

UNIVERSITÀ DI BOLOGNA



School of Engineering
Master Degree in Automation Engineering

Distributed Autonomous Systems

TITLE

Professors:
Giuseppe Notarstefano
Ivano Notarnicola

Students:
Valerio Costa
Tian Cheng Xia

Academic year 2024/2025

Abstract

Contents

Introduction	5
1 Multi-Robot Target Localization	6
1.1 Gradient tracking with quadratic functions	6
1.1.1 Different graph patterns comparison	6
1.1.2 Comparison with centralized gradient	9
1.2 Cooperative multi-robot target localization	9
1.2.1 Different graph patterns comparison	10
1.2.2 Comparison with centralized gradient	11
1.2.3 Different noises	12
2 Aggregative Optimization for Multi-Robot Systems	14
Conclusions	15
Bibliography	16

Introduction

Motivations

Contributions

Chapter 1

Multi-Robot Target Localization

1.1 Gradient tracking with quadratic functions

The first part of the task consists of implementing the gradient tracking algorithm generalized in \mathbb{R}^n and then experiment with the implementation using quadratic functions, which we define in the usual way as:

$$f(\mathbf{z}) = \frac{1}{2} \mathbf{z}^T \mathbf{Q} \mathbf{z} + \mathbf{r}^T \mathbf{z} \quad \nabla f(\mathbf{z}) = \mathbf{Q} \mathbf{z} + \mathbf{r}$$

where $\mathbf{z} \in \mathbb{R}^d$, $\mathbf{Q} \in \mathbb{R}^{d \times d}$, and $\mathbf{r} \in \mathbb{R}^d$.

We analyzed the behavior of the quadratic function through the definition of different kinds of graph patterns (in particular the followings complete, binomial, cycle, star, and path graph). In addition, we performed a comparison between the distributed gradient tracking algorithm and the centralized one.

1.1.1 Different graph patterns comparison

The experiments with different graph patterns are done on the following configurations:

- A small problem (5 agents in \mathbb{R}^3),
- A problem with higher dimensionality (5 agents in \mathbb{R}^{15}),
- A problem with many agents (15 agents in \mathbb{R}^3), and
- A problem with many agents in higher dimensionality

Firstly, with the starting small problem, we can observe from Figure 1.1 a relatively smooth behavior of the cost function and an exponentially decreasing gradient in all cases. Moreover, a result that can be expected and

is consistently persistent in all the other experiments is that consensus is reached faster with a complete graph.

Next, by experimenting with higher dimensionality, we can observe from Figure 1.2 that the behavior of both the cost and its gradient are very similar to the previous case, indicating that the dimensionality is marginal in changing the difficulty of the problem.

