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

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 $\mathbb{R}^d$ and then experiment with the implementation using quadratic functions, which we define in the usual way as:

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

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

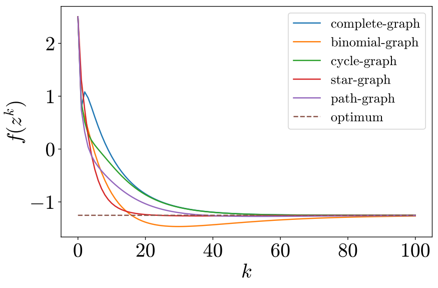
We analyzed the behavior with quadratic functions through the definition of different problems with different kinds of graph patterns (in particular complete, binomial, cycle, star, and path graph). The configurations we tested are the following:

- 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 (15 agents in $\mathbb{R}^{15}$).

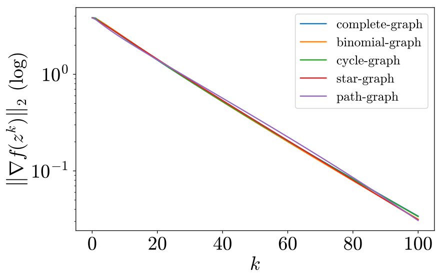
In addition, we performed a comparison between the distributed gradient tracking algorithm and the centralized one. For compactness in the discussion, in the rest of this report we show the results with a single initialization seed and, if not specified, it indicates that the results are consistent across different initializations.

1.1.1 Comparison between different graph patterns

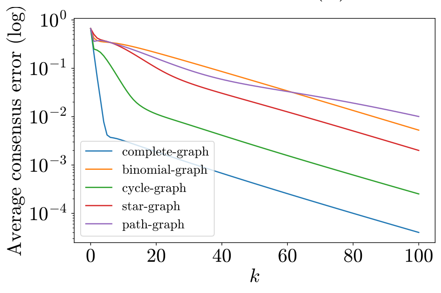
For the starting small problem, we can observe from Figure 1.1 a relatively smooth behavior of the cost function and an exponentially decreasing gradient in all cases. Moreover, a result that can be expected and is consistently persistent in all the other experiments is that consensus is reached faster 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.



