# Optimizing Multipath Routing With Guaranteed Fault Tolerance in Internet of Things

Mohammed Zaki Hasan and Fadi Al-Turjman

***Abstract*— Internet of Things (IoTs) refers to the rapidly growing network of connected objects and people that are able to collect and exchange data using embedded sensors. To guarantee the connectivity among these objects and people, fault tolerance routing has to be significantly considered. In this paper, we propose a bio-inspired particle multi-swarm optimization (PMSO) routing algorithm to construct, recover, and select *k*-disjoint paths that tolerates the failure while satisfying the quality of service parameters. Multi-swarm strategy enables determining the optimal directions in selecting the multipath routing while exchanging messages from all positions in the network. The validity of the proposed algorithm is assessed and results demonstrate high-quality solutions compared with the canonical particle swarm optimization (CPSO). Our results indicate the superiority of the multi-swarm and fully PMSO with constriction coefficient, which record an average improvement over CPSO equal to 88.45% in terms of sensors' count, and 89.15% and 86.51% under the ring and mesh topologies, respectively.**

***Index Terms*— Fault tolerant, Internet of Things, multi-swarm, multipath routing.**

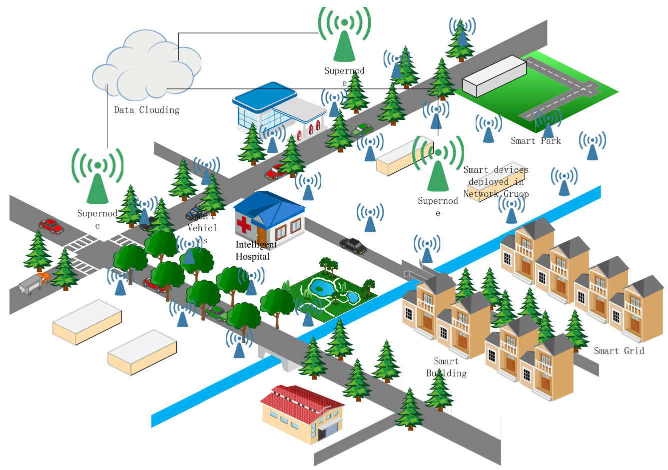


Fig. 1. The devices deployed in IoT.

## I. Introduction

WIRELESS Sensor Networks (WSNs) form the basis of the future network "Internet of Things (IoTs)". IoT is described as show in Fig. 1 as an exceptionally complex network model where varieties of components are deployed as consumer electronic devices which interact in a complex way with each other [1]. However, these devices operate under strict energy constraints making the dedicated energy budget for fault tolerant routing very limited [2]. Fault tolerant routing problem has received a significant attention in the literature [3]. We believe that the emergence needs for IoT applications such as smart homes, smart cities, healthcare, and etc, will further increase the importance of fault tolerance in various aspects, due to the required constant mode of operation and therefore special effort have been placed to develop fault tolerance in routing [3].

Often WSNs operate in an autonomous mode without a human supervision in the loop [4]. Moreover, sensor nodes are often deployed in uncontrolled and sometimes even hostile environments. Therefore, it is difficult to accurately predict the optimal way to treat fault tolerance within a particular WSN routing approach, since both technology and envisioned applications for WSNs and IoTs are changing at a rapid pace. Since the available communication energy is significantly lower than the computation energy, then it is important to develop fault tolerance routing algorithms so as to recover from path failure that will require only a limited amount of communication regardless of any additional computational energy. Otherwise, any unpredictable events may cause the devices to fail, partitioning the network and disrupting the network functions.

Basically, multipath routing protocols provide tolerance to failures and increase the network reliability [5]. Solving the fault tolerant routing problem is often formulated as multi-objective optimization problem (MOP) to establish not only for single-hop or path connectivity but also $k$-disjoint paths that guarantee the connectivity even after the failure of up to $k-1$ paths. To have a more realistic analysis of our fault model we formulate strong fault tolerant routing problem as MOP in which are treated simultaneously subject to a set of constraints.

Strong fault tolerance requires enormous computational efforts, which induce large control message overhead and a lack of scalability as the problem size increases [6]. Solving these problems on individual sensor node may require extreme memory and computational resources and yet produces average results [7]. We develop a bio-inspired particle swarm optimization (PSO) routing algorithm to provide fast recovery from path failure. Owing to its simple concept and high efficiency, PSO has been actively utilized in these problems with promising results [8]. Nevertheless having competitive performance, precisely solving the fault tolerant routing problem still remain a challeng for PSO due to premature convergence issue. Unfortunately most premature convergence traps caused

Manuscript received July 24, 2017; accepted August 9, 2017. Date of publication August 14, 2017; date of current version September 8, 2017. The associate editor coordinating the review of this paper and approving it for publication was Dr. Roozbeh Jafari. *(Corresponding author: Fadi Al-Turjman.)*

The authors are with the Department of Computer Engineering, Middle East Technical University, Northern Cyprus Campus, 99738 Mersin, Turkey (e-mail: mohammed.z.hasan@ieee.org; fadi@metu.edu.tr).

Digital Object Identifier 10.1109/JSEN.2017.2739188

by the rapid convergence characteristic and the diversity loss of the particle swarm, thereby leading to find undesirable quality solutions [9]. We face another challenge which is the ability to balance between exploration and exploitation searches. Neither exploration nor exploitation should be overemphasized as exploration inhibits the swarm convergence, whereas the exploitation tends to cause the particle swarm to hastily congregate without the feasible region that leads to premature convergence [10]. Motivated by these challenges especially the actual connectivity model inside the WSN that has been intergraded into IoT, we propose a new two routing algorithms namely fully particle multi-swarm (FPMS) and canonical particle multi-swarm (CPMS) which based on multi-swarm computationally efficient alternative to analytical methods to tolerate the failure of multipaths with reconstructed multipath that satisfy the quality of service (QoS) parameters in terms of energy consumption, delay, and throughput. To summarize we make the following contributions:

- We develop a bio-inspired particle swarm optimization (PSO) routing algorithm to provide fast recovery from path failure by make an effort to extend an existing approach for finding optimal solution for MOP. We define the objective functions and then optimize the effective values of these objective functions which computed at each sensor node selected to construct $k$-disjoint multipath.
- We investigate the performance of proposed multipath routing algorithms by compare against existing optimization algorithms namely the canonical particle swarm optimization (CPSO) [11], [12] to provide an alternative learning strategy for particles.

These algorithms are very similar to each other and only differ in a few things related to convergence, exploitation, exploration, and jumping out of the basins of attraction in optimal solutions. Overheads related to fault-tolerance have been also part of these algorithms. That is because by recovering from failures, we implicitly increase the number of in-network paths which entails more messages to be exchanged, and thus, encountering unnecessary communication overhead [13]. Therefore, we employ a complex network connectivity representation for the swarm topology population and adopt a multipath routing algorithm that balance the trade-off between fault-tolerance and communication overhead. This algorithm takes the advantage of combining proactive and reactive routing mechanisms in order to maintain and record the most demanding information on every particle in the form of an objective function value for the selected path. Then the particle are adaptively increased or decreased and connected with their matching velocity to make a proper selection by considering the optimized objective function. The performances of each objective function in terms of energy consumption, average delay, and throughput using the multi-swarm and fully particle multi-swarm optimization with constriction coefficients are evaluated and assessed.

Rest of the paper is presented as followed. Section II introduces the related works. Section III introduces the main concepts of the assumed system models. Section IV presents the proposed mechanism. Section V compares and discusses the achieved results. Finally, we conclude our paper in Section VI.

## II. Related Work

Fault-tolerant routing is usually used to provide availability, reliability and consequent independency in the network [3]. Moreover, fault-tolerant routing is important for IoTs applications since WSN's devices are attached to the environment with several QoS requirements for monitoring, and tracking vital events. There are several approaches for achieving fault tolerance in WSNs, but the most popular approach is the multipath routing since this approach is employed at high levels as one of the possible solutions with limited network resources [14]. Current studies involve different types of optimization such as meta-heuristic strategies to employ multipath routing [15]. Authors in [8] solve the fault-tolerant routing problem by constructing multipath routing to provide fast recovery from path failure. In [16] the authors propose an energy-aware multipath routing scheme based on PSO to find optimal loop-free path in order to solve the disjoint multipath problem in a Mobile Ad hoc Network (MANET). The authors in [17] present PSO to construct an appropriate path load scheduling to distribute routing load over selected paths. Meanwhile, the authors in [18] propose an Enhanced PSO-based Clustering Energy Optimization (EPSO-CEO) scheme for minimizing the power consumption of each node by constructing clusters in a centralized manner and optimize the selection of the cluster head. However, none of these mentioned studies discuss how to formulate objective function with constraints which can lead to a convergent solution as proposed in our routing algorithm.

