An integer linear programming approach for radio-based localization of shipping containers in the presence of incomplete proximity information

Stefano Abbate, *Student Member, IEEE,* Marco Avvenuti, *Member, IEEE,* Paolo Corsini, *Member, IEEE,*
Barbara Panicucci, Mauro Passacantando and Alessio Vecchio

***Abstract*—The most advanced solutions that are currently adopted in ports and terminals use RFID- and GPS-based technologies to identify and localize shipping containers in the yard. Nevertheless, because of the limitations of these solutions, the position of containers is still affected by errors and it cannot be determined in real-time. In this paper a non-conventional approach is presented: each container is equipped with nodes that use wireless communication to detect neighbor containers and to send proximity information to a base station. At the base station, geometrical constraints and proximity data are combined to determine the positions of containers. Missing information due to faulty nodes is tolerated by modeling geometrical constraints as an integer linear programming problem. Numerical simulations show that most of containers can be localized even when the number of nodes affected by faults is in the order of 30%.**

***Index Terms*—Logistics, wireless sensor network, localization, container.**

I. Introduction

PORTS and terminals undergo a continuous increase in the level of traffic. Today the port of Singapore, one of the busiest ports in the world in terms of container handling, manages more than twenty million of shipping containers per year. For this reason, ports and terminals make use of automated container handling and transportation solutions, especially in countries with high labour costs [1]. Higher productivity has been achieved through advanced terminal layouts, more efficient IT-support and improved logistics control software systems [2]. Computers are employed to schedule and control different kinds of handling operations, such as identification [3] and tracking of containers, and their localization in the yard.

A. Shipping containers

Containers, also known as shipping containers, intermodal transport units or isotainers, are used for freight cargo transport on trucks, trains and ships. Their introduction has improved cargo shipping and has driven modifications to freight-moving standards: removable truck bodies or swap bodies have been forced into standard sizes and shapes, and their use across the globe has lessened the problems caused by incompatible rail gauge sizes of different countries.

The most widespread containers are those conforming to the ISO standard, whose measures have been accepted internationally: 8 feet (2.44 m) width, 8 feet and 6 inches (2.59 m) height, and two standard lengths of 20 and 40 feet (6.10 and 12.20 m). The standard also includes specific corners used to manage containers by means of cranes; the hardiness of corners and edges permits to arrange the containers in stacks, obtaining space benefits. Various container types are available for different needs: general purpose, high cube, temperature controlled (from -25 C to +25 C) reefer, open top, open side, and many others.

B. Container terminals and storage

Seaport container terminals can be described as open systems of material flow with two external interfaces: the quayside where loading and unloading of ships take place, and the land-side where containers are loaded and unloaded on/off trucks and trains [4]. Import as well as export containers are stored in stacks and divided into a number of blocks. This facilitates the decoupling of quayside and land-side operations.

After arrival at the terminal, the container is identified in order to obtain its major data such as content, destination, outbound vessel, shipping line. Then, it is picked up by internal transportation equipment and distributed to one of the storage blocks in the yard. Once the designated vessel is ready, the container is unloaded from the yard block and transported to the berth where quay cranes load the container onto the vessel at a pre-defined stacking position. A reverse order is followed to handle an import container.

Current computer-aided solutions include real-time assignment of transportation orders to vehicles, routing and scheduling of vehicle trips for transportation, assignment of storage slots to individual containers [5]. In general, a wireless communication system provides connectivity to vehicles and operators all over the terminal. GPS- and RFID-based solutions have been extensively adopted to achieve automatic localization and/or identification. GPS receivers are not directly installed on containers, but on top of the transport and stacking equipment. The position is measured, translated into

Manuscript received ...

S. Abbate is a PhD student in Computer Science and Engineering, IMT Institute for advanced studies, Lucca, Italy, E-Mail: stefano.abbate@imtlucca.it.

M. Avvenuti, P. Corsini and A. Vecchio are with the Department of Information Engineering, University of Pisa, Italy, E-Mail: m.avvenuti@iet.unipi.it, p.corsini@iet.unipi.it, a.vecchio@iet.unipi.it

B. Panicucci is with Department of Sciences and Methods of Engineering, University of Modena and Reggio Emilia, Italy, E-Mail: barbara.panicucci@unimore.it

M. Passacantando is with the Department of Applied Mathematics, University of Pisa, Italy, E-Mail: passacantando@dma.unipi.it

yard coordinates and transmitted to a central system whenever a container is lifted or dropped. This way, database queries provide the geo-location of containers when needed. RFID technology enables a quick identification of containers, but it is less useful to determine their position. Also, RFID systems require a fixed or mobile infrastructure to read the tags, and the process, in many cases, includes human-driven or semi-automated operations.

C. Motivation

In general, shipping data about containers and their position within the yard are known in advance. However, such information is not always correct or completely up-to-date because of operational disturbances. For example, while in the yard, a container can be moved several times for content control, custom formalities, routing operations, etc. Thus, despite of currently available solutions, real-time identification and localization of containers are still error-prone activities and require human intervention to manage anomalous situations (for example, by physically searching the misplaced containers).

In these cases the RFID- and GPS-based techniques previously mentioned cannot provide a completely automated solution to the problem. Using the GPS, the position of a container can be indirectly determined from the position of the truck or quay crane when the container is lifted or dropped. But GPS receivers cannot be directly attached to containers for real-time identification and positioning, as this would prevent the receiver to be in line-of-sight with satellites when the container is not on the surface of a container stack. Instead, automatic localization in the yard can be achieved by equipping containers with smart wireless nodes and using a positioning scheme based on proximity information, as the one described in this paper. At a glance, with the proposed approach wireless sensors detect neighbor containers and send proximity information to a base station, where it is combined with geometrical constraints to determine the relative positions of containers. In a first step, a simple algorithm places those containers whose position is not ambiguous (the strawman approach). In a second step, missing proximity information due to faulty nodes is tolerated by modeling geometrical constraints as an integer linear programming problem.

Obviously this technique, based on the idea of embedding intelligence directly on containers, involves some costs. Nevertheless, it is important to notice that: i) the cost of a single container is in the order of thousands USDs, thus adding equipment increases the total cost only a small fraction; ii) beside localization, other orthogonal problems can be faced and solved through the use of intelligent devices, examples include security [6], supervision [7] and monitoring [8]. Moreover, from a more distant perspective, the authors believe that the Internet of Things (IoT) will become, sooner or later, a reality. According to the IoT vision, the world of physical objects becomes seamlessly integrated into the information network and participates actively in business, information and social processes, where smart things communicate among themselves and with the surrounding environment. Starting from this assumption, it is possible to imagine that, in a not too distant future, also the ordinary shipping containers will be enhanced with computing and communication capabilities. Then a system like the one proposed in this paper can be integrated with small additional costs.

II. Proposed approach

The localization technique described in this paper enables the automated on-line discovery of the positions of containers in the yard. This is achieved by means of wireless sensor nodes, placed on containers, that cooperate to collect proximity information and communicate with a base station, where positions are computed.

A. Assumptions and definitions

It is supposed that containers are not turned upside down, and that the long edges of containers are always parallel with each others. Containers, that are obviously parallelepipedons, are then supposed to be positioned within a three dimensional grid. These hypothesis are realistic and not over-restrictive: the placement of containers in a yard usually follows the grid model, as it maximizes the usage of the available surface. Even though the proposed approach can be extended to cope with different container types, for the sake of simplicity in this article all the containers are assumed to have the same size.

Two containers are defined as *contiguous* if they are located so that at least one edge of the first container is contiguous to an edge of the second container. Two containers are said *adjacent* if they are located face to face (i.e., four edges are contiguous). A *group* is a set of containers where every container is adjacent to, at least, another container.

More formally, given a reference system, the position of a container can be expressed through its three dimensional coordinates. Given a container A, its coordinates are indicated as $x(A)$, $y(A)$, $z(A)$. Also, since containers are placed according to a grid model, $x(A)$, $y(A)$, and $z(A)$ are three integer values that specify the element of the grid where A is placed. Without loss of generality, it can be supposed that the coordinates of the containers are non-negative, thus $x(A) \in \{0, \ldots, M_x - 1\}$, $y(A) \in \{0, \ldots, M_y - 1\}$, and $z(A) \in \{0, \ldots, M_z - 1\}$ with M_x, M_y, M_z positive values.

According to this notation, two containers A and B are contiguous if

$$\begin{cases} |x(A) - x(B)| \leq 1, \\ |y(A) - y(B)| \leq 1, \\ |z(A) - z(B)| \leq 1, \\ |x(A) - x(B)| + |y(A) - y(B)| + |z(A) - z(B)| \leq 2, \end{cases}$$

and they are adjacent if

$$|x(A) - x(B)| + |y(A) - y(B)| + |z(A) - z(B)| = 1$$

Given that a container cannot be turned upside down and that containers are placed according to a grid model where elements are parallelepipedons, a container can be oriented

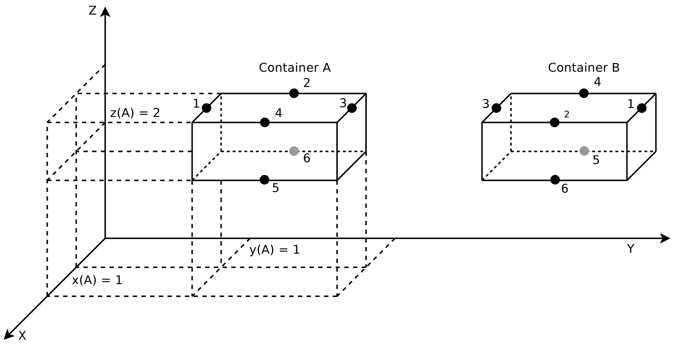


Figure 1: Position of nodes, reference system, and orientation

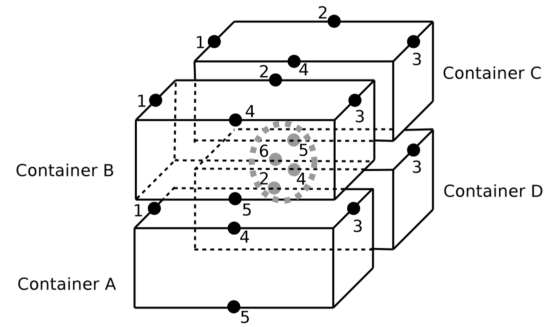


Figure 2: Every node can be close to up to three other nodes

only in two ways. Thus, the orientation of a container A can be expressed as $o(A)$, where $o(A) \in \{0, 1\}$.

