Automatic Selection of Optimization Algorithms for Energy Resource Scheduling using a Case-Based Reasoning System

Ricardo Faia[1], Tiago Pinto[1,2], Tiago Sousa[1], Zita Vale[1] and Juan Manuel Corchado[21]

[1]GECAD – Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Institute of Engineering, Polytechnic of Porto (ISEP/IPP), Porto, Portugal
{rfmfa, tabsa, tmcfp, zav}@isep.ipp.pt

[2]BISITE Research Centre, University of Salamanca (USAL), Calle Espejo, 12, 37007 Salamanca, Spain
{tpinto, corchado}@usal.es

Abstract. This paper proposes a case-based reasoning methodology to automatically choose the most appropriate optimization algorithms and respective parameterizations to solve the problem of optimal resource scheduling in smart energy grids. The optimal resource scheduling is, however, a heavy computation problem, which deals with a large number of variables. Moreover, depending on the time horizon of this optimization, fast response times are usually required, which makes it impossible to apply traditional exact optimization methods. For this reason, the application of metaheuristic methods is the natural solution, providing near-optimal solutions in a much faster execution time. Choosing which optimization approaches to apply in each time is the focus of this work, considering the requirements for each problem and the information of previous executions. A case-based reasoning methodology is proposed, considering previous cases of execution of different optimization approaches for different problems. A fuzzy logic approach is used to adapt the solutions considering the balance between execution time and quality of results

Keywords: Case Base Reasoning, Optimization Algorithm, Classification

1 Introduction

One of the main objectives of computational intelligence is to impart systems with the ability to reproduce human-like reasoning. Case-based Reasoning (CBR) is an Artificial Intelligence (AI) approach to learning and problem solving based on the past experience, which is usually stored in a case-base (CB) [1]. CBR also captures new knowledge, making it immediately available for solving new problems. AI techniques have excelled in problem-solving as a good solution over conventional techniques.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 641794 (project DREAM-GO) and a grant agreement No 703689 (project ADAPT); and from FEDER Funds through COMPETE program and from National Funds through FCT under the project UID/EEA/00760/2013

CBR has been used in many application domains, one of them being in solving power and energy systems. In [2] a CBR system for building energy prediction is proposed, with the aim at identifying operation issues and proposing better operating strategies. Simplified models based on CBR to predict the hourly electricity consumption of an institutional building are proposed in [3]. A CBR method providing online decision-making for optimization of coal-blend combustion was investigated in [4]. The estimation of the energy performance of new buildings using CBR is studied in [5]. These are relevant contributions that cover some problems in the energy domain. However, many urgently needed issues in this area are still not addressed, such as the energy resource operation and planning.

The Optimal Resource Scheduling (ORS) problem, however, requires extremely heavy computational models, depending on the amount and diversity of the considered resources, and on the depth of network validation and analysis. For this reason deterministic approaches are, most of the times, inadequate [6]. Metaheuristics are proving to be the most suitable alternative, since they are able to reach near-optimal solutions in much faster execution times [7]. These algorithms do not guarantee the optimum global solution, but in turn the response time is much lower compared to the traditional exact algorithms that guarantee it. Many of these methods have also been applied in the resolution of the ORS problem [6, 8].

The question remains, however, on how to make most use of the whole set of available algorithms, depending on the needs and characteristics of each problem. Metaheuristic methods are able to provide approximate solutions in fast execution times, while deterministic approaches need larger times to compute, but are able to provide the optimal solution. Some work has already been made with the application of CBR systems to similar problems, namely in [9], which presents a study to try finding the ideal parameters to apply in evolutionary algorithms. In this work a CBR methodology is used to estimate the best parameter setting for maximizing the performance of evolutionary algorithms. However, in the present work authors propose, not only to adapt the parameterization of a certain algorithm to meet the requirements of execution time versus quality of results, but also to choose the most appropriate algorithm and respective parameterization taking into account the availability of several distinct algorithms of different natures.

