Heurstic Approaches For Underwater Sensing And Processing Deployment

Hesham Alhumyani, Reda Ammar
Computer Science and Engineering Department,
University of Connecticut
Storrs, USA
{Hesham, Reda}@engr.uconn.edu

Raafat Elfouly
Computer Engineering Department,
Cairo University
Cairo, Egypt
Relfouly@eng.cu.edu.eg

Ayman Alharbi
Computer Engineering Department,
Umm AL-Qura University
Makkah, Saudi Arabia
aarharbi@uqu.edu.sa

Abstract—**Limited bandwidth and high propagation delay are two significant challenges in the field of underwater sensor networks. Underwater processing nodes enhance the performance of underwater sensor networks by efficiently using channel capacity, reducing end-to-end delay, and extending the network lifetime. Underwater processing nodes can perform local computations (in-network processing), such as compression, mining, and feature extraction, on the collected data before transmission. The locations of processing nodes have to be carefully chosen to maximize the benefits in a cost-effective way. In this paper, three heuristic approaches are proposed to efficiently solve the processing node deployment optimization problem. The simulation results confirm the benefits of the proposed algorithms for solving such NP problems. The results were compared with the optimal solution and suggest that these heuristics are nearly optimal for practical cases.**

I. INTRODUCTION

More than two thirds of the earth's surface is covered with water. In recent years, interest in underwater wireless sensor networks (UWSNs) and their applications has been growing rapidly. Different network applications have different requirements, which results in a variety of designs. UWSNs have many practical applications such as 3-D mapping, fish detection and classification, surveillance, intrusion detection, and military applications [1] [2]. Unlike terrestrial wireless sensor networks, UWSNs cannot use conventional communication techniques such as radio-frequency and optical waves. Although these types of waves have very high bandwidths, they attenuate and are absorbed very quickly, within a few meters (radio) or tens of meters (optical). Therefore, for communication, UWSNs use acoustic waves, which can travel for long distances; this is a practical solution for the underwater environment. However, the harsh underwater environment creates many challenges for UWSNs using acoustic waves [3]. First, acoustic waves have limited bandwidth, which can greatly increase delays and energy consumption in data transmission. Second, due to the properties of water, such as pressure, salinity, and temperature, acoustic waves move extremely slowly, with a propagation speed of around 1500 m/s. This propagation delay causes a large end-to-end delay. Such constraints can greatly affect the performance of UWSNs and limit the possible interactive and monitor applications that require large data sets.

Normal and typical underwater sensor nodes are bounded to data sensing, transmitting, and forwarding. They are dependent on batteries for their operation, but due to the difficulties in reaching the deployment sites, it is usually hard to replace or recharge these batteries. Batteries are depleted quickly when large volumes of data are transmitted, as underwater modems are energy-hungry devices. Therefore, energy consumption is a major performance metric that has to be taken into consideration when designing network architecture. As the underwater sensor nodes have limited processing capability, transmitting a large volume of data [4] requires a significant amount of time. Moreover, when the size of the dataset is above a certain threshold, the whole dataset is only recovered at the end of the mission; hence, there is no possibility of real time monitoring, and data analysis and processing must be postponed until it can be performed off shore.

It is generally easier to modify the data than to improve the communication channel. Therefore, in-network processing is one practical solution to the above problem, as it can process the collected raw data through aggregation, fusion, and mining. For example, a processing network can either transmit only the result (output) of a certain application or can eliminate less valuable bits from the dataset so that only the required information is transmitted. One fundamental task of the medium access control (MAC) protocol is to help sensor nodes to access their shared medium by scheduling a time frame for each node, thus avoiding collisions and providing sophisticated ways to reduce the waiting time for the channel access. While waiting for access to the medium (ideal time), a node can perform local processing on the collected data during the link unavailability. This can lead to a reduction in the amount of transmitted data and improved bandwidth utilization, which will improve the power consumption and end-to-end delay. Therefore, in a previous study we proposed a new type of node called a processing node [5]. However, as processing nodes have high power capability with large data storage and a high feature processor, they are also more expensive than regular underwater sensor nodes. Thus, there is a tradeoff between the number of processing nodes and network performance constraints. In our previous study [6], we developed a combinatorial optimization problem to solve the processing nodes deployment. In this paper, we present heuristic approaches to solve the processing nodes deployment optimization problem efficiently in polynomial time.

