the active and reactive power flows occurring in the transmission lines in the system. The problem of optimal power flow is the problem of finding the optimal value of the energy fuel cost, which is the objective function of the system, under the condition of satisfying all equality and inequality constraints. When the solution of this optimization problem is investigated, the power load, P_{load}, is taken as constant. Actually, in the literature it is typically assumed that the power load is the energy demand of a place and that it remains constant during 2 to 10 minutes. For this reason, the optimization of fuel cost must be done very quickly. However, the AC power flow analysis is an iterative metric and it needs to be re-calculated when the generated power values are changed, which is time and memory consuming. Therefore, if an algorithm tackles the problem using an AC power flow analysis, this analysis should be calculated at every objective function evaluation, which is very time consuming. Therefore, when testing a new algorithm for economic dispatch problems, the transmission line losses are ignored or calculated with a single equation using approximate B-loss matrices. B-loss matrices allow the transmission line losses to be handled faster than using an AC load flow for generated powers. When the system losses are ignored, the power balance calculated as follows:

$$P_{load} - \sum_{n=1}^{D} F_n(P_{G,n}) = \Delta P = 0 \quad (9)$$

where D is the total number of generators, which is also the problem dimension. If we consider the power losses, the power balance constraint is taken as in Equation 10:

$$P_{load} - P_{loss} - \sum_{n=1}^{D} F_n(P_{G,n}) = \Delta P = 0 \quad (10)$$

where P_{loss} are the power losses, which are calculated using the B-loss matrix by the following Equation [30]:

$$P_{loss} = \sum_{n=1}^{D} \sum_{j=1}^{D} P_{G,n}.B_{nj}.P_{G,j} + \sum_{n=1}^{D} B_{0n}.P_{G,n} + B_{00} \quad (11)$$

In the literature, there are problem instances of both lossy and lossless power dispatch problems. In a majority of the lossy systems, the losses are calculated using the B-loss matrices. In our study, the algorithm is tested on examples of lossy and lossless problems. B-loss matrices are preferred to use in lossy systems to compare with other algorithms in the literature.

4 APPLYING ABC-X TO EPDP

4.1 Achievement of appropriate ABC algorithm from ABC-X framework using irace

ABC-X is a generalized framework, and its parameter values must be well-defined in order to produce an algorithm suitable for the problem. In this work, parameter values of ABC-X and so an appropriate ABC algorithm from ABC-X, are determined with irace, an automatic algorithm configuration tool. irace is run with default parameter values. Problem instances are selected as training instances for irace from systems with 3, 5, 6, 13 and 40 generator units. Among these, the instances with 3, 5, and 6 generators ones are lossy and the other ones are lossless systems. In order to make the training instances different from the test cases that are available in the literature during the parameter configuration phase, problem instances were produced that differ in the power load demands from each of the problem instances from the literature. In this way, totally 30 instances (6 instances for each system) have been used as training set for irace to find appropriate parameter settings for ABC-X.

The algorithm obtained by irace from the ABC-X framework is given in Algorithm 3. This ABC algorithm does not correspond to any known ABC algorithm, as can be observed from the fact that it uses different search equations in the employed and onlooker bees steps, the specific combination of the *scoutBABC* component with others for the scout bees step, and the particular combination of the local search algorithms.