This paper thus proposes a CBR based approach that, given the problem characteristics and requirements, and considering an historic CB log of past executions of each algorithm to solve the energy resource optimization problem with different settings, suggests the most appropriate algorithm to apply and the respective parameterization. A problem-driven approach is applied in the retrieve and revise phases, considering the specificities of the different considered variables, and a fuzzy logic based approach [10, 11] is used in the revise phase to adapt the solutions to the requirements of the new problem, namely considering the balance between execution time and quality of results.

After this introductory section, section 2 describes the CBR approach proposed in this paper. Section 3 presents the experimental findings of the application of the proposed approach to a historic CB log of previous executions done by the authors' research team. Finally, section 4 presents the most relevant conclusions of this work.

2 Proposed CBR approach

In this problem, each historic case contains the set of information that is presented in Table 1. The process for which the CBR is oriented refers to choosing the method to use in the problem characterized with different parameters expressed in Table 1. There are 3 types of classification: type A indicates the parameters used for assessing the similarity between case studies, type B indicates the parameters used to determine the quality of each algorithm, and type C are the output parameters. The ID refers to the identification of each case study. The ORS problem contains the type of objective function, where 1 means single-objective optimization problem and 2 corresponds to multi-objective optimization problem. The ORS function parameter refers to which is the ORS problem for the corresponding case study, as can assume 4 states: 1 means minimizing the cost, 2 is minimizing the cost and GHG emissions, 3 is minimizing the cost and demand difference, and 4 is minimizing the cost and voltage deviation.

Table 1. Case structure

Type of parameter		Designation
		ID
A (Similarity)	**A1**	ORS problem
		ORS function
	A2	Period
		Bus
		No. DG quadratic
		EVs
		Congestion power (kW)
B (Quality)		Objective function
		Execution time (s)
C (Decision)		Algorithm
		Parameters

The Period refers to the number of periods of the ORS problem, e.g. 24 hourly periods. The Bus parameter corresponds to the number of buses that compose the distribution network of the case study. This parameter influences the execution time of the algorithms. The No. DG quadratic refers to the number of DG units using the quadratic function for their operation cost. The parameter EVs indicates the number of electrical vehicles used in each case study. The Congestion power refers to the average amount of congestion power of the case study. All these parameters are used by the CBR systems to choose the similar cases. The ORS problem and ORS function parameters have a distinct classification of **A1**, because they are firstly used to filter the cases that were solved for similar ORS problems, i.e. it is mutually exclusive: either a past case is of the same type as the case to be solved or not. On the other hand, all the other Similarity (type **A**) parameters, are classified as **A2**, which means that the similarity between past and current case can be calculated and represented by a value (in this case as a percentage of similarity for each of these parameters).

The type **B** parameters are those that enable determining the quality of the results. The Objective function indicates the objective function result obtained by the algorithm. In the case of multi-objective problems, both objective functions are stored in this parameter. Execution time contains the time that the algorithm took to solve the ORS problem for the corresponding case study. These two parameters are used to select the best algorithms after the CBR approach obtains the similar historic cases by analyzing the type **A** parameters.

Once the quality of the solutions (type **B**) of the similar cases (type **A**) is assessed, a decision is made on which methods and respective parameterizations are the most adequate (using the type **C** parameters). The Algorithm parameter is the name of the algorithm used to solve the case study. Parameters contains the parameters used in each algorithm to solve the historic case, as can be seen in Table 1. These two last parameters are type **C**, because they contain the information on which algorithm and parameters were used to solve the problem. After describing the content of each parameter in the historic cases, the different phases of CBR system is describe in following steps.

