Applied Soft Computing 48 (2016) 193–206

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Differential evolution of fuzzy controller for environmentally-powered wireless sensors

M. Prauzek[a,b], P. Krömer[a,b], J. Rodway[a], P. Musilek[a,*]

[a] Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada
[b] VŠB – Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava, Poruba, Czech Republic

ARTICLE INFO

Article history:
Received 23 January 2015
Received in revised form 22 June 2016
Accepted 29 June 2016
Available online 4 July 2016

Keywords:
Sensor network
Wireless node
Environmental monitoring
Energy management
Fuzzy control
Optimization
Differential evolution

ABSTRACT

Environmentally-powered wireless sensors use ambient energy from their environment to support their own energy needs. As such, they must operate without significant maintenance or user supervision. Due to the stochastic availability of ambient energy, its harvesting, storage and consumption must be managed by an efficient and robust controller that maintains data collection and transmission rates at desired levels, while maximizing the useful operational time of the system. To accomplish this task, the control system must observe the state of charge of an internal energy storage device, and consider the amount of energy available for harvest in the future. At the same time, the complexity of the controller must be limited so that it can be implemented on the simple embedded system of the sensor hardware. This paper presents a comprehensive synthesis of desired behavior of such controllers, and describes procedures for their design and optimization through an evolutionary fuzzy approach. The main contribution is the formalization of design objectives and development of the fitness function that drives the optimization process. Additional contributions include a comprehensive evaluation of several soft computing optimization approaches, thorough analysis of the optimized controller, its comparison to baseline control strategies, and validation of its operation with real energy availability forecasts.

1. Introduction

Wireless sensors and sensor networks [2] are often used for long-term environmental monitoring applications that take place in remote, inaccessible locations with variable ambient conditions. These environmental monitoring systems must be capable of reliable autonomous operation, and independent of externally supplied energy or human intervention. To achieve this level of self-sustainability, such systems are usually powered by energy harvested directly from the deployment environment [9]. Design of environmentally-powered wireless sensors is a complex problem with a number of conflicting goals and several design and implementation constraints [59]. The most common concerns include desired sensor sampling rates and acceptable delays between data transmissions.

Wireless sensor devices are usually designed as low-power embedded systems with a low-performance microcontroller unit (MCU) that cannot implement computationally extensive control algorithms [50]. At the same time, the complexity of the control algorithm affects the overall power consumption of the system [7]. In addition, the algorithms must be fault tolerant to satisfy the stringent requirements for autonomy and dependability. Many existing approaches include conventional rule-based algorithms [66] implemented using state machines [26] or heuristic control methods [48]. The system proposed in this article uses fuzzy control that is easy to design and implement on limited hardware, and able to deal with imprecise or missing data. The optimization method of differential evolution is then used to adapt the fuzzy control system to match the planned deployment environment for best performance. The application-specific fitness function used for optimization of the fuzzy control system is an important contribution of this work.

The dependence of environmentally-powered devices on the ambient energy suggests the possibility to use energy availability prediction to estimate the state of charge of energy storage devices [12], to improve the device performance [46], or to reconfigure the entire system [39]. In this contribution, prediction of energy available over a time horizon serves as an additional input to the fuzzy energy manager. The proposed approach is illustrated through the process of predictive fuzzy controller design. It extends

* Corresponding author. Tel.: +1 780 492 5368.
E-mail addresses: michal.prauzek@vsb.cz (M. Prauzek), pavel.kromer@vsb.cz (P. Krömer), jrodway@ualberta.ca (J. Rodway), petr.musilek@ualberta.ca (P. Musilek).

http://dx.doi.org/10.1016/j.asoc.2016.06.040
1568-4946/

previous studies by the authors on optimization of fixed parameters of wireless sensor nodes [37] and on optimization of a simple fuzzy controller without energy prediction [36]. The structure of the predictive controller is first devised manually, considering the nature of the energy management problem. The controller parameters are then optimized using differential evolution and three other soft computing methods. Strong emphasis is placed on a comprehensive evaluation of the evolved complex, predictive control strategy.

The performance of such an extended fuzzy controller is thoroughly tested through a series of simulation experiments based on an accurate hardware model of a wireless sensor node and energy availability data measured at the intended deployment site. Such experiments can provide useful information about energy management strategy performance, in a short time and at a low cost. Given the realistic models and data used in the simulations, it is reasonable to expect that the developed controllers will be applicable in a real deployment environments. They also serve to determine the optimal length of the prediction horizon. The best performing controller is then tested using real forecasts of energy availability, rather than historical measurements. Testing results confirm validity of the proposed approach and the practical value of predictive energy management for environmentally-powered wireless sensors.

This paper is organized into six sections. Section 2 provides a brief overview of the state of the art in energy management for wireless sensors and in evolutionary fuzzy systems. Section 3 a model of the optimization problem, including synthesis of energy management strategies and design of corresponding fuzzy logic controllers. Section 4 then details controller optimization by several soft computing methods, concentrating on differential evolution. Results of simulation experiments are described in detail and discussed in Section 5. The final Section 6 brings major conclusions and outlines possible directions for future work.

2. Related work

2.1. Energy management in wireless sensors

Wireless sensor networks (WSNs) are composed of a large number of wireless sensor nodes deployed inside, or in close proximity of, an area of interest [2]. Such sensor nodes are implemented as embedded systems with multiple functions, including sensing, data management, and wireless communication. They can be placed in remote locations with limited access and without energy infrastructure. To allow their sustained operation under such conditions, wireless sensor nodes are often powered using ambient energy harvested from their deployment environment [9,20,23,46,59], leading to so called environmentally-powered wireless sensor nodes (EPWSN). Energy harvesting (EH) reduces the environmental footprint of sensor nodes [63], contributes to their energy neutrality [62], and eventually leads to perpetual networks [20].

The efficient use of EH systems requires sophisticated energy management and control [63]. This involves both the node level (e.g. adaptive duty cycling [67,69] and task scheduling [27,56]), and the network level (e.g. media access control, routing, and time synchronization protocols [59]). It is a complex optimization problem with a simple objective: to maintain data sensing and transmission rates at desired levels, while maximizing the useful operational time of the system through optimal energy harvesting and consumption [63]. It must consider the properties of particular ambient energy sources (such as stochasticity and periodicity) [35], as well as the requirements of the application domain (such as robustness, reliability [5,68], and communication throughput [35]).

Various adaptive energy management strategies have been proposed for EH system control. They can be classified as local (when each node considers only local information), global (when actions are selected based on assumed complete knowledge about the entire network), and hybrid approaches that combine both [27,68]. In general, they can use information about the state of node hardware (e.g. remaining energy level [67]), heuristic information (e.g. expected usage of monitored rooms [27]), assumptions based on historical information (e.g. average amount of ambient energy available at certain location and time), or predictions (e.g. forecasting energy consumption and harvesting [63,69]). Many recent studies have concluded that predictive strategies lead to a better utilization of available ambient energy [27,46,56,63,67,69].

2.2. Evolutionary design of fuzzy systems

Artificial evolution is a well-established approach to design and optimization of intelligent systems [4,17]. Biologically inspired methods have been identified as a tool suitable for construction and tuning of accurate and interpretable fuzzy systems [17]. Differential evolution (DE) is a relatively recent evolutionary method noted for its simplicity, ease of use, good performance, and an excellent track record of real-world applications. In the last decade, DE has been used also to tailor different types of fuzzy systems [16,22,38,43,44] and neural-fuzzy systems [14,32,70] for various application [28,49].

Cheong and Lai [16] proposed the use of the DE as a part of automated design of hierarchical fuzzy controllers. The optimization algorithm was used to coordinate the outputs of subcontrollers within the hierarchical system and to optimize rule base templates. The validity and usefulness of this approach was evaluated on the classical cart-pole (inverted pendulum) control problem and it was shown that DE is a valuable method for design of symmetrical fuzzy controllers.

Another work by Eftekhari et al. [22] used DE for constructing interpretable fuzzy inference systems. More specifically, it was used to simplify fuzzy models initially generated from data by subtractive clustering and to optimize centers and widths of fuzzy membership functions of the simplified system. The proposed method used an interpretability measure to express the fitness of the system under simplification, and mean squared error as a measure of accuracy in the final step.

The evolution of fuzzy rule-based meta-schedulers for grid computing by a DE-inspired approach is due to [49]. The evolved control system used Mamdani-type fuzzy rules with Gaussian membership functions. The traditional DE was modified for this application. Every fuzzy rule was encoded as a DE vector and the entire population represented the rule-base, following the Michigan approach well-known from the domain of genetic fuzzy systems [18]. The study showed that the proposed method outperformed traditional learning strategies (i.e. the Pittsburg approach). The optimized meta-scheduler was better, in terms of training fitness, than simple scheduling strategies.

The study of [28] showed the ability of DE to tune fuzzy controllers for financial market modelling. It evolved encoded membership functions and fuzzy rules using a problem specific version of DE and showed that such optimized controller is able to embrace the complex, non-linear, and dynamic nature of the problem and that the evolved models are able to optimize market portfolio return for different types of markets and different portfolia.

Oh et al. [43] suggested the use of adaptive DE for optimization of a cascade fuzzy controller, and evaluated it on a variant of the traditional inverted pendulum problem (rotary inverted pendulum) and ball and beam system. Although focused on the evaluation of the proposed DE parameter adaptation strategy, the work also provided a clear evidence that DE is a method suitable for fuzzy controller optimization, outperforming other evolutionary approaches such as genetic algorithms.

Lai et al. [38] developed a fuzzy inference system for recognition of hypoglycaemic episodes. The fuzzy system employed Gaussian membership functions tuned using DE with a variation termed double wavelet mutation. In this algorithm, the value of the differential weighting constant is reduced with the increasing value of the iteration, according to a wavelet function. The second wavelet comes in during the crossover operation, where an additional term is used to further perturb the matrix parameters, again diminishing with increasing number of iterations. The proposed method compares favorably to other tested methods (linear regressions, evolved regressions and feed-forward neural networks), with higher values of sensitivity and specificity.

Pandit et al. [44] used an improved DE to solve the problem of environmental economic dispatch of power generation resources. The algorithm uses three different mutation strategies with a time-varying value of the differential weighting. Due to the multi-objective nature of the problem, fuzzy selection is used to rank and select the candidate solutions based on how well they meet both objectives. This method outperforms a weighted sum approach using classical differential evolution.

Chen and Yang [14] presented a method of optimizing a neural fuzzy inference system applied to a few simple applications. A cultural algorithm was used to provide a mechanism to provide feedback information to the entire population. Five different DE mutation operators are used, reflecting different methods of knowledge sources creating new potential solutions. The proposed optimization method significantly outperforms a number of other methods, including unmodified differential evolution variants, particle swarm optimization (PSO) and covariance matrix adaptation evolutionary strategies (CMA-ES).

Hung et al. [32] discussed a wavelet fuzzy neural network with asymmetric membership functions for control of an electric power steering system. Network learning rates are optimized using improved DE. The proposed method resulted in a controller with lower errors compared to a standard proportional-integral (PI) controller and to a fuzzy neural network controller using fixed learning rates.

