Multi-agent-based Structural Reconstruction of Dynamic Topologies for Urban Lighting

Félix Furger, Carole Bernon, Jean-Pierre Georgé, Nazim Pigenet, Paul Valiere

Multi-agent-based Structural Reconstruction of Dynamic Topologies for Urban Lighting

Félix Furger[1], Carole Bernon[1], Jean-Pierre Georgé[1(✉)], Nazim Pigenet[2], and Paul Valiere[2]

[1] IRIT, Université de Toulouse, CNRS, Toulouse INP, UT3, Toulouse, France
{felix.furger,carole.bernon,jean-pierre.george}@irit.fr
[2] Kawantech, Toulouse, France
{nazim.pigenet,paul.valiere}@kawantech.com
https://www.irit.fr/, https://www.kawantech.com/

Abstract. Until humanity succeeds in massively producing clean energy to satisfy its inexhaustible needs, one of its biggest challenges is to save and use its resources as efficiently as possible. With outdoor lighting being responsible for 2% of worldwide electricity consumption, smart urban lighting has recently gained a lot of attention in this respect. As an integrated part of smart cities, smart urban lighting rests on the anal-ysis of sensed data to tackle highly dynamical problems. This sensed data shapes a representation of the environment in which the smart system will have to perform. To reduce problem complexity, distributed solu-tions commonly apply local lighting policies and therefore benefit from the knowledge of the geographical positioning of the relevant streetlights in the environment. In this paper, we propose an adaptive multi-agent approach that aims at ensuring the robustness and coherence through time of the smart system's environment representation. Our approach leverages real time series data returned by streetlight sensors informing on vehicles and pedestrians traffic. We exploit this data to perform a structural reconstruction of the streetlight "fleet" topology without any *a priori* knowledge about its internal structure. We then ensure its cor-rectness through time by handling internal structure changes in order to continuously provide a coherent foundation for the smart lighting system to perform upon.

Keywords: Multi-agent systems · Self-adaptation · Structural network reconstruction · Smart lighting · Smart cities

1 Introduction

To combat the upcoming environmental crisis, our biggest challenge probably still resides in saving and using our resources as efficiently as possible. With outdoor lighting being responsible for 2% of worldwide electricity consumption, smart urban lighting recently gained some attention in this respect [15].

F. Dignum et al. (Eds.): PAAMS 2022, LNAI 13616, pp. 191–202, 2022.
https://doi.org/10.1007/978-3-031-18192-4_16

Smart Cities and Smart Lighting. Recent breakthroughs in light emitting diodes (LEDs) offered smart urban lighting the means it was lacking. Not only do LEDs offer long lifetime and low energy consumption compared to traditional streetlight technologies but they also provide much easier control opportunities [9]. First solutions benefited from the implementation of light sensors within the streetlights to strictly limit urban light use to low natural light situations. As an integrated part of smart cities however, smart urban lighting state of the art solutions nowadays rest on the analysis of more complex and informative sensed data [11] to adapt the light to different users: motion detectors can generate data informing about urban users' trajectories, their speed and their type (vehicle, cyclist or pedestrian). Moreover, in order to fully profit from such highly informative data and reduce problem complexity, an increasing number of distributed solutions focus on applying local context-aware lighting policies [9].

These solutions therefore benefit from the knowledge of the geographical positioning of the relevant streetlights in the environment. More specifically, in the already existing smart lighting solution that this research is aiming to support, each streetlight is not only capable of detecting the presence of a user in the street but also of communicating any relevant information to its nearest neighbors. This communication allows streetlights to preemptively light up before they even detect any user activity, in an effort to guarantee their comfort and safety.

To ensure fluid communication however, every streetlight needs to be aware of the identity, positioning and distance of its relevant neighbors. This requires a precise manual configuration at the time when the technology is deployed, which is hardly realistic considering how streetlights are deployed and maintained in practice. Moreover, if the neighborhood happens to change (failure, street works...) or if the sensors are moved or reset, this cumbersome configuration step has to be repeated. We therefore propose an automatic solution capable of determining each streetlight's local neighborhood.

Multi-agent Systems (MAS) are a decentralized approach, based on self-organisation mechanisms, where the calculation task is distributed over agents which are virtual or physical autonomous entities (more complete definitions of an "agent" are given in [12] and [5]). Each agent has only a local point of view of the problem it is solving, corresponding to a local goal. This particularity enables to easily distribute calculation tasks in the resolution process and consequently reduces computational costs. That is why multi-agent approaches are preferred where centralized approaches have limited flexibility and scalability.