2.1 Retrieve

Analyse the ***A1*** parameters for selecting the cases containing the same type of problem (ORS problem) and type of function (ORS function). Each historic case is filtered according to the value of the ORS problem parameter, given by equation (1).

$$F_{HC}^{1} = \begin{cases} HC_j \, , H_j\big(A1_{(i)}\big) = CS\big(A1_{(i)}\big) \\ 0 \, , Otherwise \end{cases} \quad (1)$$
$$\forall j \in \{1, \dots N_{HC}\}; i = \{ORS\ problem\}$$

Where, F_{HC}^{1} contains the historic cases that were filtered by equation (1). The terms *HC* and *CS* correspond to the historic case and current case study, respectively. The index *j* refers to the *ID* of each historic case, while index *i* corresponds to *ORS problem* parameter. N_{HC} refers to the total number of historic case studies in the database.

Then, the historic cases filtered as (F_{HC}^{1}) are also filtered if they have the same value for the ORS function, by equation (2).

$$F_{HC}^{2} = \begin{cases} F_{HC(j)}^{1}, F_{HC(j)}^{1}\big(A1_{(i)}\big) = CS\big(A1_{(i)}\big) \\ 0 \, , Otherwise \end{cases} \quad (2)$$
$$\forall j \in \{1, \dots, N_{HC}^{F1}\}; i = \{ORS\ function\}$$

Where, F_{HC}^{2} contains the historic cases that were filtered by equation (2), and index *i* corresponds to *ORS function* parameter. N_{HC}^{F1} corresponds to the total number of historic cases filtered in (1). The historic cases with *ORS problem* equal to 2 (multi-objective problems) that have *ORS function* equal to 2, 3 or 4, i.e. minimizing the cost and other competitive objective, are all considered for a current case study with the same *ORS problem* and containing the same information for the *ORS function* parameter (2, 3 or 4). The idea with this condition is to separate problems with distinct objective function.

Determine the cases that are similar to the current one through the use of *A2* parameters. For each historic case ($F_{HC(j)}^{2}$) the similarity percentage of each *A2* parameter ($P_{HC(j)}^{A2(i)}$) is calculated by equation (3).

$$P_{HC(j)}^{A2(i)} = \begin{cases} \dfrac{F_{HC(j)}^{2}\big(A2_{(i)}\big)}{CS\big(A2_{i}\big)}, CS\big(A2_{(i)}\big) \geq F_{HC(j)}^{2}\big(A2_{(i)}\big) \\ \dfrac{CS\big(A2_{i}\big)}{F_{HC(j)}^{2}\big(A2_{(i)}\big)}, F_{HC(j)}^{2}\big(A2_{(i)}\big) \geq CS\big(A2_{(i)}\big) \end{cases} \quad (3)$$
$$\forall j \in \{1, \dots, N_{HC}^{F2}\}; \forall i \in \{1, \dots, N_{A2}\}$$

where, N_{HC}^{F2} is equal to the number of historic cases filtered in previous step by equation (2), while N_{A2} corresponds to the total number of ***A2*** parameters. The similarity percentage is calculated by dividing the value of each ***A2*** parameter ($A2_{(i)}$) between the historic and current cases (or *vice versa* - allowing avoiding similarities over than 100%). Then, the average similarity is determined, which corresponds to the similarity percentage of each historic case, and is given by equation (4)

$$PC_{HC(j)} = \frac{1}{N_{A2}} \times \sum_{i=1}^{N_{A2}} P_{HC(j)}^{A2(i)} \qquad (4)$$
$$\forall j \in \{1, \ldots, N_{HC}^{F2}\};\ \forall i \in \{1, \ldots, N_{A2}\}$$

For a current case with parameters (e.g. *Period*, *Bus* or *EVs*) very close to a historic one, the similarity percentage of each historic case j ($P_{HC(j)}$) will tend to 100%. Finally, filter the historic cases with a similarity percentage ($P_{HC(j)}$) higher or equal to 75%.

$$SC_j = \begin{cases} F_{HC(j)}^{2}, P_{HC(j)} \geq 0.75 \\ 0, Otherwise \end{cases} \qquad (5)$$
$$\forall j \in \{1, \ldots, N_{HC}^{F2}\}$$

Where, set SC_j contains all the similar cases.

