Chapter 9

On the Use of Evolutionary Algorithms for Localization & Mapping

Infrastructure Monitoring in Smart Cities via Miniaturized Autonomous Sensory Agents

Ahmed Hallawa[1], Stephan Schlupkothen[1], Giovanni Iacca[2], Gerd Ascheid[1]

ABSTRACT

Miniaturized Autonomous Sensory Agents (MASAs) can play a pivotal role in smart cities of tomorrow. From monitoring underground infrastructure such as pollution in water pipes, to exploration of natural resources such as oil and gas. These smart agents will not only detect anomalies, they are expected to provide sufficient data to facilitate mapping the detected anomalies, while–cleverly–adopting their behaviour based on the changes presented in the environment. However, given these objectives and due to MASA's miniaturization, conventional designing methods are not suitable to design MASA. On one hand, it is not possible to use the widely adopted Simultaneous Localization and Mapping (SLAM) schemes, because of the hardware limitations on-board of MASA. Furthermore, the targeted environments for MASA in a smart city are typically GPS-denied, hardly accessible, and either completely or partially unknown. Furthermore, designing MASA's hardware and their autonomous on-line behaviour presents an additional challenge. In this chapter, we present a framework dubbed as Evolutionary Localization and Mapping (EVOLAM), which uses Multi-objective Evolutionary Algorithms (MOEAs) to tackle the design and algorithmic challenges in using MASAs in monitoring infrastructure. This framework facilitates offline localization and mapping, while adaptively tuning offline hardware constraints and online behaviour. In addition, we present different types of MOEAs that can be used within the framework. Finally, we project EVOLAM on a case-study, thus highlighting MOEA effectiveness in solving different complex localization and mapping problems.

KEYWORDS

Evolutionary Computation, Miniaturized Autonomous Sensory Agents, Localization, Mapping, Infrastructure Monitoring

1. Chair for Integrated Signal Processing Systems, RWTH Aachen University, Germany.
2. Department of Information Engineering and Computer Science, University of Trento, Italy.

9.1 INTRODUCTION

Smart cities of today are realized, and envisioned to continue to be adaptive to dynamically changing circumstances (Song et al., 2017). In other words, reacting to the system state optimally to provide high quality products and services to its inhabitants; This should be achieved with the least possible consumption of resources, while meeting other preset requirements. To realize this, a smart city includes a huge mesh of sensors distributed over multiple systems. This huge mesh includes static sensors and a wide range of possible active agents such as Unmanned Air Vehicles (UAVs), Autonomous Underwater Vehicles (AUVs), Autonomous Surface Vehicles (ASVs), Unmanned Surface Vessels (USVs) or Remotely Operated underwater Vehicles (ROVs), in addition to kinetically passive agents such as Miniaturized Autonomous Sensory Agents (MASAs). This network of sensors and robots are responsible for taking the adequate actions to produce the close-to-optimal output given the current state of the system. For example, MASAs may detect leakage while monitoring a water distribution network, and this then leads to AUVs moving towards in order to stop it or at least contain it. Consequently, all these agents must work together like a single living organism, which is extremely challenging. Multi-objective Evolutionary Algorithms (MOEAs) are a set of meta-heurstic algorithms that are suited for such problems.

9.1.1 Monitoring Infrastructure

One application, in the context of smart cities, is monitoring infrastructures. It is a critical task because it is linked to virtually most resources running in the city such as sewage, water, gas and petrol. However, monitoring different types of infrastructure presents hurdles. For example, most of infrastructure networks to be monitored are typically vast, e.g. a water network. Consequently, relying on static sensors in such scenarios is not efficient because it is extremely costly due to installation costs. Furthermore, they are not easily reusable in different environments and, thus, not adequate. In addition, many cities already installed their infrastructure without setting up sufficient number of static sensors and would still want to adopt smart control and monitoring. As a result, we project an increase in the use of MASAs. These agents are re-usable for different systems, e.g. water and waste pipes, moreover, they are usable indoors as well due to their miniaturized size. And as stated earlier, this is typically realized with the presence of other types of kinetically active agents (UAV, AUV, USV, ROV etc.).

However, the use of MASAs in monitoring infrastructure present challenges: Firstly, due to their scaled down size they are limited in resources. Specially, it is expected that those agents would be operating for relatively long time, as typical infrastructures are vast. This requires an optimization process on their offline tunable attributes such as communication range and/or communication

rate, in addition to their online behaviour, e.g. choosing adequate compression based on the changes in the environment such as to use lossy compression when the environment is not changing much, while using a lossless compression when the environment is highly dynamic; This will lead to saving energy. In addition, their small size and limited resources implies that localization techniques such as simultaneous localization and mapping (SLAM) techniques are not a good fit due to their need for relatively high communication and computational requirements, which is not possible with small platforms.

Another challenge in monitoring infrastructures is that it requires multiple agents with different designs, from kinetically active agents with platforms capable moving within the environment, to passive sensors only reacting to the environment. And from relatively large ones, capable of conducting complex tasks such as repair, to cm-sized sensors with high mobility such as MASAs. Furthermore, in most cases, all these agents (sensors and robots) are manufactured and designed independently without a particular standard. As a result, this produces a heterogeneous system consisting of wide range of sensors and robots embedded with completely different hardware architecture, and despite their diversity, they must cooperate in harmony to achieve the set high level goal of monitoring infrastructure.

In this last challenge, one solution is to use a centralized system, where all agents are connected to a central control unit orchestrating the whole process. However, this solution is not efficient: Firstly, it requires high volume communication between the central control unit such as fusion center and all other agents in the system, which is not possible due to energy constraints on MASAs among other communication challenges. Secondly, in a centralized system, it is preferred that the used agents are designed in a homogeneous fashion, which is not usually the case.

On the other hand, using non-centralized systems that are working independently and where the intelligence is distributed over different sub-modules in the system is sub-optimal because it is not benefiting from the available data in each sub-system.

9.1.2 State-of-The-Art

The problem of having a heterogeneous system of robots and sensors attempting to achieve a high level objective while optimizing their available resources is addressed in many currently running projects. One project that attempts to tackle this problem is *Smart and Networking UnderWAter Robots in Cooperation Meshes* (SWARMs)[3].

In this project, researchers' main objective is to offer solutions for expanding the use of AUVs, ROVs, USVs . . . etc., in order to achieve the three fundamental designs tasks in robotics: conception, planning and execution. Their targeted

3. http://www.swarms.eu/

applications platform included maritime and offshore operations, however, in achieving this goal they provided many solutions that are extremely helpful to monitoring infrastructure such as the ROS based UUV simulator presented in (Manhães et al., 2016).

Another project tackling this problem is dubbed as *Phoenix*[4]. In Phoenix, researchers have found a middle ground between the use of centralized vs distributed solution to the heterogeneous agents problem by adopting a centralized approach used only for localization and mapping. The rest of the system is running in a distributed manner. Furthermore, they proposed conducting localization and mapping offline in order to suite MASAs' limited available online resources. In that regard, they have defined what is dubbed as the Centralized Offline localization and Mapping (COLAM) problem in (Schlupkothen et al., 2018). Furthermore, they have proposed a framework, to solve this problem, dubbed as Evolutionary Localization and Mapping (EVOLAM) framework in (Hallawa et al., 2017, Schlupkothen et al., 2018).