Algorithm 3 The constructed ABC Algorithm

$SN = 22$, $limit = 2.7 * SN$
Apply *mix* initialization strategy with Henon chaotic map
for each employed bee X_i **do**
  for $t = 1$ to $m = 5$ **do**
    Select dimension j randomly
    $v_{i,j} = x_{r1,j} + \phi_1(x_{i,j} - x_{r1,j}) + \phi_2(x_{i,j} - x_{G,j})$
  if V_i is better than X_i **then**
    $X_i = V_i$, $trial_i = 0$
  else
    $trial_i = trial_i + 1$
Calculate selection probabilities with *rankBased* strategy with $SP = 1.95$
for each onlooker bee X_i **do** ▷ Select X_i according to the selection probability
  for $t = 3$ to $m = 4$ **do**
    Select dimension j randomly
    $v_{i,j} = x_{G,j} + \phi_1(x_{i,j} - x_{G,j}) + +\phi_1(x_{r1,j} - x_{r2,j}) + +\phi_2(x_{i,j} - x_{GD,j})$
  if V_i is better than x_i **then**
    $X_i = V_i$, $trial_i = 0$
  else
    $trial_i = trial_i + 1$
Apply *scoutBABC* scout bees strategy with parameter values: $wMin = 0.24$ and $wMax = 0.72$
Apply *competitive* local search with using Mtsls1 and IPOP-CMAES
fixed population size

4.2 Structure of a candidate solution

For an EPDP problem with D generators, a candidate solution X is presented as a vector

$$X = \left[P_{G,1}, \ldots, P_{G,n}, \ldots, P_{G,D}\right] \tag{12}$$

where each element $P_{G,i}$, which corresponds to the power generated by generator unit i, should satisfy the problem constraints.

4.3 Handling problem constraints

When solving the EPDP with the ABC-X framework, constraint handling mechanisms need to be defined. For the generating capacity constraints, there are two possible ways of handling them. One is to use penalize constraint violations in the fitness function [19]; another is to repair unfeasible solutions, creating feasible ones, and to work only with the repaired solutions during optimization process[24]. Here, we consider the second approach. The repair process we apply here to avoid the violation of the equality constraints defined in Eqs. 9 and 10, is given in Algorithm 4.

Algorithm 4 Power balance constraint handling procedure

while $|\Delta P| \leq \varepsilon$ **do** ▷ where ε is very small tolerance value
  Randomly select a generator i from the solution
  if $\Delta P < 0$ **then**
    Add an amount of ΔP to $P_{G,i}$ that doesn't violate $P_{G,n}^{max}$ (such as $P_{G,i} = min(P_{G,i} + \Delta P, P_{G,n}^{max})$)
  else if $\Delta P > 0$ **then**
    Subtract an amount of Δ_P from $P_{G,i}$ that doesn't violate $P_{G,n}^{min}$ (such as $P_{G,i} = min(P_{G,i} - \Delta P, P_{G,n}^{min})$)

Table 2: (Case I) The cost function coefficients of the generation units and active power generation limits

Bus no.	1	2	5	8	11	13
a	150	25	0	0	0	0
b	2	2.5	1	3.25	3	3
c	0.0016	0.01	0.0625	0.00834	0.025	0.025
d	50	40	0	0	0	0
e	0.063	0.098	0	0	0	0
$P_{min}(MW)$	50	20	15	10	10	12
$P_{max}(MW)$	200	80	50	35	30	40

5 EXPERIMENTAL RESULTS

In order to asses the effectiveness and robustness of the ABC-X framework, two test cases of economic dispatch with valve-point effect have been considered. The first system has six generators with transmission losses (Case I) and the other one is from the realistic Taipower system, which has 40 generator units without transmission losses (Case II).

ABC-X framework is implemented in C++. The code was run on a Intel Xeon E5410 quadcore CPUs running at 2.33 GHz with 2 x 6 MB L2 cache and 8 GB RAM. In each case study, the results are obtained over 30 independent runs which were terminated after $D * 1000$ and $D * 10000$ (D is the number of generators in our case) function evaluations (FEs). The reason for running the program with these two values of FEs is that (i) no standard has been specified in the literature and many methods have worked with different FEs in the past and (ii) to test the convergence speed of the algorithm. For the computation of the objective function value, ε is set to 10^{-6} to obtain feasible solutions.

5.1 Case I: 6 Generating Units System

The first case is a lossy system consisting of six generating units and the expected power demand, P_{load}, is 283.4 MW. The cost function coefficients and the B matrices related to the problem however is given in Tables 2 and 3 respectively.