II. RELATED WORK

Recently, big data retrieval has been studied extensively for UWSNs. One approach for large data acquisition is to build an underwater wired network by connecting a fixed wired link to the surface level gateway [7]. Cable-based networks are either copper or fiber optic cables. The main advantages of wired networks are higher data rates and lower power requirements than wireless transmission. However, if any part of a wire in the network is destroyed, the communication system will completely or partially fail. Another method of data transmission is to use autonomous underwater vehicles (AUV) equipped with short range optical communication equipment visit the fixed underwater nodes frequently [8] [9]. Although AUVs enhance network performance and can help to overcome the mobility problem of UWSNs, they do have several limitations. First, communication is enabled only when both the sender and the receiver are accurately optically aligned (line-of-sight data transfer). Second, they have limited storage capabilities, yet some applications require the transmission of large data sets. Third, as AUVs use optical communication, distance becomes a major factor in large data transmission due to scattering and light absorption in water. A practical solution is therefore needed that uses wireless sensor networks based on acoustic communications. In our previous study [5], a new underwater computing system architecture called underwater sensing and processing networks (USPN) was proposed. The main goal of USPNs is to allow efficient data processing techniques to be performed underwater (in-network). Two basics architectures were shown: the first architecture was composed of a single processing node communicating directly with the sink, and the second architecture was formed with a single processing node connected to the surface sink via relay nodes. In these systems, acoustic signals are used efficiently and the network performance is enhanced. Then, by taking into account more costly and complex data processing algorithms, in [10] we presented more advanced architectures for USPNs that aimed to perform big data analytics. We showed how to integrate and exploit both processing techniques and pipelining transmission to increase the system throughput and network performance speed. Both [5] and [10] used linear (chain) topology. In this paper, we study the processing nodes deployment on a mesh topology and investigate how this can improve network performance.

III. UNDERWATER PROCESSING NODES DEPLOYMENT STRATEGY

1. **Network model:** The processing nodes deployment problem is formulated as a 3-D graph optimization problem, where the nodes of the graph are defined as the set of surface-level gateways and underwater sensor nodes. The underwater sensor nodes can be replaced with any underwater processing sensor nodes, as they act as candidate locations. As illustrated in Fig. 1, the system architecture is composed of different types of nodes with different roles: regular sensor node, relay node, processing sensor node, and surface-level gateway node. To obtain the best deployment for the underwater processing

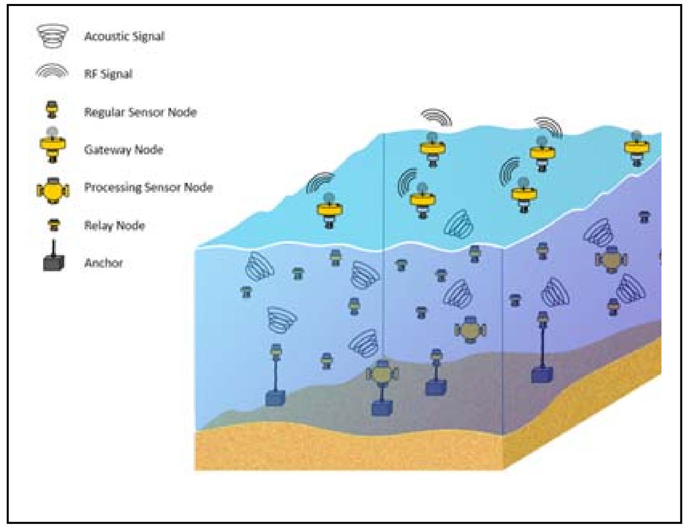

Fig.1 Network architecture

nodes, we need to carefully select their locations from the pre-existing candidate underwater sensors positions while satisfying a set of constraints such as Per-node and end-to-end flow conservation, deployment cost (budget) constraints, and data quality. The target is to optimize a set of objective functions.

2. **Assumptions:** In this paper, we assume static (fixed) underwater deployment. As some zones of the network have more phenomena than other areas, we assume that the data generated by each node is different. Furthermore, all of the acoustic transceivers are homogeneous and a fixed communication range is set for each node. At the surface level, we assume that packets delivered to any one of the surface-level gateways are also delivered to the control station.

