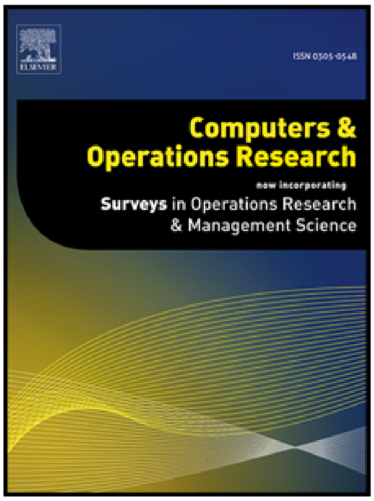


Efficient Sensor Network Management for Asset Localization

Andrei Soeanu, Sujoy Ray, Jean Berger, Mourad Debbabi

PII: S0305-0548(18)30155-2
DOI: 10.1016/j.cor.2018.06.001
Reference: CAOR 4488

To appear in: *Computers and Operations Research*

Received date: 12 October 2017
Revised date: 13 March 2018
Accepted date: 3 June 2018

Please cite this article as: Andrei Soeanu, Sujoy Ray, Jean Berger, Mourad Debbabi, Efficient Sensor Network Management for Asset Localization, *Computers and Operations Research* (2018), doi: 10.1016/j.cor.2018.06.001

Highlights

- Asset localization over resource constrained wireless sensor network (WSN).
- WSN management to assign heterogeneous sensors to localize assets with minimized error.
- Localization problem extension considering an overall WSN energy budget restriction.
- Heuristic solution approach using evolutionary learning and meta-heuristic improvements.

Available online at www.sciencedirect.com

ScienceDirect

Computers & Operations Research 00 (2018) 1–25

Computers & Operations Research

www.elsevier.com/locate/procedia

Efficient Sensor Network Management for Asset Localization

Andrei Soeanu[a,*], Sujoy Ray[a], Jean Berger[b], Mourad Debbabi[a]

[a]Concordia University, Montreal, Quebec, Canada
[b]Defence Research and Development Canada - Valcartier, Quebec, Canada

Abstract

Asset localization represents an important application over wireless sensor networks (WSN) with a wide area of applicability ranging from network surveillance to search and rescue operations. In this paper, we address a research problem of network management where resource constrained sensors, in terms of capacity, sensing range and energy, are assigned to multiple targets in order to optimally localize assets with minimized error. We consider a heterogeneous network of omnidirectional sensors, each of which has an individual capacity to focus on a number of targets and a specific range to accurately estimate its distances to the targets that it is focusing on. A proper localization of each target requires a minimum of *K* (typically three) sensors where the target location is estimated using the intersection of the *K* range circles. We further analyze the problem under the constraint of a globally specified overall WSN energy budget which limits the possible assignments for the capacitated sensors. Restricting the energy budget leads to a trade-off between energy conservation and localization performance. In this context, we propose a heuristic solution approach leveraging evolutionary learning followed by meta-heuristic improvements based on target swapping among sensors. This approach actually minimizes a quantifier that is composed of the total localization area for all targets in addition to a penalty for each target if it is assigned less than minimum sensors. We provide an illustrative case study for the proposed approach and assess its effectiveness experimentally via benchmark results obtained on a data-set derived from known vehicle routing problem instances.

Keywords: Asset Localization, Wireless Sensor Network, Multi-dimensional Assignment.

1. Introduction

Logistics planning in situations of crisis involves designing and carrying out tasks such as: movement of assets (vehicles, goods, etc.), stationing assets at specific deploy points, etc. over the available supply chain networks. Characteristically, those preplanned tasks are often executed over perilous territory in hostile environments. In such an environment, asset localization represents an important aspect due to high likelihood of potential plan deviation due to unexpected and unaccounted events. A timely assessment of such deviations may provide a cutting edge in pursuit of successful and efficient completion of planed tasks. In this setting, we address the issue of asset localization under the assumption that the tasked assets may be unable to report their own location, being required to maintain radio silence in hostile environments. Thus, we consider an external set of heterogeneous sensors to perform asset localization. In this context of

*Corresponding author
Email address: a_soeanu@encs.concordia.ca (Andrei Soeanu)

multisensor-multitarget applications the core issue is the association of information from multiple sensors in order to support target identification [1]. The underlying problem is discussed in the general class of multi-dimensional assignment problems [2, 3, 4]. We aim at elaborating a framework for multisensor-multitarget asset localization over an instrumented space containing deployed sensors.

The multisensor-multitarget assignment problem represents an extension of the focus of attention (FOA) problem [5] where each target is assigned only to a pair of sensors while each sensor, within its range, has the capacity of focusing on only one target. We address a generalized problem where omnidirectional sensors have increased capacities while the targets require a minimum number of sensors (3 or more for appropriate localization). Our problem has NP-hard complexity since it extends the FOA problem. FOA is generally NP-hard, as it is a specific case of the multi-dimensional assignment problem [6] which is inapproximable in the general case [5, 7]. In this context, we propose an evolutionary learning based heuristic aimed at finding the best sensor-target assignment combination that would minimize the overall localization error and penalty levels. The proposed approach allows to effectively employ heterogeneous sensor networks where various sensors may have different capacities and sensing ranges.

We describe next the problem and modeling in Section 2 followed in Section 3 by a discussion of related work within our scope of interest. Section 4 elaborates the solution approach followed by an illustrative case study in Section 5. Benchmark results are documented in Section 6. Finally Section 7 provides the concluding remarks and comments on future work. Detailed benchmark results are provided in the Appendix.

2. Problem Description & Modeling

The problem of asset localization through WSN is composed of two sub-problems: A sensor assignment problem for each target (typically referred as the sensor focus of attention problem) and a localization problem for each target (deriving location of each target from a particular assignment of sensors to targets). The latter problem is approached by triangulation since we consider omnidirectional sensors with different sensing ranges and target focusing capacity. Thus, we consider that at least three different sensors need to be assigned to a target asset in order to properly localize it. In this setting, we aim at finding the most suitable multisensor-multitarget assignment that would result in minimized overall localization error.

2.1. Sensor Focus of Attention

Sensor focus of attention (FOA) and related sensor assignment problems represent an aspect of key importance in finding optimized configurations over a sensor network, particularly in the case where heterogeneous sensors can focus in a given time unit only on a subset of the targets in their coverage area. In this context, distinct sensors may have different abilities in terms of detection range (e.g. due to difference in elevation level) and target focusing capacity (i.e. one or more targets per time unit). For example, more capable sensors are often placed in locations deemed of higher importance (e.g. higher trafficability potential). An effective sensor to target assignment minimizes the localization error for the targets by appropriately assigning the sensors to the targets. The heterogeneity aspect introduces an extended level of complexity due to the vast number of possible sensor assignment combinations. This requires heuristic search methods for near-optimal sensor assignment. However, since the localization error cost can be trivially reduced by not assigning sensors to some targets, such solutions can be overlooked as being invalid or otherwise can be considered along with a sufficiently large penalty to render such solutions very unattractive.

2.2. Problem Statement

We investigate the problem of target localization as an application over WSN. Over an instrumented space, with a high level of abstraction, a sensor network can be represented by a fully bipartite graph $G_{SM} = \langle S, M, E_{SM} \rangle$ where S and M denote n sensors and m targets respectively. E_{SM} denotes a set of directed edges where each edge originates from an element of S to an element of M indicating a possible assignment of a sensor to a target. Given that a sensor node i and a target node j have two defined locations, d_{ij} is considered as its estimated distance with respect to target j. In this setting, let a set of variables, $x_{ij},\ i \in S,\ j \in M$, determine the assignment of sensors to the targets. Therefore, if $x_{ij} = 1$, an assignment exists between sensor i and target j. In the localization problem, a collection of sensors assigned to a target determine the potential location of the target within an intersection area. The latter is desired to be as small as possible. However, given a limited number of sensors with restricted abilities, such an optimal assignment of sensors to targets, represents a non-linear optimization problem constrained from the aforementioned limitations. In this context, we model the problem using the following objective function and constraints:

$$min \sum_{j=1}^{m} f_j(x_{1j}, \cdots, x_{ij}, \cdots, x_{nj}) \tag{1}$$

$$\text{Subject to: } \sum_{j=1}^{m} x_{ij} \leq C_i, \qquad \forall i \in N \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} \geq K \qquad \forall j \in M \tag{3}$$

$$d_{ij} \cdot x_{ij} \leq R_i \qquad \forall i \in N, j \in M \tag{4}$$

$$x_{ij} \in \{0, 1\} \tag{5}$$

Eq. (1) is the objective function where f_j determines an area around every target as per its sensor assignments. Computing such an area, in general case, is a complex mathematical problem [8]. However, in the context of a solution finding algorithm, the values can be considered as pre-computed on demand and retrieved using memory-efficient data structures. Eq. (2) states that a sensor cannot be assigned more than its capacity of focusing on C_i targets. Similarly, Eq. (3) puts a constraint that every target should be assigned to at least K sensors. Eq. (4) assures that a sensor i can be assigned to target j if and only if the distance is within the specified coverage or monitoring range of the sensor, R_i. The aforementioned model presents two challenges. First, it requires an equivalent linear model to assess the suitability of applying heuristic solution generation techniques. Second, some problem instances may have no feasible solutions for particular K values. This relates to the classical one-dimensional bin-packing problem which has NP-hard computation complexity. This can be addressed by relaxing Eq. (3) as will be detailed in the sequel.

To address the first challenge, we rewrite the aforementioned model in a linear form using higher-dimension decision variables. Let $x_{i_1,\ldots,i_n j}$ be an assignment of all sensors $(i_1 \ldots i_n)$ to target j which yields a localized area $c_{i_1,\ldots,i_n j}$ around target j. Precisely, $x_{i_1,\ldots,i_n j}$ is a multi-dimensional binary decision variable. Then, the following model captures the aforementioned equations, Eqs. (1)-(4), in linear programming:

$$min \sum_{j=1}^{m} \sum_{i_1=0}^{1} \cdots \sum_{i_n=0}^{1} c_{i_1,\cdots,i_n j} \cdot x_{i_1,\ldots,i_n j} \tag{6}$$

$$\text{Subject to: } x_{0,\ldots,0j} = 0, \qquad \forall j \in \{1, \cdots, m\}; \tag{7}$$

$$\sum_{i_1=0}^{1} \cdots \sum_{i_n=0}^{1} x_{i_1,\ldots,i_n j} = 1, \qquad \forall j \in \{1, \cdots, m\}; \tag{8}$$

$$\sum_{j=1}^{m} \sum_{i_1=0}^{1} \cdots \sum_{i_{p-1}=0}^{1} \sum_{i_{p+1}=0}^{1} \cdots \sum_{i_n=0}^{1} x_{i_1,\cdots,i_{p-1},1,i_{p+1},\cdots,i_n j} \leq C_{i_p}, \qquad \forall p \in \{i_1 \cdots i_n\} \tag{9}$$

$$d_{i_p j} \cdot \Big(\sum_{i_1=0}^{1} \cdots \sum_{i_{p-1}=0}^{1} \sum_{i_{p+1}=0}^{1} \cdots \sum_{i_n=0}^{1} x_{i_1,\cdots,i_{p-1},1,i_{p+1},\cdots,i_n j} \Big) \leq R_{i_p}, \forall p \in \{i_1 \cdots i_n\} \text{ and } j \in \{1, \cdots, m\} \tag{10}$$

$$\begin{aligned} &\sum_{i_2=0}^{1} \cdots \sum_{i_n=0}^{1} x_{1,i_2,\cdots,i_n j} + \cdots + \sum_{i_1=0}^{1} \cdots \sum_{i_{p-1}=0}^{1} \sum_{i_{p+1}=0}^{1} \cdots \sum_{i_n=0}^{1} x_{i_1,\cdots,i_{p-1},1,i_{p+1},\cdots,i_n j} + \\ &\cdots + \sum_{i_1=0}^{1} \cdots \sum_{i_{n-1}=0}^{1} x_{i_1,\cdots,i_{n-1},1 j} \geq K \qquad \forall j \in \{1, \cdots, m\} \end{aligned} \tag{11}$$

$$x_{i_1,\cdots,i_n} \in \{0, 1\} \tag{12}$$

In this model, $x_{i_1,\ldots,i_n j}$ is an n-dimensional decision variable where i_p indicates participation of sensor p to localize target j. If $x_{i_1,\ldots,i_n j}$ is 1, the allocation of a subset of n sensors get established for target j with all participating and non-participating sensors to locate target j. Accordingly, $c_{i_1,\cdots,i_n j}$ indicates the allocation cost similar to f_j function in Eq. (1). Eqs. (7) and (8) assure that every target is assigned to at least a set of sensors. Eq. (9) specifies that sensor p is allocated to no more then C_{i_p} targets. Eq. (10) assures that a sensor can be only assigned to a target when the target is within its monitoring range. Eq. (11) requires allocation of at least K sensors per target. Eq. (12) is used to define the scope of the n-dimensional variable $x_{i_1,\cdots,i_n}$.