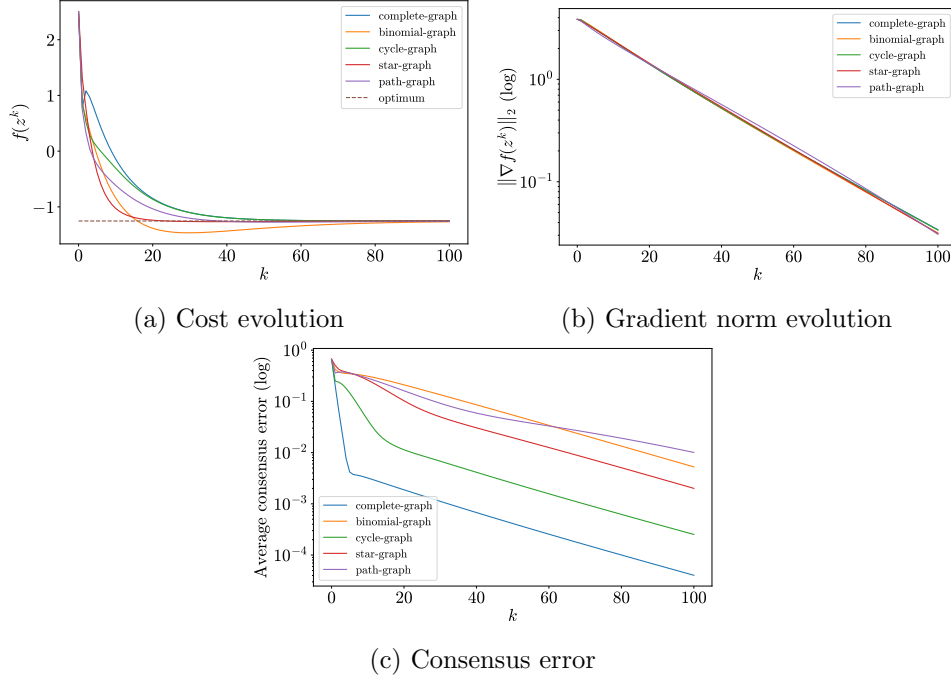


Figure 1.1: Configuration with 5 agents in \mathbb{R}^3

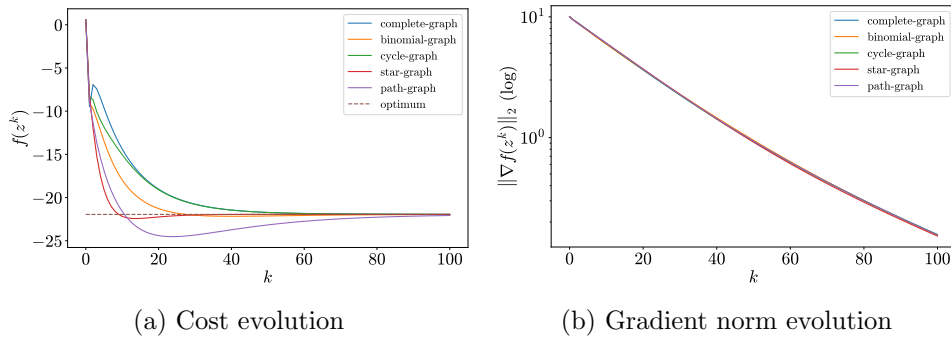


Figure 1.2: Configuration with 5 agents in \mathbb{R}^{15}

In the case of many agents with lower dimensionality, we can observe from Figure 1.3 that the cost function do not reach the optimum within

the limited number of iterations with the configurations that have a limited connectivity (e.g., path, star, and cycle graphs). Also, the gradient shows a change in terms of slope toward the end, symptom of a slow convergence. This can be explained as the fact that, by adding more agents, the overall problem includes more local losses and becomes more difficult to solve in a distributed way.

From Figure 1.4, we observe the same behavior as the previous case and also confirm that, with higher dimensionality, the problem is not significantly affected.

