A Dijkstra-Inspired Algorithm for Optimized Real-Time Tasking with Minimal Energy Consumption

Mohamed Lotfi [1,2,3], Abdelrahman Ashraf [4], Mustafa Zahran [5], Georges Samih [5], Mohammad Javadi [2], Gerardo J. Osório [6], João P. S. Catalão [1,2]

[1] Faculty of Engineering, University of Porto (FEUP), 4200-465 Porto, Portugal
[2] Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), 4200-465 Porto, Portugal
[3] Faculty of Energy Systems and Nuclear Science, Ontario Tech University, Oshawa, ON L1G 8C4, Canada
[4] Computer Science Department, German University in Cairo, New Cairo, Egypt
[5] Mechatronics Engineering Department, German University in Cairo, New Cairo, Egypt
[6] C-MAST, University of Beira Interior, 6200-358 Covilhã, Portugal

Emails: mohd.f.lotfi@gmail.com (ML), mubbyashraf@gmail.com (AA), mustafazahran1996@gmail.com (MZ), georgessami7@gmail.com (GS), msjavadi@gmail.com (MJ), gjosilva@gmail.com (GJO), catalao@fe.up.pt (JPSC)

***Abstract*— A highly versatile optimal task scheduling algorithm is proposed, inspired by Dijkstra's shortest path algorithm. The proposed algorithm is named "Dijkstra Optimal Tasking" (DOT) and is implemented in a generic manner allowing it to be applicable on a plethora of tasking problems In this study, the application of the proposed DOT algorithm is demonstrated for industrial setting in which a set of tasks must be performed by a mobile agent transiting between charging stations. The DOT algorithm is demonstrated by determining the optimal task schedule for the mobile agent which maximizes the speed of task achievement while minimizing the movement, and thereby energy consumption, cost. A discussion into the anticipated plethora of applications of this algorithm in different sectors is examined.**

Keywords*— *energy management, task optimization, Dijkstra, industrial informatics.

I. Introduction

A. Background and Motivation

In an exceedingly dynamic and digital world, the old saying of "time is money" presents itself in all modern problems. Energy systems witnessed momentous change during the past few decades [1] which in turn affected all sectors which are heavily energy-dependent, including industrial and transport sectors [2]. Despite the advancement of technologies leading to overall abundance or resources, increased socio-technical complexities associated with the availability of resources make the optimal management thereof of paramount importance [3].

In addition to the added complexity of the energy supply infrastructure, process automation levels are at an all-time high, making it necessary to develop and deploy new algorithms for optimized task management in order to guarantee cost-efficiency and reliability of these automated processes [4], [5].

B. Literature Review

This matter presents throughout a wide range of scientific research topics. A rundown of recent scientific studies is performed and subsequently presented to establish the state-of-the-art of scientific literature addressing optimal task scheduling and management in modern automated systems. As previously mentioned, two of the most affected sectors are the industrial and transport sectors [2], with maximizing cost-efficiency already of pivotal importance for the two.

For the transport sector, there has been a lot of recent focus on developing optimal management algorithms for consumer-owned electric vehicle (EV) fleets [6], with an emphasis on cost-optimal energy management in the presence of hybrid technologies [7] and considering smart homes [8] and other modern solutions. for optimal utilization of distributed energy resources (DERs) [9]. This is especially important with dynamic electricity pricing schemes adopted through demand response (DR) implementation [1], [2]. Recent research on this matter was not only confined to consumer-owned EVs, with a lot of research also investigating smart public transport systems with increased proliferation of electric buses (EBs) and smart charging infrastructures [10]. The priority is ensuring cost-efficiency of the systems [11] through optimal scheduling [12].

In industrial applications, optimal management of time and resources is of even more critical due to the profit-centered of industry, throughout the wide-ranging spectrum of industrial specializations. In the context of smart factories, multi-agents systems are proposed as a model for coordination between autonomous systems working on performing preset tasks in factories with high levels of automation and smart communication infrastructure [13]. Adoption of intelligent algorithms for optimal task scheduling, in industry has been shown to result in significant cost savings, whether performed by automated mobile agents or human labor. Such saving are crucial for industries and economic growth [14].

Algorithms for optimal task-handling are being investigated for a wide array of applications, ranging from coordinating autonomous self-driving EVs [15], to cooperative robotics [16], industrial site inspection [17], and the management of modern warehouses [18]. As such, the development of these algorithms for optimal cost-efficiency of task handling, regardless of the type of task, becomes imperative for modern industries. With increased intertwining of modern systems and cross-industry designs it becomes even more important to design algorithms for generic systems [19], which are not application-specific, and could be employed regardless of the target sector, be it smart factories, modern warehouses, EV fleet management, etc.

