A hybrid optimization algorithm based on K-means++ and Multi-objective Chaotic Ant Swarm Optimization for WSN in pipeline monitoring

Yandja Lalle
Laboratory of Technology and Smart Systems (LT2S)
National School of Engineering of Sfax (ENIS)
Sfax, Tunisia
emmanuellalle@gmail.com

Maroua Abdelhafidh
Laboratory of Technology and Smart Systems (LT2S)
National School of Engineering of Sfax (ENIS)
Sfax, Tunisia
marwaabdelhafidh@gmail.com

Lamia Chaari Fourati
Laboratory of Technology and Smart Systems (LT2S)
Higher Institute of Computer Science and Multimedia of Sfax
Sfax, Tunisia
lamia.chaari@enis.rnu.tn

Jihene Rezgui
Laboratoire Recherche Informatique (LRIMa)
Maisonneuve
Montreal, Canada
jrezgui@cmaisonneuve.qc.ca

***Abstract*—Pipeline Systems are extensively used to transport and distribute natural gas, water, oil, sewage etc. Due to the aging of these systems, leaks and pipe bursts occur frequently. Therefore, the necessity of continuous monitoring of such systems is required in order to provide early detection of a sudden problem such as leaks, before they attain the magnitude of a major disaster. Wireless Sensor Networks (WSNs) which consist of low-power consumption, low-cost and multi-functional sensor nodes for environmental conditions monitoring, present as a suitable technology to achieve this goal. As nodes in WSNs are powered by a battery, efficient energy consumption is an important factor to enable the network to operate as long as possible. At the same time, network throughput is another important metric for maintaining the Quality of Service of the network (QoS) which need suitable optimization. In this paper, considering these two objectives functions, we propose a novel optimization model based on Multi-objective Chaotic Ant Swarm Optimization (MCASO) approach aiming to optimize WSN energy efficiency and to enhance the network throughput. A K-Means++ algorithm is used to perform the clustering process while MCASO is applied during the optimization phase. The obtained simulation results confirm the enhancement of the network lifetime and the maintaining of the QoS.**

***Index Terms*—Wireless Sensor Networks (WSNs), Quality of Service (QoS), leaks, Optimization, Pareto-Front (PO), Pareto-Set (PS), Pipeline.**

I. Introduction

During these last years, monitor a pipeline system like natural gas pipeline, water or other liquid pipeline has become one important task in the world. However, with the extensive demand for pipeline system, the pipeline leaks problems have become more and more substantial. Standard techniques of pipeline monitoring require rigorous human implication because they rely on periodical examination and does not provide real-time monitoring of the pipeline. As a result, with these conventional techniques, an anomaly (for instance leak) may not be detected in time and this may lead to economic losses and environmental pollution. Therefore, monitor Pipeline System (PS) [1] in real-time should be a suitable solution to detect a leak or any other anomaly related to PS in order to preserve economic resources efficiency. To monitor PS in real-time, Wireless Sensor Networks (WSNs) can be a suitable technology. Indeed, a WSN is a network formed by low-power consumption, multi-functional, and low-cost sensor nodes which collect data from monitoring environment and send it to a base station (BS) using communication protocol like ZigBee. As in PS nodes are something deployed in hard to access environment, their batteries replenishment or charging is very difficult. Thus, once the node drains its battery, it becomes non-functional. Therefore, in order to enable the network to operate as long as possible, the available energy at each node should be optimized by employing appropriate approaches. At the same time, network throughput is another important metric for maintaining the Quality of Service of the network (QoS) which need suitable optimization [2]. In conventional optimization, the most remarkable metric among these two metrics(energy consumption and network throughput) should be considered as the objective function of the optimization problem and the second metric is treated as a constraint. This manner to do is unfair and non-realistic because it prioritize one metric at the detriment to other. A realistic way is to simultaneously consider these two metrics under a set of constraints, this is the goal of Multi-objective Optimization Algorithms (MOAs) [3]. In MOAs it is impossible for these two metrics to achieve at the same their respective optima, but there exist a set of non-dominated solutions or Pareto-optimal (PO) called Pareto-Set (PS) [4]. In this paper, we propose a Bio-inspired Multi-objective Optimization Approach named Multi-objective Chaotic Ant Swarm Optimization (MCASO) by extending a single CASO to find the set of PO, considering energy consumption and network throughput as our objective functions.

