A quantum inspired PSO algorithm for energy efficient clustering in wireless sensor networks

Pradeep Kanchan\textsuperscript{1*} and Shetty D Pushparaj\textsuperscript{1}

Abstract: Clustering is done in wireless sensor networks (WSN) to conserve the energy of sensor nodes in the network. Decreasing the energy consumption of nodes prolongs the lifetime of the WSN. A quantum bit can exist in “0” state, “1” state or a linear superposition of “0” and “1” states, unlike the binary bit which can exist in only “0” state or “1” state. In this paper, we propose a Quantum inspired PSO (particle swarm optimization) called Quantum-inspired PSO for Energy Efficient Clustering (QPSOEEC). The algorithm is tested by giving different values to the number of sensor nodes and cluster heads, varying the base station position, etc. Then, our results are compared to existing algorithms that demonstrate the superiority of our algorithm.

Subjects: Algorithms & Complexity; Computer Engineering; Computer Science; General

Keywords: clustering; PSO; quantum bits; wireless sensor networks

1. Introduction

A wireless sensor network (WSN) is a network of nodes deployed within an area in random positions. These nodes interact with each other to collect data, process it and then communicate with a base station (BS). Most often, these sensors are battery operated. Once the sensors are deployed in a specific field, say, in the deep sea or in the battlefield, it is difficult to either replace the battery or supply additional energy. Therefore, the main challenge is to conserve the energy in...
WSN’s. Various mechanisms have been used by researchers for conserving energy and one of the most effective methods is Clustering. In Clustering, the network is divided into various clusters. Each cluster has a cluster head (CH) which collects the data from all member nodes (sensors) of its cluster. Then, the CH aggregates the data and sends it to the BS.

The BS processes the data as per the requirement and performs actions like notifying an event.

The energy consumption of the network is affected by clustering. CH selection is a NP-Hard optimization problem (Latiff, Tsimenidis, & Sharif, 2007). Researchers prefer PSO to other algorithms because it is easy to implement, converges quickly and can escape from the local optima.

In this paper, we present an algorithm where position updates are performed using Quantum Inspired methods and CH selection is done using PSOECHS (Rao, Jana, & Banka, 2017). The algorithm is labeled as Quantum-inspired PSO for Energy Efficient Clustering (QPSOECC). It is compared with existing algorithms like LEACH (Heinzelman, Chandrakasan, & Balakrishnan, 2002) and PSOECHS (Rao et al., 2017).

The following are our contributions:

- Quantum PSO-based position updates followed by PSO-based CH selection
- Deriving a weight function for cluster formation
- Demonstrating the efficiency of our proposed algorithm over existing algorithms through simulation

The rest of the paper is organized as follows. Section 2 contains a review of literature related to the field of interest. Preliminaries like introduction to PSO and QPSO along with the network model and energy model are explained in Section 3. Section 4 consists of derivation of the fitness function, position update using QPSO, selection of CH, and formation of clusters. Section 5 contains discussion of the simulation results. Finally, the conclusion is explained in Section 6.

2. Review of related works

2.1. Nature inspired approaches for clustering
Tillett, Rao, and Sahin (2002) proposed a novel method called PSO (particle swarm optimization), which uses the concept of a swarm for Clustering problem. The main disadvantage of this method is that there may be energy imbalance in the network because it assigns nodes that are not CH nodes to CH nodes based on the distance only. Guru, Halgamuge, and Fernando (2005) proposed PSO-based cluster formation. There may be some residual energy in the sensor nodes that are ignored in this paper. Latiff et al. (2007) proposed the PSO-C algorithm that is used to perform energy aware CH selection. Its disadvantage is that during cluster formation, sensor nodes that are not CH nodes are assigned to the nearest CH and this may cause the network to become inefficient in terms of energy. This also has one more disadvantage that it decreases lifetime of the network. Rao et al. (2017) reported that the PSO-ECHS algorithm performs CH selection while maintaining the energy efficiency. Here, the PSO algorithm performs CH selection. In cluster formation, a weight function is calculated and used by non-CH nodes to join their CHs.