To address the second challenge of infeasible solution, we consider an objective function which minimizes localization area along with a related penalty. This model captures a capacity restricted sensor network where the total number of assignments is no more than a predetermined budget B. When B is less than $|M| \times K$, this assures that Eq. (3) is not satisfied for at least one target. Under the assumption that each sensor requires one unit of energy to focus on one target, we consider B as the energy budget since it is proportional to the total use of energy by the sensors to engage in the process of localization. The imposed penalty per target is proportional to the lack of allocated sensor(s) with respect to the requirement (K). We relax Eq. (3) and modify the objective function in Eq. (1) by including a penalty for every target where less than K sensors are assigned, in either additive (Eq. 13) or multiplicative (Eq. 14) form. In Eq. 13, we note the introduction of a base factor ρ_1 multiplied by the number of missing sensors, where ρ_1 is sufficiently large. Also, for both Eq. 13 and Eq. (14), $f_j(x_{1j}, \cdots, x_{ij}, \cdots, x_{nj})$ is expected to represent a correspondingly larger area due to lack of adequate sensors, including the case where a target has no sensors assigned.

$$min \sum_{j=1}^{m} \Big[f_j(x_{1j}, \cdots, x_{ij}, \cdots, x_{nj}) + max\Big(0, \rho_1 \times (K - \sum_{i=1}^{n} x_{ij})\Big) \Big] \tag{13}$$

$$min \sum_{j=1}^{m} \Big[f_j(x_{1j}, \cdots, x_{ij}, \cdots, x_{nj}) \times max\Big(1, (K + 1 - \sum_{i=1}^{n} x_{ij})\Big) \Big] \tag{14}$$

Similar to the previous discussed models, the objective function is subjected to constraints in Eqs. (2), (4) and (5) along with an additional energy budget constraint presented in Eq. (15). The latter can also be modified to include it in the linear model in the same manner as Eq. (4) was modified into to Eq. (11).

$$\sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} \leq B \qquad\qquad B < |M| \times K \tag{15}$$

This model is used as a guide for the performance of our approach for the under-capacitated and energy budget restricted scenarios.

3. Related Work

Various assets may have self-localization abilities such as those belonging to the same organization (e.g. self-reporting using on-board GPS system) while others may require external sensors to be localized (e.g. assets belonging to other organizations). Localization is important for surveillance and rescue efforts [9] as well as supply chain management [10, 11]. Also, organizational supply chains are exposed to potential uncertainty and unexpected events [12] typically experienced in the real-world environment and which can only be approximately considered during planning and should be monitored during plan execution. Other important aspects in this setting are related to data fusion and situational awareness [13]. Different external sources of uncertainty challenges the performance of localization procedure which includes missing assets information, imperfect sensor information, stochastic phenomena, etc. In this context, there are specific applications such as sensor deployment [14], facility location [15, 16] and object localization [17].

The use of sensors for asset localization has been thoroughly studied in wireless sensor networks (WSNs). Target localization represents a key application [18] in the context of WSNs, which are composed of multiple nodes that are deployed randomly or strategically (vantage points, junction points, etc.) [19]. Each node is typically resource constrained [20] and able to detect targets within its range by sampling various types of signals such as electromagnetic radiation, light, sound, etc. [21]. The use of distributed sensors of various capabilities can provide sensor diversity which typically enhances target localization [22].

Event detection represents a key aspect in relation to WSNs applications [22]. In this context, sensor readings must match certain conditions in order to report an event. WSNs integrated with Radio Frequency Identification (RFID) have been studied for target-tracking purposes [23]. In this work, the targets are carrying active RFID tags which enables individual target identification. A radar based sensor network is proposed in [24] for detecting targets based on ultra-wideband radio impulse. Acoustic sensors have been studied in [25] for their potential to distinguish objects that have different sound signatures. In [26], the authors discuss target sensing by analyzing the intensity of acoustic signals. In this setting, sensors generate detection events when the intensity of the acoustic signal exceeds a predefined threshold. Vibration sensors are considered in [27] in order to distinguish different types of objects. In this work, seismic and passive infrared sensors are employed in order to detect and classify humans, animals and light vehicles.

An interesting architecture aimed at tracking assets within construction sites is discussed in [28]. The employed components are cost effective and involve Radio Frequency IDentification (RFID), Bluetooth Low Energy (BLE) tags in conjunction with Android mobile devices. The core functionality is provided by a pair of applications, implementing asset searching and tracking. The key benefits of the proposed architecture reside in the ability to maximize mobile device battery lifetime, allowing an operating time enough for an entire work shift while providing good localization accuracy. In addition, the architecture allows for a trade-off between energy consumption and localization accuracy and the effectiveness of the proposed solution is demonstrated via extensive simulation. Moreover, in [29], the authors propose an efficient geographic asset tracking solution that allows conserving mobile resources such as energy used for communication by dynamically adapting the employed tracking scheme via context-aware personalized route learning. This involves distributed proactive monitoring of context information regarding asset's properties from its routes characteristics collected under different environmental conditions. The proposed approach leverages an adaptive learning scheme allowing for an optimized evaluation of data transmission with reduced overhead based on personalized tracking algorithm for mobile assets.

The problem of cooperative localization in mobile networks using non-parametric variants of belief propagation is addressed in [30] in the context of mobile networks. The paper discusses the main issues specific to non-parametric belief propagation (NPB) algorithms used for this type of problem, such as the high communication cost and the sampling techniques challenges. Also, in [9], environment surveillance using autonomous UAVs is discussed in the context of search and rescue. The paper considers key parameters that need to be taken into account including data quality, energy limitations, environmental exogenous events and the level of information exchange and coordination between different UAVs. A multi-agent system involving negotiating agents pursuing resources allocation activities is presented in [31]. This work is addressing the issues of constraint satisfaction in multi-sensor target tracking. The agents aim to optimize the use of their own consumable resources while pursuing the global goal of multi-sensor target tracking.

In order to use WSNs effectively, there are specific challenges that are arising [5]. The challenges are typically related to the limitations of individual sensors since in general, an individual sensor is not capable of estimating the location of a target. In this context, a typical challenge relates to the need to have multiple sensors per target in order to take into account sensor measurement features (e.g. omnidirectional sensing) and to mitigate effects such as noise. Thus, the key importance stays on analyzing which sensors need to be assigned to which targets and how the measurements should be combined to get an accurate estimate.

In [5], the authors consider the focus of attention problem (FOA) over a sensor network consisting of small, inexpensive and simple sensors that are used for estimating target locations. The sensors are arranged on a straight line while the targets are situated somewhere in the plane. The objective is to assign sensors to targets such that the overall expected error of the target locations is minimized. In this setting, the paper considers the special case where every target is tracked by a pair of sensors, typically representing cameras. The work presented in [32] addresses a similar problem but goes further with an extension considering the case where multiple range sensors can be arranged on a circle for monitoring a particular region.

Specific assignment problems arising from multiple target tracking are discussed in [4] in relation to data association and sensor set partitioning. Moreover, issues relating to sensor coverage are discussed in [33] where a single sensor is considered as sufficient to cover a target point but the coverage quality decreases with distance. In this context, the aforementioned work is aiming at ensuring optimal coverage over sensor network movements. Moreover, in relation to assignment problems, different variations are discussed in the literature [34, 35]. Also, in [6], the authors discuss a variant of the multidimensional assignment problem in the context of multi-sensor multi-target tracking problems, where different sensor measurements obtained from a sensor network must be matched to different targets. The Multidimensional Assignment Problem [36], also termed as the axial Multi Index Assignment Problem [3] represents a well-known optimization problem. Vehicle-target assignment is studied in [37] using game theory where a group of vehicles aim to optimally assign themselves to a given set of targets. The paper presents simulations illustrating vehicle negotiations that can lead to near-optimal assignments. A branch and bound algorithm for the multidimensional assignment problem is presented in [38] along with local search improvements. Moreover, a decomposition scheme is proposed in [39] for partitioning the multidimensional assignment problem in a manner that allows for subsequent recombination which can provide upper and lower bounds for the original problem. A memetic algorithm [40] combining a genetic algorithm with a form of local serach is proposed in [41] for the multidimensional assignment problem. In this setting, the forenamed work is employing an adjustable population size to improve the efficiency of the local search procedure.

4. Proposed Approach

In this section, we describe the proposed solution approach with main emphasis on the localization error reduction. Our aim is to elaborate advanced capabilities for an advisory asset localization system that tactical decision makers can use in the context of logistics and supply chain management.

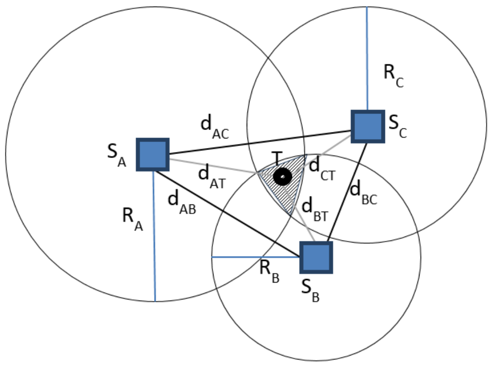


Fig. 1: Target T inside the triangle delimited by $\{S_A, S_B, S_C\}$ and marked area for possible error in target identification.

In the absence of information exchange with the targets, asset localization can be carried out using sensor data in relation to specific target signatures. In this respect, one may assume that the available sensors are able to discriminate a target signature, based for example on generated heat, reflected electromagnetic waves, sound, vibration, etc. Each sensor having a specific location and distance from a target within its detecting range can participate along with other sensors, that can detect the same target, in localizing the target with certain error corresponding to an intersection area. The assignment of at least K (typically 3) sensors to a target, establishes a possible area [8] where the target can be situated as seen in Fig. 1.

Since the problem of multi-sensor to multi-target assignment exhibits combinatorial explosion, we propose a heuristic solution generation approach whereby near-optimal solutions can be obtained in a computationally cost-effective manner. The heuristic solution search involves the exploration of various sensor-target assignments in order to identify progressively better solutions. This requires the means to compare solutions that may fail to satisfy certain constraints such as assigning at least K sensors to each target or respecting the energy budget restriction.

In order to address these situations, we allow assigning of less than K sensors to a target as well as exceeding the energy budget restriction but impose a composed penalty. Thus, we relax Eqs. (4) and (15) while modifying the objective function in Eq. (1) by including a target specific penalty whenever less than K sensors are assigned along with an overall penalty on the whole solution in the case where the energy budget restriction is exceeded. The overall penalty increases the solution cost by taking into account the increase over the energy budget restriction. According to the penalty type we have two variants corresponding to additive (Eq. 16) or multiplicative (Eq. 17) form. In Eq. (16), we note the addition of a supplementary base factor ρ_2 multiplied with the value exceeding the energy budget restriction, where ρ_2 is sufficiently large.

$$min \sum_{j=1}^{m} \Big[f_j(x_{1j}, \cdots, x_{ij}, \cdots, x_{nj}) + max\Big(0, \rho_1 \times (K - \sum_{i=1}^{n} x_{ij})\Big) \Big] + max\Big(0, \rho_2 \times (\sum_{j=1}^{m} \sum_{i=1}^{n} x_{ij} - B)\Big) \tag{16}$$

$$min \sum_{j=1}^{m} \Big[f_j(x_{1j}, \cdots, x_{ij}, \cdots, x_{nj}) \times max\Big(1, (K + 1 - \sum_{i=1}^{n} x_{ij})\Big) \Big] \times max\Big(1, \frac{\sum_{j=1}^{m} \sum_{i=1}^{n} x_{ij}}{B}\Big) \tag{17}$$

We describe in the following the key aspects of the proposed approach. In the context of solving challenging optimization problems, heuristic techniques are typically employed to obtain near-optimal solutions. The general idea of the heuristic solution generation technique is conveyed by Algorithm 1 along with a subsequent discussion. Our proposed approach is inspired by the learning based evolutionary concept described in [42] which employs an evolving population of solutions in the context of partitioning a set of customers among the depots of a supply chain in order to minimize overall routing cost. However, the scope of our problem is notably more challenging since it involves the underlying problem of multi-dimensional assignment with difficult constraints. These involve the assignment of multiple sensors to multiple targets taking into account that each sensor can be assigned to any number of targets in its range, according to its capacity. Thus, in our case, we iteratively spawn successive generations of solutions, produced using a pseudo random number choice generator that allows to obtain many multisensor-multitarget assignment combinations according to the available sensor assignment choices (initially derived from the sensing ranges and capacities of the sensors). Then, the cost and assignment combinations of the solutions generated in a given iteration allows to learn the sensor assignment choices that are cost-wise ineffective in order to remove one or more at every iteration. At the limit, the total number of available assignment choices would steadily decrease until each sensor would be left with only one assignment choice, corresponding to the final solution.

