

## Research Article

# Genetical Swarm Optimization of Multihop Routes in Wireless Sensor Networks

**Daive Caputo, Francesco Grimaccia, Marco Mussetta, and Riccardo E. Zich**

*Politecnico di Milano, Dipartimento di Energia, Via La Masa, 34, I-20156 Milano, Italy*

Correspondence should be addressed to Marco Mussetta, marco.mussetta@polimi.it

Received 19 March 2010; Revised 4 August 2010; Accepted 9 November 2010

Academic Editor: Tzung P. Hong

Copyright © 2010 Davide Caputo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, wireless sensor networks have been attracting considerable research attention for a wide range of applications, but they still present significant network communication challenges, involving essentially the use of large numbers of resource-constrained nodes operating unattended and exposed to potential local failures. In order to maximize the network lifespan, in this paper, genetical swarm optimization (GSO) is applied, a class of hybrid evolutionary techniques developed in order to exploit in the most effective way the uniqueness and peculiarities of two classical optimization approaches; particle swarm optimization (PSO) and genetic algorithms (GA). This procedure is here implemented to optimize the communication energy consumption in a wireless network by selecting the optimal multihop routing schemes, with a suitable hybridization of different routing criteria, confirming itself as a flexible and useful tool for engineering applications.

## 1. Introduction

Current sensor networks address problems of meeting standards for accuracy and also delivering data from remote locations with an appropriate level of spatial and temporal resolution.

Today, advances in sensor technology, wireless communications, and digital electronics not only unfold many possible applications but also introduce challenging issues for the collaborative processing of signals in an energy-constrained distributed sensor network [1]. The advantages are evident not only in the reduction of size, but also in the increase of functional performance and reliability and a unit-cost decrease in high-volume batch processing.

Wireless sensor networks are being designed for a wide variety of both civil and military purposes. A WSN consists of a large number of sensor nodes generally deployed in harsh environments, and it is designed essentially to detect events or natural phenomena and to collect, process, and transmit sensed data to interested users. Developing networks of low-cost, low-power, and multifunctional sensors has received increasing attention in the last years [2]. These sensors are small in size and able to sense, process, and communicate data with each other, typically over a radio frequency (RF)

channel. Because sensors are usually used to measure and monitor some parameters that may vary with place and time, a large number of sensors are typically required in order to obtain samples of these parameters at different locations and times.

Some of the applications foreseen for WSNs will require a number of devices of the order of many thousands nodes. The strategy of deploying a large number of inexpensive sensors has significant advantages, at smaller system total cost: much higher spatial resolution, higher robustness against failures through distributed operation, uniform coverage, little obtrusiveness, ease of deployment, reduced energy consumption, and, consequently, increased system lifetime. The main point is to locate sensors close to the source of a potential phenomenon, where the acquired data are likely to have the greatest benefit or impact.

A large number of sensor nodes and the dynamics of their operating environments (e.g., limited battery power and hostile physical environment) pose unique challenges in the design of sensor networks and their applications. Furthermore, questions concerning how can accessed information be queried and collected by and stored within a sensor network are still open issues. Moreover the different types of sensors may differ in size, computational and energy

capabilities, functions to be performed, parameters to be measured, and mobility patterns.

For these reasons sensor network modelling and design represent a very complex and sophisticated process, and performing system optimization is fundamental to make the employment of such new technologies for real life improvements realistic. The energy constraints of sensor networks, especially when their size increases, represent a large domain problem, and it is necessary to take into account many optimization parameters at the same time. For this reason we decided to investigate the effectiveness of evolutionary algorithms to optimize network lifetime in a limited power supply framework, mainly focusing on communication strategies [3].

In the last decades several evolutionary algorithms have been developed for the optimization of different kinds of engineering problems. All these algorithms are population-based heuristic search techniques, which can be used to solve general combinatorial optimization problems, modeled on the concepts of natural selection and evolution, or based on cultural and social behaviors typical of the swarm intelligence. In Section 2, more details and examples will be provided about different algorithms found in recent literature and some hybrid strategies will be reviewed before presenting the GSO algorithm itself. Principal attributes of the GSO algorithm class are described both analyzing its proper mechanism and showing some results of recent enhancements we propose for this specific application.

In particular, we decided to maintain the basic algorithm structure since its reliability has already been proven in previous studies and the focus of the paper does not allow us to perform a complete comparative study with respect to other heuristic approaches. For example, the driving hybridization parameter itself opens up different criteria for deciding the population quota evolving with various strategies, and many examples of the fitness diversity adaptation and Meta-Lamarckian learning can be found in literature [4–7]. More tests has been performed here to further demonstrate the algorithm robustness and a new interface has been developed in order to adapt this hybrid algorithm to WSN problem.

In Section 3, WSN optimization problem is discussed also referring to examples which can be found in literature. Section 4 describes the considered network model and some assumptions we made before performing the numerical simulations. Section 5 illustrates the procedure we developed to apply an evolutionary optimization method to WSN model. Finally, numerical implementation and results for wireless sensor network optimization are reported in Section 6.

## 2. Genetical Swarm Optimization Algorithm and Hybrid Strategies

The GSO algorithm was conceived as a hybrid evolutionary technique developed in order to combine in the most effective way the properties of GA and PSO and to overcome the problem of premature convergence. Several evolutionary algorithms have been developed in the last years for the

optimization of different kinds of engineering problems. All these algorithms are heuristic techniques, which can be used to solve combinatorial problems, modeled on the concepts of natural selection, following the classical Darwin's theory of evolution, or based on cultural and social learning, closer to the Lamarckian idea that an organism can transmit characteristics acquired during its lifetime. genetic algorithm (GA), for example, is one of the most effective evolutionary algorithms developed until now [8]. It simulates natural evolution, in terms of survival of the fittest, adopting pseudobiological operators such as selection, crossover, mutation, and many other additional operators introduced to get a faster convergence rate. On the other hand, particle swarm optimization (PSO) is based on a model of social cooperation between independent agents (particles), and it uses common knowledge to stimulate a migration of the swarm toward a global maximum or minimum [9]. Some comparisons of the performances of GA and PSO [10] emphasize the reliability and convergence speed of both methods but still keep them separate. In particular, GA and PSO show a marked application-driven characteristic for any respective technique: PSO seems to have faster convergence rate than GA early in the run, but it is often outperformed by GA for long simulations when the last one finds better solutions.

Many efforts on the enhancement of traditional methods have been proposed through modifying the structure of the population or the role that an individual plays in it, especially for GAs (e.g., distributed GA, cellular GA, and symbiotic GA), by modifying the basic operators of traditional procedures. Several other additional operators have been proposed for GAs in order to get a faster convergence rate. Hybrid genetic algorithm-local search (GA-LS) methods, for example, incorporate local improvement procedures with traditional GAs, to improve the performance of GAs in searching and solving design problems, not only finding better solutions than simple GAs, but also searching more efficiently. Such hybrid GA-LSs are also known as Lamarckian learning GAs, Baldwinian learning GAs [11], and Memetic Algorithms (MAs), which use different models of evolution to incorporate learning into a GA. The Baldwin effect allows an individual's fitness (phenotype) to be determined based on learning (i.e., the application of local improvement); like natural evolution, the result of the improvement does not change the genetic structure (genotype) of the individual. Lamarckian evolution, on the other hand, does change the genetic structure of the individual, in order to reflect the result of learning.

Many attempts to exploit the qualities of different algorithms have been done with some kind of integration in particular of the two main strategies [12, 13], but GSO aims to reach a stronger cooperation between the two techniques stressing its hybrid nature and maintaining the GA and PSO integration for the entire run. In the last few years, we have proved repeatedly that such an updating technique can improve traditional mechanisms increasing individuals' score by means of an effective combination of natural selection and knowledge sharing. This key feature makes it suitable for a wide range of applications.

