Check for updates

Binary Search Based PSO for Master Node Enumeration and Placement in a Smart Water Metering Network

Clement N. Nyirenda[1(✉)] and Samson G. Nyirongo[2]

[1] ISAT Laboratory, University of the Western Cape, Robert Sobukwe Road, Bellville, Cape Town 7535, South Africa
cnyirenda@uwc.ac.za

[2] Department of Electrical and Computer Engineering, University of Namibia, Ongwediva, Namibia

Abstract. A Binary Search based Particle Swarm Optimization (BS-PSO) algorithm is proposed for the enumeration and placement of Master Nodes (MNs) in a Smart Water Metering Network (SWMN). The merit of this proposal is that it can simultaneously optimize the number of MNs as well as their locations in the SWMN. The Binary Search (BS) Mechanism searches a pre-specified range of integers for the optimal number of MNs. This algorithm iteratively invokes the PSO algorithm which generates particles based on the chosen number of MNs. The PSO uses these particles to determine MN coordinates in the fitness function evaluation process within the underlying SWMN simulation. The packet delivery ratio (PDR) is designated as the fitness value for the particle. Results for 10 BS-PSO optimization runs show that the median optimal number of MNs is 15 and that the mean PDR of 96% can be realized. As part of future work, more optimization runs will be conducted to enhance the generalization of the results. The extension of this concept to other optimization algorithms such as Differential Evolution will also be considered.

Keywords: Smart Water · Particle Swarm Optimization · Binary Search

1 Introduction

The rapid developments in ICT have triggered widespread interest in smart water networks (SWNs) [1–3]. From the operation perspective, the major driving force behind the adoption of SWNs is the desire by water utilities to cut down Non-revenue water (NRW). NRW refers to the water that has been produced and is lost before it reaches the customer. It has been reported that more than 60% of clean water is lost due to Non-revenue Water (NRW) [4]. Apart from helping utilities to cut down on NRW, smart water networks (SWNs) also enable them to achieve their quality, productivity, and efficiency targets while,

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2020
Published by Springer Nature Switzerland AG 2020. All Rights Reserved
R. Zitouni et al. (Eds.): AFRICOMM 2019, LNICST 311, pp. 104–118, 2020.
https://doi.org/10.1007/978-3-030-41593-8_8

on the other hand, improving customer service. By implementing SWNs, it is estimated that up to $12.5 billion could be saved annually [5]. According to [1], SWN incorporated the following solution areas: residential metering, water quality monitoring, leak detection, and energy management. This work focuses on SWN based residential metering, which is generally referred to as Smart Water Metering Network (SWMN) [6,7].

1.1 Master Node Location Optimization in Smart Water Metering Networks

Smart Water Metering Networks enable water utilities to collect water consumption data from homes remotely. This cuts out the use of a human meter reader thereby preventing manual data entry errors [8]. Furthermore, meter readings can be collected as frequently as possible thereby enabling customers to keep track of their water usage on a regular basis. This gives customers the ability to optimize their use of water leading to a reduction in water bills. Furthermore, SWMNs also help with early leakage detection, provision of more accurate water rates, and easy detection of water theft.

In SWMNs, the communication infrastructure is extremely important [9]; each smart meter must be equiped with capabilities to reliably and securely transmit meter readings to a central location at the utility. The SWMN adopts hierarchical topology [6]. A limited number of Master Nodes (MNs), which act as the sinks for a group of smart meters within their vicinity, are used to collect metering data from those meters. Smart meters that are far from the MNs use intermediate Smart Meters to relay their data to MNs. MNs relay the aggregated metering data to the Control Center, where the data is processed and analyzed. If the smart meters are few and very close to each other, one MN would be sufficient. This is, however, not the case in practical situations as areas covered by the smart meters are big. In these environments, data transmission from smart meters that are far from the MN may be very poor. Therefore, it may be necessary to implement several MNs in order to solve reachability problems. A high packet delivery ratio (PDR) can be achieved easily by using more MNs and placing them evenly in the network [6]. This will, however, come at a big cost.

In [10], Particle Swarm Optimization (PSO) [12], a population based stochastic optimization technique, has been used to determine the optimal locations of MNs in Smart Water Metering Network (SWMN). The SWMN was simulated in the TinyOS Simulator (TOSSIM) [14], with the Collection Tree Protocol (CTP) [15], as routing protocol. The number of MNs is set *a priori*. In the PSO algorithm, the particle's dimensions is twice the number of MNs in order to cater for the x and y coordinates of each MN. The parameters in the final *global best* particle can easily be extracted and scaled to denote the optimal locations of the MNs. At each PSO function evaluation instant, the underlying TOSSIM script, depicting the SWMNs along with the configured MNs, is triggered and the ensuing Packet Delivery Ratios (PDRs) are recorded. This technique yields good results, but it has a shortcoming in the sense that the determination of the optimal number of MNs is not incorporated in the optimization framework.