In the context of the algorithm, the population size of each generation represents an input parameter and each individual multisensor-multitarget combination in a generation has a corresponding cost which is used to rank the individual combinations in each generation. From all individuals in a generation, only an elite number is retained where the number of elite solution represents another input parameter. The current best solution is updated in each iteration whenever a better solution is identified in the current iteration elite solutions. Moreover, the elite solutions are analyzed based on a voting scheme over the sensor-target assignment combination in order to rank the cost-effectiveness of the underlying target assignment combinations across the elite solutions. Subsequently, from the combinations with the least potential to participate in good quality (lower cost) solutions, a certain number (another input parameter) is selected to be removed from the possible assignment choices of their respective sensors before spawning the next generation. The removal of certain sensor-target assignment combination choices from various sensors after each iteration, reduces the solution search space and allows the procedure to converge to a near-optimal solution over a number (stop condition parameter) of successive iterations or until the remaining combinations are less than or equal to the population size parameter value. In this case, an exhaustive search provides the most cost effective solution in the remaining solution search space. Next, we detail the key points of Algorithm 1 from a higher level of abstraction in order to convey more effectively the underlying concept. The search procedure is initialized in line 1 with the following input parameters:

- max_iteration(*maxIter*): maximum number of generations that the procedure can spawn over successive iterations (stop condition parameter).
- remove_count(*rCnt*): number of assignment combinations choices that can be removed per iteration.

- sample_count(*sCnt*): population size to be sampled from the solution search space at each iteration.
- elite_count(*eCnt*): elite solution count to be retained from the sampled population over each iteration.
- *seed*: value to initialize the random number choice generator used for solution sampling.

At line 2, we can see that the sensor set (*sSet*) and the target set (*tSet*) are considered as global knowledge. Furthermore, at line 3, the current best solution (*crtBestSol*) is set initially empty while the random number generator (*rndGen*) is initialized with the *seed* value.

The procedure continues at line 4 with a while loop testing whether the maximum number of iterations has been reached. In the while loop, the procedure evaluates at line 5 whether the maximum number of

Algorithm 1 Heuristic Search (HS) for near-optimal multisensor-multitarget assignment

```
1: Input: max_iteration(maxIter), remove_count(rCnt), sample_count(sCnt), elite_count(eCnt), seed;
2: Global Knowledge: SensorSet(sSet), TargetSet(tSet);
3: Initially: crtBestSol ← {}, rndGen ← PseudoRandomGenerator(seed);
4: while (maxIter > 0) do
5:   if GetMaxCombinationCount(sSet, tSet) ≤ sCnt then
6:     allSensorTargetCombinations ← GenerateAllSensorTargetCombinations(sSet, tSet);
7:     for each sensorTargetCombination in allSensorTargetCombinations do
8:       if Cost(sensorTargetCombination) < Cost(crtBestSol) or crtBestSol={ } then
9:         crtBestSol ← { Cost(sensorTargetCombination), sensorTargetCombination };
10:      end if
11:    end for
12:    break;
13:  else
14:    sortedSearchMap = {};
15:    for i=0 to sCnt do
16:      exploreSol ← GeneratePseudoRandomSensorTargetCombination(sSet, tSet, rndGen);
17:      if SIZE(sortedSearchMap) < eCnt or Cost(exploreSol) < LastKey(sortedSearchMap) then
18:        Insert {COST(exploreSol), exploreSol} into sortedSearchMap;
19:        if SIZE(sortedSearchMap) > eCnt then
20:          remove LastEntry(sortedSearchMap) from sortedSearchMap
21:        end if
22:      end if
23:    end for
24:  end if
25:  if crtBestSol = {} or Cost(crtBestSol) > FirstKey(sortedSearchMap) then
26:    crtBestSol ← FirstEntry(sortedSearchMap);
27:  end if
28:  sensorCombinationVoteMap ← CountAndRankPerSensorCombinationVotes(sortedSearchMap);
29:  srCnt ← rCnt;
30:  for each {sensor, combination} in sensorCombinationVoteMap do
31:    if srCnt = 0 then
32:      break;
33:    else
34:      if crtBestSol does not contain combination and CombinationChoiceCount(sensor) > 1 then
35:        remove combination choice from sensor
36:        srCnt = srCnt - 1;
37:      end if
38:    end if
39:  end for
40:  maxIter = maxIter − 1;
41: end while
42: return crtBestSol;
```

combinations for the sensor set and target set is less than *sCnt*. In this case, an exhaustive search is used to identify the final solution before terminating the procedure (lines 6 to 12). Otherwise, at line 14, a cost-wise sorted (in ascending manner) solution search map (*sortedSearchMap*) is set to empty and a for loop over the sample count is used to explore pseudo randomly generated solutions, aggregating the elite qualifying ones in the *sortedSearchMap* (lines 15 to 23). Then, (*crtBestSol*) is updated if needed (lines 25 to 27). At line 28, a sorted sensor combination vote map (*sensorCombinationVoteMap*) is used to hold the count and rank corresponding to the per sensor combination votes obtained by analyzing the elite solutions stored in the *sortedSearchMap*. The *sensorCombinationVoteMap* is sorted in a manner that allows iterating over its elements from the least likely combinations to participate in competitive elite solutions to the most likely. Thus, a number of sensor assignment combination choices equal to *rCnt* is removed from their respective sensors by iterating over the *sensorCombinationVoteMap* (lines 29 to 39). The value of *maxIter* is decremented at line 40 and after completing the while loop, the procedure returns *crtBestSol* (at line 42).

5. Case Study

We present next an application of the proposed approach in the context of an illustrative case study problem depicted in Fig. 2 (left). The problem involves a sensor network comprising 7 sensors (`a, b, c, d, e, f, g`) that have to localize 4 targets (`P, Q, R, S`) in their coverage area. The details of the problem are shown in Table 1 which provides for each sensor the corresponding range, target focusing capacity and the targets in its range. We note that there are multiple possible assignments for each sensor ranging from 2 (sensor `c`), 3 (sensors `a, b` and `e`) and 6 (sensors `d, f` and `g`) yielding a total of 11664 possible assignments. Fig. 2 (middle) presents the solution obtained with a nearest neighbour sensor assignment while Fig. 2 (right) presents the optimal solution.

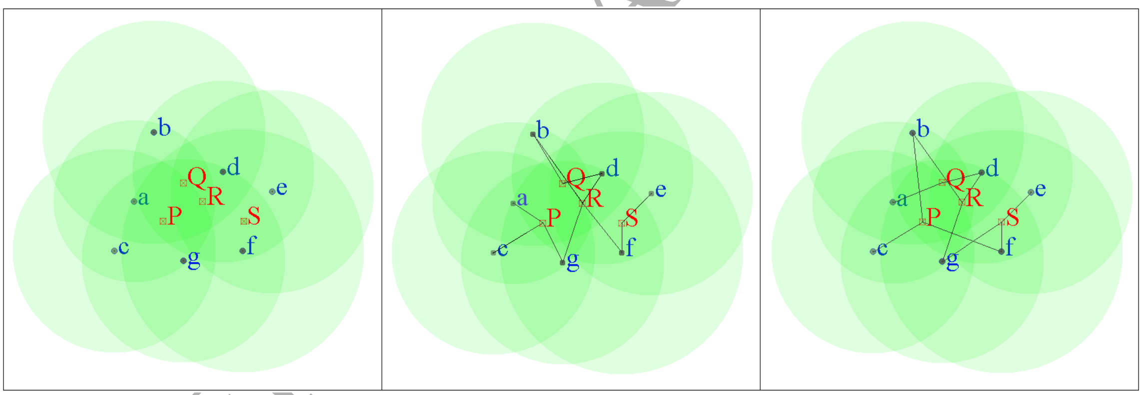
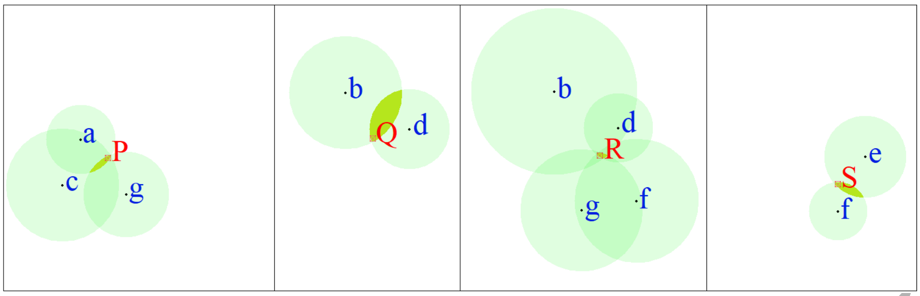


Fig. 2: Case study problem (left), nearest neighbor solution cost=2742.1184 (middle), optimal solution cost=392.1358 (right)

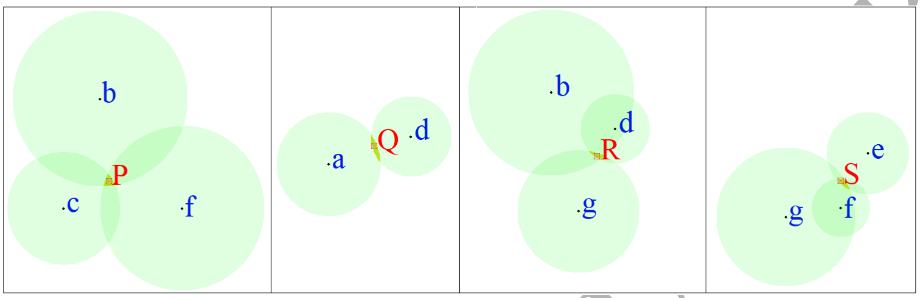
Sensor	a	b	c	d	e	f	g
Range	80	110	100	90	100	120	100
Target focusing capacity	1	2	1	2	1	2	2
Targets in range	P, Q, R	P, Q, R	P, Q	P, Q, R, S	Q, R, S	P, Q, R, S	P, Q, R, S
Target assignment combinations	3	3	2	6	3	6	6

Table 1: Case Study Problem Details

Given that we aim for at least three sensors assigned for each target, the best solution will include as much as possible corresponding sensor assignments. However, the case study problem data is chosen such that any valid solution will have at least one target with an assignment of less than 3 sensors. This allows to highlight that in the general case, there is no advantage to prune the combinations where less than 3 sensors are assigned per target since such assignment can be encountered even in the optimal solution. In



(a) { cost(P)=144.2384 + cost(Q)=2128.9001 + cost(R)=42.9048 + cost(S)=426.0751 } = 2742.1184



(b) { cost(P)=75.4355 + cost(Q)=205.1466 + cost(R)=58.7093 + cost(S)=52.8443 } = 392.1358

Fig. 3: Breakdown cost comparison between the nearest neighbor solution (a) and the optimal solution (b)

this context, a target related penalty is applied when only one or two sensors are assigned for a target. In the absence of such penalty, certain targets can have comparably large localization error cost while others can have notably smaller localization error cost (the overall localization error cost may be reduced but some targets will be localized very poorly). The lowest penalty is applied when only two sensors are assigned for a target and a higher (double) penalty is applied when only one sensor is assigned for a particular target since such assignment yields higher localization error cost. Thus, the penalty varies corresponding to the number of missing sensors and can be additive for larger problems or multiplicative for smaller problems such as the one presented in the case study.

In Fig. 3, we contrast the nearest neighbor solution obtained by assigning the sensors to the closest targets in their range against the heuristically obtained optimal solution (confirmed via exhaustive search) and provide the cost breakdown comparison per target. The sensor assignment for each target corresponding to the nearest neighbor solution is: `P:(a, c, g)`, `Q:(b, d)`, `R:(b, d, g, f)`, `S:(e,f)` while the assignment for the optimal solution is: `P:(b, c, f)`, `Q:(a, d)`, `R:(b, d, g)`, `S:(e, f, g)`. We note that with the exception of target `R` which has a slightly lower cost in the nearest neighbor solution, all other targets exhibit notably smaller cost in the optimal solution compared to the nearest neighbor solution. The latter has a lower cost for target `R:(b, d, g, f)` assigning it 4 sensors while assigning only two sensors to targets `Q:(b, d)` and `S:(e, f)` thereby incurring notably higher total penalty compared to the optimal solution which has only one assignment with penalty for sensor `Q:(a, d)`. We note that even if we disregard the penalty, the cost for target `Q` (205.1466/2=102.5733) is significantly lower in the optimal solution compared to cost for target `Q` (2128.9001/2=1064.4500) in the nearest neighbor solution.

