

Article

## Global Coverage Measurement Planning Strategies for Mobile Robots Equipped with a Remote Gas Sensor

Muhammad Asif Arain \*, Marco Trincavelli, Marcello Cirillo, Erik Schaffernicht and Achim J. Lilienthal

Mobile Robotics & Olfaction Lab, Center of Applied Autonomous Sensor Systems (AASS), School of Science & Technology, Örebro University, Örebro SE-70182, Sweden;  
E-Mails: marco.trincavelli@oru.se (M.T.); marcello.cirillo@oru.se (M.C.); erik.schaffernicht@oru.se (E.S.); achim.lilienthal@oru.se (A.J.L.)

\* Author to whom correspondence should be addressed; E-Mail: asif.arain@oru.se;  
Tel.: +46-19-30-1355; Fax: +46-19-30-3463.

Academic Editor: Markus W. Sigrist

Received: 24 November 2014 / Accepted: 25 February 2015 / Published: 20 March 2015

**Abstract:** The problem of gas detection is relevant to many real-world applications, such as leak detection in industrial settings and landfill monitoring. In this paper, we address the problem of gas detection in large areas with a mobile robotic platform equipped with a remote gas sensor. We propose an algorithm that leverages a novel method based on convex relaxation for quickly solving sensor placement problems, and for generating an efficient exploration plan for the robot. To demonstrate the applicability of our method to real-world environments, we performed a large number of experimental trials, both on randomly generated maps and on the map of a real environment. Our approach proves to be highly efficient in terms of computational requirements and to provide nearly-optimal solutions.

**Keywords:** remote gas detection; mobile robot olfaction; sensor planning; coverage planning; surveillance robots

### 1. Introduction

In a large number of situations, it is of utmost importance to assess whether a specific gas is present in an area, and establish its concentration, distribution, and the location of the source. This includes the

inspection of pipelines and chemical plants to detect the presence of dangerous gas leaks, the longterm monitoring of air pollution levels in big cities or of landfill sites to identify exploitable gas sources. It is often the case that the gases of interest are dangerous for humans and, therefore, human involvement should be minimized.

In the case of pollution monitoring in cities, the deployment of a network of stationary sensors has often proven to be an effective solution [1]. Similar solutions have also been tried in more critical environments, such as coal mines [2] and landfill sites [3]. However, whenever stationary solutions are employed in large areas, two problems arise: the measurements are inherently sparse, so that there can be no guarantee for the absence of the measured gases in the environment (e.g., in a chemical plant a leak could go undetected), and it is very difficult to decide where to place the sensors in the first place, especially if the environment is subject to changes in wind direction and speed, either caused by weather conditions or moving objects.

Mobile robotic olfaction is a recent research direction which aims at overcoming the problems of fixed sensor networks. It combines gas sensors, as used in the fixed networks, with the flexibility of a mobile platform [4]. Different platforms have been successfully tested in recent years for mapping the gas concentration and source localization, both in indoor and outdoor environments [5–7]. The use of mobile platforms has become even more appealing since the introduction of sensors which are capable of detecting gases remotely [8–10]. In particular, sensors based on tunable diode laser absorption spectroscopy (TDLAS) [11,12] allow for ranged sensing up to considerable distances (if the field of view is unobstructed), but are expensive and relatively bulky. Therefore, they are not suitable to be distributed in large numbers in the environment, but they can be used on-board a mobile platform.

The use of ranged sensing devices on mobile platforms introduces new challenges due to the robot's limited autonomy. The largest restriction is posed by battery power and hence the limited duration of monitoring activities. Thereto, it is required to explore the environment *efficiently*. Here, we focus on the problem of *gas detection in large environments*, both outdoor and indoor. We elaborate on the problem of verifying *if* a specific gas is present in the environment or not. More in particular, in this paper we present a novel approach for quickly generating exploration strategies for a mobile robot equipped with a sensor for remote gas detection. The resulting strategies are efficient, in the sense that they minimize the overall time the robot will take to inspect the whole environment.

The contributions of this paper are two-fold. Our first contribution is to formalize the remote gas sensing coverage problem for mobile robotics. Our second contribution is to present a two-step approximate solution to the aforementioned problem which relies on a novel convex optimization-based approximation. Here, we test our solution on large-scale scenarios, to demonstrate its applicability in real-world situations.

The paper is organized as follows: After presenting an overview of relevant work (Section 2), we define and formalize the gas detection coverage problem and how to obtain optimum solutions in Sections 3 and 4. We then present two alternative approaches to decompose the problem, their advantages and drawbacks (Section 5). A first experimental evaluation is detailed in Section 6, where the solution quality and the scalability of the different approaches is discussed. To overcome the scalability issues common to all previous approaches, we present in Section 7 a novel solution which uses re-weighted convex relaxation. We then evaluate this novel solution in Section 8 and conclude in Section 9.

## 2. Related Work

Stationary networks of gas sensors have been used over the years in different environments and in a variety of applications, such as pollution monitoring [13] and the measurement of the concentration of methane in the atmosphere [14]. The sensors commonly employed in such networks are *in situ* [15], which means that they have to come in direct contact with the gas they need to sense. This is obviously a strong limitation: the choice of their positioning is not trivial, new deployments could be necessary whenever the environment changes and the sensors must be re-calibrated and maintained over time. For specific gases, e.g., methane, other viable sensing technologies are available, such as infrared thermography [16], whose results, however, are highly affected by weather conditions, by the nature of the ground surface and by the distance between sensors and sources.

In recent years, many research groups have started to focus on mobile robot olfaction, a solution for gas sensing which would implicitly guarantee more flexibility than sensor networks, as it combines gas sensors with mobile platforms [17,18]. Mobile robots equipped with *in situ* gas sensors have been successfully used for mapping gas distributions [19,20] and leak detection [9,10]. A limitation of current approaches is that the robots generally move between pre-defined positions, or reactively follow gas plumes [21]. Finding good sensing positions for a mobile robot is still an open problem and it is closely related to the sensor placement problem for static networks.

In our work, we do not use *in situ* sensors, but we equip our robot with a TDLAS sensor, which can provide remote measurements up to a well defined sensing range [20]. A sensing action, that is, scanning an area of the environment with our sensor, is expensive, both in terms of time and of battery consumption. Similar costs should be accounted for whenever the robot moves from one sensing position to the next. Since here we are interested in detecting if a specific gas is at all present in a known environment, we need to scan the whole environment. Therefore, if we want to find battery- and time-efficient exploration strategies, we must take into account and minimize the costs associated both to movement and to sensing.

Neglecting the cost associated to the movements of the robot, the problem we would need to solve could be reduced to an *Art Gallery Problem* [22] or to a *View Planning Problem* [23]. The art gallery problem is NP-hard (non-deterministic polynomial-time hard [24]) in its most common variants and view planning is isomorphic to the *Set Covering Problem* [25], a well known NP-complete problem. This family of problems has been extensively studied over the past decades [26,27], but the algorithms proposed that solve the problems optimally are effective under restricted assumptions, such as not considering occlusions in the field of view of the sensors [28], or work only when the number of possible sensing configurations is relatively small [29,30]. There exist, however, efficient algorithms for calculating approximated solutions [31] for problems with a large number of possible sensing configurations. The solutions described above, however, present two major drawbacks in our context: first, they are mostly concerned with camera placement problems, which means that they rarely consider limited field of view; second, they inherently do not consider the cost of moving from one sensor position to the next. On the other hand, assuming known sensing positions, we would still need to solve a (Metric) *Traveling Salesman Problem* (TSP), that is, we should find the shortest tour that connects all the sensing

positions. This is a well-known NP-hard problem, but it is possible to optimally solve very large TSP instances, with thousands of locations [32].

Solving an art gallery problem first, and then calculating the shortest tour among the selected sensing positions by solving a TSP is an approach which has proven to be effective in many applications [33–36]. However, all the algorithms proposed rely on simplifying assumptions on the field of view of the sensors (e.g., 360° or unrestricted). More important still, the solution of an art gallery problem for a large number of candidate sensing positions with overlapping fields of view remains computationally challenging. By contrast, we use re-weighted convex relaxation to handle a high number of candidate sensing positions.

Our problem can also be related to planning for mobile robotics inspection, although in this case sensing cost is usually neglected. For instance, in [37,38], the authors present an inspection strategy for submerged ship hull with an autonomous underwater vehicle. They address the problem by developing a probabilistic planner to maximize uncertainty reduction while providing coverage of the mesh surface. In our case, we cannot ignore the cost of sensing actions and we aim at guaranteeing that the whole environment is observed. The *Generalized Covering Salesman Problem* [39] can explicitly address this last issue, as a solution of one of its instances is a tour which respects given covering constraints, but still sensing costs are not taken into accounts. A similar problem is solved in [40], where an approach based on mixed integer linear programming is used for finding a surveillance route for a mobile camera. The optimal solution approach we present here is highly reminiscent of the work of Tamioka *et al.* [40] and could be considered an extension of their algorithm: in our case, we can solve larger instances, while also considering costs associated with sensing actions.

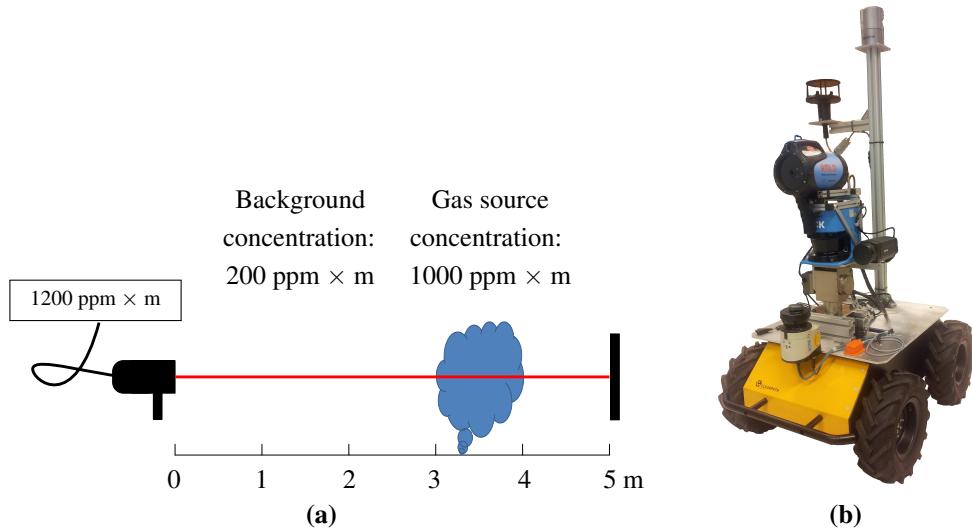
A combination of the View Planning Problem and the Metric TSP, the *Traveling View Planning Problem* has been defined and analyzed in [41,42]. It refers to finding a sequence of sensing actions while minimizing the overall cost. Here, the authors use an approximation method to find a bounded suboptimal solution which considers both traveling time and sensing time with respect to the available candidate sensing positions. However, it is not yet clear how to optimally select the candidate sensing positions and the approach presented in these papers does not scale well (in terms of computation time) when the size of the environment grows (in terms candidate sensing positions).