1.2 Contribution and Paper Organisation

The work proposed in this paper aims at solving the aforementioned problem by considering the MN placement problem as a two-fold optimization problem. In addition to finding optimal locations for the MNs, the number of MNs must also be optimized dynamically in the same optimization routine. To achieve this, the Binary Search based Particle Swarm Optimization (BS-PSO) algorithm is proposed. The optimisation of the number of MNs as well as the determination of their optimal location in the SWMN are conducted in one optimization routine. The Binary Search is implemented as high level algorithm that fixes the number of MNs, which denotes the search parameter, at every instant and invokes PSO algorithm, which functions as a low level algorithm. An arbitrary number of MNs is used depict the initial upper limit of the search range. At the start of the optimisation run, the PSO algorithm is invoked several times and the ensuing Packet Delivery Ratios (PDRs) are recorded. Then the initial upper limit is doubled and the PSO algorithm is invoked again. T-test is invoked to determine if there is a significant difference between the new upper limit and old one. If there is no significant difference, the old upper limit is construed to the upper limit of the BS approach. Otherwise, the new upper limit is doubled again and the statistical comparison is conducted again until there is no significant difference between the two potential upper limits; the lower one is always chosen. Once the upper limit is fixed, the BS algorithm is employed in order to determine the optimal number of MNs as well as their locations.

The rest of the paper is organized as follows. The SWMN topology is given in Sect. 2. An overview of Particle Swarm Optimization (PSO) is given in Sect. 3. The implementation of the BS-PSO algorithm for MN enumeration and location optimization is presented in Sect. 4. Section 5 presents the experimental setup, the results and discussion, and lastly, conclusions in Sect. 6.

2 Smart Water Metering Network Topology

The Smart Metering concept refers to the use of Smart Meters (SMs) in the collection of water usage data from clients' households. SMs, which are located at the customer's premises, measure water consumption and communicate their readings at regular intervals to the utility [11]. The utility uses this information for monitoring and billing purposes. Unlike conventional metering systems, there is no need for manual meter readers. Allowing each SM to be transmitting its data to the central location is, however, not a cost effective and scalable solution. As a result, the general trend is to deploy hierarchical network topologies in Smart Water Metering Networks [13]. Figure 1 shows a typical topology that is deployed in networks of this type.

The network consists of the following generic system components: Wireless Sensor Nodes (WSNs), Master Nodes (MNs), and the Control and Monitoring Center (CMC). In this work, Smart Meters (SMs) are deployed as a special case of WSNs. In addition to the traditional metering functions, the SMs in SWMNs are equiped with communication and routing functionalities. They are capable of

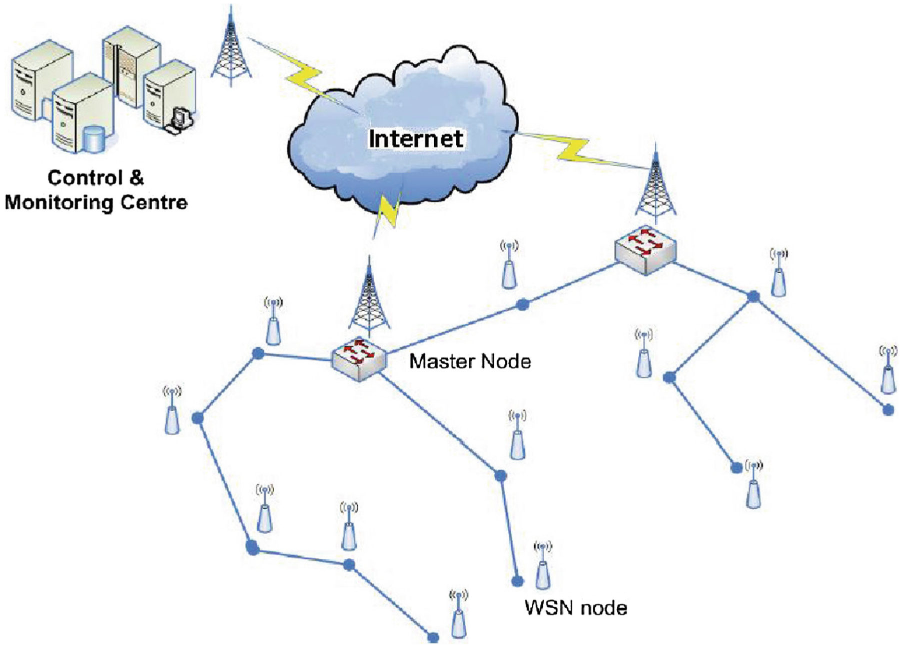

Fig. 1. Typical network topology in Smart Water Metering Networks [13].

transmitting their own metering data, as well as the metering data from neighboring SMs, to the Master Nodes at frequently. The MNs, which are equipped with more memory resources, processing and communication capabilities than the SMs, send the aggregated metering data to the CMC through a higher capacity communication link. MNs are generally fewer in number compared to SMs [11].