We illustrate next the heuristic solution generation for a sample count of 40 and an elite set size of maximum 10 solutions. In Table 2, we can see the breakdown of the received votes per sensor, ordered in increasing order of the received votes. This ordering allows to identify the combinations with the least potential to participate in competitive solutions. For the cases where the same number of minimum votes are received by different sensor combinations, the difference between the maximum and the minimum votes (of the same sensor combinations) is used as sorting criteria such that the sensors with a larger difference are considered before the ones with a smaller difference. This allows to segregate among different combinations receiving the least amount of votes. Thus, a larger difference (for the combinations of a particular sensor) corresponds to a higher confidence that the combination receiving the minimum votes (for that sensor) are less likely to be part of competitive solutions compared to other sensor combinations (receiving the same amount of minimum votes) for which the difference has lower values. In the case where two sensors with the same number of minimum votes have also the same difference between maximum and minimum votes received by their respective combinations, then the corresponding combinations receiving minimum votes have similarly low potential of participating in competitive solutions. For instance, the first and second columns of Table 2 show that sensors `d` and `g` have combinations receiving a minimum number of votes equals to 0 ([P,Q] and [P,S] for `d`, respectively [Q,R] and [R,S] for `g`). However, the maximum number of votes received by `d` is 5 for combination [Q,S] and respectively 4 for sensor `g` combination [P,S]. Since the difference between maximum and minimum number of votes is higher for `d`, the latter appears before `g`.

The combinations belonging to each sensor are also sorted in decreasing order of their received votes as shown in Table 2. For the cases where some combinations belonging to the same sensor receive the same amount of votes, the average cost of the participating solutions is taken as supplementary sorting criteria. This favors the combinations that are cost-wise beneficial across multiple solutions. For example, according to Table 2, sensor `b` receives 4 votes for combination [P,R] (with average cost of participating solutions of 3261.83) and 3 votes for [P,Q] and [Q,R] (with average cost of participating solutions of 2837.9312 and 6792.675 respectively). In this example, [P,R] appears first since it has more votes, while [P,Q] appears before [Q,R] since it has lower cost for the same votes.

The potential candidate combinations to be pruned, can be identified from Table 2, before the next sampling step. The first candidate is `[P,S]` for sensor `d` (marked in bold face). The second candidate is `[P,Q]` also for sensor `d`. The third candidate is `[R,S]` for sensor `g`. The fourth candidate is `[Q,R]` also for sensor `g`. The fifth candidate is `[P,R]` for sensor `d` rather than `[P,R]` for sensor `f` since the difference between the maximum and minimum votes for sensor `d` is greater then the one for sensor `f`. Further candidates can similarly be identified. Table 3 provides the top 10 elite solutions obtained in the first iteration along with the most amount of votes (5) provided by 5 of the solutions (with average cost of $\frac{636.9903+804.86487+1151.9899+5126.551+6206.9}{5} = 2785.4592$) for target combination `[Q,S]` of sensor `d` (targets and sensor marked in bold face). In contrast, the least amount of votes (0) have been received in the first iteration by target combinations `[P,Q]` and `[P,S]` for sensor `d` and `[Q,R]` and `[R,S]` for sensor `g` since these combinations do not appear across all of the elite solutions, thus receiving no votes. Once the combinations with least potential to be part of competitive solutions have been identified, we can prune at the end of each iteration one or more of the identified combinations from their respective sensors. This allows to progressively decrease the search space in each iteration. However, limited sampling implies that any combination considered to be pruned has a small potential to lead the solution search to a local minimum. Consequently, the more combinations are pruned per iteration, the more likely is to arrive at a local minimum solution. Thus, pruning only one combination per iteration carries the least chance of arriving at a local minimum solution. In turn, this will involve more iterations and a correspondingly larger cumulative sample amount. In this setting, one can consider pruning multiple combinations per iteration as a trade-off between lower solution search time (less number of iterations) and solution quality (gap to the optimal value).

We illustrate the heuristic solution convergence profiles (Fig. 4) and the corresponding pruned combinations over successive iterations (Table 4) for increasingly higher values of the sample count (20, 30 and 40)

Sensors: d	g	f	a	b	e	c
[Q,S]/5/2785.4592	[P,S]/4/5112.416	[P,S]/3/4941.793	[R]/4/2416.4458	[P,R]/4/3261.83	[S]/4/5006.1055	[Q]/6/4231.99
[Q,R]/2/6123.157	[P,R]/2/3387.3562	[Q,S]/2/1188.0356	[P]/4/6627.453	[P,Q]/3/2837.9312	[R]/3/2526.526	[P]/4/4136.799
[R,S]/2/7097.16	[P,Q]/2/3505.8823	[R,S]/2/3644.856	[Q]/2/2881.7705	[Q,R]/3/6792.675	[Q]/3/4778.38	
[P,R]/1/1571.2064	[Q,S]/2/3851.498	[Q,R]/1/5126.551				
[P,Q]/0/0.0	[Q,R]/0/0.0	[P,Q]/1/6114.5254				
[P,S]/0/0.0	[R,S]/0/0.0	[P,R]/1/6206.9				
min votes: 0	0	1	2	3	3	4

Table 2: Sensor-target assignment combinations: column-wise ordered by min. received votes (primary) and by the difference between max. and min. votes (secondary); row-wise ordered by vote count (primary) and by participating average solution cost (secondary)

Elite solution cost	Sensor assignment			
636.9903	P=[b; f; g]	**Q**=[a; c; **d**]	R=[b; e; g]	**S**=[**d**; f]
804.86487	P=[b; c; g]	**Q**=[b; **d**; f; g]	R=[a; e]	**S**=[**d**; f]
1151.9899	P=[b; g]	**Q**=[c; **d**; e]	R=[a; b; f]	**S**=[**d**; f; g]
1571.2064	P=[b; c; d]	Q=[b; f; g]	R=[a; d]	S=[e; f; g]
5126.551	P=[b; g]	**Q**=[a; c; **d**; e; f]	R=[b; f]	**S**=[**d**; g]
6114.5254	P=[a; c; f; g]	Q=[b; d; f]	R=[b; d]	S=[e; g]
6131.7896	P=[a; b; f]	Q=[c; d; g]	R=[b; d]	S=[e; f; g]
6137.722	P=[b; g]	Q=[b; c]	R=[a; d; e; f; g]	S=[d; f]
6206.9	P=[a; f; g]	**Q**=[b; c; **d**; g]	R=[b; f]	**S**=[**d**; e]
8056.5986	P=[a; c; f; g]	Q=[b; e]	R=[b; d]	S=[d; f; g]

Table 3: Elite solutions for the first iteration, with most amount of votes (5) provided by 5 solutions for combination `[Q,S]` of sensor `d`

Sample count/	20			30			40		
Iteration	Sensor	Remaining/Pruned combinations	Min. votes	Sensor	Remaining/Pruned combinations	Min. votes	Sensor	Remaining/Pruned combinations	Min. votes
1	d	5/[P, S]	0	d	5/[P, S]	0	d	5/[P, S]	0
2	d	4/[P, Q]	0	b	2/[Q, R]	0	b	2/[Q, R]	0
3	f	5/[P, R]	0	d	4/[P, Q]	0	g	5/[Q, S]	0
4	f	4/[Q, R]	0	f	5/[P, R]	0	f	5/[Q, R]	0
5	g	5/[P, S]	0	d	3/[Q, R]	0	a	2/[R]	0
6	g	4/[P, R]	0	a	2/[P]	1	d	4/[P, Q]	0
7	a	2/[Q]	0	f	4/[P, Q]	0	f	4/[P, R]	0
8	d	3/[P, R]	1	g	5/[Q, R]	0	g	4/[Q, R]	0
9	d	2/[R, S]	1	g	4/[Q, S]	0	f	3/[P, Q]	0
10	f	3/[Q, S]	1	d	2/[P, R]	1	d	3/[P, R]	1
11	f	2/[P, Q]	0	f	3/[Q, R]	1	g	3/[P, Q]	0
12	e	2/[R]	1	f	2/[Q, S]	1	e	2/[Q]	1
13	g	3/[Q, S]	0	e	2/[S]	1	d	2/[R, S]	1
14	a	1/[R]	2	g	3/[P, S]	0	e	1/[R]	2
15	b	2/[P, Q]	1	g	2/[R, S]	0	d	1/[Q, S]	1
16	g	2/[Q, R]	2	f	1/[P, S]	1	g	2/[P, S]	1
17	c	1/[P]	1	e	1/[R]	2	f	2/[Q, S]	2
18	e	1/[Q]	3	d	1/[Q, S]	3	N/A	N/A	N/A
Average min. votes per pruned comb.			0.72			0.61			0.53

Table 4: Pruned combinations over successive iterations

per iteration while pruning only one sensor combination in each iteration. We note that since only one candidate combination is pruned per sampling iteration, the successively pruned candidate combinations, for the sample count of 40, differ from the evaluated candidate combinations for the first iteration. This relates to the potential identification of more or less costly best solutions in successive sampling iterations, each with their respective elite solution groups and respective sensor assignments, based on which pruning candidates are determined. However, we can observe that for the sample count of 40, the successively pruned combinations, while determined over different elite group solutions, have no votes for the first 9 iterations and at most 1 or 2 votes for the subsequent iterations. In Table 4, we can also note that the average minimum votes per pruned combination is decreasing when the sample count is increasing. This reflects that the pruned combinations are segregated with better confidence when the sample count is larger. Moreover, in Fig. 4, we can see for each sample count value, the best solution values (rounded to the first decimal) obtained over successive iterations along with their respective trend-lines. We can note that for a sample count of 20, there are quite notable differences between the encountered solutions, with a maximum at 3301.4 in iteration 5 and minimum at 509.4 in iterations 15, 16, 18 and 19. Then, for a sample count of 30, there are still somewhat notable differences with a maximum at 1670.9 in iteration 8 and minimum at 413.1 in iterations 15, 17, 18 and 19. Finally, for a sample count of 40, there are comparatively less notable differences with a maximum at 956.9 in iteration 6 and minimum at 392.1 in iteration 18. We can observe that for a sample count of 20 and 30, the solution search ends up in a local minimum at 509.4 and respectively at 413.1 after 19 iterations. In contrast, for a sample count of 40, the solution search ends up in the optimal value (rounded to first decimal) at 392.1 after 18 iterations.

We also note that for each sample count value, in the last iteration, the solution search space typically contains less elements than the sample count such that an exhaustive search can be performed at this stage with the same or less computational cost than sampling but with the guarantee of providing the best solution

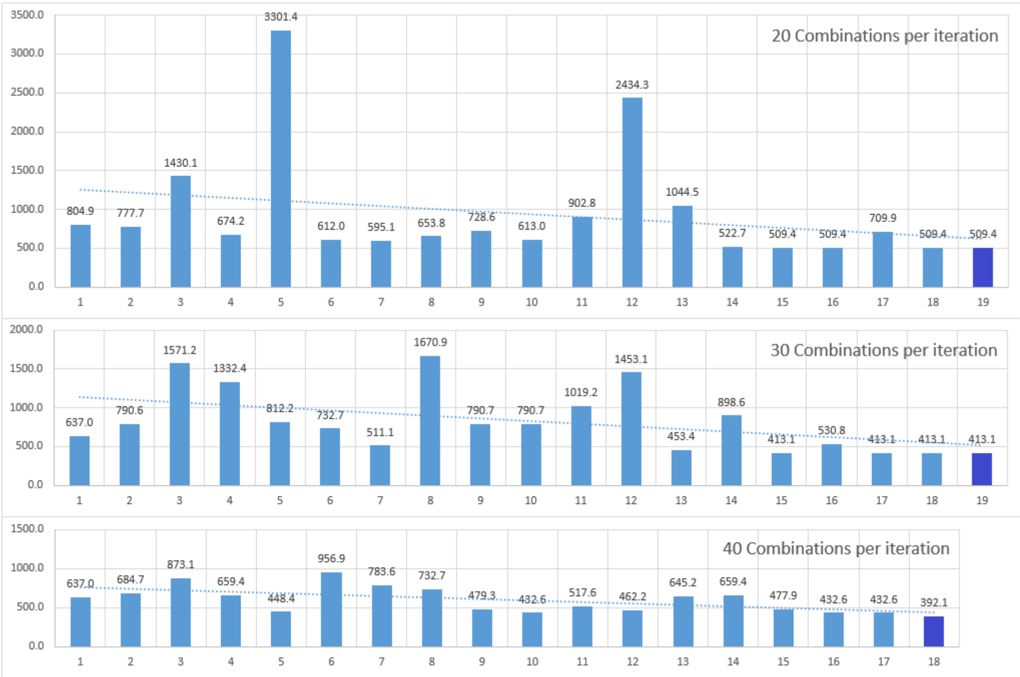


Fig. 4: Heuristic Solution Convergence Profile for increasing number of sampled combinations per iteration

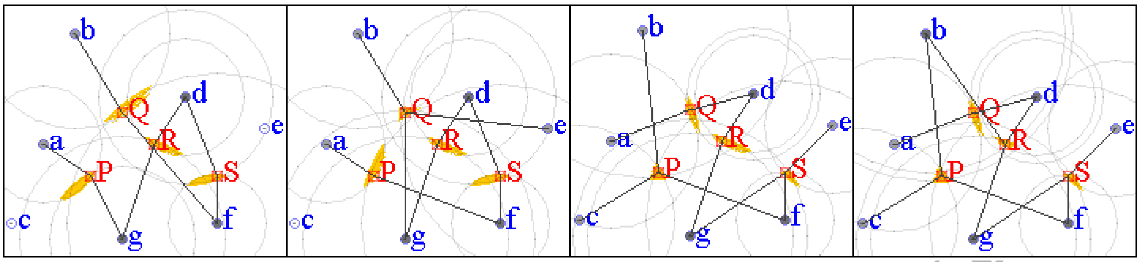
in the remaining search space. This is the reason for having 18 iterations for a sample count of 40 (rather than 19 as in the other two cases) since the remaining combination count in iteration 18 is less than 40.