The difference between these studies and our work is that we try to find optimal multipath to rout the data in two-tiered heterogeneous WSNs with multi-objective function to minimize nodes' total transmission, average delay and maximize throughput, whereas other works focus only on flat homogenous sensor network topology. Additionally, we focus on connectivity between sensor nodes and other components in network model whereas we focus on $k$-disjoint multipath to route from sensor nodes to sink node which can tolerate at least $k-1$ network failure. However, these studies do not employ $k$-disjoint multipath between sensor nodes and thus they do not guarantee fault-tolerance in case $k-1$ network failure.

Meta-heuristic in WSNs has to configure their own network topology, localize, synchronize, and calibrate; coordinate internode communication; and determine other important operating parameters. Therefore, we define a neighborhood relationship among sensor nodes according the exchanged information with each other. Indeed this motivation of using this mechanism collects the information of network topology for exploitation and converges towards finding optimal configuration, whereas the sensor nodes would help to keep diversity up. Studies such as [19] and [20] proposed a concept of diversity to avoid the premature convergence of swarm by setting a lower and upper bound of search space to ensure the swarm has a good search ability in finding optimal solutions for varying real-world applications. In [21] the authors introduce the performance of load distribution model as optimization problem in order to facilitate optimal network selection. The authors take into consideration both the network bandwidth and errors to determine the optimal load distribution among heterogeneous networks with minimal system cost. The authors in [22] aim to improve the fault tolerance and cluster lifetime of the hybrid energy-efficient distributed clustering protocol by introducing the non-probability waiting time, residue energy and central points. Meanwhile, some researchers focus on changing parameters of PSO such as inertia weight and acceleration coefficient of the updated velocity in order to improve the performance in comprehensive learning particle swarm optimizer (CLPSO) for multimode problems in [23]. Indeed, CLPSO suffers a slow resolution, however, the authors adapt the CLPSO algorithm by improving the search behavior to optimize the continuous solutions. Whereas the authors in [24] improved CLPSO algorithm by introducing a new adaptive parameter strategy. The algorithm evaluates the objective functions of individual particles and the whole swarm, based on which values of inertia weight and acceleration coefficient are dynamically adjusted to search more effectively. However, none of these mentioned studies discusses how to formulate MOP with constraints. Non of them uses this MOP to share the personal-best information for fault-tolerant routing.

## III. System Models

The proposed routing model employ fault-tolerant topology control in two-tiered heterogeneous WSNs consisting of resource-rich supernodes and simple sensor nodes with batteries of limited capacity and unmitigated QoS constraints. This mixed deployment of the two-tiered heterogeneous can provide a balance of performance and cost of WSN [25]. We consider a many-to-one traffic pattern where supernodes and simple sensor nodes are able to ensure the required connectivity degree. In the following we give the necessary definitions for our used IoT system models:

*Definition 1: n paths are said to be energy-node-disjoint* $\leftrightarrow$ *they have no common nodes.*

Every node-disjointness has to be used in building the network topology with $k$-disjoint multipath routing in order to increase the number of alternative paths and therefore the network become fault tolerant. However, to obtain strongly fault-tolerant network topology we consider a topology by construction of $k$-disjoint multipath to guarantee that a node remains connected to the sink even after the failure of up to $k-1$ paths. This leads to strong fault tolerance since a node failure may influence only one path which is a major challenge for these sensor nodes deployment. Our model is based on the observation that a node can connect and/or disconnect the links with neighbors that are not on one of the $k$-disjoint multipaths from the node to one of the supernodes. This needs to determine which neighbors are on one of such multipaths and which are not.

*Definition 2: n paths are said to be node-disjoint* $\leftrightarrow$ *they have no common nodes.*

*Definition 3: a WSN is k-vertex supernode connected if the removal of any* $k-1$ *sensor nodes does not partition the network.*

*Definition 4: Given a node* $v_\iota$*, that* $v_J$ *is the next hop toward the destination node, and forwarding selection rule F is associate the node* $v_\iota$ *with another node* $v_J$ *in* $|V|\backslash\{v_\iota\}$*, in such a way that the path* $P(v_{source}, v_\iota, v_J), \ldots, v_{destination}$*. obtained by applying the rule from source to destination.*

During adding a new node to the network or in case of failure recovery, the following fault-tolerance model is used.

### A. Fault Tolerance (FT) Model

In this section, we characterize the $FT$ parameters by exploiting the synergy between fault detection and $FT$ in IoT. The $FT$ parameters are leveraged based on Markovian model via the coverage factor and sensor failure rate [26].

*1) Coverage Factor:* The coverage factor $c$ is defined as the probability that the faulty active sensor in an IoT paradigm is correctly diagnosed, disconnected, and replaced by a good inactive spare sensor. The $c$ estimation is critical in an $FT$ IoT model and can be determined by:

$$c = c_k - c_c, \tag{1}$$

where $c_k$ denotes the accuracy of the fault detection in diagnosing faulty sensors and $c_c$ denotes the probability of an unsuccessful replacement of the identified faulty sensor with the good spare sensor. Whereas $c_c$ depends on the sensor switching circuitry and is usually a constant, $c_k$'s estimation is challenging because different fault detection approaches have different accuracies.

We analyzed the related work in the literature and observe that the accuracy of a fault detection algorithm depends on the average number of sensor node neighbors' $k$ and the cumulative probability of sensor failure $p$. Accordingly, we model $c_k$ as follows:

$$c_k = \frac{(k(1-p))}{k^{(\frac{k}{M(p)})^{1/M(p)}} + (1 - \frac{k}{M(p)})^k} \tag{2}$$

where $c_k \leq 1$ and $M(p)$ is a function of $p$ representing an adjustment parameter that may correspond loosely to the desired average number of neighboring sensor nodes required to achieve a good fault detection accuracy for a given $p$. To clarify further, we point out that our Markov models are independent of $c_k$'s determination methodology and equally applicable to any $c_k$ value.

*2) Sensor Failure Rate:* The sensor failure rate can be represented by exponential distribution with a failure rate of $\lambda_s$ over the period $t_s$ [27]. The failure rate curve approximation by piecewise exponential distributions is analogous to a curve approximation by piecewise straight-line segments. Accordingly, the exponential model works well for those inter-arrival times where the total number of events in a given time period is given by the Poisson distribution. Consequently, the Cumulative Distribution Function (CDF) for the sensors with an exponentially distributed failure rate can be represented by:

$$F_s(t_s; \lambda_s) = p = 1 - \exp^{(-\lambda t_s)}, \tag{3}$$

where $p$ denotes the cumulative probability of sensor failure and $t_s$ signifies the time over which $p$ is specified.

Solving Eq. 3 for $\lambda_s$ gives:

$$\lambda_s = -(\frac{1}{t_s}) \ln(1-p) \tag{4}$$

### B. Network Model

According to Definition 1, Definition 2, Definition 3, and Definition 4 the node-disjointness relations are modeled as a directed graph $G(V, E)$, where $|V| = \{v_1, v_2, \ldots, v_N, v_{N+1}, \ldots, v_{N+M}\}$ is a finite number of nodes i.e. particles, we advise to note both particle and sensor nodes will be interchangeable names for the same discipline in whole paper. Therefore, $N$ denote sensor node and $M$ denote supernodes. $\varepsilon$ is the matrix to defined set of $k$-disjoint multipaths from the source-destination nodes bypassing to one of the supernodes. The relation between a pair of nodes is the number of edges $E$ in $G$ which is the set of paths, whereas $E = \{(v_\imath, v_\jmath)|Hop(v_\imath, v_\jmath) \leq \tau\}$, where $\tau$ is transmission power range of node and $Hop(v_\imath, v_\jmath)$ depicts the distance between $v_\imath$ and $v_\jmath$. We indicate with $v_\jmath$ with forwarding selection rule $F$ that select the next hop of node $v_\imath$ toward the $v_d$ with $\tau$ according to mechanism of selection $F$. Therefore, $\varepsilon$ induces multipaths among any possible connection between source-destination pair in the network. Thus,

$$F : \varepsilon \longrightarrow k^{sd}_{v(\imath,\jmath)}(\varepsilon) \tag{5}$$

where $k^{sd}_{v(\imath,\jmath)}(\varepsilon) = 1$ iff the connection between node $\imath$ and $\jmath$ is part of the path between source and destination node. Moreover, a path $P(v_\imath, v_\jmath)$ from vertex $v_\imath$ to vertex $v_\jmath$ in a graph $G$ is a sequence of edges that are traversed when going from $v_\imath$ to $v_\jmath$ where $\imath \neq \jmath = 1, 2, \ldots, N + M$. Therefore is defined as set of alternative paths $p(v_\imath, v_\jmath)$. $E(v_\imath, v_\jmath \in p(v_\imath, v_\jmath))$ represents a node-disjoint between in $p(v_\imath, v_\jmath)$ and $(v_N, v_{N+M})$, whereas $e(e \in p(v_\imath, v_\jmath), (v_N, v_{N+M}))$ represents direct link between two any nodes. Thus, we can get the $k$-disjoint path in $G$, considering the QoS parameters affecting the selection mechanism of the optimal multipaths include energy consumption, delay and throughput. These parameters are used to evaluate the objective function of the selected multipaths, and we use the derivation on [28] and [29] to solve the objective functions that minimize both energy consumption, and average delay, and maximize total throughput. Table I lists all notations which have been used throughout the paper.