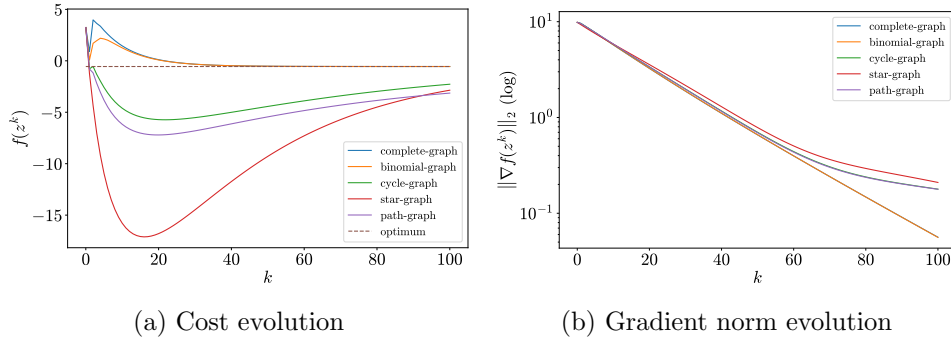


Figure 1.3: Configuration with 15 agents in \mathbb{R}^3

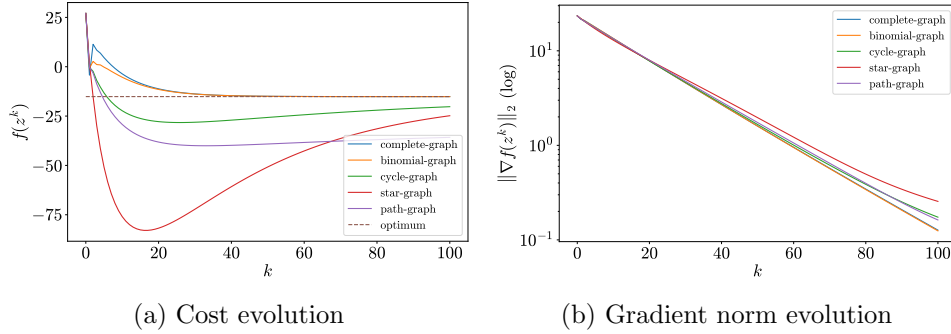


Figure 1.4: Configuration with 15 agents in \mathbb{R}^{15}

At last, we experimented with a higher number of iterations to analyze the behavior at convergence. From Figure 1.5, we can observe that the configuration with a complete graph is the one that converges with the norm of the gradient smallest to 0, while the worst performing is the path graph that is converging very slowly.

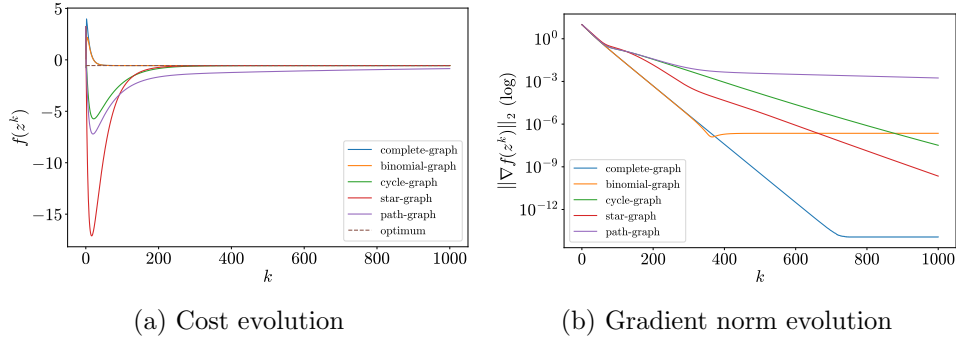


Figure 1.5: Configuration with 15 agents in \mathbb{R}^3 to convergence

1.1.2 Comparison with centralized gradient

Following the previous results, we select the configuration using the complete graph for the comparison with the centralized gradient method. In Figure 1.6, we can observe the results with 15 agents. It can be seen that, as one could expect, the centralized gradient method is faster to converge compared to a distributed algorithm as it has available all the global information.

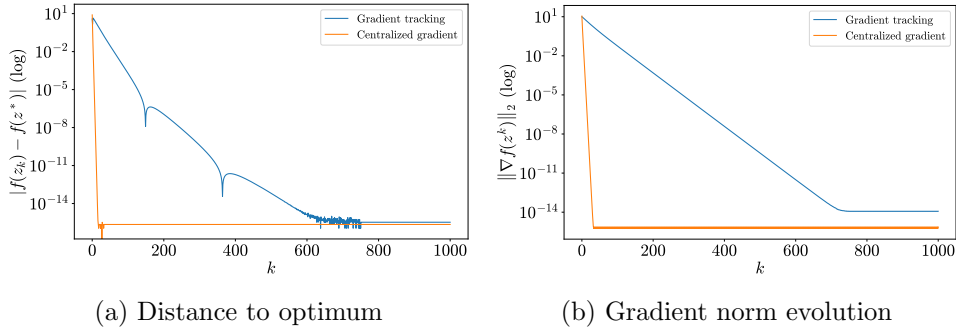


Figure 1.6: Configuration with 15 agents in \mathbb{R}^3 compared to centralized gradient

