Online Routing Algorithms for Maximum Lifetime in Wireless Sensor Networks

Mahmood R. Minhas, Sathish Gopalakrishnan, Victor C. M. Leung

Department of Electrical & Computer Engineering,

University of British Columbia,

Vancouver, BC V6T1Z4, Canada

Email: {mrminhas,sathish,vleung}@ece.ubc.ca

Abstract

The importance of energy preservation in wireless sensor networks is long established, and a number of maximum lifetime routing schemes exist in the literature. Many of the existing online routing schemes are based on the use of some edge-weight functions, but those schemes exhibit diverse performances in terms of the obtained lifetime results. The major contribution of the present work is to critically analyze the existing edge-weight functions in order to identify their shortcomings. We obtain an insight into the desired behavior of weight functions, and design better functions that are employed in the proposed approaches. We present two online routing schemes, namely the fuzzy

---

1This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant STPGP 322208-05.
maximum lifetime routing scheme (FML), and the fuzzy multiobjective routing scheme (FMO). The former tries to maximize the lifetime objective, whereas the latter strives to simultaneously optimize the lifetime as well as the energy consumption objective. Simulation results obtained under a variety of network scenarios indicate the superiority of the proposed schemes over a number of other online routing schemes in terms of the lifetime as well as the energy consumption metric. For better scalability, we also present a fully distributed implementation of the FML routing scheme. The overhead analysis shows that the distributed scheme is successful in considerably reducing the control overhead.

**keywords:** Wireless sensor networks, lifetime maximization, energy-aware routing, online routing algorithms, distributed algorithms, multiobjective optimization, fuzzy functions and operators.

1 **Introduction**

Wireless sensor networks (WSNs) have found a number of potential applications in various fields [1,7]. WSN is usually comprised of battery-operated sensor devices capable of communication and processing. In many real-life applications, sensor nodes are deployed in remote or hostile environments with a high node density, and the operational lifetime of a WSN depends on the available battery energies in the nodes. Thus there is a compelling need for energy-aware techniques and protocols at all the layers including the routing layer [7,19].

The existing routing schemes can broadly be classified into online routing and offline routing schemes (a brief review of the related energy-aware and maximum lifetime routing
schemes is given in Section 8). An online routing scheme needs to find a route for each routing request without the knowledge of the future routing requests [18]. Section 2 describes the online routing model in detail.

A number of online routing schemes are found in the literature [11, 13, 17, 18]. The two closest, in working mechanism, to the present work are the online maximum lifetime routing heuristic (OML) due to Park and Sahni [18], and the CMAX heuristic due to Kar et al. [11]. The core mechanism in both the schemes is the use of edge-weight functions. The function computes a certain weight for each edge, depending on the residual energy value of its source node (the transmitting sensor node). Followed by the weight assignment step, the maximum lifetime path is found on the basis of the assigned weights. Thus the edge-weight function has a decisive impact on the overall performance of these online routing schemes.

A major contribution of this work is an effort to investigate the shortcomings in the existing weight functions of the OML and the CMAX schemes in order to obtain an insight into the desired behavior and characteristics of a more effective weight function. The effort leads to the design of improved fuzzy-logic based weight functions that are used in the proposed routing algorithms. Section 4 presents a comparative analysis of the weight functions used by the FML, OML, and CMAX schemes, and presents an investigative insight into the improvement obtained. The results confirm our belief that the improved fuzzy-logic-based weight functions provide an edge to the proposed schemes over the aforementioned two existing online routing schemes.

We present two online routing algorithms, namely the fuzzy maximum lifetime routing algorithm (FML), and the fuzzy multiobjective routing algorithm (FMO). Moreover, we also
present a distributed FML routing implementation. Sections 3 and 5, respectively, describe algorithmic details of the FML and the FMO routing schemes, whereas a description of the distributed FML algorithm is given in Section 6.

The motivation for resorting to fuzzy logic in designing the weight functions is that the fuzzy sets and membership functions have proved to be an effective mean of expressing the costs of the various objectives involved in an optimization problem [27, 29]. Secondly, in case of the multiobjective routing, there is a need for designing a multiobjective aggregation function, and fuzzy logic offers a better alternative, namely ordered weighted operator (OWA) to the other traditional aggregation approaches such as weighted sum [26,28]. A brief introduction to fuzzy sets, membership functions, and operators is given in the appendix.

We conduct a performance comparison of the proposed routing algorithm to the OML heuristic (OML) and the CMAX heuristic. Detailed simulation results and related discussions are given in Section 7. An extensive set of simulation results obtained under various network scenarios and parameters show that the proposed FML algorithm outperforms both the OML as well as the CMAX heuristic, in terms of the network lifetime and the average energy consumption metrics. It should be noted that Park and Sahni [18] have already demonstrated the supremacy of the OML heuristic, not only over the CMAX heuristic, but also over a number of other existing maximum lifetime routing algorithms including MRPC [17] and \(z.P_{\text{min}}\) [3]. In case of the distributed FML, the overhead analysis shows that the distributed approach has a considerably lower overhead than that of the centralized FML algorithm (see Section 7.4), and hence it may offer a viable online maximum lifetime routing solution for real-life large-scale WSN deployments.
The system model used in this work is similar to the model used in our earlier work [16]. We consider a static WSN deployment, and model it as a directed graph $G(V, E)$, where $V$ is the set of nodes, and $E$ is the set of edges. All the nodes have equal initial energy $\sigma$. The node batteries are neither replaceable nor remotely rechargeable. Each node $v_i \in V$, has a set of neighbor nodes (denoted as neighborhood set $N_i$) that the node $v_i$ can reach by a single hop transmission, using a certain maximum transmission radius $r_t$. An edge $e(v_i, v_j)$ between the two nodes is defined to exist only if $v_i$ and $v_j$ are within the radio transmission range of each other, i.e., if $d_{ij} \leq r_t$, where $d_{ij}$ is the Euclidian distance between the two nodes.

The energy consumption model used in our simulations is based on the first order radio propagation model, used by many other works in the context of WSN routing [4, 9, 10]. According to this model, the energy expended by a sensor node during transmission and reception of a $k$-bit packet is given by Equations (1) and (2), respectively.

$$TX_{ij} = (A + B \cdot d_{ij}^m) \cdot k$$  

$$RX_{ij} = A \cdot k$$

In Equations (1) and (2), $A$ is distance-independent and accounts for the energy consumed by transmitter or receiver circuitry, $B$ denotes the energy required by the transmitter’s amplifier, whereas $m$ is a field constant typically in the range [2,4] and depends on certain characteristics of the wireless medium.