2.2 Reuse

Extract the algorithms that are used in the similar historic cases and their quality parameters (type ***B*** of Table 1). The same algorithms with different parameters can be considered multiple times, if it is used in multiple similar cases. Steps 3, 4 and 5 are only applied if there is any similar historic case study, otherwise, the CBR systems will select all the algorithms that can solve the chosen ORS problem.

First, filter the algorithms with different parameterization that were used to solve the similar cases, as described in equation (6).

$$Method = SC_j(C_{(i)}) \qquad (6)$$
$$\forall j \in \{1, \ldots, N_{HC}^{F2}\}; i = \{Methodology; Parameters\}$$

Where, index i indicates the parameters of type C from Table 1. Second, the average execution time (***B*** parameter) of all cases solved by the same algorithm and parameterization (equation (6)) is determined, because the same algorithm and parameterization might be used by multiple similar cases, which is given by (7).

$$Time_{Met} = \frac{1}{N_{SC}^{Met}} \sum_{j \in SC^{Met}} SC_j(B_{(i)}) \qquad (7)$$
$$\forall\ Met \in \{1, \ldots, N_{Met}\}; i = \{Execution\ time\}$$

Where, SC^{Met} refers to the set of all similar cases that were solved by the same algorithm and parameterization with index *Met*. N_{SC}^{Met} contains the number of similar cases solved by the same algorithm and parameterization with index *Met*. N_{Met} refers to the total number of algorithms with different parameterization in (6).

Finally, the average objective function (type B parameter) of all cases solved by the same algorithm and parameterization is also calculated using the previous equation (7). These values are stored in variable Fun_{Met}. Before applying this equation, the objective function values are normalized, because the cases can have objective function values with different magnitudes. The number of considered historical cases is crucial, because with many cases this process can become heavy and slow, so a good historical cases selection (retain phase) is important.

2.3 Revise

Choose the most appropriate algorithms to solve the current case study through the use of a fuzzy method. The variables $Time_{Met}$ and Fun_{Met}, determined in previous step, are used by the fuzzy method. First, create the membership function (μ^{Time}) related to time (efficiency), which is represented in Fig. 1.

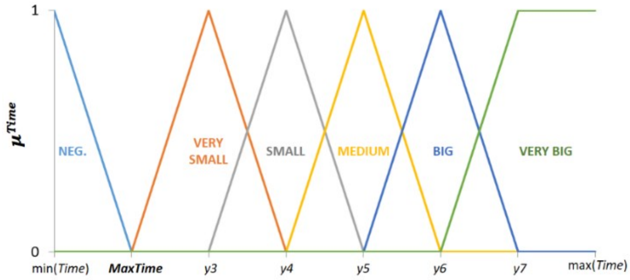

Fig. 1. Membership function of efficiency

The membership function has dynamic intervals to be adapted to every case study. The membership function starts at the minimum *Time* among all methods equation (7) , the second value of this function is the maximum time defined by the VPP in the input data, which is represented as *MaxTime*. The maximum *Time* occupies the other extreme of the membership function. The remaining values ($y3$, $y4$, $y5$, $y6$ and $y7$) are proportionally distributed between the *MaxTime* and the maximum time. The $Time_{Met}$ equation (7) of each method *Met* is classified based on this membership function, which indicates how much far the *Time* is from the *MaxTime* (i.e. NEGATIVE, VERY SMALL, SMALL, MEDIUM, BIG or VERY BIG).

Secondly, the membership function (μ^{Fun}) related with objective function (effectiveness), is created, which is represented in Fig. 2.