1.2 Cooperative multi-robot target localization

The second part of the task involves applying the gradient tracking algorithm for estimating the position of N_T fixed targets in a distributed way. Each tracking robot is located at position $\mathbf{p}_i \in \mathbb{R}^2$ and it is assumed that the distance measured from each robot is noisy. Given the position of the robot \mathbf{p}_i and of the target \mathbf{p}_τ , we model the measured distance $d_{i\tau}$ as follows:

$$d_{i\tau} = \|\mathbf{p}_i - \mathbf{p}_\tau\| + \varepsilon \cdot \text{noise}$$

where **noise** is drawn from some distribution and ε is the noise rate.

The local loss each robot i uses is the following:

$$l_i(\mathbf{z}) = \sum_{\tau=1}^{N_T} (d_{i\tau}^2 - \|\mathbf{z}_\tau - \mathbf{p}_i\|^2)^2 \quad \nabla l_i(\mathbf{z}) = \begin{bmatrix} -4 (d_{i\tau_1}^2 - \|\mathbf{z}_{\tau_1} - \mathbf{p}_i\|^2) (\mathbf{z}_{\tau_1} - \mathbf{p}_i) \\ \vdots \\ -4 (d_{i\tau_{N_T}}^2 - \|\mathbf{z}_{\tau_{N_T}} - \mathbf{p}_i\|^2) (\mathbf{z}_{\tau_{N_T}} - \mathbf{p}_i) \end{bmatrix}$$

where $\mathbf{z} = (\mathbf{z}_{\tau_1}, \dots, \mathbf{z}_{\tau_{N_T}}) \in \mathbb{R}^{2N_T}$ is the stack of decision variables of robot i containing the estimated positions of the targets $\mathbf{z}_\tau \in \mathbb{R}^2$ and $\nabla l_i(\mathbf{z}) \in \mathbb{R}^{2N_T}$ is the concatenation of the gradients computed at each target.

We approach the experimentation of such algorithm by trying different graph patterns and comparing with the centralized gradient method, similarly to the previous case. At first, we evaluated the performances of the algorithm in the following cases:

- Network of 5 robots and 1 target,
- Network of 5 robots and 3 targets, and
- Network of 15 robots and 3 targets.

Then, the focus switched to observe how much the performance changes in terms of noise. We experimented with:

- Varying Gaussian noise,
- Varying Poisson noise, and
- Varying noise rates.

1.2.1 Different graph patterns comparison

In terms of graph pattern, we can observe from Figure 1.7 and Figure 1.8 that with the same number of robots and increasing number of targets, the number of iterations required to converge is the same. This makes sense as each target is independent of the others.

Instead, by increasing the number of tracking robots, we can see from Figure 1.9 that the plateau is reached in fewer number of iteration, which intuitively means that more tracking robots help in reaching a faster convergence.