IV. PROBLEM FORMULATION

In this section, we present the detailed network model by first defining the basic theoretical graph components and then discussing the set of constraints that need to be satisfied to optimize the different object functions.

1. Definitions

The network is modeled as a graph $G = \{V, E\}$, where V represents the set of all of the nodes in the network and E is the set of all of the communication links between nodes.

1.1 Nodes: Let U be the set of all of the underwater sensor nodes, T be the set of surface gateway nodes, P be the set of processing nodes, where $P \subset U$, $pi \in U \ \forall i=1,2,....,K.$, *i.e.,* $V = U \cup T$. Let $I(u)$ be the set of nodes within the communication range of node u, i.e., $I_{(u)} = \{ v : v \quad v \in U , u \neq v \ ; d_{(u,v)} < C \}$, where $d_{(u,v)}$ denotes the Euclidean distance between u and v, and C is the communication range.

1.2 Edges: E_{out} and E_{in} are the outgoing and incoming links i.e., $E_{out}(v) = \sum_{w_i \in I(v)} e(v, w_i)$ and $E_{in}(v) = \sum_{w_i \in I(v)} e(w_i, v)$, where e(v,w) represent the link between node v and w such that $v \in V$, $w \in I(v)$.

1.3 Data flow and traffic generation: Let *f(e)* be the total data flow on edge e in (packets/s). Each node is associated with a local data generation rate (traffic load) *i.e.,* g(v) at $v \in V$. We define G as the total data generation rate for the whole network i.e., $G = \sum_{vi \in V} g(vi)$.

1.4 Processing ratio: we define the processing ratio α as the percentage of the original data volume that has been reduced after processing i.e.,

$$\alpha = \begin{cases} x, 0.1 < x < 0.9 & \textit{if the node processed data} \\ 1 & \textit{otherwise.} \end{cases}$$

2. Constraints

2.1 Per-Node flow conservation: As we aim to perform local processing on the generated data, the flow conservation for underwater sensor nodes can be defined as when the sum of the flows leaving a node is less than or equal to the sum of the flows entering that node, plus the local data generation rate.

$$\sum_{eo \in E\ out(v)} f(eo) \le \left(\sum_{ei \in E\ in(v)} f(ei) + \alpha g_v \right) \tag{1}$$

2.2 End-to-End flow conservation: Data generated by any source can be received at any surface gateways. Therefore, the flow conservation implies that the total data generation rate must be less than or equal to the total received data rate for all of the surface node sensors.

$$\sum_{tj \in T} \sum_{ei \in E\ in(vi)} f(ei) \le G \tag{2}$$

2.3 Number of processing nodes: A processing node is more expensive than a regular underwater sensor node and hence the number of processing nodes is limited to k nodes, according to the available budget.

$$\sum_{vi \in P} vi \le K \tag{3}$$

2.4 Data quality constraint: Due to underwater environment conditions, the constraints on the quality of data may vary among different zones of the network. For instance, for images the amount of light reaching the sensor is crucial and might affect image quality. Therefore, each candidate underwater sensor node has *Qi* constraints.

$$\alpha_i \le Q_i \tag{4}$$

2.5 Interference constraints: In this paper, we assume a simple interference model, in which a node cannot send and receive at the same time. Therefore, the total incoming and outgoing flow of a certain node cannot exceed the communication link capacity. Hence, the interference constraint is written as follows:

$$\sum_{ei \in E_{in(v)}} f(e_i) + \sum_{eo \in E_{out(v)}} f(e_o) \le B \tag{5}$$

3. Objective functions

Within the above constraints, we need to find the optimal processing node locations for a limited number of processing nodes, to minimizing the expected end-to-end delay and the expected power consumption.