### 3. Problem Definition

In this article, we address the problem of detecting the presence of a specific gas in large environments, both indoor and outdoor. We assume a static environment with no major changes in the average concentration of the target gas due to wind or by other means. This assumption does not imply a major limitation to the gas detection problem, since it is possible to repeat an inspection tour over and over again, in case things are changing. However, these problems need to be addressed for subsequent tasks, like gas source localization and gas distribution mapping as discussed in Section 9.

Furthermore, we assume that the gas, if present, can be detected near ground level. This assumption applies to many real world applications: for instance, methane leaks in a landfill necessarily occur close to the ground. We assume that the presence of the gas in question can be observed by means of a ranged sensor, such as TDLAS. An example of such sensors is the Remote Methane Leak Detector, which can measure the integral concentration of methane over a reflected laser beam (see Figure 1a). The sensor

emits light in the near-infrared (NIR) band and analyses the spectrum of the reflection. Each gas has a unique absorption spectrum and the sensor is tuned to the target gas [43]. Hence, the TDLAS technology is very selective and only responds to its target gas even in the presence of multiple airborne substances. Thus, we can safely neglect the complex issue of dealing with gas mixtures. If the utilized sensor is only partially selective, e.g., like the often used metal oxide (MOX) sensors, it would be required to introduce an additional gas discrimination step. Different studies dealing with this particular issue are for example [44–46].



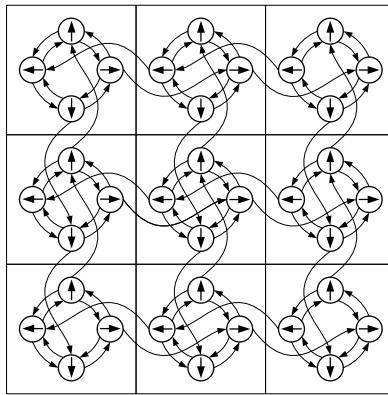
**Figure 1.** (a) the Remote Methane Leak Detector is a TDLAS sensor which can report the integral concentration of methane along its laser beam (parts per million  $\times$  meter); (b) *Gasbot*, a robotic platform for gas detection. *Gasbot* is a research platform based on a Husky A200. It is specially equipped with a methane sensitive remote sensor (RMLD) mounted in conjunction with a laser scanner on a pan-tilt unit, a laser scanner for self-localization and mapping, an anemometer and a thermal camera.

Note that methane is not the only gas that can be detected using this technology. Finally, we assume that the ranged sensor is mounted on a mobile platform, such as the one in Figure 1b, so that its beam can be directed towards the ground at a specific range from the robot and that the sensor can be rotated around its vertical axis.

Given a map of the environment in which we need to assess the presence of a specific gas, we divide it into a Cartesian grid, thus obtaining a set  $\mathcal{A}$  of  $n$  cells of identical size:  $\mathcal{A} = \{a_1, \dots, a_n\}$ . The set of all cells  $\mathcal{A}$  is partitioned into subsets  $\mathcal{O}$  and  $\mathcal{S}$ , where  $\mathcal{O}$  includes all the cells which contain an obstacle and therefore (1) are not traversable by the robot; and (2) can stop the beam of the gas sensor.  $\mathcal{S}$  includes all the cells which do not contain obstacles and are therefore traversable by the robot. A solution to the detection problem is an obstacle free closed path, or *tour*, within the map that the robot can traverse and a set of sensing actions along the tour which enables the robot to sense every cell in  $\mathcal{S}$ .

We limit the movement of the robot between a finite set of poses  $\mathcal{P}$ . Each  $p_j \in \mathcal{P}$  is defined by a two-tuple  $(a_j, \theta_j)$ , where  $a_j \in \mathcal{S}$  and  $\theta_j \in \Theta$ .  $\Theta$  is a finite set of allowed orientations, equally spaced between  $[0, 2\pi)$ . Note that the robot position within a cell corresponds to its center, regardless of the orientation. Here, we assume that the movements of the robot are limited to forward motions and

rotations. Thus, for  $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$ , the movements of the robot can be captured by a directed graph like the one represented in Figure 2—backward motions and a higher cardinality of  $\Theta$  could be obviously represented in a similar graph. In our problem definition, we use *time* as measure of cost. Given the times  $t^r$ , representing the time necessary to rotate  $2\pi/|\Theta|$ , and  $t^c$ , the cost associated to moving to the adjacent cell, the movement time  $t_{p_i \rightarrow p_j}^m$  from  $p_i$  to  $p_j$  can be easily calculated by finding the shortest path on the movement graph, where each rotation edge has a weight of  $t^r$  and every edge leading to a different cell has a weight of  $t^c$ . With this decomposition, and with a knowledge of the terrain in each cell, it would be straightforward to express the cost of movement in terms of battery consumption.



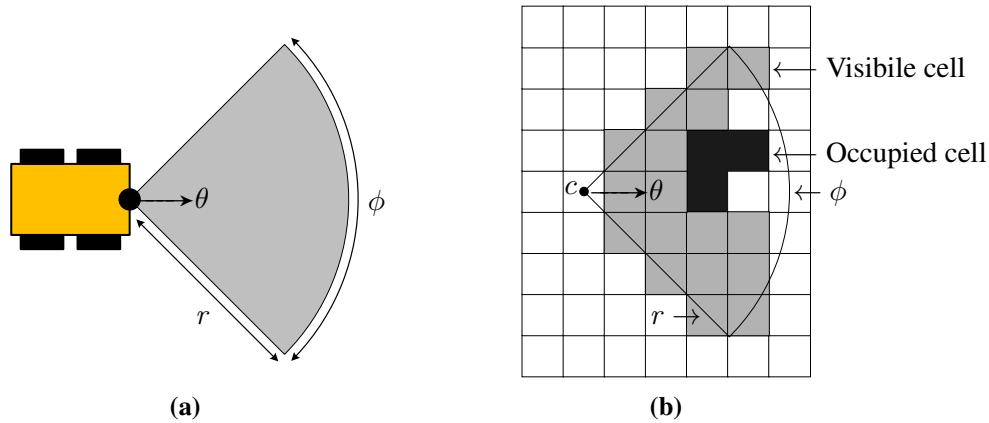
**Figure 2.** The graph captures the allowed movements of the robot on a grid map when  $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$  and only forward movements are allowed. Small circles indicate the poses where the robot can stop (the internal arrow indicates the orientation of the robot) and the directed edges its allowed movements. Note that in the figure the poses do not correspond to the centers of the cells, but this is only for clarity reasons.

We can now define a candidate sensing configuration on the movement graph described above. A candidate sensing configuration corresponds to a possible sensing action of the robot in a specific pose  $p$ , that is, it represents the possibility of performing a sensing action over a well defined area. Formally:

**Definition 1.** A candidate sensing configuration  $c_i$  is defined as a tuple  $(p_i, \phi_i, r_i)$ , where  $p_i \in \mathcal{P}$  is the robot's pose,  $\phi_i$  is the central angle of a circular sector and  $r_i$  its radius.

Hence, a candidate sensing configuration  $c_i$  would allow the robot positioned in  $p_i$  to scan a circular sector of central angle  $\phi_i$  and radius  $r_i$ , as shown in Figure 3a. Because of obstructions, not all the cells in the circular sector would be necessarily observable. Also, we need to define which portion of a cell should be swept by the laser beam to consider the cell itself as observed. Let us define  $\mathcal{C}_{\mathcal{P}}$  as the set of all candidate configurations defined over a set of robot poses  $\mathcal{P}$ . We can then define the visibility function  $v_{\mathcal{P}} : \mathcal{C}_{\mathcal{P}} \mapsto 2^{\mathbb{Z}^2}$ , such that  $v_{\mathcal{P}}(c)$  denotes the set of cells  $\in \mathcal{S}$  visible from  $c$ . In the remainder of this paper, we define  $v_{\mathcal{P}}$  so that a cell  $k$  is considered visible to a sensing configuration  $c_i$  if the line segment connecting the centers of  $c_i$  and  $k$  is in the circular sector of  $\phi_i$  and  $r_i$ , and does not intersect any occupied cell, as shown in the example in Figure 3b.

The cost associated to perform a sensing action in a candidate sensing configuration  $c$  is  $t_c^s$  and it depends on the central angle  $\phi$  associated with  $c$ .



**Figure 3.** (a) a candidate sensing configuration  $c$  allows the robot to scan a circular sector of central angle  $\phi$  and radius  $r$ ; (b)  $v_{\mathcal{P}}(c)$  is the visibility function which defines which are the cell that are observable from candidate sensing configuration  $c$ .

## 4. Finding an Optimal Solution

As defined above, given a discretized map of the environment, the sets of occupied and unoccupied cells  $\mathcal{O}$  and  $\mathcal{S}$  (such that  $\mathcal{A} = \mathcal{O} \cup \mathcal{S}$ ), the set of allowed poses  $\mathcal{P}$  for the robot, the set of candidate sensing configurations  $\mathcal{C}_{\mathcal{P}}$  defined over  $\mathcal{P}$ , the visibility function  $v_{\mathcal{P}}(c)$  and the cost functions  $t_{p_i \rightarrow p_j}^m$  (movement time from  $p_i$  to  $p_j$ , calculated as the shortest path on the movement graph) and  $t_c^s$  (sensing time of candidate configuration  $c$ ), a solution to the detection problem is an obstacle free tour defined as an ordered, finite set of sensing configurations  $\pi = \{c_1, \dots, c_k\}$  such that  $\bigcup_{c_i \in \pi} v_{\mathcal{P}}(c_i) = \mathcal{S}$ . The cost associated to a solution  $\pi$  is equal to the sum of the traveling costs (including the traveling cost to go from the last sensing configuration back to the first one) and sensing costs:

$$cost(\pi) = \sum_{i=1}^{k-1} t_{p_i \rightarrow p_{i+1}}^m + t_{p_k \rightarrow p_1}^m + \sum_{c_i \in \pi} t_{c_i}^s \quad (1)$$

Given the set  $\Pi$  of all valid solutions to a given problem instance, an optimal solution  $\pi_{opt}$  is the one with minimum cost:

$$\pi_{opt} = \operatorname*{argmin}_{\pi_i \in \Pi} cost(\pi_i)$$

#### *4.1. Problem Formulation*

Practically, we can calculate optimal solutions for the problem defined above by casting it as an optimization problem. Here, we use a formulation inspired by the work of Tomioka *et al.* [40]. More specifically, we define *flow variables* and *label variables* over the movement graph of the robot. Let us consider a problem instance with a set of candidate sensing configurations  $\mathcal{C}_P$ . Flow variables are binary variables associated with all pairs of candidate sensing configurations  $\mathcal{C}_P$ . A solution  $\pi = \{c_1, \dots, c_k\}$  to a problem instance represents a closed walk on the selected candidate sensing configurations. We can then refer to the configurations in a solution  $\pi$  by means of their indexes  $1, \dots, k$  and, as we are