Multi-Agent Systems are used in a wide variety of real-world application domains: optimization algorithms [3], power systems [8], complex networks and IoT [6], smart manufacturing [1], multi-robot systems [13].

We propose to adopt the Adaptive Multi-Agent Systems (AMAS) [2] theory where cooperation [7] is the engine that drives the behavior of an agent and the emergence of a global functionality.

2 Real-World and Simulated Environment

The research is conducted using both simulation software and real world data collected from already deployed streetlight smart sensors.

Kara Sensors. *Kara* is a smart urban lighting technology deployed in more than 40 cities across France and developed by Kawantech (www.kawantech.com). Their technology already guarantees a minimum energy saving of 55% on a LED streetlight since they ensure that streetlights fully light up only when user presence is detected. The system consists of an optical sensor such as a low definition camera combined with a calculator unit and is directly mounted into each individual streetlight. The smart system uses real time informative data about moving entities in the urban environment to monitor the streetlight's light level. Moreover, the solution implements a variety of telecommunication technologies such as Wi-Fi, LoRaWAN or DASH7. DASH7 ensures communication between *Karas* in order for streetlights to exchange information and potentially preemptively light up before they even detect any user activity.

In the scope of this paper, which objective is for each streetlight to be able to determine its local neighborhood, we leverage *Kara's* real time capabilities to inform about the presence of moving entities in the street, their positional tracking within the embedded camera's field of view as well as their speed and all associated timestamps.

Environment Simulation with GAMA. GAMA is a free, open-source simulation software designed for spatially explicit agent-based simulations development [4]. We use it to simulate the university campus in which the *Kara* technology is currently being deployed, as well as to generate and simulate various fictional urban lighting topologies. Fig. 1 shows such a fictional topology example in which simulated users can navigate and streetlights inform about real time user traffic just as if they were equipped with *Kara* sensors.

3 Objective and Methods

The objective of the research is to develop a solution that would lead each streetlight to identify and position its nearest neighbors without any prior knowledge of their identity. By taking advantage of time series data observed by *Kara* sensors, not only would the solution perform structural reconstruction of each streetlight's local neighbors topology but also continuously ensure its coherence through time. Motivated by the fact that the knowledge of each streetlight's neighborhood is the very foundation of the smart lighting system's functioning, the solution would aim at improving overall robustness through a more generic approach. It would therefore handle dynamic internal structure changes within local topologies, as well as estimate distances and relative positions between streetlights through time. Because of the dynamic and distributed nature of the problem, we tackle it with an Adaptive Multi-Agent System approach.

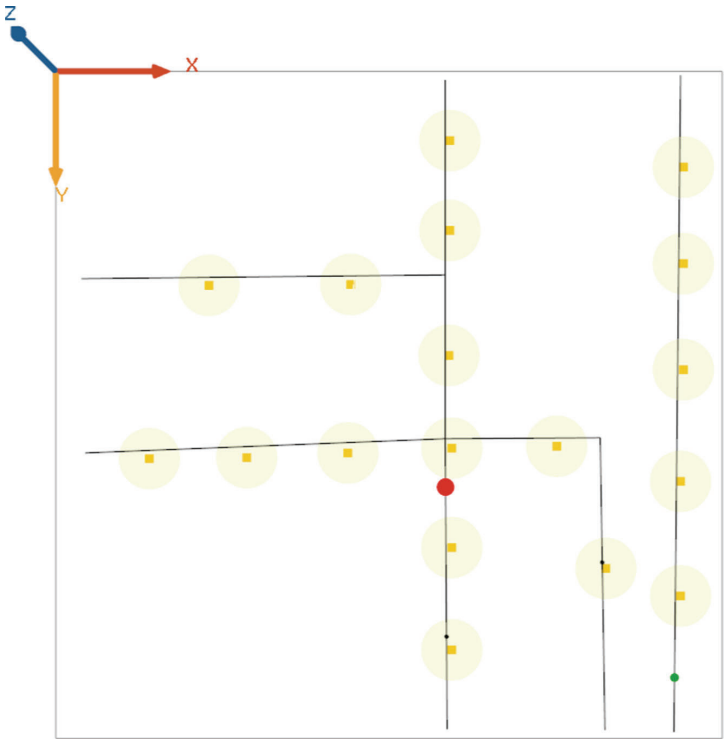

Fig. 1. Example of a fictional but realistic urban lighting topology simulation in GAMA. Yellow squares representing streetlights are positioned alongside traffic axes represented by black lines. Black, green and red circles respectively represent pedestrians, bikes and motorised vehicles (Color figure online)