In particular, in [14, 15], we presented some comparisons of GSO and classical methods performances, emphasizing the reliability and convergence speed of the first one and applying it to different case studies. In these previous experiments, for example, in large EM optimization problems, the algorithm proved to be a fast and robust technique, outperforming classical procedures.

**2.1. The Genetical Swarm Optimization Mechanism.** The basic concepts of GSO have been presented in [14], and they are clearly summarized in Figure 1: every iteration of the population is randomly divided into two parts which are processed by GA and PSO techniques, respectively. Then the fitness of the newly generated individuals is evaluated and they are recombined in the updated population which is again divided into two parts in the next iteration for the next run of genetic or particle swarm operators.

The population update concept can be easily understood thinking that some of the individuals are substituted by newly generated ones by means of GA, while the remaining are the same of the previous iteration but moved on the solution space by PSO. In Figure 1 it is possible to better understand the key steps of the GSO algorithm, through an intuitive flowchart.

The driving parameter of GSO algorithm is the *hybridization coefficient* ( $hc$ ); it expresses the percentage of population that in each iteration is processed by GA: so,  $hc = 0$  means the procedure is a pure PSO (the whole population is processed according to PSO operators),  $hc = 1$  means pure GA (the whole population is optimized according to GA operators), while  $0 < hc < 1$  means that the corresponding percentage of the population is developed by GA, while the rest with PSO technique.

In [14] GSO has been tested on problem of different dimensions: while for a small number of unknowns GSO performance is similar to GA and PSO ones ( $N = 5$ ), if the size of the problem increases ( $N = 10, 20$ ), GSO behavior improves and outperforms GA and PSO during iterations. Moreover, the best  $hc$  value found in that preliminary study does not depend on the dimension of the problem, as it has been reported also in [16]. Furthermore, the obtained best  $hc$  value (0.2) means that, for a big-sized problem, the basic PSO can be strongly improved by adding a small percentage of genetic operators to the population. In further studies a convenient value was found to be  $0.2 \leq hc \leq 0.3$  for several fitness functions, but we extended the class of GSO algorithms by considering several variation rules for  $hc(k)$ , where  $k = 1 \dots K$ , with  $k$  as the current iteration and  $K$  as the total number of iterations; this was done in order to explore different hybridization strategies for the GSO algorithm and to compare new approaches with others already present in literature: the complete set of hybridization rules considered so far has been already reported in [14].

**2.2. Recently Developed Enhancements.** In particular, in [15] we introduced new rules for varying the  $hc$  value during the run, to effectively combine the properties of GA and PSO, in order to have a more general procedure. In fact, for engineering optimization problems, the best mix of GA

and PSO operators cannot be always obvious. In particular, there are situations where a fixed  $hc$  is the right choice, and others where a variable  $hc(k)$  during the run is better. This means that also the “amount” of hybridization plays a role in affecting the performances of this procedure, but the correct value for  $hc$  is hardly known *a priori*. Therefore, we chose to let the procedure adjust the  $hc(k)$  value by itself during the iterations, according to a predefined set of rules.

In the *dynamical* approach, the  $hc$  parameter is updated during iterations according to the following rule:

$$hc(k') = \begin{cases} hc(k) + \nu \frac{e^{-\xi(k'/K)}}{N} & \text{if } \Delta \hat{f}(k') < \Delta \hat{f}(k), \\ hc(k) & \text{if } \Delta \hat{f}(k') \geq \Delta \hat{f}(k), \end{cases} \quad (1)$$

where  $k' = k + \Delta k$ ,  $\nu = \pm 1$  (*versus*),  $\xi = 2$  (*damping*),  $\Delta \hat{f}(k') = \hat{f}(k') - \hat{f}(k)$ , and  $K$  is the total number of iterations; here  $\hat{f}(k')$  is the best fitness value obtained after  $k'$  iterations and  $\Delta k = 5$ .

This approach is somehow similar to what has been presented in [4] in terms of adaptive strategies for Meta-Lamarckian learning. In that case, a roulette wheel procedure has been used, where best performing local strategies are rewarded to increase their probability to be selected. In the *dynamical* approach here presented, instead, the  $hc$  value is merely oscillating between its extreme values in order to vary when the convergence rate is approaching stagnation.

It is also worth noting that several interesting and effective strategies have been proposed in literature for the self-adaptation of Memetic Algorithms [5–7]. In the present case, however, we chose to develop a completely new self-adaptive implementation. In fact GSO cannot be defined as a Memetic Algorithm since GA and PSO are both global strategies here completely interchangeable during the run. Moreover, while in the cited papers the adaptive procedure, used to select a local strategy among several considered methods, is mainly based on evaluation of population diversity, in this paper we aim to self-adapt the  $hc$  value to increase the speed of convergence.

Thus in the so-called *self-adapting* approach, the rule implemented for the self-adaptation comes, in part, from the very simple and reliable PSO technique: in fact, if we consider the value of  $hc(k')$  in the  $k'$ th iteration, then we can call  $hc_v(k')$  the variation between  $hc(k')$  and  $hc(k)$  and so we can write

$$hc(k') = hc(k) + hc_v(k'). \quad (2)$$

Therefore, the problem is simply to find the right *velocity update* to properly change  $hc$  during the run. According to the PSO similarity, we can define a personal best  $hc_p$  value that has been obtained during the run and therefore write

$$hc_v(k') = \omega \cdot hc_v(k) + \phi \cdot \eta \cdot (hc_p - hc(k)), \quad (3)$$

where  $hc_p$  is chosen analyzing the slope of the fitness score increase, during the iterations; that is, if in iteration  $\bar{k}$  the increment of fitness is higher than in the previous history, then  $hc_p = hc(\bar{k})$ .

TABLE 1: Results of the sinc function optimization for different problem dimensions and rules of variation for the  $hc$  parameter during iterations: average final values over 100 independent trials and, in parentheses, their standard deviation  $\sigma$ .

Optimization Strategy	$hc(k)$	Dimension of the considered optimization problem					
		$N = 5$		$N = 10$		$N = 20$	
		Function evaluations		Function evaluations		Function evaluations	
		$5 \cdot 10^3$	$10^4$	$25 \cdot 10^3$	$5 \cdot 10^4$	$2 \cdot 10^5$	$4 \cdot 10^5$
Static GSO	$hc(k) = 0.2$ for all $k$	0.999 (0.000)	1.000 (0.000)	0.982 (0.122)	1.000 (0.000)	1.000 (0.000)	1.000 (0.000)
Dynamical GSO	(see (1))	0.998 (0.001)	1.000 (0.000)	0.990 (0.087)	1.000 (0.000)	0.743 (0.402)	0.832 (0.345)
Self-Adaptive GSO	(see (2), (3))	0.989 (0.081)	0.999 (0.001)	0.989 (0.087)	0.999 (0.001)	0.732 (0.408)	0.838 (0.337)
GA	$hc(k) = 1$ for all $k$	0.977 (0.139)	0.988 (0.091)	0.955 (0.169)	0.996 (0.003)	0.505 (0.412)	0.776 (0.339)
PSO	$hc(k) = 0$ for all $k$	0.884 (0.301)	0.961 (0.184)	0.415 (0.445)	0.510 (0.459)	0.020 (0.053)	0.026 (0.057)