3.1 Minimizing expected end-to-end delay: To calculate the end-to-end delay for a data packet, we need the sum of the per-hop delay at every link (edge) in the path from the data source to the destination (surface-level gateways). The per-hop delay t on each edge e can be represented as follows:

$$\tau(e) = \tau_{tr}(e) + \tau_{p(e)} + \tau_{q(e)} + \tau_{proc}(e) \quad \text{i.e.,}$$

$$\tau_p = \frac{d}{\rho}, \text{ and } \tau_{tr} = \frac{L}{B},$$

Where τ_{tr} is annotated as the transmission delay, which is equal to the packet length (L) in bits divided by channel bandwidth (B) in bits per seconds. The propagation delay (τ_p) is the distance (d) of edge e divided by the acoustic propagation speed in water. Therefore, the objective function to minimize the expected end-to-end delay for all packets is as follows:

$$Minimize\ E[\tau(e)],$$

Where $E[\tau(e)] = \frac{1}{G} \sum_{e \in E} (f(e).\tau(e))$.

3.2 Minimizing the expected power consumption: In this paper, we use the sonar equation model shown in [12]. The detailed mathematical model is shown in [6] [12] and the objective function is written as follows:

$$\text{Min}\,imize\ E[P(e)],$$

$$i.e.\ \ E[P(e)] = \frac{1}{G} \sum_{e \in E} (f(e).P(e)),$$

Where *P(e)* is the power consumed in edge *e*.

V. HEURISTIC UNDERWATER PROCESSING NODES DEPLOYMENT APPROACHES

The processing node optimization deployment problem is solved using the ILP (integer linear programming) formulation shown in Section 3. Moreover, ILP for similar problems such as the Facility Location Problem (FLP), and the Warehouse Location Problem (WLP) are considered to be NP-complete problems. When there is a large number of candidates, finding the optimal solution becomes unfeasible. Moreover, due to the limitations of computer speed, it can take several hours or even days to solve these problems. Therefore, heuristic approaches can be used when finding an optimal solution is impossible or impractical. Thus, we propose several heuristic algorithms for the purpose of finding a near optimal solution for deploying underwater processing nodes.

1. Depth-based deployment

Underwater sensor nodes that reside on the bottom of the sea (or those deployed in the deepest possible areas), can greatly increase end-to-end delays and power consumption, Data transmitted from a data source at the deepest layer travel via multi-hops along a certain path, and the communication cost is high. Moreover, the communication cost can be further increased when the number of nodes on the route is increased. Therefore, in this algorithm, the nodes deployed at great depths are given higher weights in the cost and efficiency analysis. Furthermore, underwater sensor nodes closer to the surface-level gateways are burdened with high traffic, which rapidly depletes their batteries and shortens their lifetimes. One way to alleviate this problem is to deploy many surface-level gateways, as nodes closer to the surface can be connected to the nearby surface-level gateway, which is a short distance away. Hence, before starting to add processing nodes to the network, we need to find the optimal locations for the surface-level gateway deployment based on the work presented in [11]. After that, we begin by deploying the underwater processing nodes starting from the deepest candidate underwater nodes and according to the available number. Every time a processing node is deployed, the amount of transmitted data is reduced and hence the network traffic is updated. Therefore, at each iteration (when additional processing is added), there is a need for the surface-level gateways to be redeployed, as the network traffic has been updated. Furthermore, when choosing candidate locations, we assess whether there is an improvement compared to the network performance before adding the new processing nodes. Thus, we define the improvement factor as follows:

Performance_gain =

$$\frac{\text{delay before}}{\text{delay after}} \times \frac{\text{power before}}{\text{power after}} \times \frac{\mathrm{M}_{\boldsymbol{cost}}\text{ before}}{\mathrm{M}_{\boldsymbol{cost}}\text{ after}}. \qquad (6)$$

The criteria are based on three main factors: delay improvement, power improvement, and monetary cost. If the performance gain is greater than one, then an improvement is obtained and hence the processing node is kept in the chosen candidate location. Otherwise, the regular underwater sensor node is returned to its original location. These steps are repeated until the last nodes in the last layer (the one closest to the surface) have been tested or some other constraints have been satisfied such as real time constraints, power consumption, or budget constraints. Algorithm 1 shows the detailed procedure of the depth-based deployment.