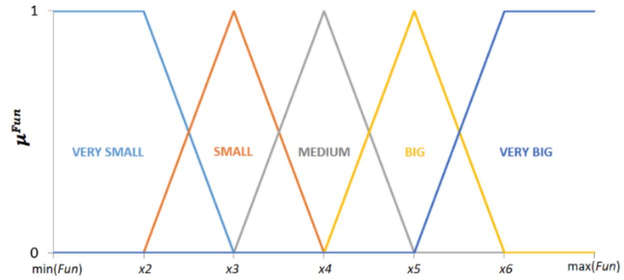

Fig. 2. Membership function of effectiveness

This membership function also has dynamic intervals, as it starts with the minimum *Fun* among all methods, while the maximum *Fun* is placed in the other extreme of the function. Just like the previous one, the remaining values are proportionally distributed between the minimum and maximum *Fun*. The Fun_{Met} is be classified based on this membership function, which also indicates how far the Fun_{Met} of each method *Met* is from the minimum *Fun*.

Then, select the algorithms considering the μ^{Time} and μ^{Fun} classifications by equation (8).

$$Method = \begin{cases} Method_{Met}, \mu_{Met}^{Time} = NEGATIVE \\ Method_{Met}, \begin{array}{l} \mu_{Met}^{Time} = \{VERY\ SMALL; SMALL\} \\ \mu_{Met}^{Fun} = \{VERY\ SMALL; SMALL\} \end{array} \end{cases} \quad (8)$$
$$\forall\ Met\ \in \{1, \ldots, N_{Met}\}$$

The methods with μ^{Time} equal to NEGATIVE, which means an execution time below the *MaxTime*, are accepted to solve the current case, without considering their effectiveness classification (μ^{Fun}). The other methods with a time slightly higher than *MaxTime*, which have VERY SMALL and SMALL efficiency classification (μ^{Time}), are accepted if they also have an objective function close to the minimum, which are VERY SMALL and SMALL classifications for the effectiveness membership function (μ^{Fun}). All methods that are classified as bigger are automatically excluded, since their execution time is too big to useful for the considered problem or the results quality is too low (big difference from the best methods).

Finally, the fuzzy confusion matrix, which joins the two membership functions (μ^{Time} and μ^{Fun}), is applied to take actions regarding the methods with VERY SMALL and SMALL classifications. Basically, these methods are changed in terms of their parameterization to reach a lower execution. The amount of these changes will be given by the fuzzy confusion matrix, which can be consulted in the fuzzy confusion matrix presented in Table 2. This enables to consider methods that would be excluded because they are above *MaxTime*, but have good objective function results.

Table 2. Fuzzy confusion matrix for small and very small classifications of effectiveness and efficiency

Efficiency Classification	Effectiveness Classification	Action to take
VERY SMALL	VERY SMALL	Very small reduction
	SMALL	Small reduction
SMALL	VERY SMALL	Small reduction
	SMALL	Big reduction

2.4 Retain

Evaluate the possibility of storing the results of the current case study in the database of historic cases. Determine the similarity of the current case study (P_{CS}) by applying the equations (4) and (5), include the current case in the database of historic cases, if its similarity percentage is lower or equal to 95%, defined as equation (9).

$$HC = \begin{cases} CS, P_{CS} \leq 0.95 \\ 0, Otherwise \end{cases} \quad (9)$$

A current case with a percentage higher than 95% is not adding new value to the historic cases, since it is only bringing useless information to the processed.

3 Results

This section presents the experimental findings concerning the application of the proposed methodology to a new case. 21 previous cases are considered in the CB, which refer to different executions of several algorithms with different parameterizations, to different variations of the ORS problem. The new case is defined by the next conditions: {ID=_; ORS problem=1; ORS function=1; Period=24; Bus=37; No DG quadratic= 3; EV's=2000 and Congestion power= 730}, B and C parameters present in Table 1 will be find by CBR methodology.

To carry out the CBR process, the new case must contain all elements of group **A** (Similarity). Table 3 shows the results of the different methods selected by the equations corresponding to the group of similarities. Please refer to [8] for a detailed description of the optimization methods shown in the last column of Table 3.