(a) Cost evolution

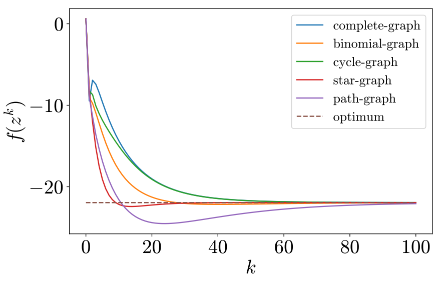


(b) Gradient norm evolution

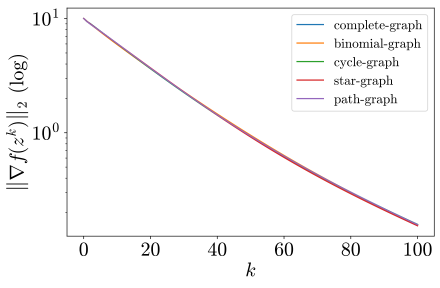


(c) Consensus error

Figure 1.1: Quadratic function minimization with 5 agents in $\mathbb{R}^3$



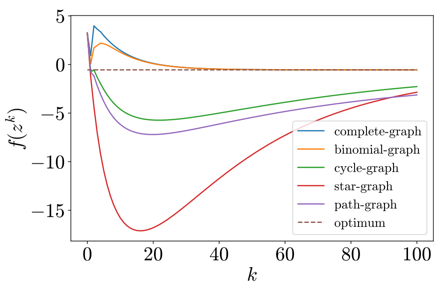
(a) Cost evolution



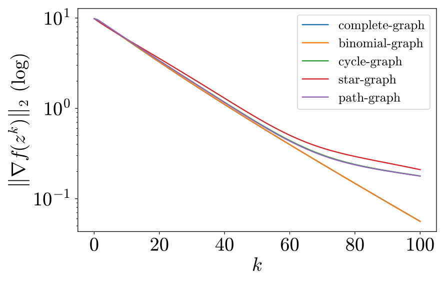
(b) Gradient norm evolution

Figure 1.2: Quadratic function minimization 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 does not reach the optimum within the given number of iterations with the configurations that have a limited connectivity (i.e., path, star, and cycle graphs). Also, the gradient shows a change in terms of slope toward the end, symptom of a slower convergence. This can be explained by 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 convergence behavior as in the previous case and also confirm that, with higher dimensionality, the problem is not significantly affected.

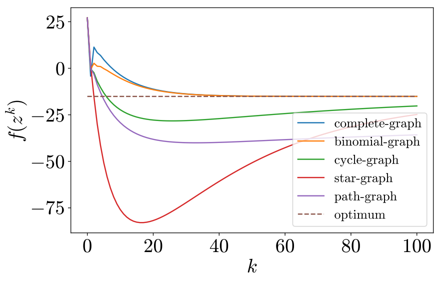


(a) Cost evolution

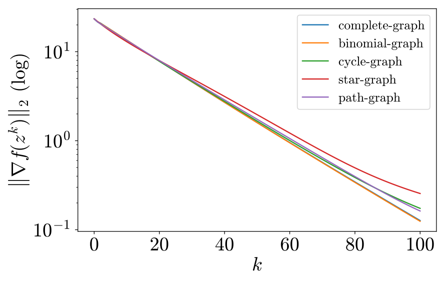


(b) Gradient norm evolution

Figure 1.3: Quadratic function minimization with 15 agents in $\mathbb{R}^3$



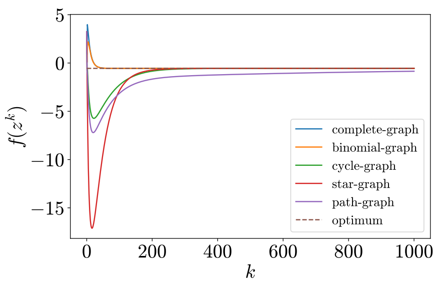
(a) Cost evolution



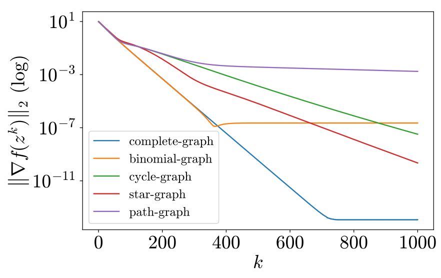
(b) Gradient norm evolution

Figure 1.4: Quadratic function minimization 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 most precise gradient, while the worst performing is the path graph that is the slowest to converge.



(a) Cost evolution

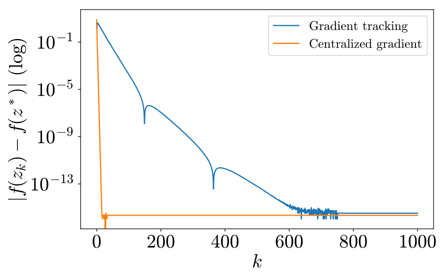


(b) Gradient norm evolution

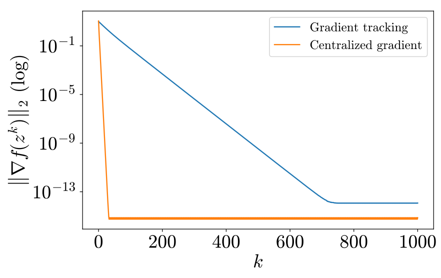
Figure 1.5: Quadratic function minimization with 15 agents in $\mathbb{R}^3$ to convergence

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, but the overall behavior is the same for all configurations. 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 and does not rely on information exchange with the neighbors and estimates.