3 The Particle Swarm Optimization Algorithm

The Particle Swarm Optimization (PSO) algorithm was introduced by Kennedy and Eberhart in 1995. It draws inspiration from the social behavior of animals living in swarms, such as flocks of birds [12]. PSO is initialized with a population of particles that are generated randomly. Each particle denotes a candidate solution to a problem and is characterized by three main parameters in the search space: its current position, current velocity and the best position ever found by the particle during the search process. The particles fly in the search space in order to find the optimal solution. The trajectory of a particle is influenced by the particle's own experience as well as it's neighboring particles. For a population of N particles, the velocity of the i-th particle is updated at every iteration by using

$$v_i(t) = \omega * v_i(t-1) + c_1 r_1 (p_i^b - x_i(t)) + c_2 r_2 (g^b - x_i(t)), \tag{1}$$

where $i = 1, 2, \ldots, N$; c_1 and c_2 are constants denoting cognitive and social parameters respectively. The values of c_1 and c_2 are chosen in the range $[0.5, 2.5]$. They are applied to accommodate the influence of the particle's previous best position p_i^b and the best position g^b among all particles in the neighborhood of the i-th particle respectively. Parameters r_1 and r_2 are random numbers uniformly distributed within $[0, 1]$. Parameter ω, known as the inertia weight, helps to dampen the velocities of the particles to assist in the convergence to the optimum point at the end of the optimization iteration.

A further arbitrary parameter $V_m = (v_{m1}, v_{m2} \ldots v_{mD}) \in S$, where D denotes the dimensions of the search space S, was defined to be limit of the velocity. Whenever, a vector element exceeds the corresponding element of V_m, that element is reset to its upper limit. The position of each particle is updated at each iteration by using

$$x_i(t+1) = x_i(t) + v_i(t+1). \tag{2}$$

A basic algorithm for PSO technique is illustrated in Algorithm 1. The algorithm begins setting the values for N, c_1, c_2, ω, and G, which denotes the maximum number of iterations. Then it randomly generates an initial population of N particles and initial velocities for each particle. The fitness function values for all N particles are evaluated based on their current positions. The current positions of each particle are set as the personal best positions for the respective particles, and overall best position found so far is set as best solution for the swarm. Then the algorithm goes into the iterative process, which keeps on going on until G iterations are completed or until the stopping criterion is met. At each generation, the particle positions in the search space are updated using Eqs. 1 and 2; fitness function values of all particles are updated based on their new positions. If necessary, personal best and global best values are updated accordingly.

4 The Proposed BS-PSO Algorithm for MN Enumeration and Placement

The proposed Binary Search based PSO algorithm for Master Node (MN) enumeration and optimal placement is based on the approach proposed in [10]. A review of the MN placement optimization problem, conceptualized in [10], is, therefore, revisited in the next subsection before the components of the BS-PSO approach are presented in the subsequent subsection.

4.1 The MN Placement Optimization Problem

A Smart Water Metering Network is assumed to contain n_{sm} smart meters in a rectangular area defined by $L \times M$, where L and M are in meters. Let n_{sm} denote the number of MNs in the area. The location of each MN will be determined by the x and y coordinates. Therefore, the number of variable parameters that have to be encoded in the particle, in order to realize optimal performance, will

Algorithm 1. Particle Swarm Optimization (PSO)

Initialize the values for N, c_1, c_2, ω, and G, which denotes the maximum number of iterations
Randomly generate N particles
$F(g^b) \leftarrow 0$
for $i \leftarrow 1, N$ **do**
    Evaluate the fitness function value, $F(x_i)$, for each particle i
    $p_i^b \leftarrow x_i$
    $F(p_i^b) \leftarrow F(x_i)$
    if $F(x_i) > F(g^b)$ **then**
        $g^b \leftarrow x_i$
        $F(g^b) \leftarrow F(x_i)$
    end if
end for
while $t \leq G$ **do**
    Update v_i and x_i by using Equations 1 and 2
    for $i \leftarrow 1, N$ **do**
        Evaluate the fitness function value, $F(x_i)$, for each particle i
        if $F(x_i) > F(p_i^b)$ **then**
            $p_i^b) \leftarrow x_i$
            $F(p_i^b) \leftarrow F(x_i)$
        end if
        if $F(x_i) > F(g^b)$ **then**
            $g^b \leftarrow x_i$
            $F(g^b) \leftarrow F(x_i)$
        end if
    end for
    Stop the algorithm if a sufficiently good fitness function value is realized
end while

$2{*}n_{mn}$. For instance, if 5 MNs are used, the number of parameters in the particle will be 10. The goal of the optimization process is to that the best particle must achieve the highest packet delivery ratio (PDR), where PDR is defined by

$$PDR = P_r / P_s, \tag{3}$$

where P_r is the number of packets received at the MNs, excluding duplicate packets, while P_s denotes the number of packets sent by smart meters. For poorly located DAPs, the PDR will be very low due to reachability issues. The $2 * n_{mn}$ MN coordinate information can be encoded in a particle by using

$$\mathbf{p} = (p_0, p_1 \ldots p_{D-1}), \tag{4}$$

where p_0 and p_1 are the coordinates of the first MN; p_2 and p_3 are the coordinates of the second MN; p_D and p_{D-1} are the coordinates of the final MN; $D = 2{*}n_{mn}$. Each element p_i is defined within $[0, 1]$. To obtain the actual coordinates, the even indexed elements, denoting the x-coordinate, are multiplied by L while odd-indexed elements, denoting the y-coordinate, are multiplied by M.