The results of equations (1), (2), (4) and (5) are related to the retrieve process, and equation (6) is already the initial phase of the reuse process, where similar cases are identified. As can be seen, the cases filtered by the ORS problem and ORS function are the same (20 cases). By applying the calculation of the total similarity (equation (5)) 4 cases are excluded, being 16 cases considered similar to the new case.

Table 3. Results similarity

Equation (1) - case ID	Equation (2) - case ID	Equation (4)	Equation (5)	Equation (6)
1	1	0,4842822	X	-
2	2	0,2817593	X	-
4	4	0,9922183	✓	RSA
5	5	0,9953471	✓	HSA
6	6	0,9956742	✓	ERS^2A
7	7	0,995853	✓	$PERS^2A$
8	8	0,9958794	✓	SADT
9	9	0,9942107	✓	GA
10	10	0,9928798	✓	PSO
11	11	0,9953506	✓	PERSGA
12	12	0,9953364	✓	PERSPSO
13	13	0,9956334	✓	GADT
14	14	0,995663	✓	PSODT
15	15	0,9960788	✓	MINLP
16	16	0,9613657	✓	$PERS^2A$
17	17	0,9615992	✓	SADT
18	18	NaN	X	-
19	19	0,9988149	✓	$PERS^2A$
20	20	0,9998657	✓	SADT
21	21	NaN	X	-

Table 4 presents the efficiency classification, as result of the efficiency fuzzy variable, and the effectiveness classification of the application of each of the selected methods to the selected cases. In Table 4, the 16 similar cases are filtered by method, and may have different configurations within the same method. In this case, the average of these configurations (execution time and objective function) is made. In Table 4, 12 methods are present which means that there are repeated methods. Being that $PERS^2A$

and SADT repeated three times. The values are sorted by execution time value, in an ascending order. The fuzzy results related to the value of the objective function, i.e. the effectiveness of each method and respective parametrization in solving the previous problem identified as similar to the new case. Table 4 also presents the decision results, which are a direct output from the confusion matrix that combines the fuzzy results for efficiency and effectiveness of each method.

Table 4. Efficiency and effectiveness results

Method	Equation (7) Time (s)	Equation (7) Objective function	Confusion Matrix		Type of modification
			Efficiency	Effectiveness	
ERS^2A	54,1	23944,94	NEG.*		-
RSA	174,28	24375,45	NEG.*		-
$PERS^2A$	189,43	25415,76	NEG.*		-
SADT	393,4367	25446,97	NEG.*		-
PERSPSO	482,88	23986,35	VERY SMALL	VERY SMALL	Very small reduction
PSODT	544,11	23946,32	VERY SMALL	VERY SMALL	Very small reduction
PSO	550,91	24291,86	VERY SMALL	SMALL	Small reduction
HSA	598,35	23985,04	VERY SMALL	VERY SMALL	Very small reduction
PERSGA	635,67	23984,61	VERY SMALL	VERY SMALL	Very small reduction
GADT	673,47	23949,94	VERY SMALL	VERY SMALL	Very small reduction
GA	1731,54	24125,38	-	-	Excluded
MINLP	94941,85	23895,53	-	-	Excluded

*Equation (8)

In Table 4 are expressed the decision results obtained by the CBR system. As it can be seen, if the classification in the efficiency process is Negative, the method will be accepted without any change. On the other hand, if the classification is any other, the value of the objective function is analyzed, the classifications medium, big and very big, are excluded at the beginning. The confusion matrix is only executed for the methods classified as very small and small. The result of the confusion matrix gives the type of modification that is required to execute so that the given method can obtain an execution time value lower than the one defined as *MaxTime* by 400 seconds.

By applying the rules of the fuzzy processes, the possible methods to solve the problems went from 12 to 10, and the MINLP and GA were excluded. The ERS2A, RSA, PERS2A and SADT methods were accepted without any change. The remaining methods are subjected to a certain type of change to be performed, which regards the adaptation of the method's parameterization, e.g. using a smaller number of iterations or a smaller number of particles in the PSO to achieve faster results.