3.1 Agent Modelling and Formalisation

We define an agent as an independent entity corresponding to a streetlight on which a *Kara* sensor has been mounted. An agent can therefore acquire data about user traffic happening in the street where it is located, and communicate its observations to other agents. The criticality of an agent is defined by how far an agent is from its local goal. In order to maintain a low level of criticality, each agent has to accurately identify and position its nearest neighbors in the environment. By comparing their respective sequences of observations and quantifying their correlation, two agents can gather knowledge about the likelihood of them being neighbours as well as, if applicable, their relative positioning in the neighborhood (distance and direction).

An agent is able to detect user activity in the street where it is located and can store this information in its local memory, hence forming time series data referred as O, an array of observations:

$$O = (o_1, o_2, o_3, ..., o_n) \qquad \text{with } o_i = (s_i, \overrightarrow{v_i}, t_i) \qquad i \in [1, n] \tag{1}$$

where s_i is the user's speed and $\overrightarrow{v_i}$ its average direction at t_i, the associated timestamp. An observation o_i is removed from the internal memory of the agent after a given time Δ_t. The agent also holds a list of contacts $C = (c_1, c_2, c_3, ..., c_p)$, representing other agents with whom communication is possible. The agent is

able to send its observations O to its contacts at any time as well as to receive its contacts' observations. With every contact c_k, the agent associates three values $conf_k$, $dist_k$ and $\overrightarrow{dir_k}$ where:

- $conf_k$ is the confidence with which the agent assumes c_k to be part of its neighborhood,
- $dist_k$ is the estimated distance between the agent and c_k,
- $\overrightarrow{dir_k}$ is the estimated direction in which c_k is situated relatively to the agent.

If $conf_k$ happens to become greater than an empirically determined threshold δ_{conf}, the agent considers c_k as a neighbor situated at a distance $dist_k$ in the direction $\overrightarrow{dir_k}$. The objective of the agent is to accurately estimate these three values for every contact in order to correctly reconstruct its neighborhood's topology. For the specific estimation of $dist_k$, the agent uses an additional *exploration* distance λ_k, as well as a memory of optimal estimated distances Λ_k.

3.2 Local Neighborhood Discrimination

In order to understand the topology of its neighborhood and discriminate neighbors from simple contacts, the agent compares its observations with its contacts' observations through communication. This comparison enables the agent to update $conf_k$, $dist_k$, λ_k and $\overrightarrow{dir_k}$ associated with the involved contact c_k.

***Ghost* Observations.** When an agent receives observations $O_k = (o_{k_1}, o_{k_2}, o_{k_3}, ..., o_{k_m})$ with $o_{k_i} = (s_{k_i}, \overrightarrow{v_{ki}}, t_{k_i})$ as defined in Eq. (1) from a contact c_k, four arrays G_1, G_2, G_3 and G_4 referred as *ghost* observations are computed as follows:

$$\begin{aligned} &G_j = (g_{j_1}, g_{j_2}, g_{j_3}, ..., g_{j_m}) \quad j \in [1,4] \\ &\text{with } g_{j_i} = (\tau_{j_i}, \overrightarrow{v_{k_i}}) \quad i \in [1,m] \quad \text{where } \tau_{j_i} = t_{k_i} - \frac{d_j}{s_{k_i}} \end{aligned} \tag{2}$$

$$\begin{aligned} \text{and } d_1 &= dist_k \\ d_2 &= \lambda_k - \Delta_\lambda \\ d_3 &= \lambda_k \\ d_4 &= \lambda_k + \Delta_\lambda \qquad \text{with } \Delta_\lambda \text{ a given small distance.} \end{aligned}$$

G_j therefore represents the observations the agent should have made if c_k is indeed a neighbor and if the distance d_j between them is accurate.