In addition to conducting offline localization and mapping, EVOLAM offers a methodology to conduct optimization of the tunable off-line and on-line parameters in MASAs using Evolutionary Algorithms (EAs), while attempting to achieve a high level task such as detecting anomaly in an unknown environment. This is a suited methodology for monitoring infrastructure in smart cities due to the following available attributes:

- EVOLAM adopts EAs for optimization. This is suited for the problem because the mathematical formalization between the tunable parameters in MASAs and the high level objectives is too complex and usually not available, i.e. it is a blackbox optimization problem.
- EVOLAM conducts localization and mapping offline, which is suited for MASAs limited available online resources.
- EVOLAM takes into consideration the reality gap associated with using a simulation.

9.1.3 Chapter Organization

Despite the fact that this chapter focuses on the of EAs for localization and mapping for infrastructure monitoring, the chapter first presents the EVOLAM framework, as it is the general layout that facilitates the use of MOEAs for this presented problem. The chapter, further, highlights the different types of MOEAs that can be adopted within this framework. Thus, the chapter is organized as follows: Firstly, a formal definition of the COLAM problem is presented in the next section. Afterwards, in Section 9.3 we shed light on the EVOLAM framework. This is followed with a dedicated section (Section 9.4) on the different types of MOEA possible to be used within EVOLAM. Finally, we project the framework on an illustrative problem, which is defined in Section 9.5.

4. https://phoenixh2020.wordpress.com/

Moreover, results are shown in Section 9.5.5 and finally, Section 9.6 concludes the chapter with closing remarks.

9.2 CENTRALIZED OFFLINE LOCALIZATION AND MAPPING

In the definition of COLAM problem, it is assumed that a group of agents are inserted to an environment $\mathcal{E}$ with the objective of collecting map-based data to obtain information about it. An environment contains a *Region of Interest* (ROI), which defines the domain of the required information. For example, an environment can be an underground river, and a ROI can be one branch of this river. Each agent i inserted in the environment collects a set of measurements at different time instances k, i.e. $m_{i,k}$, and undergo two-way time of flight ranging with each surrounding agent j, within range, to facilitate localization, i.e. collecting distance $d_{i,j,k}$. It is important to highlight that due to lack of resources of the used agents, they can not perform online localization and mapping. Furthermore, the information about the environment they are exploring is very limited, i.e. the exact boundaries of the ROI is not known.

The collected measurements are then processed in order to produce an estimated environment $\hat{\mathcal{E}}$. This environment includes an estimated ROI on which an estimated mapping function $\widehat{\mathcal{M}}_{\hat{\boldsymbol{x}},\tilde{\boldsymbol{p}}}$ is defined. This function relates estimated physical properties $\tilde{\boldsymbol{p}}$ such as pressure or temperature with estimated positions $\hat{\boldsymbol{x}}$. The main objective is to close the reality gap between the estimated environment $\hat{\mathcal{E}}$ and the true explored environment $\mathcal{E}$. The objective can be formulated as follows:

$$\min_{\boldsymbol{c}} \int_{ROI} \Phi\left(\widehat{\mathcal{M}}_{\hat{\boldsymbol{x}}(\boldsymbol{c}),\tilde{\boldsymbol{p}}(\boldsymbol{c})}(\boldsymbol{x}), \mathcal{M}(\boldsymbol{x})\right) d\boldsymbol{x}, \tag{9.1}$$

where Φ is a function to evaluate a metric that reflects the difference between the estimated map $\widehat{\mathcal{M}}_{\hat{\boldsymbol{x}}(\boldsymbol{c}),\tilde{\boldsymbol{p}}(\boldsymbol{c})}(\boldsymbol{x})$ and the real one $\mathcal{M}(\boldsymbol{x})$ over the ROI, $\boldsymbol{c}$ is the tunable configurations of the used agents, e.g. communication range. To achieve this objective presented in (9.1), many other sub-objectives have to be tackled. Firstly, the estimation of ROI–since it is unknown–has to be as close as possible to the real ROI. This can be achieved by having an accurate localization process, which is function of the agents' configuration ($\hat{\boldsymbol{x}}(\boldsymbol{c})$). Furthermore, the measured readings $m_{i,k}$, which produces the estimated physical properties $\tilde{\boldsymbol{p}}$ must be as close as possible to the real one. This estimation process is function of the agent's configuration ($\tilde{\boldsymbol{p}}(\boldsymbol{c})$). In addition, the mapping accuracy plays an important role, because it is not possible to gather all readings in all position in ROI. Mapping is used to generalize the information of the estimated physical properties on the whole ROI based on the collected measurements from the environment. Finally, since the reason we applied localization and mapping offline is mainly driven by the fact that the used agents have limited resources, then localization and mapping must be conducted with the least consumption of resources as possible.

EVOLAM framework is designed to help achieve all this objectives using

MOEAs. In the next section we give an overview on its constituting blocks.

9.3 EVOLUTIONARY LOCALIZATION AND MAPPING (EVOLAM)

In this section we shed light on what is dubbed as EVOLAM. This framework is designed to conduct optimization on MASA while fulfilling a monitor or investigation task on an unknown environment. This may include, not exclusively, detection of different anomalies such as pollution levels. In this section we start with an overview of the framework going through it block by block. Despite the fact that EVOLAM is dedicated to solve COLAM, there are further assumptions set in order to ensure that EVOLAM is sufficiently general and thus can be used on wide range of scenarios. These assumptions are as follows:

1. The environment under investigation is unknown, i.e. very limited information about it is known beforehand (including ROI).
2. The environment under investigation is too complex to explore without optimizing the used agents.
3. The environment under investigation is GPS denied.
4. The used agents are limited in resources.

It is important to highlight that EVOLAM facilitates the use of MOEAs as it fits the problem definition, however, it does not specify which MOEA to use, details regarding the MOEA block in EVOLAM is in Section 9.4. Next the an overview on EVOLAM is presented.

9.3.1 EVOLAM Overview

A block diagram of the framework is shown in Figure 9.1. The framework starts with a real run where the agents are inserted in the unknown environment. Since we are dealing with a completely unknown environment, it can not be assumed that we can establish a simulating environment without conducting first one real run on the real environment.

Furthermore, the settings of the tunable parameters of the agents used in this exploratory run are picked with the objective to gather as much as possible information that would help in establishing a rough estimate of the real environment. Thus, the objective of this first real run is *not* to detect anomalies, since the agents are not optimized yet to do that, and the general assumption is it will not be possible to detect fully such anomalies without optimizing the agents adequately for this task. Thus, the main objective of this real run is to help establishing a rough estimate of the environment's general properties to be able to create a simulation for it. For obvious reasons, this real run is followed by post processing.