4.2 BS-PSO Algorithm for MN Enumeration and Placement

In [10], many separate optimization procedures and statistical comparisons were done in order to determine the optimal number of MNs. The proposed approach aims at optimizing the number of MNs along with their placement in the SWMNs in an automated fashion as opposed to [10], where the number of MNs has to be fixed *a priori*. The architecture of the BS-PSO system, as shown in Fig. 2, is composed of three levels. Level 1 depicts the BS algorithm; Level 2 and Level 3 depict the PSO algorithm and the SWMN simulation respectively.

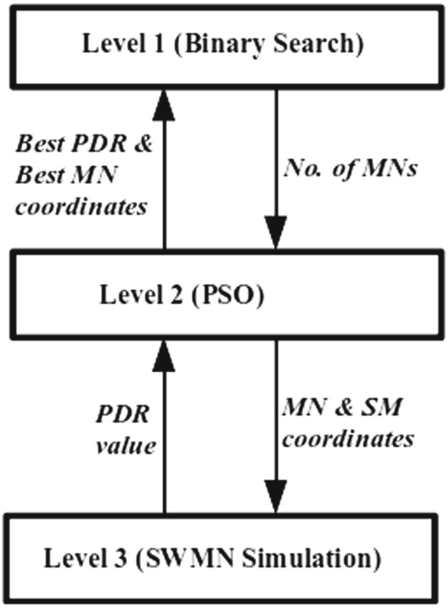

Fig. 2. BS-PSO architecture for SWMNs

The BS algorithm (Level 1) searches a pre-specified range of integers for the optimal number of MNs. At every chosen number of MNs m, it invokes the PSO algorithm (Level 2) k times. The PSO algorithm generates the particle of size $2 * m$. Every function evaluation in the PSO algorithm invokes the SWMN simulation at Level 3, using the MN coordinates extracted from the particle along with the fixed coordinates of the smart meters. The SWMN is implemented in Python in the Tiny Operating System Simulator (TOSSIM) [14], as presented in [10]. The function evaluation routine invokes the Python script several times and computes the average Packet Delivery Ratio (PDR), which denotes the fitness of the particle. Each PSO run generates the *global best* particle, which depicts the optimal locations of the MNs, along with its associated Packet Delivery Ratio (PDR) (see Eq. 3). Invoking the PSO algorithm k times for m MNs generates k particles along with their k associated PDRs. These PDR values are passed from Level 3 to Level 2 by using text files. The Packet Delivery Ratios (PDR) can be presented by using

$$\mathbf{PDR}(m) = (PDR_0(m), PDR_1(m) \dots PDR_{k-1}(m)), \tag{5}$$

where $PDR_i(m)$ denotes the PDR of the i-th run of the PSO algorithm, when the number of MNs is set to m; $i = 0, 1, \dots k-1$.

In the BS algorithm, there is a need to fix the lower bound s_{lb} and the upper bound s_{ub} of the search range. The lower bound s_{lb} is easily set to 1 since this is the lowest possible number of MNs allowed. The upper bound of the range s_{ub} can, however, not be fixed easily. Setting s_{ub} to the same value as the number of the smart meters, n_{sm}, would be a good idea but it will lead to higher computational costs. This is because it might take long time before the search process reaches the optimal value. In this proposal, s_{ub} is, therefore, set to a random integer value, which is uniformly drawn from the range [1, n_{sm}]. If s_{ub} is set to a value that is closer to n_{sm}, the behavior will be more or less similar to when $s_{ub} = n_{sm}$. The probability for such an occurrence is, however, lower than when s_{ub} is outrightly set to n_{sm}. On the other hand, if s_{ub} is set to a value that is lower than the desired optimal value, the search will suboptimally converge to s_{ub}. To cater for this latter condition, the Binary Search Upper Bound Initiliazation routine in Algorithm 2 is used. This algorithm scans the upper side of the initial s_{ub} by iteratively comparing $\mathbf{PDR}(\hat{s}_{ub})$ with $\mathbf{PDR}(s_{ub})$, where $\hat{s}_{ub}$ is the potential upper bound. When the condition, $\mathbf{PDR}(\hat{s}_{ub}) \sim \mathbf{PDR}(s_{ub})$, is realized, it implies that increasing the upper bound beyond s_{ub} will not yield any meaningful performance improvement. Therefore, s_{ub} is deemed to be the upper bound. It's associated particle, $\mathbf{p}(s_{ub})$, and $\mathbf{PDR}(s_{ub})$ are also saved for latter use in Algorithm 3.