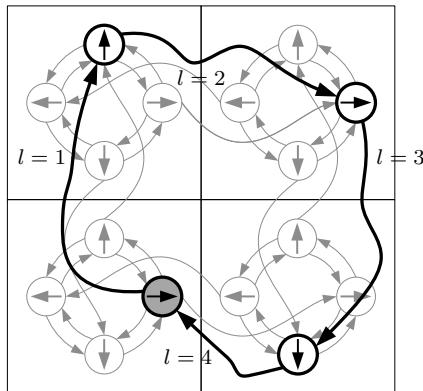
considering a tour, we can define the configuration of index  $k + 1 = 1$ , that is,  $c_{k+1} = c_1$  in  $\pi$ . Given a solution  $\pi$ , flow variable  $f_{c_i, c_j}$  represents whether  $c_i$  and  $c_j$  are consecutively traversed in  $\pi$ :

$$\forall c_i \in \mathcal{C}_P, \forall c_j \in \mathcal{C}_P, c_i \neq c_j \quad f_{c_i, c_j} = \begin{cases} 1 & \text{if } c_i = c_h \in \pi, c_j = c_{h+1} \in \pi. \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Label variables  $l_{c_i, c_j}$  are non-negative, real variables. They are assigned to the shortest path on the movement graph between any two candidate sensing configurations  $\mathcal{C}_P$ . As detailed in [40], label variables enforce flow conservation constraints, that is, constraints which guarantee that a valid solution to a problem instance is composed by closed walks in the movement graph. Given a solution  $\pi = \{c_1, \dots, c_k\}$ , label variables are assigned in such a way that, given  $c_i \in \pi$ , the label variable assigned to the (shortest) path between the predecessor in  $\pi$  of  $c_i$  has a lower value than the one assigned to the (shortest) path between  $c_i$  and its successor in  $\pi$ . This rule has a single exception in the *special configuration*, or *special vertex*, which is unique for each solution  $\pi$ . All other label variables are set to 0. Formally:

$$\forall c_i \in \mathcal{C}_P \quad \begin{cases} l_{c_h, c_i} < l_{c_i, c_j} & \text{if } c_i = c_g \in \pi, c_h = c_{g-1} \in \pi, c_j = c_{g+1} \in \pi \\ l_{c_h, c_i} > l_{c_i, c_j} & \text{if } c_i = c_g \in \pi, c_h = c_{g-1} \in \pi, c_j = c_{g+1} \in \pi, c_i \text{ special vertex of } \pi \\ l_{c_h, c_i} = l_{c_i, c_j} = 0 & \text{otherwise} \end{cases} \quad (3)$$

Finally, special vertex variables  $v_{c_i}$  are binary variables defined over every candidate sensing configuration in  $\mathcal{C}_P$ , where  $v_{c_i} = 1$  if configuration  $c_i$  is the unique special vertex in a given solution,  $v_{c_i} = 0$  otherwise. An example of a solution  $\pi$  where the shortest paths between configurations are annotated with label variables is shown in Figure 4.



**Figure 4.** An example of a solution to a problem instance. Configurations in the solution are represented by thick circles and the shortest paths between them by thick lines. Each path between two consecutive candidate sensing configurations in the solution is annotated with a label variable  $l$ , and the marked configuration corresponds to the special vertex.

We can find an optimal solution to a problem instance by casting it as a mixed integer linear programming problem, which we define as described in Equation (4). Given the set of candidate sensing

configurations  $\mathcal{C}_P$ , flow and label variables are represented in matrices  $F$  and  $L$ , both of size  $|\mathcal{C}_P| \times |\mathcal{C}_P|$ , where  $F[i, j]$  and  $L[i, j]$  represent flow variable  $f_{c_i, c_j}$  and label variable  $l_{c_i, c_j}$ , respectively. Please note that  $|A|$  denotes the cardinality of  $A$ . Special vertex vector  $S$  is a boolean vector of size  $|\mathcal{C}_P|$  with each element corresponding to a configuration in  $\mathcal{C}_P$ .

$$\text{minimize } \mathbf{1}^T (F \circ T_m) \mathbf{1} + \mathbf{1}^T (FT_s) \quad (4a)$$

subject to

$$(VF)\mathbf{1} \succeq \mathbf{1} \quad (4b)$$

$$F\mathbf{1} = (\mathbf{1}^T F)^T \quad (4c)$$

$$F \preceq L \preceq uF \quad (4d)$$

$$-\mathbf{u}S \preceq ((L\mathbf{1} - (\mathbf{1}^T L)^T) - F\mathbf{1}) \preceq -S \quad (4e)$$

$$\mathbf{1}^T S = 1 \quad (4f)$$

The objective function (Equation (4a)) minimizes total cost, expressed as the sum of traveling cost and sensing cost. In this function, matrix  $T_m$  of size  $|\mathcal{C}_P| \times |\mathcal{C}_P|$  captures the traveling costs ( $T_m[i, j] = t_{p_i \rightarrow p_j}^m$ ) and  $T_s$  is a column vector of size  $|\mathcal{C}_P|$  represents the sensing cost associated to each candidate sensing configuration ( $T_s[i] = t_{c_i}^s$ ).

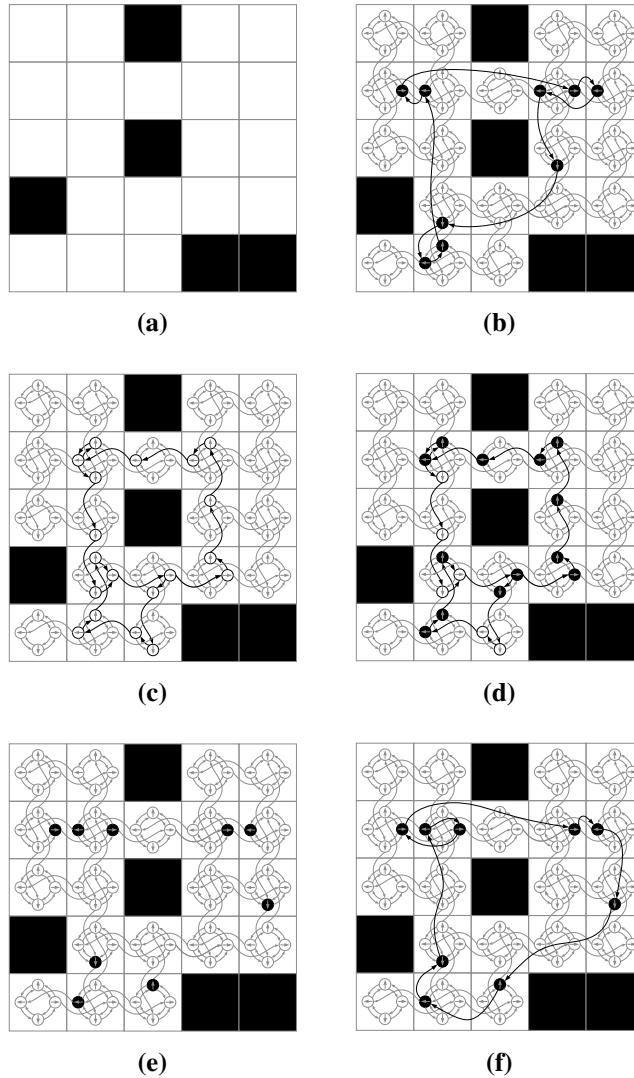
The optimization problem is subject to the following constraints:

- **Coverage constraints** (Equation (4b)) require that each cell  $a_i \in \mathcal{S}$  is visible from at least one of the candidate sensing configurations selected in the solution  $\pi$ . Given the problem definition, we can calculate  $V$  as a binary matrix of size  $n \times |\mathcal{C}_P|$ , where  $n$  is the number of cells in the problem. In particular:

$$V[a, c] = \begin{cases} 1 & \text{if } a \in v_P(c) \\ 0 & \text{otherwise} \end{cases}$$

- **Flow conservation constraints** (Equation (4c)) ensure that the solution will consist of one or more closed paths among the selected candidate sensing configurations.
- **Traveling route constraints** (Equations (4d) and (4e)) ensure that each candidate sensing configuration is visited only once in a solution, and, therefore, a solution consists of a single closed path.  $u$  is an upper limit constant for label variables.
- **Special vertex constraint** (Equation (4f)) restricts the number of special vertices to one and only one for each solution.

To reduce the size of the problem, without affecting the quality of the solution, it is possible to identify and disregard all those candidate sensing configurations which would not, in any case, belong to an optimal solution. For instance, all those sensing configurations  $c_i$  that do not observe any other cells  $v_P(c_i) = \emptyset$  (for instance, the ones placed in the cells at the border of the map) can be safely disregarded. Figure 5a is an example test map whose optimal solution is given in Figure 5b.



**Figure 5.** (a) a simple test map, where obstructed cells are represented in black and traversable ones in white. Candidate sensing configurations are defined over poses in the cells such that  $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$  and have identical  $\phi$  and  $r$  ( $r = 2$  cells,  $\phi = \frac{\pi}{2}$ ). In this example setup, the movement from one cell to the next requires 1 s, the rotation of  $\frac{\pi}{2}$  requires 0.5 s, and a sensing action takes 4 s; (b) the optimal solution, when traveling and sensing costs are considered at the same time. Here, the curved arrows represent the minimum distance from a sensing configuration to the next on the underlying graph. In this case, the total exploration time is 52.5 s (16.5 s for traveling and 36 s for sensing). (c,d): Here, traveling time is minimized first (see Section 5.1). (c) the minimum cost closed path from which all cells can be observed is calculated; (d) and then the minimum set of sensing configurations is selected, yielding an overall exploration time of 66.5 s. (e,f): Here, sensing time is minimized first (see Section 5.2). (e) the set of minimum cost sensing configurations is selected from which all cells are observable; (f) and then connecting them with the shortest closed path. This approach yields to an overall exploration time of 55 s.

Solving the mixed integer linear programming problem as described above yields an optimal solution to a given problem instance, with respect to the total operation time of the robot (traveling and sensing time). However, intuitively speaking, this corresponds to solving a combination of a *Watchman Route Problem* (that is, the problem of computing the shortest route to guard a known area) and an *Art Gallery Problem* (that is, selecting the minimum number of observation points to completely observe a known area). Therefore, the combined problem is NP-hard and, as shown in Section 6, this approach becomes practically unfeasible as the number of candidate sensing configurations grows.

## 5. Decomposing the Problem

Solving the problem so that the overall exploration time is minimized is of no practical applicability. However, a viable solution would be to sacrifice optimality by solving the problem in a two-step approach. This can be done in two different ways: (1) minimizing the overall traveling time (while guaranteeing complete coverage) and then selecting the sensing configurations along the route for minimum sensing time; or, (2) minimizing the sensing time, by selecting the sensing configurations necessary to cover the map and then finding the shortest path to connect them. In this section, we formally define such approaches.

### 5.1. Minimizing the Traveling Time

The core idea behind this decoupled approach is to find a minimum cost closed path for the robot which connects adjacent candidate sensing configurations. This path is subject to the requirement that all cells in  $\mathcal{S}$  are visible by at least one candidate sensing configuration included in the path. This first step roughly corresponds to solving a Watchman Route Problem. The second step of this approach consists of selecting the minimum cost sensing configurations which would allow the coverage of the whole environment among the ones included in the path.