It is important to highlight that we do not assume that this alone will be able to fulfill the task of producing an estimate for the environment. However, as stated in the introduction, we assume that this system is used on wide range

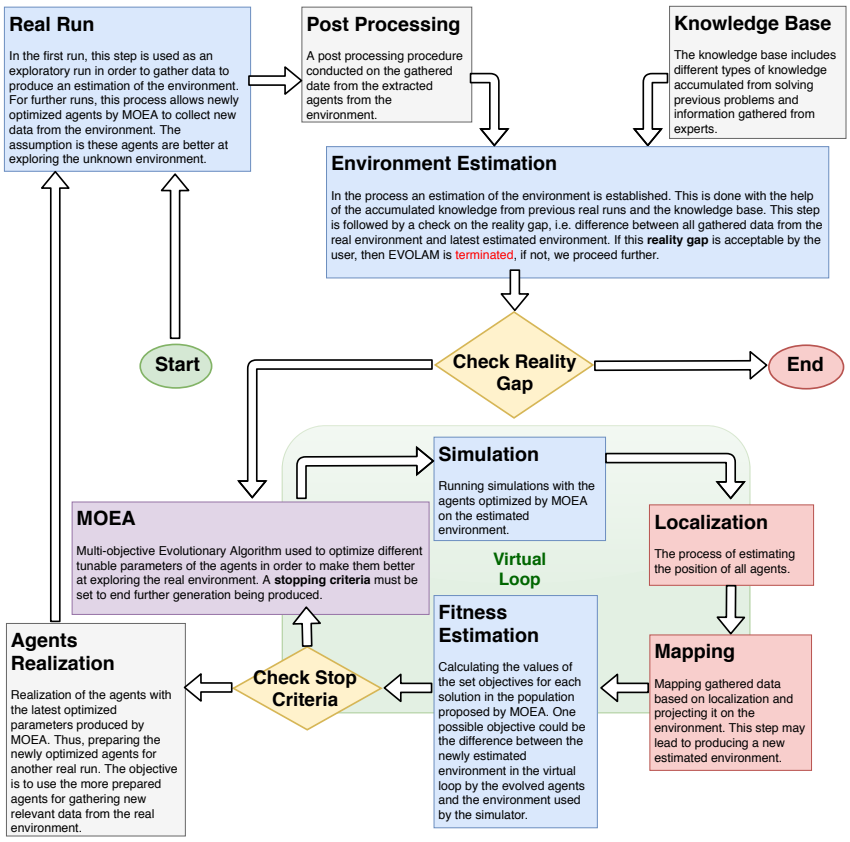


FIGURE 9.1 EVOLAM Block Diagram

of applications to decrease costs. Consequently, it is valid to assume we would have a knowledge base accumulated by running the system multiple times, which as a result can provide with some data that can help establishing a simulating environment along with the collected data from the environment.

The next block is a *check reality gap* block. This block is designed to check how good is our estimated environment relative to the real one. In other words, is the gap between the real collected data and the estimated environment is close enough to stop and exit the system or not? Obviously, in the first run, this will return false because we do not have any thing to compare it with. This "false" state will trigger the initialization of the virtual loop.

The virtual loop objective is to find the best agent configurations to detect the unknown environment anomalies while performing well relative to predetermined objectives and/or constraints such as energy consumption or mapping accuracy. The virtual loop consists of five steps: Multi Objective Evolution-

ary Algorithm (MOEA), simulation, off-line localization, mapping, and finally fitness estimation. Next we shed light on each module in the virtual loop.

9.3.2 MOEA

Due to the importance of this block, the next section is dedicated to cover different MOEAs. Generally, MOEAs belongs to the family of Evolutionary Algorithms (EAs). EAs are a general purpose meta-heuristic set of algorithms. These algorithms are well suited for non-convex optimization problems. Furthermore, EA has proven to perform well on black-box optimization problems, where the mathematical formalization between the tunable parameters and the defined objectives is not available. Needless to mention that our targeted set of problems, i.e. exploring unknown environments with recourse limited agents, fall under the black-box optimization category.

9.3.3 Simulation

The objective of this block is to run simulations with the agents optimized by MOEA on the estimated environment. The simulation process will then used for further analysis on the performance of the proposed solutions by MOEA. However, simulation is one of the challenging parts EVOLAM because fluid dynamics are extremely complex to model and most simulators fall between too complex but slow, and too simplified but fast. One possible tool for Computation Fluid Dynamics (CFD) is OpenFOAM[5]. OpenFOAM is open source and free. However, having OpenFOAM in the loop is computational exhaustive. Another possibility is to use Robot operating System (ROS)[6].

9.3.4 Localization

The localization is essential in the sense that without accurate positioning information, the measurements recorded by the agents are of very limited use. To ensure localizability in the case of pipeline or groundwater applications, the agents are equipped e.g. with ultra-sonic transceivers with facilitate ToF measurements among agents. However, a global localization based on agent-to-agent measurements is not possible unless sufficient beacons are present in the system. These beacons are usually static and with known absolute position, such that the relative distance-based information gathered via ToF measurements among beacons, is enriched by agent-to-beacon measurements. These beacons provide, due to their known absolute position, references for the localization process. Consequently, a successfully and accurate localization in an unknown environment requires an optimization of:

5. https://www.openfoam.com/
6. http://www.ros.org/

- The number of beacons and their positions, as graph-theoretical constraints must be met to ensure global localizability.
- The number of agents, as too few beacons in some areas of the environments can be compensated to some extend by more agent-to-agent measurements.
- The agents' transmit power, as connectivity plays a pivotal role to ensure localizability.

The localization itself is obtained in the fusion center after all environment and time-of-flight measurements have been gathered. Thus, requiring efficient algorithms that are able to handle large scale optimization problems.

9.3.4.1 Measurement Model

The inaccuracy as introduced by the nature of time-of-flight measurements, is subsequently be modeled as:

$$\tilde{d}_{i,j} = d_{i,j} + \eta, \qquad \eta \sim \mathcal{N}\left(0, \sigma_\eta^2\right), \tag{9.2}$$

where the actual distance $d_{i,j}$ is computed from the the ToF measurements

$$d_{i,j} = \frac{v_s}{2}\left(\tau_{i\to j} + \tau_{j\to i}\right), \tag{9.3}$$

where v_s is the speed of the signal in the considered medium and $\tau_{i\to j}$ is the ToF of the signal from i to j.

9.3.4.2 Motion Model

In the localization, the motion of the agents is assumed to follow a stochastic motion model which is derived from limited-order kinematics. More precisely, the following state vector is consider for each agent at time instance k:

$$\mathbf{x}_k = \begin{bmatrix} x_k & y_k & v_k & \phi_k & \omega_k \end{bmatrix}^\mathsf{T}, \tag{9.4}$$

where speed v and heading angle ϕ are defined in continuous time by

$$v(t) = \sqrt{\dot{x}^2(t) + \dot{y}^2(t)}, \qquad \phi(t) = \arctan\left(\frac{\dot{y}(t)}{\dot{x}(t)}\right). \tag{9.5}$$

Under the assumption that the acceleration and angular acceleration is zero-mean Gaussian, the following non-linear state evolution model can be derived (Li and Jilkov, 2003):

$$\mathbf{x}_k = f\left(\mathbf{x}_{k-1}\right) + \nu_k \tag{9.6}$$

$$= \begin{bmatrix} x_k + \frac{2v_k}{\omega_k}\sin[\frac{\omega_k T}{2}]\cos[\phi_k + \frac{\omega_k T}{2}] \\ y_k + \frac{2v_k}{\omega_k}\sin[\frac{\omega_k T}{2}]\sin[\phi_k + \frac{\omega_k T}{2}] \\ v_k \\ \phi_k + \omega_k T \\ \omega_k \end{bmatrix} + \nu_k, \tag{9.7}$$

where process nooise ν_k is zero-mean Gaussian with covariance matrix

$$\Sigma_\nu = \text{blkdiag}\left[\text{diag}\left[\sigma_x^2, \sigma_y^2\right],\ T^2\sigma_{\dot{v}}^2,\ \sigma_{\dot{\omega}}^2 \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T^2 \end{bmatrix}\right]. \tag{9.8}$$