2.2. Heuristic approaches for clustering
LEACH (Heinzelman et al., 2002) is a very popular algorithm used for clustering. Here, clusters are formed by the nodes with one node performing as the CH. Those nodes that are not CH (non-CH nodes) perform the job of transmitting data to CH and the CH receives the data. The CH has to collect the data received and transmit it to a remote BS. The disadvantage of this
algorithm is the possibility of a CH with low energy being selected. This may cause it to stop functioning, affecting the network. In PEGASIS (Lindsey & Raghavendra, 2002), each node communicates only with a node that is its close neighbor and takes turns transmitting to BS. As far as energy efficiency is concerned, PEGASIS is better than LEACH but it may become unstable when there are networks of large size. TL-LEACH (Loscri, Morabito, & Marano, 2005) introduces the concept of using a Two level hierarchy. Here, local cluster BSs (primary CHs and secondary CHs) are rotated randomly. The disadvantage is that electing secondary CHs causes extra overhead. In addition, there may be an energy imbalance in the network because non-CH nodes are assigned to CHs based on the distance only. M-LEACH (Zhou, Jiang, & Xiaoyan, 2006) and LEACH are almost similar except that M-LEACH forwards data to the next hop CH node; it does not send directly to the BS. The disadvantage is that it does not take into account the cluster formation phase. In V-LEACH (Yassein, Khamayseh, & Mardini, 2009), some CHs are selected as vice CHs. These vice CHs become CHs when the main CHs die. Disadvantage is the additional energy required for selecting Vice CHs. In E-LEACH (Xiangning & Yulin, 2007), nodes having more residual energy are made as CHs for the next round. This can extend the life of the network.

2.3. Quantum computing based algorithms

Sun, Feng, and Xu (2004) proposed a Quantum-based PSO. They consider an individual particle of a PSO system moving in a quantum multidimensional space. In (Yang, Wu, & Min, 2015), the authors propose Improved Quantum-behaved PSO with Elitist Breeding (EB-QPSO). Here, elitist breeding strategy is used to escape from local optima and perform efficient search by the swarm. Sun, Fang, Wu, Palade, and Xu (2012) performed an analysis of QPSO. A particles behavior is influenced by a parameter called contraction expansion (CE) coefficient. The upper bound of CE coefficient, within which the value of the CE coefficient selected can guarantee the convergence of the particles position, is found by the authors. Pant, Thangaraj, and Abraham (2008) developed a variant of QPSO which they call Q-QPSO. Their method uses interpolation based recombination operator. Yin, Li, Zhang, and Huo (2010) proposed a Quantum PSO in which Quantum rotation gates do the updation of Quantum bits and mutation is done by quantum non-gates. Their algorithm performs better than GA and classical PSO.

3. Preliminaries

3.1. PSO introduction

Real life swarms inspire PSO (Kennedy, 2011). An example is a flock of birds searching for food and shelter. Each bird is analogous to a particle. Here, each bird will be having a position and velocity; in the same way, a particle of the swarm will be having a position and velocity. In PSO, N denotes the total number of particles. A Position Vector and a Velocity Vector define each particle P. The position vector \( \mathbf{X} = (x_1, x_2, \ldots, x_D) \) signifies a solution and velocity vector \( \mathbf{V} = (v_1, v_2, \ldots, v_D) \) is responsible for exploration of search space. Here, \( D \) denotes dimensionality of the search space. It has the same value for all particles. A fitness function, which depends on the application for which the algorithm is being used, is used to evaluate each particle. The objective of PSO is to find that position of a particle, which results in the best value of fitness function. In the initialization stage of PSO, a position and velocity are assigned to each particle. During each iteration, the best/partial best of a particle denoted as \( P_{best} \) and global best of the swarm denoted as \( G_{best} \) are calculated. The aim is to reach a global best solution. To attain this, the position of the particle, \( X \) and velocity, \( V \) are updated by the following equations:

\[
V_{i,d}(t+1) = \omega V_{i,d}(t) + C_1 r(X_{P_{best,d}} - X_{i,d}) + C_2 r(X_{G_{best,d}} - X_{i,d}) \quad (1)
\]

\[
X_{i,d}(t+1) = X_{i,d}(t) + V_{i,d}(t+1) \quad (2)
\]

where \( \omega = \) inertia weight \((0<\omega<1)\)
C1, C2 = acceleration coefficients,

r, R = random numbers uniformly distributed in the interval (0,1),

d = dimension component, and

i = particle number.