In the online routing model, a routing algorithm needs to find a route for each routing
request without the knowledge of the future routing requests. Further, there are no prefixed source nodes, and also no assumptions are made about data generation rate. On the other hand, an offline routing algorithm finds a route for a given routing request with the full knowledge of the future routing requests (a routing request is initiated by a sensor node to indicate its intention to send its sensed data to a sink node). Moreover, some offline routing schemes assume that only a prefixed node (or a subset of nodes) in the entire network may act as source node(s), and that the data generation rates are known a priori. Several offline routing approaches based on analytical and heuristic solutions have been proposed [4,15,23].

We believe while an offline routing model may suit certain WSN applications, there are numerous WSN scenarios, where an online routing model captures the characteristics of a real-time event-driven WSN application more accurately, i.e., an event may occur anywhere at any point in real-time, and any node that detects that event may need to act as a source node. There are numerous examples of such applications including surveillance in a battle or disaster field (person or object detection, target detection, fire outbreak), environmental monitoring (temperature or humidity rise at a certain location, activity in a target space such as a nest or a plant) [1,7,25].

We have a set of source nodes performing sensing task. At any time, a source node $v_m$ may initiate a routing request $r_h(v_m, v_n)$, where $h = \{1, 2, \cdots\}$, for sending its sensed data to a sink node $v_n$. A routing request does not imply a single data message (packet), rather it represents a sequence of data packets to be sent from the source node to a sink node. We assume there are numerous routing requests $\{r_1, r_2, r_3, \ldots\}$ during the lifetime of WSN. The goal of the proposed online routing algorithms is to efficiently route each routing request $r_h$,
without the knowledge of any future routing requests $r_q$ (where $q > h$), in such a manner that maximizes the number of successful routing requests before the end of the WSN lifetime.

We employ a simple, but commonly used, WSN lifetime definition: The WSN lifetime is equal to the minimum of the lifetimes of all the sensor nodes in the network, i.e., the network lifetime ends as soon as any node in WSN runs out of its battery [4,6,15,23]. If the lifetime of a WSN node is denoted by $T_{v_i}$, the WSN lifetime may be expressed as given by Equation (3).

$$T = \min_i \{T_{v_i}\} \quad \forall v_i \in V$$ (3)

3 The Fuzzy Maximum Lifetime Routing Algorithm (FML)

The FML algorithm makes use of an edge-weight function like some other online routing schemes, but as mentioned earlier, for the proposed FML algorithm, we explicitly formulate the weight function based on a novel use of fuzzy sets and membership functions.

In order to apply fuzzy logic to the maximum lifetime routing problem, a linguistic variable *residual energy* of a node $v_i \in V$, and a fuzzy set *high lifetime* are defined. A fuzzy membership function (depicted in Figure (1)) is designed to map a value of the residual energy to its corresponding fuzzy lifetime membership value $\mu_{lt}^{ij}$ in the fuzzy set high lifetime.

As may be observed, the function assigns a high membership value to an edge having a large amount of residual energy ($re$) at its starting node $v_i$. Initially, when the residual
energy of a node is equal to $\sigma$, each of its outgoing edges is assigned a membership value of 1.0. As the node residual energy decays, the corresponding membership value falls, initially at a slow rate and then at a sharp rate. This behavior of the membership function strongly discourages the inclusion, on the selected routing path, of those intermediate nodes that have depleted their energy beyond a certain threshold value. The threshold point may be altered by adjusting the value of $\alpha$. An expression for the fuzzy lifetime membership function can be derived using the equation of line, and is given by Equation (4).

$$
\mu_{lt}^{ij} = \begin{cases} 
1 - \left( \frac{1 - \gamma}{1 - \alpha} \right) \cdot \left( 1 - \frac{re(v_i)}{\sigma} \right) : \\
\text{if } \alpha \cdot \sigma < re(v_i) \leq \sigma \\
\frac{\gamma}{\alpha \cdot \sigma - TX_{ij}} \times (re(v_i) - TX_{ij}) : \\
\text{if } TX_{ij} < re(v_i) \leq \alpha \cdot \sigma \\
0 : \text{if } re(v_i) \leq TX_{ij}
\end{cases}
$$

(4)
Here

\[ re(v_i) = ce(v_i) - TX_{ij} \] (5)

where \( re(v_i) \) and \( ce(v_i) \) denote residual energy and current energy of node \( v_i \) respectively, and \( \alpha \in [0, 1] \), \( \gamma \in [0, 1] \) are algorithmic parameters. Then a weight is assigned to each edge \( e(v_i, v_j) \) using the following equation:

\[ w_{ij} = 1 - \mu_{lt}^{ij} \] (6)

\( G (V, E) \) is the given directed graph
For each routing request \( r_h(v_m, v_n) \)
For each edge \( e(v_i, v_j) \) in \( V \)
    Compute fuzzy lifetime membership \( \mu_{lt}^{ij} \)
    Assign weight \( w_{ij} = 1 - \mu_{lt}^{ij} \)
End For
Find minimum weight path \( p_h \) from \( v_m \) to \( v_n \)
Send data along path \( p_h \)
Compute the minimum node energy in \( G (V, E) \)
IF a node has run out of energy, stop.
End For

Figure 2: The fuzzy maximum lifetime (FML) routing algorithm.

The proposed fuzzy routing algorithm (shown in Figure (2)) works as follows: When a routing request \( r_h(v_m, v_n) \) is initiated, a fuzzy lifetime membership value is computed for each edge in WSN using Equation (4), and hence a weight is assigned to each edge using Equation (6). Following the weight assignments, the maximum lifetime path \( p_h \), between \( v_m \) and \( v_n \), is found using the Dijkstra’s shortest path algorithm [8]. It should be noted that FML algorithm has a complexity advantage over the OML algorithm, since for each routing request, the FML algorithm requires only one shortest path search whereas the OML
4 A Comparative Analysis of the FML, OML and CMAX Weight Functions

As mentioned earlier, the proposed fuzzy-logic-based weight function is the distinguishing feature of the FML routing algorithm that contrasts it from the OML and the CMAX heuristics. We believe that the above mentioned feature is the underlying reason for the superiority of FML over the other two heuristics (see the simulation results and comparisons given in Section 7). Therefore, it seems natural to investigate and comprehend the cause of supremacy of FML. In this section, we follow this natural instinct, and make an effort to identify the possible strength in FML mechanism.