If the containers are organized in the yard as separated groups, the localization procedure described in the following can be applied separately to each group.

B. Container equipment

Every container is equipped with wireless nodes that i) detect the presence of nodes belonging to other containers; ii) calculate their relation of proximity on the base of measures of the Received Signal Strength (RSS) of the wireless communication channel.

As known, localization techniques based on RSS are characterized by limited accuracy, in particular when used in the presence of obstacles [9], [10]. To overcome this limitation, the system has been conceived according to the following guidelines:

- the proximity relation between two nodes is modeled as a value in a binary domain: *close*/*far*;
- the proximity relation between nodes belonging to different containers can be used to infer contiguity and/or adjacency conditions between such containers.

An obvious placement strategy can be based on the use of six wireless nodes placed at the center of the container faces. In this way, proximity between two nodes can be easily translated into a condition of adjacency between the corresponding faces, and thus between containers.

Unfortunately, an initial set of experiments showed that wireless communication between nodes of the same container was prevented by its metallic nature. Thus, in order to guarantee a line of sight between nodes belonging to the same container, and considering that there is always a small amount of space between adjacent containers, the nodes have been moved to the edges.

Figure 1 shows the placement of the nodes on a container. Four sensors are placed on the edges of the upper face, whereas two other sensors are placed on the edges of the bottom face of the container. The unsymmetrical distribution of nodes with respect to the horizontal plane is justified by the fact that containers cannot be turned upside down. Every node is identified by a unique node identifier, called *NodeId*. The NodeId is composed of two parts: the *ContainerId*, that identifies the container a node belongs to, and the *EdgeId*, that identifies the edge where the node is placed.

Figure 1, besides the EdgeId of nodes, also shows the direction of the axes of the reference system and two containers with different orientations ($o(A) = 1$ and $o(B) = 0$).

With this second placement strategy, proximity relations between nodes can still be used to derive contiguity information about the edges of containers, and in turn adjacency between containers. To this purpose, two nodes are considered close if they belong to contiguous edges, far in all other cases.

C. Detection of proximity information between nodes

Given two containers *A* and *B*, there are, in theory, 36 proximity relations between the six nodes of *A* and the six nodes of *B*. Actually, because of the storage rules previously mentioned (grid model where long edges are always parallel), and because of the position of nodes on the edges of containers, the number of significant proximity relations is equal to 20. For example, node 1 of container *A* cannot be close to nodes 2, 4, 5, 6 of container *B*, as it would violate the parallelism of long edges of the two containers. Similarly, node 3 of container *A* cannot be close to nodes 2, 4, 5, 6 of container *B*, and nodes 2, 4, 5, 6 of container *A* cannot be close to nodes 1 and 3 of container *B*. In other words, the grid model defines compatibility rules between edges and, in turn, between nodes.

Every node maintains a table that contains the NodeId of nodes in close proximity (NCP table). Because of the storage rules and grid model, every node can be in close proximity with up to three other nodes, as shown in Figure 2. Thus, the NCP table can contain a maximum of three entries. Also note that, from the nodes that are close, every node is from a different container.

To determine its proximity with respect to other nodes, every node periodically broadcasts a beacon. The beacon contains the NodeID of the sender. The emission of the beacons is performed with the same period T for all the nodes in the network, but there is no synchronization among different nodes (T is equal to $30s$ in the implemented prototype). At the same time, every node listens to the radio channel for the possible reception of beacons generated by other nodes. On receiving a beacon, the NodeID of the sender is analyzed, and it is checked if the EdgeID of the sender belongs to the set of edges compatible with the edge of the receiver (i.e., the proximity relation between the sender and the receiver is significant). If the edges are not compatible, the beacon is not further analyzed. For example, if a node with EdgeID equal to 1 receives a beacon from a node with EdgeID equal to 1 or 3, then the beacon is further analyzed as described in the following, otherwise the beacon is discarded. If the edges are compatible, the receiver node compares the RSS of the received beacon against a fixed threshold (experiments in a real setting have been carried out to tune this threshold to reflect an approximate distance of $1m$). If the RSS is greater than the threshold, the receiver inserts the NodeId of the sender node within its NCP table.

To improve the stability of the system and make it tolerant to possible packet losses, a node is removed from the NCP table only if no beacons are received from that node during a time longer than kT, where k is a configurable parameter (set to 3 in the implemented prototype). Because of the movement of containers, it may happen that a node receives the beacons from its new neighbors while its NCP table still contains the entries of its old neighbors (that are removed after kT time). To manage this situation the following policy is adopted: in case a beacon coming from a new node is received, let e be the EdgeId of the new node and let be S the subset of entries of the NCP table of the receiver that are compatible, in terms of edges, with e; if S already contains the maximum amount of entries (i.e. 3 if $e \in \{2, 4, 5, 6\}$ or 1 if $e \in \{1, 3\}$), the oldest entry in S is discarded and an entry containing the information coming from the new node is added to the table. In other words, with scope limited to sets of compatible edges, recent information is preferred to old data.

D. Collection and storage of proximity information

Each node must send the content of its NCP table to the base station. This operation, called *collection*, runs with period hT, where h is a configurable parameter (equal to k in the implemented prototype). Transfer of data to the base station is achieved through the standard Collection Tree Protocol (CTP), provided by the TinyOS operating system (TinyOS has been used as the base platform for the implementation of the system, as described in Section VII). In CTP, transfer of data is performed through a multi-hop routing tree that converges at the base station. Every node takes part in the forwarding activity and routing is based on a shortest-path algorithm (together with mechanisms that take into account the quality of a link). The tree is built and maintained independently with respect to the operations aimed at the detection of proximity information, and its topology does not depend on the position of containers (Figure 3). If a node gets broken, that node will be excluded from the routing tree and it will not participate in the forwarding activity (CTP detects broken links and selects another node as the next hop). This is not a problem until the number of broken nodes gets so high that large parts of the network become unreachable.

It is important to notice that the focus of this paper concerns the localization mechanisms and that transfer of information from nodes to the base station can be achieved through standard routing protocols. Thus, any routing protocol that is able to provide connectivity to the base station and that is able to dynamically re-arrange the routes in case of faults could be suitable (literature about routing protocols for wireless sensor networks is quite abundant, see [11] for a survey).

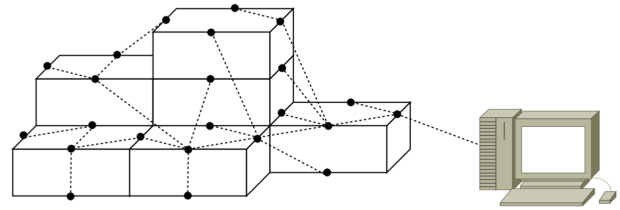


Figure 3: The routing tree is independent from the position of containers

Since the NCP table has a maximum of three entries, it is possible to insert all its data in a packet (NCP packet) of fixed size and format. In more detail, the packet contains the NodeId of the sender and the NodeIds of all the nodes in its close proximity. If the number of nodes in close proximity is lower than three, the remaining fields are set to zero.

Table I shows the content of a NCP packet that represents the relations of close proximity of node 2 of container A with nodes 6, 5, 4 of containers B, C, D respectively, as also depicted in Figure 2 (NodeIds are expressed in the form ContainerId, EdgeId).

$A, 2$	$B, 6$	$C, 5$	$D, 4$

Table I: Example of a NCP packet

E. Representation of proximity information at the base station

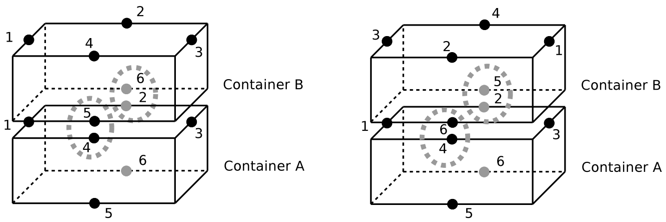
The base station maintains a set $\mathcal{C}$ that includes the ContainerIds of all known containers. The set is initially empty and it is managed as follows: $i)$ each time a NCP packet is received, all the NodeIds contained in the packet are extracted; $ii)$ the ContainerIds of the extracted NodeIds are added to $\mathcal{C}$; $iii)$ if a ContainerId is already included in $\mathcal{C}$, the corresponding element is refreshed; $iv)$ elements of $\mathcal{C}$ are removed from the set when not refreshed for a time equal to z times kT (with z set to 3 in the implemented prototype).

Let the relation of close proximity between node *i* of container *A* and node *j* of container *B* be represented as $R_{A,B}(i, j)$. The base station maintains also a set $\mathcal{R}$ of relations of close proximity, initially empty, and managed as follows: $i)$ each time a NCP packet is received, the relations of close proximity derived from the packet content are added to $\mathcal{R}$; $ii)$ if a relation of close proximity is already included in $\mathcal{R}$, the corresponding element is refreshed; $iii)$ elements of $\mathcal{R}$ are removed from the set when not refreshed for a time equal to z times kT.

Given the symmetrical nature of proximity relations, every time $R_{A,B}(i, j)$ is inserted into $\mathcal{R}$, then $R_{B,A}(j, i)$ is inserted as well. This helps to make the system more resilient to the possible loss of packets during the collection phase.

F. Inferring the position of containers from the relations of close proximity

As mentioned, two containers are adjacent if they are located face to face. Thus given two adjacent containers A and B, the latter can be adjacent to one of the six faces of A



(a) Same orientation (b) Opposite orientation

Figure 4: Containers with the same and opposite orientation

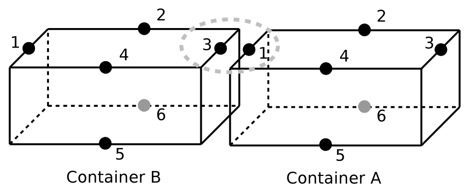


Figure 5: Rule 1 can be applied to determine the position of B if the position of A is known

(and viceversa). Since every container can have two different orientations and considering that the possible positions of B with respect to A are six, it is possible to distinguish a total of twelve forms of adjacency: six if B has the same orientation of A and six if they have opposite orientation.

Figure 4 shows two adjacent containers when they have the same and opposite orientation. In the first case, the relations of close proximity that are generated are $R_{A,B}(4,5)$, $R_{A,B}(2,6)$, $R_{B,A}(5,4)$, and $R_{B,A}(6,2)$, while in the second case are $R_{A,B}(4,6)$, $R_{A,B}(2,5)$, $R_{B,A}(6,4)$, and $R_{B,A}(5,2)$.