The updation process is repeated until an acceptable value of G_best is generated. Here, ω, the Inertia weight, is used to control the velocity. C1 is the Cognitive scaling parameter and C2 is the Social scaling parameter. Usually, C2 = C1. P_best and G_best are calculated as follows:

\[ P_{\text{best}} = P_i \text{ if } \text{Fitness}(P_i) < \text{Fitness}(P_{\text{best}}) \]
\[ = P_{\text{best}}, \text{ otherwise} \] (3)

\[ G_{\text{best}} = P_i \text{ if } \text{Fitness}(P_i) < \text{Fitness}(G_{\text{best}}) \]
\[ = G_{\text{best}}, \text{ otherwise} \] (4)

Each particle will try to attain its “personal best” while at the same time trying to move the group towards the Global Best.

3.2. QPSO introduction

Clerc and Kennedy (2002) stated that convergence of the PSO occurs when each particle converges to its local attractor \( p_{t,i}^d \), which is defined as follows:

\[ p_{t,i}^d = \psi_d^t + \left( 1 - \psi_d^t \right) G_{\text{best}}^d \]
\[ = C_1 r \left( C_1 r + C_1 R \right) \] (5)

where \( \psi_d^t = \frac{C_1 r \left( C_1 r + C_1 R \right)}{C_0 m_{\text{best}}^t} \ln \left( \frac{1}{u_{t,i}^d} \right) \), if \( \text{randv} \geq 0.5 \)
\[ = p_{t,i}^d - \alpha \left| x_{t,i}^d - m_{\text{best}}^d \right| \ln \left( \frac{1}{u_{t,i}^d} \right), \text{ if } \text{randv} < 0.5 \] (6)

Here, C1, C2, r and R are as in (1). Each particle will converge to its own local attractor. At the same time, various parameters like the current position of the particle, its personal best position, its local attractor and the global best position of the particle will also converge to one point. This leads to convergence of the algorithm.

Quantum PSO (QPSO) (Sun et al., 2004) is developed based on the above analysis.

Each single particle in QPSO is a spin-less particle moving in an N-dimensional Hilbert space with some energy. Its state is characterized by a wave function which depends only on its position. The probability of a particle appearing at a position denoted by \( x_t^i \) in the iteration \( t \) is determined from a probability density function (Liu, Sun, & Xu, 2006).

Using the Monte Carlo method, a particle will fly according to the following formulas:

\[ x_{t+1,i}^d = p_{t,i}^d + \alpha \left| x_{t,i}^d - m_{\text{best}}^d \right| \ln(1/u_{t,i}^d), \text{ if } \text{randv} \geq 0.5 \]
\[ = p_{t,i}^d - \alpha \left| x_{t,i}^d - m_{\text{best}}^d \right| \ln(1/u_{t,i}^d), \text{ if } \text{randv} < 0.5 \] (6)

where

\( \alpha = \text{contraction expansion (CE) coefficient} \)
\( \alpha \text{ and randv are uniformly distributed random numbers in the range } [0,1] \)

m_best = mean best
m_{\text{best}} is calculated as follows:

\[ m_{\text{best}} = \left( \frac{1}{N} \right) \sum_{i=1}^{N} P_{\text{best}_i} \]  

(7)

where \( N \) = swarm size.

The CE coefficient is a parameter that can be adjusted to maintain a balance between local and global search of the algorithm during the search process. It is controlled by using a time-varying decreasing method (Sun et al., 2012) as follows:

\[ \alpha = \alpha_1 + \left( \frac{T}{C_0} \right) \left( \frac{\alpha_0}{C_0} \right) \]  

(8)

where

\( \alpha_0 = \) initial value of \( \alpha \)

\( \alpha_1 = \) final value of \( \alpha \)

\( \alpha = \) number of maximum possible iterations

\( t = \) current iteration number.

The QPSO algorithm is a probabilistic optimization algorithm. A major advantage of QPSO algorithm over PSO is that in QPSO, only the position vector is required for calculations whereas in PSO, both Position and velocity vectors are needed.

3.3. The energy model used

The Energy model used is the same as the one used in LEACH (Heinzelman et al., 2002) and PSO-ECHS (Rao et al., 2017). The energy consumed by a node to transmit a \( l \)-bit data packet is as follows:

\[ E_{\text{Transmit}} = lE_{\text{elec}} + l\epsilon_{fs}d^2, \text{ if } d < d_0 \]

\[ = lE_{\text{elec}} + l\epsilon_{mp}d^4, \text{ if } d \geq d_0 \]  

(9)

where

\( l = \) no. of bits in the data packet,

\( E_{\text{elec}} = \) the energy dissipated/bit for running the transmitter/receiver circuit,

\( \epsilon_{fs} = \) amplification energy (using free space method),

\( \epsilon_{mp} = \) amplification energy (using multipath model),

\( d = \) propagation distance, and

\( d_0 = \) threshold distance.