Algorithm 2. Binary Search Upper Bound Initiliazation

Input: s_{lb} and n_{sm}
Output: $s_{ub}, \mathbf{p}(s_{ub}), \mathbf{PDR}(s_{ub})$
$s_{ub} \leftarrow \mathcal{U}(s_{lb}, n_{sm})$
Set potential upper bound,$\hat{s}_{ub} = 2 * s_{ub}$
if $\hat{s}_{ub} > n_{sm}$ **then**
    $\hat{s}_{ub} \leftarrow n_{sm}$
end if
while (1) **do**
    Invoke PSO in Algorithm 1, k times with $N = s_{ub}$ to generate $\mathbf{PDR}(s_{ub})$
    Invoke PSO in Algorithm 1, k times with $N = \hat{s}_{ub}$ to generate $\mathbf{PDR}(\hat{s}_{ub})$
    if $\mathbf{PDR}(\hat{s}_{ub}) \sim \mathbf{PDR}(s_{ub})$ **then**
        break
    else
        $s_{ub} \leftarrow \hat{s}_{ub}$
    end if
end while

The BS-PSO algorithm, presented in Algorithm 3 uses s_{lb} as the input. It calls Algorithm 2 in order to determine s_{ub}, $\mathbf{p}(s_{ub})$, and $\mathbf{PDR}(s_{ub})$. It then calls the PSO in Algorithm 1, k times with $N = s_{lb}$ to generate $\mathbf{PDR}(s_{lb})$. Since only one MN is used in the latter case, the PDR realized is normally expected to be the worst. Once the performance at these two extremes of the search range is fixed, the iterative search and PSO optimization process starts. At the start of every iteration, the midpoint s_{mid} of the search range is computed and the PSO algorithm is called several times to generate $\mathbf{PDR}(s_{mid})$. If $\mathbf{PDR}(s_{mid}) \sim \mathbf{PDR}(s_{ub})$, s_{ub} is updated to s_{mid}; $\mathbf{p}(s_{ub})$ and $\mathbf{PDR}(s_{ub})$ are updated to $\mathbf{p}(s_{mid})$ and $\mathbf{PDR}(s_{mid})$ respectively. Once that is done, if the termination condition, $\mathbf{PDR}(s_{mid}) \sim \mathbf{PDR}(s_{lb})$, is reached, the algorithm terminates. Otherwise, if $!(\mathbf{PDR}(s_{mid}) \sim \mathbf{PDR}(s_{ub}), s_{ub})$, s_{lb} is updated to s_{mid}; $\mathbf{p}(s_{lb})$ and $\mathbf{PDR}(s_{lb})$ are updated to $\mathbf{p}(s_{mid})$ and $\mathbf{PDR}(s_{mid})$ respectively, and the iterative search process continues.

Algorithm 3. BS PSO Algorithm

Input: $s_{lb}, s_{ub}, \mathbf{p}(s_{ub}), \mathbf{PDR}(s_{ub})$
Output: $s_{lb}, \mathbf{p}(s_{lb}), \mathbf{PDR}(s_{lb})$
Determine s_{ub} by invoking Algorithm 2
Invoke PSO in Algorithm 1, k times with $N = s_{lb}$ to generate $\mathbf{PDR}(s_{lb})$
while (1) **do**
  $s_{mid} \leftarrow \mathrm{int}(s_{lb} + s_{ub})/2$
  Invoke PSO in Algorithm 1, k times with $N = s_{mid}$ to generate $\mathbf{PDR}(s_{mid})$
  if $\mathbf{PDR}(s_{mid}) \sim \mathbf{PDR}(s_{ub})$ **then**
    $s_{ub} \leftarrow s_{mid}$
    $\mathbf{p}(s_{ub}) \leftarrow \mathbf{p}(s_{mid})$
    $\mathbf{PDR}(s_{ub}) \leftarrow \mathbf{PDR}(s_{mid})$
    if $\mathbf{PDR}(s_{mid}) \sim \mathbf{PDR}(s_{lb})$ **then**
      break
    end if
  else
    $s_{lb} \leftarrow s_{mid}$
    $\mathbf{p}(s_{lb}) \leftarrow \mathbf{p}(s_{mid})$
    $\mathbf{PDR}(s_{lb}) \leftarrow \mathbf{PDR}(s_{mid})$
  end if
end while

5 Experiment Setup and Results

5.1 The SWMN Topology and Simulation Parameters

Just like in [6,7,10], the SWMN topology, depicting the Tsumeb East area in Northern Namibia, is used. The area, which is a 400.5 m × 400 m square, has 140 houses, which implies that there are 140 smart meters. Parameters L and

M are, therefore, both set to 400.5 m and 400 m respectively, while n_s is set to 140. The SWMN is set up in TOSSIM using Zuniga's Link Layer Model [16], which is used to generate the Python simulation scripts. Figure 3 illustrates the SWMN; the blue circles denote 140 SMs.

Most PSO parameters, in Eqs. 1, 2 and Algorithm 1, are adopted from [17] and set as follows: $N = 20$, $c_1 = 2.0$, $c_2 = 2.0$, $\omega = 0.9$, and $G = 50$. The particles are initilized between 0 and 1, as described in Sect. 3. These particle parameters are converted to MN coordinates by multiplying them by 400 m. The parameters used in the TOSSIM Simulation are shown in Table 1.