### C. Problem Formulation

We aim to construct a $k$-disjoint multipath routing in fault-tolerant topology network topology to route collected by sensor nodes to the supernodes for two-tiered WSNs with the aforementioned network model. Hence, we model the topology control as a transmission range assignment problem for each sensor node in the network. The objective is to minimize the assigned QoS parameters in terms of transmission power range and the average delay for all sensors while maintaining $k$-disjoint multipaths from each sensor to the set of supernodes to determine the optimal multipaths. In this topology, each sensor node in the network must be connected to at least one supernode with $k$-disjoint multipaths to exchange information with each other. We can define the problem as follows: given

TABLE I

NOTATION

| Symbol | Quantity |
|---|---|
| $N$ or particle | Sensor nodes |
| $M$ | Supernodes |
| $\tau$ | transmission range of node |
| $Hop(v_\imath, v_\jmath)$ | Distance between indicating two nodes $v_\imath$ and $v_\jmath$ |
| $c$ | Coverage factor |
| $c_k$ | Accuracy if fault detection |
| $c_c$ | The probability of an unsuccessful replacment of the identified faulty sensor with good spare one |
| $p$ | Adjustment parameter |
| $\lambda_s$ | Failure rate |
| $t_s$ | Specific time period |
| $\varepsilon$ | The definition matrix of $k$-disjoint multipaths |
| $F$ | Forwarding rule to select next hop of node |
| $k$ | Connectivity indicator between node $\imath$ and node $\jmath$ |
| $P(v_\imath, v_\jmath)$ | The path in network |
| $\aleph_{\imath,\jmath}$ | The set of nodes disjoint of the $k$-disjoint path |
| $P^o_v$ | Operational power of node |
| $d^\alpha_{v(\imath,\jmath)}$ | Distance between node $\imath$ and node $\jmath$ |
| $L_p$ | Bits of data frame |
| $\lambda^a_v$, $\lambda^b_v$, and $\lambda^\mu_v$ | The time duration of data acquisition , processing and data packet transmission time |
| $P_{min}$ and $P_{max}$ | Lower and upper energy constraints value bound of selected path |
| $\wp(\xi_\imath, \xi_\jmath)$ | Delay of routing data packets from source to destination |
| $L^\wp_e$ | Hop delay requirement along the path from the source to the sink |
| $\Delta\wp$ | Bounded delay |
| $f$ | Objective function |
| $Z$ | Set of feasible solutions |
| $B$ | Bandwidth |
| $p_{best}$ | Personal-best position |
| $g_{best}$ | Global-best position |
| $\phi_1$ | Personal best coefficient |
| $\phi_2$ | Neighbor best coefficient |
| $\chi$ | Constriction coefficient |
| $v$ | Velocity toward selecting optimal solution |

a $k$-disjoint multipath constructed by connecting a group of supernode and energy-constrained sensor nodes that can adjust their transmission range up to a predefined optimal value. The transmission range of each sensor is set such that the total transmission power is minimized and the resulting topology is still $k$-disjoint multipath connected in order to assure ensure QoS satisfaction.

### D. Energy Model

To obtain the appropriate constraints value we depend on the determination of two variables; the number of hops and the intermediate distance between two sensor nodes along the selected path, where $\tau_{v(\imath,\jmath)}$ is the distance from one node to the next hop node. To define a neighborhood relationship among the nodes as mentioned in section III, neighborhood topology is used. Determining the next hop is achieved via exploit the nearest neighbor, and therefore each sensor node have transmission range that can communicate with the neighbors which are within that range. Suppose that the transmission range is given by $\tau > 0$, then representing its next neighborhood is given by

$$\aleph_{\imath,\jmath} = \{\imath, \jmath \neq \imath | \parallel v_\imath - v_\jmath \parallel \leq \tau_v\} \tag{6}$$

Here $\aleph_{\imath,\jmath}$ denotes the set of nodes disjoint of the $k$-disjoint path. Please note that this can be vary during transmission and reception of information. It might lead to disconnect several neighbors and partition multipaths, unless the constraints are satisfied. Hence, these constraints can be considered as a dynamic point of the objective function which might be in the feasible region. Therefore, these constraints can extremely change the topology connectivity degree which can be used to solve the objective function of energy consumption as the optimization process evolves. The lower bound and upper bound values of number of hops and transmission range determined by [28] according to method for determination of cut-off method values. Cut-off method for integer programming (IP) problem as referred in [28] and [30] is closely related to certain global optimization problem. There are several modifications and combinations methods of cut-off with other methods were introduced, such as lines searches, and quadratic approximation [31]. Main focus challenging question with cut-ff methods has always been the question of constraint dropping strategies. Since, in each iteration a new constraints is added to the existing set of constraints but no constraints is ever deleted, therefore the size of the problem to be solved increases from iteration to iteration. Thus, we depend on determination of lower and upper bounded for constraints as in the work reported [28] whereas the number of constraints in each iteration is bounded. Further the energy dissipation to run the transmitter and receiver circuitry is denoted as the operational power $P^o_\upsilon$. We assume that

$$P^{trans}_\upsilon = P^{rec}_\upsilon = P^o_\upsilon \tag{7}$$

Meanwhile, the transmit power level that should be assigned to sensor node $\imath$ to connect sensor node $\jmath$ with acceptable signal to noise ratio (SNR) is denoted as is $P^t_\upsilon = \varepsilon_{mp} d^\alpha_{\upsilon(\imath,\jmath)}$, where $\alpha$ energy loss due to channel transmission under the assumption that the WSN is relatively free of obstacles where $d^\alpha_{\upsilon(\imath,\jmath)}$ is distance between sensor node $\imath$ and sensor node $\jmath$ and $\tau_{\upsilon(\imath,\jmath)}$ is transmission range. Hence, the overall expression for power consumption simplifies to

$$P^t_\upsilon = 2P^o_\upsilon + \varepsilon_{mp} d^\alpha_{\upsilon(\imath,\jmath)} \tag{8}$$

Whereas $d^\alpha_{\upsilon(\imath,\jmath)} = \tau_{\upsilon(\imath,\jmath)}$ is sensor $\imath$'s transmission range, $s.t.\ \tau_{max} \geq \tau_{\upsilon(\imath,\jmath)} \geq \tau_{min}$, where $\tau_{max}$ is a fixed maximum communication distance, which is considered by maximum power that sensor nodes can transmit that is $P_{max}$ while the $\tau_{min}$ is a minimum communication distance which consider by minimum power that sensor node can transmit that is $P_{min}$. Therefore, the optimal hop theoretical hop number is obtained as an integer number from

$$HOP = \sqrt[\alpha]{\tau_{\upsilon(\imath,\jmath)}\left(\frac{3\varepsilon_{mp} d^\alpha_{\upsilon(\imath,\jmath)}}{2P^o_{\upsilon_{(\imath,\jmath)}}}\right)} \tag{9}$$

The total energy consumption of given path at specific interval period $\lambda$ is given

$$Energy_{\upsilon_{sd}}(\lambda) = L_p\{\sum_{\upsilon_{(\imath,\jmath)}=0}^{N+M} (\lambda^a_\upsilon + \lambda^b_\upsilon)P^o_\upsilon + \lambda^\mu P^t_\upsilon\} \tag{10}$$