Before presenting the comparative study, it is necessary to describe brief details of the OML and the CMAX heuristics. CMAX [11] formulates an edge-weight function, which is used to assign a weight to each edge in the network. The weight for an edge $e(v_i, v_j)$ is computed as $TX_{ij} (\lambda \eta(i) - 1)$, where $\eta(i) = 1 - ce(v_i)/\sigma$, and $\lambda$ is an algorithmic parameter. As can be seen, CMAX due to term ‘$\eta(i)$’ assigns a high weight to an edge $e(v_i, v_j)$, if the node $v_i$ has consumed a large amount of its initial energy. Then Dijkstra’s algorithm is used to find the minimum cost path. CMAX algorithm needs only one shortest path search, whereas $zP_{\text{min}}$ requires two to find the maximum lifetime path.

Park and Sahni [18] presented an online maximum lifetime algorithm (OML) that is also based on edge weight assignment like CMAX. The OML heuristic works as follows: Let
$G(V, E)$ be the given graph, and $G'(V, E')$ be the graph after removing all edges $e(v_i, v_j)$ such that $ce(v_i) < TX_{ij}$. For a routing request to send data from a source $v_m$ to a sink $v_n$, the minimum energy path $P'$ is found, and the minimum residual energy $\text{minRE}$ along $P'$ is computed. A second pruned graph $G''(V, E'')$ is obtained by removing all edges $e'(v_i, v_j)$ from $G'$ such that $ce(v_i) - TX_{ij} < \text{minRE}$. The weight $w''$ to be assigned to each edge in $G''$ is computed using the following equation.

$$w'' = (TX_{ij} + \rho(v_i)) \cdot (\lambda^{n(i)} - 1) \quad (7)$$

where $\lambda$ is an algorithmic parameter of OML, and $\rho(v_i)$ is given by

$$\rho(v_i) = \begin{cases} 0 & \text{if } ce(v_i) - TX_{ij} > eMin(v_i) \\ c & \text{otherwise} \end{cases} \quad (8)$$

where $eMin(v_i)$ is the energy required by node $v_i$ to transmit to its closest neighbor and $'c > 0'$ is an algorithmic parameter.

Figure 3: A comparison of the proposed FML weight function to the OML and the CMAX weight functions. In our opinion, the CMAX weight assignment behavior is overly-pessimistic, whereas that of the OML is too optimistic (or too late) to have a timely effect on the lifetime objective.
A visible similarity in all the three heuristics is the use of edge weight functions, i.e., all three schemes proceed by assigning a weight to each edge in the network and then finding the minimum cost path. Therefore, a comparative study of these weight functions may lead us to an insight into the performance difference among FML and the other heuristics.

The OML and the CMAX weight functions are formulated implicitly to attain a certain behavior of the edge-weight corresponding to a given value of the residual energy. On the other hand, the proposed weight functions are explicitly designed based on use of fuzzy membership functions, which offer a flexible way of representing the cost of a given objective as a fuzzy membership value [29].

The weight functions used by OML and CMAX seem very similar apparently, since both use a common term $\lambda$, however $\lambda$ is defined in different ways by both the approaches. A brief discussion on the differences among these two is given by [18]. The FML weight function is defined in an altogether different manner, i.e., the weight assigned to an edge is computed based on fuzzy lifetime membership value of the edge that in turn ultimately is a function of the residual energy level of the originating node. Figure (3) shows the weight assigned to an edge as a function of the residual energy of the by each of the three heuristics.

As discussed by Park and Sahni in [18], it is relatively easier to explain the performance lag of CMAX since based on its definition of $\lambda$ term, it prematurely starts assigning a higher weight to an edge that has depleted a large fraction of its initial battery power (notice the rise in weight curve as soon as $re$ falls below values of 0.5). The authors argued that this behavior of CMAX weight function is premature, and therefore suggested that even though a node may have depleted a large fraction of its energy, this should not imply its exclusion
from the route search process since it may rule out a number of good routing choices. In an
effort to rectify this behavior of CMAX, OML authors proposed that, for achieving a longer
lifetime, the edge weight should better be a function of the ratio of the residual energy of the
node to the minimum energy required to transmit to the closest neighbor. As a result, they
relaxed the constraint and obtained a weight function curve that starts rising only after the
residual energy decays beyond 5% of the initial energy level.

We argue that the above described threshold used in the OML scheme for raising the
weight is probably too late to have a considerable effect on the maximum lifetime routing
path search process: The OML weight function keeps assigning a negligible (almost zero)
weight to an edge until its originating node has depleted almost 95% of its battery power. The
FML weight function tries to attain a balance between the two extreme behaviors of CMAX
and OML, and strives to respond to the continuous decay of residual energy throughout
the network operation as described in following: The weight is increased from 0 to 0.1 at a
steady lower rate when the residual energy is in range \([1.0, 0.2]\) (notice that OML assigns an
almost 0 weight at even \(re(v_i) = 0.2\)). There is a breakpoint at \(re(v_i) = 0.2\), as beyond that
point, the weight increases at a much sharper rate.

To further verify our above mentioned proposition, we improvised the FML weight function
to simulate as closely as possible the behavior of OML and CMAX weight functions. The
resulting \textit{OML-Like} and \textit{CMAX-Like} weight functions are shown in Figure (4). Then we
employed the improvised weight functions in our FML heuristic to study the effect on the
obtained lifetime values (Figure (5)). As can be observed, the lifetime results had an adverse
effect by using the improvised functions. Thus it can be concluded that a careful weight
function design for FML is one of its strengths that helped in achieving better lifetime.