(a) Distance to optimum



(b) Gradient norm evolution

Figure 1.6: Quadratic function minimization 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 through N_R tracking robots. Each robot is located at position $\boldsymbol{p}_i \in \mathbb{R}^2$ and it is assumed that the distance measured from each robot is noisy. Given the positions $\boldsymbol{p}_i$ and $\boldsymbol{p}_\tau$ of the i-th robot and the τ-th target, respectively, we model the measured noisy distance $d_{i,\tau}$ as follows:

$$d_{i,\tau} = \|\boldsymbol{p}_i - \boldsymbol{p}_\tau\| + \varepsilon \cdot \texttt{noise}$$

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

The local loss each robot i uses is the following:

$$l_i(\boldsymbol{z}) = \sum_{\tau=1}^{N_T} \left(d_{i,\tau}^2 - \|\boldsymbol{z}_\tau - \boldsymbol{p}_i\|^2\right)^2 \quad \nabla l_i(\boldsymbol{z}) = (\nabla l_{i,1}(\boldsymbol{z}_1), \ldots, \nabla l_{i,N_T}(\boldsymbol{z}_{N_T}))$$
$$\nabla l_{i,j}(\boldsymbol{z}_j) = -4\left(d_{i,j}^2 - \|\boldsymbol{z}_j - \boldsymbol{p}_i\|^2\right)(\boldsymbol{z}_j - \boldsymbol{p}_i)$$

where $\boldsymbol{z} = (\boldsymbol{z}_{\tau_1}, \ldots, \boldsymbol{z}_{\tau_{N_T}}) \in \mathbb{R}^{2N_T}$ is the stack of decision variables of robot i containing the estimated positions of the targets $\boldsymbol{z}_\tau \in \mathbb{R}^2$ and $\nabla l_i(\boldsymbol{z}) \in \mathbb{R}^{2N_T}$ is the concatenation of the gradients computed with respect to 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 performance of the algorithm in the following cases, all with the same type of noise:

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

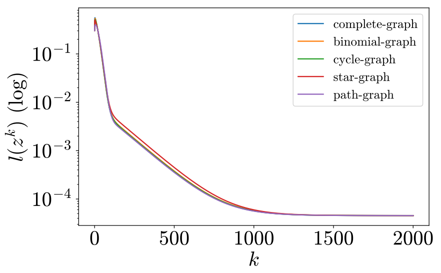
Then, the focus switched to observe how much the performance changes in terms of noise. By fixing the problem configuration, we experimented with varying Gaussian noises, Poisson noises and noise rates.

1.2.1 Comparison between different graph patterns

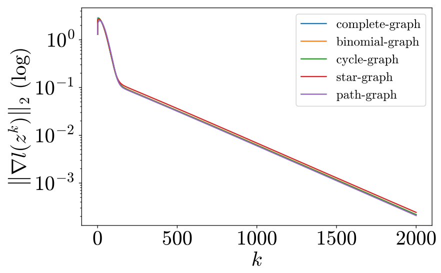
In terms of graph pattern, we can observe from Figure 1.7 and Figure 1.8 that with the same number of robots and increasing number of targets, the number of iterations required to converge is the same. This makes sense as each target is independent to the others and can be tracked in parallel.

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.

Moreover, as the total loss is a summation, we must note that the overall loss is higher in the case of more agent, but this does not indicate worse tracking results. We report in Figure 1.10 the average distance between the estimated and real target positions for a fixed configuration with varying number of robots. It can be seen, as intuition would suggest, that on average the tracking error becomes smaller by increasing the tracking robots.