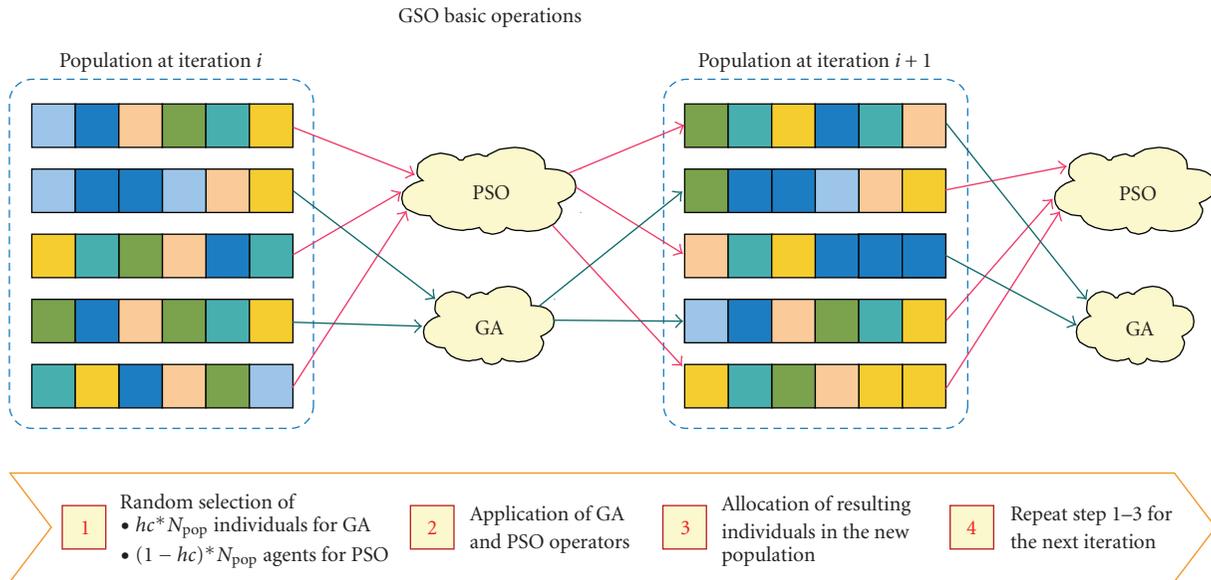


FIGURE 1: Basic steps in hybridizing GA and PSO and setting up population for GSO algorithm.

What is different, in this case, is the fact that there is only one single agent for a monodimensional problem; therefore, the personal best of  $hc$  is also the global best, and so the definition of (3) is simpler than the traditional expression for the velocity update. Therefore this updating procedure cannot take advantage of the cooperation between the particles, differently from PSO, and so we chose, if the slope of the fitness curve and the velocity are equal to zero, to add a random *mutation* to the  $hc$  value to help it to explore other possibilities.

2.3. *Additional Performance Analysis.* In [14] several preliminary numerical results have been reported to test the ability of GSO to optimize standard analytical benchmark functions. To further investigate the effectiveness of the

adaptive approaches proposed above, Table 1 summarizes some additional results obtained for the optimization of an  $N$ -dimensional analytical test function defined as follows:

$$f(X) = \prod_{i=1}^N \frac{\sin(x_i - x_{0,i})}{(x_i - x_{0,i})}, \quad (4)$$

where  $N$  represents the dimension of the problem (i.e., the number of parameters to optimize). This is the well-known *sinc* function that has a single maximum value and several local maxima. The location of the maximum is  $X_0 = \{x_{0,1}, \dots, x_{0,N}\}$  and  $f(X_0) = 1$ .

The results reported in Table 1 represent the value of  $f(X)$  obtained by different hybridization strategies (GA and PSO included for comparison) for different dimensions of

the problems and for different number of iterations: these results are the average over 100 independent trials, and therefore the standard deviation  $\sigma$  is reported too (in parentheses). It is clear from these results that the considered version of GSO is faster than GA and PSO in finding the maximum value: in fact, for the same problem dimension and number of function evaluations, the average results obtained by GSO techniques are always greater than those obtained by GA and PSO, and the improvement is more relevant when the dimension of the problem increases; moreover, the reported values of standard deviation show that also the reliability of the considered methods is better than the one of traditional techniques, thus confirming the robustness of the developed procedure. The overall results reported here and in cited papers show that, although the static GSO is generally the fastest and the most robust strategy in order to optimize multimodal functions, an adaptive approach is a suitable and reliable solution especially when the proper *hc* value is not known *a priori* for a specific problem.

### 3. Wireless Sensor Network Optimization Issues

As described in the introduction, sensor networks are emerging as an important field of study in different technological areas and offer a rich domain of active research. These complex systems use technologies essentially from sensing, communication, and computing fields and combine at the same time many design issues of wireless communication and mobile computing. They still present significant system challenges since the network utilizes large numbers of sensors operating essentially as resource-constrained nodes and exposed to potential failures. Advances in sensor production technologies have made building very compact and autonomous sensor nodes possible, each containing one or more sensor devices, computation and communication capabilities, and limited power supply, promising for WSNs better coverage, higher resolution, fault tolerance, and robustness. Networking the sensors can facilitate the transmission and dissemination of the measured parameters to some collector sites at which the information is further processed for decision-making purposes. The distributed nature of the network and *deploy-and-leave* future vision make it even more attractive in military and other sensitive applications. Performing the processing at the source can dramatically reduce the computational cost of networking and management; therefore, WSN organization should be autonomously performed with a minimum of human interference.

Minimizing energy consumption certainly is the key requirement when designing sensor network protocols and algorithms. Since sensor nodes are equipped with small and very limited batteries, it is crucial for the network to be energy efficient in order to maximize network lifetime. However further requirements for a well-designed system are represented by the fault tolerance property, network scalability, reliability, and of course low production costs as much as possible. Even if nowadays analog and digital technologies make it possible to develop cheap, small, and

low-power sensor nodes, these capabilities cannot make the sensor network completely energy autonomous. It is possible to network microsensors where each sensor individually senses the environment, but collaboratively achieves complex information gathering and dissemination tasks [17]. All these features, integrated on a single device, contribute to define a *smart sensor*. Built to be energetically autonomous and to perform local filtering and preprocessing of the sensed data, a smart sensor is able to communicate and to share its information with all the other nodes that constitute the sensor network.

Generally, sensor network applications require quality of service guarantees suggesting the need for global network cost optimization. Many different approaches have been used in the WSN optimization research field. For example, in [18] an approach for optimizing global cost in sensor networks through greedy local decisions at each node is presented, and they explore the benefits of this approach in reducing the idle listening at individual nodes in order to reduce the global network energy cost. They consider essentially two representations for the local node state: the number of descendants in the routing tree and the number of descendants and duty cycle. In [19] and later in [20], an approach based on genetic algorithms is proposed to optimize design of application-specific WSNs with connectivity and energy conservation limitations. We develop a fitness function to incorporate different aspects of network performance, including the status of sensor nodes (active or inactive) and the choice of appropriate cluster heads but still referring to a single-hop routing scheme.

A model of metaheuristic optimization algorithm such as ACO (ant colony optimization), basically inspired by the behaviour of ants in finding paths from the colony looking for food, has been used in [21] in data gathering and communication for WSNs, assuming unlimited energy for the base station and comparing its results with LEACH and PEGASIS algorithms [22, 23], again following a cluster-based approach. A similar approach can be found in [24], but this time the ant colony optimization is applied with respect to an *ad hoc network* where nodes are not fixed. In [25] a global multihop routing strategy for heterogeneous sensor networks is proposed mixing tiered and clustered architectures through some heuristics.