The best and worst total fuel costs obtained for the first case over $D * 1000$ and $D * 10000$ FEs are given in Table 4. As seen in Table 4, both the best and the worst results of ABC-X are similar in both runs with low and high FEs. This indicates a low variability of the results and to a fast convergence of ABC-X at each run.

In Table 5, we compare the results achieved by ABC-X with various algorithms that have been applied to the same problem instance in the literature before. The algorithm generated from

Table 3: (Case I) B loss matrix values

$$[B] = \begin{bmatrix} 0.0224 & 0.0103 & 0.0016 & -0.0053 & 0.0009 & -0.0013 \\ 0.0103 & 0.0158 & 0.001 & -0.0074 & 0.0007 & 0.0024 \\ 0.0016 & 0.001 & 0.0474 & -0.0687 & -0.0060 & -0.0350 \\ -0.0053 & -0.0074 & -0.0687 & 0.3464 & 0.0105 & 0.0534 \\ 0.0009 & 0.0007 & -0.0060 & 0.0105 & 0.0119 & 0.0007 \\ -0.0013 & 0.0024 & -0.0350 & 0.0534 & 0.0007 & 0.2353 \end{bmatrix}$$

$$[B_0] = \begin{bmatrix} -0.0005 & 0.0016 & -0.0029 & 0.006 & 0.0014 & 0.0015 \end{bmatrix}$$

$$[B_{00}] = \begin{bmatrix} 0.0011 \end{bmatrix}$$

Table 4: (Case I) The results obtained over 30 trials

	Results over $D * 1000$ FEs		Results over $D * 10000$ FEs	
Generator	Output power	Output power	Output power	Output power
$P_1(MW)$	199.6	199.6	199.6	199.6
$P_2(MW)$	20.00	20.01	20.00	20.00
$P_3(MW)$	24.01	25.24	23.96	23.96
$P_4(MW)$	18.68	17.51	18.86	19.54
$P_5(MW)$	18.53	17.28	18.17	18.17
$P_6(MW)$	13.54	14.69	13.82	13.47
$P_{loss}(MW)$	11.22	9.237	11.18	10.77
$F_{total}(R/h)$	925.0 (Best Cost)	925.2 (Worst Cost)	925.0 (Best Cost)	925.0 (Worst Cost)

Table 5: (Case I) Comparison between the results of ABC algorithm produced by ABC-X and other algorithms in literature

Methods	Best Cost (R/h)	Worst Cost (R/h)	FEs
MSG-HS [30]	925.6	928.6	1000000
GA [18]	996.0	111.7	6120
DE [22]	963.0	–	10000
ABC [21]	928.4	–	10000
EP [20]	955.5	959.4	–
IEP [20]	953.6	958.3	–
SADE ALM [28]	944.0	964.8	20000
TS-SA [20]	959.6	966.0	–
ITS [20]	969.1	985.5	–
ABC-X	**925.0**	**925.2**	6000
ABC-X	**925.0**	**925.0**	60000

ABC-X outperforms all other algorithms in terms of achieving the best and worst costs even though most of the considered algorithms have run with more function evaluations.

5.2 Case I: 40 Generating Units System

This case study includes 40 generators with quadratic cost functions, together with the valve point effects. In this case, the transmission losses are ignored and the required power demand to be met by all the forty generating units is 10500 MW. The unit data (cost function coefficients) for the system are available in literature and so we have taken them directly from [27]. The Taipower system is one of the largest problems that can be found in the literature on the EPDP with valve point effects.The best and worst total fuel costs obtained for the two case studies with $D * 1000$ and $D * 10000$ FEs are listed in Table 6 and 7.

Since this is a larger system with more non-linear elements, it has also more local minima and, thus, is also more difficult to solve. This situation is reflected by the results in Tables 6 and 7, where, unlike in the first case study, the results obtained with $D * 1000$ FEs are clearly worse than those obtained with $D * 10000$ FEs and also the variability of the results obtained is higher.