Zheng et al. [70] presented a hybrid neuro-fuzzy system for classification of earthquake victims. A proposed differential biogeography-based optimization (DBBO) algorithm is used to train the neuro-fuzzy classifier. This method combines aspects of differential evolution into the migration equation of a biogeography-based optimization in order to exploit the solution-space exploration present in that method. The DBBO approach implemented outperforms methods including multispecies PSO and self-adaptive DE.

This contribution presents an application-driven study that employs DE as a simple, proven, and non-intrusive tool for optimization of fuzzy control systems. In particular, it considers the complex problem of predictive energy management for environmentally-powered wireless sensors. It develops a straightforward approach to controller representation and parameter optimization, and focuses on comprehensive evaluation of the proposed complex control strategy.

3. Problem model

This section presents the synthesis of the desired behavior of an energy manager for EPWSN, and describes the design of the corresponding fuzzy controller.

3.1. Energy management system

A typical EPWSN contains the following components [64]: a sensor or sensor array to capture the phenomena of interest; a microcontroller unit, to interface with sensors and manage data and energy within the system; a transceiver, to transmit collected and other data, and to receive data for time synchronization, scheduling and routing purposes; a data storage device to protect the collected data, and to improve node efficiency by taking advantage of varying energy costs of data storage and transmission; energy source(s), including primary batteries and a solar panel or other energy harvester; and energy storage element(s) such as rechargeable batteries or supercapacitors. The energy management system (EMS), a part of the microcontroller unit (MCU), is crucial for efficient operation of the EPWSN. It controls the conversion, storage, and use of energy within the node.

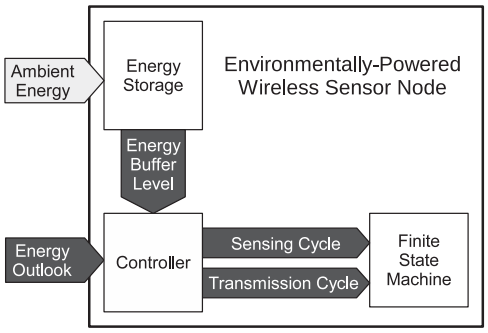


Fig. 1. A block diagram of the environmentally-powered wireless sensor node operation.

For the purpose of EMS design, many details can be abstracted away, as shown in Fig. 1. The MCU can be treated as a finite state machine (FSM) that facilitates transitions between different states of EPWSN operation. The EMS itself can be modeled as a controller that can observe internal and external states of the system and provides control signal(s) to the FSM. The observed states include *energy buffer level* (an internal state describing the amount of energy remaining in the energy storage, or its state of charge), and *energy outlook* (an external state describing the amount of ambient energy expected to be available for harvest from the environment). The control signals are *sensing cycle* (the time between two consecutive measurements), and *transmission cycle* (the time between two consecutive data transmissions).

The energy buffer level describes the relative state of charge of the energy storage device. In addition to its primary meaning, this variable also serves as a proxy to the energy harvested from the environment during the recent past, and thus to the ambient energy availability.

The energy outlook is based on predictions of energy availability over certain time horizon. The duration of the horizon depends on a number of factors, including the type and size of the energy harvester and energy storage elements, and on the patterns of ambient energy availability. As a result, determination of the duration is a non-trivial task. Its solution can be found, for example, using an optimization procedure described in Section 4.

The control signals adjust the sensor measurement and data transmission rates through the FSM. The general behavior of the controller can be described using several scenarios describing its desired response to specific values of the state variables. When the energy buffer is full and the outlook indicates high amount of available energy, the node should take measurements as often as desired and transmit collected data without delay. On the other hand, when the energy buffer level is low and the outlook is close to zero, the node should take measurements only at a minimal acceptable rate and data transmission should be delayed to conserve energy.

To design an efficient and robust energy management system, its objectives have to be formalized. This involves definition of an objective function, specification of domains of involved decision

variables, and identification of constraints. These elements are, in general, dependent on the type of sensor node and its application. For example, objectives of a sensor node for monitoring environmental conditions of an ecosystem could be formalized as follows:

- The *sensing rate* can be specified in terms of how far apart should individual measurements be spaced. This involves a *desired* sensing cycle, e.g. $T_s^{\text{des}} = 1$ min, and *maximum acceptable* sensing cycle, e.g. $T_s^{\max} = 60$ min. The *actual* sensing cycle then varies in range $T_s \in [1, \infty)$ min, because its values can extend beyond $T_s^{\max}$ in case of node failure. In this sense, $T_s^{\max}$ can be considered a soft constraint.
- To obtain monitoring data that are *as complete as possible*, a monitoring device should minimize the occurrence of cases when the actual sensing cycle is longer that the maximum acceptable cycle, i.e. $T_s > T_s^{\max}$. This requirement can be expressed as minimization of the total length of time when the measurements are spaced more than $T_s^{\max}$ apart.
- When the controller schedules a measurement, but it cannot be executed due to the lack of energy, a *monitoring failure* occurs. The number of failures, n_f, over a period of time should also be minimized.
- In addition to sensing, a wireless sensor node has to transmit the collected data to a base station or to other nodes. The *transmission rate* can be specified in terms of how far apart should individual transmissions be spaced. This involves a *desired* transmission cycle, e.g. $T_t^{\text{des}} = 2$ min, and *maximum acceptable* transmission cycle, e.g. $T_s^{\max} = 1440$ min (or 24 h). The *actual* transmission cycle then varies in range $T_s \in [2, \infty)$ min, because its actual values can extend beyond $T_s^{\max}$ in case of node failure. Similarly to the sensing rate, $T_t^{\max}$ is a soft constraint.

These objectives are used to design the structure and parameters of a controller implementing the energy management system. They also form the basis of an objective function used for evaluation and optimization of the controller.

3.2. Fuzzy controller

Fuzzy logic can be considered an abstract language suitable for efficient definition and synthesis of intelligent control systems [10]. Fuzzy logic controllers use fuzzy sets to describe control conditions and actions. The ensuing representation allows for easy, high-level definition of control laws in the form of IF-THEN rules. For example, a rule describing relation between variables of an energy management system can have the following form

$$\text{IF } E_b \text{ is VH THEN } T_s \text{ is VS}, \tag{1}$$

where VH (for *very high*) is a fuzzy set defined on the input variable E_b (level of energy buffer), and VS (for *very short*) is another fuzzy set defined on the output variable T_s (sensing cycle, i.e. the time between two consecutive measurements).

A basic rule-based controller can be designed manually by a domain expert. For the purpose of energy management, one can define the following linguistic variables and corresponding fuzzy sets. The first input variable is *normalized energy buffer level*, E_B. This domain is partitioned into five fuzzy sets (a.k.a. terms) shown in Fig. 2: VL (*very low*), L (*low*), M (*medium*), H (*high*), VH (*very high*).

The second input, E_n, is the *energy harvesting outlook* for n hours ahead, E_n^o, multiplied by the ratio of the predicted outlook and the maximum amount of energy available for harvest at given site over the same time period, $E_n^{\max}$,

$$E_n = E_n^o \frac{E_n^o}{E_n^{\max}(t)} = \frac{(E_n^o)^2}{E_n^{\max}(t)}. \tag{2}$$

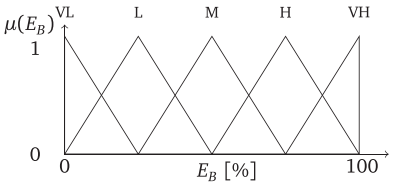


Fig. 2. Fuzzy partition of energy buffer level E_B.

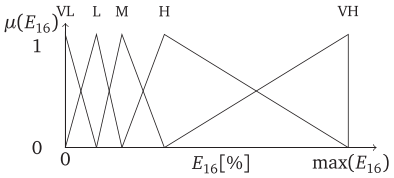


Fig. 3. Fuzzy partition of normalized energy harvesting outlook E_n (example for $n = 16$).

This scaling by $E_n^o/E_n^{\max}(t)$ takes into account diurnal and annual distribution of solar radiation, and ensures that the energy outlook is considered relative to the amount of energy expected over the horizon of n hours, starting at time t. The value of $E_n^{\max}(t)$ is constant for any particular location on Earth that can be calculated based on spatial coordinates and time of year [55].

The domain of E_n is also divided into five terms, with the same labels as in the previous case. However, their partition is derived from the distribution of actual energy availability for given time horizon, so that each term covers 20% of the values. Consequently, the terms are not distributed evenly, as shown in Fig. 3 for $n = 16$. The maximum amount of energy, $\max(E_n)$, differs for each horizon n and depends on the actual environmental conditions at the deployment site.

The fuzzy controller has two outputs: *sensing cycle* and *transmission cycle*. Sensing cycle, T_s, is the time between two consecutive measurements. Instead of general fuzzy sets, the output terms are fuzzy singletons (fuzzy sets whose support is a single point with grade of membership equal to 1) ranging from s_1 (short interval ≡ high sensing rate) to s_5 (long interval ≡ low sensing rate). To follow the example introduced in Section 3.1, the sensing cycle has been set between 1 and 60 min, corresponding to the range between 1 measurement per minute and 1 measurement per hour. Transmission cycle is the time between two wireless transmissions. The five fuzzy singletons range from t_1 (short interval ≡ high transmission rate) to t_5 (long interval ≡ low transmission rate). The transmission cycle is set between 2 and 1440 min, corresponding to a transmission taking place every 2 min to once a day. Relation between the input and output variables is described using a rule base consisting of rules in form (1), with consequents represented as fuzzy singletons of sensing/transmission cycles. The entire fuzzy rule base can be represented in a matrix form, as shown in Table 1.

Table 1
Fuzzy rule base of the energy management controller.

	E_B				
	VL	L	M	H	VH
E_n					
VL	s_5, t_5	s_5, t_5	s_5, t_5	s_4, t_4	s_3, t_3
L	s_5, t_5	s_5, t_5	s_4, t_4	s_3, t_3	s_2, t_2
M	s_5, t_5	s_4, t_4	s_3, t_3	s_2, t_2	s_1, t_1
H	s_4, t_4	s_3, t_3	s_2, t_2	s_1, t_1	s_1, t_1
VH	s_3, t_3	s_2, t_2	s_1, t_1	s_1, t_1	s_1, t_1

The rule base is evaluated by comparing the actual values of inputs against the input fuzzy sets

$$\mathrm{Poss}(E, k) = \sup_{E}(\min(\{E(t)\}, \mu_k(E))), \tag{3}$$

where Poss is a possibility measure implementing the comparison, E is a relative measure of energy (either E_B or E_n), $E(t)$ is a fuzzy singleton corresponding to the actual input value at time t, and $\mu_k(E)$ is the membership function of one of the fuzzy sets defined in the corresponding input space (i.e. $k = \{\mathrm{VL}, \mathrm{L}, \mathrm{M}, \mathrm{H}, \mathrm{VH}\}$).

Results of these individual evaluations are then combined using a **t**-norm operation, and the resulting value determines the level of activity of the corresponding rule

$$\lambda_r = \mathrm{Poss}(E_B, k) \quad \mathbf{t} \quad \mathrm{Poss}(E_n, l), \tag{4}$$

where r is a rule that relates fuzzy sets k and l, defined on input domains E_B and E_n, respectively. The **t**-norm operation is implemented using algebraic product $\mathbf{t}(a, b) = a \cdot b$.