Similarly, energy consumption by the receiver to receive \( l \)-bit of data is given by

\[ E_{\text{Receive}} = lE_{\text{elec}} \]  

(10)

Several factors such as digital coding, modulation, filtering, and signal spreading influence the \( E_{\text{elec}} \).

Overall energy that is spent by the sensor node in order to transmit data and receive
data is given as follows:

\[ E_{\text{Total}} = E_{\text{Transmit}} + E_{\text{Receive}} \] (11)

3.4. The network model
The sensors are deployed randomly and, once deployed, are assumed to be stationary. A single node can operate as both Sensor as well as CH. Each node performs sensing at regular intervals of time. A node sends data to its CH or BS. Usually the number of sensors will be greater than the number of CHs. The sensors (CH or BS) use different levels of transmission power depending upon the distance to which data will be sent. An assumption is that sensor nodes are homogeneous and have equal capacity for processing and communication.

4. Proposed algorithm
The QPSOEEC is a nature-inspired algorithm, which combines the good features of both PSO and Quantum computing. The main advantage of our proposed algorithm over existing algorithms is that in the position-updating phase, only one parameter, the position vector, is required. The QPSOEEC algorithm has the following three steps:

(1) Position updating
(2) CH selection
(3) Formation of clusters

The position updating is done using QPSO (Yang et al., 2015) and selection of CH is done using PSO-ECHS (Rao et al., 2017). The advantage of using QPSO for position updating is that many parameters are not needed as in Rao et al. (2017); only the position vector is needed. In the CH selection phase, the sensor nodes send information, which indicates their location and residual energy to the BS. Only those nodes, which meet a threshold energy, become eligible to be a CH. The CH selection algorithm is run at the BS. Then, cluster formation phase starts. In cluster formation, a weight function is derived based on factors like distance, node degrees of CHs and energy as in Rao et al. (2017). The sensor node joins the CH with the highest weight value.

4.1. Deriving the fitness function
The fitness function is derived in the same way as in Rao et al. (2017). \( f_1 \) is dependent on the average distance among clusters and average distance between CH and BS. Our aim is to minimize \( f_1 \):

\[ f_1 = \sum_{j=1}^{m} \left( \frac{1}{l_j} \right) \sum_{i=1}^{l_j} \left( \text{dist}(s_i, CH_j) + \text{dist}(CH_j, BS) \right) \] (12)

where

\[ m = \text{no. of CH's}, \]

\[ l_j = \text{no. of sensor nodes in cluster, } j \]

\[ j = \text{distance between sensor } s_i \text{ and its selected CH, and } CH_j \]

\[ \text{dist}(CH_j, BS) = \text{distance between CH } CH_j \text{ and the BS.} \]

\( f_2 \) is defined as the reciprocal of total energy of all selected CHs. Our aim is to minimize \( f_2 \).
\[ f_2 = \frac{1}{m} \sum_{j=1}^{m} E_{CH_j} \]  

(13)

where

\( E_{CH_j} \) is the energy of CH \( CH_j \)

Fitness = \#f_1 + (1 - \#f_2), 0 < \# < 1

(14)

Our aim is to minimize the fitness.

4.2. Position update

Position update is done using QPSO (Yang et al., 2015). It is based on quantum computing. The state of the particle can be described with only a position vector unlike the PSO (Rao et al., 2017) where we need position vector and velocity vector. The updating of the position of particle is done by (6).

4.3. Cluster head selection

CH selection is done as in PSO-ECHS (Rao et al., 2017). The CHs are selected from the normal sensor nodes based on the energy efficiency. The fitness function is designed such that total energy consumption is minimized.