In Table 8, a comparison of the results of ABC-X with those of several algorithms from the literature is given. When the comparison results in Table 8 are examined, it can be observed that although each algorithm is run with different FEs, our approach gives very good results when compared to the other algorithms and the best costs reported for ABC-X with $1000 * D$ function evaluations are better than those of all other algorithms except for NPSO-LRS [25], ST-HDE [29], and DEC(2)-SQP(1) [6]. On the other hand, the worst results by ABC-X are in all cases better than the worst results achieved by any of the other algorithms and in many cases also better than their best ones. The results by ABC-X with $10000 * D$ function evaluations indicate that it can still benefit significantly from the additional computation time. Hence, overall the results achieved with the proposed approach of automatically configuring a specialized, ABC-X algorithm (or any other continuous optimizer from a flexible, component-wise framework) for a specific, industrial-style problem shows to be highly effective without requiring any significant algorithm engineering effort.

6 CONCLUSION

ABC-X is a flexible and configurable ABC framework that can be adjusted by automatic algorithm configuration tools to specific problems. ABC-X has previously been applied to several benchmark functions and has demonstrated superior performance over known ABC algorithms. In this study, ABC-X was tested on a non-convex economic dispatch problem with valve point effects that serves as a case study for a potential industrial application in power management. The obtained results on two case studies shows that the algorithm generated from the ABC-X framework reaches very high performance and that its structure did not exist previously in the literature. Future planned work will be to extend the ABC-X algorithm by adding new components and realize its usability in different application areas. In addition, we want to implement the ideas underlying the design of the ABC-X framework to other metaheuristics such as Differential Evolution (DE) and Particle Swarm Optimization (PSO).

REFERENCES

[1] 2013. Solution to non-convex economic dispatch problem with valve point effects by incremental artificial bee colony with local search. *Applied Soft Computing* 13, 5 (2013), 2456 – 2466.

[2] Dogan Aydın and Thomas Sffltzle. 2015. A configurable generalized artificial bee colony algorithm with local search strategies. In *Evolutionary Computation (CEC), 2015 IEEE Congress on.* IEEE, 1067–1074.

[3] Doğan Aydın, Gürcan Yavuz, and Thomas Stützle. 2017. ABC-X: a generalized, automatically configurable artificial bee colony framework. *Swarm Intelligence* 11, 1 (2017), 1–38.

[4] C. H. Chen and S. N. Yeh. 2006. Particle Swarm Optimization for Economic Power Dispatch with Valve-Point Effects. In *2006 IEEE/PES Transmission Distribution Conference and Exposition: Latin America.* 1–5. DOI: http://dx.doi.org/10.1109/TDCLA.2006.311397

[5] C-L Chiang. 2007. Genetic-based algorithm for power economic load dispatch. *IET generation, transmission & distribution* 1, 2 (2007), 261–269.

[6] Ld S Coelho and Viviana Cocco Mariani. 2006. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with

Table 6: (Case II) The best and worst results obtained over 30 trials with $D * 1000$ FEs