In applying evolutionary techniques to WSN design, critical issues must be considered to trade-off among different objectives:

- (i) to reduce the amount of power wasted by wireless devices,
- (ii) to increase the network lifetime,
- (iii) to achieve fault tolerance in case of individual node failure,
- (iv) to allow network scalability and deployment,
- (v) to reduce bandwidth requirements, enhancing collaboration among nodes (e.g., data fusion) since the limited wireless channel bandwidth must be shared among all the sensors in the network.

Before building the objective function and performing the optimization, it is important to understand the main sources of energy loss in operation of wireless sensor networks in order to make the optimization process itself more effective. Basically the two main sources of power consumption are represented by communication or computation tasks [26]: communication involves essentially the use of transceivers while computation is related with processing aspects due to the chosen protocol and data compression techniques. A proper medium access control (MAC) protocol should be able to balance between computation and communication costs, trying to solve all the typical problems that cause energy waste in WSNs [27], namely:

- (i) *collision* (packet loss and packet retransmission),
- (ii) *overhearing* (reception of a packet destined to another node),
- (iii) *control packet overhead* (“MAC protocol’s overhead”: frame headers, signaling, etc.),
- (iv) *idle listening* (active listening to an idle channel waiting for a packet to arrive).

The radio transceiver of a node can operate in any of the following four different modes.

- (i) *Idle mode*: when a node is neither transmitting nor receiving a packet. This mode consumes power because the node has to listen to the wireless medium continuously in order to detect a packet that it should receive, so that the node can then switch into receive mode.
- (ii) *Transmit mode*: when a node is transmitting a packet.
- (iii) *Receive mode*: when a node is receiving a packet.
- (iv) *Sleep mode*: sleep mode has very low power consumption. The node’s transceiver when it is in sleep mode can neither transmit nor receive packets.

The goal of protocol development for environments with limited power resources is to optimize the transceiver usage for a given communication task. Thus, to optimize such a complex system is a major issue especially in terms of energy savings. As mentioned before, some attempts have been done in this direction using different evolutionary techniques, like genetic algorithm (GA) [8] and particle swarm optimization (PSO) [9]. In this context, the advantages of evolutionary computation and, generally speaking, artificial intelligence are the capability of using iterative progress to find an energy global optimum, without being trapped in local optima and the possibility to face nonlinear and discontinuous problems with a great number of variables.

Certainly GAs are used more frequently in route optimization due to the combinatorial nature of the problem, and generally they are limited to a single-hop cluster approach as in [20]. Both PSO and GA can be found in literature applied to target localization and sensor placement [28, 29], even if PSO seems to be less common than the other algorithm in this field.

We chose a multihop routing strategy for its intrinsic capability to avoid hidden terminals and saving energy at the

same time. Multihop wireless networks essentially use two or more wireless hops to convey information from a source to a destination, and this strategy allows for saving energy on condition that a proper path optimization is performed. After having first identified several design issues, the next section describes in more detail the network model of the specific infrastructure for being optimized.

#### 4. Network Model and Assumption

In this paper we study the multihop routing problem in static wireless sensor networks. We define the network area as a flat two-dimensional square where we randomly place  $n$  nodes. In order to assure a complete coverage of the selected area and a medium connectivity of the network, we also put some other constraints on the minimum and maximum distance among nodes. After the deployment, the position of the nodes remains the same (mobility issues are not addressed) and one of the nodes is arbitrarily chosen to be the base station. We suppose the whole network to be within the communication range of the base station allowing it to broadcast packets to all the nodes. That is possible since we assume that the base station is connected to the grid as in many commercial wireless sensor systems [30]. The base station controls the entire network being aware of the battery level of the nodes and their spatial coordinates. Moreover the base station has higher computational capabilities and is equipped with a different type of transceiver with a bigger transmitting range that allows it to reach the whole network.

Each node is equipped with a transceiver and a battery. For these simulations we adopted the same radio model proposed in [31] with

$$E_{\text{elec}} = 50 \text{ nJ/bit} \quad (5)$$

as the energy being dissipated to run the transmitter or receiver circuitry and

$$\varepsilon_{\text{amp}} = 100 \text{ pJ/bit/m}^2 \quad (6)$$

as the energy dissipation of the transmission amplifier.

Thus, transmission  $E_{Tx}$  and receiving  $E_{Rx}$  cost energies of the nodes that communicate in a round of the simulation are calculated as follows when a  $k$ -bit packet is transmitted along a distance  $d$ :

$$E_{Tx}(k, d) = E_{\text{elec}} \cdot k + \varepsilon_{\text{amp}} \cdot k \cdot d^\lambda, \quad (7)$$

$$E_{Rx}(k) = E_{\text{elec}} \cdot k. \quad (8)$$

We consider that the energy loss due to channel transmission, represented by the path-loss exponent  $\lambda$  in (7), is proportional to the square distance between sender and receiver (Figure 2).

We make the assumption that the energy needed for the transmission of one bit of data from node  $P$  to node  $Q$  is the same as to transmit one bit from  $Q$  to  $P$  (symmetric propagation channel). All of the nodes are assumed to be functionally equivalent and to have a uniform transmission range defined as a circle centered on the node. As mentioned

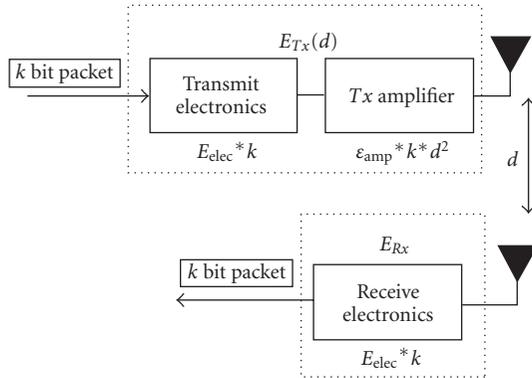


FIGURE 2: TX/RX channel transmission model (from [31]).

before, only the base station has a bigger transmission range that enables it to directly transmit packets to every node in the network. Each node is equipped with a GPS receiver such that it can establish its own location. We also consider that each node is aware of its battery status and of the position of its neighbors. In the set-up phase, after the choice of the base station, each node has to construct a local table of signal strengths recorded from all its neighbors, which should be a direct correlation to the distance between those nodes. This set-up phase must be performed only once at the startup of the system; therefore, it can be considered as a constant cost and should not affect the algorithm's performance beyond the set-up phase.

We simulate the transmission of data from every node to the base station through a random sequence of senders. At each round of simulation, a sender has to communicate the sensed data to the base station performing a multihop communication to reach it. At the beginning of each round, the base station updates the topology of the network removing the dead nodes and calculates the next optimal path. Since it has no power constraints, the base station can broadcast the results of its search to the entire network and synchronize the network clock. The nodes receive the information sent so they know if they will participate in the next route. The base station can easily update the battery level of the network at each step since collision and packet retransmission are avoided and only few nodes participate to communication. Thus, it is possible to calculate directly the energy waste associated to a single transmission (proportional to the distance between sender and receiver) automatically updating the battery level table after each round. In this way the control packet overhead needed to organize network communication task is reduced.

To describe the operative modes a node can assume, we follow the model presented in the previous paragraph, considering that transmit and receive modes all consume considerably more energy than the idle state, while the sleep mode does not imply any power consumption. For this reason, all the nodes not taking part into the next scheduled multihop path switch to sleep mode. They wake up switching to idle state only after a time calculated using information provided by the base station at the beginning of the round.

When the nodes are isolated or their battery level is close to zero, we considered them as dead nodes and we eliminated them from the sequence of senders and from the successive evaluations of the best possible paths. The simulation ends when the percentage of dead nodes is so high that the base station cannot be reached from the radio of the survived nodes. We did not take into account data aggregation that can be performed during the multihop path but further considerations can be done also on this important aspect.