9.3.4.3 Maximum a Posteriori Optimal Localization

Under the assumption of the above mentioned models, which are henceforth rewritten in terms of density functions

$$p\left(\tilde{d}_{i,j} \,\middle|\, \mathbf{x}_i, \mathbf{x}_j\right) = \mathcal{N}\left(d_{i,j}, \sigma_\eta^2\right) \tag{9.9}$$

$$p\left(\mathbf{x}_{i,k} \,\middle|\, \mathbf{x}_{i,k-1}\right) = \mathcal{N}\left(f\left(\mathbf{x}_{k-1}\right), \Sigma_\nu\right), \tag{9.10}$$

the following Maximum a Posteriori optimal localization problem can be derived:

$$\widehat{\mathbf{X}}_{MAP} = \arg\max_{\mathbf{X}_{0:K}} p\left(\mathbf{X}_{0:K} \,\middle|\, \widetilde{\mathcal{D}}_{1:K}\right) \tag{9.11}$$

$$= \arg\max_{\mathbf{X}_{0:K}} p\left(\widetilde{\mathcal{D}}_{1:K} \,\middle|\, \mathbf{X}_{0:K}\right) p\left(\mathbf{X}_{0:K}\right) \tag{9.12}$$

$$= \arg\max_{\mathbf{X}_{0:K}} \left[\prod_{k=1}^{K} p\left(\widetilde{\mathcal{D}}_k \,\middle|\, \mathbf{X}_k\right)\right]\left[\prod_{k=1}^{K} p\left(\mathbf{X}_k \mid \mathbf{X}_{k-1}\right)\right] p\left(\mathbf{X}_0\right) \tag{9.13}$$

$$= \arg\max_{\mathbf{X}_{0:K}} \sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{E}} \log p\left(\tilde{d}_{i,j} \,\middle|\, \mathbf{x}_i, \mathbf{x}_j\right) \tag{9.14}$$

$$+ \sum_{k=1}^{K} \sum_{i\in\mathcal{A}} \log p\left(\mathbf{x}_{i,k} \,\middle|\, \mathbf{x}_{i,k-1}\right) + \sum_{i\in\mathcal{A}} \log p\left(\mathbf{x}_{i,0}\right),$$

where $\mathbf{X}_{k:l}$ is the collection of all agents' state vectors for the time period k to l. Moreover, $\widetilde{\mathcal{D}}_{k:l}$ is the collection of all measurements in the same time period and $\mathcal{A}$ is the set of all agents Equation (9.14).

9.3.5 Environment Mapping

Topology mapping based on localization is presented in literature in many works such as (Kim et al., 2010, Kortenkamp and Weymouth, 1994) and (Thrun and Bücken, 1996). However, in all these works there is the assumption that the used agents for mapping are not restricted in terms of on-line resources. For resource constrained cases, one widely used scheme in mapping is Vietoris–Rips (VR) complex. This is adopted in (Ahmed et al., 2015) and (Dirafzoon et al., 2014). In our work, (Hallawa et al., 2017), we conducted modifications on VR-complex (dubbed as Trajectory Incorporated VR complex (TIVR) and used it for mapping in the context of MASA and COLAM. In our adaptation of VR-complex, we took into consideration the trajectory of the agents as part of the simplex itself.

Generally, we start the mapping process given a set of position estimates $\hat{\boldsymbol{p}}_{i,t} \in \mathbb{R}^d$ from all agents i and at all time instances t from the localization process. Then, we introduce the following:

$$\mathcal{V} = \{\hat{\boldsymbol{p}}_{i,t} \mid \forall i, \forall t\}, \quad \mathcal{T}_i = \{(\hat{\boldsymbol{p}}_{i,t}, \hat{\boldsymbol{p}}_{i,t+1}) \mid \forall t\} \tag{9.15}$$

where $\mathcal{V}$ is the input vertex set and $\mathcal{T}_i$ set with the edges defining the estimated trajectory of agent i. As a result, the neighboring vertices set can be defined as:

$$\mathcal{S}_i^{d,\epsilon} = \left\{ \arg\max_{\Delta \subseteq \mathcal{V}} |\Delta| \;\middle|\; \forall u, v \in \Delta, u \neq v, \|u - v\|_2 \leq \epsilon \right\} \tag{9.16}$$

which is the set that includes all vertices where the pairwise distance among them is at most ϵ, i.e. all distances between agents that are less that ϵ. Consequently, the set of all these neighbor set is denoted as $\mathcal{V}_{\mathcal{H}} = \bigcup_i \mathcal{S}_i^{d,\epsilon}$. The map is then defined via the four-tuple $\mathcal{M} = (\mathcal{V}_{\mathcal{H}}, \mathcal{T}, \mathcal{H}, w)$, where $\mathcal{T} = \bigcup_i \mathcal{T}_i$ is the union of trajectory edges, motivated by TIVR, $\mathcal{H} = \bigcup_i \mathcal{H}_i$ and $w : \mathcal{S}_i^{d,\epsilon} \to \mathbb{R}$ is the weight function for each neighbor set $\mathcal{S}_i^{d,\epsilon}$. Moreover, $\mathcal{H}_i \subseteq \mathbb{R}^d$ is the convex hull of the points in $\mathcal{S}_i^{d,\epsilon}$: $\mathcal{H}_i = \text{Conv}(\mathcal{S}_i^{d,\epsilon})$.

Moreover, it is important to include certainty in mapping, because ending the virtual loop may rely on it. One obvious assumption is to set more agent dense spaces in the reconstructed map with higher certainty relative to less dense spaces. Consequently, we have introduced introduce a weight (certainty) function w as follows:

$$w(\mathcal{S}) = \frac{1}{2} \sum_{(u,v) \in \mathcal{S}^2, u \neq v} \|u - v\|_2^{-1} \tag{9.17}$$

Therefore, the final output map will be in the form of weighted hull representing spaces where the agents have explored including their trajectory, in addition to weights reflecting certainty for each of these hulls.

9.4 MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS (MOEAS)

In this section we give a quick overview on general typers of MOEAs. Generally, given k objective functions and j variables, the multi-objective problem can be formalized as follows:

$$\max/\min \{F_1, \ldots, F_k\} \tag{9.18a}$$

$$\text{s. t. } \boldsymbol{d} \in \mathcal{D} \tag{9.18b}$$

where $\{F_1, \ldots, F_k\}$ is the set of objective functions, $\boldsymbol{d}$ is a decision vector, and $\mathcal{D}$ is the variable space, where $\mathcal{D} \subset \mathbb{R}^j$. Note that:

$$(F_1(\boldsymbol{d}), \ldots, F_n(\boldsymbol{d})) \in \mathcal{O} \tag{9.19a}$$

where O is the objective space, where $O \subset \mathbb{R}^k$ and $k > 2$. To project this in monitoring infrastructure using MASA, the set of objectives could be minimizing the consumed energy, and minimizing the localization error, while the decision vector could include the range of each used agent and the ranging frequency for example. Abstractly, the main objective of any MOEA is to produce a Pareto front while:

1. Minimizing the distance to the true Pareto front (PF convergence)
2. Covering diverse parts of the Pareto front (PF diversity)

To achieve these objective there are many approaches, next we highlight some of the widely used approaches.