Our contributions, in this paper, can be summarized as follows: (1) We propose a multi-objective optimization model that minimize the energy consumption in WSN; the proposed model includes 2 objective functions and therefore provides a better energy usage and maximize the overall throughput; (2) we integrate a K-Means++ algorithm that performs the process clustering ; (3) We also extend the single CASO into Multi-objective CASO. The remain of this paper is organized as follows: In Section II, we present the layout of pipeline monitoring with a WSN while Section III formulates our problem. Section IV presents the basic concepts of Multi-objective Optimization while Section V details the Chaotic Ant Swarm Optimization (CASO). In Section VI the single CASO is extended into Multi-objective CASO (MCASO). Section VII presents the simulation results while VIII concluded our paper.

II. Related Works

Underwater pipelines are very hard to access, and are sometimes subject to harsh environments like salt water. Therefore, as sensor nodes installed along these pipelines are powered by batteries, the regular replacement of these batteries is very difficult. In order to prolong their lifetime, several works are carried out focusing on the optimization of energy consumption. Especially, Abdelhafidh et al. [5] proposed an hybrid clustering algorithm based on K-means and Ant Colony Optimization (ACO); called K-ACO to improve the WSN Lifetime in PS. Moreover, in [6], authors combined data aggregation and bio-inspired clustering algorithm in order to enhance the WSN Lifetime. vesting devices (EHD). Specifically, Qureshi et al. [7] presented a method for near-optimal piezoelectric bimorph energy harvester module aiming to allow a self-powered wireless sensor node for in-pipe monitoring using kinetic energy of water flow. In [8], authors developed the energy scavenging based architecture enabling nodes to recharge their batteries. They also proposed mathematical model to manage efficiently energy at each node.

III. Layout of pipeline monitoring with a WSN

A. WSN topology

A WSN topology can be defined as the positioning of a network, including its nodes and connecting lines. There are several topologies, including Linear topology, point-to-point topology, bus topology, ring topology, star topology tree topology and mesh topology. In this paper, we use Linear topology because of its potential advantages in pipeline architecture such as: fast/cost efficient deployment, reduced requirements for maintenance, increased reliability, and the ability to efficiently adapt multi-hop routing protocols.

B. Pipeline monitoring Architecture

A pipeline monitoring system with a WSN is formed by a great number of nodes, the CHs and a Base Station (BS). In this paper, the clustering process is performed using algorithm detailed in **Algorithm 2**. sensors nodes deployed along the pipeline monitor chemical parameters such as pressure, flow, temperature, pH, conductivity, turbidity. Then the monitoring data information are transmitted to the CHs and CHs at their turn sent it to a BS following one-hop or multi-hop communications depending of the position of the node. The monitoring networks take an in-tube communication mode for information exchange between sensor nodes. The CHs are responsible for managing sensor nodes in its cluster, combining pre-processing information, apply data aggregation techniques and sent it to the BS as shows in Fig. 1. The BS collect all information from all CHs and send it to the remote control or end-user.

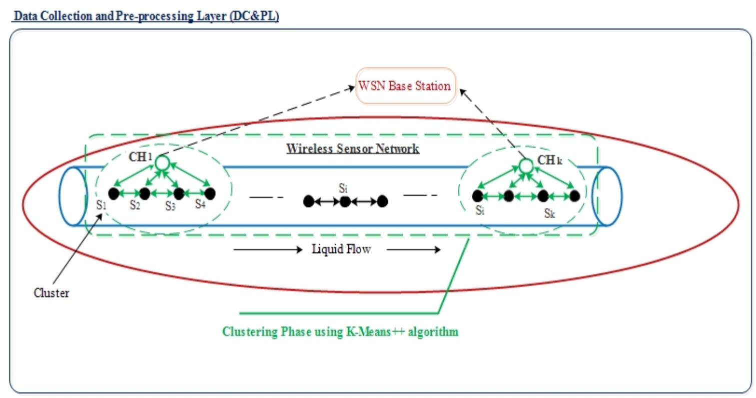

Fig. 1. Pipeline monitoring Architecture

C. Clustering process

Typically, in WSNs, the random clustering mechanism is often used for creating clusters in the network. Unfortunately, this technique suffers from the optimal CHs selection problem. This drawback motivated us to find another clustering method called K-Means. In fact, the K-Means algorithm is an unsupervised learning algorithm that commonly used in network planning and also can be used to solve classification problems. This method groups in our case nodes using Euclidean distance. K-Means divides the data set (nodes) into k clusters using cluster mean value, minimizing thus the inter-cluster similarity and maximizing the intra-cluster similarity. The steps of this algorithm are represented in **Algorithm 1**. The main inconvenient of K-Means is its sensibility to initial CHs selection. This drawback motivated us to search a new variant of K-Means named K-Means++ which has been proposed in 2007 by David Arthur and Sergei Vassilvitskii. The K-means++ algorithm tries to solve the above problem, by spreading the initial centers evenly. **Algorithm 2** details this algorithm.