Intuitively, this approach should yield good results whenever the time required for sensing actions is significantly less than the one for movement. Figure 5c,d show an example of this two phase approach, where first a tour is calculated, such that it connects contiguous candidate sensing configurations from which the whole environment is observable (Figure 5c). Then, in the second phase, the algorithm selects the set of minimum cost candidate sensing configurations on the path necessary to observe the whole environment (Figure 5d). The robot will follow the tour as generated in the first phase, but will only stop for sensing according on those sensing configurations selected in the second phase.

We use the mixed integer linear programming problem as defined by Tomioka *et al.* [40] to find a traveling path that provides full coverage without considering sensing cost. In order to then select the set of minimum cost candidate sensing configurations, we solve a sensor placement problem where the only variables considered are the candidate sensing configurations included in the traveling route. The formulation we use to solve the sensor placement problem is presented in the next section.

## 5.2. Minimizing the Sensing Time

The core idea behind this approach is dual with respect to the one presented above and consists in first selecting a set of minimum cost candidate sensing configurations from which all cells in  $\mathcal{S}$  are visible, and then solving a traveling salesman problem over them to find the shortest tour. This approach should therefore yield to better solutions than the previous one whenever the time necessary to perform sensing actions is much higher compared to the time required for the robot to move.

Figure 5e,f show how this approach would work compared to the ones described above: First the set of minimum cost candidate sensing configurations is selected (Figure 5e), and then a shortest closed path is calculated so that the robot will visit all the configurations (Figure 5f).

We find the set of minimum cost candidate sensing configurations by solving the following integer linear programming problem:

$$\underset{C}{\text{minimize}} \quad C^T T_s \quad (5a)$$

subject to

$$VC \succeq 1 \quad (5b)$$

$$C \in \{0, 1\} \quad (5c)$$

where  $C$  is a column vector of cardinality  $|\mathcal{C}_P|$  whose elements are binary variables representing if a given candidate sensing configuration is selected or not. Equation (5b) captures the necessary coverage constraints. For calculating the minimum traveling route, we use the Repetitive Nearest-Neighbor Algorithm [47] for calculating a Hamiltonian cycle.

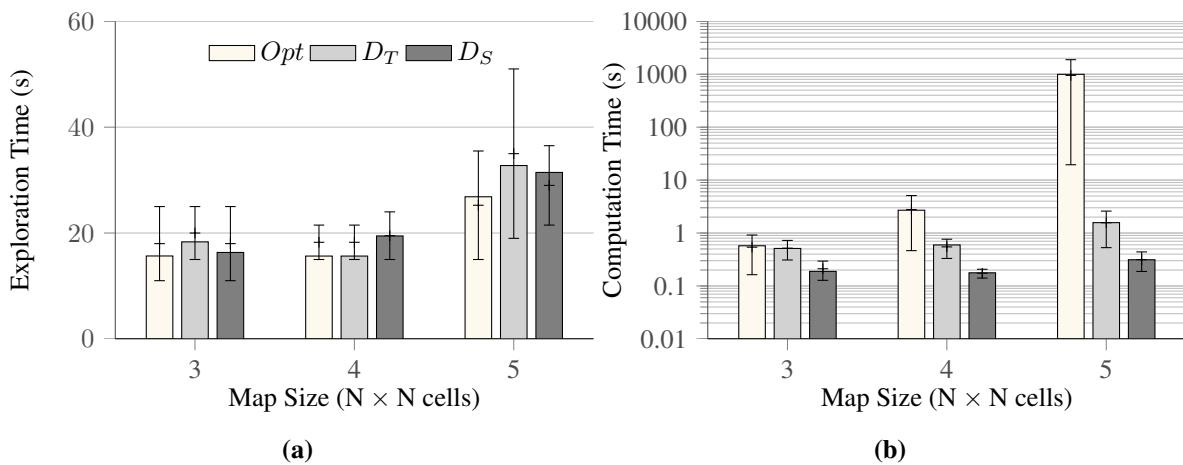
## 6. Experimental Evaluation

We present an evaluation of the three approaches described above: the one in which traveling time and sensing time are minimized at the same time (Section 4), and the decoupled ones, in which either traveling time or sensing time are minimized first (Section 5). The evaluation is conducted in sets of randomly generated maps of varying size, whose cells have a fixed side length of 1 m. Candidate sensing configurations are defined over poses in the cells such that  $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$  and have identical  $\phi$  and  $r$  ( $r = 15$  meters,  $\phi = \frac{\pi}{2}$ ). Sensing and movement times are set based on our previous experience with our mobile platform [20]: Each sensing action takes 4 s, movement from one cell to the adjacent towards which the robot is oriented requires 1 s, while a rotation of the mobile platform of  $\frac{\pi}{2}$  requires 0.5 s.

For the first part of the experimental evaluation, we prepared 6 sets of 10 randomly generated maps and one truncated set. Each set contains maps of identical size, starting from small  $4 \times 4$  cell maps, up to  $9 \times 9$  cell maps of which 10% cells contain an obstacle. For  $3 \times 3$  cells there are only three unique maps with a single occupied cell (approx. 10%), all other maps are just rotations. Thereto, the results for  $3 \times 3$  maps are averaged using the unique 3 maps only. Furthermore the maps are checked, whether the observable area is a single connected component. This means, there are no enclosed spaces, which cannot be reached by the robot.

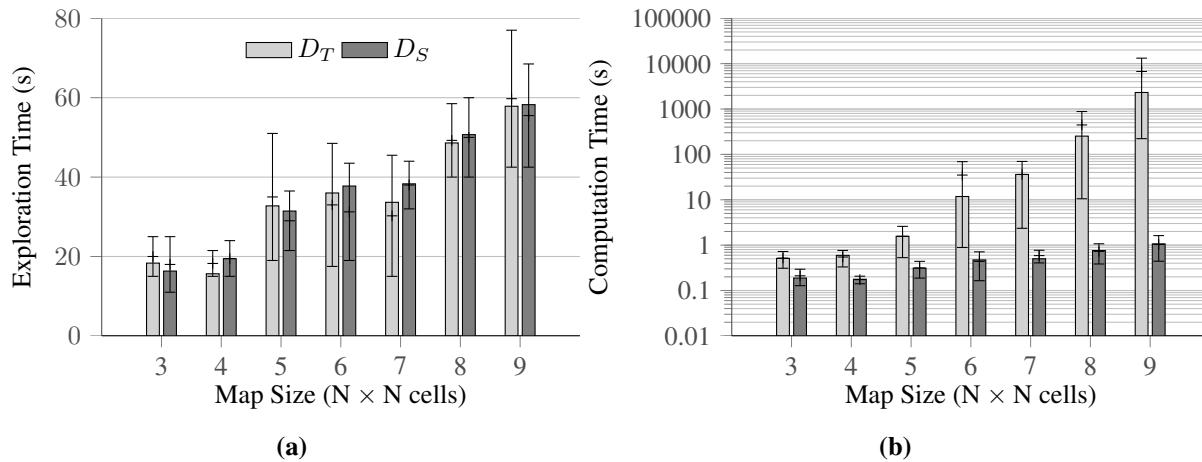
The evaluation focuses on two aspects: solution quality (in terms of overall exploration time) and time required to compute the solutions. However, finding the optimal solution for maps as small as

$6 \times 6$  becomes already an unpractical task, as the computation of 12 h on an Intel i7 Quad Core 2.60 GHz computer with 8 GB RAM was not enough to find the solution. The aggregated results for the comparison of all three methods up to maps of size  $5 \times 5$  are presented in Figure 6. Here,  $Opt$  indicates the optimal solution,  $D_T$  the decoupled approach in which traveling time is minimized first, and  $D_S$  the decoupled approach in which sensing time is minimized first. In particular, Figure 6a illustrates the aggregate quality of the solutions provided by the two decoupled approaches and the optimal one. In this limited set of examples, the disjoint approach where traveling time is minimized first shows better solution quality whenever it is possible to discard high number of redundant sensing configurations along the traveling route, whose field of view overlap. On the other hand, the disjoint approach where sensing time is minimized first shows better solution quality whenever the traveling path along the sensing configurations in the solution is small. When we consider the time necessary to compute a solution, the disjoint approach where the sensing times are minimized first is the most efficient, while the optimal solution becomes soon too expensive to be of any practical use (Figure 6b).



**Figure 6.** Comparison of all approaches, optimal and disjoint ( $Opt$ ,  $D_T$ ,  $D_S$ ), on three sets of maps. Each set contains maps of varying size, from  $3 \times 3$  to  $5 \times 5$ . The solid bars indicate the average values over the 3 maps for  $3 \times 3$  and 10 maps for the rest, and error bars show the minimum and maximum values observed during the trial. (a) shows the solution quality; and (b) shows the computation time taken by the all three approaches on a logarithmic scale.

We tested the decoupled approaches on the remaining 4 sets of maps (of size  $6 \times 6$  to  $9 \times 9$ ), and the results are presented in Figure 7. Since we could not calculate the optimal solutions for those runs, we directly compare the quality of the solutions found by the two disjoint approaches. In this second set of tests,  $D_T$ , in average, generates solutions of slightly better quality (Figure 7a), that is, solutions where the overall exploration times are lower. On the other hand, the computation times for this decoupled approach grow quickly as the size of the maps increase, as it is clearly shown in Figure 7b. It is very important to note that  $9 \times 9$  maps are still not large enough for representing real-world problems, and that  $D_T$  requires a time in the order of hours to solve such simple instances. On the other hand,  $D_S$  generates solutions for the same instances in less than two seconds, with a reasonable loss of quality (in this set of experimental runs, between 0.69% and 13.81% to the  $D_T$ ).



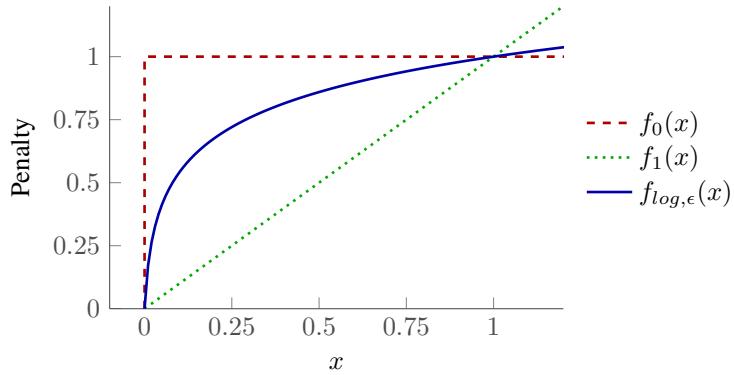
**Figure 7.** Comparison of two disjoint approaches up to maps of size  $9 \times 9$ . Notations are similar as in the previous figure, *i.e.*, bars indicate the average values over the 3 maps for  $3 \times 3$  and 10 maps for the rest, and error bars indicate the minimum and maximum values observed during the trial. (a) shows the solution quality of two disjoint approaches ( $D_T$ ,  $D_S$ ); and (b) shows the average computation time on a logarithmic scale for both approaches.

We continued our tests focusing on  $D_S$  up to maps of size  $26 \times 26$  when also this decoupled approach begins to require computation times in the order of hours. The bottleneck of the computation lies in finding a set of minimum cost candidate sensing configurations which can cover the whole map. This is a Sensor Placement Problem (SPP), which has been studied extensively in the past decades [27]. In the following, we propose a new convex relaxation method which allows us to drastically curtail the computation times.