Best solution				Worst solution			
Generator	Output power	Generator	Output power	Generator	Output power	Generator	Output power
$P_1(MW)$	114.0	$P_{21}(MW)$	523.0	$P_1(MW)$	114.0	$P_{21}(MW)$	523.5
$P_2(MW)$	114.0	$P_{22}(MW)$	524.1	$P_2(MW)$	114.0	$P_{22}(MW)$	550.0
$P_3(MW)$	102.3	$P_{23}(MW)$	523.1	$P_3(MW)$	120.0	$P_{23}(MW)$	523.4
$P_4(MW)$	190.0	$P_{24}(MW)$	523.1	$P_4(MW)$	190.0	$P_{24}(MW)$	550.0
$P_5(MW)$	97.00	$P_{25}(MW)$	523.0	$P_5(MW)$	97.00	$P_{25}(MW)$	526.3
$P_6(MW)$	140.0	$P_{26}(MW)$	525.6	$P_6(MW)$	140.0	$P_{26}(MW)$	550.0
$P_7(MW)$	269.9	$P_{27}(MW)$	10.00	$P_7(MW)$	300.0	$P_{27}(MW)$	10.00
$P_8(MW)$	285.7	$P_{28}(MW)$	10.00	$P_8(MW)$	300.0	$P_{28}(MW)$	10.00
$P_9(MW)$	285.8	$P_{29}(MW)$	10.00	$P_9(MW)$	300.0	$P_{29}(MW)$	10.00
$P_{10}(MW)$	130.0	$P_{30}(MW)$	97.00	$P_{10}(MW)$	130.0	$P_{30}(MW)$	97.00
$P_{11}(MW)$	169.3	$P_{31}(MW)$	190.0	$P_{11}(MW)$	94.00	$P_{31}(MW)$	190.0
$P_{12}(MW)$	94.00	$P_{32}(MW)$	190.0	$P_{12}(MW)$	94.00	$P_{32}(MW)$	190.0
$P_{13}(MW)$	125.0	$P_{33}(MW)$	190.0	$P_{13}(MW)$	214.7	$P_{33}(MW)$	190.0
$P_{14}(MW)$	304.6	$P_{34}(MW)$	200.0	$P_{14}(MW)$	215.5	$P_{34}(MW)$	200.0
$P_{15}(MW)$	391.9	$P_{35}(MW)$	200.0	$P_{15}(MW)$	304.6	$P_{35}(MW)$	200.0
$P_{16}(MW)$	394.7	$P_{36}(MW)$	200.0	$P_{16}(MW)$	394.3	$P_{36}(MW)$	200.0
$P_{17}(MW)$	489.5	$P_{37}(MW)$	110.0	$P_{17}(MW)$	500.0	$P_{37}(MW)$	110.0
$P_{18}(MW)$	491.3	$P_{38}(MW)$	110.0	$P_{18}(MW)$	489.8	$P_{38}(MW)$	110.0
$P_{19}(MW)$	517.5	$P_{39}(MW)$	110.0	$P_{19}(MW)$	511.6	$P_{39}(MW)$	110.0
$P_{20}(MW)$	511.4	$P_{40}(MW)$	512.2	$P_{20}(MW)$	513.3	$P_{40}(MW)$	512.4
$F_{total}(R/h)$	121770 (Best Cost)			$F_{total}(R/h)$	122356 (Worst Cost)		

Table 7: (Case II) The best and worst results obtained over 30 trials with $D * 10000$ FEs