9.4.1 Dominance-Based MOEA

This category of MOEA uses dominance to decide which solutions are more fit than the other, which generally in EA context affects the production of the next generation of solutions. Generally, using dominance can be categorized into tree groups: Dominance rank, dominance count, and dominance depth.

In dominance ranking, the fitness is calculated based on how many solutions are dominating the solution under investigation. This has been adopted in MOGA as shown in (Murata and Ishibuchi, 1995) for example. On the other hand, in dominance count the fitness is based on counting the number of solutions which the solution under investigation is dominating. This is adopted for example in Strength Pareto Evolutionary Algorithm (SPEA) and its modified version in (Zitzler et al., 2001, Kim et al., 2004). In dominance depth, the fitness of a solution is based on which front the solution is located on. One popular algorithm based on dominance depth is Non-dominated Sorting Genetic Algorithm 2 (NSGA-2) shown in (Deb et al., 2002). Generally, dominance depth has proven superiority in performance over dominance rank and count, this is reflected on literature state-of-the-art. In our case-study in Section 9.5, we use NSGA-2 in the MOEA block in EVOLAM.

9.4.2 Indicator-Based MOEA

Modern MOEA literature proposed the use of indicators rather than dominance for comparing solutions. The idea behind using indicator is MOEA is to transform the Mutli-objective problem to a single objective problem by using this indicator as its fitness. Two widely adopted indicators are Hypervolume (HV) and $R2$ indicators.

To understand the HV indicator, an example is presented. In Figure 9.2, four solutions ($S_1 \ldots S_4$) are given. Each color represent the HV space each solution is contributing relative to a reference point. In this example we are dealing with $2d$, thus the HV space is simply a rectangle. The HV indicator represents the added HV space by a certain solution. For example, if we remove S_2 the overall

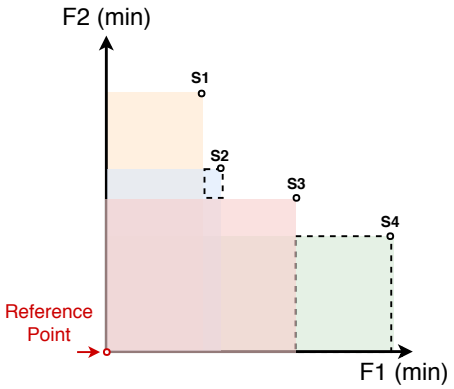


FIGURE 9.2 Hypervolume example

HV of the set of solutions will not be affected compared to S_4 for example. Note that the dotes square is not the HV itself but the added HV by that solution to the overall Pareto front. As shown, this indicator captures both diversity and fitness. In fact, only HV is known to be strictly monotonic, in other words, if given two sets of solutions m and n, where m is strictly dominating n, then HV indicator of m, $I_{HV}(m)$, will always be greater than $I_{HV}(n)$ as shown in (Beume et al., 2009, Zitzler et al., 2007, Emmerich et al., 2005).

HV indicator suffer from two main problems: It is computationally demanding with high number of objectives, and it has a bias w.r.t. regions with knee shaped form relative to other regions as presented in (Brockhoff et al., 2012).

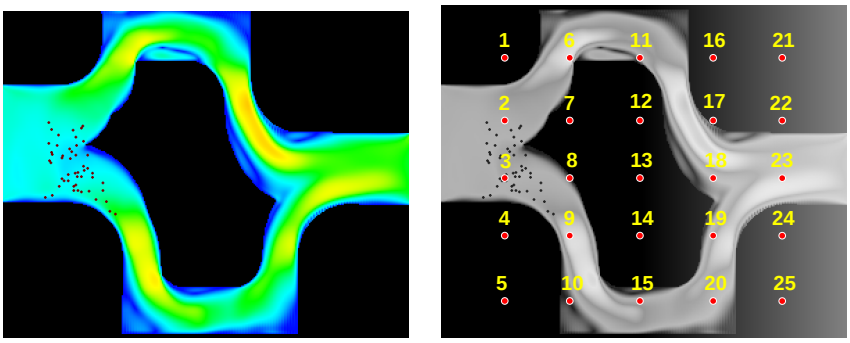
On the other hand, the $R2$ indicator is from the R indicator family presented in (Hansen and Jaszkiewicz, 1994). The $R2$ indicator uses utility functions that are faster to calculate, and less biased. Literature is rich with $R2$ indicator based algorithms and their analysis such as in (Zitzler and Künzli, 2004, Brockhoff et al., 2012)

Furthermore, there are other indicator based algorithms such as IBEA presented (Zitzler and Künzli, 2004), but they are less popular because they are out performed by HV and $R2$ indicator based algorithms.

9.4.3 Other Approaches

There are a lot of other approaches such as decomposition. This approach is based on the idea of transforming the multi-objecitve problem to to multiple single objective problems. One widely used example is (MOEA/D), which is introduced in (Zhang and Li, 2007). Furthermore, Non-dominated Sorting Genetic Algorithm III (NSGA-3), which adopts the idea of decomposition along with non-dominated sorting as shown in (Mkaouer et al., 2015).

9.5 CASE-STUDY



Agents Exploring the Environment (12m×12m) Possible Beacon Position (5 × 5)

FIGURE 9.3 Case-Study Environment

In this section we introduce the case-study. It is an illustrative problem we are using to project EVOLAM on and highlight its properties. Generally, the problem is exploration of an underground environment to detect changes exerted on an island in the middle of the a flow as shown in Figure 9.3a. The environment area is 12m × 12m. This environment is chosen for tests purposes for several reasons: Agents' trajectories have dynamic behaviour w.r.t. longitudinal and angular velocities, and acceleration. In addition, agents physically split in the environment into two different groups which makes the localization a more challenging task. Finally, the island has an irregular shape, which makes the mapping procedure more challenging. Next we formally define the problem statement.

9.5.1 Problem Statement

In this case-study, the objective is to reconstruct the original map (Figure 9.3) using a swarm of agents and a set of placed beacons. These beacon positions can take one or more of the possible available locations as shown in Figure 9.3b. Thus, the procedure goes as follows:

- Agent swarms are injected into the unknown environment (see Figure 9.3)
- There are two possible off-line tunable parameters on each agent and beacon: Ranging rate f_r and communication range R and both can not exceed $f_{r,\max}$ and $R_{\max}$, respectively.
- A set of beacons can be placed on the environment in one or more preset positions ($b_{p,max}$ positions spread equally on the environment).
- The number of used agents on the environment n_a is tunable and has to be between $n_{a,min}$ and $n_{a,max}$.

- The number of used beacons n_b is tunable and has to be between $n_{b,min}$ and $n_{b,max}$.
- The positions of the beacons $\boldsymbol{b}_p$ is tunable, but chosen from one or more of the possible $b_{p,max}$ positions on the environment.
- The problem has two objectives: Decreasing agent's energy ($E(R, f_r, n_a, \boldsymbol{b}_p)$), while increasing environment mapping quality ($F_{\text{map}}(\hat{\mathcal{E}})$).