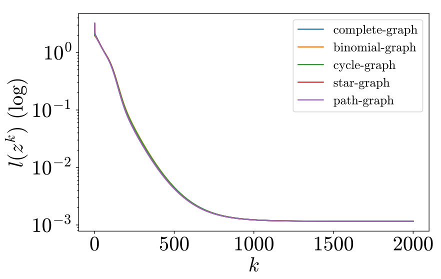


(a) Loss evolution

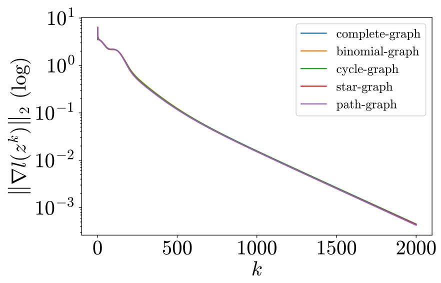


(b) Gradient norm evolution

Figure 1.7: Tracking with 5 robots and 1 target

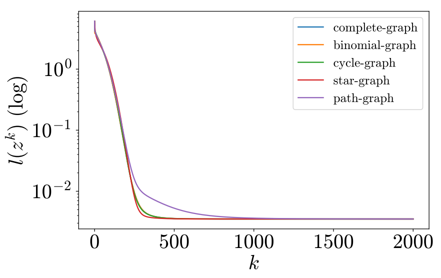


(a) Loss evolution

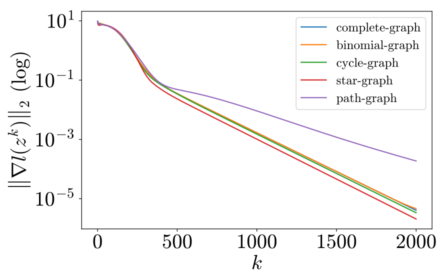


(b) Gradient norm evolution

Figure 1.8: Tracking with 5 robots and 3 targets



(a) Loss evolution



(b) Gradient norm evolution

Figure 1.9: Tracking with 15 robots and 3 targets

However, this behavior also shows a plateau, which can be explained by the presence of noise and numerical reasons.

Finally, an observation consistent in all experiments is that, as shown in Figure 1.11, the overall behavior of this system is to reach consensus in the first few iterations and then optimize the loss while improving and preserving consensus.

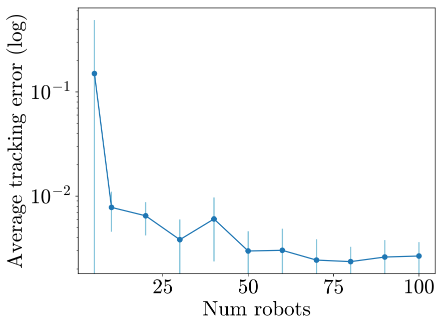


Figure 1.10: Average tracking error for different number of robots (average of 10 runs). Vertical bars represent the standard deviation.

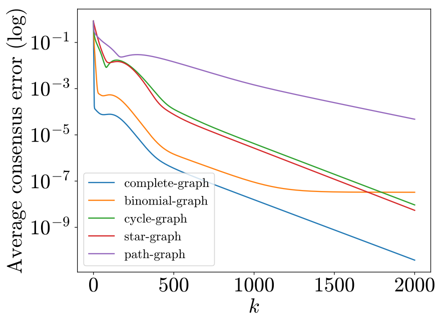
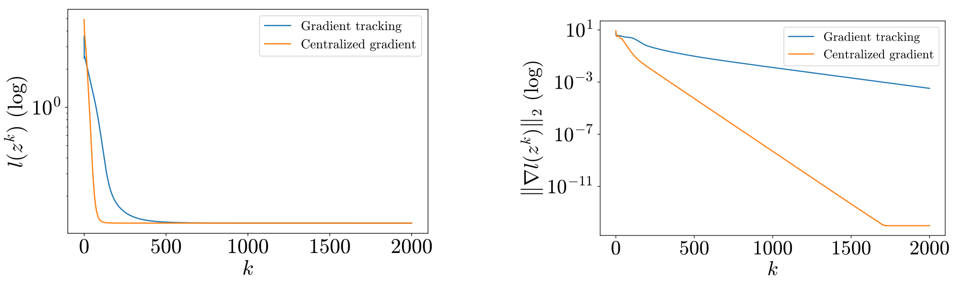


Figure 1.11: Tracking with 15 robots and 3 targets

1.2.2 Comparison with centralized gradient

Compared with the centralized gradient algorithm, the plots in Figure 1.12 confirm what we observed before in the case of quadratic functions. The convergence speed is faster and more accurate in a centralized approach, which also results in a lower average tracking error.