***Paired* Observations.** Each G_j is then compared to the actual agent's observations $O = (o_1, o_2, o_3, ..., o_n)$ as defined in Eq. (1). This comparison is performed to determine which distance d_j ensures the best correlation between the agent's observations O and its contact's observations O_k. The objective is therefore to converge with steps of size Δ_λ towards the real distance, if it only exists, after

several communication rounds. Four arrays P_1, P_2, P_3 and P_4 referred as *paired* observations are then computed from G_1, G_2, G_3, G_4 and O as follows:

$$P_j = (p_{j_1}, p_{j_2}, p_{j_3}, ..., p_{j_m}) \quad j \in [1,4] \quad \text{with } p_{j_i} = (g_{j_i}, \omega_{j_i}) \quad i \in [1,m] \tag{3}$$

where $\omega_{j_i} \in O = (o_1, o_2, o_3, ..., o_n)$ is the agent's observation for which $|\tau_{j_i} - t_l|_{l \in [1,n]}$ is minimum, τ_{j_i} being the timestamp of the current *ghost* observation g_{j_i} and t_l the timestamp of o_l for $l \in [1,n]$. This results in every *ghost* observation $g_{j_i} \in G_j$ being matched with the temporally closest agent's observation $\omega_{j_i} \in O$. Not every $o_i \in O$ is however necessarily matched with a *ghost* observation and several *ghost* observations can be matched with the same agent's observation.

Each array of *paired* observations P_j is then used to compute three scores referred as $match_j$, gap_j and $rate_j$.

Match. This score corresponds to the proportion of observations that have successfully been paired with a *ghost* observation out of all observations. We note $match_j = \frac{N_j}{n}$ with N_j the number of observations in O that have been paired with a *ghost* observation in G_j and n the overall number of observations in O. We consider $match_A$ better than $match_B$ if $match_A > match_B$.

Gap. This score corresponds to the average absolute time difference between a *ghost* observation's timestamp and its associated observation's timestamp.

$$gap_j = \frac{1}{m} \sum_{i=1}^{m} |\tau_{j_i} - t_{j_i}| \tag{4}$$

with τ_{j_i} the timestamp of g_{j_i} and t_{j_i} the timestamp of the associated ω_{j_i}, for $i \in [1,m]$ and m the number of *paired* observations in P_j. We consider gap_A better than gap_B if $gap_A < gap_B$.

Rate. This score corresponds to the proportion of *paired* observations for which the absolute time difference between the *ghost* observation's timestamp and its associated observation's timestamp is less than or equal to δ_t, a given tolerance threshold, out of all *paired* observations. We note $rate_j = \frac{M_j}{m}$ with M_j the number of *paired* observations in P_j for which $|\tau_{j_i} - t_{j_i}| < \delta_t$ and m the overall number of *paired* observations in P_j. We consider $rate_A$ better than $rate_B$ if $rate_A > rate_B$.

The *paired* observations P_{best} is chosen out of P_1, P_2, P_3 and P_4 resulting in the best *rate*, *match* and *gap*, by this order of priority. This represents the criticality of the agent whose goal is thus to improve it by modifying its position as described in the following.

Confidence Update. The confidence $conf_k$ with which the agent assumes c_k to be part of its neighborhood holds a gliding average of the S most recent *rates*

the agent evaluated. The current $rate_{best}$ resulting from P_{best} is therefore used to update $conf_k$ as follows:

$$conf_k \leftarrow conf_k - \frac{conf_k - rate_{best}}{S} \tag{5}$$

Moreover, if $conf_k$ becomes greater than the threshold δ_{conf} then c_k is now considered as a neighbor by the agent. The contrary leads c_k to be considered as a regular contact.

Direction Update. The estimated direction $\overrightarrow{dir_k}$ in which c_k is situated relatively to the agent holds a gliding average of the S most recent relevant directions the agent evaluated. Are considered relevant the users directions in the *paired* observations in P_{best} for which $|\tau_{best_i} - t_{best_i}| < \delta_t$ for $i \in [1, n]$. The average direction $\overrightarrow{\mu_{best}}$ resulting from these relevant users directions is therefore used to update $\overrightarrow{dir_k}$ as follows:

$$\overrightarrow{dir_k} \leftarrow \overrightarrow{dir_k} - \frac{\overrightarrow{dir_k} - \overrightarrow{\mu_{best}}}{S} \tag{6}$$

Distance Update. The *exploration* distance λ_k previously used to compute the *ghost* observations G_2, G_3 and G_4 is simply the most recent best evaluated distance d_{best} from which P_{best} is originated. This distance d_{best} is therefore used to update λ_k as follows: $\lambda_k \leftarrow d_{best}$.