We discuss next the same case study problem in the context of using an available energy budget to limit the energy use of the sensor network. We consider without loss of generality that each sensor requires an amount of one energy unit to focus on a particular target. Moreover, we consider that sensors with the capacity of focusing on more than one target, require the corresponding amount of energy units matching their capacities. However, the availability of a limited energy budget may restrict some sensors from using all of their available target focusing capacity or even prevent some sensors from focusing on any target.

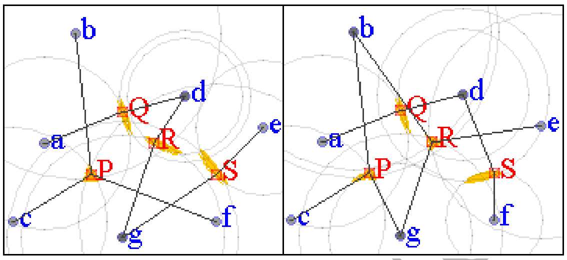
In order to better illustrate the effect of an energy budget in the context of the case study problem, we reconsider the problem in a setting where less precise target localization is allowed as trade off for energy saving. Consequently, we allow for a minimum of 2 sensors per target as acceptable instead of 3. While this results in extra network capacity compared to the case where a minimum of 3 sensors are required per target, the solution search space of the problem actually increases since each sensor can partially use its capacity, thus increasing the number of possible sensor to target assignment combinations.

Having the requirement of minimum 2 sensors per target and a total of 4 targets corresponds to a minimum energy budget of 8 energy units. Furthermore, the summed target focusing capacities of the 7 sensors is given by `a(1) + b(2) + c(1) + d(2) + e(1) + f(2) + g(2)`, resulting in a maximum energy budget of 11 units. Thus, we solve the problem for an energy budget ranging from 8 to 11 units of energy.

Fig. 5(a) depicts the solutions obtained for increasing energy budget values (8,9,10,11) when using a sample count of 40 (initially chosen given that solution search space is larger than before as previously mentioned). However, only for energy budget values of 8 and 11, the obtained solutions have been confirmed as optimal after exhaustive search. Increasing the sample count to 50, allowed the generation of better solutions for energy budget values 9 and 10 presented in Fig. 5(b), solutions confirmed as optimal after exhaustive search. We can observe in each solution presented in Fig. 5 the corresponding sensor-target assignment (lines connecting sensors to targets) and the related localization cost for each target (area delimited by the range circle intersection of the assigned sensors). We note as expected that increasing the energy budget leads to more cost effective solutions. We also note in Fig. 5(a) that for the smallest energy budget value of 8, two sensors (c and e) are not used while for energy budget value of 9, only one sensor (c) is not used.

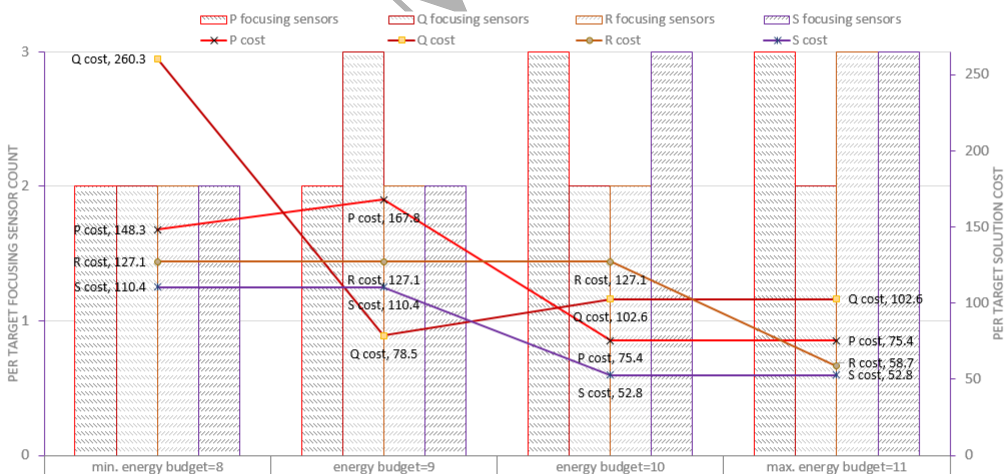


(a) SC=40: { eb=8,cost=646.1188; eb=9,cost=483.7962; eb=10,cost=357.9978; eb=11,cost=289.5625 }



(b) SC=50: { eb=9,cost=459.4762; eb=10,cost=352.2106 }

Fig. 5: Cost comparison for increasing energy budget (eb) between the solutions obtained with a sampling count (SC) of 40 (a) and the more cost effective solutions obtained for SC=50 (b).



	min. energy budget=8	energy budget=9	energy budget=10	max. energy budget=11
P focusing sensors	2	2	3	3
Q focusing sensors	2	3	2	2
R focusing sensors	2	2	2	3
S focusing sensors	2	2	3	3
P cost	148.3	167.8	75.4	75.4
Q cost	260.3	78.5	102.6	102.6
R cost	127.1	127.1	127.1	58.7
S cost	110.4	110.4	52.8	52.8

Fig. 6: Solution characteristics for increasing energy budget and sample count=40 relative to 'per target focusing sensor count' (primary axis) and 'per target solution cost' (secondary axis).

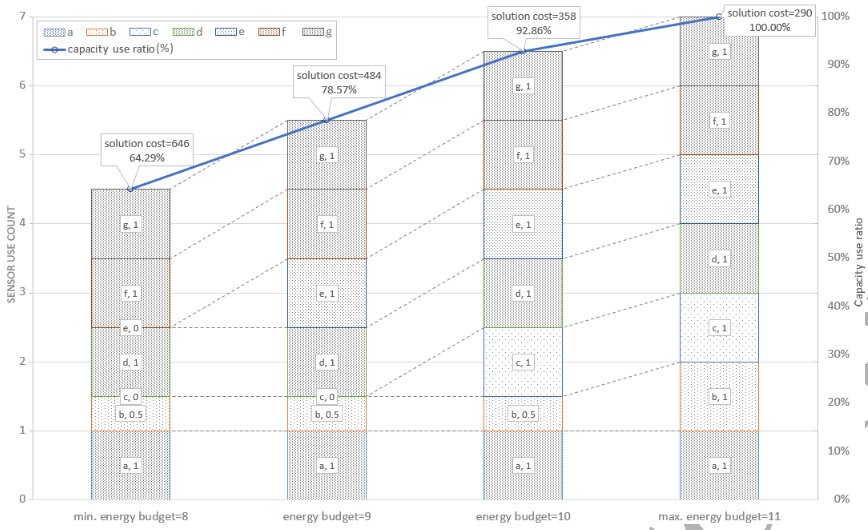


Fig. 7: Solution characteristics for increasing energy budget and sample count=40 relative to 'sensor use count' (primary axis) and 'network capacity use ratio' (secondary axis).

Fig. 6 depicts the characteristics of the obtained solutions (using sample count of 40), for increased energy budget in terms of the count of focusing sensors per target (target assignment count) as well as the solution cost breakdown per target. We can note that the cost for each particular target does not always decrease with increased energy budget such as in the case of target `Q` which has the highest cost for energy budget of 8 and the lowest cost for energy budget of 9. Also, target `P` has the highest cost for energy budget of 9 and the lowest cost for energy budget values of 10 and 11. Moreover, we note that increased values of energy budget allows the assignment of 3 sensors for an increasing number of targets, thereby lowering the cost. We can also observe that for energy budget of 11 (which allows obtaining the best solution), the assignment of sensors to targets is the same as in the best solution for the original case study problem even though the latter has a higher cost since it requires 3 sensors per target, thus incurring a penalty for one target, namely `Q`.

Fig. 7 depicts for increased energy budget the characteristics of the obtained solutions (using sample count of 40) in terms of the sensor capacity usage as well as the sensor network capacity use ratio (the corresponding solution cost is also shown as integer value for brevity). The smallest capacity use ratio is achieved for the minimum energy budget of 8 where sensor `b` is used at 50% capacity while sensors `c` and `e` are used at 0% capacity (not used). This corresponds to an overall sensor network capacity use of 64.29% given by `(a(1)+b(0.5)+c(0)+d(1)+e(0)+f(1)+g(1))/7 = 4.5/7` (4.5 sensors are used out of 7). The localization cost however is the highest (646). Bringing the energy budget value to 9 allows sensor `e` to be used at 100% capacity which results in a sensor network capacity use of 78.57% and an improved localization cost (484). Increasing the energy budget to 10 also allows sensor `c` to be used at 100% capacity resulting in 92.86% network capacity use and better localization cost (358). Finally, for the maximum energy budget, all sensors are used at 100% capacity with the best localization cost (290).

In Fig. 8, we can see a comparison between the solutions obtained for increasing energy budget with sample count of 40 and respectively 50, relative to the sensor network capacity use and localization solution cost. We note that for the energy budget of 9, the contrast of the obtained solutions is the most pronounced as follows. For sample count of 40, the solution cost is 484 and the network capacity use is 78.57% given by `(a(1)+b(0.5)+c(0)+d(1)+e(1)+f(1)+g(1))/7 = 5.5/7`. In contrast, the solution obtained with a sample count of 50 provides a better solution cost of 459 and a network capacity use given by `(a(1)+b(0.5)+c(1)+d(1)+e(1)+f(0.5)+g(1))/7 = 6/7` (85.71%). This results from bringing sensor `c` to 100% use while reducing the use of sensor `f` to 50% capacity. The solution with cost of 484 spends the

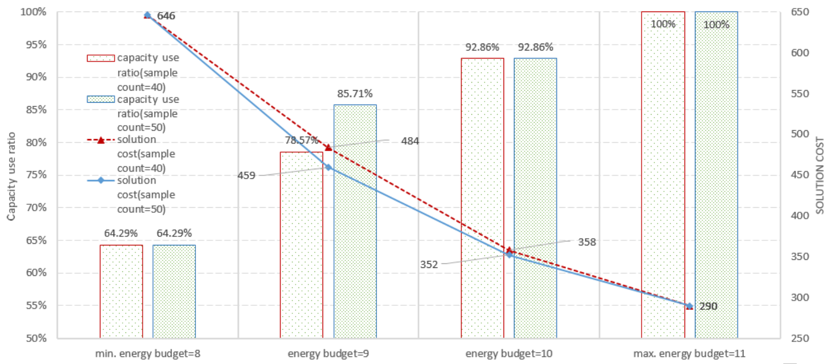


Fig. 8: Comparison for increasing energy budget between the solutions obtained with sample count=40 and the solutions obtained with sample count=50 with respect to 'network capacity use ratio' (primary axis) and 'solution cost' (secondary axis).

energy budget by focusing 2 sensors on targets `P,R,S` and 3 sensors on target `Q` whereas the solution with cost 459 spends the energy budget by focusing 2 sensors on targets `Q,R,S` and 3 sensors on target `P`.

6. Benchmark Results

In context of the present problem and associated model, we assessed the proposed approach experimentally using a problem set generated from the node arrangements of known vehicle routing problem (VRP) instances [43]. The forenamed VRP data set was selected since it provides node arrangements relevant for supply chain activities and contains instances with sizes ranging from 20 to 101 nodes.