IV. Problem formulation

Let's consider a Linear WSN composed by a set of sensor nodes installed along a pipeline of L length as shown in

Algorithm 1 K-Means

Input: A Set $\mathbf{S} = \{s_1, s_2, s_3, ...s_n\}$ of sensor and a number k $(k > 1)$ of clusters to form
Output: A Set of k clusters generated
Step 1: Take randomly **k** nodes belong to **S** as initial centroids
Step 2: Compute the Euclidean distance from each node to all centroids and affect it to the closest centroid. By doing so, **k** initial clusters are created
Step 3: Recompute the positions of centroids in each cluster by taking the mean of all data points (nodes) assigned to that centroid's cluster.

$$new_{center} = \frac{1}{c_i}.\sum_{j}^{c_i} s_i \tag{1}$$

where c_i denoted the number of nodes in i^{th} cluster and s_i represents sensor *i*
Step 4: If there is change in position of any centroid then go to Step 2, else the clusters are finalized and the clustering process ends.

Algorithm 2 K-Means++

Input: A Set $\mathbf{S} = \{s_1, s_2, s_3, ...s_n\}$ of sensor and a number k $(k > 1)$ of clusters to form
Output: A Set of k clusters generated
Step 1: Take one center c_1, chosen uniformly at random from **S**
Step 2: Take a second center c_i, corresponding to s$\in$ **S** with probability $\frac{D(s)^2}{\sum D(s)^2}$
Step 3: Repeat Step 2 until the k desired centers are taken.
Step 4: Applied steps (2), (3) and (4) of standard K-Means to finish clustering process

Fig. 1. The nodes transmit packet with μ_{tput} rate using E_i energy. This paper purpose is to minimize energy consumption and maximize network throughput. The models of these two objectives are represented in the following subsections.

A. Energy Consumption

The energy consumed to transmit packet between two nodes is detailed in [9] and we abstract it by:

$$E_i = 2E_{star} + \frac{L}{R}(P_{tx} + P_{rx} + 2P_{cir} + P_{amp}) \tag{2}$$

where:

E_{star} corresponds to the energy for startup the radio;

P_{tx} and P_{rx} are respectively the power consumption of the radio in transmission mode and receive mode;

P_{cir} is the power consumption of the electronic circuitry; L denotes the payload size in bits;

R represents the transmission data rate;

$P_{amp} = \frac{(cd)^n}{BER}$ denotes the energy consumption of the power amplifier, which is calculated by transmission range and BER (bit-error-rate). c is a constant depending on channel attenuation and non-linear effect of the power amplifier; d is the transmission range; and n is the poss loss exponent;

B. Packet throughput

Network throughput is the rate of successful packet that is delivered over a communication channel. So, if this parameter increases then the network efficiency will be increased [10]. This factor is affected by two important factors like packet error rate and packet length and is represented by:

$$\mu_{tput} = \frac{L.(1 - PER)}{T_{flow}} \tag{3}$$

where:

$PER = 1 - (1 - BER)^L$: packet Error Rate.

T_{flow}: is the end-to-end latency .

L: denotes the Packet length.

V. Basic aspects on Multi-objective optimization (MOO)

We give here the basic concepts of MOO. Generally, Multi-objective Problem (MOP) is formed by a set of objective functions under a set of inequality and equality constraints. In order to conserve generality, multi-objective minimization or maximization problem having *n* variables and *m* $(m\ > 1)$ objectives can be formulated as:

$$\begin{cases} min/max f = min/max[f_1(x), f_2(x), f_3(x), ...f_n(x)] \\ s.t \quad g_i(x) \leq 0, i = 1, 2, ...mie \\ h_j(x) = 0, j = 1, 2, ...mee \end{cases} \tag{4}$$

with $x \in R^n$ being the decision space, and $f_i(x)$ denoted the objectives space. mie denoted the multiple inequality equation, while mee represents the multiple equality equation. The objective functions of (2) are typically in conflict with each other in the real optimization problems. Clearly, the improvement of one of the objectives may result in the degradation of other objectives, thus it is important to achieve the Pareto-optimality, which represents the conditions where no one of the objective functions can be optimized without sacrificing at least one of the other objectives [11]. For any optimization problem, we have the following concepts:

- Non-dominated solutions (ND): A solution **a** is said to dominate a solution **b** if and only if: $f_i(a) \leq f_i(b) \forall i \in \{i = 1, 2, ...m\}$,
  $f_i(a) = f_i(b) \exists i \in \{i = 1, 2, ...m\}$.
- Local optimally: In the Pareto sense, a solution **a** is locally optimal, if there exists a real $\epsilon > 0$ such that there is no other solution **b** dominating the solution **a** with $\mathbf{b} \in R^n \bigcap \mathbf{B}\left(\mathbf{a}, \epsilon\right)$
- Global-optimality: A solution **a** is globally optimal in the Pareto sense, if there does not exist any vector **b** that dominates the vector **a**.
- Pareto-optimality: When a solution is not dominated by any other solution in the search space, this solution is called PO. The set of all PO solutions and their

corresponding images in the objective space is termed Pareto-Front (PF).

VI. Single Chaotic Ant Swarm Optimization (SCASO)

CASO is a global optimization approach based on the chaotic behavior of natural ants and the intelligent organization of the ant colony. Initially, each ant perform chaotic search and the self-organization process of the ant colony is achieved by introducing a organization variable r_1. The impact of this variable on the ant behavior is relatively small at the beginning. Once the organization variable increases gradually, the chaotic behavior of ants decreases gradually. With the increasing of organization variable and exchange of information between neighbor ants, the individual ant forget its position and moves to the best one. The neighbors selection concept is introduced in this algorithm to simulate the behavior of ants species in the nature. The chaotic search has been proposed in [12] and is detailed as follows : The search area of ants represents the problem search space. Solution of the considered problem are searching in the search space R^l. A population containing **K** ants is considered. These ants, localized in a search space **S** try to optimize a function f : $\mathbf{S} \longrightarrow \mathbf{R}$. Each value s in **S** is a possible solution to the considered problem. The position of each ant i is represented by:

$S_i = (z_{i1}, z_{i2}, ...z_{il})$. where i=1, 2, . . ., **K**. During moving to the best site (best positions), each individual ant is impacted by the organization process of the ant swarm. Mathematically, the strategy used by an ant to move toward the best positions is supposed to be a function of four parameters such as the current position of this ant, the best position found by itself and any member of its neighbors and the organization variable. The chaotic system is represented by the following equations:

$$\begin{cases} y_i(n) = y_i(n-1)^{1+r_i} \\ z_{id}(n) = (z_{id}(n-1) + \frac{7.5}{\psi_d}.V_i)exp(1-exp(-ay_i(n)) \\ (3-\psi_d(z_{id}(n-1) + \frac{7.5}{\psi_d}.V_i))) - \frac{7.5}{\psi_d}.V_i + \\ exp(-2ay_i(n)+b)(p_{id}(n-1) - z_{id}(n-1)) \end{cases} \tag{5}$$

Where:
$y_i(n)$ represents the current state of the organization variable,
a corresponds to a sufficiently large positive constant and can be selected as **a** = 200
[12], **b** is a constant and 0$\leq b \leq$ 2/3 [12],
$r_i \in [0, 1/2]$ is a positive constant less than one and designs the organization factor of ant i,
$z_{id}(n)$ the current state of the d^{th} dimension of the individual ant i,
d=1,2,............l,
ψ_d determines the selection of the search range of d^{th} element of variable in search space and
0$\leq V_i \leq$ 1 determines the search region of i^{th} ant.
The value of V_i should be appropriately chosen according to the concrete optimization problems [12]. In this model the initial position of individual ant can be selected as:

$$z_{id}(0) = \frac{7.5}{\psi_d}.(1-V_i)rand() \tag{6}$$

where $\psi_d > 0$. During ants movement, they select their neighbors using the following ways: The first way is the nearest fixed number of neighbors. The nearest m ants are defined as the neighbors of single i_{th} ant. The second way to select neighbors is to consider the case where the number of neighbor increase each iteration. This is due to the impact of self-organization of ants. As the organization variable increases with time, the neighbors of every ant also increase gradually. The number of neighbors of each ant is defined to be increase for each iteration. Due to the self-organizing behavior, every individual ant will follow their neighbors as time evolves and the ant swarm converges. All the pre-defined values given in this paper are drawn from [12].