Thus, if the coordinates and the orientation of a container A are known, it is possible to infer the coordinates and the orientation of a container B by using the relations of close proximity included in $\mathcal{R}$. Table II contains twelve rules that can be used to compute the coordinates and the orientation of B with respect to A.

A rule can be applied if the correspondent condition is true. Figure 5 shows a case where, since $R_{A,B}(1,3) \in \mathcal{R}$, rule 1 can be used to infer the coordinates and orientation of B with respect to A. Container *B* has the same x and z coordinates of *A* and same orientation, whereas the coordinates $y(A)$ and $y(B)$ are related as follows: if $o(A) = 1$ then $y(B) = y(A) - 1$, otherwise $y(B) = y(A) + 1$ (Figure 5 depicts the case where $o(A) = 1$). These relations can be merged in the equation $y(B) = y(A) - 2\,o(A) + 1$.

Figure 6 shows another example where rule 5 can be used. This rule can be applied if at least one relation of proximity out of two (logic *OR*) is in $\mathcal{R}$ ($R_{A,B}(2,4)$ or $R_{A,B}(6,5)$). Container *B* has the same y and z coordinates of A and the same orientation, whereas $x(B) = x(A) - 2\,o(A) + 1$. In this case the system exhibits a basic form of fault tolerance: even if one of the two relations of proximity is not present in $\mathcal{R}$, because of faulting nodes or packet losses, it is still possible to determine the position of a container *B*. Similar considerations can be drawn about rules 6, 7, and 8.

In other cases, a single relation of proximity is not sufficient to resolve the possible ambiguities and to infer the position and orientation of B with respect to A. Figure 7 shows one of such cases: if $R_{A,B}(2,6) \in \mathcal{R}$ then both the solutions depicted in Figure 7 are possible. Only the presence in $\mathcal{R}$ of $R_{A,B}(4,5)$ (or its symmetrical $R_{B,A}(5,4)$) can resolve the ambiguity and, in that case, the position of B can be computed through rule 9. This also explains the presence of the logic *AND* in rule 9. Similar considerations can be made for rules 10, 11, and 12.

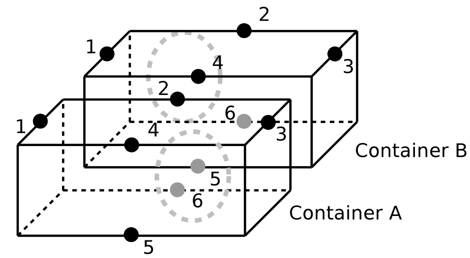


Figure 6: One of the two relations of close proximity is sufficient to determine the position and orientation of B with respect to A

III. The strawman approach

This section describes a simple approach for the localization of containers (the strawman approach) [12]. Then, in the following sections, some problems of the strawman approach are discussed and solved through Integer Linear Programming (ILP) techniques.

The localization procedure begins its execution with a container with known position and orientation. Then it finds all the adjacent containers by examining the content of $\mathcal{R}$ and uses the rules shown in Table II to compute the position of adjacent containers and their orientation. The same operations are repeated for all the containers whose position and orientation has been determined, until all the containers have been localized.

More in detail, for every container the system must store its identifier, its coordinates and its orientation. Thus, for every container A it is possible to define a tuple

$$\tau(A) := \{A, x(A), y(A), z(A), o(A)\}$$

that contains such information.

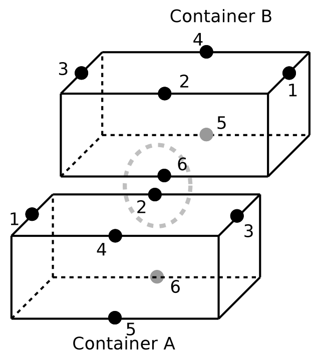
Besides $\mathcal{C}$ and $\mathcal{R}$ previously introduced, the localization procedure makes use of the following sets:

- $\mathcal{K}$: the set of known containers at the time the localization procedure is executed. It is initialized with the value of $\mathcal{C}$.
- $\mathcal{P}$: the set of tuples of containers whose position and orientation have been computed by the algorithm.
- $\mathcal{D}$: the set of tuples of containers whose neighbors still have to be discovered.

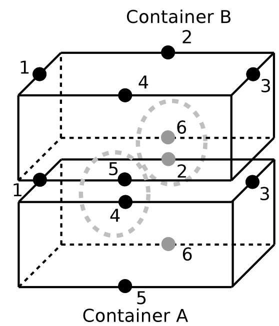
Initially, $\mathcal{P}$ and $\mathcal{D}$ are empty. The algorithm starts performing the following preliminary operations: it inserts in both $\mathcal{P}$ and $\mathcal{D}$ the tuple of a container I with known position and orientation. Then, it periodically executes the following actions: it extracts the tuple of a container A from $\mathcal{D}$ and for each container B that is contiguous to A (i.e, $R_{A,B}(i,j) \in \mathcal{R}$,

#	Condition	Orientation of B	Coordinates of B (equal to A if not specified)
1	$R_{A,B}(1,3) \in \mathcal{R}$	$o(B) = o(A)$	$y(B) = y(A) - 2\,o(A) + 1$
2	$R_{A,B}(1,1) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$y(B) = y(A) - 2\,o(A) + 1$
3	$R_{A,B}(3,1) \in \mathcal{R}$	$o(B) = o(A)$	$y(B) = y(A) + 2\,o(A) - 1$
4	$R_{A,B}(3,3) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$y(B) = y(A) + 2\,o(A) - 1$
5	$R_{A,B}(2,4) \in \mathcal{R} \vee R_{A,B}(6,5) \in \mathcal{R}$	$o(B) = o(A)$	$x(B) = x(A) - 2\,o(A) + 1$
6	$R_{A,B}(2,2) \in \mathcal{R} \vee R_{A,B}(6,6) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$x(B) = x(A) - 2\,o(A) + 1$
7	$R_{A,B}(4,2) \in \mathcal{R} \vee R_{A,B}(5,6) \in \mathcal{R}$	$o(B) = o(A)$	$x(B) = x(A) + 2\,o(A) - 1$
8	$R_{A,B}(4,4) \in \mathcal{R} \vee R_{A,B}(5,5) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$x(B) = x(A) + 2\,o(A) - 1$
9	$R_{A,B}(2,6) \in \mathcal{R} \wedge R_{A,B}(4,5) \in \mathcal{R}$	$o(B) = o(A)$	$z(B) = z(A) + 1$
10	$R_{A,B}(2,5) \in \mathcal{R} \wedge R_{A,B}(4,6) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$z(B) = z(A) + 1$
11	$R_{A,B}(5,4) \in \mathcal{R} \wedge R_{A,B}(6,2) \in \mathcal{R}$	$o(B) = o(A)$	$z(B) = z(A) - 1$
12	$R_{A,B}(5,2) \in \mathcal{R} \wedge R_{A,B}(6,4) \in \mathcal{R}$	$o(B) = 1 - o(A)$	$z(B) = z(A) - 1$

Table II: Coordinates and orientation of a container B with respect to a container A



(a)



(b)

Figure 7: A single relation of close proximity is not sufficient to infer the position and orientation of B with respect to A

for some (i, j)), it verifies if B is adjacent to A. In such case it computes the coordinates and orientation of B using the rules previously defined, $\tau(B)$ is updated and inserted into both $\mathcal{D}$ and $\mathcal{P}$ whereas B is removed from $\mathcal{K}$. If B is not adjacent to A, it is simply ignored. Once all the containers that are contiguous to A have been checked, the algorithm starts again by extracting another element of $\mathcal{D}$ and performs the same operations. The algorithm stops when $\mathcal{D} = \emptyset$ and returns the sets $\mathcal{P}$ and $\mathcal{K}$: $\mathcal{P}$ contains the identifier, position and orientation of all localized containers, whereas $\mathcal{K}$ contains the identifier of all known containers whose position and orientation cannot be computed. If there is no missing information about the set of adjacent containers under observation, because of faults or lost packets, then all the containers are localized. The pseudo-code of the algorithm is shown in Figure 8.

```
I is a container with known position and orientation
K ← C − {I}
D ← {τ(I)}
P ← {τ(I)}
for all τ(A) ∈ D do
   D ← D − {τ(A)}
   for all B ∈ K do
      for i = 1 to 12 do
         if Rule i can be applied then
            Compute coordinates and orientation of B
            P ← P ∪ {τ(B)}
            D ← D ∪ {τ(B)}
            K ← K − {B}
            break
         end if
      end for
   end for
end for
return  P, K
```

Figure 8: Pseudo-code of the localization procedure in the strawman approach.

IV. Localization by means of Integer Linear Programming

The strawman approach does not guarantee the localization of all containers in case of node faults or lost packets (in general, in case of incomplete proximity information in $\mathcal{R}$). However, the redundancy of contiguity information of a group of containers can be used to tolerate faults and localize a larger number of containers. This can be done by modeling localization as an ILP problem. The solutions of the ILP problem provide a tuple for each container which is compatible with the tuples of all the other containers and with the proximity information.

A. Variables and geometrical constraints of the ILP problem

The variables of the ILP problem are the coordinates $x(A)$, $y(A)$, $z(A)$, and the orientation $o(A)$ of each container $A \in \mathcal{C}$. In addition to the previous variables, a new variable $p(\tau(A))$

is introduced for each tuple $\tau(A)$:

$$p(\tau(A)) = \begin{cases} 1 & \text{if the coordinates and the orientation} \\ & \text{of } A \text{ are those contained in } \tau(A) \\ 0 & \text{otherwise.} \end{cases}$$

The variables x, y, z, o, p are linked by geometrical and operational constraints as follows:

$$\sum_{i=0}^{M_x-1}\sum_{j=0}^{M_y-1}\sum_{k=0}^{M_z-1}\sum_{l=0}^{1} i\, p(A,i,j,k,l) = x(A) \qquad \forall\ A \in \mathcal{C},$$

$$\sum_{i=0}^{M_x-1}\sum_{j=0}^{M_y-1}\sum_{k=0}^{M_z-1}\sum_{l=0}^{1} j\, p(A,i,j,k,l) = y(A) \qquad \forall\ A \in \mathcal{C},$$

$$\sum_{i=0}^{M_x-1}\sum_{j=0}^{M_y-1}\sum_{k=0}^{M_z-1}\sum_{l=0}^{1} k\, p(A,i,j,k,l) = z(A) \qquad \forall\ A \in \mathcal{C},$$

$$\sum_{i=0}^{M_x-1}\sum_{j=0}^{M_y-1}\sum_{k=0}^{M_z-1}\sum_{l=0}^{1} l\, p(A,i,j,k,l) = o(A) \qquad \forall\ A \in \mathcal{C}.$$

Moreover, a unique tuple is associated to each container, therefore:

$$\sum_{i=0}^{M_x-1}\sum_{j=0}^{M_y-1}\sum_{k=0}^{M_z-1}\sum_{l=0}^{1} p(A,i,j,k,l) = 1 \qquad \forall\ A \in \mathcal{C},$$

and, at the same time, each location of the yard can host at most one container:

$$\sum_{A\in\mathcal{C}}\sum_{l=0}^{1} p(A,i,j,k,l) \le 1 \qquad \forall\ (i,j,k).$$

Reduction of problem size. The size of the ILP problem can be reduced by using, as pre-processing phase, the procedure described as the strawman approach. In this way, it is possible to localize the subset of containers whose position and orientation can be unambiguously determined (coordinates and orientation are contained in the tuples of the $\mathcal{P}$ set). For every container A such that $\tau(A) \in \mathcal{P}$, the corresponding coordinates and orientation are considered as known values.