The rule activities, λ_k, are finally used as coefficients that weight the contributions of the output fuzzy singletons towards the control value, i.e.

$$T_s(t) = \frac{\sum \lambda_r s_r}{\sum s_r}, \quad \text{and} \quad T_t(t) = \frac{\sum \lambda_r t_r}{\sum t_r}, \tag{5}$$

where s_r and t_r are the fuzzy singletons assigned to rule r.

The structure of the manually designed controller corresponds to the matrix representation from Table 1, with fuzzy singletons $s_5 = t_5 = 1$, $s_4 = t_4 = 0.75$, $s_3 = t_3 = 0.5$, $s_2 = t_2 = 0.25$, and $s_1 = t_1 = 0$. These value represent relative duration of the sensing cycle, T_s, and transmission cycle, T_t, within the ranges defined earlier. This controller is described by a control surface shown in Fig. 8(a).

4. Controller optimization

Although the manually designed (set) controller may operate well, its performance can be further improved by an appropriate modification of its parameters. However, fuzzy systems do not naturally support learning or adaptation. The lack of an effective learning mechanism can be overcome by combining fuzzy systems with evolutionary computing. This contribution uses an evolutionary approach for the optimization of the controller introduced in the previous section. During the optimization process, the expert-defined, domain-specific rule base is fixed and the fuzzy singletons s_1, t_1, …, s_5, t_5 are tuned by the evolutionary algorithm.

4.1. Optimization method

Based on the literature review, differential evolution was selected as the method of choice for tuning the controller parameters. In contrast to the traditional evolutionary methods such as genetic algorithms, DE was designed for solving continuous, real-valued problems. It has been successfully used for optimization of various fuzzy models including center-of-gravity, standard additive, and Takagi-Sugeno model [8,15,60]. It was shown that the DE is in some fuzzy control-related applications superior to other evolutionary optimization methods including traditional and elitist genetic algorithms [8]. This choice has also been confirmed through experimental evaluation of the ability of several widely-used meta-heuristic optimization methods to evolve fuzzy controllers for EPWSN. DE gained popularity as a highly efficient method for both, real-parameter [51] and discrete-parameter [54] optimization. Since its inception, it has been used to solve many practical problems [19].

DE evolves a population of candidate solutions by their iterative modification through differential mutation and crossover. In each iteration, mutation is applied to the current population to form so called trial vectors. These vectors are further modified by various crossover operators. At the end of each iteration, the trial vectors compete with existing candidate solutions for survival in the population. A high-level outline of the DE algorithm is summarized in Algorithm 1. A detailed description of different DE variants and particular mutation and crossover strategies can be found, e.g., in [25,51].

Algorithm 1. A summary of classic differential evolution

```
1  Initialize a population, P^0, consisting of M vectors;
2  Evaluate an objective function, f_obj, ranking the vectors in the population;
3  while Termination criteria is not satisfied do
4      for i ∈ {1, ..., M} do
5          Differential mutation: Create trial vector v_i^t according to some mutation strategy;
6          Validate the range of coordinates of v_i^t; optionally adjust coordinates of v_i^t so, that v_i^t is valid solution to given problem;
7          Perform crossover; select randomly one parameter l in v_i^t and modify the trial vector using some crossover strategy;
8          Evaluate the trial vector.;
9          if trial vector v_i^t represents a better solution than target vector x_i then
10             add v_i^t to P^{t+1}
11         else
12             add x_i to P^{t+1}
13         end
14     end
15 end
```

Although DE performed well in relevant prior studies [8,15,60], the efficiency of meta-heuristic optimization is problem dependent [65] and the problem domain and controller representation in this work are different from the previous applications. To alleviate this problem and to provide comparison with other existing approaches, three other common soft computing methods have been used to search for EPWSN controllers: genetic algorithms [1], particle swarm optimization [34], and covariance matrix adaptation evolution strategy [30,33].

The first method, Genetic Algorithm (GA), is the paramount evolutionary optimization method [25,41]. It mimicks genetic evolution by iterative application of operators simulating natural selection and biological reproduction on encoded candidate solutions. The GA has been used to tune [31] and synthesize [11] fuzzy controllers for nearly two decades.

The second approach, PSO, is a global, population-based algorithm for continuous optimization based on emulations of swarming behaviour of bird flocks, fish schools and even human social groups [34,25]. It uses a population of mobile candidate particles characterized by their position, x_i, and velocity, v_i, inside an n-dimensional search space they collectively explore. The method

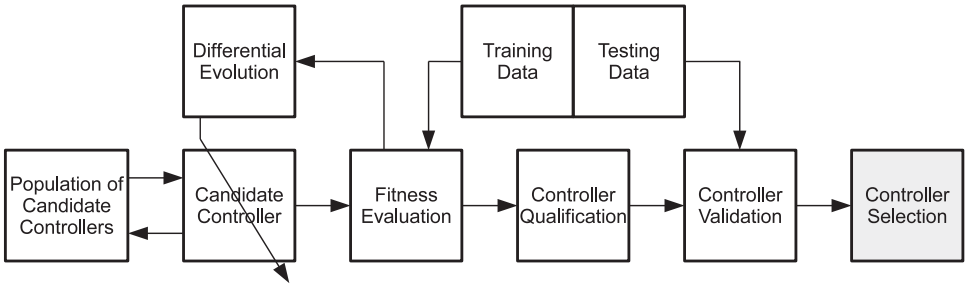


Fig. 4. The process of controller development using differential evolution.

has a number of variants and numerous successful industrial applications [40], including optimization and tuning of different types of fuzzy controllers [13,42].

Finally, Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [30,33] is a recent variant of evolution strategies with derandomized self-adaptation (i.e. strategy parameter control). CMA-ES builds and maintains a covariance matrix of mutation distribution so that the probability of application of successful mutation steps is increased during the evolution. It uses a principal component analysis (PCA) of past mutation steps to derive the new mutation distribution. The algorithm was developed for real-world search problems with significantly non-separable objective functions [30]. In the past, it was successfully used for the optimization of, e.g., closed loop [57] and multivariable fractional order [61] fuzzy controllers.

4.2. Optimization process

For the purpose of DE, the controller is encoded as a vector of 10 floating-point numbers $c_{\{s_i,t_i\}} \in [0, 1], i \in \{1, \ldots, 5\}$

$$\mathbf{c} = (c_{s_1}, c_{s_2}, c_{s_3}, c_{s_4}, c_{s_5}, c_{t_1}, c_{t_2}, c_{t_3}, c_{t_4}, c_{t_5}). \quad (6)$$

The actual output fuzzy singletons are decoded from this representation as follows

$$s_j = \begin{cases} c_{s_j}, & j = 1 \\ c_{s_j}\left(1 - \sum_{l=1}^{j-1} s_l\right), & j > 1, \end{cases} \quad (7)$$

and analogously for t_j. This decoding ensures continuity of the controller [45] at no extra cost, and allows the use of traditional DE operators.

The process of controller optimization is outlined in Fig. 4. First, an initial population of controllers is formed by randomly perturbing the output fuzzy singletons of the set controller (i.e. by multiplying the manually encoded values of s_j and t_j by *rand*(0.9, 1.1)). The controller was optimized using the */DE/rand/*1 variant of the algorithm. The parameters of the algorithm were set according to the best practices, past experience, and initial trials: scaling factor $F = 0.95$, mutation rate $C = 0.9$, and population size $M = 10$. The rather small population size was used due to the low dimensionality of the problem and relatively high computational costs of the simulations. It has been shown, however, that many practical low-dimensional problems can be successfully solved by DE with small population sizes [52].

The remaining methods used to evolve PRWSN controllers were configured as follows, based on the past experience and initial optimization trials. The *steady-state* GA with generation gap 2 (parents were instantly replaced by superior offspring chromosomes) used one-point crossover with probability $p_c = 0.8$, and uniform mutation with probability $p_m = 0.02$. Parent selection was a combination of roulette-wheel and elitist and population size, M, was set to 10 as in DE. PSO was the traditional *global* variant (*gbest* PSO) with local and global weights equal to 1.49445 and inertia weight 0.729. The number of particles was set to 10. For CMA-ES, a quasi parameter-free algorithm, the randomly perturbed set controller was used as the initial solution and the initial standard deviation of all tuned parameters was set to 0.05. CMA-ES population size, λ, was set to 10. Unlike the other algorithms, CMA-ES did not require any other explicit parameters. All algorithms were limited by the same maximum number of fitness function evaluations (1000) and all optimization runs (DE, GA, PSO, and CMA-ES) were independently executed 50 times to enable a fair comparison. Parameters of all compared algorithms are summarized in Table 2.

Table 2
Parameters of all compared algorithms.

Algorithm	Parameters
*/DE/rand/*1	population size $M = 10$, scaling factor $F = 0.95$, mutation rate $C = 0.9$
steady-state GA	population size $M = 10$, generation gap 2, one-point crossover operator, probability $p_c = 0.8$, uniform mutation, probability $p_m = 0.02$
gbest PSO	population size $M = 10$, inertia weight $c_0 = 0.729$, local (cognitive) weight $c_1 = 1.49445$, global (social) weight $c_2 = 1.49445$
CMA-ES	population size $\lambda = 10$, initial standard deviation $\sigma_i = 0.5$ for $i \in \{1, \ldots, 10\}$

4.3. Fitness function

Each candidate controller, c, is evaluated using a fitness function, f_{obj}, and then submitted to the DE process described by Algorithm 1. The fitness function developed in this contribution is based on the design objectives for EPWSN energy management systems introduced in Section 3.1. It is based on experience from the field deployment of a recently constructed prototype EPWSN [50]. However, it can be used for a broad class of problems in the domain of environmentally powered devices with energy harvesting and adaptive duty cycling.

The fitness is evaluated based on the performance of the candidate controllers in simulations using real data collected at the location of planned deployment of the EPWSN. In general, this data should cover one or multiple complete years, to account for the variability of energy availability through seasons. The fitness function to be minimized takes into account four aspects of the system performance

$$f_{\text{obj}}(c) = w_1 \cdot f_1 + w_2 \cdot f_2 + w_3 \cdot f_3 + w_4 \cdot f_4. \quad (8)$$

Its components, f_i, correspond to the previously introduced objectives, while the weights, w_i, adjust the relative importance of each objective in the context of the problem [6]. Such combination of several disparate system characteristics (e.g. percentage, absolute number and time, as in our case) into a single scalar fitness function is quite common [47] and well founded in multiattribute

utility theory [21]. The first component, f_1, expresses the average actual sensing cycle relative to the maximum acceptable sensing cycle, $T_s^{\max}$. To account for variability of the actual cycle, the number of measurements over a period of time is used instead

$$f_1 = \frac{n_s^{\min}}{n_s}, \tag{9}$$

where $n_s^{\min}$ (derived from $T_s^{\max}$) is the minimum acceptable number of measurements collected over the period, and n_s is the actual number of collected measurements. Although T_s cannot be set to a value longer than $T_s^{\max}$ by the controller, the actual sensing cycle can be longer due to node failures caused by the unavailability of energy and thus inability to perform a scheduled measurement. The ratio $n_s^{\min}/n_s$ takes value 1 when the actual number of measurements equals $n_s^{\min}$, and grows/decreases, respectively, with their decreases/increases.