VII. From SCASO to Multi-objective CASO (MCASO)

let's consider a set of x_i, i = 1, 2,...n as the decision variables of our optimization problem and $f_1(x), f_2(x), f_3(x), ...f_m(x)$ as a set of functions to optimize. As we mentioned in section III, these functions are typically conflict with each other in real-optimization because of the constraints imposed by the problem. For this reason, it is infeasible for multiple functions to achieve their optima at the same time. Therefore, Pareto-dominance is used to find a set of FO. To extend SCASO to MCASO, we naively employed the basic concepts of MOO define in section III. We also used the archive-base **A** to store the updated ND solutions found so far. **A** is set to be empty initially. In order to update the content of **A**, we used the ND sorting algorithm proposed in [13] which classify solutions into Dominated (D) and ND solutions at each iteration of the algorithm. This algorithm determines which solution will be stored in **A**(containing only ND solutions). The "crowding distance" concept detailed in [13] is additionally used in order to enable MCASO to distribute the obtained solutions on the Pareto front evenly. The details of non-dominated sorting algorithm and crowding distance are presented as follows:

nonDominatedSortProcedure(Population **P**)
Begin Procedure
for each p $\in$ **P**
S_p = Ø /*S_p is a set of solutions that the solution p dominates*/
n_p = 0 /* n_p is the number of solutions which dominate the solution p*/
for each q $\in$ **P**
if(p dominates q)
$S_p = S_p \bigcup \{q\}$
else if(q dominates p)
n_p = n_p+1
F_1 = Ø /*First front*/
if(n_p = 0) then
p_{rank} = 1

$F_1 = F_1 \bigcup \{p\}$
i = 0 /*Front counter initialization*/
While ($F_i \neq \emptyset$)
$\mathbf{Q} = \emptyset$
for each $p \in \mathbf{F}_i$
for each $q \in \mathbf{S}_p$
$n_q = n_q - 1$
if(n_q=0) then
q_{rank} = i+1
$\mathbf{Q} = \mathbf{Q} \bigcup \{q\}$
i = i+1
F= Q
end Procedure

crowdingDistanceProcedure(SND) /*SND is a set of non-dominated solutions*/
Begin Procedure
l = $\|SND\|$ /*number of solution in SND*/
for each i in SND, $SND[i]_{dist} = 0$
for each objective m
sort(SND, m) /*sort each non-dominated solution in SND based on objective m*/
$SND[1]_{dist} = SND[l]_{dist} = \infty$ /*boundary points*/
for i = 2 to (l-1)
$SND[i]_{dist} = SND[i]_{dist} + \frac{SND([i+1]).m - SND([i-1]).m}{f_m^{max} - f_m^{min}}$
end Procedure

The details of our final algorithm are in **Algorithm 3**.

VIII. Simulation results

A. Clustering process

The K-Means++ algorithm is employed to perform clustering process. K-Means++ has been evaluated through simulations in python 3.6. 50 sensor nodes have been deployed along the pipeline. The problem that strikes the eye here is the choice of the **k** value. Often, the random manner is used to choose this value. In this paper, we used the "Elbow method" to find optimal **k**, given a set of nodes to cluster. Indeed, The "Elbow method" is a method of interpretation and validation of consistency within cluster analysis designed to help to find the appropriate number of clusters in a dataset [14]. Under python evaluation, "Elbow method" provides with 50 nodes, results represented on Fig. 2.

As we can see in Fig. 2, the optimal number of cluster to form is **k = 4**. With this value of **k**, K-Means++ gives results represented in Fig. 3. We consider that nodes are initially deployed randomly along the pipe with a specific transmission range (T_r) and a specific sensing range (S_r). TABLE I provided the number of nodes in each cluster using K-Means++.