4.4. Cluster formation

Cluster formation is done using a weight function as in Rao et al. (2017). Sensor nodes use a weight function to join a CH which is denoted as \( CH_{\text{weight}} \). It is defined as follows:

\[ CH_{\text{weight}}(s_i, CH_j) = L \times \text{Energy factor} \]  

(15)

where

\[ \text{Energy factor} = \frac{E_{\text{residual}}(CH_j)}{\text{dist}(s_i, CH_j) \times \text{dist}(CH_j, BS) \times \text{deg}(CH_j)} \]  

(16)

Here

\( L \) is a constant, assumed to be 1

\( L \) = residual energy of CH \( CH_j \)

\( \text{dist}(s_i, CH_j) \) = distance between sensor \( s_i \) and \( CH_j \)

\( \text{dist}(CH_j, BS) \) = distance between \( \text{dist}(CH_j, BS) \) and BS

\( \text{deg}(CH_j) \) = degree of node \( CH_j \)

During cluster formation, each sensor node calculates the \( CH_{\text{weight}} \) using (15) and joins the CH with highest weight value.

Table 1 shows the comparison of energy consumption in LEACH, PSO-ECHS, and QPSOEEC when no. of sensors is 300, no. of CH’s is 15, and BS is at (100,100).

As can be seen from the table, the energy consumption using QPSOEEC is better than LEACH and PSO-ECHS. Similar results were obtained for the cases where number of sensors were varied from 400 to 700 and number of CH’s varied from 30 to 50. The resultant graphs are shown later. The overall algorithm is explained in Algorithm 1.
Table 1. Comparison of energy consumption in LEACH, PSOECHS, and QPSOEEC when no of sensors = 300 and CH = 15 and BS is at (100, 100)

| Rounds | LEACH | PSOECHS | QPSOEEC |
|--------|-------|---------|---------|
| 0      | 0     | 0       | 0       |
| 200    | 423.05| 21.03   | 6.93    |
| 400    | 505.86| 40.77   | 11.76   |
| 600    | 508.68| 58.96   | 15.32   |
| 800    | 541.05| 78.51   | 21.53   |
| 1000   | 544.79| 96.95   | 28.35   |
| 1200   | 563.71| 113.12  | 31.26   |
| 1400   | 571.39| 131.37  | 36.46   |
| 1600   | 584   | 146.16  | 42.02   |
| 1800   | 585   | 169.7   | 44.86   |
| 2000   | 587.4 | 189.53  | 52.12   |
| 2200   | 589.28| 209.83  | 58.67   |
| 2400   | 592.3 | 227.95  | 61.82   |
| 2600   | 595   | 246.95  | 66.77   |
| 2800   | 595.3 | 266.57  | 73.07   |
| 3000   | 598.21| 286.81  | 76.01   |
| 3200   | 598.85| 303.54  | 80.31   |
| 3400   | 599.8 | 310.75  | 85.3    |
| 3600   | 600   | 316.79  | 86.39   |
| 3800   | 600   | 321.58  | 91.48   |
| 4000   | 600   | 329.03  | 94.84   |
| 4200   | 600   | 335.02  | 97.56   |
| 4400   | 600   | 341.28  | 100.74  |
| 4600   | 600   | 347.23  | 104.83  |
| 4800   | 600   | 353.41  | 108.15  |
| 5000   | 600   | 359.5   | 111.4   |

Algorithm 1: QPSOEEC

**Input:** Sensor nodes $S = \{s_1, s_2, \ldots, s_{\text{sen}}\}$, size of swarm $N$, no. of dimensions of particle $D$

**Output:** Optimal positions of CHs with minimum energy consumption

1 begin
2 Initialize the particles
3 Calculate fitness using (14)
4 for $i = 0$ to No. of Rounds do
  5 Update position using QPSO
  6 Calculate fitness using (14)
  7 Find personal best, $P_{\text{best}}$ and global best, $G_{\text{best}}$
  8 Use $G_{\text{best}}$ and $G_{\text{best}}$ for CH selection
  9 Form clusters
10 end
11 Calculate total energy consumed at the end of predefined No. of Rounds
12 Stop
13 end
5. Performance evaluation

5.1. Simulation environment

Our algorithm was tested using C (Dev C++) and results plotted using MATLAB (R2015a) on Windows 8 Pro Platform. For our simulation, the number of sensor nodes was varied from 300 to 700 and CHs from 15 to 50. Each Sensor node is initialized with energy of 2J. The sensing field is of $200 \times 200$ m$^2$ (Table 2). In the test environment, comparison of the Total Energy Consumption is done while varying the number of sensors from 300 to 700 and CHs from 15 to 50. For the first case of 300 Sensors and 15 CHs, the BS was placed at (100,100), (200,200), and (300,300) and the results were obtained. Furthermore, the number of sensors was changed to 400, 500, and 700. The number of CHs was also varied as 35, 40, and 50.