Problem	p-n20	p-n23	p-n40	p-n45	p-n50	p-n51	p-n55	p-n60	p-n65	p-n70	p-n76	p-n101
nodes	20	23	40	45	50	51	55	60	65	70	76	101
sensors	10	12	20	23	25	26	28	30	33	35	38	51
targets	10	11	20	22	25	25	27	30	32	35	38	50
sensors/targets ratio	1.000	1.091	1.000	1.045	1.000	1.040	1.037	1.000	1.031	1.000	1.000	1.020
min. sensor capacity	1	1	1	1	1	1	1	1	1	1	1	1
max. sensor capacity	4	4	4	4	5	5	5	5	6	6	6	6
avg. sensor capacity	3	2.75	3	2.87	3	2.88	2.89	3	2.9	3	3	2.94
min. target coverage	4	4	4	4	4	4	4	4	4	4	4	4
max. target coverage	7	7	7	7	10	9	8	9	12	8	10	11
avg. target coverage	4.7	5.2	4.5	5.4	5.6	5.7	5.7	5.6	6.2	5.7	5.9	6.5
solution search space order of magnitude >	10^{8}	10^{9}	10^{14}	10^{22}	10^{24}	10^{25}	10^{28}	10^{31}	10^{38}	10^{39}	10^{40}	10^{63}

Table 5: Benchmark problems

The aforementioned data set provides a more suitable context than using purely random generated instances as previously adopted in the context of related problems and their specific associated models. For each instance, half of the nodes have been considered as sensors and the other half as targets. For the instances with odd number of nodes, we selected one more sensor compared to the number of targets. Only the selection of the sensor nodes was performed pseudo-randomly but in a manner whereby their capacity and range would make each problem challenging in the following sense. Each sensor has more targets in its range that its capacity and every target is covered by 4 or more sensors that have the target in their sensing range. The detailed breakdown for each of the benchmark problems is provided on a separate column in Table 5. In the latter, we can see that the sensor to target ratio is greater than unity for the odd node count instances since one more sensor node is available. Moreover, the problems are divided in three subsets according to the maximum target focusing ability of the sensors, as follows. The problem set with 20, 23, 40 and 45 nodes have sensors with target focusing capacities from 1 to maximum 4 targets, with an average target coverage ranging from 4.5 to 5.4. The problem set with 50, 51, 55 and 60 nodes have sensors with target focusing capacities from 1 to maximum 5 targets, with an average target coverage ranging from 5.6 to 5.7. The problem set with 65, 70, 76 and 101 nodes have sensors with target focusing capacities from 1 to maximum 6 targets, with an average target coverage ranging from 5.7 to 6.5. In order to make the problems challenging to solve without penalty, no reserve sensor capacity was considered. Thus, for all problems, we have an average sensor capacity of 3 or slightly lower than 3 for the odd node count instances (due to the presence on an additional sensor node). We can also note in Table 5, the progressively increasing solution search space order of magnitude with each larger problem instance, ranging from 10^8 combinations for the smallest instance of 20 nodes, up to 10^{63} combinations for the largest instance of 101 nodes.

6.1. Parameter Exploration and Performance Assessment

Fig. 9 depicts for different parameter values, the performance assessment of 20, 23, 40 and 45 node problems where the maximum sensor capacity is 4. Fig. 10 depicts for different parameter values, the performance assessment of 50, 51, 55 and 60 node problems where the maximum sensor capacity is 5. Fig. 11 depicts for different parameter values, the performance assessment of 65, 70, 76 and 101 node problems where the maximum sensor capacity is 6. In each of the aforementioned three figures, we have 4 sub-figures included, with the results for each of their respective problems. Each sub-figure depicts five *hs* and *hsmh* results which represent average values obtained after running the heuristic with eight different random seeds for different values of the pruned combinations and sample count per iteration, as follows.

In the leftmost side of each sub-figure, we have one pruned combination and 1000 samples (1/1000). After that, we have two pruned combinations and 300 samples (2/300) followed by two pruned combinations and 100 samples (2/100). Then, we have three pruned combinations and 300 samples (3/300). Finally, in the rightmost side, we have three pruned combinations and 100 samples (3/100). For each of the parameter values, we can see two line graphs and two bar graphs. The line graphs depict the gap to the best solution for *hs* (blue) and *hsmh* (gray) results while their total number of combinations explored during solution search is depicted by the red and orange bar graphs respectively. The corresponding values are provided in the tables situated below the graphs. For each problem instance, the best solution to compare against was selected as

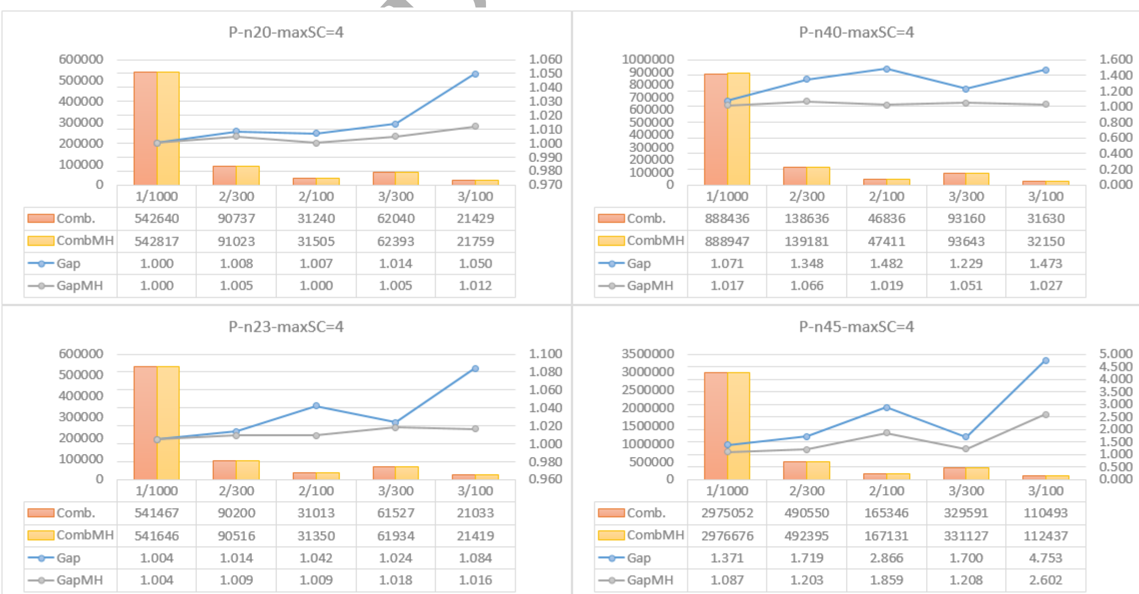


Fig. 9: Performance assessment for the 20, 23, 40 and 45 node problems where the maximum sensor capacity is 4

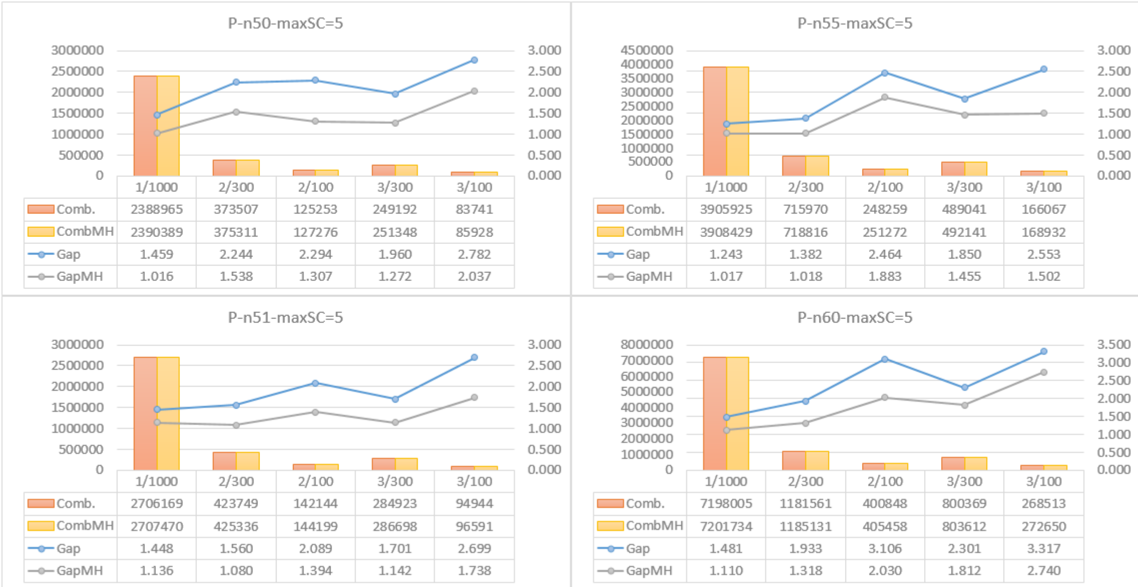


Fig. 10: Performance assessment for the 50, 51, 55 and 60 node problems where the maximum sensor capacity is 5

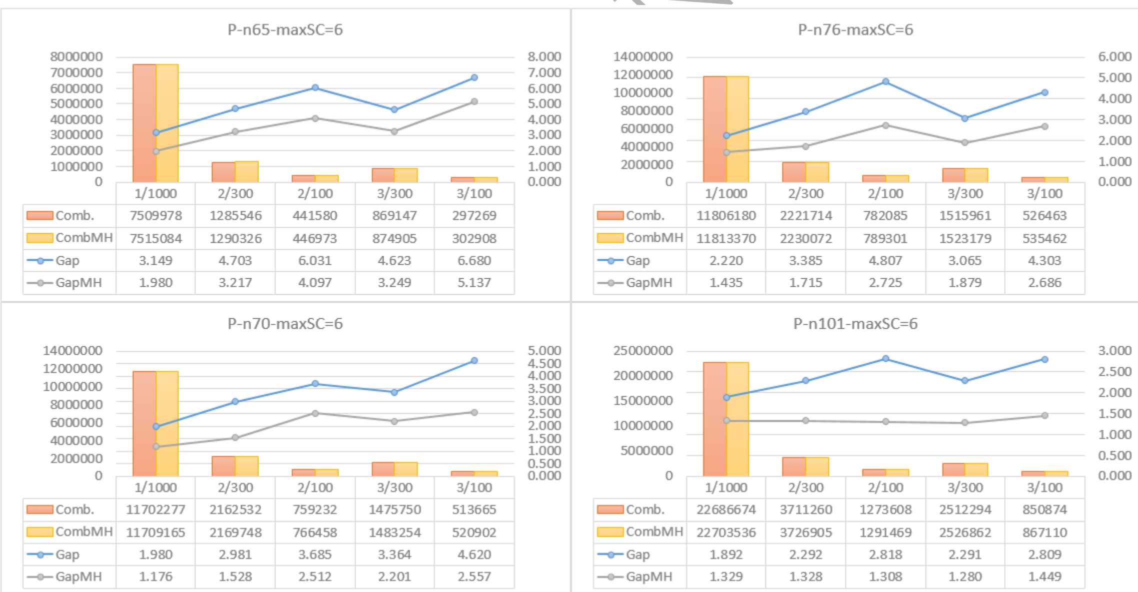


Fig. 11: Performance assessment for the 65, 70, 76 and 101 node problems where the maximum sensor capacity is 6

the one with the smallest value among all experimental results. Since for a given problem, the evaluation of any particular sensor-target assignment is on the average equally costly in terms of computation, this can serve as an appraisal of the computing budget corresponding to particular parameter values. As reference for the performed experiments, the average number of sensor-target assignment combinations that can be evaluated on a Core i7 platform, using a Java implementation, is on the order of 1000000 per second.

We can see that the application of the meta-heuristic improvements slightly increases the total number of sensor-target assignment combinations. The selected parameter values have been chosen based on the following considerations. For 1/1000, we let the search procedure explore a large number of samples while pruning only one combination per iteration. This allows to obtain the best results but the total number of explored combinations was large, which corresponds to large execution time. For 2/300, the search procedure is notably faster but the results are not as good compared to 1/1000. However, we can note that the meta-heuristic improvements have notable benefits on the results at the expense of only minor increase in the explored sensor-target assignment combinations due to the application of the meta-heuristic. For 2/100, the search procedure is even faster but the results are degraded and the application of the meta-heuristic improvements is not as effective. For 3/300, the results are somewhat less competitive in general compared to 2/300, hinting that pruning an additional combination per iteration allows for faster result generation at the expense of slightly more costly results. Finally, for 3/100, the search procedure is fastest but the results are the least competitive while the meta-heuristic improvements have limited benefits due to insufficient sampling and more aggressive pruning of sensor-target combinations.