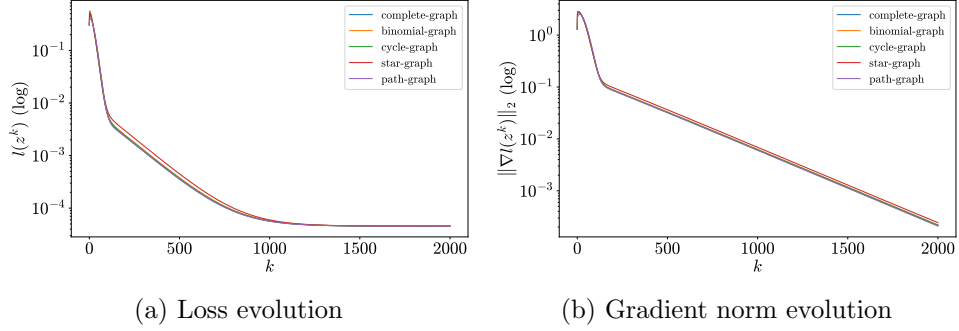


Figure 1.7: Configuration with 5 robots and 1 target

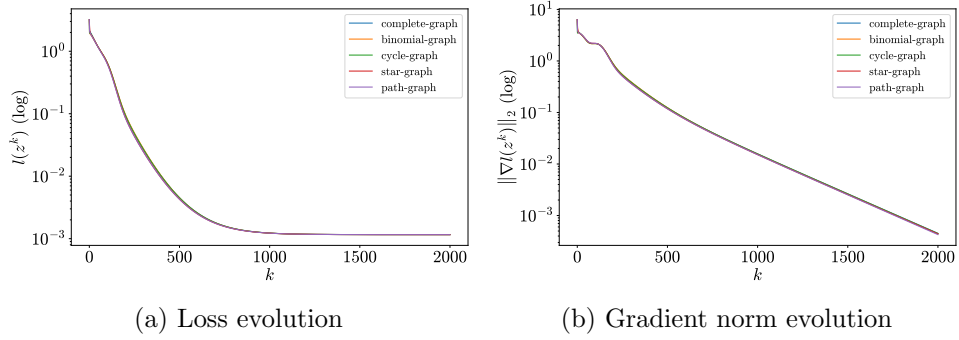


Figure 1.8: Configuration with 5 robots and 3 targets

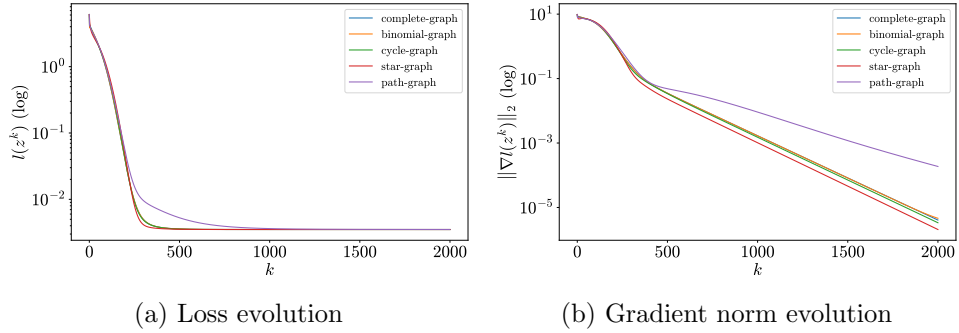


Figure 1.9: Configuration with 15 robots and 3 targets

1.2.2 Comparison with centralized gradient

The plots in Figure 1.10 confirms what we observed before in the case of quadratic functions, as the convergence speed is faster in the centralized approach.

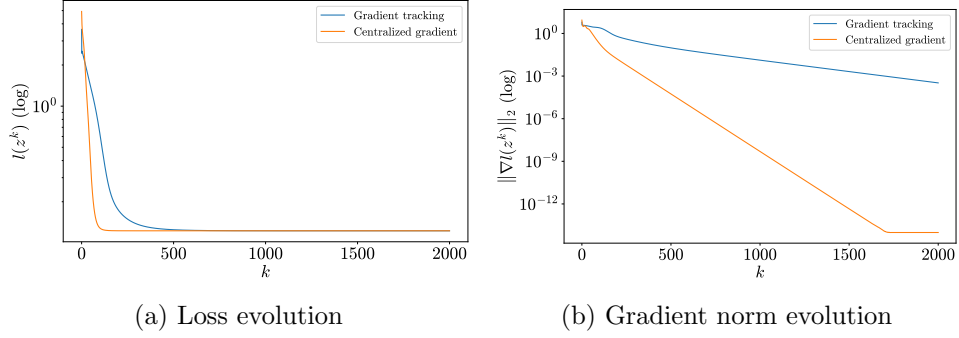


Figure 1.10: Configuration with 5 robots and 3 targets with centralized gradient

1.2.3 Different noises

From the experiments with different noises, we observed a worsening in performance that is proportional to the amount of noise injected into the distance measurement, implying as one could expect that more noise means worse results. This behavior is consistent with noise drawn from different distributions as in Figure 1.11 and Figure 1.12, and also when the noise rate is increased as in Figure 1.13.