Where $\lambda^a_\upsilon$, $\lambda^b_\upsilon$, and $\lambda^\mu_\upsilon$ indicate the time duration of data acquisition, processing and data packet transmission time that taking at sensor node $\imath$, respectively. We can assume the energy cost for each path $P(\upsilon_\imath, \upsilon_\jmath)$ can be expressed as

$$Energy_{(\aleph)}(\lambda) = \sum_{(s,d)\in\Pi_{P(\upsilon_\imath,\upsilon_\jmath)}(\aleph)} L_p\{(\lambda^a_\upsilon + \lambda^b_\upsilon)P^o_\upsilon + \lambda^\mu P^t_\upsilon\} \tag{11}$$

with

$$\Pi_{P(\upsilon_\imath,\upsilon_\jmath)}(\aleph) = \{(s,d)\ s.t.\ k^{sd}_{\upsilon(\imath,\jmath)}(\varepsilon) = 1\ for\ at\ least\ one\ \jmath\} \tag{12}$$

The set $\Pi_{P(\upsilon_\imath,\upsilon_\jmath)}(\aleph)$ defined binary constraints that ensure all links exists between the source-destination pair as well as the link exits between the two sensor nodes $\imath$ and $\jmath$. Suppose, transmission by sensor node can be received by all sensor nodes within its transmission range; that is $\iff$ there is a connection between $\imath$ and $\jmath$, and if $d^\alpha_{\upsilon(\imath,\jmath)} \geq \tau_{\upsilon(\imath,\jmath)}$. Then, the connectivity constraints is:

$$d^\alpha_{\upsilon(\imath,\jmath)} = \tau_{\upsilon(\imath,\jmath)} = \begin{cases} 1 & \text{if a connection exist between a node } \imath \text{ and } \jmath \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Finally, the objective function for minimizing the energy spent by a node to transmit a data packet of length $L_p$ bits a distance $\tau$ is given by:

$$\min Energy_{\upsilon_{sd}}(\lambda) \tag{14}$$

subject to

$$d^\alpha_{\upsilon(\imath,\jmath)} = \tau_{\upsilon(\imath,\jmath)} \tag{15a}$$

$$\tau_{\upsilon(\imath,\jmath)} \leq HOP \tag{15b}$$

$$\tau_{min} \leq \tau_{\upsilon(\imath,\jmath)} \leq \tau_{max} \tag{15c}$$

$$2 \leq \alpha \leq 4 \tag{15d}$$

$$\sum_{\upsilon_{(\imath,\jmath)}} d^\alpha_{\upsilon(\imath,\jmath)} \leq HOP \tag{15e}$$

This value of energy of the selected path can adjust their transmitting power within a closed interval of lower and upper value bound such as $[P_{min}, P_{max}]$, where $P_{min}$ and $P_{max}$ determining the minimum and maximum constraints values of the path.

### E. Delay Model

The definition of delay depend on the optimal hop number which could have different delay guarantees, denoted as $\wp(\xi_\imath, \xi_\jmath)$. Increased number of nodes results in more paths becoming available for simultaneously routing packets to their destinations which is beneficial for reducing the delay. Meanwhile, this may also increase proportionally to number of nodes on the invoked path. In order to calculate the delay among multipaths from source to destination, we considered nodal's load as well as delays of total paths. This depends on handling packet loss during initial network topology as well as data routing. Each supernode receive ACK message from its neighbors indicating that it is ready for transmitting and receiving. Otherwise, supernodes keep retransmitting until they receive an ACK message, for a specified period of time (user parameter). After the time expires, the supernodes proceed with passing ACK message to next node. Then supernodes select its next hop based on optimal hop-distance, updates and topology changes. Therefore, suppose, determine the optimal number of hops as in Eq. 9 that minimizes the delay of the successful transmission of a packet, then jointly optimize the hops and the estimation of delay constraint to derive a scaling for minimizing the delay. Hence, to solve the optimization problem, all the source nodes and the intermediate nodes periodically calculate the delay when generated from the one-hop neighborhood of each node, because the one-hop is easier to acquire [28]. Suppose that one QoS requirement is be satisfied at each hop, then the end-to-end QoS requirement is also met [28]. More precisely, a node can satisfy the hop requirement by selecting the next hop. This allows the bounded delay to be evenly divided at each hop. The end-to-end delay between any two sensor nodes $\xi_{Source}$ and $\xi_{Destination}$ over the set of paths $P$ is given by

$$\wp_{sourcessink}(L_p) = \min\{\sum_{\xi_\imath} \wp(\xi_\imath, \xi_\jmath)\}, \tag{16}$$

where $\wp_{SourcesDestination}$ is the minimum achievable delay when the generated data are routed along the set of paths between $\xi_{Source}$ and $\xi_{Destination}$. The delay $\wp(\xi_\imath, \xi_\jmath)$ between two nodes is the time required to successfully transmit a packet after the first node receives it. This time might include queuing, contention, transmission, retransmission, idle, propagation, load, and processing. The mean delay of each sensor node can be computed as

$$\begin{aligned}\xi = {} & D_{quening} + D_{propagation} + D_{processing} \\ & + D_{transmission} + D_{retransmission} \\ & + D_{load} + D_{idle}\end{aligned} \tag{17}$$

subject to

$$\sum_{\upsilon=1}^{N+M} \wp(\xi_\imath, \xi_\jmath) \leq \Delta\wp, \tag{18}$$

where $\Delta\wp$ is the bounded delay, which depends on two factors: the number of hops taken and the delay of a node, which are of additive form and denoted as $\eta_{\imath\jmath}$ and $\wp^e$, respectively. Therefore,

$$\Delta\wp = \wp_0^{Source} + \wp_{\eta_1}^{\xi+1} + \wp_{\eta_2}^{\xi+2} + \ldots + \wp_{\eta_{N+M}}^{Destination} \tag{19}$$

$L_e^{\wp}$ is the hop delay requirement along the path from the source to the sink, which is composed of $\eta_\imath$ and depends on the partition requirements at sensor node $\xi_\imath$. The hop delay requirement is equal to

$$L_e^{\wp} = \frac{\Delta\wp - \wp^e}{\eta_\imath} \tag{20}$$

Then rewrite the constraint

$$\sum_{\upsilon=1}^{N+M} \wp(\xi_\imath, \xi_\jmath) \leq L_e^{\wp}, \tag{21}$$

*F. Throughput Model*

Each sensor has a maximum transmitting power $P_{max}$ and a maximum bandwidth. Bandwidth denoted as $B$ is defined as the sum of total of all transmitted and received loads, i.e. their sum should not exceed the bandwidth capacity of the node. We use the defined of throughput in referred in [29] as the amount of data packets successfully transmitted. Therefore, the total amount of data packet successfully transmitted the optimal number of hops is calculated as

$$\begin{aligned}Throughput = {} & (\frac{D_{transmission} + D_{retransmission} + D_{load}}{\xi}) \\ & * Tx_{datarate}\end{aligned} \tag{22}$$

The objective function for maximizing the throughput for all the outgoing packet at the selected path is given by:

$$\max Throughput_{\upsilon_{sd}}(\lambda) \tag{23}$$

subject to

$$\sum_{(s,d)} \aleph_{s,d} \leq B \tag{24}$$

## IV. Convergence Behavior of Different Particle Swarm Optimization Algorithm

The $k$-disjoint multipath algorithm assigns each sensor particle/node with transmission range level according to hop-distance as referred in Eq. 9 for each neighbors to search on diversity characteristics of particle swarm. Each node has ability to improve cooperative learning behavior by exchanging the messages with their neighbors. Upon receiving these messages, each node computes the disjoint paths and further the local path information updates according to constraints as referred in Eqs. 15, 18 and 24. According to calculation of the objectives functions in Eqs. 14, 16, and 23 a new potential paths will generate, construct, select and maintain, therefore, nodes that select adaptively with a velocity $\upsilon_{(\imath,\jmath)}$ that is updated every iteration to satisfies the right direction of selected paths.

Assume $Z_\imath = (z_1, z_2, z_3, \ldots, z_m)_\imath$ is the feasible solution, and $Z = (z_1, z_2, z_3, \ldots, z_w)$ is solution space of multi-objective problem with $k$-disjoint multipath has $m$ variables values, the position and velocity of particle $\upsilon$ represented by $m$ dimensional vector $|V| = \{\upsilon_1, \upsilon_2, \ldots, \upsilon_N, \upsilon_{N+1}, \ldots, \upsilon_{N+M}\}$ and $\nu = \{\nu_1, \nu_2, \ldots, \nu_N, \nu_{N+1}, \ldots, \nu_{N+M}\}$. Two positions, named personal-best position $p_{best}$ and global-best $g_{best}$ are defined in proposed algorithm.