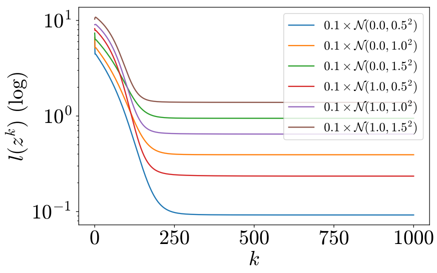
(a) Loss evolution

(b) Gradient norm evolution

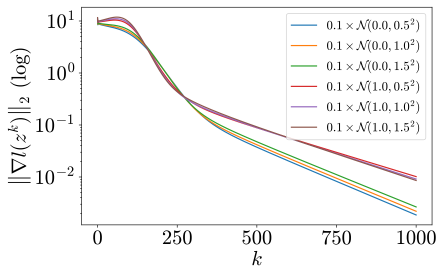
Figure 1.12: Tracking with 5 robots and 3 targets with centralized gradient

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 leads to worse results. This behavior is consistent with noise drawn from different distributions as in Figure 1.13 and Figure 1.14, and also when the noise rate is increased as in Figure 1.15.

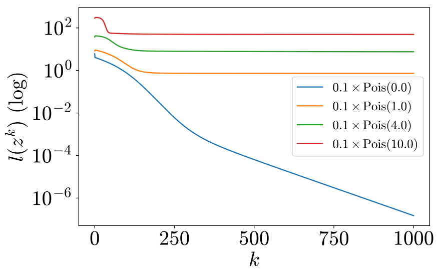


(a) Loss evolution

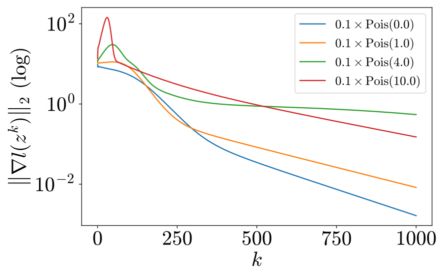


(b) Gradient norm evolution

Figure 1.13: Tracking with 15 robots and 3 targets with noise drawn from Gaussian distributions

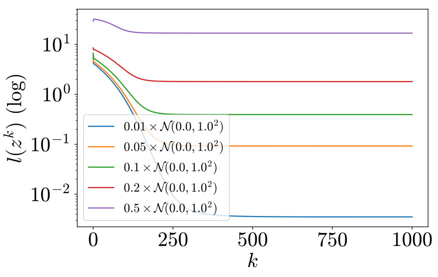


(a) Loss evolution

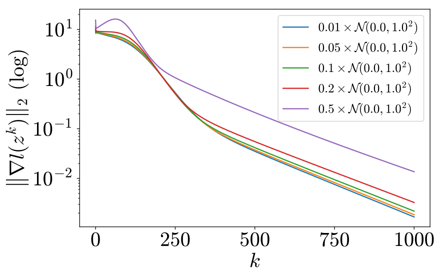


(b) Gradient norm evolution

Figure 1.14: Tracking with 15 robots and 3 targets with noise drawn from Poisson distributions



(a) Loss evolution



(b) Gradient norm evolution

Figure 1.15: Tracking with 15 robots and 3 targets with different rates of Gaussian noise

Chapter 2

Aggregative Optimization for Multi-Robot Systems

Conclusions

Bibliography