Table 1. Tossim Simulation Parameters [10]

Parameter	Value
Transmission range	70 m
Number of nodes	140
Number of invocations per particle evaluation	3
Simulation time	10 s per simulation

The Binary Search PSO code is written in C language. A personal computer with an Intel Core i7-2670QMCPU@2.20 GHz*8 processor with 4 GB RAM, running on Ubuntu 18.01, is used. Ten BS-PSO optimization routines are conducted in the current implementation due to computing constraints. Each routine incorporated Algorithms 1, 2, and 3 as described in the previous sections.

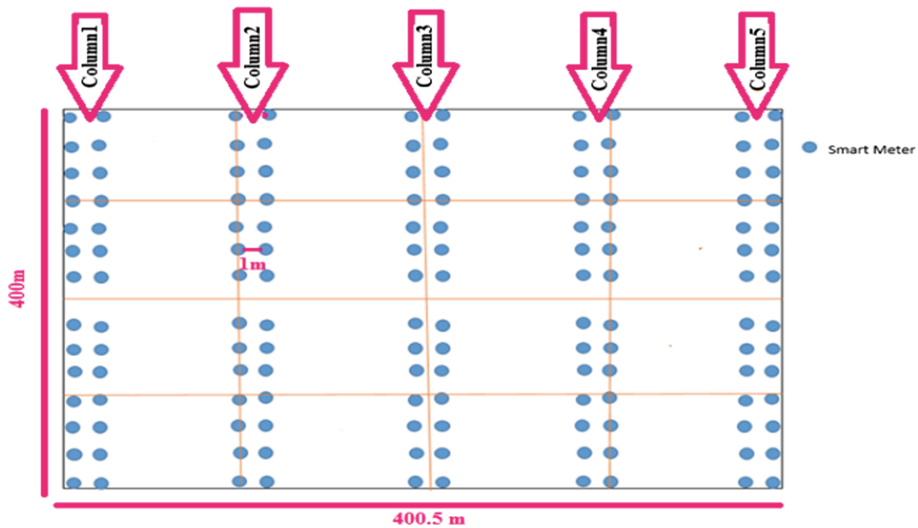

Fig. 3. Smart meters in the Tsumeb SWMN [6]

5.2 Results and Discussion

Ten optimization routines were implemented. For each of the routines, parameter k in Algorithms 2 and 3 is set to 10. Therefore, each optimization routine generates $\mathbf{PDR} = (PDR_0, PDR_1(m) \ldots PDR_9)$ for the realized number of MNs. Table 2 shows the statistical results for the 10 optimization routines.

Table 2. Optimization results from 10 BS-PSO Optimization Routines

Routine no.	Mean PDR (%)	Min. PDR (%)	Max. PDR (%)	No. MNs
1	94.24	96.87	95.95	17
2	94.83	96.19	95.55	15
3	94.19	96.62	95.65	15
4	95.27	97.23	96.06	15
5	94.61	96.26	95.39	12
6	95.55	97.82	96.60	15
7	95.49	97.50	96.51	15
8	92.82	97.89	95.91	15
9	95.60	96.60	96.15	15
10	93.73	97.69	96.03	15

From Table 2, the average PDR is around 96%. The most consistent number of MNs over the 10 routines is 15. Only the first and the fifth routines achieve different results, i.e. 17 and 12 respectively. This is due to the stochastic nature of the Binary Search and the PSO algorithms. From these results it can, therefore, be concluded that the optimal number of MNs for this network is 15, when the BS-PSO algorithm is employed.

To illustrate the spread of the locations of the enumerated MNs in the SWMN, plots of MNs have made on the SWMN based on the realized coordinates from the optimisation process. For brevity of space, only three plots from the best runs for routines 1, 5 and 8 are presented. Figure 4 shows the locations of the 17 MNs realized in the best run in first routine, while Figs. 5 and 6 show the locations of the 12 MNs and 15 MNs realized in the best runs in the fifth and the eighth routines respectively. In some places, the MNs are appear to be cluttered in one place. The crowding of SWMNs in one place, in as much as it is coming from the optimization process, is not a good development. This is because the MNs are actually overlapping, which lead to more interference when packets are being relayed to the MNs. Therefore, there is a need to incorporate an algorithm that will reduce crowding of the MNs in the optimization process in order to maintain a good spread of the MNs in the SWMN topology.