During solving the multi-objective functions Eqs. 14, Eq. 16 and Eq. 23 in terms of energy consumption, average delay, and throughput respectively. The nodes will connected in the whole searching range in each iteration to generate arbitrary feasible solutions i.e. paths of each objective function. However, solution $Z_1$ may be dominate $Z_2$ and/or $Z_3$ if and only if $\{\forall_\imath = 1, 2, \ldots, n, f_\imath(Z_1) \leq f_\imath(Z_2)\} \wedge \{\exists\jmath = 1, 2, \ldots, n, f_\jmath(Z_1) \leq f_\jmath(Z_2)\}$; this is denoted $Z_1 \succ Z_2$. These nodes are influenced by individual exchange message which denoted as extreme value point and global extreme value point, which lead to the nodes to select next hop toward extreme value point within the scope of the search space in each iteration process. We indicate with $\upsilon_N = d_\upsilon^F(\upsilon_{N+1}), (p_{best(\upsilon_N)})$ or $\upsilon_N = d_\upsilon^F(\upsilon_{N+1}), (g_{best(\upsilon_{N+M})})$ is the next hop of node $\upsilon_\imath$ towards $\upsilon_\jmath$ with extreme value of either $p_{best}$ or $g_{best}$ within $\tau_{(\imath,\jmath)}$ according $F$ rule of updating the velocity as referred in Eq. 25 and 27. Therefore, the nodes will deviate from the constraints domain and hard to converge to the extreme value point of constraint domain. The whole personal-best positions of the swarm imply the distribution of good objective functions as referred Eqs. 14, 16 and 23 in that related to exchange messages under satisfying constraints in Eqs. 15, 18 and 24, that use to update the values of velocities and then select the optimal multipath route. In each path the personal-best position of particle $\upsilon_{(\imath,\jmath)}$ denoted as $p_{best(\upsilon_{(\imath,\jmath)})} = (p_{best(\upsilon_1),best(\upsilon_2),\ldots,best(\upsilon_N),best(\upsilon_{N+M})})$ and global-best position denoted as $g_{best(\upsilon_{(\imath,\jmath)})} = (g_{best(\upsilon_1)}, best(\upsilon_2), \ldots, best(\upsilon_N), best(\upsilon_{N+M}))$.

The degree of influence the personal-best position $p_{best}$ defined by coefficient of constraints $\phi_1$. Likewise the influence of the best global $p_g$ is defined by coefficient of constraints $\phi_2$. The velocity update function, which drives the CPSO, is defined mathematically.

$$\overrightarrow{\upsilon_\upsilon} := \chi \cdot (\upsilon_\upsilon \tag{25a}$$
$$+ \overrightarrow{Z}(0, \phi_1) \bigotimes (\overrightarrow{p_\upsilon} - \overrightarrow{\upsilon_{N+M}}) \tag{25b}$$
$$+ \overrightarrow{Z}(0, \phi_2) \bigotimes (\overrightarrow{g_\upsilon} - \overrightarrow{\upsilon_\upsilon}) \tag{25c}$$

where $\overrightarrow{Z}$ is a distribution of objective functions sampled under satisfying constraints in Eqs. 15, 18 and 24. $\chi$ is a constriction coefficient, which help to balance global exploration and local exploitation. It is defined as

$$\chi = \frac{2}{\phi + \sqrt{\phi^2 - 4\phi}}, \quad with\ \phi = \phi_1 + \phi_2 > 4 \tag{26}$$

During the evolution process, the velocity update function, Eq. (25a) is referred as momentum that represents the node's current selection direction, Eq. (25b) is referred as social component, which the force of being attracted towards the best solution so far evaluated by the neighbors, and Eq. (25c) is referred as the cognitive component which the force of being attracted towards the previous solutions the node was know about. The variety between the CPSO and FPMSO algorithms is the velocity update function, which describes that not just the best position node is taken into account but all of its neighbors, which aims to increase diversity the search space. Additionally, the multi-swarm algorithm uses to generate a multipaths when velocity update function from the main path does not have a change in its objective function. Non-change of objective function for select path is detected when the path is not dominated by any path among the set feasible paths. However, when a new multipath is formed from the main path, the sensor nodes which triggered the construction and selection get merged into a new the multipath are removed from the main path. Algorithm 2 show the process of construction of a new multipath. therefore the velocity update function is defined as

$$\overrightarrow{\upsilon_\upsilon} = \chi(\overrightarrow{\upsilon_\upsilon} + \frac{1}{N}\sum_{\upsilon=1}^{N+M} \overrightarrow{Z}(0, \phi_1) \bigotimes ((\overrightarrow{p_\upsilon} - \overrightarrow{\upsilon_\upsilon}))) \tag{27}$$

The details of the proposed algorithm 1 are follows

**Algorithm 1**: Main Multi-Swarm Algorithm

1: Calculate $p_{best}$'s objective function as referred in Eq. 6 and Eq. 8 in terms of energy consumption and average delay. Then figure out the minimal value of objective function among these $p_{best}$'s objective function for $k$-disjoint multipath.
2: Calculate a constriction coefficient $\chi$ as referred in Eq. 14 in order to help in preventing velocity explosion.
3: Update the velocity value.
4: The node will select the optimal solution to improve the fault-tolerant multipath routing.

We devise an canonical particle multi-swarm optimization as shown in Algorithm 3 in order to solve the objective function. Basically, the idea behind using multi-swarm is apply multi-swarm after construct and select the paths accordingly to Algorithm 2. Multi-swarm employs a well-known multi-swarm to solve the tolerance problem. Specifically, by taking the full advantage of exchange information of all personal-best messages will contribute to ignoring several nodes fault tolerance error messages trapping in local optimal solution. This will lead to to defined to strengthen the node's ability to learn from other nodes' experience to guide its selection direction. Therefore, the performance of each algorithm depends on the way the nodes is influenced of select in the search space to the analysis and achieve the goal objective functions.

**Algorithm 2**: Construct and Selection Mechanism

1: input: The network topology;
2: for $\imath \in \aleph, \jmath \in \aleph$ where $\imath \neq \jmath$;
3: Calculate distance range $d^\alpha_{\upsilon(\imath,\jmath)} = \tau_{\upsilon(\imath,\jmath)}$ measure between $\imath$ and $\jmath$;
4: if $d^\alpha_{\upsilon(\imath,\jmath)} \leq \tau$ then;
5: Initialize candidate particle to be solution for construct and select the multipaths
6: Add the candidate to a candidate set
7: set the connectivity $k^\alpha_{\upsilon(\imath,\jmath)} = 1$ in $G$
8: end if
9: end for
10: Assign the fitness function for each paths in term of
11: Applying the canonical particle multi-swarm to compute a minimal energy consumption, average delay and maximize throughput
12: return selected the candidate set of construction and selection paths

The pseudo-code of the proposed CPMSO algorithm is shown in Algorithm 3. As we can see in lines 2 and 3, each sensor node initializes the cognitive, social acceleration coefficients of current velocity, and the current local-best information bounded within range $(\overrightarrow{lowerbound}, \overrightarrow{upperbound})$ to communicates with its neighbors. Then, after it evaluates the objective function in line 4 each sensor node starts to access the information of its neighbors according to the specified