Moreover, if $rate_{best} > 0$, d_{best} is saved in the optimal estimated distances memory Λ_k. The update of the estimated distance $dist_k$ between the agent and c_k then requires to sample a random value λ_{sp} from this optimal estimated distances memory Λ_k. This sampled distance λ_{sp} is then used to update $dist_k$ as follows: $dist_k \leftarrow dist_k + sgn(\lambda_{sp} - dist_k) \cdot \Delta_{dist}$ where sgn is the sign function and Δ_{dist} a given small distance.

Although the *exploration* distance λ_k is quite sensitive to dynamic noise and can quickly deviate from the real distance between the agent and c_k, $dist_k$ is at all time converging with steps of size Δ_{dist} towards one of the most optimal estimated distances stored in Λ_k. We therefore consider λ_k as a good tool to ensure exploration and $dist_k$ as the most relevant and robust estimated distance between the agent and c_k.

3.3 Evaluation Metrics

We use a number of metrics to evaluate our topology reconstruction performance. Three criteria are taken into account:

- the correctness of the neighbors discrimination,
- the accuracy of the estimated distances between the agent and its neighbors,
- the accuracy of the estimated directions in which the neighbors are situated relatively to the agent.

Neighbors Discrimination. For a given agent with N true neighbors and N_e estimated neighbors, we name P_T the number of true positives and P_F the number of false positives within the estimated neighbors. We use these basic quantities to build two standard metrics referred as R_{TP} (true positive rate) and $Precision$ [10] as follows: $R_{TP} = \frac{P_T}{N}$ and $Precision = \frac{P_T}{P_T+P_F} = \frac{P_T}{N_e}$

Distances Estimation Error. For a given agent with N true neighbors, the local distances estimation error is the average absolute difference between the estimated distances and the real ones, for all N neighbors.

Directions Estimation Error. For a given agent with N true neighbors, the local directions estimation error is the average absolute difference between the estimated angles and the real ones, for all N neighbors.

4 Results and Discussion

Confidence Threshold Determination. For the solution to function once deployed, there is no strict need for the streetlights to be able to discriminate neighbors from regular contacts in a binary way. Indeed, since each streetlight associates a neighbor confidence score with each one of its contacts, such a score can be used in a flexible manner to infer a variety of different lighting policies depending on the specific needs of the user. However, for the metrics to accurately evaluate the reconstruction performance of the solution, it is necessary to perform a binary discrimination between estimated neighbors and regular contacts. We thus provide the agents with a confidence threshold δ_{conf}, whose role is to ensure such a discrimination during the performance evaluation task only.

For a given network whose structure is to be reconstructed, Fig. 2 exhaustively shows all confidence scores within the agent population after a simulated period of 48 h (in the following, all durations are simulated). In this example, we see that true neighbors have been given significantly higher confidence scores by the system than regular contacts.

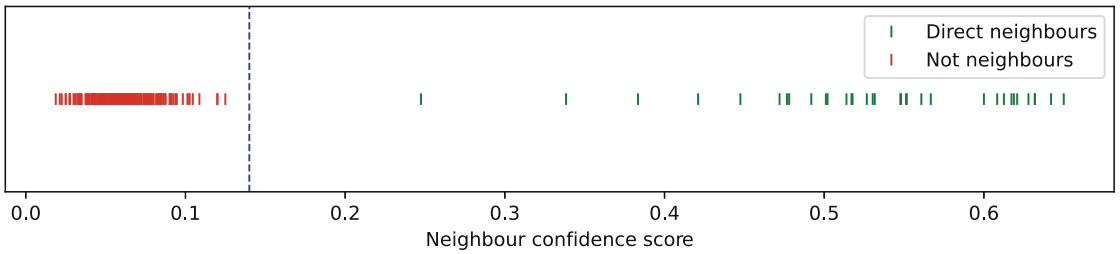

Fig. 2. Exhaustive confidence score distribution within the agent population after a simulated period of 48 h for a realistic topology. Green lines represent confidence scores associated with agents that have been validated as neighbors by humans (true neighbors) whereas red ones represent those that are not. Indirect neighbors (*i.e.* a direct neighbor is situated between them and the agent) are not displayed. The empirically chosen threshold δ_{conf} is represented in blue. High confidence is correlated with true neighborhood. (Color figure online)

Confidence score therefore appears to be a good tool to whether or not consider a contact as a neighbor. Empirically, we choose $\delta_{conf} = 0.14$ since this value appears to accurately discriminate neighbors from regular contacts.