6.2. Result Analysis

Table 6 shows the experimentally obtained benchmark results (integer rounded) presented in a separate column for each problem instance. The experiments have been conducted using additive penalty ($\rho_1 = 5000$) for the cases where less than 3 sensors are assigned to a target. We mention that in the table, the penalty values are displayed in multiples of 1000. For restricted energy budget, we used $\rho_2 = 6$, however any solution that exceeds the energy budget restriction can be adjusted to one respecting it by deassigning the cost-wise least impacting targets from the sensor(s) that use(s) the most capacity.

Problem	p-n20	p-n23	p-n40	p-n45	p-n50	p-n51	p-n55	p-n60	p-n65	p-n70	p-n76	p-n101
With maximum budget ($K \times \sum_{i=1}^{n} C_i$), full (100%) energy budget - all sensors used at full capacity												
nn [no penalty]	2791	2880	8061	5061	66407	64109	6726	7310	74289	13211	14054	99381
nn penalty ×1000	30	30	35	40	40	40	45	70	80	75	105	95
Tgts [0 sensors]	0	0	0	0	1	1	0	0	1	0	0	0
Tgts [1-2 sensors]	4	5	5	8	7	7	7	10	13	10	12	15
Tgts [3 sensors]	1	2	9	8	10	10	12	7	7	14	8	18
Tgts [4+ sensors]	5	4	6	6	7	7	8	13	11	11	18	17
nnmh [no penalty]	<u>1371</u>	4267	3433	7362	5519	3376	2903	4048	5952	4937	4519	86518
nnmh penalty ×1000	0	10	10	20	20	35	25	35	35	40	35	80
Tgts [1-2 sensors]	0	1	2	4	4	6	5	5	5	7	5	13
Tgts [3 sensors]	10	8	16	15	17	12	17	18	22	20	26	22
Tgts [4+ sensors]	0	2	2	3	4	7	5	7	5	8	7	15
hs [no penalty]	<u>1124</u>	<u>2181</u>	<u>1575</u>	<u>1024</u>	<u>5002</u>	<u>2538</u>	<u>1631</u>	<u>3042</u>	<u>4584</u>	<u>3131</u>	<u>3798</u>	<u>22869</u>
hs penalty ×1000	0	0	0	0	0	0	0	0	0	0	0	0
Tgts [3 sensors]	10	11	20	22	25	25	27	30	32	35	38	50
hsmh [no penalty]	<u>1124</u>	<u>2181</u>	<u>1529</u>	<u>938</u>	<u>3794</u>	<u>1900</u>	<u>1438</u>	<u>2513</u>	<u>2519</u>	<u>1799</u>	<u>2194</u>	<u>14632</u>
hsmh penalty ×1000	0	0	0	0	0	0	0	0	0	0	0	0
Tgts [3 sensors]	10	11	20	22	25	25	27	30	32	35	38	50
With 75% of the energy budget												
hsmh [no penalty]	1172	2283	1725	1171	3068	2029	1620	2274	2094	2072	2229	21721
hsmh penalty ×1000	40	45	75	85	95	95	105	115	120	135	145	190
Tgts [1 sensors]	0	0	0	0	1	0	1	1	0	0	0	0
Tgts [2 sensors]	8	9	15	17	17	19	19	21	24	27	29	38
Tgts [3 sensors]	2	2	5	5	7	6	7	8	8	8	9	12
Capacity use ratio	76%	76%	78%	75%	76%	80%	76%	77%	76%	71%	79%	76%
Sensor use	10/10	12/12	20/20	23/23	23/25	26/26	27/28	28/30	33/33	31/35	37/38	49/51
With 50% of the energy budget												
hsmh [no penalty]	5104	4730	14514	15105	16294	14738	14159	17235	16941	21829	16805	114216
hsmh penalty ×1000	75	85	150	165	190	190	205	225	240	265	285	375
Tgts [1 sensors]	5	6	10	11	13	13	14	15	16	18	19	25
Tgts [2 sensors]	5	5	10	11	12	12	13	15	16	17	19	25
Capacity use ratio	59%	52%	51%	48%	53%	57%	55%	51%	53%	55%	58%	56%
Sensor use	10/10	10/12	15/20	19/23	23/25	26/26	28/28	30/30	33/33	35/35	38/38	50/51

Table 6: Benchmark results

The table is divided into three sections presenting the results obtained using full energy budget (100%), three quarters budget (75%) and half budget (50%) respectively. The first section contrasts four types of results, as follows: nearest-neighbor (*nn*) solutions, nearest-neighbor solutions followed by meta-heuristic improvements (*nnmh*), heuristic solutions (*hs*) and heuristic solutions followed by meta-heuristic improvements (*hsmh*). The meta-heuristic solution improvements are carried out after restoring the initial allocation choices for all the sensors followed by an iterative attempt at target swapping among sensor pairs covering the same targets in pursuit of lowering the overall cost. First, we discuss the results obtained by using the myopic *nn* sensor assignment. We can see that such assignment is very unfavorable since there are instances where certain targets have no (0) sensors assigned. There is also a large number of targets that have either under-assigned (1-2 sensors) or over-assigned (4+) sensors while the number of targets with 3 assigned sensors is comparatively less. Next, we discuss the effect of subjecting the *nn* solutions to the meta-heuristic improvements. We can note that the *nnmh* solutions no longer contain targets with no sensors assigned. Also, the number of targets that have either under-assigned or over-assigned sensors is lower compared to the number of targets with 3 assigned sensors. We can note that for the 20 node instance in particular, all targets have 3 sensors assigned. Thus, the obtained *nnmh* solution results indicate the usefulness of the meta-heuristic improvements. However, the scope of improvement over the *nn* solutions is limited by their initial unfavorable sensor-target assignments. This becomes evident when we inspect the *hs* solutions which are notably better compared to the *nnmh* solutions. Thus, all *hs* solutions have 3 sensors assigned for each target, thus incurring no penalties. We can see that even for the 20 node instance, where the *nnmh* solution has no penalty, the *hs* solution still has a better cost. This hints that the heuristic search is not only able to avoid the penalties but the obtained assignment is also cost effective. With respect to the *hsmh* solutions obtained after subjecting the *hs* solutions to the meta-heuristic improvements, we can note that they exhibit for the most part improved costs, especially for the larger instances. Also, the *hsmh* solution values for the 20, 23 and 40 node instances are optimal for the first two (10^8 and 10^9 order of magnitude) and most likely optimal for the third one (10^{14}). We confirmed via exhaustive search the first two instances while the third one was assessed by removing certain combinations (with no votes across millions of sampled cost effective solutions) and then exhaustively searching the remaining (10^{12} order of magnitude) space. We also note that for the 20 and 23 node instances, the *hs* solutions obtained are also optimal. Thus, the meta-heuristic improvements offer most benefits when the solution search space order of magnitude is larger.

The second and third sections of Table 6, present the results obtained for the same problem instances when the energy budget is restricted. Since the benchmark problems are tight (no reserve capacity), the restriction of the energy budget involves the application of penalties which will be discussed next.

6.3. Impact of Energy Budget Restriction

The restriction of the energy budget results in solutions where certain sensors are unused or underused (used at less than full capacity). Such restriction allows to save energy with the benefit of allowing the network to operate longer but depending on the degree to which the energy budget is restricted, the localization cost may increase. In addition, it may no longer be possible to respect the requirement to have a certain number of sensors assigned per target, which corresponds to solutions exhibiting penalties. Also, for larger problem instances, separate penalty may be incurred at the heuristic search stage in case that the solution exceeds the energy budget restriction. However, such penalty can be mitigated by removing a number of sensor-target assignments (from the sensors with the highest capacity use) whereby the solution cost is allowed to increase such that the energy restriction can be satisfied. Consequently, this procedure is applied, if needed, during the meta-heuristics such that any remaining penalty is resulting from the assignment of less than 3 sensors per target.

In Table 6, second section (75% energy budget), we observe that all solutions exhibit penalties since the sensor network cannot use its full capacity. As such, the sensor network capacity use ratio varies from 71% (p-n70) to 80% (p-n51). Regarding the sensor use, some sensors are unused in certain cases, ranging from 1 (p-n55, p-n76) to 4 (p-n70). Moreover, we note that for most problems, the solutions predominantly assign 2 sensors for most targets while the number of targets with 3 sensors assigned is notably smaller. There are also a few problems (p-n50, p-n55 and p-n60) where one target is assigned only one sensor. If we compare the solutions disregarding the penalties, we can see that in most cases the solution values have higher cost compared to those obtained in the case where 100% energy budget is used. There are a few exceptions (p-n50, p-n60 and p-n65) but in such solutions, while some targets can be localized relatively well even with 2 sensors (given the particular node arrangements of those problems), a few targets are very poorly localized, especially when being assigned only one sensor for some targets. This stems from the use of assignment choices typically unavailable when assigning 3 sensors for each target (avoiding the penalty).

In the third section of Table 6 (50% energy budget), we can observe that all solutions exhibit significant penalties (the sensor network is severely restricted). The sensor network capacity use ratio varies from 48% (p-n45) to 59% (p-n20). We also note that for all problems, the obtained solutions assign only 1 or 2 sensors per target. Even if we disregard the penalty, the corresponding solutions have significantly higher cost even when compared to those obtained in the case where 75% energy budget is used. With respect to sensor use, some sensors are also unused in certain cases, ranging from 1 (p-n101) to 5 (p-n40).

For all three energy budget levels (100%, 75% and 50%), Fig. 12 shows the average sensor capacity use corresponding to different energy budgets used to generate the solutions for each of the benchmark problems. Fig. 12 also shows the minimum and maximum sensor capacity for each problem.

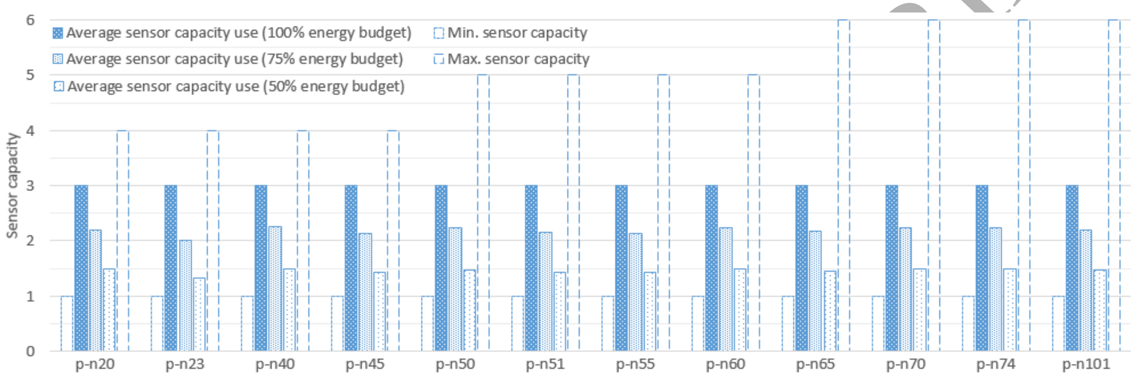


Fig. 12: Average sensor capacity use relative to the benchmark results obtained for each problem instance with different energy budgets

We illustrate in the following the effect of energy budget restriction on the solution obtained for the first problem (p-n20) from the benchmark results presented in Table 6. The problem has 10 sensors with an average capacity of 3 and 10 targets such that the full energy budget (needed to assign 3 sensors for each target) has a value of 30. Table 7 presents the problem data (sensor and target locations as well as target coverage) with the sensor-target assignment solution for full energy budget where 3 sensors are assigned for each target (target assignment count = 3) and each sensor is used at full capacity (capacity use = 100%).

Table 8 and Table 9 provide the sensor-target assignment solutions for restricted energy budgets of three quarters (75%) and respectively half (50%). Instead of locations, these two tables provide the target assignment count (TAC) and capacity use (CU) values for each target and sensor respectively.