B. Additional constraints derived from the relations of close proximity

Additional constraints to the ILP problem can be inferred from $\mathcal{R}$. Rules 9–12 of Table II express the position of B with respect to A, on the base of their adjacency relation. However, such rules can be re-written to express the position of B with respect to A on the base of their contiguity relation. For example, rule 9 of Table II can be split into rules 9a and 9b of Table III, where the proximity relations are taken into account separately. Orientation is not considered in Table III as it cannot be uniquely determined. Rule 9a provides only partial information about the position of B:

$$\begin{cases} x(B) = x(A) - o(A) + o(B) \\ y(B) = y(A) \\ z(B) = z(A) + 1 \end{cases}$$

#	Condition	Coordinates of B (equal to A if not specified)
9a	$R_{A,B}(2,6) \in \mathcal{R}$	$x(B) = x(A) - o(A) + o(B)$ $z(B) = z(A) + 1$
9b	$R_{A,B}(4,5) \in \mathcal{R}$	$x(B) = x(A) + o(A) - o(B)$ $z(B) = z(A) + 1$
10a	$R_{A,B}(2,5) \in \mathcal{R}$	$x(B) = x(A) - o(A) - o(B) + 1$ $z(B) = z(A) + 1$
10b	$R_{A,B}(4,6) \in \mathcal{R}$	$x(B) = x(A) + o(A) + o(B) - 1$ $z(B) = z(A) + 1$
11a	$R_{A,B}(5,4) \in \mathcal{R}$	$x(B) = x(A) + o(A) - o(B)$ $z(B) = z(A) - 1$
11b	$R_{A,B}(6,2) \in \mathcal{R}$	$x(B) = x(A) - o(A) + o(B)$ $z(B) = z(A) - 1$
12a	$R_{A,B}(5,2) \in \mathcal{R}$	$x(B) = x(A) + o(A) + o(B) - 1$ $z(B) = z(A) - 1$
12b	$R_{A,B}(6,4) \in \mathcal{R}$	$x(B) = x(A) - o(A) - o(B) + 1$ $z(B) = z(A) - 1$

Table III: Contiguity rules for a container B with respect to a container A

since $x(B)$ is a function of the orientation of B, that is unknown and cannot be established through the single proximity relation $R_{A,B}(2,6)$. Obviously, if also rule 9b can be applied, the resulting system of equations can be solved and the result is the one expressed by rule 9.

Similar considerations can be made about rules 10, 11, and 12, that can be replaced by the couples of rules {10a, 10b}, {11a, 11b} and {12a, 12b}.

V. ILP-BASED LOCALIZATION

The localization procedure of the strawman approach returns two sets: $\mathcal{P}$, that contains the tuples of all the containers that have been successfully localized, and $\mathcal{K}$, that contains the identifiers of known containers whose position is still undetermined. If $\mathcal{K}$ is empty, all the containers have already been localized and the algorithm terminates. When the set $\mathcal{K}$ is nonempty, a larger number of containers can be localized by applying an algorithm which iteratively solves an ILP problem with the constraints described in Section IV and with a suitable objective function.

At the beginning, the algorithm finds any solution satisfying the constraints described in Section IV. Such solution gives a tuple (i.e. position and orientation) for each container of set $\mathcal{K}$. However, such a solution is not sufficient to localize the containers of $\mathcal{K}$, because there could be different solutions satisfying the constraints. On the other hand, we accept as localized a container only if it has a unique possible position and orientation. Thus, we have to verify if the obtained tuples are unique. This can be done by solving a suitable ILP problem with the same set of constraints. An objective function is added to such model whose aim is to maximize the number of containers which could have a different tuple from the initial one. If such a number is equal to zero, all the containers are localized. Otherwise, the set of containers can be divided into two subsets. Some of them keep the same tuple as in the initial solution and therefore we have to further verify the uniqueness of their tuple. Others change their tuple and therefore correspond to containers which can not be localized. The process is iterated. In the following the ILP-based algorithm reported in Figure 9 is described.

1: Consider $\mathcal{P}$ and $\mathcal{K}$ returned by the strawman approach.
2: **if** $\mathcal{K} \neq \emptyset$ **then**
3: Find $\bar{p}$ satisfying the constraints given by $\mathcal{P}$, geometrical constraints, and $\mathcal{R}$.
Let $\bar{\tau}(A)$, $A \in \mathcal{C}$ be the tuple such that $\bar{p}(\bar{\tau}(A)) = 1$
4: $\mathcal{Q} \leftarrow \mathcal{K}$
5: **while** $\mathcal{Q} \neq \emptyset$ **do**
6: Find p_{opt} that solves the following ILP problem:

$$v = \min \sum_{A \in \mathcal{Q}} p(\bar{\tau}(A))$$

subject to the constraints given by $\mathcal{P}$, geometrical constraints, and $\mathcal{R}$.
Let $\{\tau_{opt}(A),\ A \in \mathcal{C}\}$ be the set of tuples such that $p_{opt}(\tau_{opt}(A)) = 1$
7: **if** $v = |\mathcal{Q}|$ **then**
8: break
9: **else**
10: **for all** $A \in \mathcal{Q}$ **do**
11: **if** $\tau_{opt}(A) \neq \bar{\tau}(A)$ **then**
12: $\mathcal{Q} \leftarrow \mathcal{Q} - \{A\}$
13: **end if**
14: **end for**
15: **end if**
16: **end while**
17: **for all** $A \in \mathcal{Q}$ **do**
18: $\mathcal{P} \leftarrow \mathcal{P} \cup \{\bar{\tau}(A)\}$
19: $\mathcal{K} \leftarrow \mathcal{K} - \{A\}$
20: **end for**
21: **end if**
22: **return** $\mathcal{P}$, $\mathcal{K}$

Figure 9: Pseudo-code of the ILP-based algorithm

The content of $\mathcal{P}$ is used to assign a value to the variables corresponding to the containers already localized, and a feasible layout for all the containers is computed (see line 3). In other words, the algorithm finds suitable initial values $\bar{p}$, for the variables p, so that the constraints given by $\mathcal{P}$, the proximity relations contained in $\mathcal{R}$, and the geometrical constraints are satisfied. Moreover for each container $A \in \mathcal{C}$ we denote by $\bar{\tau}(A)$ the tuple such that $\bar{p}(\bar{\tau}(A)) = 1$. Then, an auxiliary set $\mathcal{Q}$ is initialized with the value of $\mathcal{K}$ (line 4). The set $\mathcal{Q}$ represents the set of containers for which the uniqueness of their tuples must be verified. Among all the feasible layouts of containers, the ILP-based algorithm looks for the one where the number of containers whose tuple is equal to the initial one is minimum (which corresponds to maximize the number of containers which have a different tuple from the initial one). In other words, the following ILP problem is solved:

$$\min \sum_{A \in \mathcal{Q}} p(\bar{\tau}(A)),$$

subject to the constraints given by $\mathcal{P}$, the geometrical constraints, and the relations of close proximity (line 6). The value of the objective function represents the total number of containers which have the same tuple as the initial solution. If this minimum value is equal to the cardinality $|\mathcal{Q}|$ of the set $\mathcal{Q}$, then no container in $\mathcal{Q}$ can have a tuple that is different from the initial one, i.e. all the containers in $\mathcal{Q}$ are unambiguously localized (lines 7-8). Otherwise, each container A having a tuple $\tau_{opt}(A)$ that is different from the initial one $\bar{\tau}(A)$ cannot be localized; hence the set $\mathcal{Q}$ is updated by removing such containers (lines 10-14) and the ILP problem is solved again. When the procedure completes, the set $\mathcal{Q}$ contains all the containers which can be localized and their tuples are the ones specified by the initial solution; the containers in $\mathcal{Q}$ are removed from $\mathcal{K}$ and their tuples are inserted in $\mathcal{P}$ (lines 17-20). The ILP-based algorithm returns the set $\mathcal{P}$ of tuples of all the containers which can be localized, and the set $\mathcal{K}$ which contains the containers which cannot be localized given the partial information (line 22). It is worth noting that the procedure always provides the tuples of all containers which can be unambiguously localized based on the available information.

As an example of the proposed methodology, let six containers be placed as depicted in Figure 10a, while Figure 10b shows the X-Z plane and the working nodes. Figure 10c, for the sake of clarity, represents the same information in a different way: containers are depicted as circles and the relations of close proximity are depicted as dashed lines.

It is assumed that A is a container with known position and orientation. The localization procedure of the strawman approach is not able to localize other containers, thus produces only the following result:

$$\mathcal{P} = \{(A, 0, 0, 0, 1)\}.$$

Containers in the set $\mathcal{K} = \{B, C, D, E, F\}$ are not localized because no one of the rules of Table II can be applied. Subsequently, the ILP-based algorithm starts with a feasible layout $\bar{\tau}$ of the containers as the one shown in the figure and sets $\mathcal{Q} = \mathcal{K}$. Then, the ILP problem is solved and the solution is $v = 3 < |\mathcal{Q}|$ with

$\tau_{opt}(B) = \{(B, 2, 0, 0, 0)\} \neq \bar{\tau}(B)$
$\tau_{opt}(C) = \{(C, 1, 0, 0, 0)\} \neq \bar{\tau}(C)$
$\tau_{opt}(D) = \bar{\tau}(D)$
$\tau_{opt}(E) = \bar{\tau}(E)$
$\tau_{opt}(F) = \bar{\tau}(F)$.