In the scope of the neighbors discrimination evaluation task in this simulated environment, we can therefore consider any contact with a confidence score higher than 0.14 as an estimated neighbor, the contrary leading it to be considered as a regular contact.

Global Reconstruction Performance (Simulation). To validate the global reconstruction performance of our solution in a simulated environment, we run multiple simulations on the same fictional urban lighting topology, shown in Fig. 1. In Fig. 3, we demonstrate that with the chosen threshold $\delta_{conf} = 0.14$, both our neighbors discrimination metrics R_{TP} (true positive rate) and *Precision* quickly increase and exceed 0.9 after a simulated period of 2–3 hours. *Precision* tends to reach a rate of 1 after 2 h, which means that every estimated neighbor is indeed an actual neighbor. However, it takes about a day for the system to fill the gap between an R_{TP} of 0.95 and an R_{TP} of 1. This shows that even if 5% of the true neighbors seem harder to identify, the system eventually discriminates them correctly after some time. Moreover, we show that once the system converges, it is not impacted by dynamic noise and durably maintains both a *Precision* and an R_{TP} of 1. This points out the absence of false positives, as well as an exhaustive identification of all true neighbors.

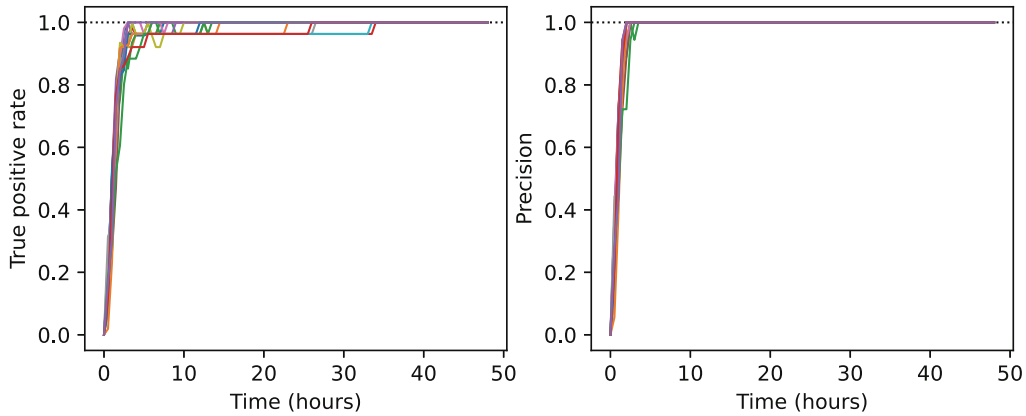

Fig. 3. Evolution through time of R_{TP} (true positive rate, on the left) and *Precision* (on the right) for 15 simulation runs on the same fictional urban lighting topology. An R_{TP} of 1 indicates that all true neighbors are identified by the system and a *Precision* rate of 1 indicates that all estimated neighbors are true neighbors (*i.e.* true positives)

In addition, Fig. 4 clearly shows the distances estimation convergence, with an average absolute error smaller than 1 m after 12 simulated hours. The system then reaches an average minimum distances estimation error of about 30 cm (streetlights are usually situated at a distance between 15 and 35 m from each others). With regard to the directions estimation error, it also significantly drops

in the first few hours, before reaching 1 degree of error in average after about 7 h. These low error values are maintained durably.