In all three tables, each column provides the related information for each sensor while each row provides the details for each target. Each target row has markings in round brackets indicating the sensors covering

Table 7: P-n20 (100% energy budget)

.	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	Range	55	45	55	55	50	40	50	40	50	50
Target	Location	107,125	85,70	76,123	186,125	168,104	152,57	131,61	122,102	183,74	171,85
T1	152,72					(-)	(-)	(x)		(x)	(x)
T2	143,87	(-)				(-)	(x)	(-)	(x)	(-)	(x)
T3	152,116	(x)			(-)	(x)			(-)		(x)
T4	107,93	(-)	(-)	(x)				(x)	(x)		
T5	122,72		(-)				(x)	(x)	(x)		
T6	183,114				(x)	(x)				(x)	(-)
T7	171,45						(x)	(-)		(x)	(x)
T8	88,112	(x)	(x)	(x)					(-)		
T9	180,57					(-)	(x)	(x)		(x)	(-)
T10	125,121	(x)		(-)		(x)			(x)		

Table 8: P-n20 (75% energy budget)

Target\	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	TAC \ CU	3/3=1	1/1=1	1/2=0.5	1/1=1	2/3=0.66	2/4=0.5	3/4=0.75	4/4=1	2/4=0.5	3/4=0.75
T1	2					(-)	(-)	(x)		(-)	(x)
T2	2	(-)				(-)	(-)	(-)	(x)	(-)	(x)
T3	2	(x)			(-)	(x)			(-)		(-)
T4	3	(-)	(x)	(-)				(x)	(x)		
T5	2		(-)				(-)	(x)	(x)		
T6	2				(x)	(-)				(-)	(x)
T7	2						(x)	(-)		(x)	(-)
T8	2	(x)	(-)	(x)					(-)		
T9	2					(-)	(x)	(-)		(x)	(-)
T10	3	(x)		(-)		(x)			(x)		

Table 9: P-n20 (50% energy budget)

Target\	Sensor	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
	TAC \ CU	2/3=0.66	1/1=1	1/2=0.5	1/1=1	3/3=1	1/4=0.25	3/4=0.75	1/4=0.25	1/4=0.25	1/4=0.25
T1	2					(-)	(-)	(x)		(-)	(x)
T2	2	(-)				(x)	(-)	(x)	(-)	(-)	(-)
T3	2	(x)			(-)	(x)			(-)		(-)
T4	2	(-)	(x)	(-)				(-)	(x)		
T5	1		(-)				(-)	(x)	(-)		
T6	1				(x)	(-)				(-)	(-)
T7	1						(x)	(-)		(-)	(-)
T8	1	(-)	(-)	(x)					(-)		
T9	1					(-)	(-)	(-)		(x)	(-)
T10	2	(x)		(-)		(x)			(-)		

the target. In this setting, simple sensor-target coverage is marked by (-) while assignment to a covered target is marked by (x). The corresponding benchmark solution cost is 1124(+0 penalty) for 100% energy budget, 1172(+40000 penalty) for 75% energy budget and 5104(+75000 penalty) for 50% energy budget.

We note that the sensor network is using progressively less capacity of its sensors with decreased energy budget according to the energy budget limitation. Thus, in Table 7, the use of full energy budget corresponds to a sum of target assignments of 30 while the network capacity use ratio is 100%. Then, in Table 8, for 75% energy budget, the target assignment sum decreases to $2+2+2+3+2+2+2+2+2+3 = 22$ (the limit value since $30 \times 0.75 = 22.5$) while the network capacity use ratio decreases to $(1+1+0.5+1+0.66+0.5+0.75+1+0.5+0.75)/10 = 0.766$ ($\approx 76\%$). Finally, in Table 9, for 50% energy budget, the target assignment sum decreases to $2 + 2 + 2 + 2 + 1 + 1 + 1 + 1 + 1 + 2 = 15$ while the network capacity use ratio decreases to $(0.66 + 1 + 0.5 + 1 + 1 + 0.25 + 0.75 + 0.25 + 0.25 + 0.25)/10 = 0.591$ ($\approx 59\%$).

The Appendix contains an extensive amount of additional tables providing the data for the benchmark problems with node counts ranging from 23 to 101 in terms of sensor and target locations as well as the best sensor-target assignment solutions for full energy budget.

7. Conclusion and Future Work

In this work, we proposed an evolutionary learning heuristic approach suitable for wireless sensor network management which allows an efficient assignment of multiple sensors with different capabilities, in terms of sensing ranges and focusing capacities, to multiple target assets, in order to minimize the overall asset localization error cost. Moreover, the technique allows to effectively manage limited WSN capabilities whereby an energy budget restriction limits the full use of the sensors capacities in order to conserve energy. Employing the energy budget restriction as a part of WSN management for asset localization provides a trade-off between energy conservation and localization performance. An illustrative case study was presented both with and without energy budget restrictions. Furthermore, we experimentally assessed our technique via extensive benchmarks in various parameter settings. The obtained results indicate the effectiveness of the proposed approach which allows for an user applicable trade-off in terms of solution quality versus computation time. In this context, sufficiently good solutions can be obtained in shorter time when needed, using a smaller sample population size and a larger amount of removed combinations per generation. Conversely more competitive solutions can be found at the expense of more computing time when using a larger sample population size and a smaller amount of removed combinations per generation.

In future work, a further research direction can be pursued toward the extension of the procedure in order to improve the coverage by determining the sensor locations over an instrumented space. Another potential research direction relates to considering the choice of sensors to be deployed in terms of cost/benefit ratio, taking into account the acquisition cost of the sensors whereby more capable sensors may be more expensive.

References

[1] Y. Kuroki, T. Matsui, An approximation algorithm for multidimensional assignment problems minimizing the sum of squared errors, Discrete Applied Mathematics 157 (9) (2009) 2124–2135.

[2] P. P. Storms, F. C. R. Spieksma, An lp-based algorithm for the data association problem in multitarget tracking, Computers & Operations Research 30 (7) (2003) 1067–1085.

[3] H.-J. Bandelt, A. Maas, F. C. Spieksma, Local search heuristics for multi-index assignment problems with decomposable costs, Journal of the Operational Research Society 55 (7) (2004) 694–704.

[4] A. B. Poore, S. Gadaleta, Some assignment problems arising from multiple target tracking, Mathematical and computer modelling 43 (9) (2006) 1074–1091.

[5] D. Goossens, S. Polyakovskiy, F. C. Spieksma, G. J. Woeginger, The focus of attention problem, Algorithmica 74 (2) (2016) 559–573.

[6] J. L. Walteros, C. Vogiatzis, E. L. Pasiliao, P. M. Pardalos, Integer programming models for the multidimensional assignment problem with star costs, European Journal of Operational Research 235 (3) (2014) 553–568.

[7] R. E. Burkard, R. Rudolf, G. J. Woeginger, Three-dimensional axial assignment problems with decomposable cost coefficients, Discrete Applied Mathematics 65 (1-3) (1996) 123–139.

[8] F. Librino, M. Levorato, M. Zorzi, An algorithmic solution for computing circle intersection areas and its applications to wireless communications, Wireless Communications and Mobile Computing 14 (18) (2014) 1672–1690.

[9] S. Waharte, N. Trigoni, Supporting search and rescue operations with UAVs, in: Emerging Security Technologies (EST), 2010 International Conference on, IEEE, 2010, pp. 142–147.

[10] F. A. Özsoy, M. Ç. Pinar, An exact algorithm for the capacitated vertex p-center problem, Computers & Operations Research 33 (5) (2006) 1420–1436.

[11] S. Ray, M. Debbabi, M. Allouche, N. Léchevin, M. Bélanger, Energy-efficient monitor deployment in collaborative distributed setting, IEEE Transactions on Industrial Informatics 12 (1) (2016) 112–123.

[12] L. Birnbaum, G. Collins, M. Freed, B. Krulwich, Model-based diagnosis of planning failures., in: AAAI, Vol. 90, 1990, pp. 318–323.

[13] J. Salerno, M. Hinman, D. Boulware, Building a framework for situation awareness, Tech. rep., DTIC Document (2004).

[14] Y. Z. Zhao, O. P. Gan, Distributed design of RFID network for large-scale RFID deployment, in: 2006 IEEE International Conference on Industrial Informatics, Singapore, 2006, pp. 44–49.

[15] A. Murat, G. Laporte, V. Verter, A global shooting algorithm for the facility location and capacity acquisition problem on a line with dense demand, Computers & Operations Research 71 (Supplement C) (2016) 1–15.

[16] S. Li, A 1.488 approximation algorithm for the uncapacitated facility location problem, Information and Computation 222 (2013) 45–58.

[17] P. Yang, W. Wu, M. Moniri, C. Chibelushi, Efficient object localization using sparsely distributed passive RFID tags, IEEE Transactions on Industrial Electronics 60 (12) (2013) 5914–5924.

[18] E. L. Souza, A. Campos, E. F. Nakamura, Tracking targets in quantized areas with wireless sensor networks, in: Local Computer Networks (LCN), 2011 IEEE 36th Conference on, IEEE, 2011, pp. 235–238.

[19] R. Boutaba, N. Achir, M. St-Hilaire, E. F. Nakamura, Planning and deployment of wireless sensor networks (2014).

[20] E. F. Nakamura, A. A. F. Loureiro, A. Boukerche, A. Y. Zomaya, Localized algorithms for information fusion in resource constrained networks, Information Fusion (15) (2014) 2–4.

[21] F. Gustafsson, F. Gunnarsson, Localization in sensor networks based on log range observations, in: Information Fusion, 2007 10th International Conference on, IEEE, 2007, pp. 1–8.

[22] É. L. Souza, E. F. Nakamura, R. W. Pazzi, Target tracking for sensor networks: A survey, ACM Computing Surveys (CSUR) 49 (2) (2016) 30.

[23] A. Oka, L. Lampe, Distributed target tracking using signal strength measurements by a wireless sensor network, IEEE Journal on Selected Areas in Communications 28 (7) (2010) 1006–1015.

[24] B. Sobhani, M. Mazzotti, E. Paolini, A. Giorgetti, M. Chiani, Multiple target detection and localization in uwb multistatic radars, in: Ultra-WideBand (ICUWB), 2014 IEEE International Conference on, IEEE, 2014, pp. 135–140.

[25] Y. K. An, S.-M. Yoo, C. An, B. E. Wells, Rule-based multiple-target tracking in acoustic wireless sensor networks, Computer Communications 51 (2014) 81–94.

[26] W.-P. Chen, J. C. Hou, L. Sha, Dynamic clustering for acoustic target tracking in wireless sensor networks, IEEE transactions on mobile computing 3 (3) (2004) 258–271.

[27] X. Jin, S. Sarkar, A. Ray, S. Gupta, T. Damarla, Target detection and classification using seismic and pir sensors, IEEE Sensors Journal 12 (6) (2012) 1709–1718.

[28] I. Bisio, A. Sciarrone, S. Zappatore, A new asset tracking architecture integrating RFID, Bluetooth Low Energy tags and ad hoc smartphone applications, Pervasive and Mobile Computing 31 (2016) 79–93.
[29] D. Balakrishnan, A. Nayak, An efficient approach for mobile asset tracking using contexts, IEEE Transactions on Parallel and Distributed Systems 23 (2) (2012) 211–218.
[30] V. Savic, S. Zazo, Cooperative localization in mobile networks using nonparametric variants of belief propagation, Ad hoc networks 11 (1) (2013) 138–150.
[31] L.-K. Soh, C. Tsatsoulis, Reflective negotiating agents for real-time multisensor target tracking, in: International Joint Conference On Artificial Intelligence, Vol. 17, LAWRENCE ERLBAUM ASSOCIATES LTD, 2001, pp. 1121–1127.
[32] V. Isler, S. Khanna, J. Spletzer, C. J. Taylor, Target tracking with distributed sensors: The focus of attention problem, Computer Vision and Image Understanding 100 (1) (2005) 225–247.
[33] J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage control for mobile sensing networks, IEEE Transactions on robotics and Automation 20 (2) (2004) 243–255.
[34] D. W. Pentico, Assignment problems: A golden anniversary survey, European Journal of Operational Research 176 (2) (2007) 774–793.
[35] R. Burkard, M. Dell'Amico, S. Martello, Assignment problems: revised reprint, SIAM, 2012.
[36] W. P. Pierskalla, Letter to the editor-the multidimensional assignment problem, Operations Research 16 (2) (1968) 422–431.
[37] G. Arslan, J. R. Marden, J. S. Shamma, Autonomous vehicle-target assignment: A game-theoretical formulation, Journal of Dynamic Systems, Measurement, and Control 129 (5) (2007) 584–596.
[38] M. Larsen, Branch and bound solution of the multidimensional assignment problem formulation of data association, Optimization Methods Software 27 (6) (2012) 1101–1126.
[39] C. Vogiatzis, E. L. Pasiliao, P. M. Pardalos, Graph partitions for the multidimensional assignment problem, Computational Optimization and Applications 58 (1) (2014) 205–224.
[40] W. E. Hart, N. Krasnogor, J. E. Smith, Memetic evolutionary algorithms, in: Recent advances in memetic algorithms, Springer, 2005, pp. 3–27.
[41] D. Karapetyan, G. Gutin, A new approach to population sizing for memetic algorithms: a case study for the multidimensional assignment problem, Evolutionary computation 19 (3) (2011) 345–371.
[42] A. Soeanu, S. Ray, M. Debbabi, J. Berger, A. Boukhtouta, A learning based evolutionary algorithm for distributed multi-depot vrp, in: KES, 2012, pp. 49–58.
[43] P. Augerat, J. Belenguer, E. Benavent, A. Corberan, D. Naddef, G. Rinaldi, Computational results with a branch and cut code for the capacitated vehicle routing problem, Tech. rep., Universite Joseph Fourier, Grenoble, France (1995).