It is important to highlight that agent's energy $E(R, f_r, n_a, \boldsymbol{b}_p)$ is directly associated with the tunable parameters R, f_r, n_a, $\boldsymbol{b}_p$, while mapping quality ($F_{\text{map}}(\hat{\mathcal{E}})$) is indirectly associated with them via localization. In addition, beacon positions b_p plays a pivotal role on the quality of localization and consequently mapping. Therefore, the objective of EVOLAM is to tune R, f_r, n_a, n_b and b_p in order to produce the Pareto front w.r.t two objective: Agent energy $E(R, f_r, n_a, \boldsymbol{b}_p)$ and mapping quality ($F_{\text{map}}(\hat{\mathcal{E}})$). Formally, the problem is defined as follows:

$$\max_{f_r, R, n_a, n_b, \boldsymbol{b}_p} F_{\text{map}}(\hat{\mathcal{E}}) \quad \text{and} \quad \min_{f_r, R, n_a, n_b, \boldsymbol{b}_p} E(R, f_r, n_a, \boldsymbol{b}_p) \tag{9.20a}$$

$$\text{s.t. } n_{a,\min} \leq n_a \leq n_{a,\max} \tag{9.20b}$$

$$n_{b,\min} \leq n_b \leq n_{b,\max} \tag{9.20c}$$

$$\boldsymbol{b}_p \in \{1, \ldots, b_{p,max}\}^{n_b} \tag{9.20d}$$

$$f_r \leq f_{r,\max}, \quad R \leq R_{\max} \tag{9.20e}$$

$$E_i(R, f_r, n_a, n_b) \leq E_{\max}, \ \forall i = 1, \ldots, n_a \tag{9.20f}$$

Next we formally define the objective functions: Agent energy and mapping quality.

9.5.2 Objective Functions

The objectives considered in this work account for the mapping quality F_{map} and the total energy consumption. Firstly, we define the swarm energy consumption by:

$$E(R, f_r, n_a, \boldsymbol{b}_p) = \sum_{i=1}^{n_a} \underbrace{\sum_t n_{i,t}(n_a, \boldsymbol{b}_p, f_r) \cdot T_r \cdot p_t(R)}_{E_i(R, f_r, n_a, \boldsymbol{b}_p)}, \tag{9.21}$$

where R is the communication range in meters, n_a is the number of agents and $\boldsymbol{b}_p$ the position vector of beacons. Note that n_b is implicit in $\boldsymbol{b}_p$. Moreover, T_r is the period of a single ranging pulse, f_r is the rate describing how frequently rangings are performed, $n_{i,t}$ is the number of ranging partners of agent i and time t and $p_t(R)$ is the transmit power required to achieve communication range R. In order to calculate the transmit power, the following equation has been used:

$$p_t(R) = p_{r,min} \cdot 10^{\frac{1}{10}\alpha R} \tag{9.22}$$

where α is the attenuation coefficient and $p_{r,min}$ is minimum receive power needed by the receiver for ranging. On the other hand, for mapping quality $F_{\text{map}}(\hat{\mathcal{E}})$, we adopted F_1-score proposed by (Powers, 2011). One reason behind this choice is that F_1-score is capable of capturing both precision and recall of the mapping process. In our adaptation of the F_1-score we assume the part of the environment with the fluid to be the area of interest, thus $F_{\text{map}}(\hat{\mathcal{E}})$ is set as follows:

$$F_{\text{map}}(\hat{\mathcal{E}}) = \frac{2|\mathcal{E}_f \cap \hat{\mathcal{E}}_f|}{2|\mathcal{E}_f \cap \hat{\mathcal{E}}_f| + |\mathcal{E}_{\bar{f}} \cap \hat{\mathcal{E}}_f| + |\mathcal{E}_f \cap \hat{\mathcal{E}}_{\bar{f}}|} \tag{9.23}$$

where $\mathcal{E}$ is the actual environment including both area of interest and non-interest, e.g. a fluid are $\mathcal{E}_f$ and non-fluid area $\mathcal{E}_{\bar{f}}$. On the other hand, $\hat{\mathcal{E}}$ is the estimated environment which includes both the area classified as fluid $\hat{\mathcal{E}}_f$ and the one classified as non-fluid $\hat{\mathcal{E}}_{\bar{f}}$. In the next section, we discuss the adopted MOEA.

9.5.3 Adopted MOEA

In this work we adopted multi-objective Non-Sorting Genetic Algorithm NSGA-2. The reason behind the use of NSGA-2 in the MOEA module is that the number of objectives is exactly two, swarm power and mapping quality. NSGA-2 does not perform well with objectives higher than two.

For the settings of NSGA-2 we adopted a population 20 individual per generation and ran all simulations with fixed computational budget of 15 generations, i.e. total number of evaluations per Pareto front is 225 (20×15) evaluations. For cross over and mutation, we used the Simulated Binary Crossover (SBX) operator explained in (Agrawal et al., 1995) and polynomial mutation operator.

Each solution offered by NSGA-2 adaptation answers the following questions:

1. How many agents (n_a) to be used for exploring the environment?
2. How many beacons (n_b) to be placed in the environment?
3. What are the positions of each beacon?
4. What is the agents' and beacons' ranging range (R), given that it is the same for all agents and beacons?
5. What is the agents' and beacons' ranging rate (f_r), given that it is the same for all agents and beacons?

9.5.4 Simulation

All simulations are conducted on MATLAB and Table 9.1 summarizes the simulation settings and NSGA2 initialization parameters. Furthermore, for communication between agents, no considerations regarding interference or multi-path is assumed. In addition, regarding mapping, the mapping technique presented in Section 9.3.5 is adopted, and for simplicity all non-zero weight regions are

TABLE 9.1 Simulation Parameters

Parameter	Value	Note
ϵ	2m	Mapping threshold, (9.16)
α	0.22dB/m	Attenuation coefficient (9.21) (Pinkerton, 1949)
σ_x	3.2×10^{-1}cm	Std. dev. in covariance matrix Σ_ν (9.8)
σ_y	7.1×10^{-1}cm	Std. dev. in covariance matrix Σ_ν (9.8)
S_{w_v}	3.2×10^{-3}cm/s^2	PSD of $\dot{a}(t)$ (9.8)
S_{w_ω}	3.2×10^{-2}rad/s^2	PSD of $\dot{\omega}(t)$ (9.8)
σ_η^2	0.01m	Std. dev. of measurement noise (9.2)
$n_{a,\text{min}}, n_{a,\text{max}}$	4, 10	Min./max. number of agents
$n_{b,\text{min}}, n_{b,\text{max}}$	2, 24	Min./max. number of beacons
$b_{p,max}$	25	Max available beacon positions
R_{max}	200cm	Maximum communication range

classified as fluid region.

9.5.5 Results

In this section we highlight the results. As explained, we have tested EVOLAM on a mapping task for environment shown in Figure 9.3 with different alowed upper limit for the used beacons $n_{b,max}$: From 2 beacons to 24 beacons as shown in Table 9.1. It is important to mention that we used steps of 2 when setting the values of maximum number of beacons, i.e. $n_{b,max} = \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24\}$. Consequently, the final output of the simulations is a Pareto front with 15 solutions for each possible $n_{b,max}$, thus 12 Pareto fronts in total. Each of these Pareto fronts has evolved over 20 generations, i.e. 20 runs on the virtual loop. Each solution on this Pareto front includes a value for each of following parameters: Agents' ranging range R, agents' ranging rate f_r, number of agents to be used n_a, number of beacons to be used n_b and the position for each beacon of the n_b beacons. Please note that all position are chosen from the 25 available positions distributed equally on the environment.