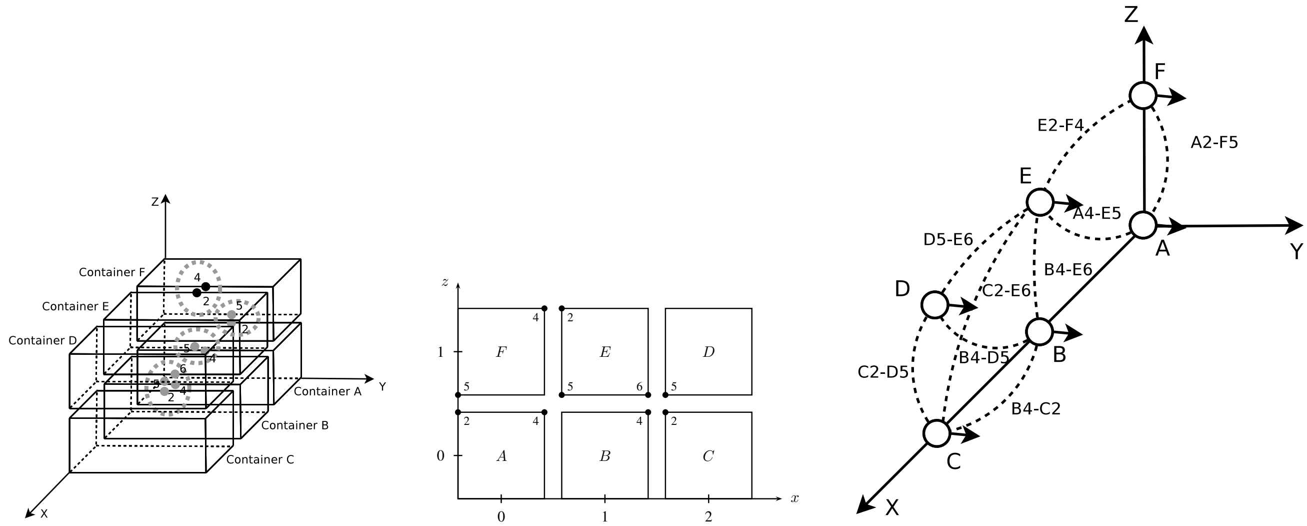
The ILP problem is solved again with $\mathcal{Q} = \{D, E, F\}$ and in this case $v = 3 = |\mathcal{Q}|$, so the algorithm terminates. The result of the algorithm is

$\mathcal{P} = \{(A, 0, 0, 0, 1), (D, 2, 0, 1, 1),$
$(E, 1, 0, 1, 1), (F, 0, 0, 1, 1)\}$,
$\mathcal{K} = \{B, C\}$.

In the end, the ILP-based algorithm increases the number of localized containers as it is able to determine also the position of containers D, E, F. However, it is not able to localize B and C because of the limited number of relations of close proximity (both $\{(B, 1, 0, 0, 1), (C, 2, 0, 0, 1)\}$ and $\{(B, 2, 0, 0, 0), (C, 1, 0, 0, 0)\}$ are possible solutions).

VI. Simulation and Results

The effectiveness of the ILP-based algorithm has been evaluated by means of simulations. The numerical results have been obtained applying the previously described model and



(a) Six containers

(b) X-Z plane

(c) Schematic representation: containers are depicted as circles with an arrow that specifies their orientation, relations of close proximity are depicted as dashed lines

Figure 10: Example of ILP-based localization

considering different levels of node faults. The ILP model has been implemented using the AMPL 8.1 modeling language [13] and solved using the CPLEX 9.1.0 solver [14] on a normal PC. The CPLEX solver has also been used to compute the initial solution ($\bar{p}$ and $\bar{\tau}$). The experiments have been carried out considering two different scenarios.

In the first scenario, the containers have been disposed in a group with a box-like shape, with different size: $4 \times 4 \times 4$, $5 \times 5 \times 5$, $10 \times 2 \times 6$, $2 \times 10 \times 6$, $10 \times 1 \times 6$, and $1 \times 10 \times 6$. Moreover, the percentage of faulty nodes have been varied between 15% and 40%. For each disposition and for each percentage of faults, the algorithm has been run on ten different instances randomly generated. The results of the tests are shown on Table IV. The first column reports the size of the group, the remaining columns report the average percentage of localized containers depending on the percentage of faulty nodes.

In the second scenario, the space that contains the containers is not completely full. More precisely, the group has been organized as follows: a part of the inner volume with size $I_x \times I_y \times I_z$ is assumed to be completely full of containers, then the remaining part of the volume up to size $V_x \times V_y \times V_z$, contains a number of containers placed randomly. To test the algorithm in this scenario, different configurations of $V_x \times V_y \times V_z$, have been considered: $5 \times 5 \times 5$ (with inner volume $3 \times 3 \times 3$), $10 \times 2 \times 6$ (with inner volume $5 \times 2 \times 3$), $10 \times 2 \times 6$ (with inner volume $10 \times 2 \times 2$), $2 \times 10 \times 6$ (with inner volume $2 \times 5 \times 3$), and $2 \times 10 \times 6$ (with inner volume $2 \times 10 \times 2$). The total number of containers in the group has been randomly varied to be in the interval $125 - 150\%$ of the number of containers contained in the inner volume. Also in this case, for each configuration and for each percentage of faults, the algorithm has been executed on ten different instances randomly generated. The results of the tests are shown on Table V.

It is worth noting that, for a given percentage of faults, the average number of localized containers decreases when the configuration is less compact. This is reasonable because if the containers are less compactly disposed, the redundancy of relations of close proximity that are correctly detected is reduced as well. For each test the run time was of about few seconds. With respect to the strawman approach, the ILP-based algorithm increases the number of localized container up to 18.5 times in the first scenario and up to 6.7 times in the second scenario.

Figure 11 shows a summary of the results for the two scenarios. The curves depict the ratio between the number of containers localized using the ILP-based approach and the number of containers localized using the strawman approach, averaged over all the different configurations. In other words, Figure 11a and 11b show the average gain obtained through the ILP-based approach with respect to the strawman approach (a value equal to 1 means that there is no gain). In both scenarios the gain is small when the percentage of faults is small (in such situations, also the strawman approach is able to localize almost all the containers), but it becomes very relevant when the number of faults increases. This is particularly evident for the first scenario where containers are disposed in a more compact way. In few cases, even if the gain with respect to the strawman approach is large, the performance of the ILP-based approach could be considered not very satisfactory because the absolute percentage of localized containers is not close to 100%. Nevertheless, it is important to notice that the ILP-based algorithm provides the optimal result, compatibly with the set of geometrical constraints previously introduced (better solutions can be achieved only changing the set of constraints, e.g. increasing the number of nodes attached to a container).

Table IV: Average percentage of localized containers (first scenario)

(a) Strawman approach

group size	faulty nodes (%) 15	20	25	30	35	40
4x4x4	99.7	98.9	85.8	64.4	41.3	29.7
5x5x5	99.8	89.2	96.6	81.2	57.0	21.6
10x2x6	99.6	98.7	96.3	38.3	40.3	6.9
2x10x6	89.1	87.3	65.5	33.8	6.8	8.4
10x1x6	97.7	94.5	46.7	28.2	10.0	8.0
1x10x6	68.5	24.2	12.0	6.7	3.5	2.8

(b) ILP-based algorithm

group size	faulty nodes (%) 15	20	25	30	35	40
4x4x4	100.0	100.0	99.7	98.9	97.8	90.3
5x5x5	100.0	100.0	99.4	98.1	95.5	95.1
10x2x6	100.0	99.8	99.7	98.5	95.8	91.6
2x10x6	100.0	99.3	99.0	98.1	95.1	85.9
10x1x6	100.0	99.3	98.0	93.5	88.8	80.5
1x10x6	100.0	99.3	95.7	89.3	54.2	52.5

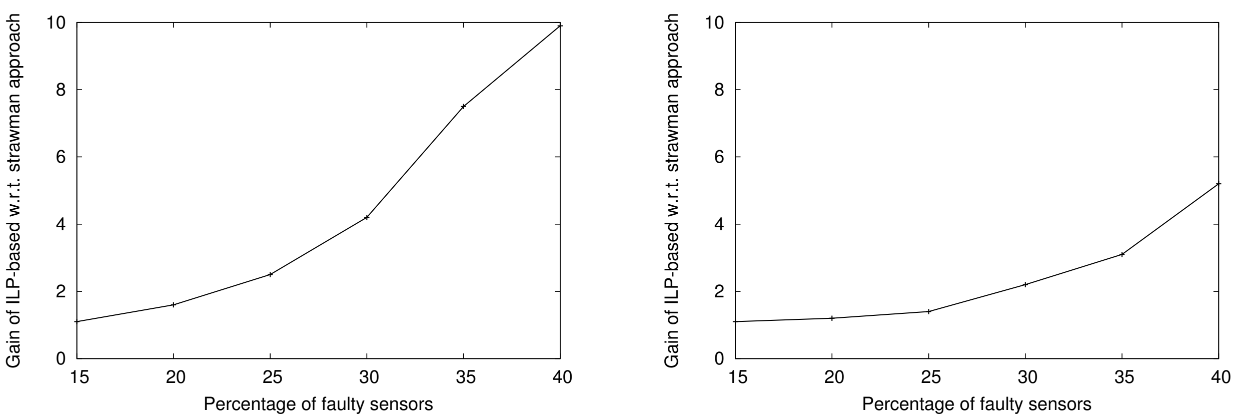
Table V: Average percentage of localized containers (second scenario)

(a) Strawman approach

inner vol.	outer vol.	faulty nodes (%) 15	20	25	30	35	40
3x3x3	5x5x5	96.2	75.7	81.9	68.6	27.0	18.8
5x2x3	10x2x6	84.2	91.3	69.4	59.4	41.3	12.6
10x2x2	10x2x6	89.5	75.6	73.2	54.3	21.5	13.6
2x5x3	2x10x6	73.3	74.1	52.8	42.5	15.6	8.4
2x10x2	2x10x6	91.8	84.0	49.8	17.3	21.0	7.7

(b) ILP-based algorithm

inner vol.	outer vol.	faulty nodes (%) 15	20	25	30	35	40
3x3x3	5x5x5	99.1	97.2	92.9	92.4	88.6	64.1
5x2x3	10x2x6	95.6	95.4	92.6	91.4	80.6	81.0
10x2x2	10x2x6	96.8	95.3	88.5	79.7	74.0	60.4
2x5x3	2x10x6	97.2	96.2	81.3	77.5	55.6	56.4
2x10x2	2x10x6	96.2	89.7	90.2	86.9	67.5	37.9