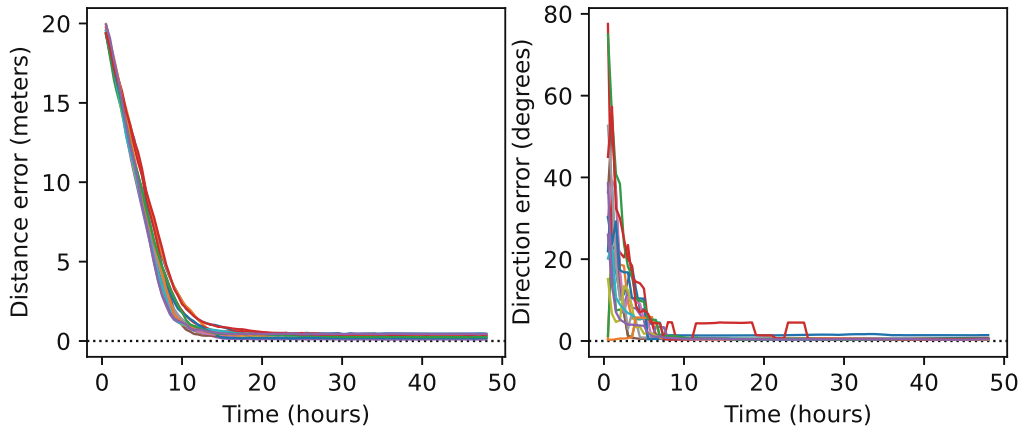

Fig. 4. Evolution through time of absolute distances estimation error (on the left) and absolute directions estimation error (on the right) for 15 simulation runs on the same fictional urban lighting topology

Global Reconstruction Performance (Real World). To test the global reconstruction performance of our solution in a real world situation, we ran experiments on real data collected by *Kara* sensors on two different occasions in the same street. The urban lighting topology in question only consists of a single street along which are located four streetlights, separated by a distance of about 28 m in average. Since the topology is so simple, all streetlights share the same neighborhood. Therefore, we do not study the neighbor discrimination performance of our solution but focus on the distances and directions estimation evaluation.

Figure 5 shows the distances estimation convergence on such real data, with an average absolute error dropping somewhere between 3 and 5 m after 40 operating hours. With respect to the directions estimation error, it significantly drops in the first few hours before reaching about 4 degrees of error in average after 14 operating hours. These values are maintained durably.

Discussion. We demonstrate that our solution performs particularly well in the case of a simulated urban lighting topology, with close to perfect neighbors discrimination evaluation rates and negligible absolute distances and directions estimation errors after a few simulated hours. Because the studied topology is sufficiently varied and realistic, these results suggest that the solution would perform well on a grand variety of urban lighting topologies. However, further simulations are needed to explore more specific topologies as well as scaling-up.

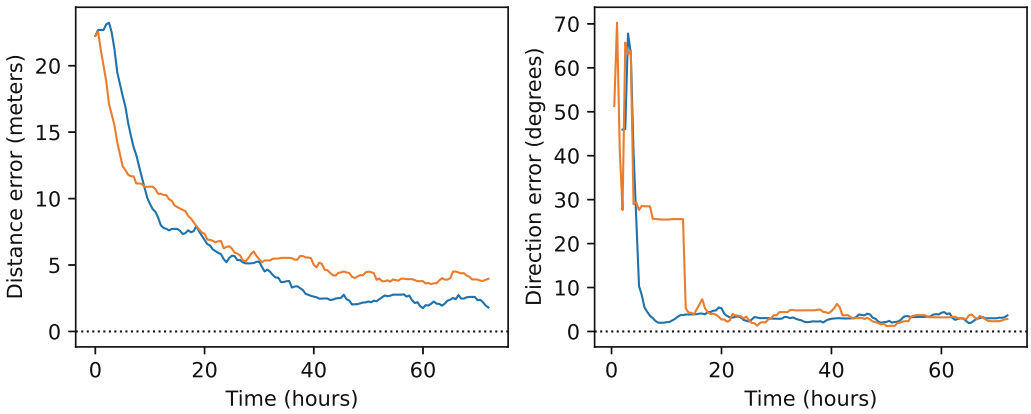

Fig. 5. Evolution through time of absolute distances estimation error (on the left) and absolute directions estimation error (on the right) on two real world data sets

It is noteworthy that the agents only communicate the events they perceive, and have no other information about their position nor do they sense the other agents, as is the case in multi-robot topology reconstruction research [14].

When we test our solution in a real world situation, although the system still converges, results are not as accurate as in a simulated environment, especially in terms of distances estimation. While simulated user traffic may not be as realistic as real user traffic, simulated *Kara* sensors certainly collect more accurate data about a simulated world than real *Kara* sensors do about the real one. In this respect, uncertainties about the users detection timestamps, their speed as well as their positional tracking in the real world are elements possibly explaining the difference between our system's performance in a simulated and a real environment. In addition, we did not test our system's neighbors discrimination performance in a real world situation as *Kara* sensors are still being installed on a larger scale.