The second term, f_2, considers the total length of time when the measurements are spaced more than $T_s^{\max}$ apart

$$f_2 = \sum_{T_s > T_s^{\max}} T_s. \tag{10}$$

This way, the failures that yield actual sensing cycle T_s longer than $T_s^{\max}$ are penalized. Such long sensing cycles can be caused by one failure during periods of low sensing rates, or by a series of consecutive failures during periods of higher sensing rates.

The third component, f_3, represents the total number of failures, n_f, over a period of time

$$f_3 = n_f. \tag{11}$$

The last term, f_4, penalizes actual transmission cycles, T_t, longer than the maximum acceptable transmission cycle, $T_t^{\max}$

$$f_4 = \sum_{T_t > T_t^{\max}} T_t. \tag{12}$$

The values of weights in (8) have been determined experimentally during initial trials. They adjust the relative values of individual fitness component to the order of single digit for a typical node performance ($w_1 = 100$, $w_2 = w_3 = 0.01$), and adjust the importance of some components within the overall fitness value ($w_4 = 0.0004$, representing the relatively low priority of data transmission compared to data collection). In practice, the values of these constants can be adjusted to represent the importance of individual fitness components based on application requirements or designer preferences.

5. Experiments and results

The EPWSN controllers were evolved and examined through a series of simulation experiments described in this section.

5.1. Experiment design and data description

The computational experiments involved simulations performed using an enhanced version of a recently developed software simulator of the wireless sensor node [36,37]. The simulator was extended for this study to implement a model of hardware developed during laboratory and field characterization of the prototype EPWSN described by [50]. The model provides accurate description of energy conversion, storage and consumption within the device. It considers various measured characteristics, such as the amount of energy available for harvest, the capacity and leakage of the energy storage elements, and the amount of energy required by the node in different operating states. The simulator has also been modified so it can process and use solar irradiation forecast data for the purpose of modelling predictive energy management methods proposed in this contribution.

The operations considered by the simulator include analog and digital sensor readings (performed together at every sampling cycle), transfer of data between internal memories (there are a small, energy efficient data buffer, and a larger, but more power-demanding permanent data storage in a non-volatile memory device), and wireless data transmission between the node and a base station. At each simulation step (corresponding to device wake-up), the simulator records a number of node attributes that are used for evaluating fitness function and for analyzing node performance.

The evolutionary process, driven by the fitness function described above, is stopped after 100 generations. The fitness of the best performing individual in each experimental run is retained, and the average fitness of the best performers over 50 runs is calculated. This is repeated for each controller configuration and each optimization method applied. For each method, the average fitness values are used to determine the configuration that is best qualified to manage energy consumption of EPWSN.

The controller simulator requires environmental data to determine the conditions the platform will be exposed to when deployed. The experiments use five years of data from the ACIS "Fairview AGDM" site located at 56.0815° latitude, −118.4395° longitude, and 655.00 m elevation. This site was selected due to its proximity to the EMEND Project in the north-west part of Alberta, Canada, concerned with forest ecosystem monitoring [24]. The data set, obtained from the ACIS service [3], has 5 min solar irradiance measurements (in W/m^2) continuous from January 1, 2009 until December 31, 2013. To facilitate testing of the developed controllers, the entire data set was divided to two parts: 40% (2 years) for training, and 60% (3 years) for validation.

In addition, to allow validation of the controllers using real forecasts, solar irradiance predictions for the location of interest were generated using the numerical weather prediction model WRF [58] from the Global Forecasting System (GFS) grid 003, for the year of 2012. The outputs for the variable of interest (downward solar radiation) were recorded for every hour, and the values at the spatial point corresponding to "Fairview AGDM" site were extracted.

5.2. Optimal length of the prediction horizon

Simulations were run for a number of controller configurations: the 'simple' controller considered only energy buffer level, while controllers 'E_n' were harvesting-aware, considering energy outlook for $n \in \{2, 3, 4, 5, 6, 8, 16, 24, 32\}$ hours ahead. The results of the optimization, in terms of average number of annual measurements and error rate of evolved controllers, are shown in Figs. 5 and 6, respectively. The plots indicate that the DE is in most configurations the best optimization strategy combining the highest average number of measurements and the lowest error rate of evolved controllers. Controller E_3, with 3 h energy availability outlook, appears to be the most successful energy availability-aware control

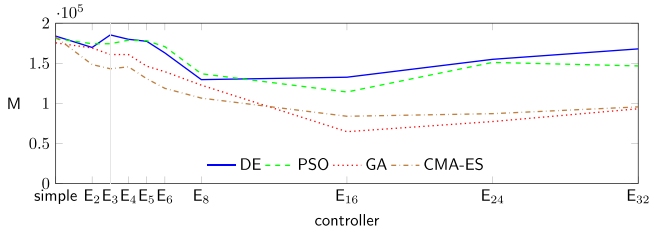


Fig. 5. Comparison of the average number of measurements collected per year (training data set).

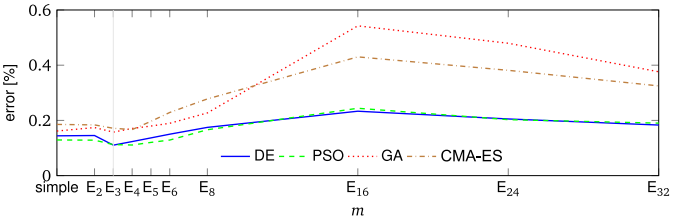


Fig. 6. Comparison of error rate of every energy management controller (training data set).

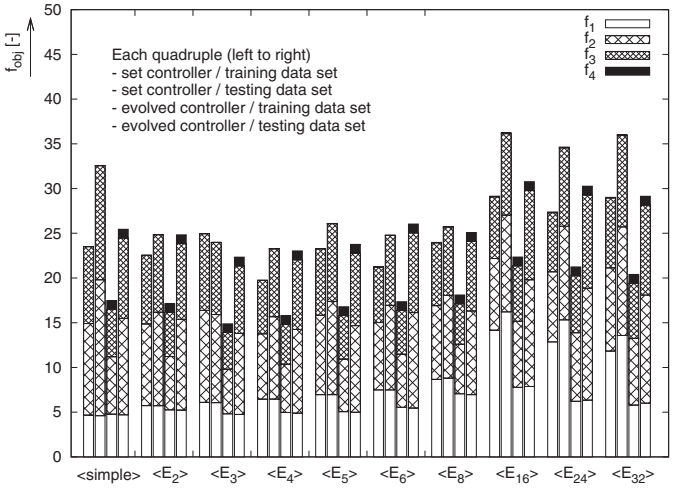


Fig. 7. Summary of average fitness for all considered configurations of energy management controller.

strategy with the maximum amount of average annual measurements and minimum average error rate.

5.3. Controller validation

To allow validation and comprehensive comparison of different types of evolved controllers, each obtained control strategy was applied to training (labeled 'trn') and testing (labeled 'tst') data sets. The two data sets cover the periods of 2 and 3 years, respectively, as described in Section 5.1. Simulations were also run separately for the manually set (labeled 'set') and evolved (labeled 'evo') controllers. Results of all simulation experiments are shown in Fig. 7. The height of each bar represents the fitness of corresponding controller/data set combination, while the segments of each bar correspond to individual fitness components $f_1 - f_4$.

Table 3 details the results illustrated in Fig. 7. In addition to the fitness and its components, the table lists the average amount of energy harvested per year, E_H, the average number of measurements collected per year, M, the relative percentage of failures, Fail, and the average durations of the sensing cycle, $\bar{T}_s$, and transmission cycle, $\bar{T}_t$. As mentioned earlier, the optimal prediction horizon has been determined to be $n = 3$ h. The corresponding group of experiments attained the lowest fitness on both the training and testing data sets (cf. rows 'evo trn' and 'evo tst' under group E_3 in Section 3). Accordingly, this controller has also yielded a combination of the maximum number of annual measurements and minimum percentage of failures. This confirms the conclusions obtained by a visual inspection of the average annual number of measurements (maximum at E_3) and error rate (minimum at E_3) of all examined controllers shown in Figs. 5 and 6, respectively.

5.4. Analysis of the best performing controller

From the 50 independent optimization runs, controller labeled 'E_3^{best}' achieved the overall lowest values of fitness $F_{\text{obj}} = 11.69$. The control surfaces for the best-performing controller are shown in Fig. 8(b) and (c). Both sensing and transmission control surfaces have similar strategies for VH and VL values of E_B and E_n. For the intermediate values (H, M and L), the controller prefers sensing before transmitting the data.

Detailed analysis of the winning controller performance is shown in Figs. 9 and 10. Fig. 9 shows a histogram of intervals between measurements for the winning controller. The controller adjusts the sensing cycle to 1 min when the storage is fully charged and ample ambient energy is available, and between 1 and 2 min most of the time. During the periods of low energy availability (e.g. during long nights), the controller sets the sensing cycle to longer periods. Only at 147 (or 0.03%) occasions is the actual cycle longer than the maximum acceptable sensing cycle $T_s^{\max} = 60$ min.

The transmission cycle histogram is shown in Fig. 10. The cycle is set to the minimum length of 2 min most of the time, with longer periods engaged in situations analogous to the sensing cycle. Only at 4 (or 0.0016%) occasions is the transmission cycle longer than the maximum acceptable transmission cycle $T_t^{\max} = 24$ h.

Behavior of the winning controller through a complete year of operation is illustrated in Fig. 11. The top graph shows the time series of daily number of measurements (solid line) and distribution of the sensing cycles (gray dots corresponding to individual values of sensing cycle for each day). Similarly, the second graph shows the daily number of transmissions and individual values of transmission cycle. In both cases, the distribution scatter plots correspond to the histograms shown in Figs. 9 and 10, but unfolded in time. The third and fourth plots show the daily average values of the energy and data buffers, respectively.

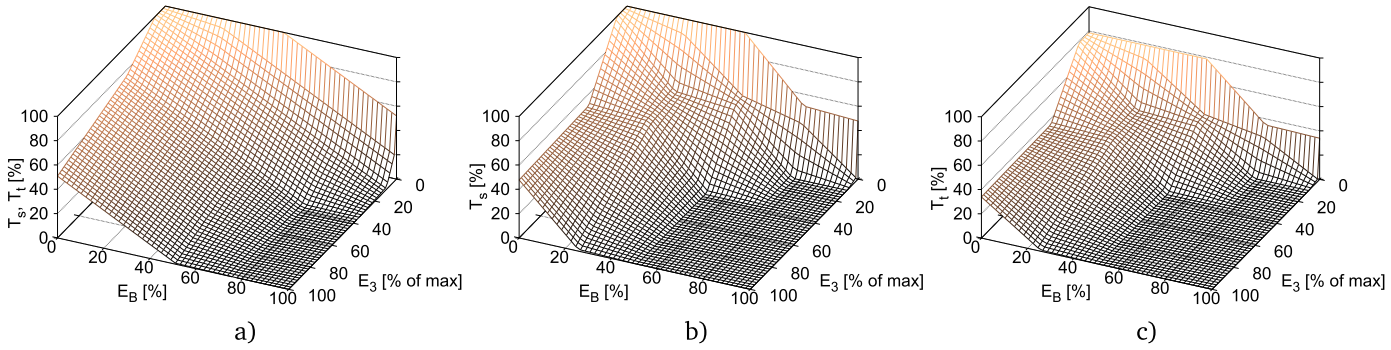


Fig. 8. Controller surfaces: (a) sensing and transmission surfaces of the basic controller (before optimization), (b) sensing surface after optimization (controller E_3^{best}), (c) transmission surface after optimization (controller E_3^{best}).