5 The Fuzzy Multiobjective Routing Algorithm (FMO)

In order to incorporate the minimum energy consumption objective in our routing algorithm, another linguistic variable \textit{required energy} along an edge \(e(v_i, v_j)\), and a corresponding fuzzy set \textit{low energy consumption} are defined. A fuzzy membership function (Figure (6)) is proposed to map a value of the variable \textit{required energy} to its corresponding fuzzy minimum
energy membership value \( \mu_{me}^{ij} \). As may be seen, the function assigns the lowest (highest) membership value to an edge requiring the maximum (minimum) transmission energy among all the neighboring edges. This behavior of membership function encourages the selection of such edges that require lesser transmission energy. The lowest membership value may be altered by adjusting the value of algorithmic parameter \( \Delta \). An expression for the fuzzy minimum energy membership function is given by Equation (9).

\[
\mu_{me}^{ij} = 1 + \frac{(\Delta - 1) \times TX_{ij}}{\max(TX_{ij})}
\]

where

\[
\max(TX_{ij}) = \max_j \{TX_{ij}\} \quad \forall \ j \ s.t. \ v_j \in N_i
\]

\[
\min(TX_{ij}) = \min_j \{TX_{ij}\} \quad \forall \ j \ s.t. \ v_j \in N_i
\]

In order to formulate a fuzzy multiobjective aggregation function for an edge, the following fuzzy rule is proposed:

**IF** an edge

\( \text{has start node with high lifetime} \quad \text{AND} \)

Figure 6: A depiction of the fuzzy membership function for minimum energy.
requires low energy consumption

THEN it is a good edge.

The above fuzzy rule translates to the following ‘and-like’ function by employing the OWA operator:

\[
\mu_{ij} = \beta \times \min(\mu_{ij}^{lt}, \mu_{ij}^{me}) + (1 - \beta) \times \left( \frac{\mu_{ij}^{lt} + \mu_{ij}^{me}}{2} \right)
\] (12)

where \( \mu_{ij} \) is the fuzzy multiobjective membership value of the edge \( e(v_i, v_j) \), and \( \beta \in [0, 1] \) is a constant. As can be observed, the above OWA function due to the term \( \beta \times \min(\mu_{ij}^{lt}, \mu_{ij}^{me}) \), asserts a preference on the objective having the least membership value. Also, it may be noted that due to nature of our carefully designed membership functions, the minimum value of \( \mu_{ij}^{lt} \) is 0, whereas that of \( \mu_{ij}^{me} \) is \( \Delta > 0 \). Therefore it is easy to infer that if the residual energy (and the corresponding lifetime membership value \( \mu_{ij}^{lt} \)) is high, a similar preference level is given to both the objectives; otherwise if \( \mu_{ij}^{lt} < \Delta \), the preference shifts to the maximum lifetime objective. As a conclusion, the value of parameter \( \Delta \) affects the relative preference of the two routing objectives.

The proposed fuzzy multiobjective routing algorithm (Figure (7)) operates as follows: When a routing request \( r_h(v_m, v_n) \) is initiated, a fuzzy lifetime membership value is computed for each edge using Equation (4). Also, a fuzzy minimum energy membership value for each edge is computed using Equation (9), followed by computation of a multiobjective membership value using Equation (12). Then a weight is assigned to each edge using
Figure 7: The fuzzy multiobjective (FMO) routing algorithm.

Following the weight assignment, the multiobjective path \( p_h \) between \( v_m \) and \( v_n \) is found using Dijkstra’s shortest path algorithm [8].

### 6 The Distributed FML Algorithm

The proposed fuzzy routing algorithms described in the earlier sections are source-directed in their working, and thus require a centralized implementation. This implies a considerable protocol control overhead since, during each shortest path search phase, a possibly large number of control packets need to be sent to the source node from each of the intermediate nodes on the partially searched path. This fact results in poor scalability of the routing algorithm which may hinder its practical application to large-scale WSNs. In addition, the
need for transmission and reception of a large number of control packets implies an implicit energy consumption overhead that has a direct adverse effect on the energy conservation as well as the lifetime maximization objectives. Therefore, for a better scalability and a lower energy consumption overhead, we propose a distributed fuzzy maximum lifetime (FML) algorithm shown in Figure (8).

\[ G(V, E) \text{ is the given directed graph} \]
\[ \text{For each routing request } r_h(v_m, v_s) \]
\[ \quad \text{For each edge } e(v_i, v_j) \text{ in } V \]
\[ \quad \text{Compute fuzzy lifetime membership } \mu_{ij} \]
\[ \quad \text{Assign weight } w_{ij} = 1 - \mu_{ij} \]
\[ \text{End For} \]
\[ \text{Setup Phase:} \]
\[ \text{The sink } v_s: \]
\[ \quad \text{Initiate by broadcasting a setup packet} \]
\[ \text{Each receiving node:} \]
\[ \quad \text{Update forwarding table FT (route to } v_s, \text{ cost to } v_s) \]
\[ \quad \text{If not the source } v_m: \]
\[ \quad \quad \text{Broadcast FT information to upstream neighbors} \]
\[ \text{End If} \]
\[ \text{Data Transmission Phase:} \]
\[ \text{The source } v_m: \]
\[ \quad \text{Send the packets to the best immediate neighbor} \]
\[ \text{Each receiving node:} \]
\[ \quad \text{If not the sink } v_s: \]
\[ \quad \quad \text{Make the best local forwarding decision using FT} \]
\[ \text{End If} \]
\[ \text{Compute the minimum node energy in } G(V, E) \]
\[ \text{IF a node has run out of energy, stop.} \]
\[ \text{End For} \]

Figure 8: The distributed FML routing algorithm.

The distributed FML algorithm works as follows: Each node in WSN computes its ‘fuzzy membership value’ using the fuzzy maximum lifetime membership function given by Equation (4). For each routing request, a setup phase is triggered during which, a sink nodes initiates by broadcasting a setup packet to its immediate neighbor nodes. On receiving the said packet, each of the neighbor nodes registers itself as the sink’s neighbor and updates its forwarding table (FT) by storing the route to the sink node along with an accumulated cost of the route to the sink node. Then it broadcasts a setup packet containing the above stated
routing information to its following upstream neighbors that perform the similar updates in their forwarding tables. This local broadcast process is carried out until the source node (that initiated the routing request) is reached. As a result of the setup phase, each node discovers a route as well as an accumulated cost to the sink node.

The setup phase is followed by the data transmission phase during which, starting from the source node, each intermediate node makes a local forwarding decision based on its forwarding table and thus forwards the data packet to the best downstream neighbor (the one having the least accumulated cost to the sink). This forwarding scheme continues until the sink node is reached. The route found is used for the entire duration of the routing request (all source to sink packets in the routing request are routed along the same route).

7 Performance Evaluation and Discussions

We present a comprehensive evaluation of the proposed fuzzy routing algorithms. For the comparison purpose, we implemented two other online routing heuristics namely, the online maximum lifetime (OML) [18] and the CMAX [11] heuristics.