M. Lotfi acknowledges the support of the MIT Portugal Program (in Sustainable Energy Systems) by Portuguese funds through FCT, under grant PD/BD/142810/2018. G.J. Osório acknowledges the support by UIDB/00151/2020 research unit (C-MAST) funded by FCT. M.S. Javadi and J.P.S. Catalão acknowledge the support by FEDER funds through COMPETE 2020 and by Portuguese funds through FCT, under POCI-01-0145-FEDER-029803 (02/SAICT/2017).

978-1-7281-7455-6/20/$31.00 ©2020 IEEE

C. Novel Contributions

In this study, we propose, implement, and validate a novel algorithm for optimal task management. The proposed algorithm is generic in nature, meaning it can be adapted to different problems in which a limited number of mobile agents are required to perform a set number of tasks, with the minimal energy consumption (and thereby cost). The contributions of this work can thus be summarized in the following list:

- Design an algorithm of optimal task management by a limited number of mobile agents.
- A generic tasking problem is modeled using graph theory to guarantee applicability to a wide range of modern problems (particularly in the industrial and transportation sectors).
- Ensure the computational efficiency of the algorithm to enable application in real-time.
- Demonstrate the proposed algorithm considering a generic case study in an industrial setting.

D. Paper Organization

This manuscript is organized as follows: In Section I, the motivation behind this work, literature review, and novel contributions were presented. In Section II, the conceptual model is introduced, describing how a generic tasking problem was computationally modeled using a graph theory approach. In Section III, the proposed algorithm is detailed and the implementation is presented. In Section IV, a case study based on an industrial application is presented, demonstrating the validity of the proposed algorithm and performing a sensitivity analysis of different parameters. Finally, in Section V, the conclusions of this work are listed and prospects for work following up on this study are discussed.

II. Conceptual Model

A. Tasking Problem with Mobile Agents

In Fig. 1, a generic tasking problem is illustrated. The main elements of a tasking problem in this study can be defined and listed as follows:

Map: The map is a confined space where all the tasks and mobile agents are located. All events and scheduling is performed within this local environment.

Mobile Agent: is an agent which can move around the map and perform tasks. The agent is electric, meaning it consumes electric energy on a local battery as a cost of movement. This agent could be autonomous, or human-operated (e.g. Segway, electric pallet jack, golf cart, etc.)

Charging Station: is where the mobile agent can go to recharge its batteries. Commonly found on the edges of most maps, they can be located anywhere across the traversable map.

Tasks: must be reached by a mobile agent in order to be accomplished. This is generic: I.e., in inspection problems the task is merely for the agent to be there every period of time. In handling problems the agent must stay until the task is complete.

Obstacles: are untraversable parts of the map. The mobile agent must plan a path around them to reach tasks or charging stations.

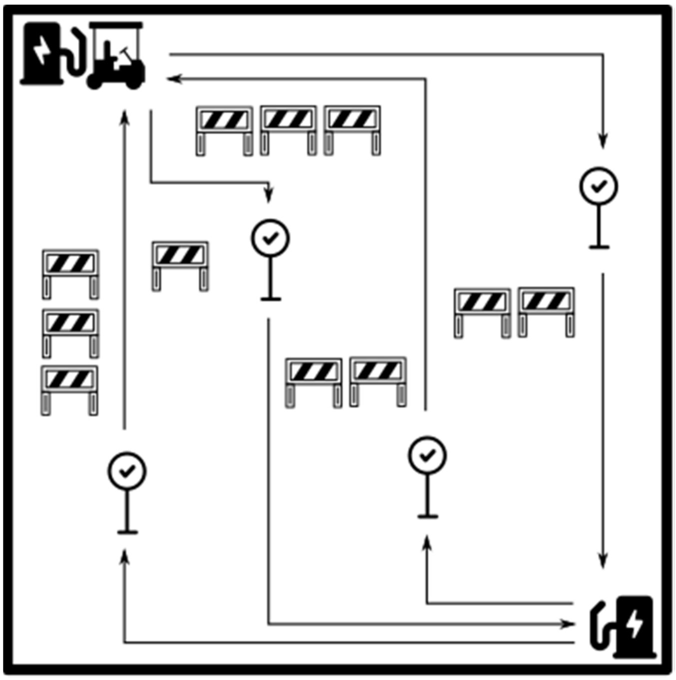


Fig. 1. An illustration of a generic tasking problem: A movile agent needs to perform specified tasks located at different locations in a confined map, in the presence of intraversible obstacles. The movement is associated with energy consumption, and recharging is performed at set locations.