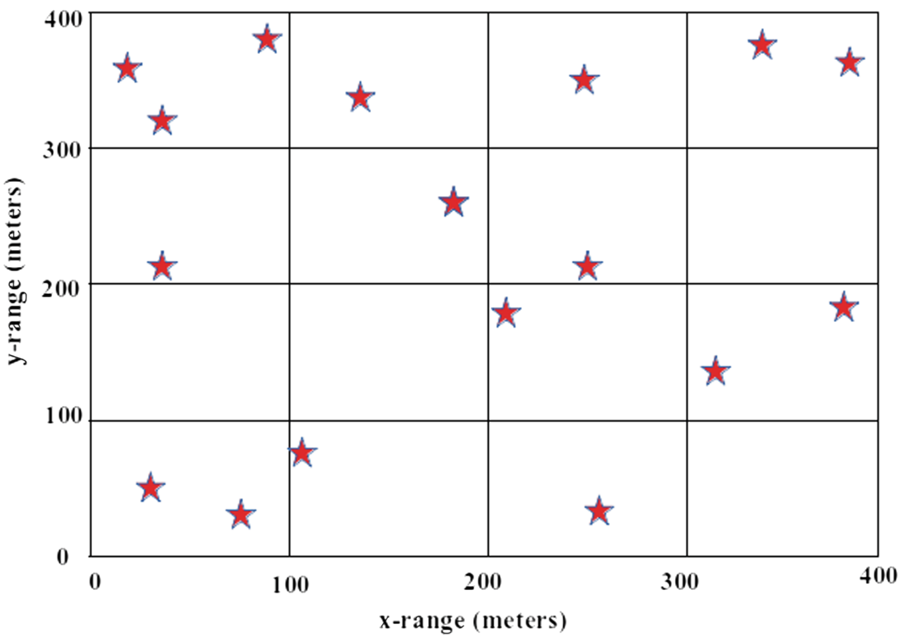

Fig. 4. MN Placement for Run 1, with 17 MNs realized from the optimization process

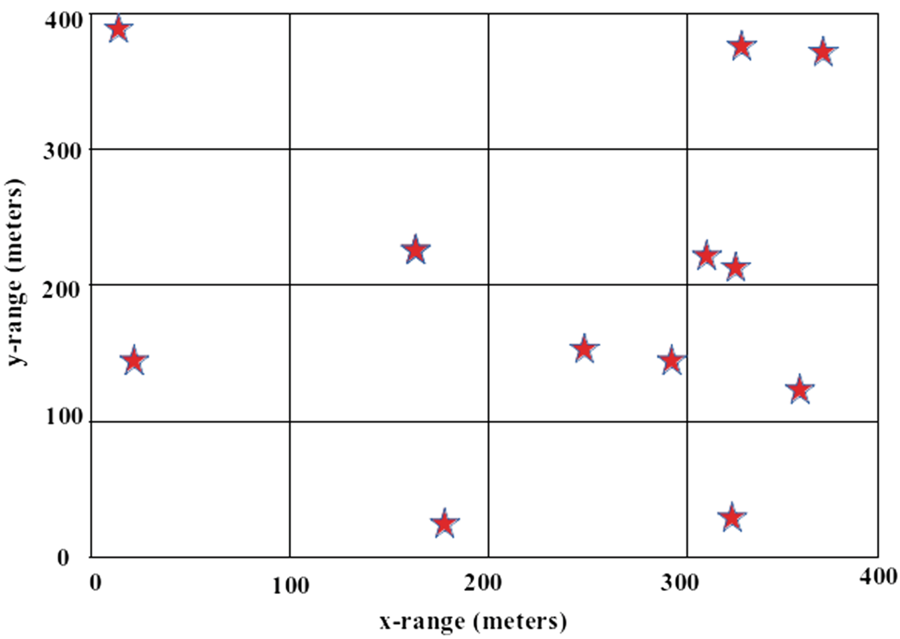

Fig. 5. MN Placement for Run 5, with 12 MNs realized from the optimization process

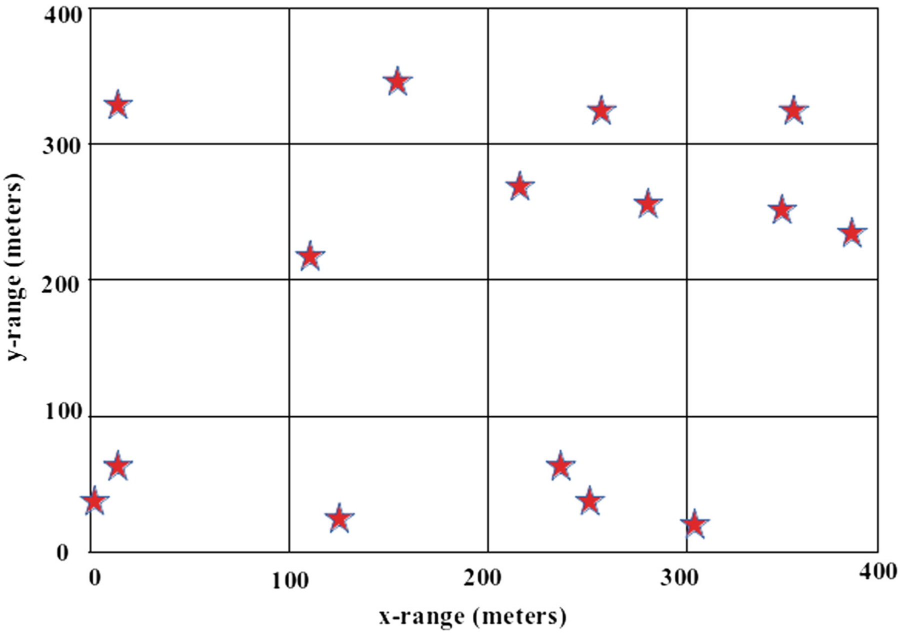

Fig. 6. MN Placement for Run 8, with 15 MNs realized from the optimization process