## 7. A New Re-Weighted Convex Relaxation for Solving Sensor Placement Problems

To solve the problem of finding the set of sensing configurations that covers the whole environment at minimum cost (as we do in the decoupled approach described in Section 5.2) for instances with a high number of variables, we propose an iterative convex relaxation method which introduces intense sparsity, thus drastically reducing the number of variables to consider. The reduced problem can then be cast into an integer linear programming problem as formulated in Equation (5).

We call this algorithm *SPP with Re-weighted Convex Relaxation*, or in short *conv-SPP*. The SPP described in Equation (5) is NP-hard and corresponds to  $\ell_0$ -minimization whose solutions are binary variables, *i.e.*, sensing configurations. The combinatorial search soon becomes impracticable as the size of the problem grows. It is because  $\ell_0$ -minimization penalizes non-zero elements identical and the overall solution cost is based on the cardinality of the set.  $\ell_1$ -minimization on the other hand is much faster than  $\ell_0$ -minimization, and therefore, can handle relatively high number of variables. However, it penalizes each element proportional to the magnitude and thus the solution is no longer a binary vector (see Figure 8).



**Figure 8.** The concave loss function  $f_{log,\epsilon}(x)$  approximates better the  $\ell_0$  sparsity count  $f_0(x)$  than by the traditional convex  $\ell_1$  relaxation  $f_1(x)$  [48].

Candès *et al.* [48] proposed a re-weighted (iterative) formulation of  $\ell_1$ -minimization to handle large instances of a discrete problem in continuous domains. The approach effectively uses a concave loss function ( $f_{log,\epsilon}(x)$ ), which is much closer to the penalty function of  $\ell_0$ -minimization (see  $f_0(x)$  in Figure 8). The essential part of the approach is to solve an overall non-convex optimization problem through iterations of convex optimization. To sparsify the solution vector along the curve  $f_{log,\epsilon}(x)$  in Figure 8, a weight vector is introduced which evolves iteratively. The resultant vector is much sparser than the result of simple  $\ell_1$ -minimization with a linear penalty function. Afterwards, the reduced set of non-zero elements forms a reduced search space in which the combinatorial problem in its integer linear formulation (Equation (5)) can be solved in a feasible time.

Formally, our *conv-SPP* formulation, inspired by [48] is:

$$\underset{C}{\text{minimize}} \quad (W \circ C)^T T_s \quad (6a)$$

subject to

$$VC \succeq 1 \quad (6b)$$

$$0 \preceq C \preceq 1 \quad (6c)$$

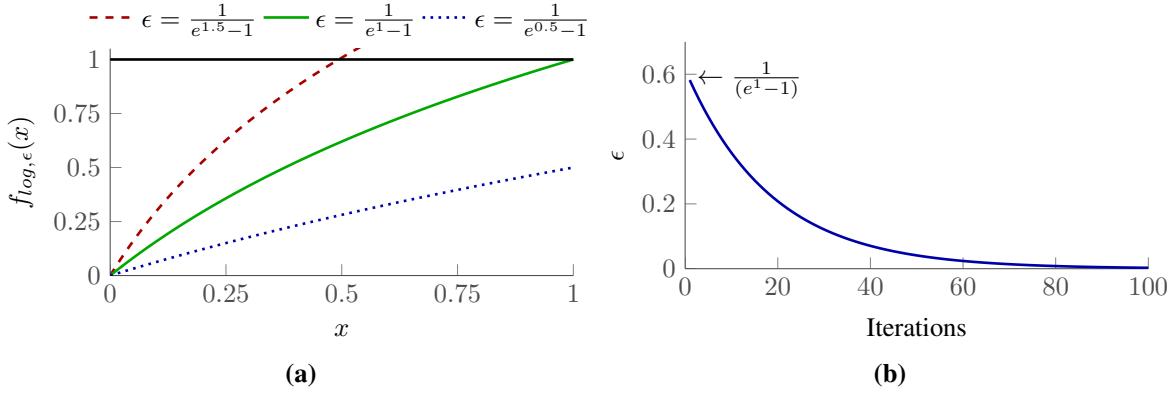
$W$  is a weight vector of cardinality  $|\mathcal{C}_P|$ . At the beginning, all the elements of  $W$  are equal to 1, *i.e.*,  $W = \mathbf{1}$ . In subsequent iterations, the weights are updated according to Equation (7).

$$W_{(i)} = \frac{\epsilon}{C_{(i)} + \epsilon} \quad i = 1, \dots, |W| \quad (7)$$

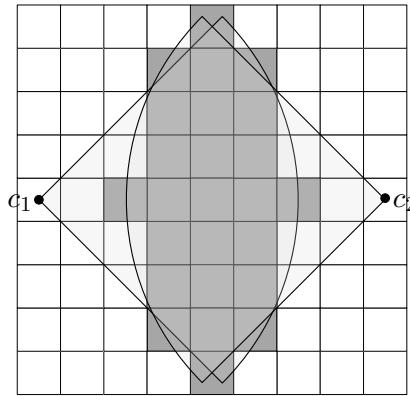
The parameter  $\epsilon$  is strictly positive and determines the rate of convergence. Smaller values of  $\epsilon$  allow for faster convergence (see Figure 9a) but at an increased risk of getting trapped in a local-minimum.

A local-minimum may occur whenever two or more variables have identical values and none of them is converging to the maximum or minimum limit. This could happen, for example, when two or more candidate sensing configurations have overlapping field of view and share a common subset of coverage. In Figure 10,  $c_1$  and  $c_2$  are two sensing configurations facing each other. The set  $\mathbb{O}$  contains elements (cells) under the overlapping field of view, *i.e.*  $\mathbb{O} \subset v_P(c_1)$  and  $\mathbb{O} \subset v_P(c_2)$ . If  $v_P(c_1) \setminus \mathbb{O}$  and  $v_P(c_2) \setminus \mathbb{O}$  are partially observed by another selected sensing configuration(s) in the solution. The configurations  $c_1$  and  $c_2$  will hold half of the magnitude to minimize the overall solution cost in continuous domain, and

any further iteration will not improve the situation, since the algorithm can not decide which of the two sensing configurations it prefers. This problem has then to be resolved in the subsequent combinatorial optimization step.



**Figure 9.** (a) Smaller value of  $\epsilon$  defines steeper penalty functions; (b) The exponential decay of  $\epsilon$  during the iterative procedure.



**Figure 10.** The sensing configurations  $c_1$  and  $c_2$  are trapped in a local minimum. The cells under the overlapping field of view are indicated in gray. Assuming that the remaining visible cells of both configurations, indicated in white under the coverage window, are partially (50%) observed by another configuration(s) in the solution. The  $c_1$  and  $c_2$  are set to 0.5 in the solution vector and any subsequent iteration does not improve the situation.

In our implementation we adapt the value of  $\epsilon$  in each step  $i$  according to Equation (8), so that it sets relatively high value at beginning to avoid a local minimum in early iterations to explore globally better solution, and then deceases to exponential decay with each iteration to speed up the convergence (see Figure 9b).

$$\epsilon_{(i)} = \left( \frac{1}{e^1 - 1} \right)^{1 + ((i-1) \times 10^{-1})} \quad (8)$$

The procedure stops if no improvement in sparsity for the last  $n$  ( $n = 5$  in our implementation) iterations is observed, or if a predefined number of iterations (150 in our case) is reached. We use

$\ell_\tau^0$  sparsity measure with  $\tau$  equals  $10^{-2}$ . In addition, we stop if the number of non-zero elements is low enough so that the combinatorial search is feasible (the value was empirically set to 80). Finally, the reduced vector of candidate sensing configurations in  $C$  is solved with the combinatorial method according to Equation (5). The overall procedure is summarized in Algorithm 1.

---

**Algorithm 1** *conv-SPP*


---

- 1: Set  $W = 1$ ;
  - 2: Solve Equation (6);
  - 3: Update  $W$  as in Equation (7) and  $\epsilon$  as in Equation (8);
  - 4: Go back to step 2, if none of the stopping criteria is true;
  - 5: Discard zero elements of  $C$ ;
  - 6: Solve Equation (5) with updated  $C$ ;
- 

In addition to the solution itself, our algorithm provides an interval in which the optimal solution for the sensing time minimization lies. The upper bound for this interval is the found solution of Algorithm 1, since the optimal solution is either equal to our solution or faster. The lower bound is obtained by considering the continuous problem instead of the combinatorial one. Simple  $\ell_1$ -minimization with linear penalty function  $f_1(x)$ ; which is the first iteration of step 2 in Algorithm 1, will provide a solution, which is always less than or equal to the optimal solution of combinatorial search in terms of sensing time.

The limits can be used to assess the quality of the solution when the optimal solution is not available. However, in practice, the optimal sensing time is very close to the upper bound of *conv-SPP*, as shown in the following section (Section 8).