Table 3
Simulation results for different controller configurations (simple and predictive), prediction horizons of energy outlook (values of n in E_n), and data sets (training and testing). Reported values are averages over 50 simulation runs.

Group	Data	Fitness	f_1	f_2	f_3	f_4	E_H [kJ]	M [–]	Fail [%]	$\bar{T}_s$ [s]	$\bar{T}_t$ [s]
	set trn	23.51	4.68	10.23	8.60	0.00	179.6	187,285	0.23%	168	378
	set tst	32.57	4.61	15.20	12.76	0.00	171.7	178,971	0.24%	166	373
Simple	evo trn	16.58	4.78	6.42	5.31	0.97	178.5	183,865	0.14%	172	374
	evo tst	24.47	4.71	10.77	8.99	0.98	170.7	175,818	0.17%	169	368
	set trn	22.55	5.73	9.14	7.68	0.00	159.6	152,811	0.25%	206	454
	set tst	24.84	5.72	10.46	8.66	0.01	151.1	144,278	0.20%	205	454
E_2	evo trn	16.19	5.26	5.96	4.94	0.97	166.7	169,709	0.15%	189	437
	evo tst	23.85	5.23	10.15	8.46	0.98	158.1	160,922	0.17%	188	436
	set trn	24.95	6.09	10.28	8.57	0.00	153.2	143,733	0.30%	219	488
	set tst	23.98	6.06	9.85	8.06	0.01	145.2	135,999	0.20%	218	487
E_3	evo trn	14.11	4.81	5.00	4.10	0.97	169.0	185,345	0.11%	173	455
	evo tst	21.35	4.75	9.05	7.54	0.99	160.9	176,876	0.14%	171	453
	set trn	19.74	6.46	7.30	5.98	0.01	147.4	135,56	0.22%	232	523
	set tst	23.27	6.47	9.20	7.59	0.01	139.1	127,42	0.20%	233	526
E_4	evo trn	14.89	4.97	5.41	4.44	0.98	165.8	179,859	0.12%	178	474
	evo tst	22.04	4.90	9.34	7.79	0.99	157.9	171,638	0.15%	176	471
	set trn	23.26	6.96	8.87	7.43	0.02	140.6	125,828	0.29%	250	571
	set tst	26.07	6.95	10.43	8.69	0.03	132.9	118,596	0.24%	250	572
E_5	evo trn	16.01	5.06	5.89	4.85	0.98	162.8	177,381	0.14%	182	499
	evo tst	22.76	4.99	9.68	8.08	1.00	155.4	169,418	0.16%	179	493
	set trn	21.23	7.49	7.53	6.21	0.02	134.4	116,976	0.26%	269	620
	set tst	24.76	7.49	9.46	7.82	0.03	127.2	110,179	0.24%	269	622
E_6	evo trn	16.48	5.54	5.94	4.89	0.98	155.5	162,845	0.15%	199	533
	evo tst	25.05	5.46	10.67	8.91	0.99	148.5	155,611	0.19%	196	526
	set trn	23.93	8.67	8.24	7.02	0.04	123.3	101,076	0.35%	312	735
	set tst	25.70	8.82	9.25	7.64	0.05	115.4	93,556	0.27%	317	757
E_8	evo trn	17.23	7.05	5.53	4.54	0.98	140.7	129,569	0.17%	254	599
	evo tst	24.10	6.95	9.36	7.79	0.99	134.5	124,169	0.21%	250	593
	set trn	29.08	14.17	8.01	6.90	0.08	93.6	61,841	0.55%	510	1466
	set tst	36.14	16.21	10.82	9.11	0.10	84.1	50,871	0.59%	584	1755
E_{16}	evo trn	21.50	7.78	7.38	6.20	0.99	117.1	132,581	0.23%	280	1232
	evo tst	29.77	7.87	11.93	9.97	0.99	110.1	125,265	0.26%	283	1324
	set trn	27.31	12.85	7.85	6.61	0.07	96.5	68,167	0.48%	463	1350
	set tst	34.54	15.32	10.47	8.75	0.09	84.7	53,836	0.54%	552	1732
E_{24}	evo trn	20.33	6.22	7.66	6.37	0.98	127.2	154,885	0.21%	224	1059
	evo tst	29.27	6.35	12.51	10.41	1.00	118.1	145,819	0.24%	228	1238
	set trn	28.95	11.84	9.30	7.81	0.07	101.7	74,004	0.52%	426	1167
	set tst	35.94	13.58	12.15	10.21	0.10	90.5	60,738	0.56%	489	1400
E_{32}	evo trn	19.73	5.80	7.45	6.16	0.98	136.3	167,912	0.18%	209	931
	evo tst	28.15	6.01	12.08	10.05	0.99	126.1	157,564	0.21%	216	1089

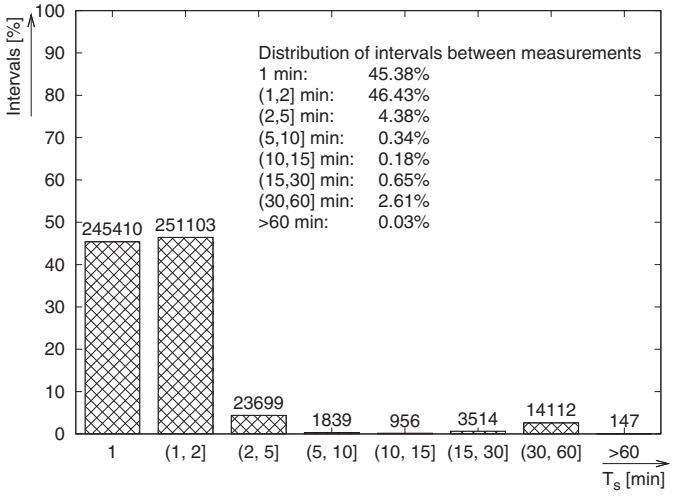


Fig. 9. Histogram of measurement periods for the best evolved controller E_3^{best}.

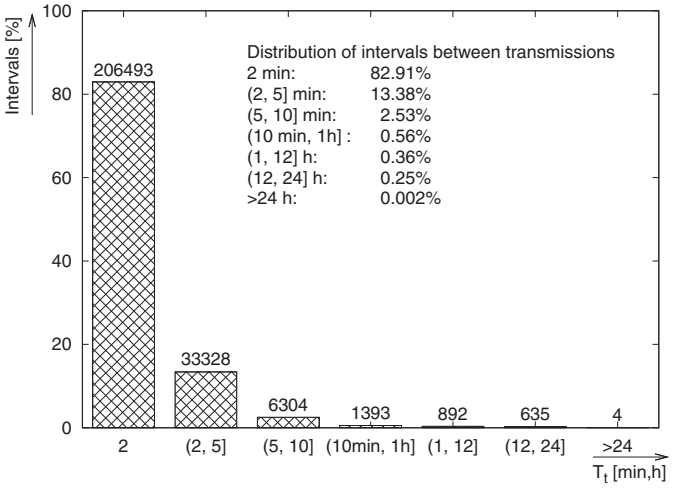


Fig. 10. Histogram of transmission periods for the best evolved controller E_3^{best}.

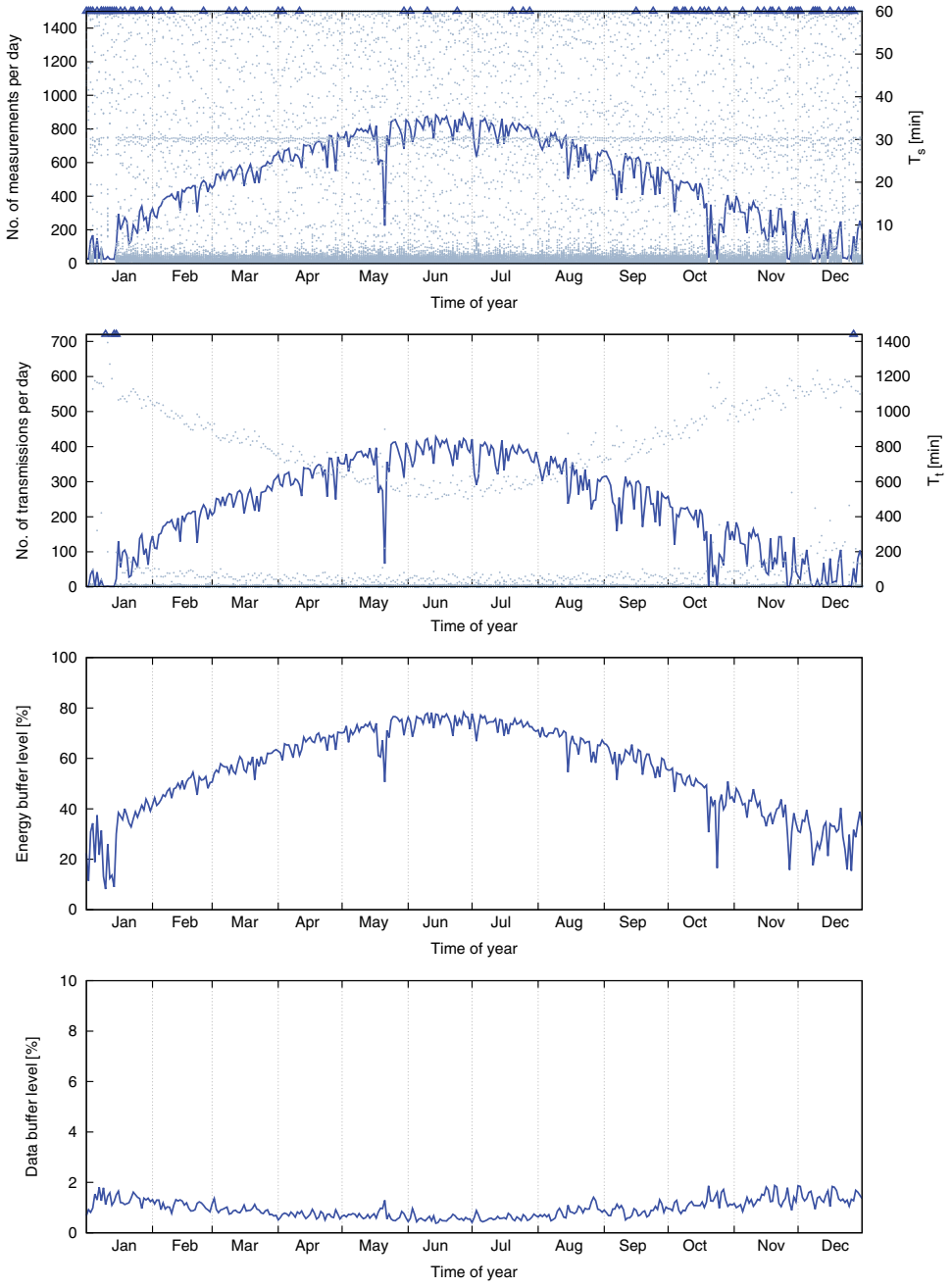


Fig. 11. Performance of the winning controller through different seasons (top to bottom: graph of the daily number of measurements and scatter plot of sensing cycle; graph of the daily number of transmissions and scatter plot of transmission cycle; daily average of energy buffer level; daily average of data buffer level). The triangle (▲) symbols in scatter plots indicate outliers, i.e. the values of corresponding cycles that are greater than the maximum acceptable duration.