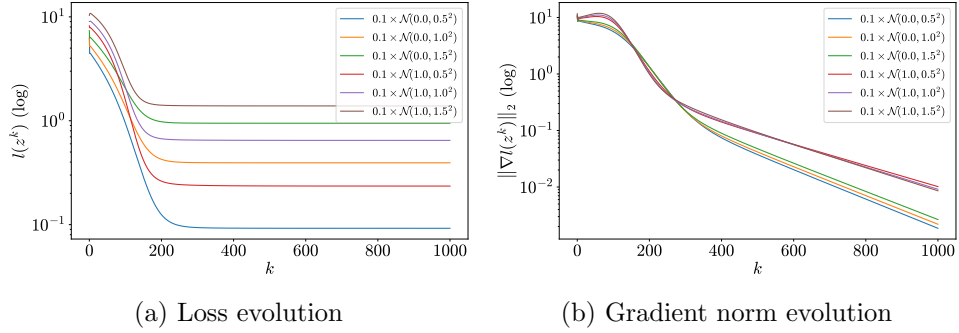


Figure 1.11: Configuration with 15 robots and 3 targets with noise drawn from Gaussian distributions

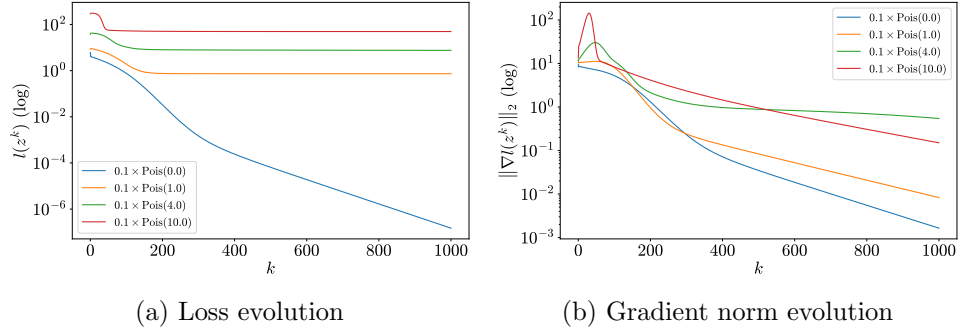


Figure 1.12: Configuration with 15 robots and 3 targets with noise drawn from Poisson distributions

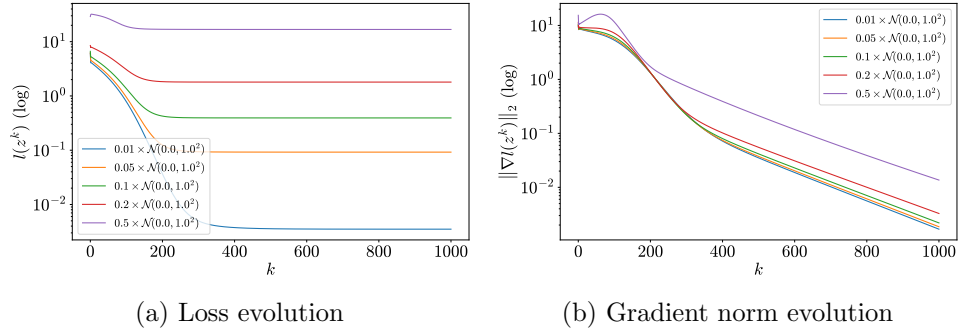


Figure 1.13: Configuration with 15 robots and 3 targets with different rates of Gaussian noise

Chapter 2

Aggregative Optimization for Multi-Robot Systems

Conclusions

Bibliography