The PSO parameters used as in Rao et al. (2017) are given in Table 3.

5.2. Performance metric used

The performance metric used is Energy Consumption. We have considered the Total Energy consumption for a particular number of Rounds (in our case 5000). The Energy consumption increases as the number of Rounds increases. The LEACH, PSO-ECHS and QPSOEEC algorithms are ranked by comparing the Total energy consumption while varying the number of sensors from 300 to 700 and CHs from 15 to 50. QPSOEEC outperforms both LEACH and PSO-ECHS by a large margin. The values of Total Energy consumption using QPSOEEC are 89% better than LEACH and 71% better than PSO-ECHS. Similar results were obtained when BS was placed at (200,200) and (300,300) with QPSOEEC outperforming LEACH and PSO-ECHS. QPSOEEC got better results when number of sensors was varied from 300 to 700 and number of CHs was varied from 15 to 50.

The graphs plotted for the case when number of Sensors is 300 and number of CH's is 15 with BS position varying as (100,100), (200,200), and (300,300) are shown in Figures 1–3. The graphs

| Table 2. Network parameters |
|-----------------------------|
| Parameters        | Value                  |
| Area              | $200 \times 200$ m$^2$ |
| Base station      | (100,100), (200,200), (300,300) |
| No. of sensors    | 300–700                |
| No. of CH's       | 15–50                  |
| $\epsilon_{\text{elec}}$ | 50 nJ/bit             |
| $\epsilon_{\text{fs}}$ | 10 pJ/bit/m$^2$       |
| $\epsilon_{\text{mp}}$ | 0.0013 pJ/bit/m$^4$   |
| Packet length     | 4000 bits              |
| Message size      | 500 bits               |

| Table 3. PSO parameters |
|-------------------------|
| Parameters             | Value |
| No. of particles       | 30    |
| $C_1$                  | 2.0   |
| $C_2$                  | 2.0   |
| $\alpha$               | 0.3   |
| $\omega$               | 0.7   |
| $D$                    | 15–50 |
| No. of iterations      | 100   |
Figure 1. No. of sensors = 300, CH = 15, and BS at (100,100).

Figure 2. No. of sensors = 300, CH = 15, and BS at (200,200).

Figure 3. No. of sensors = 300, CH = 15 and BS at (300,300).
plotted when number of Sensors is 300 and number of CHs is 30 with BS position varying as (100,100), (200,200), and (300,300) are shown in Figures 4–6. Figures 7–9 depict the scenario when number of sensors is 400 and number of CHs is 40 with the BS position varied as (100,100), (200,200), and (300,300). Figures 10–12 depict the scenario when number of Sensors is 500 and number of CH’s is 50 with the BS position varied as (100,100), (200,200), and (300,300). The graphs obtained when number of Sensors is 700 and number of CHs is 35 and BS position varied as (100,100), (200,200), and (300,300) are shown in Figures 13–15.

6. Conclusion and future work
In this paper, an algorithm called QPSOEEC is proposed. The proposed algorithm uses only one parameter, the position vector whereas PSO requires both position and velocity vectors. The simulation results of QPSOEEC algorithm are compared with existing algorithms LEACH and PSO-ECHS. The no. of sensors and CHs and also the position of the BS were varied. The experimental results show that our algorithm performs better than LEACH and PSO-ECHS in terms of Total Energy Consumption. The values of total energy consumption using QPSOEEC are 89\% better than LEACH and 71\% better than PSO-ECHS.
Figure 6. No. of sensors = 300, CH = 30, and BS at (300,300).

Figure 7. No. of sensors = 400, CH = 40, and BS at (100,100).

Figure 8. No. of sensors = 400, CH = 40, and BS at (200,200).
Figure 9. No. of sensors = 400, CH = 40, and BS at (300,300).

Figure 10. No. of sensors = 500, CH = 50, and BS at (100,100).

Figure 11. No. of sensors = 500, CH = 50, and BS at (200,200).
Figure 12. No. of sensors = 500, CH = 50, and BS at (300,300).

Figure 13. No. of sensors = 700, CH = 35, and BS at (100,100).

Figure 14. No. of sensors = 700, CH = 35, and BS at (200,200).
The challenge with Quantum computing is that as of now Quantum computers don’t exist in a full-fledged form. So, it is not possible to test our algorithm on an actual quantum computer. Once Quantum computers become available, the power of our algorithm can be realized.