(a) First scenario

(b) Second scenario

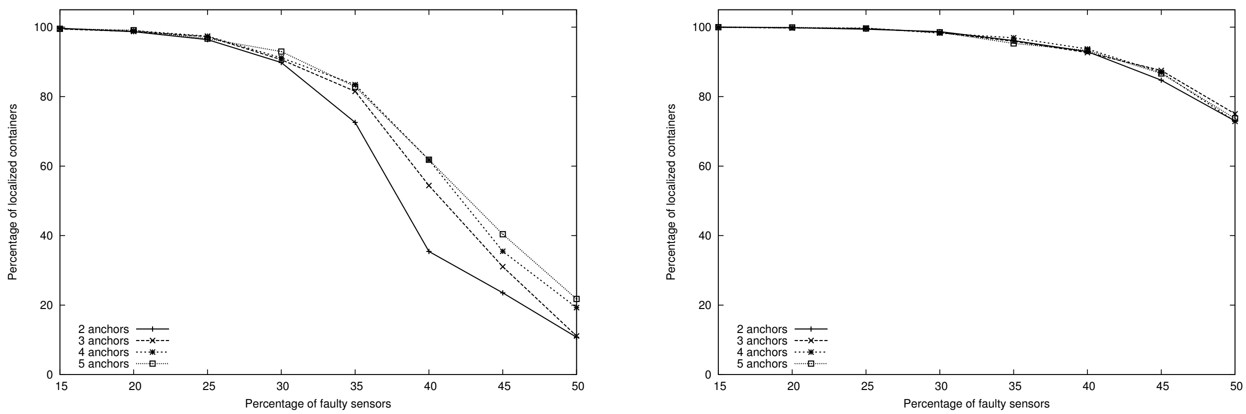
Figure 11: Average gain in localization using the ILP-based approach w.r.t. the strawman approach

A. Varying the number of containers with known position

Further experiments have been performed to study the effects of varying the number of containers with known position. Such containers will be called *anchor containers* from now on. Experiments have been carried out on a subset of the configurations previously presented where the number of anchor container has been varied between 2 and 5. Considering that increasing the number of anchor containers allows better localization, the percentage of faults has been pushed up to 50%. The results are the average values obtained for ten random instances, in terms of container placement and position of anchors.

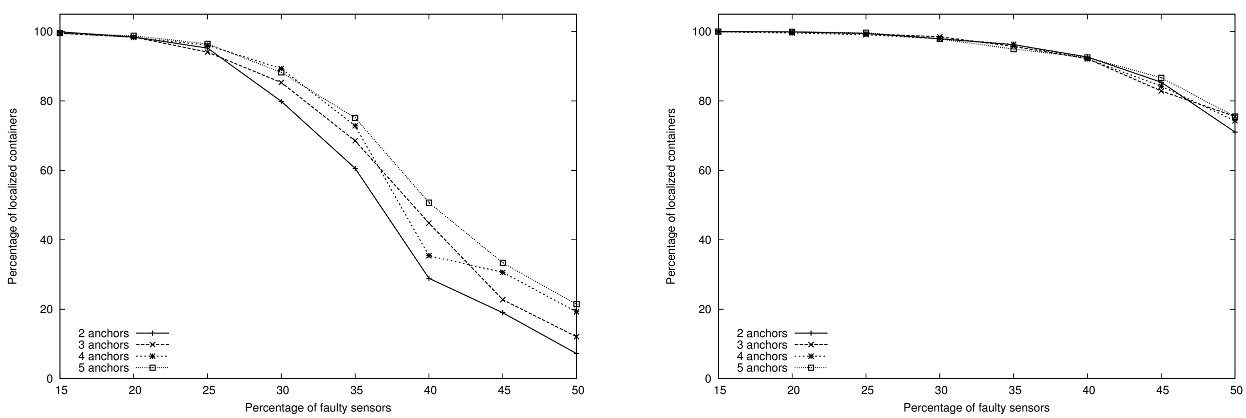
In the first scenario the containers were placed in a box-like shape of size $5 \times 5 \times 5$ and $10 \times 2 \times 6$. Figures 12 and 13 show the fault tolerance for the strawman approach (on the left) and the ILP-based algorithm (on the right). Note that in both figures the strawman approach is influenced by the number of anchor containers. This because an increase of the number of anchor containers contributes to fill in the missing information and therefore localization improves. On the other hand, the ILP-based algorithm is able to localize a greater number of containers and the number of anchor containers does not affect significantly the localization procedure. This is due to the fact that in this scenario the containers are compactly disposed and the geometrical constraints suffice for the requirements of the ILP-based algorithm.

In the second scenario two configurations have been considered: $5 \times 5 \times 5$ (with inner volume $3 \times 3 \times 3$) and $10 \times 2 \times 6$ (with inner volume $10 \times 2 \times 2$). Figures 14 and 15 show the two configurations for the strawman approach (on the left) and the ILP-based algorithm (on the right). Both suffer a decrease of localization performance with respect to the first scenario. This is caused by the sparse placement of the containers that is translated in a lower number of adjacencies. This is more evident, for example, in a very sparse configuration such as $10 \times 2 \times 6$. Note also that the lack of information of the second scenario is counterbalanced by the increasing number of anchor containers which, this time, influences also the



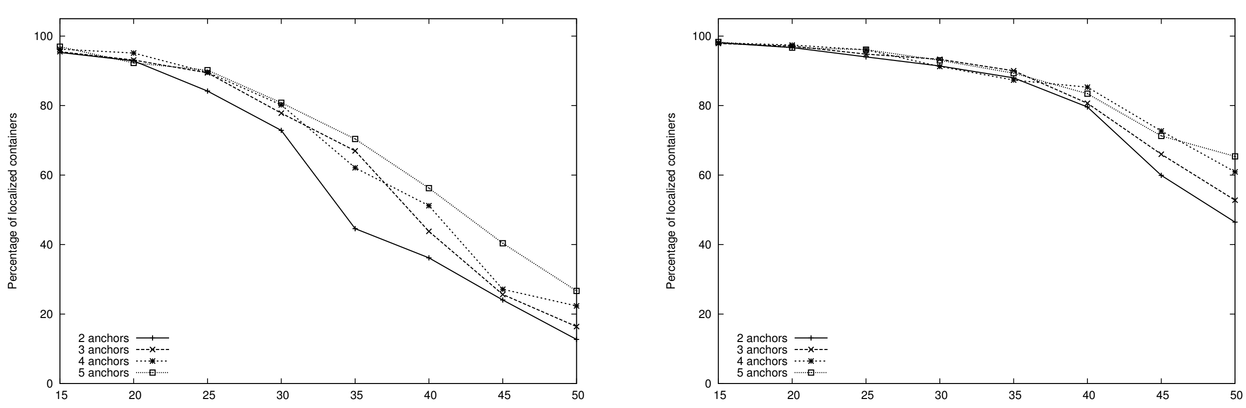
(a) Strawman approach (b) ILP-based algorithm

Figure 12: Comparison between the strawman approach and ILP-based algorithm (first scenario, $5 \times 5 \times 5$)



(a) Strawman approach (b) ILP-based algorithm

Figure 13: Comparison between the strawman approach and ILP-based algorithm (first scenario, $10 \times 2 \times 6$)



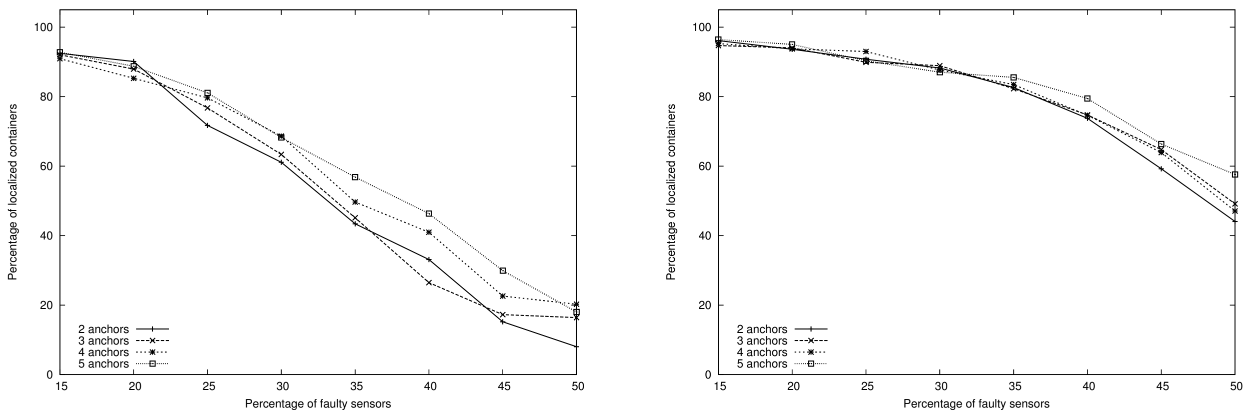
(a) Strawman approach (b) ILP-based algorithm

Figure 14: Comparison between the strawman approach and ILP-based algorithm (second scenario, $5 \times 5 \times 5$ with inner volume $3 \times 3 \times 3$)

performance of the ILP-based algorithm.

VII. Implementation and prototyping

A completely working prototype of the system has been built using the *Tmote Sky* nodes commonly available for the



(a) Strawman approach

(b) ILP-based algorithm

Figure 15: Comparison between the strawman approach and ILP-based algorithm (second scenario, $10 \times 2 \times 6$ with inner volume $10 \times 2 \times 2$)

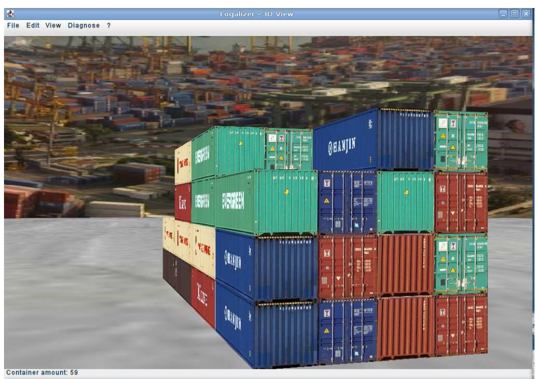


Figure 16: The graphical user interface

realization of wireless sensor networks. A wireless sensor network (WSN) is a wireless network composed of a large number of distributed autonomous sensors capable not only of measuring real world phenomena but also filtering, sharing, combining and aggregating such readings [15]. Each node of the network is equipped with a radio transceiver, a microcontroller, one or more sensing devices, and is powered by batteries. The nodes organize themselves in a wireless ad-hoc network: each node supports a multi-hop routing algorithm that allows it to forward data packets to a sink node directly connected to a base station. All these features eased the production of the prototype. Considering the aim of the localization system, the sensing features of the devices have not been used. However, it should be noted that the localization system can be easily extended, with the proper sensing equipment, to automate some existing procedures. For example, containers could be sensed to measure the level of CO_2 (to detect hidden people), or to continuously monitor the temperature of refrigerators. The application that is executed on the nodes is written in nesC [16], while the operating system is TinyOS [17].