B. MCASO phase

After clustering process, the MCASO takes place with the aim of optimizing our two objectives (energy consumption and network throughput). The simulation parameters are provided by TABLE II. The maximum packet payload size that supported by ZigBee is 114 bytes [10]. We compared the results

Algorithm 3 K-MCASO

Step 1: Generate randomly the initial population **P**. Generate randomly the positions of each ant, set $y_i(0)$ = 0.999 and **A**=∅
Step 2: Do **Algorithm 2** to perform clustering process
Step 3: Evaluate each individual in the population
Step 4: Classification of all individual in **P** into D and ND solutions by using:
nonDominatedSortProcedure(**P**)
Step 5: Compute crowding distance of all non-dominated solution by using:
crowdingDistanceProcedure(SND) /*SND is the output of non-Dominated sorting procedure*/
Step 6: Update each ant position using:

$$\begin{cases} y_i(n) = y_i(n-1)^{1+r_i} \\ z_{id}(n) = (z_{id}(n-1) + \frac{7.5}{\psi_d}.V_i)exp(1 - exp(-ay_i(n)) \\ (3 - \psi_d(z_{id}(n-1) + \frac{7.5}{\psi_d}.V_i))) - \frac{7.5}{\psi_d}.V_i + \\ exp(-2ay_i(n) + b)(p_{id}(n-1) - z_{id}(n-1)) \end{cases} \quad (7)$$

Step 7: Update the archive **A**
A' = **A** $\bigcup \{SND\}$
Sort **A'** into D' and ND' solutions by using:
nonDominatedSortProcedure(**A'**)
A = ND'
Step 8: Generate new population by (4)
Step 9: If k_{max} iteration reach
Output the population
else
goto (3)

Cluster	C1	C2	C3	C4
Population	14	13	10	13

TABLE I
Total number of nodes in each cluster

TABLE II
Parameters of simulation

	Parameters	Values
1	E_{star}	1.0 uJ
2	P_{tx}	19.1 mW
3	P_{rx}	14.6 mW
4	T_{flow}	30us
5	L	114bits
6	Number of generation	250
7	Number of nodes	50
8	BER	5.10^-4
9	n	3.5
10	d	20m
11	P_{cir}	12mW
12	C	19.2
13	R	20bps

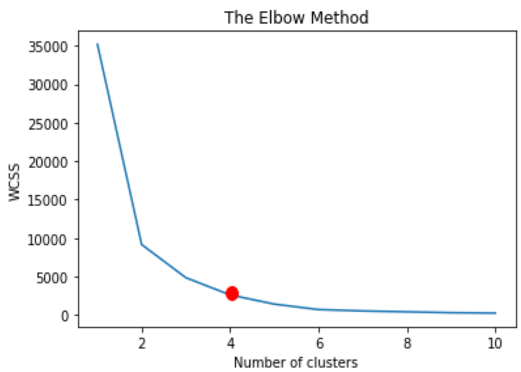

Fig. 2. Elbow method with 50 sensor nodes

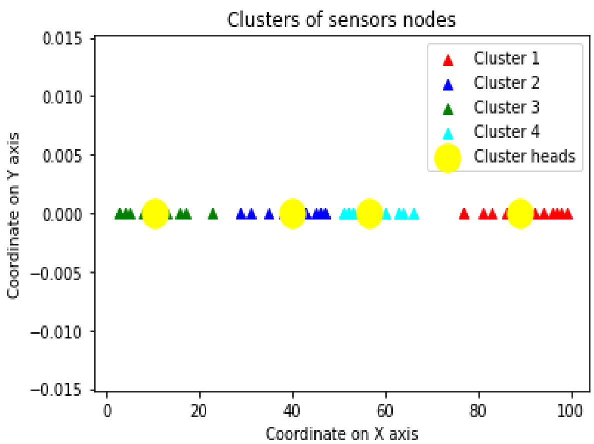

Fig. 3. Cluster formation with k=4 using K-Means++

of K-MCASO with results obtained by Non-Sorting Genetic Algorithm II (NSGA-II) and the results have shown that our algorithm provides optimal solutions in terms of PO finding as presented in Fig.4