**Algorithm 3**: Canonical Particle Multi-Swarm Optimization

```
1: input:Objective functions $\overrightarrow{Z}$;
2: $V := \{\overrightarrow{v}_1, \ldots, \overrightarrow{v_{N+M}}\} := InitNode(\overrightarrow{lowerbound}, \overrightarrow{upperbound}) \rightarrow \forall_v \in \{1, \ldots, N+M\} : \overrightarrow{N+M_v} := \overrightarrow{U}(\overrightarrow{lowerbound}, \overrightarrow{upperbound})$
3: $\overrightarrow{V} := \{\overrightarrow{v}_1, \ldots, \overrightarrow{v_v}\} := InitParticleVelocities (\overrightarrow{lowerbound}, \overrightarrow{upperbound}) \rightarrow \forall_v \in \{1, \ldots, N+M\} : \overrightarrow{v_{1,\ldots,N+M}} := (\overrightarrow{upperbound} - \overrightarrow{lowerbound}) \bigotimes \overrightarrow{U}(0,1) - \frac{1}{2}(\overrightarrow{upperbound} - \overrightarrow{lowerbound})$
4: $\Upsilon = \{\overrightarrow{\Upsilon}_1, \ldots, \overrightarrow{\Upsilon_{N+M}}\} := EvaluateObjectfunction(\overrightarrow{Z}) \rightarrow \forall_v \in \{1, \ldots, N+M\} : \Upsilon_v := f(\overrightarrow{Z_v})$
5: $P = \{\overrightarrow{p_1}, \ldots, \overrightarrow{p_{N+M}}\} := Initllocallyoptimal(Z) \rightarrow Z$
6: $P = \{p_1^Z, \ldots, p_v^Z\} := InitObjeectivefunction(\Upsilon) \rightarrow \Upsilon$
7: $G = \{\overrightarrow{g_1}, \ldots, \overrightarrow{g_{N+M}}\} := Initgloballyoptimal(P, T) \rightarrow P$
8: $G = \{g_1^Z, \ldots, g_1^Z\} := Initgloballyoptimal(P^Z, T) \rightarrow P^Z$
9: while termination condition nor met do
10:     for each particle/node $v$ of $N+M$ do
11:         $\overrightarrow{v}_v = \chi.(\overrightarrow{v}_v + \overrightarrow{Z}(0, \varphi_1) \bigotimes (\overrightarrow{p_v} - \overrightarrow{v}_{N+M}) + \overrightarrow{Z}(0, \varphi_2) \bigotimes (\overrightarrow{g_v} - \overrightarrow{v}_v))$
12:         $\overrightarrow{v}_p := \overrightarrow{Z}_p + \overrightarrow{v}_p$
13:     end for
14:     $\Upsilon := EvaluateObjectivefunction\ (Hop, Z)$
15:     $P, P^Z := Updatelocallyoptimal(Hop, Z) \rightarrow \forall_p \in \{1, \ldots, N+M\} : \overrightarrow{p}_p, p_p^Z := \begin{cases} \overrightarrow{Z}_p, y_i & \text{if } \Upsilon_v \text{ better than } p_p^f \\ \overrightarrow{p}_p, p_p^f & \text{otherwise} \end{cases}$
16:     $G, G^f := Updategloballyoptimal(P, P^Z, T) \rightarrow \forall_p \in \{1, \ldots, N+M\} : \overrightarrow{g_p}, g_p^Z := best(P_{T_p}, P_{T_p}^Z),\ where\ T_p\ are\ the\ neighbors\ of\ p$
17: end while
18: best solution found
```

local-best topology as seen in lines $5-8$. The number of iterations of the algorithm to achieve the best objective function value begins from line 9. After each step starting from line 10, each sensor node considers some neighboring states of the current desired state of fault-tolerance, and probabilistically decides between either selecting, accessing or exchanging the information according to different local-best information with one neighboring node based on the communication overhead or staying in the current state. Indeed, these probabilities ultimately lead the proposed model to move to states of lower energy consumption, lower delay and higher throughput. Because of each sensor node has a $k$-disjoint multipath connected to neighboring node in its topology according to local-best information. Therefore, this topology assures that the selected sensor nodes have full diversity in updating their global optimal for all their neighbors which can lead to less communication overhead as described in line 16.

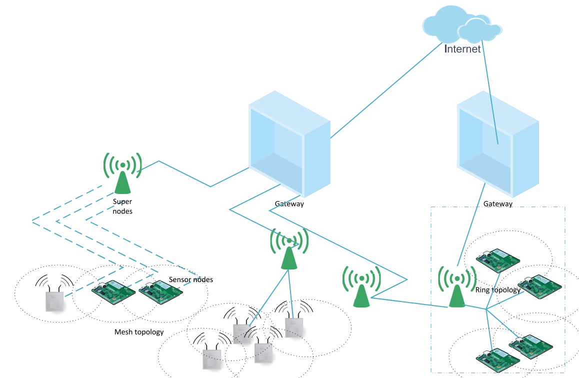


Fig. 2. The devices deployed in IoT.

## V. Performance Evaluation

In order to assess the performance of the proposed algorithm, we performed extensive simulations. We have implemented our algorithms using MatLab [32] to develop, generate network topology, evaluates the objective functions and visualize the outputs of the evolution. We employ 500 sensor nodes distributed uniformly over the area of $1000 * 1000m$ as seen in Fig. 2. The supernodes are also distributed uniformly in this area. The path loss exponent for wireless channel $\alpha$ is between 2 and 4. The initial value of transmission range of the nodes is set to be $12.00m$ in order to guarantee the connection among the supernodes and nodes with satisfy the constraints of problems. For each algorithm CPMSO, FPMSO, and CPSO a three main test experiments were run to investigate the usability of these algorithms in finding and selecting robust optimal multipath according the objective functions have been devised and executed. The first experiment investigates the performance in terms of energy consumption, average delay, and total throughput in general. Meanwhile the second the performance while increasing the number of sensor nodes. Finally, the third experiment the behavior of sensor nodes under different swarm topologies and processing constraints. The estimation of robust objective function in terms of energy consumption, average delay, and throughput is done via the approximation of the effective or expected fitness function using $k$-disjoint multipath for the approximation. The coefficients $\phi_1 = 2.8$ and $\phi_2 = 1.3$ are set so that $\phi_1 + \phi_2 \geq 4$. And thus, to ensure that these algorithms optimize with bounds of feasible search space, a cut-off boundary constraints processing is used, whereas $lb$ and $ub$ as derived in [28]. All variables, parameters like positions, velocities and other constants values are summarized in Table II.

### A. Experiment (1) General Performance Investigation

We report our first simulation results presenting the performance of finding robust optimal multipath in terms of energy consumption, average delay and throughput for the employment 30 sensor nodes and supernodes in resulting network topology obtained after executing three algorithms CPMSO, FPMSO, and CPSO with maximum number of hops (hop=5).

*1) Total Energy Consumption:* Figure 3 present the total energy consumption resulting topology obtained after executing CPMSO, FPMSO, and CPSO. We notice that the total

TABLE II
DEFINITION OF PARAMETERS

| Parameter | Value |
|---|---|
| $E_{elec}$ | $50nJ/bit$ |
| $\varepsilon_{fs}$ | $10pJ/bitm^2$ |
| $\varepsilon_{mp}$ | $0.0013pJ/bitm^2$ |
| Topology structure | Square ($1000m * 1000m$), sensor node distributed uniformly |
| Total number of sensor nodes | $30, 40, 50$ sensor nodes |
| Message payload | 64 bytes |
| Data length $p$ | 2000 bits |
| Transmission range | $12.00m$ |
| Tx data rate | $250kbps$ |
| Personal best coefficient $\phi_1$ | 2.8 |
| Neighbor best coefficient $\phi_2$ | 1.3 |
| Robust fitness estimation | $f_{eff}$ |
| Initialization | Energy consumption, delay, and throughput |
| Boundary processing | Cut-off method and ignore cut-off method |
| Degree of Disjoint connectivity $k$ | 2,3,4 |
| Number of hops | $hops = 1, 2, 3, 4, 5$ |

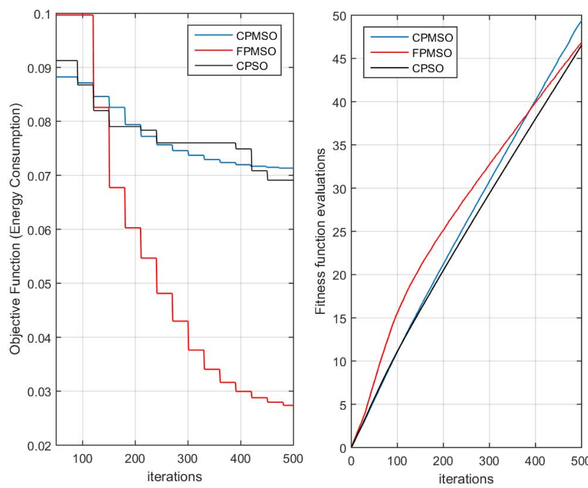


Fig. 3. Multi-swarm optimization routing algorithm for optimizing energy consumption.

energy consumption in the $k$-disjoint generated by propose algorithms is better than CPSO. This because that solving objective function utilized by CPSO has difficulties in discovering $k$-disjoint multipath after recovery the fault tolerance error messages since the search space is long resulting unable to substituted with $k$-disjoint multipath and thus high total transmission energy. Another important observation that related to exchange the messages for fault tolerance between supernodes and nodes is CPSO perform significantly worse than FPMSO and CMPSO, since CPSO requires significantly more control messages to exchange between the neighbors. Therefore, CPSO need to find $k$-disjoint multipath in its reachable neighborhood whereas the FPMSO and CPMSO can directly search for paths using the less control messages between the reachable neighborhood. Thus, the $k$-disjoint multipath for FPMSO and CPMSO can achieve lower total energy consumption compare to CPSO algorithm.