The software executed on the base station, which processes the data coming from the WSN and which determines the position of containers, has been implemented in Java. To verify its outcome, the program has been interfaced with a GUI that provides a visual representation of the yard. The GUI interface (shown in Figure 16) allows the user to easily locate the position of each container. It is possible to interact with the interface to manipulate the view (rotation, zoom in and out, change from textured mode to wire-frame and vice versa). The user can also select one of the containers with the mouse pointer to retrieve its specific information, or he can search for a given container by specifying its ID. The selected container is then displayed in texture mode, while the other containers are switched in wire-frame mode (this is useful to find containers that are completely hidden by others). The user can also move within the virtual environment.

VIII. Related work

Research about smart containers gained momentum in the last years, pushed not only by recent advances in emerging sensor technologies and miniaturization, but also by governative initiatives and regulations (such as the Advanced Container Security Device program or the Marine Asset Tag Tracking program of the Department of Homeland Security of the USA [18]). In particular, radio frequency identification and wireless sensor network have been the primary technological solutions used to explore new research directions, such as enhancement of security and intrusion detection [19], [20], [21], and detection of damages to goods [22].

A. *Localization of shipping containers*

A paper that explicitly deals with the problem of localizing containers in a harbor is [23]. The authors describe a system, VAPS, that takes into account the physical characteristics of large objects as a way to define constraints useful for the purpose of localization: $i)$ the metallic surface and the grid-like arrangement of containers cause a waveguide effect along some directions and a blocking effect along other directions; $ii)$ objects are not dimensionless and cannot overlap. In VAPS, containers are equipped with two wireless devices (for the two horizontal axes), while communication between the devices located on the same container is achieved through wires. Each device is able to distinguish a (small) number of different RSS levels. Simulative results show that VAPS performs better than two competitors: an RSS-based method using a open-space propagation model and a hop-count based method. In the end, VAPS confirms the importance of the problem of container localization and the use of geometrical constraints as an effective technique over geometrically blind approaches. With respect to the technique described in this paper, in [23] the analysis is limited to a bi-dimensional scenario and the presence of faults is not considered.

The problem of automatically identify and locate containers in the yard has been faced also in [24]. The proposed system, MOCONT, relies on GPS positioning acquired by reach-stackers that communicate the position of containers to a base station each time they are moved. The system also includes an inertial navigation system that, by using accelerometers, gyroscopes, and ground speed sensors, provides positioning information when the GPS system cannot operate (for example when the satellite signal is shielded by high container stacks). Container identification is performed through digital image analysis techniques.

Tracking of container position on a large scale can be also achieved using non-GPS technologies. In [25], the authors propose a tracking system that is based on the analysis of FM broadcast signal: a low-cost and low-power FM receiver, attached to containers, records the frequency spectrum and compare it to known data to determine the path of the container. The purpose of this approach is to overcome some of the limitations of GPS-based devices such as high power consumption, cost and the need of line-of-sight with satellites. However, the position is determined with a rather large error level (in the order of kilometers) and, in any case, its purpose is different from the approach proposed in this paper, as it is aimed at tracking the movement of containers during ground transportation by means of trucks or trains.

The adoption of WSNs for container tracking and monitoring has been discussed also in [26]. The authors propose an architecture where sensor nodes are placed both inside and outside containers. The internal nodes are used to monitor the status of goods or to recognize some possible dangers (fire, water, etc.). Each container is also equipped with an external node, called container monitor, that is responsible of collecting the data coming from internal nodes and communicating with other monitors. Container monitors are supposed to have global connectivity, through GSM links, and be equipped with a GPS receiver. The architecture also includes the presence of a special node, with an unlimited power supply, that can be used to reduce the energy spent for communication by container monitors.

In [27] the authors designed a system to identify and localize containers. The system is based on a tablet PC equipped with a camera, a GPS unit and a digital compass. Image processing techniques are used to recognize the containers pointed by the camera, and an extended Kalman filter is used to fuse the data coming from the two sensors. The device is used by an operator that manually has to move within the yard and communicates with a database by means of a wireless connection. Thus, the purpose of the system is somehow different from the one proposed in this paper that, as mentioned, aims at achieving automatic and and continuous monitoring of container position.

B. Use of WSNs with shipping containers

WSNs have been studied, for their sensing capability, as a way to gather information related to the status of goods inside containers. In [28], the authors focus on the integration of WSNs for monitoring of goods with the enterprise systems that are used by the different partners that belong to the supply chain. The paper describes the design of a middleware system that can accomplish this task under the assumption that a localization system is available.

Security of container transportation is another important research topic. Container's position, provenience and destination can be used to program targeted inspections and discover security threats more effectively. IBM's Secure Trade Lane [6] introduces a solution to make container shipment more predictable and more secure. An embedded wireless device called TREC (Tamper Resistant Embedded Controller) is applied to the door of a container; built-in sensors detect the door's opening, temperature, humidity, acceleration; localization is GPS-based. An advanced back-end system analyzes and processes data acquired by TREC using different technologies such as GSM/GPRS, satellite or short range wireless communication based on IEEE 802.15.4 (useful for extending coverage in a busy container yard or when containers are stacked in a vessel). Back-end processing basically translates such data into business information that is accessible by accredited supply chain participants. TRECs are able to store data in the case they cannot communicate with the back-end, and to signal predefined alarms. The configuration of TRECs is done by means of handheld devices (that can be also used for reading data).

MASC [8] (Monitoring and Security of Containers) proposes the use of smart containers that allow officers at ports of arrival to detect whether a container has been tampered during the transport. Each container is instrumented with MASC units, composed by an antenna placed outside the container and connected to the sensors that are located inside the container. Like the IBM solution, MASC units have autonomous logging and signaling capabilities. The discussion is mostly focused on the MASC architecture, based on trusted third parties that control access to data collected by sensor nodes; the use of a tree structure to represent the supply chain, allows the root (a forwarding agency, holding the overall responsibility), to assign parts of the containers transport to lower service providers.

Another approach, useful for easing the enterprise-level integration, is proposed in [29]. Here, Web services facilitate the communication between containers and involved parties. The authors assume that containers have a set of sensors able to monitor the position, temperature, pressure, and other physical characteristics of containers. The usefulness and the need of an autonomous localizing system is emphasized.

C. Localization of nodes in a WSN

The localization of sensor nodes is a general problem of WSNs, as for many applications sensed data are meaningless if their origin is unknown. Some approaches are range-based, like time-of-arrival, angle-of-arrival, and received-signal-strength (described, for example, in [30]), while others are range-free (as the ones described in [31]). The reader is also forwarded to [32] for a survey of localization techniques under limited measurement capabilities.

A technique partially similar to the one we adopted is described in [33]. This algorithm, called ROCRSSI (Ring Overlapping based on Comparison of Received Signal Strength Indicator), requires that every node having known position (*anchor* node) periodically broadcasts an array containing the RSS of packets received from other nodes. Nodes having unknown position (*strayed* nodes) collect these vectors and trigger the localization process. Comparing the signal strength of transmissions coming from anchor nodes, a strayed node determines a ring (with a radius that is proportional to the detected power) centered in one of the anchor nodes. The procedure is repeated for every other anchor and the node is finally located in the middle of the region where the largest number of rings intersect.

In [34], the authors propose a localization technique that uses connectivity information to derive the position of WSN nodes. The approach, which is based on multidimensional scaling (a data analysis technique), is able to take advantage of connectivity distance between all the nodes that still have to be localized (instead of localizing the nodes one by one as done in other approaches). The use of discrete values (the number of hops) to express the distance between nodes is similar to the idea of using binary proximity information proposed in this paper. Nevertheless, the use of geometrical constraints as a way to improve the performance of the system is not considered (the problem of localization is discussed from a rather general point of view, and it is not contextualized in a specific scenario).

A localization scheme that makes use of additional knowledge derived from the placement strategy is described in [35]. Nodes are supposed to be placed according to a grid topology and the distance from anchor nodes is used to select the position of the grid with maximum probability. The presence of constraints increases the localization accuracy of the system with respect to similar approaches. This confirms that the idea of incorporating geometrical information into localization problems is successful.

The technique proposed in this paper, that is neither completely range-based nor completely range-free, contributes to the body of existing literature by showing that the use of geometrical constraints can be effective for the purposes of a localization system. Moreover, all this research work highlights the need for advanced solutions in the area of container logistics and that many currently applied processes, such as transport and monitoring of containers, could benefit from the adoption of WSN-based systems, where the technique proposed in this paper could fulfill the localization needs.

IX. Conclusion

In this paper, a non-conventional system for the localization of containers in the yard of ports and terminals has been presented. This localization solution represents an alternative or an integration with respect to the currently available systems, that are based on GPS and RFID technologies. In particular, the use of a wireless sensor network overcomes some problems that may arise from the use of these two technologies: first the need to guarantee a line of sight towards satellites, that limits the use of GPS systems only to outdoor environments or to the containers positioned on the surface of a stack, second the need of an infrastructure and an explicit action for reading RFID tags. Moreover, for both technologies real-time localization of containers is not possible.

The proposed localization system is characterized by high scalability. In fact, when the number of container increases the amount of signaling traffic generated by a single container is not subject to changes, since the maximum number of adjacent container is still equal to six. Obviously, because of data collection, the traffic injected into the network increases linearly with the number of nodes, since they all produce their NCP packets. However, it is important to consider that the movement rate of containers is generally low, and that it is not needed to have an extremely fast reaction time. Thus, the system can be easily tuned to tolerate the size of stacks that are typically found in real scenarios.