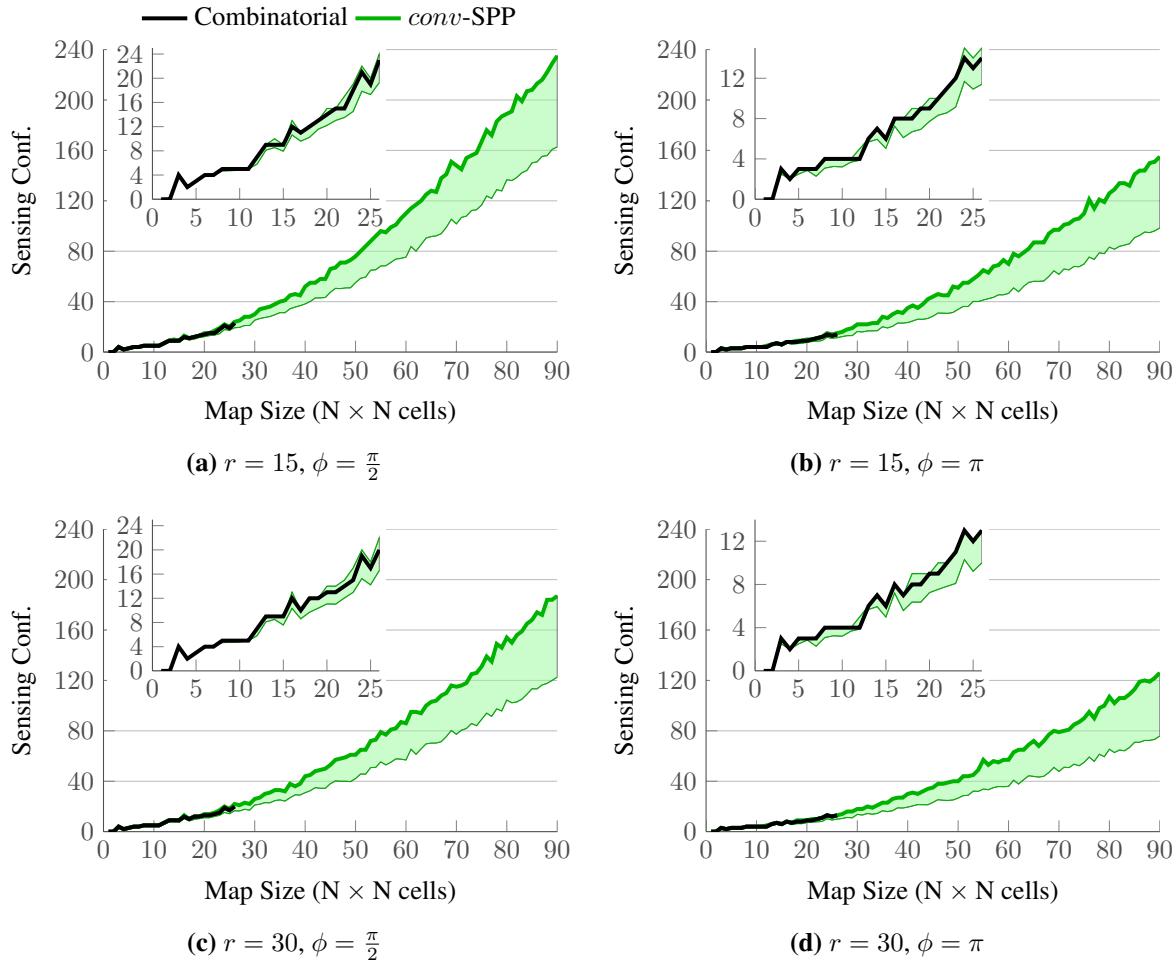
## 8. Evaluation of the New Method

We evaluated *conv-SPP* on sets of randomly generated grid maps of varying size, from  $3 \times 3$  to  $90 \times 90$  cells. In each map, 10% of the cells are occupied by obstacles, and we assume that cells have an identical side length of 1 m. Each set is composed by 10 different maps of identical size, with the only exception of the set of maps of size  $3 \times 3$ , where, considering all symmetries, only 3 unique instances are possible.

In each test run, the candidate sensing configurations are defined such that  $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$ , and  $\phi$  and  $r$  are fixed. Therefore, the sensing time for each candidate sensing configuration is set to 1 and the algorithm minimizes the number of sensing configurations. To evaluate the dependency of the algorithm with respect to the sensing parameters, we consider two different sensing radii ( $r = \{15, 30\}$  meters) and two central angles ( $\phi = \{\frac{\pi}{2}, \pi\}$ ).

To solve the optimization problem, we used the Gurobi Solver [49], with CVX, a package for specifying and solving convex programs [50,51] in Matlab. For these runs, we used a computer with an Intel i7 Quad Core 2.60 GHz and 8 GB RAM.

We compared the optimal solutions of the sensor placement problem instances with the solutions provided by *conv-SPP* on maps up to the size of  $26 \times 26$  cells. Larger maps required computation times in the order of hours to solve a single instance optimally.

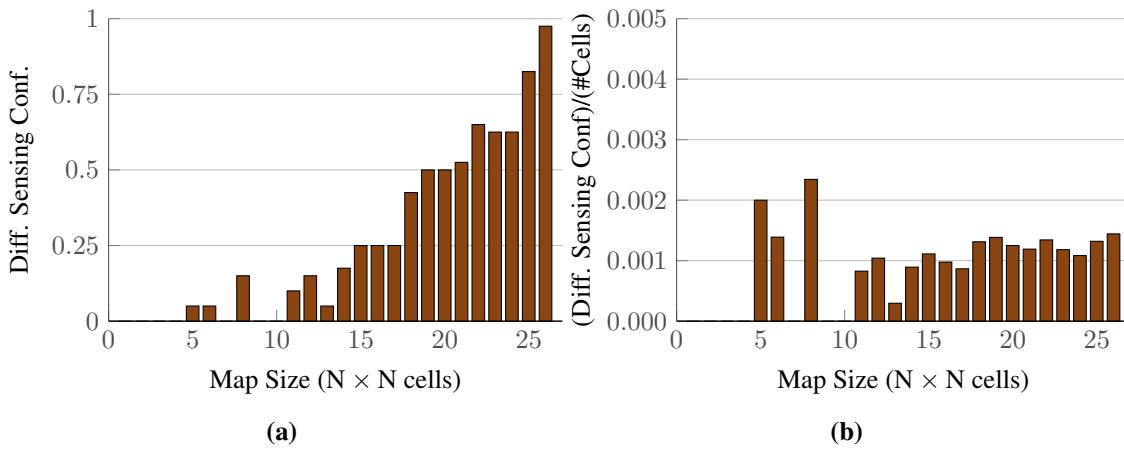


**Figure 11.** Comparison of the quality of the solutions obtained with *conv-SPP* and the combinatorial method on randomly generated maps. Sensing parameters are fixed to  $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$ ,  $r = \{15, 30\}$ , and  $\phi = \{\pi, \pi/2\}$ . The optimal solutions (thick black lines) are close to the solutions provided by *conv-SPP* (thick green lines) with respect to the number of sensing configurations selected. The lower bounds of the green intervals represent the result of a single  $\ell_1$ -minimization step.

The results for a single map randomly chosen from each set are shown in Figure 11. The thick black lines represent the number of sensing configurations selected in the optimal solutions, while the upper bounds of the green-colored intervals represent the number of sensing configurations selected by *conv-SPP*. The lower bounds of the intervals represent the number of configurations after the first iteration of the  $\ell_1$ -minimization. The inset graphs show the results up to maps of size  $26 \times 26$ , for which the optimal solutions are available. *conv-SPP* obtains results which are very close to the optimal solutions in terms of the number of sensing configurations selected, as our method does not add more than two sensing configurations for any instance. Figure 12 shows the average difference of solution quality, which, for the available results, is always less than one sensing configuration.

Figure 13 shows the aggregated computation times for both approaches: The thick lines represent the average computation times for each set of maps, while the colored intervals are bounded by the minimum and maximum times it took to solve all the instances in one set. The inset graphs show the

computation times on a logarithmic scale: As it can be seen, the time required to calculate the optimal solution of the problem increases steeply with the size of the maps considered. It is worth noticing that the calculation of the optimal solutions for maps up to  $13 \times 13$  cells takes less time than when *conv-SPP* is employed. This is because the number of variables is small and the combinatorial method can find a solution quickly. On the other hand, *conv-SPP* runs at least one iteration of convex relaxation and then performs combinatorial optimization for the remaining variables. Therefore, the iterations of convex relaxation simply require additional time. We stopped computing the optimal solutions for a set of maps whenever at least one instance of the set required more than one hour to be solved optimally. By contrast, our method could solve all the instances in less than 800 s.

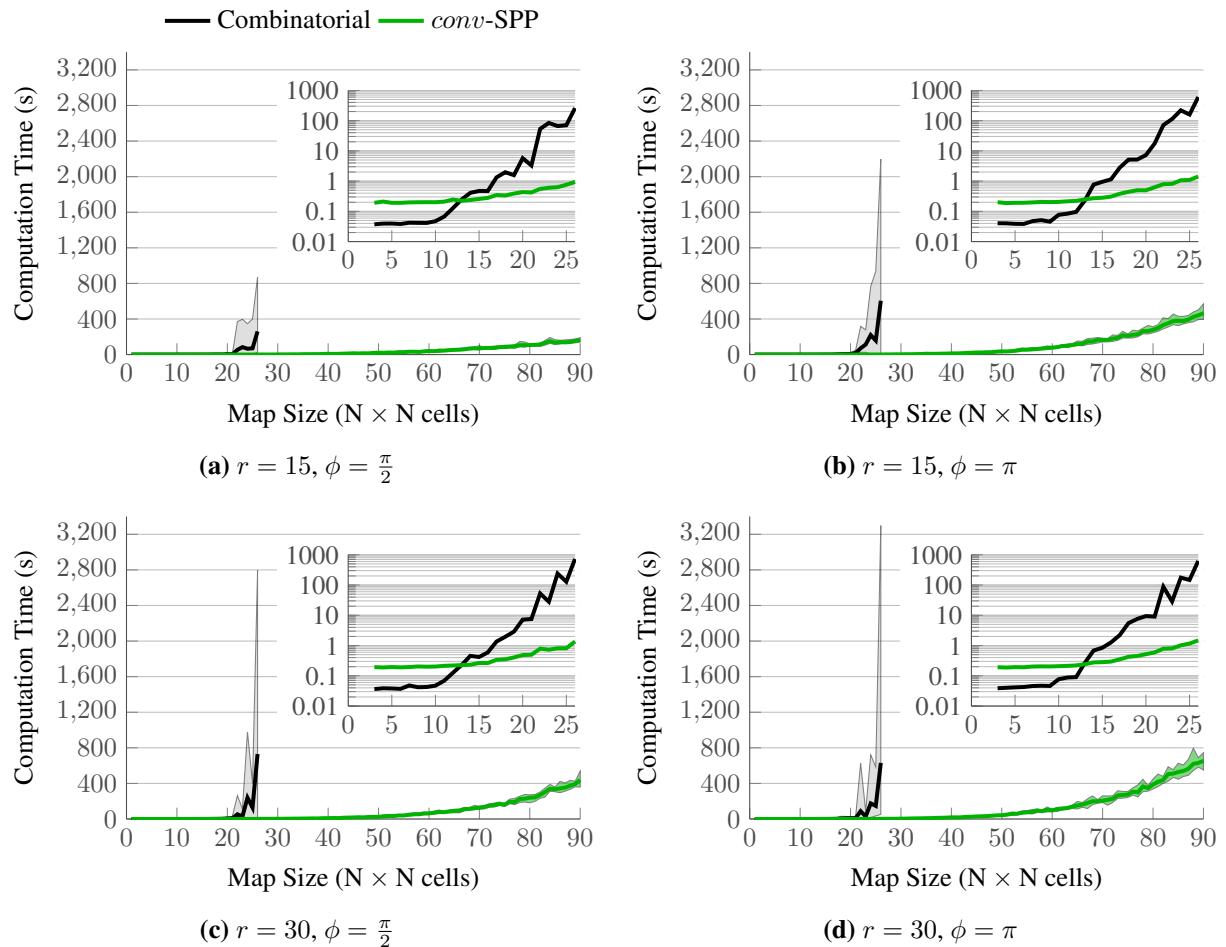


**Figure 12.** The solution quality gap between *conv-SPP* and combinatorial optimization. **(a)** shows the average difference for all possible combinations of the sensing parameters  $r = \{15, 30\}$  and  $\phi = \{\pi, \pi/2\}$ ; **(b)** shows the average difference divided by the number of cells in the maps.