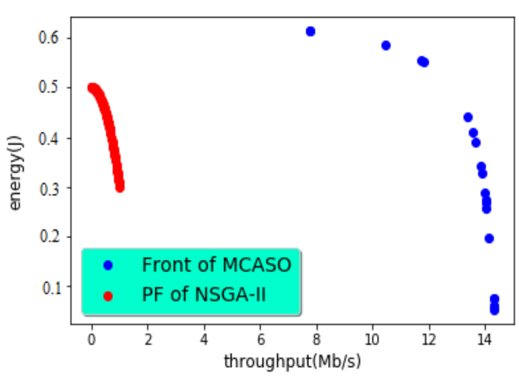

Fig. 4. Pareto-Front provide by K-MCASO and NSGA-II

IX. Conclusion

A hybrid K-Means++ MCASO was proposed in this paper for two main purposes: optimal clustering, and extending Network Lifetime (NL) and maintaining the QoS. We achieved these purposes by considering two phases. Firstly, a K-Means++ algorithm is used to perform optimally clustering process. Secondly, we extended SCASO into MCASO by introducing in SCASO the following concepts: Parto-dominance, non-dominated sorting, crowding distance and archive-base, detailed in section IV. Our optimization problem is formulated as a mathematical model and MCASO is used to find a set of Pareto-Front called Pareto-Set which constitute the solutions of our problem. Numeric simulations proven that our proposed method outperformed NSGA-II in terms of finding Pareto-Optimal.

References

[1] Maroua Abdelhafidh, Mohamed Fourati, Lamia Chaari Fourati, and Amor Laabidi. An investigation on wireless sensor networks pipeline monitoring system. *International Journal of Wireless and Mobile Computing*, 14(1):25–46, 2018.

[2] Mohammad Hammoudeh and Robert Newman. Adaptive routing in wireless sensor networks: Qos optimisation for enhanced application performance. *Information Fusion*, 22:3–15, 2015.

[3] BY Qu, YS Zhu, YC Jiao, MY Wu, Ponnuthurai N Suganthan, and JJ Liang. A survey on multi-objective evolutionary algorithms for the solution of the environmental/economic dispatch problems. *Swarm and Evolutionary Computation*, 38:1–11, 2018.

[4] Kalyanmoy Deb and Kalyanmoy Deb. *Multi-objective Optimization*, pages 403–449. Springer US, Boston, MA, 2014.

[5] Maroua Abdelhafidh, Mohamed Fourati, Lamia Chaari Fourati, Adel Ben Mnaouer, and Mokhtar Zid. Linear wsn lifetime maximization for pipeline monitoring using hybrid k-means aco clustering algorithm. In *2018 Wireless Days (WD)*, pages 178–180. IEEE, 2018.

[6] Maroua Abdelhafidh, Mohamed Fourati, Lamia Chaari Fourati, Adel Ben Mnaouer, and Mokhtar Zid. Lifetime maximization for pipeline monitoring based on data aggregation and bio-inspired clustering algorithm. In *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 666–671. IEEE, 2018.

[7] Fassahat Ullah Qureshi, Ali Muhtaroğlu, and Kağan Tuncay. Near-optimal design of scalable energy harvester for underwater pipeline monitoring applications with consideration of impact to pipeline performance. *IEEE Sensors Journal*, 17(7):1981–1991, 2017.

[8] V Balasubramanian, Nikolaos Kouvelas, K Chandra, RV Prasad, Artemios G Voyiatzis, and W Liu. A unified architecture for integrating energy harvesting iot devices with the mobile edge cloud. In *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pages 13–18. IEEE, 2018.

[9] Huimin She, Zhonghai Lu, Axel Jantsch, Dian Zhou, and Li-Rong Zheng. Analytical evaluation of retransmission schemes in wireless sensor networks. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th*, pages 1–5. IEEE, 2009.

[10] Mohammad Hamdan, Muneer Bani Yassein, and Hisham A Shehadeh. Multi-objective optimization modeling of interference in home health care sensors. In *Innovations in Information Technology (IIT), 2015 11th International Conference on*, pages 219–224. IEEE, 2015.

[11] Carlos A Coello Coello, Gregorio Toscano Pulido, and M Salazar Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 8(3):256–279, 2004.

[12] Jiejin Cai, Xiaoqian Ma, Lixiang Li, Yixian Yang, Haipeng Peng, and Xiangdong Wang. Chaotic ant swarm optimization to economic dispatch. *Electric Power Systems Research*, 77(10):1373–1380, 2007.

[13] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In *International Conference on Parallel Problem Solving From Nature*, pages 849–858. Springer, 2000.

[14] Yoshimi Fukuoka, Mo Zhou, Eric Vittinghoff, William Haskell, Ken Goldberg, and Anil Aswani. Objectively measured baseline physical activity patterns in women in the mped trial: Cluster analysis. *JMIR public health and surveillance*, 4(1), 2018.