5 Conclusion

In order to reduce computational costs as well as problem complexity, smart urban lighting solutions commonly aim to adopt distributed approaches, implementing local lighting policies within autonomous streetlights. In some cases, the possibility for streetlights to exchange information offers opportunities for more flexible and relevant lighting policies.

Motivated by the fact that the effectiveness of such local lighting policies highly depends on the quality of each streetlight's environmental representation, we proposed, implemented and evaluated a solution that enables each streetlight to continuously identify and position its nearest neighbors. We demonstrate the performance of our solution when embedded in simulated environments and obtain encouraging results when tested in a real world situation.

As *Kara* sensors are currently being deployed on the campus of our university, we will be able to upscale real-world testing in the following months.

Acknowledgements and Data. This work is part of the LightCampus project supported by the *Région Occitanie* (www.laregion.fr) through EU (ERDF-ESF) funds. We thank the *GIS neOCampus* for the experimental environment (www.neocampus.org). The GAMA models used for generating the data used in the simulation tests can be found here https://cloud.irit.fr/index.php/s/cTQG4K8bNsn8CIi.

References

1. Bendul, J.C., Blunck, H.: The design space of production planning and control for industry 4.0. Comput. Ind. **105**, 260–272 (2019)
2. Capera, D., Georgé, J.P., Gleizes, M.P., Glize, P.: The amas theory for complex problem solving based on self-organizing cooperative agents. In: WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. IEEE (2003)
3. Couellan, N., Jan, S., Jorquera, T., Georgé, J.P.: Self-adaptive support vector machine: a multi-agent optimization perspective. Expert Syst. Appl. **42**(9), 4284–4298 (2015)
4. Drogoul, A., et al.: GAMA: a spatially explicit, multi-level, agent-based modeling and simulation platform. In: Demazeau, Y., Ishida, T., Corchado, J.M., Bajo, J. (eds.) PAAMS 2013. LNCS (LNAI), vol. 7879, pp. 271–274. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38073-0_25
5. Ferber, J., Weiss, G.: Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence, vol. 1. Addison-Wesley Reading, Boston (1999)
6. Fortino, G., Russo, W., Savaglio, C., Shen, W., Zhou, M.: Agent-oriented cooperative smart objects: from IoT system design to implementation. IEEE Trans. Syst. Man Cybern. Syst. **48**(11), 1939–1956 (2017)
7. Georgé, J.P., Gleizes, M.P., Camps, V.: Cooperation. In: Serugendo, G.D.M., Gleizes, M.P., Karageorgos, A. (eds.) Self-organising Software. Natural Computing Series book series (NCS), Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-17348-6_9
8. González-Briones, A., De La Prieta, F., Mohamad, M.S., Omatu, S., Corchado, J.M.: Multi-agent systems applications in energy optimization problems: a state-of-the-art review. Energies **11**(8), 1928 (2018)
9. Juntunen, E., Sarjanoja, E.M., Eskeli, J., Pihlajaniemi, H., Österlund, T.: Smart and dynamic route lighting control based on movement tracking. Build. Environ. **142**, 472–483 (2018). https://doi.org/10.1016/j.buildenv.2018.06.048
10. Ma, C., Chen, H.S., Lai, Y.C., Zhang, H.F.: Statistical inference approach to structural reconstruction of complex networks from binary time series. Phys. Rev. E **97**(2) (2018). https://doi.org/10.1103/PhysRevE.97.022301
11. Paz, J.F.D., Bajo, J., Rodríguez, S., Villarrubia, G., Corchado, J.M.: Intelligent system for lighting control in smart cities. Inf. Sci. **372**, 241–255 (2016)
12. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. Knowl. Eng. Rev. **10**(2), 115–152 (1995)
13. Yan, Z., Jouandeau, N., Cherif, A.A.: A survey and analysis of multi-robot coordination. Int. J. Adv. Rob. Syst. **10**(12), 399 (2013)
14. Yi, S., Luo, W., Sycara, K.: Distributed topology correction for flexible connectivity maintenance in multi-robot systems. In: Proceedings of (ICRA) International Conference on Robotics and Automation (2021)
15. Zissis, G.: Energy consumption and environmental and economic impact of lighting: the current situation. In: Handbook of Advanced Lighting Technology, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-319-00295-8_40-1