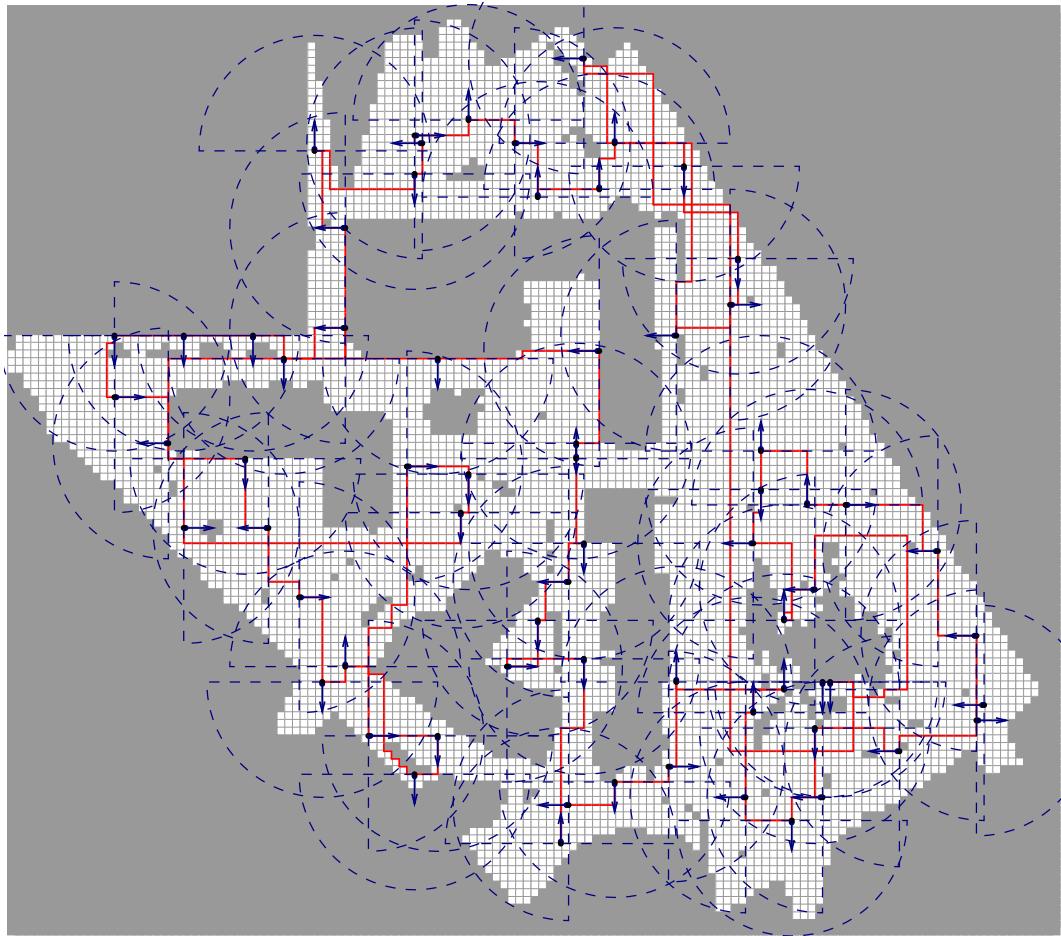
Within the same set of maps of identical size, different instances can require very different computation times for computing the optimal solution. Conversely, this is not the case when using *conv-SPP*. This is because few changes in the map layout can completely change the discrete optimization problem, while, on the other hand, the continuous problem is much less affected, and in *conv-SPP* the combinatorial solver is used only on a reduced set of variables. Hence, another advantage of *conv-SPP* is that the computation time is largely independent of the particular problem instance within a given problem size.

We also used the Freiburg University campus map [54] to demonstrate our algorithm on a real map. We discretized the map into a Cartesian grid where each cell covers  $1 \text{ m}^2$ . The map size is  $120 \times 136$  cells, 6113 of which are unoccupied (see Figure 14). For the sensing parameters  $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$ ,  $r = 15 \text{ m}$ , and  $\phi = \pi$ , our algorithm generated a solution of 68 sensing configurations in 90.87 s with a lower bound of 53.03 configurations. This scenario is evaluated in based on real map data (but without actual gas sources) to verify that our approach is able to efficiently cover a large real environment. In our ongoing work, we are performing actual gas detection tasks using the described algorithm on a robot in different environments (see Figure 15a for an example). Initial results are quite

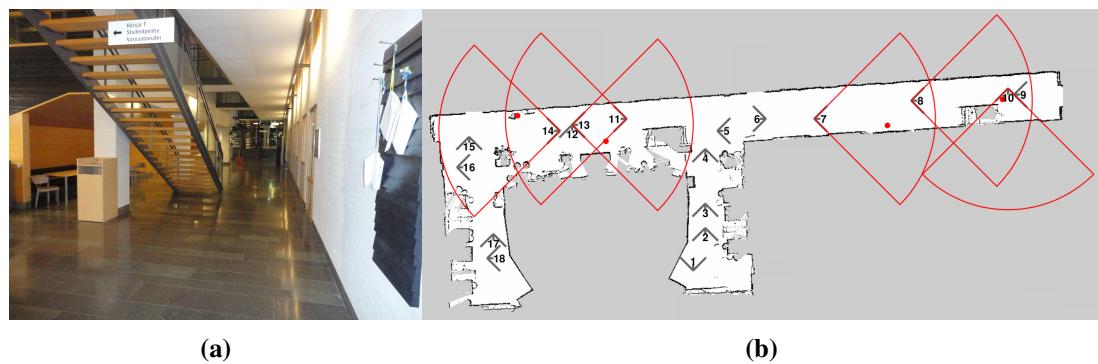
promising, as shown in Figure 15b. For more details on the experiments and the evaluation, we refer the reader to our forthcoming publication [52].



**Figure 13.** Computation times to calculate the optimal solutions and to solve the instances with *conv*-SPP. The thick lines represent the average computation times for each set of maps, while the colored intervals are bounded by the minimum and maximum times it took to solve all the instances in one set. The inset graphs show the computation times on a logarithmic scale.



**Figure 14.** The exploration plan generated for the map of the Freiburg University campus (obstacles are represented in gray). The positions of the sensing configurations are represented with blue dots, blue arrows indicate their orientation and dashed-lines their field of view. A closed path through the configurations is shown in red.



**Figure 15.** (a) an indoor setting for the gas detection task; (b) graphical results: the gas sources are indicated by red dots, the planned sensing configuration by truncated cones in gray, and the small number associated to each configuration indicates the order in which they are visited. The configurations plotted in red indicate the positive readings and the actual area they cover.

## 9. Conclusions

In this paper, we discussed the problem of gas detection in large environments with a mobile robot equipped with a remote gas sensor. Generally speaking, given an environment where a specific gas may be present in detectable concentrations, a solution to the problem would be a cost-effective exploration plan for the robot so that it could ascertain whether this is the case or not. Here, we consider cost to be directly proportional to the time required to complete the exploration plan.

We provided a formal definition of the problem and suggested several possible methods for solving it. The main contribution of this paper is *conv*-SPP, an algorithm for quickly finding an exploration plan which guarantees a complete coverage of the environment. Our algorithm leverages a novel method based on convex relaxation that drastically reduces the number of variables and it is therefore effective for solving large problem instances. Furthermore, *conv*-SPP offers guarantees on the quality of the solution, as it is always possible to assess the theoretical maximum distance of the solution provided to the optimal one. In practice, we show that in all the cases for which we could experimentally solve the problems optimally, our algorithm always generated results very close to the optimal ones.

We performed extensive validation of our approach in simulation, both on randomly generated maps of various sizes and on the map of the Freiburg University campus. We also compared our methods with other approaches, both in terms of solution quality and of computational requirements. In our future work, we want to test our algorithm in real world scenarios with a mobile platform. We also intend to extend our problem definition to take further advantage of remote sensing and include non-traversable but observable cells.

In this article, we are addressing the planning problem for gas detection only. For addressing this problem a pregenerated plan, like the approach we presented here, is sufficient. Future work will address how to deal with actual gas detections. Possible questions of interest include the problem of gas source localization [17,53], which often requires an active strategy to follow the trail to the source. Robot assisted gas tomography for gas distribution mapping with remote sensors [19,46] requires the consideration of different sensing positions in order to meet constraints regarding the overlap of the sensing fields and intersection angles that are necessary to reconstruct the gas distribution. Optimizing the sensing geometry for tomographic measurements [55] will be one focus of our future work. Extending the presented work to multi-robot scenarios, in which a fleet of robots cooperatively monitors a target area [56–60] is another interesting direction for future work.

## Author Contributions

- M.A. Arain: Problem formalization, algorithm design, experimental design, evaluation and manuscript writing.
- M. Trincavelli: Problem formalization, algorithm design and experimental design.
- M. Cirillo: Problem formalization, algorithm design, experimental design, evaluation and manuscript writing.
- E. Schaffernicht: Problem formalization, experimental design, evaluation and manuscript writing.
- A.J. Lilenthal: Problem formalization, evaluation and manuscript writing.

## Conflicts of Interest

The authors declare no conflict of interest.

## References

1. Asadi, S.; Badica, C.; Comes, T.; Conrado, C.; Evers, V.; Ilie, S.; Jensen, J.S.; Lilienthal, A.; Milan, B.; Neidhart, T.; *et al.* ICT solutions supporting collaborative information acquisition, situation assessment and decision making in contemporary environmental management problems: the DIADEM approach. In Proceedings of the International Conference on Innovations in Sharing Environmental Observation and Information (EnviroInfo), Ispra, Italy, 5–7 October 2011; pp. 920–931.
2. Zhou, G.; Zhu, Z.; Chen, G.; Hu, N. Energy-Efficient Chain-Type Wireless Sensor Network for Gas Monitoring. In Proceedings of the Second International Conference on Information and Computing Science (ICIC), Manchester, UK, 21–22 May 2009; pp. 125–128.
3. Kiernan, B.M.; Beirne, S.; Fay, C.; Diamond, D. Landfill Gas Monitoring at Borehole Wells using an Autonomous Environmental Monitoring System. In Proceedings of the 5th International Conference on Climate Change and Global Warming, Heidelberg, Germany, 24–26 September 2008; pp. 166–171.
4. Trincavelli, M.; Reggente, M.; Coradeschi, S.; Loutfi, A.; Ishida, H.; Lilienthal, A.J. Towards Environmental Monitoring with Mobile Robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Nice, France, 22–26 September 2008; pp. 2210–2215.
5. Ishida, H. Blimp Robot for Three-Dimensional Gas Distribution Mapping in Indoor Environment. *AIP Conf. Proc.* **2009**, *1137*, 61–64.
6. Hernandez Bennetts, V.; Lilienthal, A.J.; Neumann, P.P.; Trincavelli, M. Mobile robots for localizing gas emission sources on landfill sites: is bio-inspiration the way to go? *Front. Neuroeng.* **2012**, *4*, 1–20.
7. Neumann, P.; Hernandez Bennetts, V.; Bartholmai, M. Adaptive Gas Source Localization Strategies and Gas Distribution Mapping using a Gas-sensitive Micro-Drone. In Proceedings of the 16th GMA/ITG-Fachtagung Sensoren und Messsysteme, Nuremberg, Germany, 22–23 May 2012; pp. 800–809.
8. Baetz, W.; Kroll, A.; Bonow, G. Mobile Robots with Active IR-Optical Sensing for Remote Gas Detection and Source Localization. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; pp. 2773–2778.
9. Soldan, S.; Welle, J.; Barz, T.; Kroll, A.; Schulz, D. Towards Autonomous Robotic Systems for Remote Gas Leak Detection and Localization in Industrial Environments. In *Field and Service Robotics*; Yoshida, K., Tadokoro, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 92, pp. 233–247.