All four plots clearly express the following seasonal trends. During summer (corresponding to the middle sections of the plots), when ambient energy is abundant, the daily numbers of measurements and transmissions are relatively high (and, conversely, the durations of sensing and transmission cycles are short). The increased activity of the node is supported by the higher average level of the energy buffer, and excess energy is also used to perform more frequent data transmissions (as shown by the relatively lower levels of data buffer). In winter (corresponding to the beginning and end parts of the plots), the trends are reversed: sensing and transmission activities are suppressed due to the lower availability of ambient energy reflected by the lower average levels of the energy buffer. The data buffer is relatively full, because of the less frequent transmissions to conserve stored energy.

5.4.1. Comparison with baseline controllers

To put the performance of the optimized controller to a perspective, it has been compared with two static control strategies. Both static controllers have fixed sensing and transmission cycles representing low- and high-intensity monitoring: $S^{60/1440}$ (one measurement every hour and data transmission once a day), and $S^{1/2}$ (one measurement a minute and data transmission every 2 min).

These controllers have been compared with E_3^{best} in simulations involving all three data sets employed earlier in this work to evolve and test the new controller. The simulations using *trn* and *tst* spanned across two (2008–2009) and three years (2010–2012), respectively, while the *real* data set covered one year (2012). The results of this comparison, on a per-year basis, are shown in Table 5.

Table 4
Performance of the best controller evolved by the DE, E_3^{best}, on training (evo trn) and testing (evo tst) data, and on actual forecasts of solar energy availability (evo real).

Algorithm	Data	Fitness	f_1	f_2	f_3	f_4	E_H [kJ]	M [–]	Fail [%]	$\bar{T}_s$ [s]	$\bar{T}_t$ [s]
DE	evo trn	11.69	4.64	3.90	3.14	0.004	184.61	188,800	0.083%	167	360
	evo tst	18.42	4.58	7.57	6.27	0.002	176.30	180,265	0.116%	164	354
	evo real	12.30	4.71	4.27	3.31	0.001	183.70	185,837	0.178%	169	363

Table 5
Performance of static controllers and E_3^{best} on all data sets.

Controller	Data	Fitness	f_1	f_2	f_3	f_4	E_H [kJ]	M [–]	Fail [%]	$\bar{T}_s$ [s]	$\bar{T}_t$ [s]
E_3^{best}	*trn*	11.69	4.64	3.90	3.14	0.004	184.61	188,800	0.083%	167	360
$S^{1/2}$		5629.89	3.35	90.32	5536.22	0.000	189.23	261,606	51.412%	120	375
$S^{60/1440}$		122.49	104.66	9.71	8.12	0.279	60.67	837	4.626%	3771	90559
E_3^{best}	*tst*	18.42	4.58	7.57	6.27	0.002	176.30	180,265	0.116%	164	354
$S^{1/2}$		7837.06	3.30	125.46	7708.31	0.000	181.02	250,272	50.658%	120	369
$S^{60/1440}$		127.56	104.24	12.76	10.56	0.393	57.92	7912	0.044%	3755	90612
E_3^{best}	*real*	12.30	4.71	4.27	3.31	0.001	183.70	185,837	0.178%	169	363
$S^{1/2}$		2837.95	3.38	45.57	2789.00	0.000	187.26	259,193	51.831%	121	380
$S^{60/1440}$		112.92	104.50	4.67	3.75	0.140	60.63	8383	4.282%	3759	90492

The high values of f_2 and f_3 for $S^{1/2}$ correspond, respectively, to the contiguous nature of controller failures and their very high number. This shows that there is not enough ambient energy to support high-intensity sensing with short, fixed measurement and transmission cycles. The second static controller, $S^{60/1440}$, collects only a small number of all possible measurements (cf. the high value of f_1), yet still does not guarantee error-free operation of the monitoring system.

The average weekly numbers of monitoring failures (i.e. measurement periods longer than 60 min) over one year are illustrated in Fig. 12, for all three data sets. The graphs show that the adaptive, harvesting-aware controller E_3^{best} operates with very few monitoring failures per week throughout the year. The static controllers have, on average, 3–4 times more failures. In the simulation using real weather forecasts, E_3^{best} worked 12 weeks without a single failure. In contrast, both static control strategies encountered 4-11 monitoring failures every week. The fast controller $S^{1/2}$ had 7 weekly failures throughout the year, with the exception of the second week with 5 failures and 24th week with 9 failures. This corresponds, on average, to one failure every day. The slow static controller $S^{60/1440}$ performed even worse, despite its significantly longer sensing cycle.

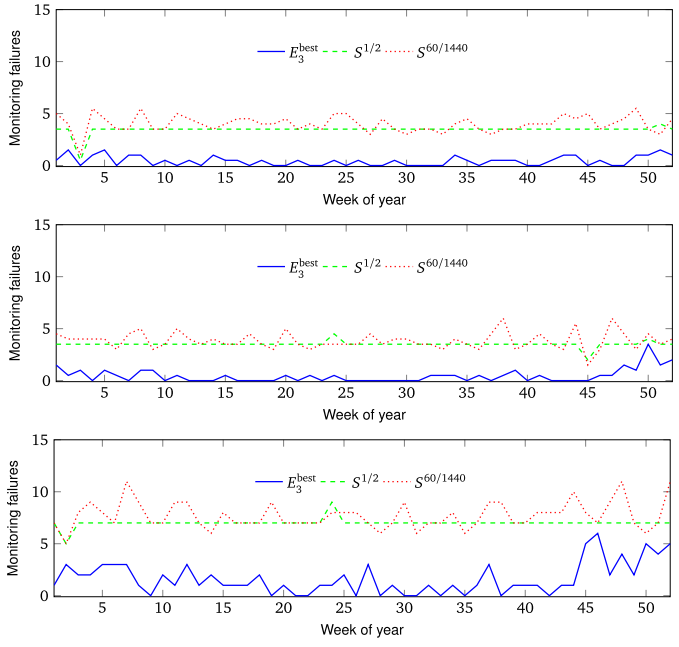


Fig. 12. Average number of monitoring failures per week of year (for three different data sets).

Table 6
Longest measurement and transmission cycles (in minutes)/hours without measurement and transmissions for all three controllers.

Measurements			Transmissions		
$S^{1/2}$	$S^{60/1440}$	E_3^{best}	$S^{1/2}$	$S^{60/1440}$	E_3^{best}
1404/23	780/13	780/13	1405/0	2160/1	1644/1
1235/20	780/13	780/13	1235/0	1980/1	1503/1
1205/20	720/12	780/13	1205/0	1860/1	1448/1
1199/20	720/12	780/13	1200/0	1740/1	1447/1
1185/20	720/12	780/13	1185/0	1560/1	1430/0

The summary of the five longest periods between measurements and transmissions for each controller, under the realistic simulation scenario, *real*, is provided in Table 6. The table illustrates that the adaptive, harvesting-aware sensing and transmission strategy of E_3^{best} reaches similar number of hours without measurement as the slow static controller $S^{60/1440}$, but collects more than 22 times more measurements over the course of the year. The fast controller $S^{1/2}$ collects approximately 1.4 times more measurements than E_3^{best}, over the same time period. However, the measurements scheduled and executed by E_3^{best} are more evenly distributed over time, as shown by the significantly lower number of hours without measurements. The evolved controller thus provides much more consistent coverage of the monitored phenomena.

From the perspective of data availability, E_3^{best} experienced a total of one day without transmission. This is worse than $S^{1/2}$ that transmitted data every day, but better than $S^{60/1440}$ that failed to send data in the required daily intervals multiple times.

To evaluate the inconsistency caused by missing measurements, we introduce an *inconsistency score*

$$IS(m) = \left(1 - \frac{M}{M_{\max}}\right) \cdot \left(\frac{2(m/m_{\max})}{1 + (m/m_{\max})}\right), \tag{13}$$

where $M_{\max}$ is the number of measurements that could be collected over the considered time period (here, one year) with the shortest possible sensing cycle of 1 min, M is the actual number of measurements collected by a controller over the same period (year), $m_{\max}$ is the maximum number of missing measurements for given monitoring failure event across all compared controllers, and m is the actual number of missing measurements for given event and

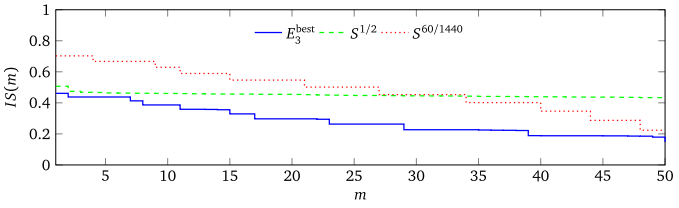


Fig. 13. Inconsistency score of 50 longest periods between measurements of E_3^{best}, $S^{1/2}$, and $S^{60/1440}$ (lower is better).

particular controller. The first term reflects the proportion of the maximum possible number of measurements actually collected by a controller. It evaluates to 0 when all possible measurements are collected, and to 1 when all measurements failed. The second term describes the severity of a monitoring failure for particular controller using the relative number of missing measurements with respect to all compared controllers. This term evaluates to 0 when there are no measurements missing (i.e. when no failure occurred), and to 1 if no measurements are collected during the failure period.

The higher $IS(m)$, the greater the inconsistency in terms of time coverage of measurements, weighted by the success in collecting their maximum desirable number. A visual comparison of 50 longest intervals between measurements in terms of $IS(m)$, is shown in Fig. 13. It shows that for the monitoring periods with the longest failures, E_3^{best} substantially reduces inconsistency of collected data, compared to both $S^{1/2}$ and $S^{60/1440}$.

The analysis of the longest measurement and transmission periods shows that adaptive, harvesting-aware sensor node control allows consistent, predictable power management that yields small number of failures with a well spread temporal distribution. In contrast to static control strategies, the harvesting-aware approach substantially reduces the number of monitoring failures during which the phenomena of interest is not sensed for more than 60 min. This enables the use of complex dynamic sensing and monitoring strategies powered by ambient energy that can respond to various application requirements.