Figure 9.4 presents the Pareto optimal front for environment Figure 9.3 across different $n_{b,max}$ and interpolated over the rest of the $n_{b,max}$ axis to generate a surface for visualization purposes (green surface). Furthermore, the figure includes a projection of Pareto front surface on the three planes to help with the analysis.

Furthermore, Table 9.2 summarizes solutions from five Pareto fronts (at $n_{b,max} = 2, 8, 14, 20, 24$) out of the produced 12. From each Pareto front we present six solutions from the 15 solutions proposed by MOEA after 20 generations. It is important to highlight that $\bar{n}_s$ is the average of time instances, where rangings with surrounding agents has been established within the communication range R. Form Figure 9.4 and Table 9.2 the following points are

TABLE 9.2 Selected solutions from Pareto front for max beacons ($n_{b,max}$)= 2, 8, 14, 20, and 24

Solution ID	n_a	n_b	$\bar{n}_s$	R [cm]	Beacons Position IDs	E/T_r [mW]	F_{map} [%]
2a	10	2	6.38	191.7	4, 23	16.8	42.4
2b	10	2	6.31	191.5	4, 23	16.8	38.76
2c	10	2	6	192.6	2, 24	16	20.83
2d	9	2	7.07	193.6	5, 24	11	18.43
2e	10	2	9.46	82.6	6, 9	5.5	11.6
2f	6	2	9.72	116.4	6, 11	**1.7**	5.98
8a	10	5	7.1	179.9	3, 4, 12, 13, 14, 23	14.3	45.39
8b	10	5	6.96	181.7	3, 4, 12, 13, 14, 21	14.1	32.31
8c	10	8	8.75	138.3	1, 2, 4, 5, 6, 11, 18, 25	8.9	22.94
8d	10	7	8.7	122.1	1, 2, 5, 6, 11, 18, 24	7.5	18.43
8e	10	7	8.7	121.6	1, 2, 5, 6, 11, 18, 23	7.5	9.8
8f	10	7	8.8	121.5	1, 2, 5, 6, 11, 17, 22	7.3	7.2
14a	10	13	4	121.2	1, 2, 6, 9, 10, 11, 12, 13, 14, 15, 18, 20, 25	20.5	42.4
14b	10	12	9.77	158.1	1, 2, 6, 9, 10, 11, 12, 13, 14, 15, 18, 20	9	32.13
14c	6	11	9.9	138.2	1, 2, 6, 9, 10, 11, 12, 13, 14, 15, 18	2.5	22.09
14d	6	14	9.8	137	1, 2, 6, 9, 10, 11, 12, 13, 14, 15, 18, 20, 25, 22	2.5	19.57
14e	6	14	9.9	135.7	1, 2, 6, 9, 10, 11, 12, 13, 14, 15, 18, 20, 25, 22	2.5	14.06
14f	6	14	9.8	135.2	1, 2, 6, 9, 10, 11, 12, 13, 14, 15, 18, 20, 25, 22	2.5	11.7
20a	10	14	4.41	200	2, 3, 4, 5, 7, 8, 11, 12, 17, 18, 19, 20, 23	32.3	66.43
20b	10	15	4.34	194.3	1, 2, 3, 4, 5, 8, 11, 12, 13, 16, 17, 18, 19, 20, 23	31.2	54.83
20c	10	12	4.31	194.7	1, 2, 3, 4, 8, 12, 13, 15, 18, 19, 20, 22	30.9	45.18
20d	10	12	7.07	193.6	2, 3, 4, 5, 8, 12, 13, 17, 18, 19, 21, 22	23.1	41.07
20e	10	14	9.46	82.6	1, 3, 4, 6, 7, 8, 12, 13, 14, 17, 19, 21, 22, 23	7.6	32.74
20f	9	14	9.85	144.3	2, 3, 6, 7, 8, 9, 12, 15, 16, 17, 20, 21, 22, 25	6.6	25.45
24a	9	16	4.2	199.9	2, 5, 6, 7, 8, 9, 11, 12, 16, 17, 18, 19, 21, 22, 23, 24	26.8	**66.8**
24b	9	15	6.31	200	2, 5, 6, 7, 8, 9, 11, 12, 14, 16, 17, 19, 21, 22, 24	25.1	53.65
24c	10	18	7.44	187	1, 3, 4, 5, 6, 7, 8, 9, 10, 12, 16, 17, 18, 21, 22, 23, 24, 25	16.4	45.65
24d	8	14	4.21	134.7	2, 4, 5, 6, 7, 8, 9, 12, 13, 14, 16, 20, 21, 22	13	37.12
24e	9	15	6.95	94.7	3, 7, 9, 10, 11, 12, 15, 16, 17, 19, 20, 21, 22, 23, 24	3.5	31.08
24f	7	14	8.69	94.7	2, 4, 5, 6, 8, 9, 13, 16, 17, 19, 20, 21, 22, 23	2.7	24.39

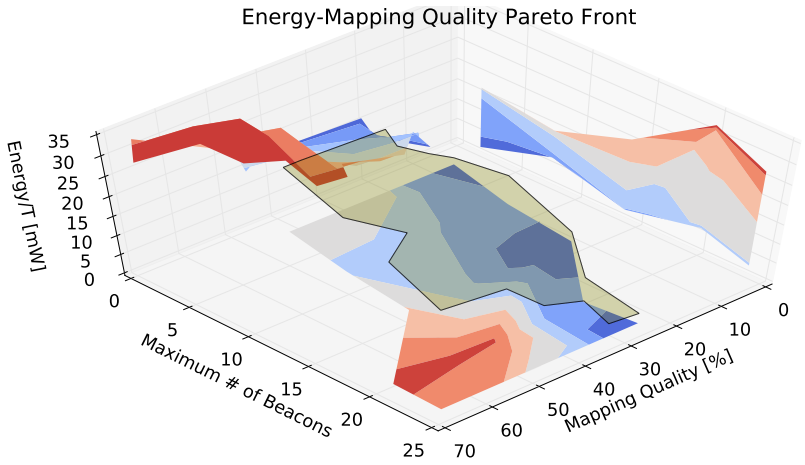


FIGURE 9.4 Energy-Mapping Quality Pareto Front with different $n_{b,max}$

concluded:

- The Pareto front is well distributed w.r.t the objectives (Energy and mapping quality), which is a positive indication regarding the performance of MOEA as it shows it was able to capturing diversity adequately.
- The mapping quality reaches values up to 66.8% (solution 24a in Table 9.2) and swarm power can go as low as 1.7 mW (solution 2f in Table 9.2).
- The areas with high swarm power consumption is associated with the areas with high $n_{b,max}$ and high mapping performance, which is consistent with theoretical grounds of the problem.
- Diversity range between mapping quality and swarm power consumption for each $n_{b,max}$ shifts in a non linear fashion, e.g. at $n_{b,max} = 14$.
- MOEA did not always converge to solutions where all the available $n_{b,max}$ are used. It proposed less n_b for solutions associated with low swarm power e.g. solutions 24e, 24f. Furthermore, in some cases it proposed solutions with less n_b but with better positions, which lead to higher quality mapping, e.g. solutions 8a, 14a, 20a and 24a.
- MOEA proposes different beacon positions, and thus exploits the symmetric nature of the environment, e.g. solution 2c vs 2f in Table 9.2.