10. Bonow, G.; Kroll, A. Gas leak localization in industrial environments using a TDLAS-based remote gas sensor and autonomous mobile robot with the Tri-Max method. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 987–992.
11. Lackner, M. Tunable Diode Laser Absorption Spectroscopy (TDLAS) in the Process Industries- A Review. *Rev. Chem. Eng.* **2007**, *23*, 65–147.
12. Frish, M.B.; Wainner, R.T.; Green, B.D.; Laderer, M.C.; Allen, M.G. Standoff gas leak detectors based on tunable diode laser absorption spectroscopy. In Proceedings of the SPIE Optics East, Boston, MA, USA, 23–26 October 2005.
13. Tsujita, W.; Yoshino, A.; Ishida, H.; Moriizumi, T. Gas sensor network for air-pollution monitoring. *Sens. Actuators B: Chem.* **2005**, *110*, 304–311.
14. Somov, A.; Baranov, A.; Savkin, A.; Spirjakin, D.; Spirjakin, A.; Passerone, R. Development of wireless sensor network for combustible gas monitoring. *Sens. Actuators A: Phys.* **2011**, *171*, 398–405.
15. Liu, X.; Cheng, S.; Liu, H.; Hu, S.; Zhang, D.; Ning, H. A Survey on Gas Sensing Technology. *Sensors* **2012**, *12*, 9635–9665.
16. Lewis, A.W.; Yuen, S.T.S.; Smith, A.J.R. Detection of gas leakage from landfills using infrared thermography - applicability and limitations. *Waste Manag. Res.* **2003**, *21*, 436–447.
17. Kowadlo, G.; Russell, R.A. Robot Odor Localization: A Taxonomy and Survey. *Int. J. Robot. Res.* **2008**, *27*, 869–894.
18. Ishida, H.; Wada, Y.; Matsukura, H. Chemical Sensing in Robotic Applications: A Review. *IEEE Sens. J.* **2012**, *12*, 3163–3173.
19. Hernandez Bennetts, V.M.; Lilienthal, A.J.; Khaliq, A.A.; Sesé, V.P.; Trincavelli, M. Towards Real-World Gas Distribution Mapping and Leak Localization Using a Mobile Robot with 3D and Remote Gas Sensing Capabilities. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 2335–2340.
20. Hernandez Bennetts, V.; Schaffernicht, E.; Stoyanov, T.; Lilienthal, A.J.; Trincavelli, M. Robot Assisted Gas Tomography—Localizing Methane Leaks in Outdoor Environments. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6362–6367.
21. Lilienthal, A.; Streichert, F.; Zell, A. Model-based Shape Analysis of Gas Concentration Gridmaps for Improved Gas Source Localisation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain, 18–22 April 2005; pp. 3564–3569.
22. Lee, D.T.; Lin, A.K. Computational Complexity of Art Gallery Problems. *IEEE Trans. Inf. Theory* **1986**, *32*, 276–282.
23. Scott, W.R. Model-based view planning. *Mach. Vis. Appl.* **2009**, *20*, 47–69.
24. Korte, B.; Vygen, J. *Combinatorial Optimization; Theory and Algorithms*, 2nd ed.; Springer-Verlag: Berlin/Heidelberg, Germany, 2002; pp. 327–259.
25. Erdem, U.M.; Sclaroff, S. Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. *Comput. Vis. Image Underst.* **2006**, *103*, 156–169.

26. Tarabanis, K.A.; Allen, P.K.; Tsai, R.Y. A Survey of Sensor Planning in Computer Vision. *IEEE Trans. Robot. Autom.* **1995**, *11*, 86–104.
27. Ramsden, D. Optimization Approaches to Sensor Placement Problems. Ph.D. Thesis, Rensselaer Polytechnic Institute, Troy, NY, USA, December 2009.
28. An, W.; Shao, F.M.; Meng, H. The coverage-control optimization in sensor network subject to sensing area. *Comput. Math. Appl.* **2009**, *57*, 529–539.
29. Angella, F.; Reithler, L.; Gallesio, F. Optimal Development of Cameras for Video Surveillance Systems. In Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), London, UK, 5–7 September 2007; pp. 388–392.
30. Chen, X.; Davis, J. An occlusion metric for selecting robust camera configurations. *Mach. Vis. Appl.* **2008**, *19*, 217–222.
31. Chiu, P.L.; Lin, F.Y.S. A Simulated Annealing Algorithm to Support the Sensor Placement for Target Location. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, Ontario, Canada, 2–5 May 2004; pp. 867–870.
32. Applegate, D.L.; Bixby, R.E.; Chvátal, V.; Cook, W.J. *The Traveling Salesman Problem: A Computational Study*; Princeton University Press: Princeton, NJ, USA, 2007; pp. 1–58.
33. Faigl, J.; Kulich, M.; Přeučil, L. A Sensor Placement Algorithm for a Mobile Robot Inspection Planning. *J. Intell. Robot. Syst.* **2011**, *62*, 329–353.
34. Danner, T.; Kavraki, L.E. Randomized Planning for Short Inspection Paths. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), San Francisco, CA, USA, 24–28 April 2000; pp. 971–976.
35. Kazazakis, G.D.; Argyros, A.A. Fast Positioning of Limited-Visibility Guards for the Inspection of 2D Workspaces. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Lausanne, Switzerland, 30 September–4 October 2002; pp. 2843–2848.
36. Vieira Neto, H.; Nehmzow, U. Real-time Automated Visual Inspection using Mobile Robots. *J. Intell. Robot. Syst.* **2007**, *49*, 293–307.
37. Hollinger, G.A.; Englot, B.; Hover, F.; Mitra, U.; Sukhatme, G.S. Uncertainty-Driven View Planning for Underwater Inspection. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Saint Paul, MN, USA, 14–18 May 2012; pp. 4884–4891.
38. Englot, B.; Hover, F.S. Three-dimensional coverage planning for an underwater inspection robot. *Int. J. Robot. Res.* **2013**, *32*, 1048–1073.
39. Golden, B.; Naji-Azimi, Z.; Raghavan, S.; Salari, M.; Toth, P. The Generalized Covering Salesman Problem. *INFORMS J. Comput.* **2012**, *24*, 534–553.
40. Tomioka, Y.; Takara, A.; Kitazawa, H. Generation of an Optimum Patrol Course for Mobile Surveillance Camera. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 216–224.
41. Wang, P.; Krishnamurti, R.; Gupta, K. *View Planning Problem with Travel Costs: Program Formulation, Hardness of Approximation, and Approximation Algorithms*; Simon Fraser University: Burnaby, BC, Canada, 2006.
42. Wang, P.; Krishnamurti, R.; Gupta, K. View Planning Problem with Combined View and Traveling Cost. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Roma, Italy, 10–14 April 2007; pp. 711–716.

43. Adam, H.; Bauer, J.; Thornton, E.; Tulip, J. Greenhouse Gas Emissions and Reduction Studies Using a Portable TDL Gas Monitor. In Proceedings of the climate change and GHG technology conference and tradeshow, Calgary, AB, Canada, 22–24 May 2002.
44. Trincavelli, M. Gas Discrimination for Mobile Robots. *Künstliche Intell.* **2011**, *25*, 351–354.
45. Fonollosa, J.; Rodríguez-Luján, I.; Trincavelli, M.; Vergara, A.; Huerta, R. Chemical Discrimination in Turbulent Gas Mixtures with MOX Sensors Validated by Gas Chromatography-Mass Spectrometry. *Sensors* **2014**, *14*, 19336–19353.
46. Hernandez Bennetts, V.; Schaffernicht, E.; Pomareda, V.; Lilienthal, A.J.; Marco, S.; Trincavelli, M. Combining Non Selective Gas Sensors on a Mobile Robot for Identification and Mapping of Multiple Chemical Compounds. *Sensors* **2014**, *14*, 17331–17352.
47. Kizilates, G.; Nuriyeva, F. On the Nearest Neighbor Algorithms for the Traveling Salesman Problem. In *Advances in Computational Science, Engineering and Information Technology*; Nagamalai, D.; Kumar, A.; Annamalai, A., Eds.; Springer International Publishing: Cham, Switzerland, 2013; Volume 225, pp. 111–118.
48. Candès, E.J.; Wakin, M.B.; Boyd, S.P. Enhancing Sparsity by Reweighted  $\ell_1$  Minimization. *J. Fourier Anal. Appl.* **2008**, *14*, 877–905.
49. Gurobi Optimizer, Version 5.6. Available online: <http://www.gurobi.com> (accessed on 10 December 2014).
50. Grant, M.; Boyd, S. CVX: Matlab Software for Disciplined Convex Programming, Version 2.1. Available online: <http://cvxr.com/cvx> (accessed on 10 December 2014).
51. Grant, M.C.; Boyd, S.P. Graph Implementations for Nonsmooth Convex Programs. In *Recent Advances in Learning and Control*; Blondel, V., Boyd, S.P., Kimura, H., Eds.; Springer: London, UK, 2008; Volume 371, pp. 95–110.
52. Arain, M.A.; Cirillo, M.; Hernandez Bennetts, V.; Schaffernicht, E.; Trincavelli, M.; Lilienthal, A.J. Efficient Measurement Planning for Remote Gas Sensing with Mobile Robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015;
53. Fukazawa, Y.; Ishida, H. Estimating Gas-Source Location in Outdoor Environment Using Mobile Robot Equipped with Gas Sensors and Anemometer. In Proceedings of the IEEE Sensors Conference, Christchurch, New Zealand, 25–28 October 2009; pp. 1721–1724.
54. OpenSLAM GMapping. Available online: <https://www.openslam.org/gmapping.html> (accessed on 10 December 2014).
55. Hartl, A.; Song, B.C.; Pundt, I. 2-D reconstruction of atmospheric concentration peaks from horizontal long path DOAS tomographic measurements: parametrisation and geometry within a discrete approach. *Atmos. Chem. Phys.* **2006**, *6*, 847–861.
56. Lochmatter, T. Bio-Inspired and Probabilistic Algorithms for Distributed Odor Source Localization using Mobile Robots. PhD Thesis, École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, February 2010.

57. Rocco, M.D.; Reggente, M.; Saffiotti, A. Gas Source Localization in Indoor Environments using Multiple Inexpensive Robots and Stigmergy. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, 25–30 September 2011; pp. 5007–5014.
58. Li, F.; Meng, Q.H.; Bai, S.; Li, J.G.; Popescu, D. Probability-PSO Algorithm for Multi-robot Based Odor Source Localization in Ventilated Indoor Environments. In *Intelligent Robotics and Applications*; Xiong, C., Huang, Y., Xiong, Y., Liu, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5314, pp. 1206–1215.
59. Zou, Y.; Luo, D. A Modified Ant Colony Algorithm Used for Multi-robot Odor Source Localization. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*; Huang, D.S., Wunsch II, D.C., Levine, D.S., Jo, K.H., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5227, pp. 502–509.
60. Liu, Z.; Lu, T.F. Multiple Robots Plume-Tracing in Open Space Obstructed Environments. In Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO), Guilin, China, 19–23 December 2009; pp. 2433–2439.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).