Algorithm 1: Depth-based approach

Give an id number to each node 1, 2… n
Surface –level gateways *= K*
ProcessingNodes *= V*
Gateway deployment *(K)*
$\boldsymbol{d_0}$ = *calculateDelay*
$\boldsymbol{p_0}$ = *calculatePower*
$\boldsymbol{m\_cost_0}$ =*calculateDeploymentCost*
$\boldsymbol{cost_0} = \boldsymbol{d_0} \times \boldsymbol{p_0} \times \boldsymbol{m\_cost_0}$
For *i* = *1* **to** *n* **do**
Select the candidate nodes according to their depth
Set the selected node to be processing node
Gateway deployment *(K)*
$\boldsymbol{cost_i} = \boldsymbol{d_i} \times \boldsymbol{p_i} \times \boldsymbol{m\_cost_i}$
performance_gain = $\frac{cost_{i-1}}{cost_i}$
if (***performance_gain*** *>=1*)
Keep the processing node at this location
ProcessingNodes *=* ***ProcessingNodes****-1*
else
Set this candidate node to be a regular node
if *(ProcessingNodes* *=0* **||*Performance*** *>***threshold)***
exit
End For

2. Data volume-based deployment

Underwater sensor nodes are sparsely deployed and they work independently. In addition, different zones of the network can observe different phenomena, leading to variation in the amount of collected data at each underwater sensor node. Hence, an underwater processing node deployment using the depth-based algorithm might not always obtain the near optimal solution. Moreover, in a certain path there might be some nodes that collect larger data volumes and have a greater effect on the network performance than the deepest nodes. Therefore, in this section we present an algorithm that depends on the size of the data. As for the previous algorithm, we start with the surface-level gateway deployment procedure given in [11], to ensure that the optimal network performance is obtained before the deployment of any processing node and

Algorithm 2: Data volume-based approach

Give an id number to each node 1, 2… n
Surface –level gateways *= K*
ProcessingNodes *= V*
Gateway deployment *(K)*
$\boldsymbol{d_0}$ = *calculateDelay*
$\boldsymbol{p_0}$ = *calculatePower*
$\boldsymbol{m\_cost_0}$ =*calculateDeploymentCost*
$\boldsymbol{cost_0} = \boldsymbol{d_0} \times \boldsymbol{p_0} \times \boldsymbol{m\_cost_0}$
Sort all candidate nodes according to their data size
Repeat
Select the candidate nodes with largest data volume
Set the selected node to be processing node
Gateway deployment *(K)*
$\boldsymbol{cost_i} = \boldsymbol{d_i} \times \boldsymbol{p_i} \times \boldsymbol{m\_cost_i}$
performance_gain = $\frac{cost_{i-1}}{cost_i}$
if (***performance_gain*** *>=1*)
Keep the processing node at this location
ProcessingNodes *=* ***ProcessingNodes*** *- 1*
else
Set this candidate node to be a regular node
if *(ProcessingNodes* *=0* **||*Performance*** *>***threshold)***
exit
Until (*the end of sorted list is reached*)

then we improve on that baseline. Then, the candidate locations are sorted according to their collected data volumes. After that, we iterate over the sorted list and begin with the underwater sensor node (candidate position) with the largest volume and replace that regular underwater sensor node with an underwater processing node. Again, redeployment of the surface-level gateways is required as the network traffic will be updated. Furthermore, the decision to deploy an underwater processing node in a certain candidate location depends on (6). If no improvement is gained, then the regular underwater sensor is returned, otherwise the underwater processing node is deployed. We continue to iterate over the sorted list until there are no further available processing nodes or specific network performance constraints are reached. Algorithm 2 shows the detailed procedure for the data volume-based deployment.

3. Hybrid depth and data-volume approach

Based on the above two approaches, we propose a hybrid algorithm that integrates the depth-based and data volume-based algorithms. In this case, the effect of the deepest deployed underwater sensor nodes in increasing the communication cost could be alleviated while also taking data volumes into consideration. Algorithm 3 shows the detailed procedure for the hybrid approach. The first phase of hybrid approach is to start the surface-level gateway procedure for the optimal network performance before performing any data processing. Underwater sensor nodes are floated at different depths to observe a given phenomenon. Furthermore, in this paper, we deploy the underwater sensor nodes at different layers, such that each layer ID is associated with a certain depth. Therefore, we divide the network into different layers and begin with the deepest layer. Next, the candidate underwater sensor nodes at the deepest layer are sorted according to their data volumes. The next step is to iterate over the sorted list and use the objective function in (6) to test whether an underwater processing node should be used at the candidate location. If there are more available processing nodes when the sorted list is completed, we jump one further step to a higher layer and perform the same procedure as above. These steps are repeated until the last layer (closest to the surface) is reached, no more processing nodes are available, or some of the network performance constraints are satisfied.