The performance metrics used for comparison are network lifetime and average energy consumption. Network lifetime is computed as the number of successful sessions (end to end, i.e., source to sink packet deliveries) before any sensor node in WSN runs out of its battery (Recall from Section 2 that network lifetime ends as soon as the first node in WSN dies). Average energy consumption is computed as the total energy expended during a routing request divided by the number of sessions (source to sink packet deliveries) in the routing request. Total energy expended during a routing request refers to the sum of transmission
energies and reception energies of all the intermediate nodes on the path selected for the routing request. It is worth mentioning that even the transmission energy of the source node is added to the above sum. However, the reception energy at the sink node is not included since the sink node is usually a base station not powered by a battery.

Our simulation setup consists of 2-dimensional grid of size $X \times Y$ populated with $n$ sensor nodes randomly deployed within the region. In our simulations, sink nodes are assumed to have infinite energy (usually the sink node is a base station powered by a fixed power source) and predetermined locations, whereas all other nodes (including the source nodes) have an equal level of initial residual energy equal to $\sigma$. A sample random WSN topology consisting of 50 nodes with a single sink node spread in a $25m \times 25m$ region is shown in Figure (9).

![Figure 9: A sample random topology of 50 nodes. An ‘o’ indicates a sensor node, whereas ‘X’ shows a sink node.](image)

In the scope of a routing algorithm, we assume a MAC layer with lower energy losses due to bounded number of retransmission attempts (it may appear to be an optimistic assumption considering typical wireless MAC protocols that offer a probabilistic nature of contention-based channel access mechanism, but some recently proposed MAC protocols are able to offer deterministic contention-free or lesser-than-typical contention channel access guarantees [5,21,22]).
Our traffic model is online, where there are no known source nodes or predetermined traffic flows; rather we assume that any of the non-sink nodes on detecting an event in its sensing range is able to initiate a routing request in order to start sending its sensed data to a sink node.

7.1 Finding Values for the Simulation Parameters

There are number of simulation parameters to which, we need to assign suitable values. For wireless medium related parameters including $A$, $B$ and $m$, we used typical values as were used by many other works [4,10]. We randomly placed between 30 to 100 sensor nodes in a fixed-size square region measuring $(X \times Y)$ $m^2$, whereas the transmission range of a sensor node’s radio was varied from $7m$ to $15m$. A list of the simulation parameters and their values is given in Table 1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma$</td>
<td>1 $J$</td>
</tr>
<tr>
<td>$A$</td>
<td>100 nJ/bit</td>
</tr>
<tr>
<td>$B$</td>
<td>50 pJ/bit/m$^4$</td>
</tr>
<tr>
<td>$m$</td>
<td>4</td>
</tr>
<tr>
<td>$X$</td>
<td>25</td>
</tr>
<tr>
<td>$Y$</td>
<td>25</td>
</tr>
<tr>
<td>$n$</td>
<td>{30, 40, 50, . . . , 100}</td>
</tr>
<tr>
<td>$r_t$</td>
<td>{7, 8, . . . , 15}</td>
</tr>
</tbody>
</table>

In addition, the FML scheme has two algorithmic parameters ($\alpha$ and $\gamma$), for which the best values (e.g., those resulting in the maximum obtained lifetime) were determined empirically. For this purpose, we experimented with all values for $\gamma \in \{0.1, 0.2, . . . , 0.9\}$ using numerous possible combinations of values for $n$ and $r_t$, and the best value found for $\gamma$ was 0.9, and hence we used this value in the rest of our simulations. Then, we conducted a series of experiments
to determine the best value for \( \alpha \). We tried 10 values for \( \alpha \in \{0.1, 0.2, \ldots, 1.0\} \), and for each value, we obtained the lifetime results using 10 random 50-node topologies, whereas 10 random routing sequences were generated for each topology. In other words, for each value of \( \alpha \), we averaged our results over 100 runs. The results obtained are shown in Figure (10). As can be seen, \( \alpha = 0.2 \) obtained marginally best lifetime, and hence we set the value of parameter \( \alpha \) at 0.2 throughout our simulations. In our implementations of OML and CMAX, we used the best value for the parameter \( \lambda = 10^{11} \), as reported by the OML authors in [18].

![Figure 10: Effect of \( \alpha \) on the lifetime metric obtained by FML routing algorithm. It may be noted that setting \( \alpha = 0.2 \) results in the highest lifetime value.](image)

### 7.2 Performance Comparison of the FML Scheme with the OML and the CMAX Heuristics

We execute runs of FML and OML on 10 randomly generated 100-node topologies, and performed a comparison in terms of lifetime metric. Table (2) lists the lifetime values obtained by both approaches along with some statistical measures. Each row corresponds to a topology and 10 random sequences of routing requests were generated for each topology. The columns Average and SD respectively, show the average and the standard deviation of
lifetime values obtained over 10 sequences, whereas the last two columns respectively, show
the average and standard deviation of percentage improvement achieved by the FML over the
OML. It may be noted that the FML consistently outperforms the OML by a considerable
margin ranging from 10% to 20% (average is about 14%). In other words, this experiment,
carried out on a variety of topologies and routing request sequences, exhibits that the use of
FML prolongs the network lifetime by a allowing a considerable more number of sessions.

Table 2: A statistical comparison of lifetime results obtained by FML and OML for 10
different topologies. The FML achieves lifetime values that are 10% to 20% better than
those achieved by the OML.

<table>
<thead>
<tr>
<th>Topology</th>
<th>Average</th>
<th>SD</th>
<th>Avg. Imp. (%)</th>
<th>SD Imp. (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FML</td>
<td>8020</td>
<td>382.39</td>
<td>11.15</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>7220</td>
<td>229.98</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>FML</td>
<td>9000</td>
<td>740.87</td>
<td>11.87</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>8050</td>
<td>596.75</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>FML</td>
<td>8480</td>
<td>507.28</td>
<td>10.61</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>7670</td>
<td>188.86</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>FML</td>
<td>9480</td>
<td>1018.50</td>
<td>10.67</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>8610</td>
<td>479.47</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>FML</td>
<td>7670</td>
<td>271.01</td>
<td>13.30</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>6780</td>
<td>261.62</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>FML</td>
<td>7870</td>
<td>371.33</td>
<td>14.34</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>6910</td>
<td>510.88</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>FML</td>
<td>9260</td>
<td>400.56</td>
<td>15.01</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>8060</td>
<td>313.40</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>FML</td>
<td>10930</td>
<td>992.25</td>
<td>16.85</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>9380</td>
<td>850.88</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>FML</td>
<td>11190</td>
<td>636.75</td>
<td>20.45</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>9320</td>
<td>723.88</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>FML</td>
<td>9380</td>
<td>385.28</td>
<td>14.45</td>
</tr>
<tr>
<td></td>
<td>OML</td>
<td>8220</td>
<td>480.28</td>
<td></td>
</tr>
</tbody>
</table>

Now we include CMAX heuristic in our comparisons. Park and Sahni [18] have already
demonstrated the supremacy of OML over CMAX, but it would be interesting to study where
FML stands in comparison to CMAX as well. In case of each routing approach, all the results
reported are averaged over 100 runs – 10 random network topologies were generated, and 10 random request sequences were generated for each topology.