*2) Total Average Delay:* Figure 4 shows the average delay of the selected optimal multipath from source to the sink. We can observe that the proposed algorithms; FPMSO and

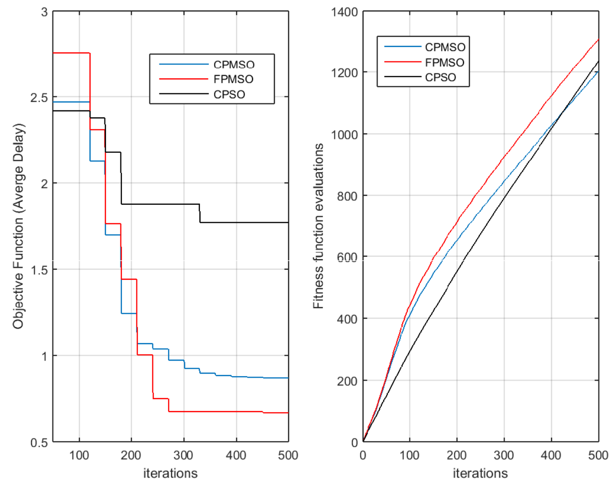


Fig. 4. Multi-swarm optimization routing algorithm for optimizing average delay.

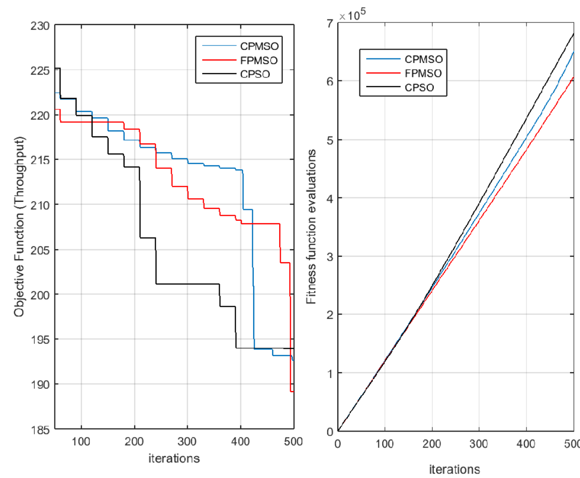


Fig. 5. Multi-particle swarm optimization routing algorithm for optimizing throughput.

CPMSO, have demonstrated a lower delay per hop compared to CPSO. This can be returned to the selection and maintenance of $k$-disjoint multipath for fault tolerance which can satisfy the hop requirement by selecting the next hop in the neighborhood of each node. Consequently, it requires significantly less control messages for fault tolerance compared to CPSO for selecting and maintaining 1-hop neighborhood. Therefore, this indicates that both FPMSO and CPMSO are more feasible for $k$-disjoint multipath than CPSO.

*3) Total Throughput:* Throughput may be frequent due to high-bit error rate or other conditions such as the environmental ones. Therefore, we present in Fig. 5 the effect of solving the objective function as referred in Eq. 8 on throughput. We observe that when minimizing delay with the increasing optimal number of hops, throughput degrades significantly. Actually, this is an expected result as minimizing delay under the aforementioned constraints with optimal number of hops leads to lower number of exchanged control messages for fault tolerance and thus nodes can be obtained completely. Finally, we concluded that the performance attained depends on the network topology construction. Although that CPSO achieve fully connected topology i.e. when each node has connection

TABLE III
ALGORITHMS COMPARISON

| Algorithm | Number of node deployment | $\arg(100 - \frac{\arg(f_{eff})}{\max(f_{eff})} \cdot 100)$ |
|---|---|---|
| CPMSO | 30 | $\approx 84.52\%$ |
| FPMSO | 30 | $\approx 92.39\%$ |
| CPSO | 30 | $\approx 71.83\%$ |
| CPMSO | 40 | $\approx 83.64\%$ |
| FPMSO | 40 | $\approx 89.78\%$ |
| CPSO | 40 | $\approx 68.21\%$ |
| CPMSO | 50 | $\approx 81.39\%$ |
| FPMSO | 50 | $\approx 87.65\%$ |
| CPSO | 50 | $\approx 67.06\%$ |

TABLE IV
ALGORITHMS COMPARISON

| Algorithm | Topology | $\arg(100 - \frac{\arg(f_{eff})}{\max(f_{eff})} \cdot 100)$ |
|---|---|---|
| CPMSO | ring | $\approx 85.11\%$ |
| FPMSO | ring | $\approx 93.02\%$ |
| CPSO | ring | $\approx 75.48\%$ |
| CPMSO | mesh | $\approx 83.74\%$ |
| FPMSO | mesh | $\approx 89.28\%$ |
| CPSO | mesh | $\approx 70.15\%$ |

to all nodes in the swarm as neighbors, it has exhibited a particularly bad performance compared to others. This is because the simultaneous attraction of $k$-disjoint multipath provoking a random behavior from each node to discover, construct an select the paths. Meanwhile this behavior could support optimal performance in FPMSO and CPMSO with fully connectivity.

### *B. Experiment (2) Objective Function Performance Investigation*

In this experiment the objective function is conducted with varying number of sensor nodes in the network. The overview of the results that have been produced for each algorithm is summarized in Table III while increasing the number of sensor nodes. To get an idea of how the performance of the three algorithms varies we compare them in terms of the obtained solutions under the same settings which are used for CPSO. Results of the comparison show that the average objective function values developed for the algorithm FPMSO followed by CPMSO are approximately equal to 92.39% and 84.52%, respectively. Thus, it can be observed that the best performing algorithm is the FPMSO algorithm, followed by the CPMSO algorithm for 30 nodes. Meanwhile, CPSO has the worst value which is approximately equal to 71.83% under the same setting. This is because CPSO can construct and select optimal paths from unfavorable regions in the search space. The low number of generated paths for the objective function can be the reason behind CPSO's divergence from the global optimal solution than other algorithms. Finally, these results show that increased exploration can indeed be beneficial in finding better solutions from a quality stand-point for specific objective function (i.e. energy consumption).

### *C. Experiment (3) Topology Performance Investigation*

Aside from the experiment V-B, this experiment V-C reports the performance of each algorithm while varying the network topology. Table IV presents the difference in quality of each algorithm while applying mesh and ring topologies respectively. The information in mesh and ring topology is capsuled more locally in their multipaths. However, the information in ring topology has been shared between only two neighboring sensor nodes, while in mesh topology the information is shared among the sensor nodes of per-defined size of the multipaths. In ring topology FPMSO algorithm had a value approximately equal to 93.02% which represents better performance in finding optimal solution compared to CPMSO and CPSO algorithms which are approximately 85.11% and 75.48% respectively. Meanwhile, the mesh topology does not have all their information shared, however, the sensor nodes tend to explore more and this tend to construct and select more multipaths, which could be beneficial for finding the optimal paths. Generally, all algorithms present promising performance in both mesh and ring topologies. In CPSO the mesh topology presents approximately 70.15% which is slightly better than FPMSO and CPMSO algorithms.

Indeed, a little bit difference of quality could be attributed to the amount of exchanged information between the nodes which defined by the connectivity definition among the sensor nodes and supernodes. However, in all topologies the information between nodes is available, while in the ring topology the information is shared between only two neighboring nodes. Meanwhile, in mesh topology the information is shared among nodes of a pre-defined connectivity degree. Hence, ring topology has enough information shared, and it tends to explore more neighboring nodes in order to create more paths. In FPMSO and CPMSO similar setting to CPSO are used to make a fair comparison between them. Therefore, several things can be concluded. Firstly, both FPMSO and CPMSO are able to present better performance than CPSO, because of the mesh topology in multi-swarm algorithm which performs the local search and records all information from the neighbors available in order to find the optimal paths. Finally, as we referred in Eq. 25 which is divided into two phases. The first phase seems to depict the convergence of the current node selection, which using cognition in normal swarm. In multi-swarm, the main swarm use cognition and momentum components of the standard velocity updates. The second phase depicts the convergence behavior of the multi-swarm, which provides not only the best position node but all best neighbors. This lead to the assumption that the nodes need a finite time to stabilize their objective function values and positions and this in turn would mean that by setting optimal transmission range of a node which uses a full PSO algorithm to increases the diversity of the search space and generates much better results.