A schematic description of a round of the algorithm is shown in Figure 3. As can be easily seen, the key ideas of this algorithm are to address all the computational costs to the base station (while sensor nodes are involved only in communication and sensing tasks) and to switch the sensor node to sleep mode whenever possible. The other main idea is to use the GSO presented in Section 2 to perform the choice of the optimal path between base station and nodes. In this study, we wanted to check the real capability of the algorithm in discovering the optimal routes without exploring the entire range of possible routes (the whole connection graph, described in more detail in Section 5).

In literature there are different routing protocols that use one or more criterion to evaluate the efficiency of a route in terms of power consumption, link quality [32], message overhead, time delay in delivering packets, and throughput. Five different metrics to determine optimal paths are presented in [33] (even if here only two of them are implemented). The most common criteria used in current routing protocols have two different basic ideas. Since radio communication cost is a function of the squared transmission distance, it is better to send information in several smaller hops rather than using a single-hop transmission over a long communication distance, but as analyzed in [31], in some situations, multihop routing (in particular MTE, minimum transmission energy routing protocol) could shorten the wireless network lifetime, performing worse than single-hop communication. Both techniques, anyway, cause an opposite mortality rate of the nodes of the network in the base of their spatial position: in fact, in single-hop communication, nodes that are far away from the base station die earlier, while in multihop communication the nodes closer to the base station consume their battery sooner, making the base station inaccessible for other sensor nodes (fast energy depletion problem).

In order to prove the performances of the GSO, we adopted some of the metrics described in [2] to choose the best possible paths.

(i) *Maximum Available Power (MAP)*. The route with maximum total available power is preferred. The total AP is calculated by summing the values of the batteries of each node along the route. Based on this approach, route SACFG could be selected in Figure 4, but since it includes the nodes in SCFG and an extra node, it is not power efficient even if it has a higher total AP. The best route becomes route SDEG. In our algorithm, we calculate the total power of the network summing all the values of the batteries except the battery of the base station. To evaluate the performance of the multihop

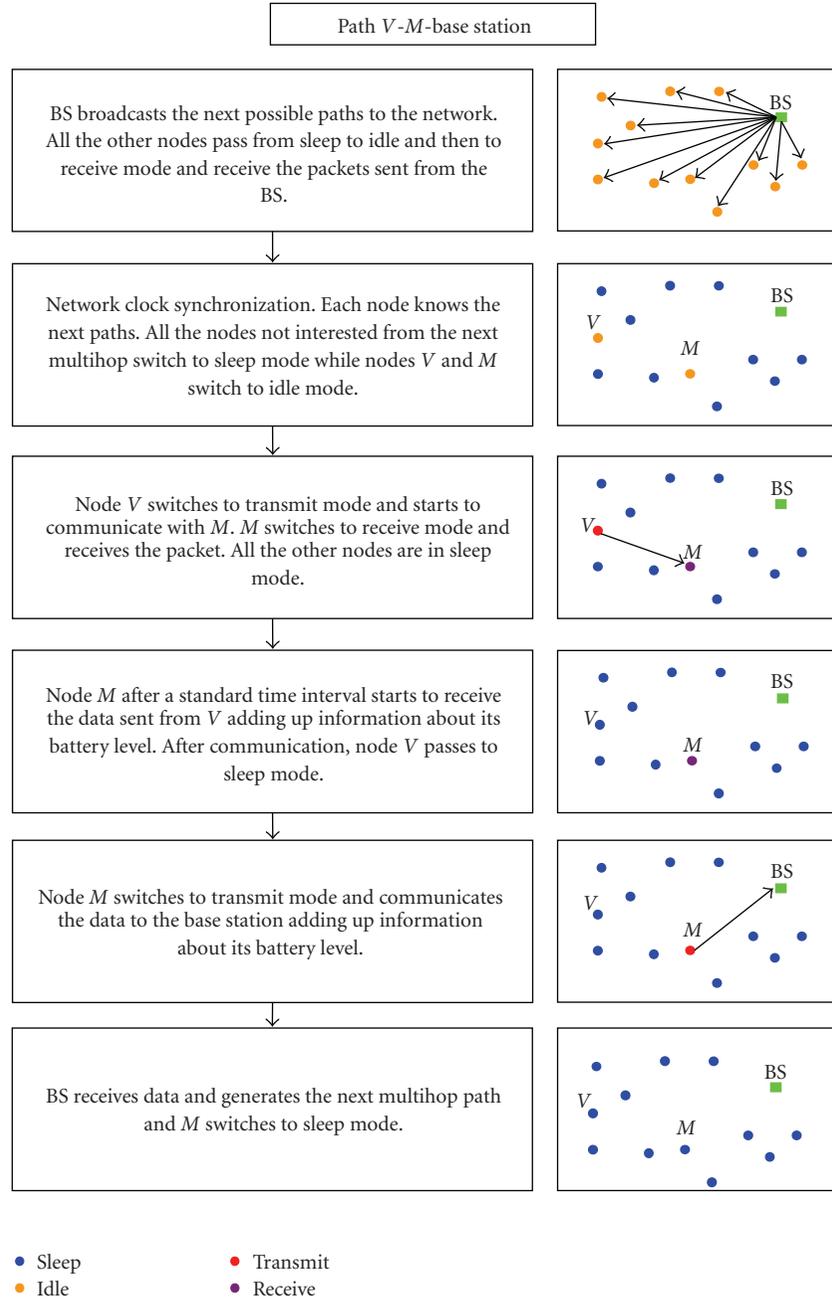


FIGURE 3: Schematic description of a basic multihop path.

path, we choose to define the first criterion by the following expression:

$$C_1 = \frac{(P_{\text{med}} - P_{\text{min}})}{(P_{\text{max}} - P_{\text{min}})}, \quad (9)$$

where  $P_{\text{med}}$  is the average power of the nodes that participate to the specific multihop (always excluding the gateway). We chose an average value so we did not take into account the problem of routes deriving from the extension of other possible routes because we wanted to study the efficiency of this metric without any superimposition with the other criteria (for example, criterion 3).  $P_{\text{max}}$  and  $P_{\text{min}}$  are the

maximum and minimum battery levels among all the sensors in a specific round.

(ii) *Minimum Communication Power (MCP)*. The MCP route requires minimum communication power to transmit the data packets from the sender to the base station. As shown in Figure 4, route  $SCFG$  is the MCP route. We considered the power expenses associated to transmission and reception of a standard packet of a fixed number of bit following (7) and calculated the total power consumption determined by the selected multihop ( $P_{h,\text{tot}}$ ). Then we calculated two reference power consumption,  $P_{\text{ref,max}}$  associated



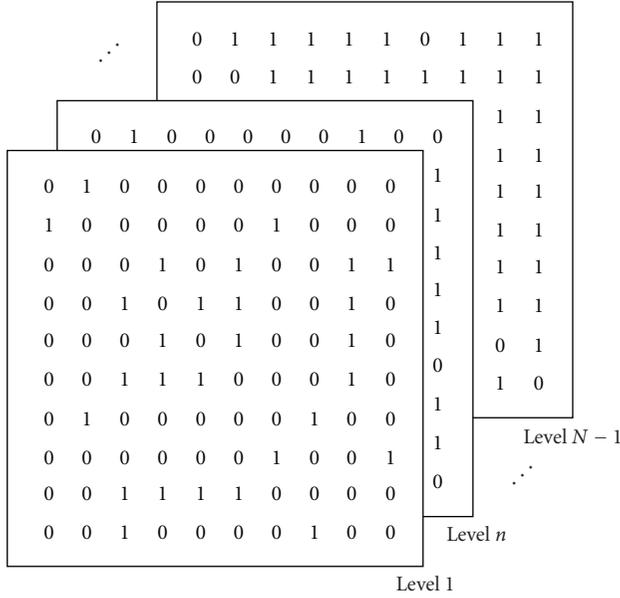


FIGURE 5: Different levels of adjacency matrix  $A_n$  for a network of  $N = 10$  nodes: nonzero values indicate that a link between two nodes  $(i, j)$  is possible in  $n$  steps.