Figure 11: A comparison of FML to OML and CMAX in terms of (a) network lifetime and (b) average energy consumption for a varying transmission radius \( r_t = \{7, 8, \cdots, 15\} \) for a 50-node WSN topology. The FML is able to achieve the highest lifetime values as well as the least average energy consumption values.

Figure 12: A comparison of FML to OML and CMAX in terms of (a) network lifetime and (b) average energy consumption for a varying transmission radius \( r_t = \{7, 8, \cdots, 15\} \) for a 100-node WSN topology. The FML is able to achieve the highest lifetime values as well as the least average energy consumption values.

7.2.1 The Effect of the Transmission Radius

We study the effect of transmission radius on the network lifetime obtained by FML and the two other approaches. Figures (11) (a) and (b), respectively, show the lifetime and
the average energy consumption obtained by all the three approaches in case of 50 nodes for a range of transmission radius values. Again, it may clearly be seen that FML obtains better lifetimes than both OML and CMAX. However, the lifetime obtained by FML and OML increases until $r_t = 12$ and then remains constant, whereas in case of CMAX, lifetime reaches a maximum value at $r_t = 9$ and then starts falling sharply. It should be noted that very similar trends of OML and CMAX are reported by Park and Sahni [18]. The initial increasing trend in the lifetime can be explained as follows: with the increase in transmission radius $r_t$, each node is able to communicate farther, and to discover more neighboring nodes, and thus a higher network connectivity is resulted that offers more routing choices at each node. As a result, a routing algorithm is able to find more cost-effective (maximal lifetime) routes. However at the same time, energy consumption also rises because a node now may transmit to a farther neighbor that requires more transmission energy. Also in terms of energy consumption, it can be seen that for the entire range of transmission radius values used in this experiment, routes found by FML consume considerably lesser energy than those
found by OML or CMAX.

Figures (12) (a) and (b), respectively, show the lifetime and the average energy consumption obtained by all the approaches in case of 100 nodes for a range of transmission radius values. The lifetime as well energy consumption trends here are similar to ones obtained for 50-node case discussed above. However, the worth mentioning fact is that the improvement margin of FML over OML is even larger in case of 100-nodes. This is a healthy behavior that FML performs even better with the increasing network density which is a typical feature of real life WSNs.

7.2.2 The Effect of the Node Density

We study the effect of sensor node density on the results by randomly deploying \( n \in \{30, 40, \ldots, 100\} \) sensor nodes in a region of fixed size. Figures (13) (a) and (b), respectively, show the lifetime and the average energy consumption obtained for a varying node density in case of \( r_t = 12 \). It may be seen that FML was able to obtain higher lifetime values than the other two approaches. Moreover, with the rising node density, FML shows a consistently sharper increasing trend in the obtained lifetime. On the other hand, OML is not able to show a similar competitive increasing trend at higher transmission radii, and thus the performance difference grows to be higher. As far as the energy metric is concerned, FML clearly proved to be more energy frugal than OML and CMAX approaches. It can be noted that FML exhibits a relatively sharper decreasing trend in terms of energy consumption as the node density grows.
Figure 14: A comparison of FML and OML in terms of ‘current energy’ distribution at the end of network lifetime for (a) \( n=50 \) and (b) \( n=100 \). The results give an insight into the better performance of the FML: longer lifetimes are obtained by achieving a more uniform consumption pattern across the network.

7.2.3 A comparison in Terms of Energy Consumption Distribution

Moreover, we conducted a set of experiments to obtain an insight into energy consumption pattern across the nodes in a topology. Figures (14) (a) and (b) show a comparison of current energy distribution at the end of WSN lifetime obtained by FML and OML for 50 and 100 node network, respectively. In both the figures, X-axis shows the normalized current energy level of nodes, whereas Y-axis shows the percentage of total nodes having a certain level of current energy at the end of network lifetime. For both topologies, we find that FML behaves better than OML as in the FML case, at the end of network lifetime, the number of nodes having lower current energy levels, i.e., in range \([0.1, 0.3]\), is much larger than that obtained in the OML case. For instance, in case of 50-node network, almost 40% of the nodes have their remaining energy levels at 30% or below, whereas the same measure is just around 20% in case of OML. This means that a considerably larger number of nodes have depleted 70% of their battery power before reaching the point when a node in the network runs fully out of its battery (which translates to end of lifetime as per the definition used in
this work). In other words, FML achieves a prolonged network lifetime by choosing a more uniform distribution of routing paths (and thus a more uniform energy consumption pattern across all the nodes in the network). On the other hand, OML reaches the end of lifetime much earlier, i.e., a node runs out of battery as early as only 20% of the nodes have depleted 70% of their battery power.

Again as was observed in the previous comparisons discussed above in this section, FML behaves relatively better for larger network sizes, the similar current-energy distribution plot for a 100-node topology shows an even a bigger performance gap between FML and OML. It is observed that in the FML case, at the end of network lifetime, a large portion (about 40%) of nodes have their current energy levels at 30% or lower, whereas in the OML case, a only 12% on the nodes fall in this range. Moreover, it can be seen that a large portion of nodes have current energy levels between 50% and 90% which is not a good sign at the end of network lifetime since the routing algorithm should have tried to exploit the energy reserve at those nodes to ensure a prolonged lifetime.

Another way of looking at the energy consumption pattern is to compare the *average current levels* of the nodes obtained by FML and OML at the end of network lifetime as depicted by Figure (15). The figure shows the values of the said metric for network sizes ranging from 50 to 100 nodes. As can be observed, at the end of lifetime, FML consistently achieves a considerably lower average energy level than the OML does. Again the difference becomes wider with the increasing network size (from about 10% for 50-node network to about 15% for 100-node network). This trend also supports our claim regarding the FML mechanism that it exercises a more uniform energy consumption across the network nodes.
and thus extracts its success from this fact and hence delays the time to the failure of first node in the network.