Acknowledgements
The authors thank National Institute of Technology Karnataka (NITK), Surathkal for giving us the resources in terms of Lab support to do the experiments and NMAMIT, Nitte for supporting the Research.

Funding
The authors received no direct funding for this research.

Author details
Pradeep Kanchan
E-mail: pradeepis_2000@yahoo.com
ORCID ID: http://orcid.org/0000-0002-9320-5288
Shetty D Pushparaj
E-mail: prajshetty@gmail.com
1 Department of MACS, National Institute of Technology Karnataka (NITK), Surathkal, India.

Citation information
Cite this article as: A quantum inspired PSO algorithm for energy efficient clustering in wireless sensor networks, Pradeep Kanchan & Shetty D Pushparaj, Cogent Engineering (2018), 5: 1522086.

References
Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6(1), 58–73. doi:10.1109/6426.985692
Guru, S., Halgamuge, S., & Fernando, S. (2005). Particle swarm optimisers for cluster formation in wireless sensor networks. Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on (pp. 319–324). IEEE.
Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications, 1(4), 660–670. doi:10.1109/TWC.2002.804190
Kennedy, J. (2011). Particle swarm optimization. In Encyclopedia of machine learning (pp. 760–766). Springer.
Latiff, N. A., Tsimenidis, C. C., & Sharif, B. S. (2007). Energy-aware clustering for wireless sensor networks using particle swarm optimization. Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on (pp. 1–5). IEEE.
Lindsey, S., & Raghavendra, C. S. (2002). Pegasis: Power-efficient gathering in sensor information systems. In Aerospace conference proceedings, 2002 (Vol 3, p. 3). IEEE.
Liu, J., Sun, J., & Xu, W. (2006). Quantum-behaved particle swarm optimization with adaptive mutation operator. Advances in Natural Computation, 959–967.
Loscri, V., Morabito, G., & Marano, S. (2003). A two-levels hierarchy for low-energy adaptive clustering hierarchy (tl-leafch). In Vehicular Technology Conference, 2005. VTC-2005-Fall. 2005 IEEE 62nd (Vol 3, pp. 1809–1813). IEEE.
Pant, M., Thangaraj, R., & Abraham, A. (2008). A new quantum behaved particle swarm optimization. In Proceedings of the 10th annual conference on Genetic and evolutionary computation (pp. 87–94). ACM.
Rao, P. S., Jana, P. K., & Banka, H. (2017). A particle swarm optimization based energy efficient cluster head selection algorithm for wireless sensor networks. Wireless Networks, 23(7), 2005–2020. doi:10.1007/s11276-016-1270-7
Sun, J., Fang, W., Wu, X., Palade, V., & Xu, W. (2012). Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection. Evolutionary Computation, 20(3), 349–393. doi:10.1162/EVCO_a_00049
Sun, J., Feng, B., & Xu, W. (2004). Particle swarm optimization with particles having quantum behavior. In Evolutionary Computation, 2004. CEC2004. Congress on (Vol 1, pp. 325–331). IEEE.
Tillett, J., Rao, R., & Sohain, F. (2002). Cluster-head identification in ad hoc sensor networks using particle swarm optimization. In Personal Wireless Communications, 2002 IEEE International Conference on (pp. 201–205). IEEE. doi:10.1049/i1059-0889(2002/er01)
Xiangning, F., & Yulin, S. (2007). Improvement on leach protocol of wireless sensor network. In Sensor
Yang, Z.-L., Wu, A., & Min, H.-Q. (2015). An improved quantum-behaved particle swarm optimization algorithm with elitist breeding for unconstrained optimization. Computational Intelligence and Neuroscience, 41.

Yassein, M. B., Khamayseh, Y., & Mardini, W. (2009). Improvement on LEACH protocol of wireless sensor network (VLEACH). International Journal of Digital Content: Technology and Its Applications, 3, Citeseer. doi:10.4156/jdcta

Yin, Q., Li, W., Zhang, X., & Huo, F. (2010). Continuous quantum particle swarm optimization and its application to optimization calculation and analysis of energy-saving motor used in beam pumping unit. In Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on (pp. 1231–1235). IEEE.

Zhou, H., Jiang, Z., & Xiaoyan, M. (2006). Study and design on cluster routing protocols of wireless sensor networks.