5.4.2. Verification using real forecasts

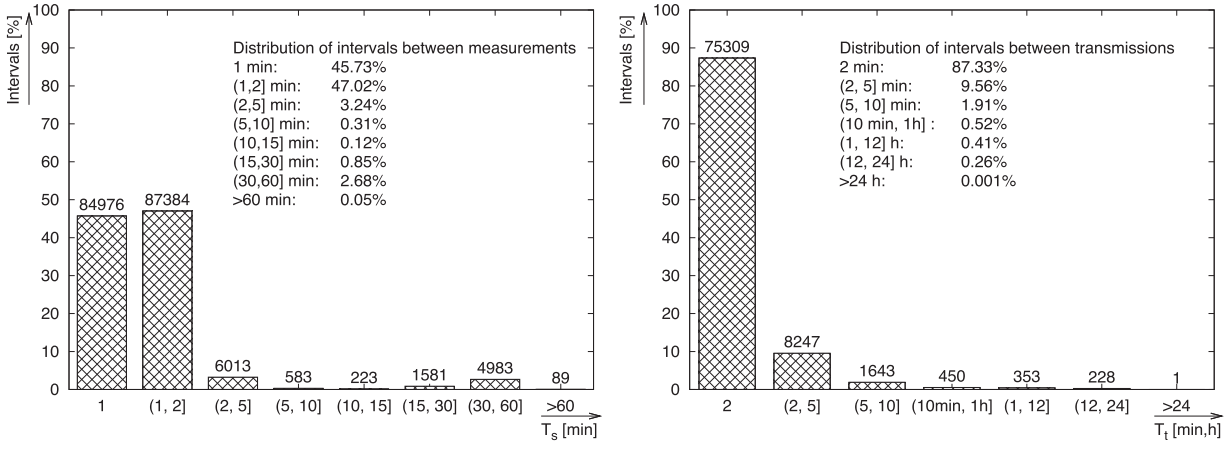
To verify the performance of the selected controller under realistic conditions, an additional simulation has been executed using solar irradiation values obtained from the WRF numerical weather prediction model, rather than from historical measurements used for training and testing. This corresponds to a real deployment scenario, when irradiation forecasts could be generated off-site and provided to the nodes from a central location (e.g. from a data aggregator).

The results of this experiment, along with detailed performance indicators of the winning controller on the training and testing data sets, are shown in Table 4. The overall fitness of the controller using real forecasts is only slightly higher than the fitness obtained during training. Values of all other indicators (including the amount of harvested energy, number of measurements, and average sensing and transmission cycles) are also comparable to the training performance.

Histograms of measurement and transmission periods from this experiment are shown in Fig. 14. It can be seen that the distribution of intervals is similar to those obtained in previous experiments using historical irradiation measurements, corresponding to a perfect forecast. For example the number of measurement intervals longer than 60 min slightly increased from 0.03% to 0.05%. In other words, any errors of numerical weather prediction for the test period (full year of 2012) did not significantly compromise the performance of the controller. This clearly confirms the validity of the proposed approach, and its practicality for use with actual weather forecasts.

5.5. Alternative optimization approaches

For an in-depth comparison of the outcome of different optimization strategies, three alternative approaches (cf. Section 4.1) were also applied to optimize the most promising type of energy management controller E_3 with 3-h energy availability outlook. The performance of the best controllers with 3 h energy availability outlook, optimized by PSO, GA, and CMA-ES, is summarized in Table 7. It can be seen that the different optimization strategies (cf. Table 4) have triggered evolution of controllers with different high-level properties. When used with the alternative approaches, the objective function f_{obj} (8) developed in this work yields controllers that harvest less energy and perform smaller number of measurements, as compared to DE. Only PSO produced a controller comparable with that obtained by DE. Although the final value of training fitness of PSO is slightly lower than that of DE, PSO performs worse on the test data and also using real forecasts. This is likely caused by overfitting of the PSO obtained controller, often associated with a quick convergence clearly visible from the graphs shown Fig. 15. In the case of GA, the evolved controllers also transmitted data, on average, with approximately 20 times longer cycle. With CMA-ES, even the best evolved controller featured long average transmission periods of more than 1.5 h (5400 s). This observation is in line



(a) Histogram of measurement periods

(b) Histogram of transmission periods

Fig. 14. Histograms of measurement and transmission periods for the best evolved controller E_3^{best} driven by numerical weather prediction forecasts (year 2012).

Table 7
Performance of the best controllers with 3 h energy outlook evolved by PSO, GA, and CMA-ES on training (trn) and testing (tst) data, and on actual forecasts of solar energy availability (real).

Algorithm	Data	Fitness	f_1	f_2	f_3	f_4	E_H [kJ]	M [–]	Fail [%]	$\bar{T}_s$ [s]	$\bar{T}_t$ [s]
PSO	trn	10.78	4.64	3.43	2.70	0.004	184.50	188,665	0.072%	167	361
	tst	19.78	4.58	8.28	6.92	0.000	176.15	179,983	0.128%	165	355
	real	12.59	4.72	4.42	3.45	0.000	183.57	185,642	0.185%	169	363
GA	trn	13.78	5.81	4.38	3.59	0.005	161.80	150,765	0.119%	209	433
	tst	21.73	6.60	8.24	6.89	0.002	141.17	124,909	0.184%	237	495
	real	11.16	6.36	2.75	2.05	0.001	153.27	137,781	0.149%	228	472
CMA-ES	trn	14.53	5.49	4.97	4.00	0.07	108.26	159,625	0.125%	198	6027
	tst	22.79	5.42	9.50	7.87	0.001	103.23	152,199	0.172%	195	5967
	real	12.20	5.50	3.80	2.91	0.0004	108.93	159,337	0.182%	198	5636

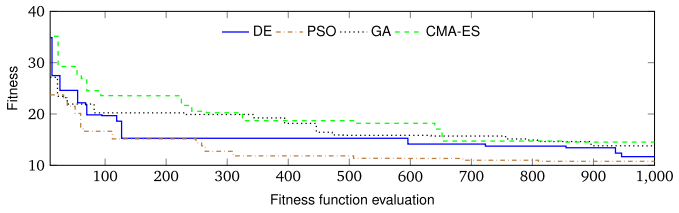


Fig. 15. Evolution of the best controllers with 3 h energy outlook by DE, PSO, GA, and CMA-ES.

with previous results on CMA-ES that show that it can be outperformed by other (more randomized) algorithms on multimodal functions that are not highly non-separable [29,53]. Although the fitness function might be tuned to better correspond to the high-level search approach provided by these algorithms, such research is beyond the scope of this application-focused study.

6. Conclusion and future work

This article introduced a new predictive energy management strategy for environmentally-powered wireless sensor nodes. The strategy has been designed in the form of a fuzzy control system, and optimized using evolutionary computing. The controller manages how energy is invested with respect to data collection and transmission. The environmentally-powered wireless sensor node uses energy outlook that estimates future energy available for harvest using location-specific predictions of solar irradiation.

The proposed system has been thoroughly examined using simulations based on a realistic, hardware-derived model of sensor node operation and energy-related environmental data collected at a particular deployment site. The proposed approach allows optimization of energy management strategy suitable for intended sensor node deployment sites without the need for expensive and time-consuming long-term hardware evaluation. The same data has been used for controller optimization and for determination of the optimal length of prediction horizon used to calculate the energy outlook. The obtained energy management strategies have also been compared to two static, baseline controllers and validated using real weather forecast data.

Conducted experiments demonstrate that the proposed controller design and optimization result in excellent monitoring performance with respect to sensing and transmission rates. This ensures effective acquisition of environmental data and appropriate hardware optimization (solar panel configuration, energy storage type a capacity, etc.) of sensor nodes or entire sensor networks.

Future work will consider application of alternative optimization approaches. Given the multi-objective nature of the optimization problem, the weighted aggregation approach [6] used in the current work could be replaced by a suitable population- or pareto-based method [25]. However, these alternatives also face the problem of determining the relative importance or dominance of individual sub-objectives that will have to be resolved with respect to the requirements of particular application.

In the long term, substantial effort will concentrate on real-world, field validation of the simulation results presented in this contribution. It is expected that the experience obtained during such field trials will allow further optimization of sensor node hardware and potential improvement of the developed simulation environment.

Acknowledgements

This work has been supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, by TECTERRA project 1209-UNI-013, and by projects SP2015/154 and SP2016/97 of the Student Grant System of VSB-TU Ostrava.

References

[1] M. Affenzeller, S. Winkler, S. Wagner, A. Beham, Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications, Chapman & Hall/CRC, 2009.
[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38 (4) (2002) 393–422.
[3] Alberta Agriculture and Rural Developement, AgroClimatic Information Service, 2013 http://agriculture.alberta.ca/acis/.
[4] P. Angelov, D.P. Filev, N. Kasabov, Evolving Intelligent Systems: Methodology and Applications, Wiley-IEEE Press, 2010.
[5] A. Aprem, C. Murthy, N. Mehta, Transmit power control policies for energy harvesting sensors with retransmissions, IEEE J. Sel. Top. Signal Process. 7 (October (5)) (2013) 895–906.
[6] B.V. Babu, M.M.L. Jehan, Differential evolution for multi-objective optimization, in: 2003 Congress on Evolutionary Computation, CEC 2003 – Proceedings, vol. 4, 2003, pp. 2696–2703.
[7] C. Bergonzini, D. Brunelli, L. Benini, Algorithms for harvested energy prediction in batteryless wireless sensor networks, in: 3rd International Workshop on Advances in Sensors and Interfaces, IWASI 2009, 2009, pp. 144–149.
[8] M. Bodur, A. Acan, T. Akyol, Fuzzy system modeling with the genetic and differential evolutionary optimization, in: International Conference on Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, vol. 1, 2005, November, pp. 432–438.
[9] R. Bogue, Solar-powered sensors: a review of products and applications, Sens. Rev. 32 (2) (2012) 95–100.
[10] P. Bonissone, K. Chiang, Fuzzy logic controllers: from development to deployment, in: IEEE International Conference on Neural Networks, vol. 1, 1993, pp. 610–619.
[11] B. Carse, T.C. Fogarty, A. Munro, Evolving fuzzy rule based controllers using genetic algorithms, Fuzzy Sets Syst. 80 (3) (1996) 273–293.
[12] A. Castagnetti, A. Pegatoquet, C. Belleudy, M. Auguin, An efficient state of charge prediction model for solar harvesting WSN platforms, in: 2012 19th International Conference on Systems Signals and Image Processing, IWSSIP, 2012, pp. 122–125.
[13] O. Castillo, R. Martínez-Marroquín, P. Melin, F. Valdez, J. Soria, Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot, Inf. Sci. 192 (2012) 19–38, Swarm Intelligence and Its Applications.
[14] C. Chen, S. Yang, A knowledge-based cooperative differential evolution for neural fuzzy inference systems, Soft Comput. 17 (5) (2013) 883–895.