If  $d_{ij}$  represents the Euclidean distance between nodes  $i$  and  $j$  and  $d_{\max}$  represents the maximum transmission range, a link between nodes  $i, j$  exists only if  $d_{ij} < d_{\max}$ . According to this feature, the adjacency matrix  $A$  can be dynamically updated by updating sensor power level after each communication, in order to remove connections associated to dead nodes: if  $A$  is the adjacency matrix of the undirected graph  $V$  of the network, then the matrix  $A_n$  (i.e., the matrix product of  $n$  copies of  $A$ ) has an interesting property: the entry in row  $i$  and column  $j$  gives the number of paths of length  $n$  from vertex  $i$  to vertex  $j$  (which is always finite if there are no directed cycles). Therefore, here the matrix  $A_n$  is defined to be

$$A_{n+1} = A_n \cdot A_1, \quad (13)$$

where, in our particular case, to avoid loops, the main diagonal of every level of the adjacency matrix is forced to have all zero entries, as shown in Figure 5, where all nonzero entries have been indicated with 1.

The generation of feasible multihop routes is therefore straightforward: if the number of nodes is  $N$ , the maximum number of arcs in a route is  $N - 1$ , therefore, knowing that  $A_1 = A$  (updated step by step, as aforementioned), it is just a matter of computing  $A_n$  for all  $n \in [1, N - 1]$ .

At each iteration of GSO, therefore, each individual solution is selected and its “genes” are decoded in a suitable route according to the following chromosome representation, where  $X = (x_1, x_2, \dots, x_M)$  (with  $M = N - 1$ ) is the real-coded chromosome and  $x_m \in [0, 1]$  is the value of each gene, which represents one connected arc in the selected route, as explained with these simple rules: starting from the sender  $i$ , if  $i$  is connected to the destination node  $k$ , a nonzero value is present in the corresponding entry of at least one  $A_{N-m}$

TABLE 2: Fitness function weights.

Fitness function	$\alpha_1$	$\alpha_2$	$\alpha_3$
$F_1$	1	0	0
$F_2$	0	1	0
$F_3$	0	0	1
$F_4$	0.33	0.33	0.33
$F_5$	0	0.5	0.5
$F_6$	0.5	0.5	0
$F_7$	0.5	0	0.5

( $m = 1$  for the first arc). Therefore, a node  $j$  is chosen among those connected to  $i$  (by looking at  $A_1$ ), according to the value of the gene  $x_1$ : for instance, if 3 possible nodes are available for the first arc, the first will be selected if  $x_1 < 1/3$ , the second if  $1/3 < x_1 < 2/3$ , and the third if  $x_1 > 2/3$ ; the process is then repeated for node  $j$ , considering now  $m = 2$  and  $A_{N-2}$ . This iterative procedure ends to node  $k$  in at most  $M = N - 1$  steps, thanks to  $A_n$  features, therefore enabling the creation of consistent routes of variable lengths. Loops can be easily removed by postprocessing the node sequence of the generated route.

Once a feasible path has been identified, the GSO evaluates it associating a fitness score to it. In the evolutionary computation field a fitness function is a particular type of objective function that quantifies the optimality of solutions for a specific optimization problem. The fitness function helps the optimization procedure to select the best candidate solutions in order to properly search for a global optimum. This results in an efficient search among the best possible solutions, avoiding less promising results. Fitness function correlates closely with the algorithm’s goal: in our case the objective is the maximization of the following fitness score  $F$ , defined as a linear combination of  $C_1$ ,  $C_2$ , and  $C_3$ :

$$F = \alpha_1 \cdot C_1 + \alpha_2 \cdot C_2 + \alpha_3 \cdot C_3, \quad (14)$$

where multiple combination of  $\alpha_i$  can be considered. In our numerical simulations, we adopted  $\alpha_i$  values shown in Table 2. The fitness score value  $F$  is defined in the domain  $[0, 1]$ . Therefore the GSO technique is going to find out the best possible route among several ones by considering the above-mentioned criteria, without having to explore the entire graph of all the feasible connections.

## 6. Numerical Results

To validate the effectiveness and the reliability of the proposed technique for optimizing a wireless sensor network in a multihop routing scheme, the sample network of  $N = 25$  nodes shown in Figure 6 has been generated. Some other tests have been performed for networks with a larger numbers of nodes, but we decided to report numerical results for  $N = 25$  essentially to avoid ambiguity in the graphic interpretation. We perform a series of  $N_s = 10$  independent simulations using the same random sequence of senders in order to evaluate and compare the performances of the selected metrics in terms of energy savings and enhancement

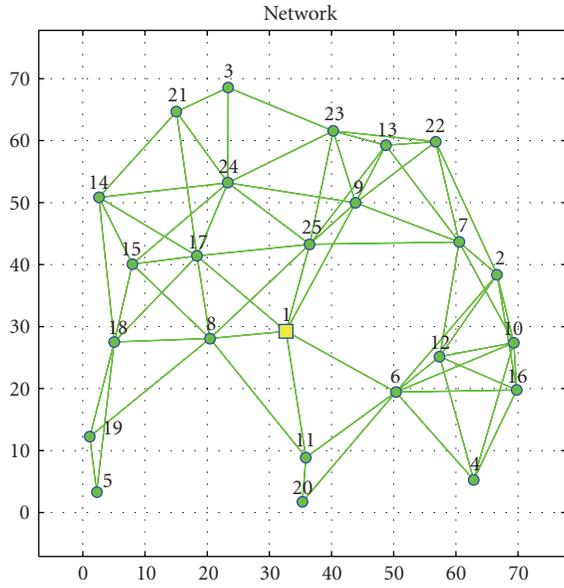


FIGURE 6: The considered sample network of 25 nodes.

of the connectivity of the network. For each sender of the sequence, the GSO performs  $N_i = 100$  iterations in search of the best possible path accordingly with the adopted criteria. Having 10 as the number of individuals in the considered population, the optimization initialization starts with the generation of 10 random solutions (i.e., routes) and the total number of function calls is limited to 1000 per each GSO run.

In this work, the GA engine implemented in GSO uses real encoding of genes with tournament selection on 5 individuals; uniform crossover occurs with probability  $P_{cross} = 80\%$  while the random mutation rate is  $P_{mut} = 5\%$ . For the PSO implementation, we use  $\phi_1 = 2$ ,  $\phi_2 = 2$ , and  $\omega = 0.7$  in the velocity update equation described in [15].

For each sender, an optimal route is defined by a single run of GSO, and the network status and power level of each node are updated consequently. The best route is chosen by GSO according to fitness function which implements the set of rules presented in Section 4 and in [3]. Figure 7 describes the depletion of the nodes batteries, for instance, when the fifth metric ( $F_5$ ) is adopted and is obtained as a mean of the values of each simulation performed using the same criterion. The power level of the sensors is updated after each round of the simulation and, as expected, the first nodes to exhaust their power are some of those positioned close to the base station and in particular nodes 6, 9, 17, 25.