Figure 15: A comparison of FML and OML in terms of ‘average current energy’ at the end of network lifetime for a varying node density \( n = \{30, 40, \ldots, 100\} \). It may be seen in all the cases that FML reaches the end of lifetime only when the average current energy of the nodes in WSN has fallen to a considerably lower level than that observed in the case of OML.

### 7.3 FMO Performance: Effect of Multiobjective Routing on the Lifetime and the Energy Consumption

In this section, we present the results of FMO approach with a view of studying the effect of incorporating the minimum energy consumption objective on the lifetime maximization process. To start with, a series of experiments were conducted for determining suitable values for the parameters \( \Delta \) and \( \beta \) using various combination values for the two parameters, and the lifetime and the average energy consumption values obtained from FMO are shown in Figures (16) (a) and (b), respectively. As may be observed, \( \beta = 0.2 \) results in the maximum lifetime as well as the least energy consumption, hence we set its value at 0.2 in all the subsequent runs. Also it may be noted that for smaller values of \( \Delta \), the preference of the energy consumption objective is high, and as a consequence, the average energy consumption
is low, but at the same time, the network lifetime also falls. Hence, there is a visible tradeoff between the two routing objectives, and the parameter $\Delta$ can be used to achieve a desired balance.

![Figure 16: The effect of choosing various possible combinations of the two FMO parameters, namely $\Delta$ and $\beta$, on (a) Lifetime and (b) average energy consumption obtained by FMO. The FMO multiobjective framework is able to offer a flexible tradeoff between the two optimization objectives (lifetime vs. energy consumption): Setting $\Delta = 0.2$ achieves the least energy consumption values, but at the same time results in the worst lifetime values.](image)

Next, we conduct experiments to study the effect of $\Delta$ in a scenario of varying node density. Figures (17) (a) and (b) show the lifetime and the average energy consumption, respectively, obtained by FML and FMO (for $\Delta \in \{0.2, 0.4, 0.6, 0.8\}$) for a varying node density $n$. Again a lifetime-energy consumption tradeoff is clearly visible in the results, i.e., for instance, FMO obtains the least energy consumption when $\Delta = 0.2$, but also obtains the least lifetime in this case. Thus, FMO is able to offer a flexible control over choosing a desired balance between the two routing objectives.
Figure 17: A comparison with the FML, in terms of (a) Lifetime and (b) average energy consumption, to the FMO for various values for the parameter $\Delta$ for a varying node density $n$. The results again show the flexible lifetime vs. energy consumption tradeoff offered by the FMO.

7.4 Reducing Control Overhead: A Comparison with Distributed FML

The major objective of designing the distributed FML algorithm is to reduce the control overhead that is inherent in the proposed FML algorithm due to the use of centralized Dijkstra’s shortest path search process. A high control overhead poses a scalability question to algorithm, and thus may prohibit its application to large-scale WSNs. The proposed distributed FML tries to counter the problem by cutting down the number of control packets that need to be sent across the network.

In this section, we perform a quantitative study to assess performance gain of the distributed FML in terms of control traffic reduction. We conduct simulations to compute the amount of control bits as well as data bits transmitted in case of centralized and distributed FML algorithms. The computations were recorded over a range of sensor node density, and the results are plotted in Figure (18). As expected, the centralized implementation has a higher control overhead than the distributed scheme. Moreover, the centralized case
overhead rises much sharply with an increasing number of sensor nodes, whereas in the distributed case, the overhead increase is reasonable. As a result, the performance gap grows to a significant amount for a 100-node network.

It should be noted that this large difference in the number of transmitted control bits directly translates to a considerable performance difference in terms of the magnitude of energy consumption, and thus the effect on lifetime objective. Therefore, the analysis in this section shows that distributed FML algorithm is a strong candidate for maximum lifetime routing in large scale WSNs.

Figure 18: A comparison between the control overhead of the centralized and the distributed FML schemes in terms of the amount of the control traffic needed during the path search phase. The distributed FML successfully lowers the control overhead by a considerable margin.

8 Related Work

Comprehensive reviews of existing routing techniques for WSNs have been presented by Perillo et al. [19] and Al-Karaki et al. [2]. Al-Karaki et al. classified the existing protocols into many categories based on two criteria namely, network structure and protocol operation. However the existing energy-aware routing schemes for WSNs can be broadly classified on
the criterion of routing objective into minimum energy (ME) routing and maximum lifetime (ML) routing.

Most of the earlier reported attempts related to energy efficient routing in WSN fall in the category of minimum energy routing. Minimum energy (ME) routing algorithms and protocols focus on finding a ‘shortest’ or ‘minimum energy’ path between a source node and the sink node. The main objective of such techniques is to minimize the total amount of energy consumption for a given routing session. The biggest drawback of typical ME routing schemes is that battery power of nodes on the favorite paths is quickly drained. As a consequence, a WSN may prematurely suffer from a network partitioning or at least a considerable degree of loss in sensing coverage. In other words, these schemes usually do not cater for a balanced energy consumption pattern among sensor nodes of a WSN.

Maximum lifetime (ML) routing algorithms on the other hand strive to delay as much as possible the time until sensor nodes runs out of battery. The basic philosophy of this class of techniques is to favor routes consisting of nodes with higher residual energy, though various ML routing algorithms may use different underlying path selection mechanisms. For instance, a simple comparison of ME and ML routing algorithms is shown in Figure (19). Three of the possible paths between source and sink nodes are shown with their total available current energies (ce). The number shown along an edge is the energy required to transmit along the edge. A ME routing scheme selects path2 as it results in minimum total energy consumption, whereas a possible ML routing algorithm may select path3 depending on max-min(ce) criterion, i.e., the path along which the minimum ‘ce’ is maximum among all the paths.
Figure 19: A high-level comparison between mechanism of the minimum energy and the maximum lifetime routing techniques (adopted and modified from [1]).

We present a brief review of some of the related existing maximum lifetime routing schemes for WSN. An online routing algorithm – namely CMAX – is presented in [11]. Park and Sahni [18] presented an online maximum lifetime heuristic (OML).