Best solution				Worst solution			
Generator	Output power	Generator	Output power	Generator	Output power	Generator	Output power
$P_1(MW)$	112.0	$P_{21}(MW)$	523.2	$P_1(MW)$	114.4	$P_{21}(MW)$	523.3
$P_2(MW)$	111.0	$P_{22}(MW)$	523.2	$P_2(MW)$	113.9	$P_{22}(MW)$	523.7
$P_3(MW)$	97.39	$P_{23}(MW)$	523.3	$P_3(MW)$	120.2	$P_{23}(MW)$	523.6
$P_4(MW)$	179.7	$P_{24}(MW)$	523.2	$P_4(MW)$	179.7	$P_{24}(MW)$	549.8
$P_5(MW)$	88.29	$P_{25}(MW)$	523.2	$P_5(MW)$	96.99	$P_{25}(MW)$	523.6
$P_6(MW)$	140.0	$P_{26}(MW)$	523.2	$P_6(MW)$	140.6	$P_{26}(MW)$	523.3
$P_7(MW)$	300.0	$P_{27}(MW)$	10.00	$P_7(MW)$	300.0	$P_{27}(MW)$	10.00
$P_8(MW)$	284.7	$P_{28}(MW)$	10.00	$P_8(MW)$	300.0	$P_{28}(MW)$	10.00
$P_9(MW)$	284.6	$P_{29}(MW)$	10.00	$P_9(MW)$	290.5	$P_{29}(MW)$	10.00
$P_{10}(MW)$	130.0	$P_{30}(MW)$	94.03	$P_{10}(MW)$	130.0	$P_{30}(MW)$	96.99
$P_{11}(MW)$	94.00	$P_{31}(MW)$	190.0	$P_{11}(MW)$	94.00	$P_{31}(MW)$	190.0
$P_{12}(MW)$	94.00	$P_{32}(MW)$	190.0	$P_{12}(MW)$	94.00	$P_{32}(MW)$	190.0
$P_{13}(MW)$	214.7	$P_{33}(MW)$	189.9	$P_{13}(MW)$	125.0	$P_{33}(MW)$	190.0
$P_{14}(MW)$	304.5	$P_{34}(MW)$	199.9	$P_{14}(MW)$	394.2	$P_{34}(MW)$	199.9
$P_{15}(MW)$	394.2	$P_{35}(MW)$	199.9	$P_{15}(MW)$	304.6	$P_{35}(MW)$	200.0
$P_{16}(MW)$	394.2	$P_{36}(MW)$	199.9	$P_{16}(MW)$	394.2	$P_{36}(MW)$	200.0
$P_{17}(MW)$	489.2	$P_{37}(MW)$	109.9	$P_{17}(MW)$	489.2	$P_{37}(MW)$	110.0
$P_{18}(MW)$	489.2	$P_{38}(MW)$	109.9	$P_{18}(MW)$	489.2	$P_{38}(MW)$	110.0
$P_{19}(MW)$	511.2	$P_{39}(MW)$	109.9	$P_{19}(MW)$	511.2	$P_{39}(MW)$	110.0
$P_{20}(MW)$	511.2	$P_{40}(MW)$	511.2	$P_{20}(MW)$	511.2	$P_{40}(MW)$	511.3
$F_{total}(R/h)$	121465 (Best Cost)			$F_{total}(R/h)$	121751 (Worst Cost)		

Table 8: (Case II) Results in literature and the solution values obtained by the proposed ABC-X .

Methods	Best Cost (R/h)	Worst Cost (R/h)	FEs
HGPSO [15]	124797	–	100000
SPSO [15]	124350	–	100000
CEP [27]	123488	126903	60000
HGAPSO [15]	122780	–	100000
FEP [27]	122680	127246	60000
MFEP [27]	122648	124356	60000
IFEP [27]	122624	125741	60000
HPSOM [15]	122112	–	100000
PSO-LRS [25]	122036	123462	20000
Improved GA [16]	121916	123334	100000
HPSOWM [15]	121915	–	100000
IGAMU [5]	121819	–	450000
HDE [29]	121813	–	42500
PPSO [4]	121788	124998	20000
DEC(2)-SQP(1) [6]	121742	122839	18000
ST-HDE [29]	121699	–	42500
NPSO-LRS [25]	121664	122982	40000
ABC-X	121770	122356	40000
ABC-X	121465	121751	400000