B. Modeling a Generic Environment with Graph Theory

The first step is to mathematically model the generic environment for a tasking problem before proceeding with computational implementation to optimize task handling. The maps shown in Fig. 1 can be modeled by graph theory using the following steps:

Step 1: Divide the map into a "mesh" of equidistant and isomorphic cells. The size of each cell should be based on the smallest element in the map.

Step 2: Construct a graph in which each cell becomes a node V_i in a set of nodes $\mathbf{V}$.

Step 3: Each node is connected to adjacent nodes through an edge E. In this sense:

$$E \subseteq \left\{ \{V_i, V_j\} \mid (V_i, V_j) \in \mathbf{V}^2 \land i \neq j \right\} \tag{1}$$

In other words, an edge $\{V_i,V_j\}$ connects nodes V_i and V_j given that V_i and V_j are adjacent in the graph and they are not the same node.

An illustrative example for a five-by-five map is shown in Fig. 2, in which each cell is assigned as a node, and connected to adjacent nodes through the edges of the graph.

At this stage the graph model of the map is complete, and properties of nodes and edges can be defined to reflect different map elements using object-oriented programming (OOP). For instance, the movement cost through a certain edge for the mobile agent can be represented as a function of a property value of the two connected nodes, and the values can be updated to represent a dynamic environment with the movement of the agent or other events.

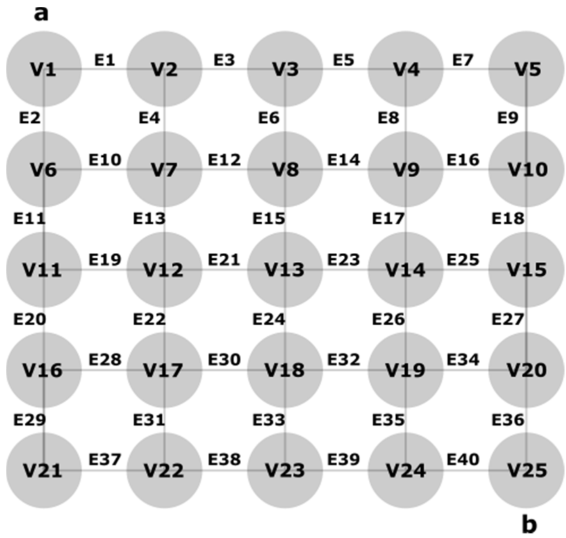


Fig. 2. An illustration of five-by-five map modeled using graph theory. Cells are assigned as nodes in the graph, and connections between adjacent cells are modeled as .edges of the graph.

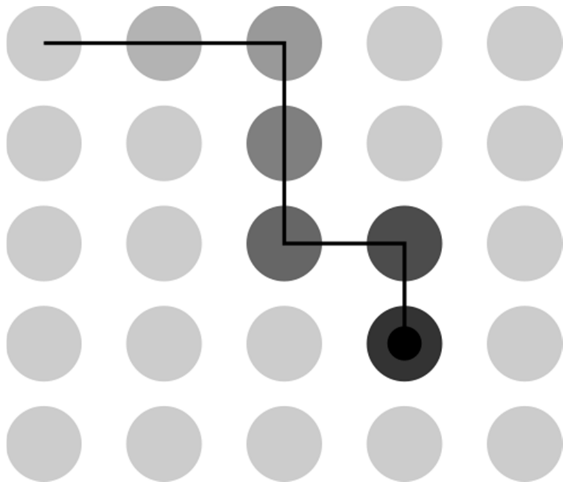


Fig. 3. . Node properties being updated as the mobile agent moves and follows a path through the map.

III. Proposed Algorithm

A. Dijkstra-Inspired Algorithm

The proposed algorithm was inspired by Dijkstra's shortest path first (SPF) algorithm [20], a well-known fundamental algorithm in graph theory. The algorithm is a highly computationally efficient algorithm finding the shortest path between two nodes in a graph. By modelling the generic tasking problem using graph theory, Dijkstra's algorithm can be used as the main pillar of an algorithm for optimal task scheduling. This novel algorithm is the main contribution of this study is explained subsequently, by elaborating how each element of the problem can be modeled in the graph representation using OOP.