Figures 8, 9, 10, and 11 depict the status of the wireless network, respectively, in terms of percentage of residual power, alive nodes, number of single-hop links, and percentage of nodes connected to the base station. All the reported results are the average values after 10 simulations under the same initial conditions. Figure 8 shows that  $F_2$  allows for saving more power even beyond the  $N = 600$  round, but after that it is outperformed by  $F_5$  and  $F_3$ . As can be seen in Figure 9, the best criterion changes as the number of rounds increases, and while at the beginning almost all the

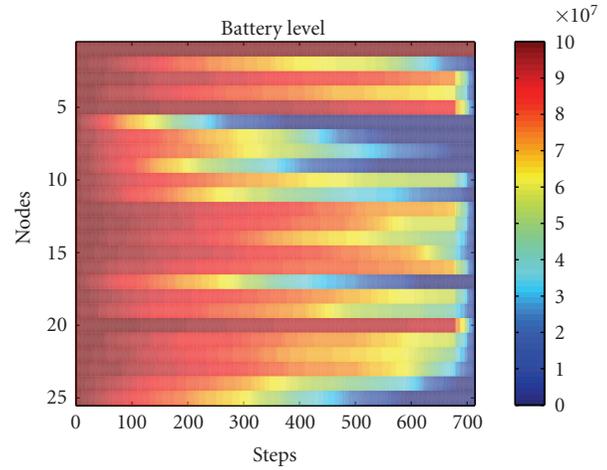


FIGURE 7: Battery level status for each sensor during iterations of the network, optimized considering  $F_5$ , defined in Table 2 (average results over 10 trials).

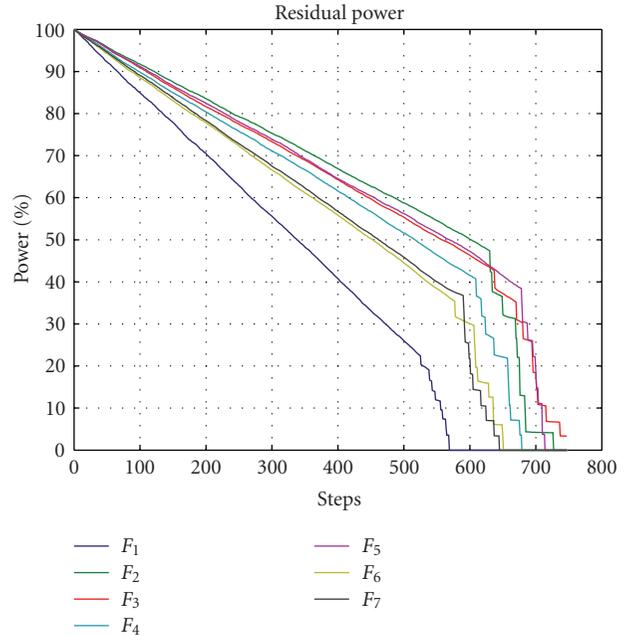


FIGURE 8: Percentage of network residual power during iterations for different criteria, defined in Table 2 (average results over 10 trials).

criteria perform well after almost  $N = 500$  rounds,  $F_4$  assures better results. After almost  $N = 150$  steps, the best criterion changes and becomes  $F_2$ , but only for few rounds, since in the last part of the simulation  $F_5$  and  $F_3$  outperform all the others.

In particular, it is worthwhile to look at the behavior of the third criterion since at the beginning it has the worst performance while it ends up being the second best criterion. The worst performance is obtained when adopting  $F_1$  but that is probably caused by our choice not to evaluate extended paths with a different score (see Section 4).

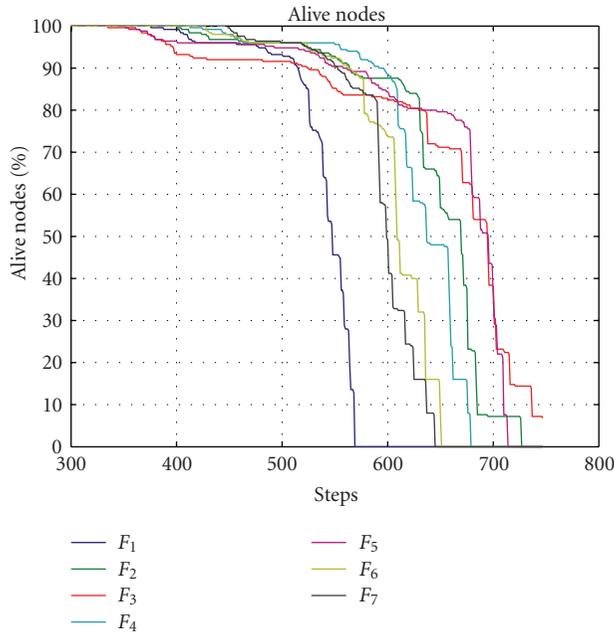


FIGURE 9: Percentage of alive nodes during iterations for different criteria, defined in Table 2 (average results over 10 trials).

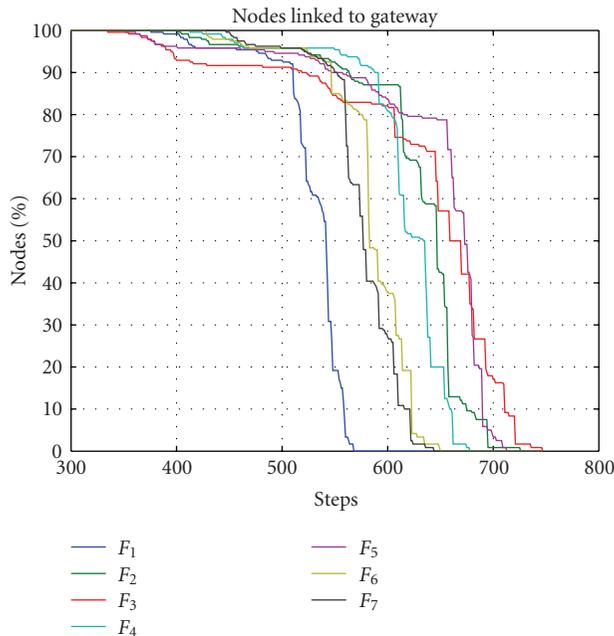


FIGURE 10: Percentage of nodes connected to gateway during iterations for different criteria, defined in Table 2 (average results over 10 trials).

The other figures describe the connectivity of the network and describe almost the same behaviour noticed in Figure 9, underlining an overall major efficiency of  $F_5$  and  $F_3$  with respect to all the other criteria and showing a different best metric at different stages of the simulation. The obtained results show the effectiveness of the proposed approach, which is based on the application of an indirect evolutionary

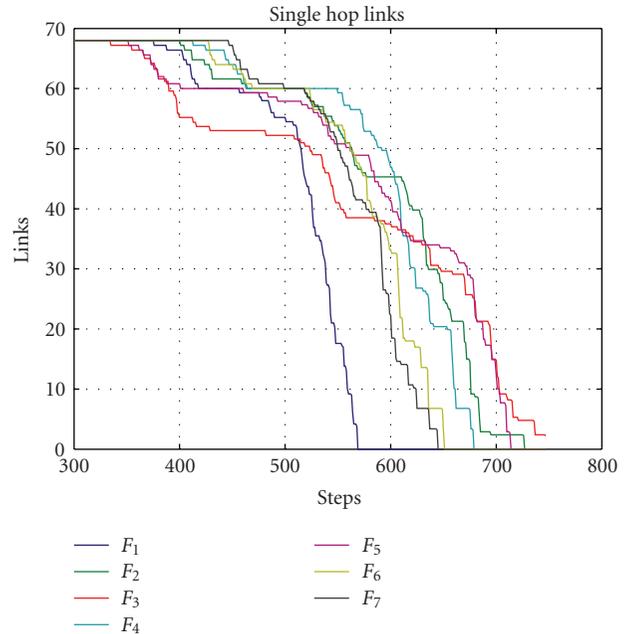


FIGURE 11: Number of single-hop connections in the considered network during iterations for different criteria, defined in Table 2 (average results over 10 trials).

technique in order to apply an optimal path search based on hybrid routing metrics. In fact, the best results are obtained using a hybrid criterion. Moreover, these results suggest that future works will have to take into account variable weights  $\alpha_i$  for the considered objective functions, in order to explore an extended range of possibilities to optimize WSN lifetime.