6 Conclusion

A Binary Search based Particle Swarm Optimization (BS-PSO) algorithm is proposed for the enumeration and placement of Master Nodes (MNs) in a Smart Water Metering Network (SWMN). This proposal was motivated by the desire to simultaneously optimize the number of MNs and their locations in a single run. The system architecture has three levels. Level 1 consists of the Binary Search (BS) algorithm while the PSO algorithm and the SWMN simulation are at Levels 2 and 3 respectively. The Binary Search (BS) Mechanism fixes the upper bound of the search range and then starts searching that range for the optimal number of MNs. The BS algorithm iteratively invokes the PSO algorithm (Level 2) for every potential number of MNs. The PSO algorithm generates particles based on the number of MNs that it receives from the BS algorithm. These particles are used to determine MN coordinates, which it passes to the SWMN simulation (Level 3) in order to determine the Packet Delivery Ratio, which is designated as the fitness function value in this proposal. The Binary Search process uses t-test compare two potential numbers of MNs. Results for 10 BS-PSO optimization routines show that the median optimal number of MNs is 15 and that the mean PDR of 96% can be realized.

The advantage of the proposed BS-PSO algorithm is that the determination of the optimal number of MNs is done automatically in a single run unlike in [10]. The computational requirements are massive to achieve a single optimization.

This is the reason why the number of optimization runs was limited to 10. As part of future work, high performance computing techniques will be employed in order to increase the number of optimization routines and runs to enhance the generalization of the results. Furthermore, future work will also involve the extension of this concept to other optimization algorithms such as Differential Evolution. Dynamic algorithms to cater for the even spreading of MNs in the SWMN will also be developed and analyzed.

References

1. Cahn, A.: An overview of smart water networks. J. Am. Water Works Assoc. **106**(7), 68–74 (2014)
2. Beach, T., Howell, S., Terlet, J., Zhao, W., Rezgui, Y.: Achieving smart water network management through semantically driven cognitive systems. In: Camarinha-Matos, L.M., Afsarmanesh, H., Rezgui, Y. (eds.) PRO-VE 2018. IAICT, vol. 534, pp. 478–485. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99127-6_41
3. de Azevedo, M.T., Martins, A.B., Kofuji, S.T.: Digital transformation in the utilities industry: industry 4.0 and the smart network water. In: Technological Developments in Industry 4.0 for Business Applications, pp. 304–330. IGI Global (2019)
4. Malcolm, F., Gary, W., Zainuddin, G.: The Manager's Non-revenue water Handbook - A Guide to Understanding Water Losses. USAID, USA (2008)
5. Sensus Research: WATER 20/20: Bringing smart water network into focus. Sensus, North American Headquarters (2012)
6. Shitumbapo, L.N., Nyirenda, C.N.: Simulation of a smart water metering network in Tsumeb East, Namibia. In: International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), pp. 44–49 (2015)
7. Nyirenda, C.N., Nyandowe, I., Shitumbapo, L.: A comparison of the collection tree protocol (CTP) and AODV routing protocols for a Smart Water Metering Network in Tsumeb, Namibia. In: IST-Africa Week Conference, pp. 1–8 (2016)
8. McNabb, J.: Vulnerabilities of wireless water meter networks. J. New Engl. Water Works Assoc. **126**(1), 31–37 (2012)
9. Albentia, S.: Proposal for Smart Metering Networks Solution, ALB-W012-000en, UK (2012)
10. Nyirenda, C.N., Makwara, P., Shitumbapo, L.: Particle swarm optimization based placement of data acquisition points in a smart water metering network. In: Bi, Y., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2016. LNNS, vol. 16, pp. 905–916. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-56991-8_66
11. Mudumbe, J.M., Adnan, M., Abu-Mahfouz, M.: Smart Water Meter System for user-centric consumption measurement. In: 13th International Conference on industrial Informatics (INDIN 2015) (2015)
12. Kennedy, J.: Particle swarm optimization. In: Sammut, C., Webb, G.I. (eds.) Encyclopedia of Machine Learning, pp. 760–766. Springer, Boston (2011). https://doi.org/10.1007/978-0-387-30164-8_630
13. Spinsante, S., Pizzichini, M., Mencarelli, M., Squartini, S., Gambi, E.: Evaluation of the wireless M-Bus standard for future smart water grids. In: 9th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 382–1387 (2013)

14. Levis, P., Lee, N., Welsh, M., Culler, D.: TOSSIM: accurate and scalable simulation of entire TinyOS applications. In: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, pp. 126–137 (2003)
15. Fonseca, R., Gnawali, O., Jamieson, K., Kim, S., Levis, P., Woo, A.: The collection tree protocol (CTP). TinyOS TEP **123**(2) (2006)
16. Zuniga, M.: Building a network topology for Tossim. USC Technical Report (2011)
17. Takahama, T.: PSO code. http://www.ints.info.hiroshima-cu.ac.jp/~takahama/download/PSO.html. Accessed 19 July 2019