valve-point effect. *IEEE Transactions on power systems* 21, 2 (2006), 989–996.
[7] Swagatam Das and Ponnuthurai N Suganthan. 2010. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. *Jadavpur University, Nanyang Technological University, Kolkata* (2010).
[8] Mohammed El-Abd. 2011. Opposition-based artificial bee colony algorithm. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*. ACM, 109–116.
[9] Wei-feng Gao and San-yang Liu. 2012. A modified artificial bee colony algorithm. *Computers & Operations Research* 39, 3 (2012), 687–697.
[10] HH Happ. 1977. Optimal power dispatch?A comprehensive survey. *IEEE Transactions on Power Apparatus and Systems* 96, 3 (1977), 841–854.
[11] F. Hutter, H. H. Hoos, and K. Leyton-Brown. 2011. Sequential Model-Based Optimization for General Algorithm Configuration. In *Proc. of LION-5*. 507?523.
[12] Frank Hutter, Holger H Hoos, Kevin Leyton-Brown, and Thomas Stützle. 2009. ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research* 36, 1 (2009), 267–306.
[13] Fei Kang, Junjie Li, and Zhenyue Ma. 2011. Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Information Sciences* 181, 16 (2011), 3508–3531.
[14] Dervis Karaboga, Beyza Gorkemli, Celal Ozturk, and Nurhan Karaboga. 2014. A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review* 42, 1 (2014), 21–57.
[15] Sai-Ho Ling, Herbert HC Iu, Kit Yan Chan, Hak-Keung Lam, Benny CW Yeung, and Frank H Leung. 2008. Hybrid particle swarm optimization with wavelet mutation and its industrial applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 38, 3 (2008), 743–763.
[16] Sai-Ho Ling and FH Frank Leung. 2007. An improved genetic algorithm with average-bound crossover and wavelet mutation operations. *Soft Computing* 11, 1 (2007), 7–31.
[17] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Leslie Pérez Cáceres, Mauro Birattari, and Thomas Stützle. 2016. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives* 3 (2016), 43–58.
[18] Tahir Nadeem Malik, Azzam ul Asar, Mudasser F Wyne, and Shakil Akhtar. 2010. A new hybrid approach for the solution of nonconvex economic dispatch problem with valve-point effects. *Electric Power Systems Research* 80, 9 (2010), 1128–1136.
[19] Mostafa Modiri-Delshad and Nasrudin Abd Rahim. 2014. Solving non-convex economic dispatch problem via backtracking search algorithm. *Energy* 77 (2014), 372 – 381. DOI: http://dx.doi.org/10.1016/j.energy.2014.09.009
[20] W Ongsakul and T Tantimaporn. 2006. Optimal power flow by improved evolutionary programming. *Electric Power Components and Systems* 34, 1 (2006), 79–95.
[21] S Özyön, C Yaşar, G Özcan, and H Temurtaş. 2011. An artificial bee colony algorithm (abc) aproach to nonconvex economic power dispatch problems with valve point effect. In *National Conference on Electrical, Electronics and Computer (FEEB11) pp.* 294–299.
[22] S Özyön, C Yaşar, and H Temurtaş. 2011. Differential evolution algorithm approach to nonconvex economic power dispatch problems with valve point effect. In *6th International Advanced Technologies Symposium (IATS11).* 181–186.
[23] Serdar Özyön, Celal Yaşar, and Hasan Temurtaş. 2011. Particle swarm optimization algorithm for the solution of nonconvex economic dispatch problem with valve point effect. In *Electrical and Electronics Engineering (ELECO), 2011 7th International Conference on.* IEEE, I–101.
[24] Jong-Bae Park, Ki-Song Lee, Joong-Rin Shin, and Kwang Y Lee. 2005. A particle swarm optimization for economic dispatch with nonsmooth cost functions. *IEEE Transactions on Power systems* 20, 1 (2005), 34–42.
[25] A Immanuel Selvakumar and K Thanushkodi. 2007. A new particle swarm optimization solution to nonconvex economic dispatch problems. *IEEE transactions on power systems* 22, 1 (2007), 42–51.
[26] A Immanuel Selvakumar and K Thanushkodi. 2008. Anti-predatory particle swarm optimization: solution to nonconvex economic dispatch problems. *Electric Power Systems Research* 78, 1 (2008), 2–10.
[27] Nidul Sinha, R Chakrabarti, and PK Chattopadhyay. 2003. Evolutionary programming techniques for economic load dispatch. *IEEE Transactions on evolutionary computation* 7, 1 (2003), 83–94.
[28] C Thitithamrongchai and B Eua-Arporn. 2007. Self-adaptive differential evolution based optimal power flow for units with non-smooth fuel cost functions. *Journal of Electrical Systems* 3, 2 (2007), 88–99.
[29] SK Wang, JP Chiou, and CW Liu. 2007. Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm. *IET Generation, Transmission & Distribution* 1, 5 (2007), 793–803.
[30] Celal Yaşar and Serdar Özyön. 2011. A new hybrid approach for nonconvex economic dispatch problem with valve-point effect. *Energy* 36, 10 (2011), 5838–5845.