- Mobile agents are created as an object which has property variables corresponding to location (represented by node number in the graph), battery charge level, and any other features depending on the type of the problem.

- The charging stations are modeled are the start and end path of the mobile agent, since they must eventually return to recharge. The Dijkstra SPF is then employed to calculate the optimal path through the map, provided the subsequent modeling aspects.

- Each node has a specific "heat" property, which is the main premise of the proposed algorithm. The value of this "heat" property is used to establish all other relations. The movement cost through an edge is defined as the difference between the heat values of the two connected nodes. The initial heat value of all nodes is zero.

- Obstacles can therefore be easily modeled by setting a very large number as the heat value. In this manner, obstacles can be modeled in a very computationally efficient way (as opposed to the use of exceptions or conditional statements), since the movement cost to/from that node would never be chosen over the alternative.

- On the contrary, tasks can be modeled by setting a very low value, even a negative one, depending on the type, and/or urgency of each task. This acts as an "attractor" for the mobile agent, since the Dijkstra algorithm will be attracted to pass through that node when constructing the path.

- Once a robot has reached a task, the heat value of this node is incremented. In this way, the movement cost to/from this location is increased, so it is not selected again as the task is accomplished.

- Whenever a new task arises, its location on the map can be assigned by setting the heat value of the corresponding node, based on its urgency. For recurring tasks, this value can be updated based on the frequency

- A timer is used to simulate the flow of time. With equidistant and isomorphic cells being used to model the map, the movement time from one node to the other can be used as the unit of time. As such, a "cool-down" effect can be made by decrementing the global values in the map every iteration, to keep the continuous running of the algorithm.

All implementation was performed using Python 3.6.7 on a standard laptop computer with an Intel Core i7-8550U CPU @ 1.80 GHz, 16.0 GB RAM, Windows 10 64-bit operating system.

Algorithm I. The proposed algorithm for optimal tasking.

```
Input Map, TaskList, MobileAgents
while isRunning do
  t += 1
    for each A in MobileAgents
      if A.atStation then
        A.path ←
DijkstraSPF(A.orSta,A.deSta)
        A.atStation := false
      else
        A.loc[t] ← A.move(path,loc[t-1])
        if map[A.loc[t]].isStation then
          A.atStation := true
        else
          Map.nodeHeat[loc[t]] += inc
        end if
        for each N in Map.nodeHeats
          N-=Map.cooldownRate
        end for
      end if
    end for
end while
```

IV. VALIDATION

A. Generic Case Study: Industrial Site Inspection Problem

To demonstrate the .proposed algorithm a generic case study is used based on the facility inspection problem [17]. In this problem, a mobile agent such as illustrated in Fig.1 is stationed at either of the charging stations on the edges of the map. The mobile agent must inspect the site, making sure all areas are frequently inspected and no areas are ignored. The agent must head after each "patrol" to the charging station at the other side of the map in order to report and to recharge the vehicle used or switch to another.

In this case, all nodes in the map are "tasks", since the objective is to patrol the full map continuously. The constraint in this case is to avoid any areas of the map being overlooked on the long term (i.e., avoid some areas being visiting more than other as much as possible).