VI. PERFORMANCE EVALUATION

To verify the benefits of the proposed approaches, we simulated our techniques. To evaluate the performance of each approach, we compared the generated solutions with the optimal solution.

1. Simulation settings

In our simulation experiments, 141 nodes were deployed at different layers (depths) in a *500 m × 500 m × 500 m* three-dimensional topology. The number of layers was fixed as six layers and the distance between each layer was set to 100 m. Each layer consisted of a set of 25 randomly deployed underwater sensor nodes. We defined the first layer as the deepest layer with a depth of 500 m and the last layer was the surface-level layer with a depth of 0 m. In the last layer, 10 surface-level gateways were chosen from 16 candidate locations that were uniformly deployed with a 4 × 4 planar mesh according to the surface-level gateway deployment procedure given in [13]. In our experiments, we fixed some parameters to test and evaluate the work. At the physical layer parameters, we fixed the signal-to-noise-ratio (SNR) at 15 dB, the frequency at 20 kHz, and the spreading factor y at 15. The communication range was fixed for all of the nodes at 150 m. We fixed the packet length as 400 bits and the channel capacity as 10 kbps. The number of processing nodes used in the experiments was 25 nodes with initial energy ($\vec{E}_{initial}$) 1e6 joules, whereas the initial energy (E_{intial}) for the regular underwater node was 1e4 joules. For the data processing technique, we considered and tested the LZW compression algorithm using an ARM Cortex-A8 processor.

Algorithm 3: Hybrid approach

Give an id number to each layer(depth)
Each layer has M candidate nodes
LayerCount *= R*
Surface –level gateways *= K*
ProcessingNodes *= V*
Gateway deployment *(K)*
d_0 = *calculateDelay*
p_0 = *calculatePower*
m_cost_0 =*calculateDeploymentCost*
$cost_0 = d_0 \times p_0 \times m\_cost_0$
  For *j = 1* **to** *R* **do**
Sort all candidate nodes in layer [*j*] *according to their data size*
    For *i = 1* **to** *M* **do**
Select the candidate nodes with largest data volume
Set the selected node to be processing node
Gateway deployment *(K)*
$cost_i = d_i \times p_i \times m\_cost_i$
$performance\_gain = \frac{cost_{i-1}}{cost_i}$
if (***performance_gain*** *>=1*)
  Keep the processing node at this location
  ProcessingNodes = ***ProcessingNodes****-1*
  else
  Set this candidate node to be a regular node
if *(ProcessingNodes =0 ||Performance >threshold)*
  exit
    End For
  End For

2. Simulation results

To accurately assess the quality of the solutions for the proposed algorithms, we performed extensive simulations for different scenarios; the data sources collected different amounts of data in each scenario and took the average of all of the runs. Fig. 2 shows the quality of the solutions for the expected end-to-end delay; it is clear that the quality loss (increased delay) was less than 10%. The quality of the solutions with regards to the expected power consumption is shown in Fig. 3. The percentage of the quality loss (increased power) was less than 6%. Therefore, the obtained results were very close to the optimal solution.

VII. CONCLUSIONS

In this study, we developed algorithms for determining the best placements for processing nodes, and identified a better solution for the processing node optimization problem. Using these solutions can increase the probability of finding a nearly optimal solution for the node placement problem, thus increasing the networks' ability to solve large problems accurately. The three presented approaches show that the obtained results were close to the optimal solution. Finally, more sophisticated algorithms are currently being studied, and will be presented in the future papers.