## VI. CONCLUSION

In this paper we propose a bio-inspired particle multi-swarm optimization (PMSO) strategy to construct, recover and select $k$-disjoint multipath routes. Two position-information in terms of personal-best position, and the global position are introduced in the form of velocity update to enhance the performance of routing algorithm. To validate this strategy, we assessed objective function which considers the average energy consumption and average in-network delay. Our results show that the strategy using the characteristics of all personal-best information is a valid strategy for the purposes of improving the PMSO performance. Moreover, the proposed algorithm has also been compared with similar algorithms, which optimize the energy consumption and average delay over the explored paths.

## REFERENCES


[1] G. Singh and F. Al-Turjman, "A data delivery framework for cognitive information-centric sensor networks in smart outdoor monitoring," *Comput. Commun. J.*, vol. 74, no. 1, pp. 38–51, 2016.

[2] V. Petrov *et al.* (2017). "When IoT keeps people in the loop: A path towards a new global utility." [Online]. Available: https://arxiv.org/abs/1703.00541

[3] M. Z. Hasan, H. Al-Rizzo and F. Al-Turjman, "A survey on multipath routing protocols for QoS assurances in real-time multimedia wireless sensor networks," *IEEE Commun. Surveys Tuts.*, 2017, doi: 10.1109/COMST.2017.2661201.

[4] Y. Zeng, L. Xu, and Z. Chen, "Fault-tolerant algorithms for connectivity restoration in wireless sensor networks," *Sensors*, vol. 16, no. 1, p. 3, 2016.

[5] A. Hadjidj, A. Bouabdallah, and Y. Challal, "HDMRP: An efficient fault-tolerant multipath routing protocol for heterogeneous wireless sensor networks," in *Proc. 7th Int. Conf. Heterogeneous Netw. Quality, Rel., Secur. Robust. (QShine)*, 2010, pp. 469–482.

[6] M. I. Akbas, M. R. Brust, D. Turgut, and C. H. C. Ribeiro, "A preferential attachment model for primate social networks," *Comput. Netw.*, vol. 76, pp. 207–226, Jan. 2015.

[7] S. Jiang, "Linear decision fusion under the control of constrained PSO for WSNs," *Int. J. Distrib. Sensor Netw.*, vol. 8, no. 1, p. 871596, 2012.

[8] Y. Hu, Y. Ding, and K. Hao, "An immune cooperative particle swarm optimization algorithm for fault-tolerant routing optimization in heterogeneous wireless sensor networks," *Math. Problems Eng.*, vol. 2012, pp. 1–19, Dec. 2012, Art. no. 743728, doi: 10.1155/2012/743728.

[9] W. H. Lim and N. A. M. Isa, "Particle swarm optimization with adaptive time-varying topology connectivity," *Appl. Soft Comput.*, vol. 24, pp. 623–642, Nov. 2014.

[10] C.-H. Wu and Y.-C. Chung, "Heterogeneous wireless sensor network deployment and topology control based on irregular sensor model," in *Advances in Grid and Pervasive Computing*, C. Cérin and K.-C. Li, Eds. Berlin, Germany: Springer, 2007, pp. 78–88.

[11] J. K. Vis. (Jan. 15, 2015). Particle swarm optimizer for finding robust optima. Leiden, The Netherlands. [Online]. Available: http://www.liacs.nl/assets/Bachelorscripties/2009-12JonathanVis.pdf

[12] M. A. Montes de Oca and T. Stützle, "Convergence behavior of the fully informed particle swarm optimization algorithm," in *Proc. 10th Annu. Conf. Genetic Evol. Comput.*, Atlanta, GA, USA, 2008, pp. 71–78.

[13] H. Bagci, I. Korpeoglu, and A. Yazici, "A distributed fault-tolerant topology control algorithm for heterogeneous wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 914–923, Apr. 2015.

[14] M. I. Akbas, M. R. Brust, C. H. Ribeiro, and D. Turgut, "Deployment and mobility for animal social life monitoring based on preferential attachment," in *Proc. IEEE 36th Conf. Local Comput. Netw. (LCN)*, Oct. 2011, pp. 484–491.

[15] M. A. Adnan, M. A. Razzaque, I. Ahmed, and I. F. Isnin, "Bio-mimic optimization strategies in wireless sensor networks: A survey," *Sensors*, vol. 14, no. 1, pp. 299–345, 2014.

[16] Y. H. Robinson and M. Rajaram, "Energy-aware multipath routing scheme based on particle swarm optimization in mobile ad hoc networks," *Sci. World J.*, vol. 2015, pp. 1–9, Dec. 2015, Art. no. 284276. [Online]. Available: http://dx.doi.org/10.1155/2015/284276

[17] M. Azharuddin *et al.*, "A PSO based fault tolerant routing algorithm for wireless sensor networks," in *Information Systems Design and Intelligent Applications*, vol. 1, J. K. Mandal, Ed. New Delhi, India: Springer, 2015, pp. 329–336.

[18] C. Vimalarani, R. Subramanian, and S. N. Sivanandam, "An enhanced PSO-based clustering energy optimization algorithm for wireless sensor network," *Sci. World J.*, vol. 2016, pp. 1–11, Jan. 2016, Art. no. 8658760. [Online]. Available: http://dx.doi.org/10.1155/2016/8658760

[19] M. Pant, T. Radha, and V. P. Singh, "A simple diversity guided particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 3294–3299.

[20] H.-L. Shieh, C.-C. Kuo, and C.-M. Chiang, "Modified particle swarm optimization algorithm with simulated annealing behavior and its numerical verification," *Appl. Math. Comput.*, vol. 218, no. 8, pp. 4365–4383, 2011.

[21] G. Singh and F. Al-Turjman, "Learning data delivery paths in QoI-aware information-centric sensor networks," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 572–580, 2016.

[22] Y. Zhou, X. Wang, T. Wang, B. Liu, and W. Sun, "Fault-tolerant multipath routing protocol for WSN based on HEED," *Int. J. Sensor Netw.*, vol. 20, no. 1, pp. 37–45, 2016.

[23] J. J. Liang and P. N. Suganthan, "Adaptive comprehensive learning particle swarm optimizer with history learning," in *Simulated Evolution and Learning*, T.-D. Wang, Ed. Berlin, Germany: Springer, Oct. 2006, pp. 213–220.

[24] F. Al-Turjman, "Information-centric sensor networks for cognitive IoT: An overview," *Ann. Telecommun.*, vol. 72, no. 1, pp. 3–18, 2017.

[25] X. Han, X. Cao, E. L. Lloyd, and C.-C. Shen, "Fault-tolerant relay node placement in heterogeneous wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 5, pp. 643–656, May 2010.

[26] A. Munir, J. Antoon, and A. Gordon-Ross, "Modeling and analysis of fault detection and fault tolerance in wireless sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 14, no. 1, p. 3, 2015.

[27] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*. Hoboken, NJ, USA: Wiley, 1994.

[28] M. Z. Hasan, F. Al-Turjman, and H. Al-Rizzo, "Optimized multi-constrained quality-of-service multipath routing approach for multimedia sensor networks," *IEEE Sensors J.*, vol. 17, no. 7, pp. 2298–2309, Apr. 2017.

[29] M. Z. Hasan, F. Al-Turjman, and H. Al-Rizzo, "Evaluation of a duty-cycled protocol for TDMA-based wireless sensor networks," in *Proc. Int. Conf. Wireless Commun. Mobile Comput. (IWCMC)*, Sep. 2016, pp. 964–969.

[30] M. Z. Hasan and F. Al-Turjman, "Evaluation of a duty-cycled asynchronous X-MAC protocol for vehicular sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 95, pp. 1–16, 2017, doi: 10.1186/s13638-017-0882-7.

[31] B. Grimstad and A. Sandnes, "Global optimization with spline constraints: A new branch-and-bound method based on B-splines," *J. Global Optim.*, vol. 65, no. 3, pp. 401–439, 2016.

[32] *M.U.s.G. MathWorks*, The Mathworks Inc., Natick, MA, USA, 1992.



**Mohammed Zaki Hasan**, photograph and biography not available at the time of publication.

**Fadi Al-Turjman**, photograph and biography not available at the time of publication.