[15] S.-H. Chen, W.-H. Ho, J.-H. Chou, M.-H. Lin, Design of robust-optimal controllers with low trajectory sensitivity for uncertain Takagi–Sugeno fuzzy model systems using differential evolution algorithm, Optim. Control Appl. Methods 33 (6) (2012) 696–712.
[16] F. Cheong, R. Lai, Designing a hierarchical fuzzy logic controller using the differential evolution approach, Appl. Soft Comput. 7 (2) (2007) 481–491.
[17] O. Cordón, A historical review of evolutionary learning methods for Mamdani-type fuzzy rule-based systems: designing interpretable genetic fuzzy systems, Int. J. Approx. Reason. 52 (September (6)) (2011) 894–913.
[18] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, Genetic fuzzy rule-based systems based on the Michigan approach, World Sci. (2001) 153–178 (chapter 6).
[19] S. Das, P. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (February (1)) (2011) 4–31.
[20] R. Dayal, K. Modepalli, L. Parsa, A new optimum power control scheme for low-power energy harvesting systems, IEEE Trans. Ind. Appl. 49 (November (6)) (2013) 2651–2661.
[21] J.S. Dyer, Multiple Criteria Decision Analysis: State of the Art Surveys, Springer, New York, NY, 2005, pp. 265–292 (ch. Maut – Multiattribute Utility Theory).
[22] M. Eftekhari, S. Katebi, M. Karimi, A. Jahanmiri, Eliciting transparent fuzzy model using differential evolution, Appl. Soft Comput. 8 (1) (2008) 466–476.
[23] A. Elefsiniotis, M. Weiss, T. Becker, U. Schmid, Efficient power management for energy-autonomous wireless sensor nodes for aeronautical applications, J. Electron. Mater. 42 (7) (2013) 1907–1910.
[24] EMEND Project, Ecosystem-based Research into Boreal Forest Management, 2013, December http://www.emendproject.org/pages/read/about.
[25] A. Engelbrecht, Computational Intelligence: An Introduction, 2nd ed., Wiley, New York, NY, USA, 2007.
[26] N. Ferry, S. Ducloyer, N. Julien, D. Jutel, Energy estimator for weather forecasts dynamic power management of wireless sensor networks, in: Lect. Notes Comput. Sci., 2011, pp. 122–132.
[27] P. Gyorke, B. Pataki, Application of energy-harvesting in wireless sensor networks using predictive scheduling, in: 2012 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), 2012, May, pp. 582–587.
[28] N. Hachicha, B. Jarboui, P. Siarry, A fuzzy logic control using a differential evolution algorithm aimed at modelling the financial market dynamics, Inf. Sci. 181 (January (1)) (2011) 79–91.
[29] N. Hansen, S. Kern, Evaluating the cma evolution strategy on multimodal test functions, in: X. Yao, E.K. Burke, J.A. Lozano, J. Smith, J.J. Merelo-Guervós, J.A. Bullinaria, J.E. Rowe, P. Tiňo, A. Kabán, H.-P. Schwefel (Eds.), Parallel Problem Solving from Nature – PPSN VIII: 8th International Conference, Birmingham, UK, September 18–22, 2004 Proceedings, Springer, Berlin, Heidelberg, 2004, pp. 282–291.
[30] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evol. Comput. 9 (June (2)) (2001) 159–195.
[31] F. Herrera, M. Lozano, J. Verdegay, Tuning fuzzy logic controllers by genetic algorithms, Int. J. Approx. Reason. 12 (3–4) (1995) 299–315.
[32] Y. Hung, F. Lin, J. Hwang, J. Chang, K. Ruan, Wavelet fuzzy neural network with asymmetric membership function controller for electric power steering system via improved differential evolution, IEEE Trans. Power Electron. 30 (4) (2015) 2350–2362.
[33] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evol. Comput. 15 (March (1)) (2007) 1–28.
[34] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conf. on Neural Networks, 1995. Proceedings, vol. 4, 1995, pp. 1942–1948, vol.4.
[35] M.-A. Koulali, A. Kobbane, M.E. Koutbi, H. Tembine, J. Ben-Othman, Dynamic power control for energy harvesting wireless multimedia sensor networks, EURASIP J. Wirel. Commun. Netw. (2012) 158.
[36] P. Krömer, M. Prauzek, P. Musilek, Harvesting-Aware control of wireless sensor nodes using fuzzy logic and differential evolution, in: Workshop on Energy Harvesting Communications, IEEE SECON 2014, Singapore, 2014, June, pp. 51–56.
[37] P. Krömer, M. Prauzek, P. Musilek, T. Barton, Optimization of wireless sensor node parameters by differential evolution and particle swarm optimization, in: IBICA, 2014, pp. 13–22.
[38] J.C.Y. Lai, F.H.F. Leung, S.H. Ling, H.T. Nguyen, Hypoglycaemia detection using fuzzy inference system with multi-objective double wavelet mutation differential evolution, Appl. Soft Comput. J. 13 (5) (2013) 2803–2811.
[39] Y. Li, Z. Jia, S. Xie, Energy-prediction scheduler for reconfigurable systems in energy-harvesting environment, IET Wirel. Sens. Syst. 4 (2) (2014) 80–85.
[40] S. Ling, H. Iu, K. Chan, H. Lam, B. Yeung, F. Leung, Hybrid particle swarm optimization with wavelet mutation and its industrial applications, IEEE Trans. Syst. Man Cybern. B: Cybern. 38 (June (3)) (2008) 743–763.
[41] M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, MA, 1996.
[42] V. Mukherjee, S. Ghoshal, Intelligent particle swarm optimized fuzzy {PID} controller for {AVR} system, Electr. Power Syst. Res. 77 (12) (2007) 1689–1698.
[43] S.-K. Oh, W.-D. Kim, W. Pedrycz, Design of optimized cascade fuzzy controller based on differential evolution: simulation studies and practical insights, Eng. Appl. Artif. Intell. 25 (3) (2012) 520–532.
[44] M. Pandit, L. Srivastava, M. Sharma, Environmental economic dispatch in multi-area power system employing improved differential evolution with fuzzy selection, Appl. Soft Comput. J. 28 (2015) 498–510.
[45] W. Pedrycz, F. Gomide, An Introduction to Fuzzy Sets: Analysis and Design. A Bradford Book, MIT Press, 1998.
[46] J. Piorno, C. Bergonzini, D. Atienza, T. Rosing, Prediction and management in energy harvested wireless sensor nodes, in: 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology 2009, Wireless VITAE 2009, 2009, May, pp. 6–10.
[47] R. Poli, W.B. Langdon, N.F. McPhee, A Field Guide to Genetic Programming, Lulu Press, 2008.
[48] T. Prabhakar, S. Devasenapathy, H. Jamadagni, R. Prasad, Smart applications for energy harvested WSNS, in: 2nd International Conference on COMmunication Systems and NETworks, COMSNETS 2010, 2010, p. 2010.
[49] R. Prado, S. García-Galán, J. Expósito, n.A. Yuste, S. Bruque, Learning of fuzzy rule-based meta-schedulers for grid computing with differential evolution, in: E. Hüllermeier, R. Kruse, F. Hoffmann (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems. Theory and Methods. Vol. 80 of Communications in Computer and Information Science, Springer, Berlin, Heidelberg, 2010, pp. 751–760.
[50] M. Prauzek, P. Musilek, A.G. Watts, Fuzzy algorithm for intelligent wireless sensors with solar harvesting, in: Symposium on Intelligent Embedded Systems (IES 2014), 2014 IEEE Symposium Series on Computational Intelligence, Orlando, FL, 2014, pp. 1–7.
[51] K.V. Price, R.M. Storn, J.A. Lampinen, Differential Evolution A Practical Approach to Global Optimization. Natural Computing Series, Springer-Verlag, Berlin, Germany, 2005.
[52] A. Qing, C. Lee, Differential Evolution in Electromagnetics. Adaptation, Learning, and Optimization, Springer-Verlag, 2010.
[53] S. Rahnamayan, P. Dieras, Efficiency competition on n-queen problem: DE vs. CMA-ES, in: Canadian Conference on Electrical and Computer Engineering, 2008. CCECE 2008, 2008, May, pp. 000033–000036.
[54] M. Randall, Differential evolution for a constrained combinatorial optimisation problem, Int. J. Metaheurist. 1 (December (4)) (2011) 279–297.
[55] J. Rodway, P. Musilek, E. Lozowski, J. Heckenbergerova, M. Prauzek, Pressure-based prediction of harvestable energy for powering environmental monitoring systems, in: 15th IEEE Int. Conf. on Environment and Electrical Engineering (EEEIC 2015), IEEE, Rome, Italy, 2015, June, pp. 725–730.
[56] M. Severini, S. Squartini, F. Piazza, An energy aware approach for task scheduling in energy-harvesting sensor nodes, in: J. Wang, G. Yen, M. Polycarpou (Eds.), Advances in Neural Networks ïĚ! ISNN 2012. Vol. 7368 of Lecture Notes in Computer Science., Springer, Berlin, Heidelberg, 2012, pp. 601–610.
[57] M. Sheikhan, S.A. Ghoreishi, Antiviral therapy using a fuzzy controller optimized by modified evolutionary algorithms: a comparative study, Neural Comput. Appl. 23 (6) (2012) 1801–1813.
[58] W.C. Skamarock, J.B. Klemp, J. Dudhia, D.O. Gill, D.M. Barker, M.G. Duda, X.-Y. Huang, W. Wang, J.G. Powers, A Description of the Advanced Research WRF Version 3. NCAR Technical Note 113, 2008.
[59] J.A. Stankovic, T. He, Energy management in sensor networks, Philos. Trans. A: Math. Phys. Eng. Sci. 370 (January (1958)) (2012) 52–67.
[60] H. Su, Y. Yang, Differential evolution and quantum-inquired differential evolution for evolving Takagi–Sugeno fuzzy models, Expert Syst. Appl. 38 (June (6)) (2011) 6447–6451.
[61] S. Sudalaiandi, B. Subramanian, Design of multivariable fractional order PID controller using covariance matrix adaptation evolution strategy, Archiv. Control Sci. 24 (June (2)) (2014) 119–251.
[62] M.K. Szczodrak, O. Gnawali, L.P. Carloni, Modeling and implementation of energy neutral sensing systems, in: Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems. ENSSys'13, ACM, New York, NY, USA, 2013, 9:1–9:6.
[63] D. Vergados, G. Stassinopoulos, Adaptive duty cycle control for optimal stochastic energy harvesting, Wirel. Pers. Commun. 68 (1) (2013) 201–212.
[64] A.G. Watts, Managing energy-harvesting environmental monitoring systems, University of Alberta, 2014 (Master's thesis).
[65] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (August (1)) (2002) 67–82.
[66] S. Yang, X. Yang, J. McCann, T. Zhang, G. Liu, Z. Liu, Distributed networking in autonomic solar powered wireless sensor networks, IEEE J. Sel. Areas Commun. 31 (12) (2013) 750–761.
[67] H. Yoo, M. Shim, D. Kim, Dynamic duty-cycle scheduling schemes for energy-harvesting wireless sensor networks, Commun. Lett. IEEE 16 (February (2)) (2012) 202–204.
[68] B. Zhang, R. Simon, H. Aydin, Harvesting-aware energy management for time-critical wireless sensor networks with joint voltage and modulation scaling, IEEE Trans. Ind. Inf. 9 (February (1)) (2013) 514–526.
[69] Y. Zhang, S. He, J. Chen, Y. Sun, X. Shen, Distributed sampling rate control for rechargeable sensor nodes with limited battery capacity, IEEE Trans. Wirel. Commun. 12 (June (6)) (2013) 3096–3106.
[70] Y. Zheng, H. Ling, S. Chen, J. Xue, A hybrid neuro-fuzzy network based on differential biogeography-based optimization for online population classification in earthquakes, IEEE Trans. Fuzzy Syst. 23 (4) (2015) 1070–1083.