4 Conclusions

This paper presented a CBR methodology to support the choice of the methods to use in solving the energy ORS problem. The proposed method includes a fuzzy based process to determine the changes in parameterization that should be applied to each method that is considered promising to solve a new case with specific characteristics.

It is clear that this method brings advantages when compared to a manual process, because choosing manually hardens the effectiveness of the choice, and the time spent, e.g. in the choice of parameters.

The performance of CBR systems is highly correlated with the number of cases that it imbues. Even so, the presented results suggest as final result a considerable number of methods to solve the problem, all of which with expected small execution times and good quality of results for the envisaged problem. This means that the presented methodology was effective in the selection and classification of the methods. The modifications to be performed in the methods, as result from the fuzzy process, enlarge the scope of possible methods to be applied, as rather than excluding such methods for being just a bit slower or presenting a bit worst quality of results than other methods, it still considers the most promising ones as possible solutions, subject to a degree of changes that would make them suitable to solve the problem as well.

As future work, it is intended to develop a method for deciding which parameters to modify to obtain the given value of maximum execution time, according to the results of the fuzzy process. It is also proposed to apply decision trees in the process of retrieve. Finally, the process of reviewing can be enhanced with the help of an expert, in order to build an expert system to perform the revision of the changed parameters.

References

1. Aamodt A, Plaza E (1994) Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun IOS Press 7:39–59.
2. Monfet D, Corsi M, Choinière D, Arkhipova E (2014) Development of an energy prediction tool for commercial buildings using case-based reasoning. Energy Build 81:152–160. doi: 10.1016/j.enbuild.2014.06.017
3. Platon R, Dehkordi VR, Martel J (2015) Hourly prediction of a building's electricity consumption using case-based reasoning, artificial neural networks and principal component analysis. Energy Build 92:10–18. doi: 10.1016/j.enbuild.2015.01.047
4. Xia J, Chen G, Tan P, Zhang C (2014) An online case-based reasoning system for coal blends combustion optimization of thermal power plant. Int J Electr Power Energy Syst 62:299–311. doi: 10.1016/j.ijepes.2014.04.036
5. Hong T, Koo C, Kim D, et al (2015) An estimation methodology for the dynamic operational rating of a new residential building using the advanced case-based reasoning and stochastic approaches. Appl Energy 150:308–322. doi: 10.1016/j.apenergy.2015.04.036
6. Soares J, Sousa T, Morais H, et al (2013) Application-Specific Modified Particle Swarm Optimization for energy resource scheduling considering vehicle-to-grid. Appl Soft Comput 13:4264–4280. doi: 10.1016/j.asoc.2013.07.003
7. Baños R, Manzano-Agugliaro F, Montoya FG, et al (2011) Optimization methods applied to renewable and sustainable energy: A review. Renew Sustain Energy Rev 15:1753–1766. doi: 10.1016/j.rser.2010.12.008
8. Sousa T, Morais H, Castro R, Vale Z (2016) Evaluation of different initial solution algorithms to be used in the heuristics optimization to solve the energy resource scheduling in smart grids. Appl Soft Comput 48:491–506. doi: 10.1016/j.asoc.2016.07.028
9. Yeguas E, Luzón MV, Pavón R, et al (2014) Automatic parameter tuning for Evolutionary Algorithms using a Bayesian Case-Based Reasoning system. Appl Soft Comput 18:185–195. doi: 10.1016/j.asoc.2014.01.032
10. Faia R, Pinto T, Vale Z (2015) Dynamic Fuzzy Estimation of Contracts Historic Information Using an Automatic Clustering Methodology. In: Bajo J, Hallenborg K, Pawlewski P, et al (eds) Highlights PAAMS Collect. SE - 23. Springer International Publishing, pp 270–282
11. Chang P-C, Fan C-Y, Dzan W-Y (2010) A CBR-based fuzzy decision tree approach for database classification. Expert Syst Appl 37:214–225. doi: 10.1016/j.eswa.2009.04.062