## 7. Conclusions

Advances in wireless communications and digital electronics not only unfold many possibilities but also introduce challenging issues for the collaborative processing of signals in an energy-constrained distributed sensor network. WSNs are suitable for several novel and existing applications such as environmental monitoring, health care, infrastructure management, public safety, security, transportation, and military strategy.

With the continuing advances in network design, optimization in the communication layer is needed not only to extend the system lifespan, but also to increase reliability and availability as much as possible. It is difficult to reach an optimal solution which takes into account simultaneously different issues such as the reliable extraction of data from distributed sensor nodes, accurate data reporting, and efficient power management to extend sensor network lifetime. In this paper, we propose an evolutionary technique to optimize the lifetime of a sensor network in such an energy-constrained context and adopt a multihop routing scheme.

The optimization algorithm, here specifically arranged to optimize the energy consumption in networks with scalable numbers of nodes, integrates the main features of GA and

PSO into the optimization process, in order to take advantage of the peculiarities of these two classical methods. The principal aim of this work is not to prove the performances of the GSO (which have been already validated in cited papers), but rather to apply it to the routing problem optimization in order to show and underline the effectiveness of the use of hybrid routing metrics that result from the fusion of multiple criteria.

The procedure developed as an indirect method integrates evolutionary capabilities in maximization of WSN lifetime and shows itself effective, fast, and reliable. In particular, the obtained behaviour reflects the ability of GSO to optimize the routes from several nodes to the base station under an energy saving point of view, in order to preserve the network functionality and to avoid premature death of the most solicited nodes. Reported results also suggest further improvements for the proposed technique.

## References

- [1] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [3] D. Caputo, F. Grimaccia, M. Mussetta, and R. E. Zich, "An enhanced GSO technique for wireless sensor networks optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '08)*, pp. 4074–4079, Hong Kong, June 2008.
- [4] Y. S. Ong and A. J. Keane, "Meta-Lamarckian learning in memetic algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 99–110, 2004.
- [5] A. Caponio, G. L. Cascella, F. Neri, N. Salvatore, and M. Sumner, "A fast adaptive memetic algorithm for online and offline control design of PMSM drives," *IEEE Transactions on Systems, Man, and Cybernetics Part B*, vol. 37, no. 1, pp. 28–41, 2007.
- [6] F. Neri, J. Toivanen, G. L. Cascella, and Y. S. Ong, "An adaptive multimeme algorithm for designing HIV multidrug therapies," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 2, pp. 264–278, 2007.
- [7] V. Tirronen, F. Neri, T. Kärkkäinen, K. Majava, and T. Rossi, "An enhanced memetic differential evolution in filter design for defect detection in paper production," *Evolutionary Computation*, vol. 16, no. 4, pp. 529–555, 2008.
- [8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, NY, USA, 1989.
- [9] J. Kennedy, "Particle swarm: social adaptation of knowledge," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '97)*, pp. 303–308, April 1997.
- [10] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 3, pp. 771–779, 2004.
- [11] K. W. C. Ku and M. W. Mak, "Exploring the effects of Lamarckian and Baldwinian learning in evolving recurrent neural networks," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '97)*, pp. 617–621, April 1997.
- [12] J. Robinson, S. Sinton, and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna," in *Proceedings of the IEEE International Symposium on Antennas and Propagation*, vol. 1, pp. 314–317, June 2002.
- [13] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 34, no. 2, pp. 997–1006, 2004.
- [14] A. Gandelli, F. Grimaccia, M. Mussetta, P. Pirinoli, and R. E. Zich, "Development and validation of different hybridization strategies between GA and PSO," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 2782–2787, Singapore, September 2007.
- [15] F. Grimaccia, M. Mussetta, and R. E. Zich, "Genetical swarm optimization: self-adaptive hybrid evolutionary algorithm for electromagnetics," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3 I, pp. 781–785, 2007.
- [16] A. Gandelli, F. Grimaccia, M. Mussetta, P. Pirinoli, and R. E. Zich, "Genetical swarm optimization: an evolutionary algorithm for antenna design," *Journal of Automatika*, vol. 47, no. 3–4, pp. 105–112, 2006.
- [17] J. C. Chen, R. E. Hudson, and K. Yao, "Joint maximum-likelihood source localization and unknown sensor location estimation for near-field wideband signals," in *Advanced Signal Processing: Algorithms, Architectures and Implementations XI*, vol. 4474 of *Proceedings of SPIE*, pp. 521–532, July 2001.
- [18] R. Jurdak, P. Baldi, and C. V. Lopes, "State-driven energy optimization in wireless sensor networks," in *Proceedings of the Systems Communications*, pp. 356–363, August 2005.
- [19] K. P. Ferentinos, T. A. Tsiligiridis, and K. G. Arvanitis, "Energy optimization of wireless sensor networks for environmental measurements," in *Proceedings of the IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA '05)*, pp. 250–255, July 2005.
- [20] K. P. Ferentinos and T. A. Tsiligiridis, "Adaptive design optimization of wireless sensor networks using genetic algorithms," *Computer Networks*, vol. 51, no. 4, pp. 1031–1051, 2007.
- [21] N. Ding and P. X. Liu, "Data gathering communication in wireless sensor networks using Ant Colony Optimization," in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 822–827, August 2004.
- [22] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.
- [23] S. Lindsey and C. S. Raghavendra, "PEGASIS: power-efficient gathering in sensor information systems," in *Proceedings of the IEEE Aerospace Conference*, vol. 3, pp. 1125–1130, 2002.
- [24] F. Ducatelle, G. D. Caro, and L. M. Gambardella, "Ant agents for hybrid multipath routing in mobile ad hoc networks," in *Proceedings of the 2nd Annual Conference on Wireless On-Demand Network Systems and Services*, pp. 44–53, January 2005.
- [25] I. Slama, M. C. Ghedira, B. Jouaber, and H. Afifi, "Cluster based wireless sensor networks' optimization under energy constraints," in *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 745–750, December 2007.
- [26] C. E. Jones, K. M. Sivalingam, P. Agrawal, and J. C. Chen, "A survey of energy efficient network protocols for wireless networks," *Wireless Networks*, vol. 7, no. 4, pp. 343–358, 2001.

- [27] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proceedings of the IEEE Infocom*, pp. 1567–1576, New York, NY, USA, June 2002.
- [28] J. Jia, J. Chen, G. Chang, J. Li, and Y. Jia, "Coverage optimization based on improved NSGA-II in wireless sensor network," in *Proceedings of the IEEE International Conference on Integration Technology (ICIT '07)*, pp. 614–618, March 2007.
- [29] C. Zhao and P. Chen, "Particle swarm optimization for optimal deployment of relay nodes in hybrid sensor networks," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 3316–3320, September 2007.
- [30] <http://www.microstrain.com/wireless-sensors.aspx>.
- [31] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceeding of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 3005–3014, Maui, Hawaii, USA, January 2000.
- [32] R. Dube, C. D. Rais, K-Y Wang, and S. K. Tripathi, "Signal stability-based adaptive routing (SSA) for Ad Hoc mobile networks," Tech. Rep. CS-TR-3646, UMIACS-TR-96-34, August 1996.
- [33] S. Singh, M. Woo, and C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks," in *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 181–190, Dallas, Tex, USA, 1998.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