However, besides the practical relevance of the implemented system, the main contributions of this paper are the following. First, the use of geometrical constraints as a way to reduce the space of possible solutions of a localization problem. As known, localization techniques based on the strength of the received signal are characterized by high error levels. The discretization of node positions makes the localization process simple and scarcely sensible to RSS errors. Second, the idea of modeling the localization problem as a ILP problem where the geometrical constraints can be easily represented and managed. The resulting ILP problem can be solved by using standard procedures and proves to be resilient to a large number of faults: in the two considered scenarios, the overall localization rate is increased by 4.45 and 2.4 times with respect to the strawman approach (average values over all configurations and percentages of faulting nodes). It also reasonable to believe that these techniques can be successfully applied to other localization scenarios characterized by geometrical constraints or to make existing techniques more tolerant to faults.

References

[1] C.-I. Liu, H. Jula, and P. Ioannou, "Design, simulation, and evaluation of automated container terminals," *IEEE Trans. on Intell. Transp. Syst.*, vol. 3, no. 1, pp. 12 –26, Mar. 2002.

[2] P. Chen, Z. Fu, A. Lim, and B. Rodrigues, "Port yard storage optimization," *IEEE Trans. Autom. Sci. Eng.*, vol. 1, no. 1, pp. 26 – 37, Jul. 2004.

[3] Z. He, J. Liu, H. Ma, and P. Li, "A new automatic extraction method of container identity codes," *IEEE Trans. on Intell. Transp. Syst.*, vol. 6, no. 1, pp. 72 – 78, Mar. 2005.

[4] A. Alessandri, C. Cervellera, M. Cuneo, M. Gaggero, and G. Soncin, "Modeling and feedback control for resource allocation and performance analysis in container terminals," *IEEE Trans. on Intell. Transp. Syst.*, vol. 9, no. 4, pp. 601 –614, Dec. 2008.

[5] H. Durrant-Whyte, D. Pagac, B. Rogers, M. Stevens, and G. Nelmes, “Field and service applications - an autonomous straddle carrier for movement of shipping containers - from research to operational autonomous systems,” *IEEE Robot. Autom. Mag.*, vol. 14, no. 3, pp. 14 –23, Sep. 2007.

[6] S. Schaefer, “Secure trade lane: a sensor network solution for more predictable and more secure container shipments,” in *Companion to the 21st ACM SIGPLAN symposium on Object-oriented programming systems, languages, and applications*. New York, NY, USA: ACM, 2006, pp. 839–845.

[7] W. Lang, R. Jedermann, D. Mrugala, A. Jabbari, B. Krieg-Brü andckner, and K. Schill, “The “Intelligent Container” – A Cognitive Sensor Network for Transport Management,” *IEEE Sensors J.*, vol. 11, no. 3, pp. 688 –698, Mar. 2011.

[8] J. O. Lauf and D. Gollmann, “Monitoring and security of container transport,” in *In Proc. of New Technologies, Mobility and Security*. Springer, 2007.

[9] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, “Experimental comparison of rssi-based localization algorithms for indoor wireless sensor networks,” in *Proc. of the workshop on real-world wireless sensor networks (REALWSN)*. New York, NY, USA: ACM, 2008, pp. 1–5.

[10] E. Elnahrawy, X. Li, and R. Martin, “The limits of localization using signal strength: a comparative study,” in *Proc. of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, Oct. 2004, pp. 406 – 414.

[11] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325–349, 2005.

[12] S. Abbate, M. Avvenuti, P. Corsini, and A. Vecchio, “Localization of shipping containers in ports and terminals using wireless sensor networks,” in *Proc. of the International Conference on Computational Science and Engineering*, vol. 2, Aug. 2009, pp. 587 –592.

[13] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, Nov. 2002. [Online]. Available: http://www.ampl.com

[14] *ILOG CPLEX 9.1 User’s Manual*. ILOG Inc., 2005. [Online]. Available: http://www.cplex.com

[15] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Computer Networks*, vol. 38, pp. 393–422, 2002.

[16] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, “The nesC language: a holistic approach to networked embedded systems,” *SIGPLAN Notices*, vol. 38, no. 5, pp. 1–11, 2003.

[17] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors,” *SIGPLAN Notices*, vol. 35, no. 11, pp. 93–104, 2000.

[18] D. of Homeland Security, `http://www.dhs.gov`.

[19] R. Craddock and E. Stansfield, “Sensor fusion for smart containers,” in *The IEE Seminar on Signal Processing Solutions for Homeland Security*, Oct. 2005, p. 12 pp.

[20] S. J. Kim, G. Deng, S. Gupta, and M. Murphy-Hoye, “Intelligent networked containers for enhancing global supply chain security and enabling new commercial value,” in *Proc. of the 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE)*, Jan. 2008, pp. 662 –669.

[21] J. Bian, R. Seker, S. Ramaswamy, and N. Yilmazer, “Container communities: anti-tampering wireless sensor network for global cargo security,” in *Proc. of the 17th Mediterranean Conference on Control and Automation*, Jun. 2009, pp. 464–468.

[22] S. Bukkapatnam, S. Mukkamala, J. Kunthong, V. Sarangan, and R. Komanduri, “Real-time monitoring of container stability loss using wireless vibration sensor tags,” in *Proc. of the IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2009, pp. 221 –226.

[23] Y. Jarma, G. Karbaschi, M. de Amorim, F. Benbadis, and G. Chelius, “Vaps: Positioning with spatial constraints,” in *Proc. of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops (WoWMoM)*, Jun. 2009, pp. 1–8.

[24] V. Recagno, A. Derito, and R. Nurchi, “Mocont: a new system for automatic identification and location of containers,” in *Proc. of the Vehicular Technology Conference*, vol. 4, 2001, pp. 2609 –2613 vol.4.

[25] R. Schmidt, A. Gagnon, and D. Allcock, “Maritime containers tracking trial results,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 24, no. 9, pp. 10–14, Sep. 2009.

[26] S. Mahlknecht and S. Madani, “On architecture of low power wireless sensor networks for container tracking and monitoring applications,” in *Proc. of the 5th IEEE International Conference on Industrial Informatics*, vol. 1, Jun. 2007, pp. 353 – 358.

[27] E. Herrera, M. Ribo, A. Pinz, R. Quiros, and G. Fabregat, “Wearable system for the localization of shipping containers in a maritime terminal,” in *Proc. of the IEEE Instrumentation and Measurement Technology Conference (IMTC)*, Apr. 2006, pp. 611–616.

[28] K. Thoelen, S. Michiels, and W. Joosen, “Tracking and tracing containers through distributed sensor middleware,” in *Proc. of the 2nd International Conference on Autonomic Computing and Communication Systems*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008, pp. 1–7.

[29] L. Baresi, D. Braga, M. Comuzzi, F. Pacifici, and P. Plebani, “A service-based infrastructure for advanced logistics,” in *Proc. of the 2nd International Workshop on Service Oriented Software Engineering*. New York, NY, USA: ACM, 2007, pp. 47–53.

[30] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, “Locating the nodes: cooperative localization in wireless sensor networks,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, 2005. [Online]. Available: http://dx.doi.org/10.1109/MSP.2005.1458287

[31] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *Proc. of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)*. New York, NY, USA: ACM, 2003, pp. 81–95.

[32] C. Wang and L. Xiao, “Sensor localization under limited measurement capabilities,” *IEEE Network*, vol. 21, no. 3, pp. 16–23, May-Jun. 2007.

[33] C. Liu, T. Scott, K. Wu, and D. Hoffman, “Range free sensor localization with ring overlapping based on comparison of received signal strength indicator,” *Int. J. Sen. Netw.*, vol. 2, no. 5/6, pp. 399–413, 2007.

[34] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz, “Localization from connectivity in sensor networks,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 961 – 974, Nov. 2004.

[35] R. Stoleru and J. Stankovic, “Probability grid: a location estimation scheme for wireless sensor networks,” in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, Oct. 2004, pp. 430 – 438.

PLACE PHOTO HERE

Stefano Abbate received his Bachelors and Master’s degrees in Computer Engineering from the University of Pisa respectively in 2005 and 2008. Currently he is a Ph.D. candidate in Computer Science and Engineering (XXIV Ph.D. Program) at the IMT Institute for Advanced Studies Lucca, Italy. His areas of research include localization techniques and designing pervasive systems based on wireless sensor networks. He is a student member of IEEE and IEEE CS.

PLACE PHOTO HERE

Marco Avvenuti is an Associate Professor with the Information of Information Engineering of the University of Pisa, Italy. He received the Laurea in Electronic Engineering (1989) and the PhD in Information Engineering from the University of Pisa (1993). In 1992 and 1996 he was a visiting scholar at the Berkeley’s International Computer Science Institute (ICSI). He is a member of the IEEE Computer Society. His current research interests are in the areas of mobile and pervasive computing and wireless sensor networks. He has been involved in several projects on distributed systems funded by the European Community, the Italian Ministry of Education and Research and by the Italian National Council of Research (CNR).

PLACE
PHOTO
HERE

Paolo Corsini achieved the Laurea in Electronic Engineering in 1969 as a member of ”Scuola Superiore di Studi Universitari e di Perfezionamento” of Pisa. Since 1972 he has held courses at the University of Pisa in the area of computer engineering, at first as associate Professor and, since 1987, as full Professor. From 1994 to 2002 he was Dean of the Faculty of Engineering of Pisa. His research interests have concentrated on algorithms for signal processing, on system architecture and on fuzzy logic. At the moment he is researching on sensor network.

PLACE
PHOTO
HERE

Barbara Panicucci received the Laurea and the Ph.D. in Mathematics from the University of Pisa. Her current research interests include equilibrium problems and Nash equilibria with applications, variational inequalities and Vehicle Routing problems. She is currently researcher at the Department of Sciences and Methods of Engineering of University of Modena and Reggio Emilia.

PLACE
PHOTO
HERE

Mauro Passacantando received the Laurea in Mathematics and the Ph.D. degree in mathematics from the University of Pisa, Italy, in 2000 and 2005, respectively. He is currently an Assistant Professor of operations research with the Department of Applied Mathematics of the University of Pisa. He has authored several scientific papers in international journals and books. From 2004 he teached several courses on operations research at the Faculty of Engineering of University of Pisa. His research is mainly devoted to variational inequalities and equilibrium Problems, concerning both theory and algorithms, and their applications to game theory.

PLACE
PHOTO
HERE

Alessio Vecchio received the Laurea in computer engineering and the Ph.D. in information engineering from the University of Pisa, Italy, in 1999 and 2003, respectively. Since 2006, he has been a Researcher with the Department of Information Engineering of the University of Pisa. His research interests include pervasive and ubiquitous systems, sensor networks, e-health and bioinformatics.