As explained in Section 9.4, MOEA has a stochastic nature thus a performance analysis, therefore it is important to highlight that this whole process of generating the 12 fronts (one front per $n_{b,max}$) was performed 15 times in order to analyze statistically. In that regard, the performance metric hypervolume indicator highlighted in (Zitzler et al., 2007) is adopted. The indicator reflects the consistency of the convergence properties of the Pareto optimal front, and the diversity of the front. Table 9.3 has the summary of the indicator mean and

TABLE 9.3 Hypervolume indicator mean and variance for 15 runs

Environment	Hypervolume Indicator Mean [%]	Hypervolume Indicator Variance [%]
2 Beacons	65.23	3.2
8 Beacons	71.23	2.2
14 Beacons	78.45	3.4
20 Beacons	81.3	4
24 Beacons	81.4	4.3

variance performance. The relative point chosen is $(0.9, 2)$.

9.6 CONCLUSION

In this chapter we shed light on the use of evolutionary algorithms for localization and mapping for monitoring infrastructure in smart cities using miniaturized autonomous sensory agents. Firstly, we started with an overview on what we envision as the role different agents and robots can play in smart cities. Specifically, their role in infrastructure inspections for detecting anomalies such as pollution. Afterwards, we presented a framework dubbed as Evolutionary Localization and Mapping (EVOLAM), highlighting how it works and how each block contributes to the overall objective(s). The framework facilitates the use of Multi-objective Evolutionary Algorithm (MOEA) in localization and mapping. In that regard, a special focus was given on the use of MOEA. In a dedicated section, we highlighted the taxonomy of different MOEA paradigms highlighting different algorithms and their variations. Finally, we projected EVOLAM on case-study. Results show how effective the framework and the use of MOEA in solving such a complex mulit-objecitve problem.

BIBLIOGRAPHY

Agrawal, R. B., Deb, K. and Agrawal, R. (1995), 'Simulated binary crossover for continuous search space', *Complex systems* **9**(2), 115–148.

Ahmed, M., Karagiorgou, S., Pfoser, D. and Wenk, C. (2015), Map construction algorithms, *in* 'Map Construction Algorithms', Springer, pp. 1–14.

Beume, N., Fonseca, C. M., López-Ibáñez, M., Paquete, L. and Vahrenhold, J. (2009), 'On the complexity of computing the hypervolume indicator', *IEEE Transactions on Evolutionary Computation* **13**(5), 1075–1082.

Brockhoff, D., Wagner, T. and Trautmann, H. (2012), On the properties of the r2 indicator, *in* 'Proceedings of the 14th annual conference on Genetic and evolutionary computation', ACM, pp. 465–472.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002), 'A fast and elitist multiobjective genetic algorithm: Nsga-ii', *IEEE transactions on evolutionary computation* **6**(2), 182–197.

Dirafzoon, A., Betthauser, J., Schornick, J., Benavides, D. and Lobaton, E. (2014), Mapping of unknown environments using minimal sensing from a stochastic swarm, *in* 'Intelligent Robots

and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on', IEEE, pp. 3842–3849.

Emmerich, M., Beume, N. and Naujoks, B. (2005), An emo algorithm using the hypervolume measure as selection criterion, *in* 'International Conference on Evolutionary Multi-Criterion Optimization', Springer, pp. 62–76.

Hallawa, A., Schlupkothen, S., Iacca, G. and Ascheid, G. (2017), Energy-efficient environment mapping via evolutionary algorithm optimized multi-agent localization, *in* 'Proceedings of the Genetic and Evolutionary Computation Conference Companion', GECCO '17, ACM, New York, NY, USA, pp. 1721–1726.

Hansen, M. P. and Jaszkiewicz, A. (1994), *Evaluating the quality of approximations to the non-dominated set*, IMM, Department of Mathematical Modelling, Technical Universityof Denmark.

Kim, J., Zhang, F. and Egerstedt, M. (2010), 'A provably complete exploration strategy by constructing voronoi diagrams', *Autonomous Robots* **29**(3-4), 367–380.

Kim, M., Hiroyasu, T., Miki, M. and Watanabe, S. (2004), Spea2+: Improving the performance of the strength pareto evolutionary algorithm 2, *in* 'International Conference on Parallel Problem Solving from Nature', Springer, pp. 742–751.

Kortenkamp, D. and Weymouth, T. (1994), Topological mapping for mobile robots using a combination of sonar and vision sensing, *in* 'AAAI', Vol. 94, pp. 979–984.

Li, X. R. and Jilkov, V. P. (2003), 'Survey of maneuvering target tracking. part i. dynamic models', *IEEE Transactions on Aerospace and Electronic Systems* **39**(4), 1333–1364.

Manhães, M. M. M., Scherer, S. A., Voss, M., Douat, L. R. and Rauschenbach, T. (2016), UUV simulator: A gazebo-based package for underwater intervention and multi-robot simulation, *in* 'OCEANS 2016 MTS/IEEE Monterey', IEEE.
URL: *https://doi.org/10.11092Foceans.2016.7761080*

Mkaouer, W., Kessentini, M., Shaout, A., Koligheu, P., Bechikh, S., Deb, K. and Ouni, A. (2015), 'Many-objective software remodularization using nsga-iii', *ACM Transactions on Software Engineering and Methodology (TOSEM)* **24**(3), 17.

Murata, T. and Ishibuchi, H. (1995), Moga: Multi-objective genetic algorithms, *in* 'Evolutionary Computation, 1995., IEEE International Conference on', Vol. 1, IEEE, p. 289.

Pinkerton, J. M. M. (1949), 'The absorption of ultrasonic waves in liquids and its relation to molecular constitution', *Proceedings of the Physical Society, Section B* **62**(2).

Powers, D. M. (2011), 'Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation'.

Schlupkothen, S., Hallawa, A. and Ascheid, G. (2018), Evolutionary algorithm optimized centralized offline localization and mapping, *in* '2018 International Conference on Computing, Networking and Communications (ICNC)', pp. 625–631.

Song, H., Srinivasan, R., Sookoor, T. and Jeschke, S. (2017), *Smart cities: foundations, principles, and applications*, John Wiley & Sons.

Thrun, S. and Bücken, A. (1996), Integrating grid-based and topological maps for mobile robot navigation, *in* 'Proceedings of the National Conference on Artificial Intelligence', pp. 944–951.

Zhang, Q. and Li, H. (2007), 'Moea/d: A multiobjective evolutionary algorithm based on decomposition', *IEEE Transactions on evolutionary computation* **11**(6), 712–731.

Zitzler, E., Brockhoff, D. and Thiele, L. (2007), The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration, *in* 'Evolutionary multi-criterion optimization', Springer, pp. 862–876.

Zitzler, E. and Künzli, S. (2004), Indicator-based selection in multiobjective search, *in* 'International Conference on Parallel Problem Solving from Nature', Springer, pp. 832–842.

Zitzler, E., Laumanns, M. and Thiele, L. (2001), 'Spea2: Improving the strength pareto evolutionary algorithm'.

List of Figures