Misra et al. [17] proposed MRPC, which is an online routing algorithm that selects the path which has the maximum available lifetime whereas the lifetime of a path is computed based on the minimum of the residual energies of the nodes that lie along that path. Max-min $z \cdot P_{min}$ [3] is another ML routing algorithm that first finds the minimum energy path $P_{min}$ between a source and the sink. Then in an effort to balance the tradeoff between the minimum energy and the maximum lifetime objective, it tries to search a path along which the total energy consumption is no more than $z \cdot P_{min}$ where $z$ is an algorithmic parameter. Sungwook et al. [12] present an online energy-efficient routing scheme that claims to take real-time online route decisions for QoS-constrained WSN application scenarios.

A number of analytical solution approaches are also found where the WSN lifetime problem is formulated as a liner program and then solved by using any of the LP solution techniques such as gradient projection method and others [4, 15, 23]. These LP techniques require a priori knowledge of the future routing requests and data rates. Cui et al. [6] present a utility
based scheme to solve ML routing problem.

Multipath routing approaches have also been a popular choice due to their underlying philosophy of distributing the data flow among multiple paths for a single routing request between a source and sink. A number of multipath algorithms have been developed [2]. The energy-aware routing (EAR) protocol [20] is a distributed multipath protocol that discovers and maintains multiple paths with their accumulated costs by local flooding. The best ML path is chosen probabilistically based on the accumulated path cost. Lu et al. present an energy-aware multipath routing protocol in [14]. Srinivasan et al. [24] proposed a data-centric routing algorithm in which, a node decides to participate in packet forwarding only if it has enough available energy.

9 Conclusions

A number of online routing scheme are presented for maximum lifetime routing in WSNs. A critical effort is made for identifying the shortcomings of the edge cost functions used in some of the existing online routing schemes. The comparative analysis of the weight functions provides us an insight to the better operating regions in terms of the relation between the residual energy and the edge weight, and hence leads to the design of more effective behavior of the cost functions.

The role of the fuzzy membership functions and operators is to offer a flexible mean of explicitly implementing the desired behavior of the cost functions, but in general, any non-fuzzy approach may also be used as a tool for achieving the task. However, fuzzy logic may be useful in formulating the cost of the objectives involved in many related optimization
problems in WSNs.

For the sake of giving a fine and complete view to the reader, some minor concluding remarks are worth mentioning. We did not investigate the sensitivity of the performance to the fine tuning of the proposed fuzzy membership functions – For instance what if two or more breakpoints (threshold points) are introduced instead of one in the maximum lifetime membership function. We did not evaluate the effect of some simulation settings such as varying the number of sink nodes in the network, placing the sink node(s) at various geographic locations, varying the area of the sensor field, to name a few.

The multiobjective optimization framework can be easily extended to incorporate other critical routing objectives. The fully distributed implementation approach presented for the FML scheme can be similarly extended to the FMO scheme.

In the immediate future, we plan to extend our work on multiobjective routing to include other crucial metrics such as latency and capacity. We shall also look into devising multipath routing approaches for seeking further improvement in terms of the lifetime and energy consumption metrics. Further, we will consider flow prioritization to determine schemes for dropping certain requests when more important requests arrive. Other aspects of interest include context-aware routing and co-design of energy-efficient routing and MAC protocols using fuzzy functions. Last but not least, we plan on investigating for defining novel/atypical but relatively realistic lifetime metrics with a view of measuring the WSN lifetime more accurately.
A short introduction to fuzzy Logic

Fuzzy Logic is a mathematical discipline invented to express human reasoning in rigorous mathematical notation. Unlike classical reasoning in which, a proposition is either true or false, fuzzy logic establishes an approximate truth value of a proposition based on linguistic variables and inference rules. The value of a linguistic variable is expressed as words or sentences in a natural or an artificial language [29]. By using hedges like ‘more’, ‘many’, ‘few’, and connectors like AND, OR, and NOT with linguistic variables, an expert can form rules, which govern the approximate reasoning.

.1 Fuzzy Membership Functions

In the context of crisp sets, a certain element is either a member or a nonmember of a set (in other words, the membership value is either 1 or 0), whereas in fuzzy logic, a certain element may have a partial membership in a set (membership value is in the range [0,1]). A fuzzy membership function is used to compute the membership value corresponding to a given value of the linguistic variable. The membership function can be designed in a flexible way in order to reflect the desired goodness behavior of an objective. For instance, as can be seen in Section 3, we design a fuzzy membership function for computing lifetime membership value $\mu_{lt}$ corresponding to the residual energy value of a node, where a higher value of $\mu_{lt}$ means a higher goodness level of the ‘lifetime’ objective.
2 Fuzzy Ordered Weighted Averaging (OWA) Operator

As in the case of the proposed fuzzy multiobjective (FMO) routing algorithm, there are multiple optimization goals that we want to optimize simultaneously, we need to formulate a multiobjective cost aggregation function that may reflect the effect of all the objectives collectively as a scalar value. A common approach is to use a weighted sum based cost function. Generally, this type of cost function is not sufficient to reach the desired solution due to certain reasons [26,28]: the formulation of multiobjective cost functions do not desire pure ‘anding’ or ‘oring’ kind of aggregation operation. The formulation of multiobjective decision function neither desires a pure ‘anding’ of t-norms, nor a pure ‘oring’ of s-norms.

Fuzzy logic offers a fuzzy aggregation operator, namely the Ordered Weighted Averaging (OWA) [26], as an alternative to weighted sum, for designing a multiobjective cost function. This operator allows easy adjustment of the degree of ‘anding’ and ‘oring’ embedded in the aggregation. ‘Or-like’ and ‘And-like’ OWA operators for two fuzzy sets A and B are implemented as given in Equations (14) and (15), respectively.

\[
\mu_{A\cup B}(x) = \beta \times \max(\mu_A, \mu_B) + (1 - \beta) \times \frac{1}{2}(\mu_A + \mu_B) \tag{14}
\]

\[
\mu_{A\cap B}(x) = \beta \times \min(\mu_A, \mu_B) + (1 - \beta) \times \frac{1}{2}(\mu_A + \mu_B) \tag{15}
\]

where \(\mu_A\) and \(\mu_B\) denote the fuzzy membership values in fuzzy sets A and B respectively, whereas \(\beta\) is a parameter in the range \([0, 1]\) that controls the degree to which OWA operator resembles a pure ‘or’ or a pure ‘and’, respectively.
Acknowledgement

This work was supported by the Natural Sciences and Engineering Research Council of Canada under grant STPGP 322208-05.

References