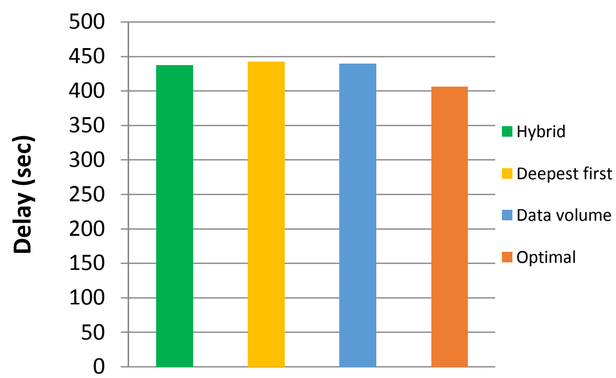

Fig. 2 Quality of solutions for the expected end-to-end delay

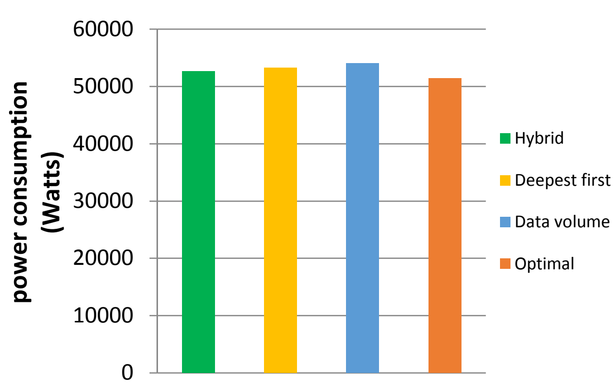

Fig. 3 Quality of solutions for the expected power consumption

VIII. REFERENCES

[1] G. Xie and J. Gibson, "A networking protocol for underwater acoustic networks," DTIC Document, Tech. Rep., 2000.

[2] J. G. Proakis, J. A. Rice, E. M. Sozer, and M. Stojanovic, "Shallowwater acoustic networks*," Encyclopedia of Telecommunications, 2003.

[3] J.-H. Cui, J. Kong, M. Gerla, and S. Zhou, "The challenges of building mobile underwater wireless networks for aquatic applications," Network, IEEE, vol. 20, no. 3, pp. 12–18, 2006.

[4] J. Proakis, E. M. Sozer, J. A. Rice, and M. Stojanovic, "Shallow water acoustic networks," IEEE Communications Magazines, pp. 114–119, November 2001.

[5] Alharbi, A.; Alhumyani, H.; Tolba, S.; Ammar, R.; Jun-Hong Cui, "Underwater Sensing and Processing Networks (USPN)," in *Computers and Communication (ISCC), 2014 IEEE Symposium on*, vol., no., pp.1-7, 23-26 June 2014 doi: 10.1109/ISCC.2014.6912472.

[6] H. Alhumyani, A. Alharbi, S. Tolba, R. Ammar, "Efficient surface level gateway deployment using underwater sensing and processing networks," OCEANS'15 MTS/IEEE Washington (Accepted).

[7] B. M. Howe, T. McGinnis, and H. Kirkham, Sensor networks for cabled ocean observatories. EGS - AGU - EUG Joint Assembly, Abstracts from the meeting held in Nice, France, 6–11 April 2003, abstract #12598 (Apr. 2003), 12598–+

[8] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke, "Data collection, storage, and retrieval with an underwater sensor network," in Proceedings of the 3rd international conference on embedded networked sensor systems. ACM, 2005, pp. 154–165.

[9] M. Tivey, "A low power, low cost, underwater optical communications system," Ridge 2000 Events (Apr. 2004), pp. 27–29.

[10] Alharbi, A.; Ammar, R.; Alhumyani, H.; Rajasekaran, S.; Jun-Hong, "Efficient pipeline architectures for Underwater Big data analytic," in Signal Processing and Information Technology (ISSPIT), 2014 IEEE International Symposium on , vol., no., pp.000161-000166, 15-17 Dec. 2014 doi: 10.1109/ISSPIT.2014.7300581

[11] S. Ibrahim, J.-H. Cui, and R. Ammar, "Surface-Level gateway deployment for underwater sensor networks" in Military Communications MilCom07, November 2007.

[12] R. Urick, "Principles of underwater sound." McGraw-Hill, New York, 3rd edition, 1983.

[13] Ibrahim, S.; Cui, J.-H.; Ammar, R., "Efficient surface gateway deployment for underwater sensor networks," in Computers and Communications, 2008. ISCC 2008. IEEE Symposium on, vol., no., pp.1177-1182, 6-9 July 2008 doi: 10.1109/ISCC.2008.4625609.