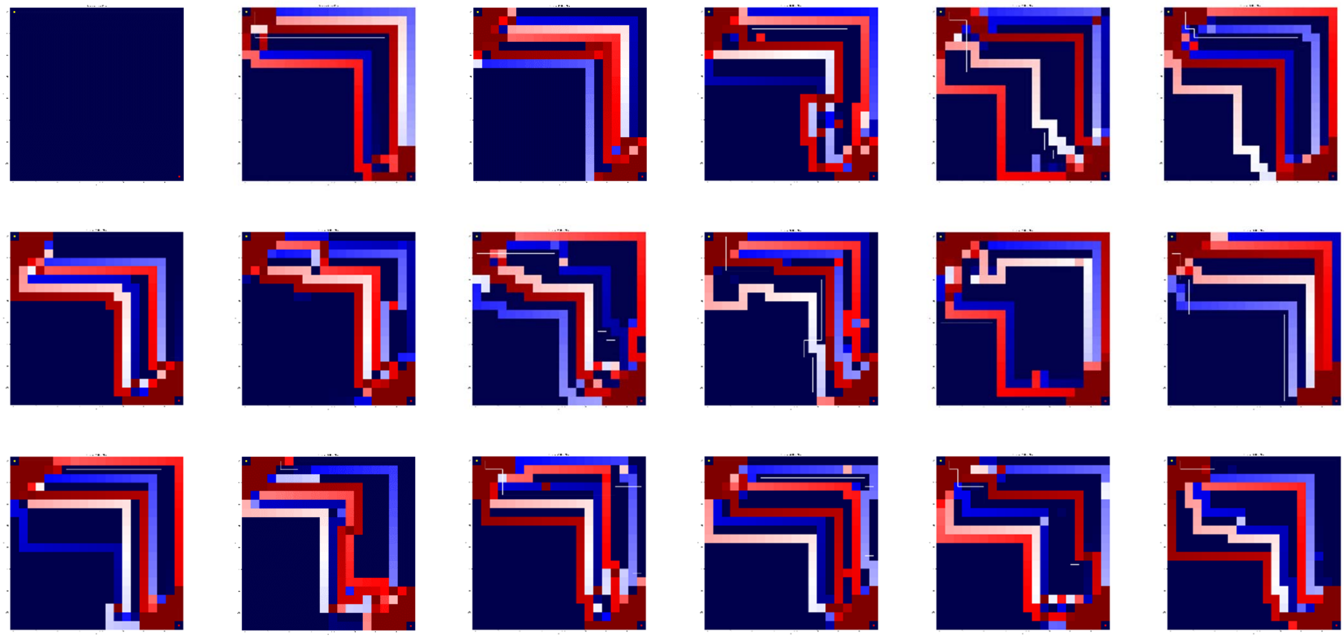


Fig. 4. Results for a 20x20 map for N={0;5;10;15;20;25;30;35;40;45;50;55;60;65;70;75;80;85}, with INC=200.

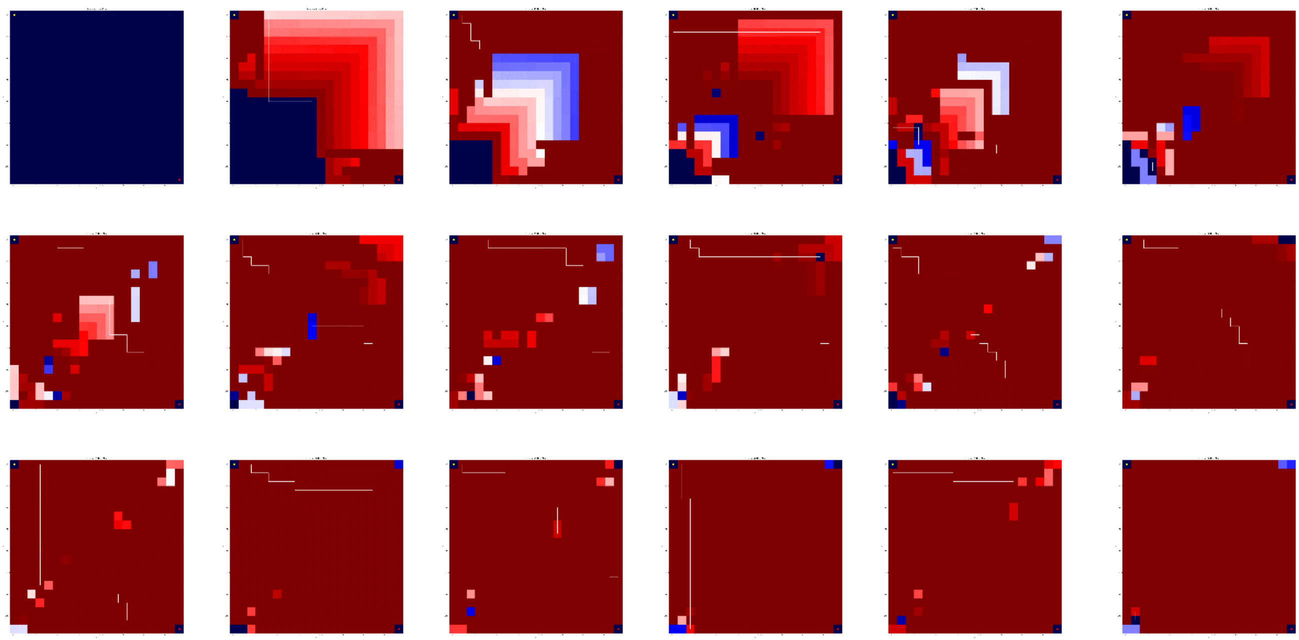
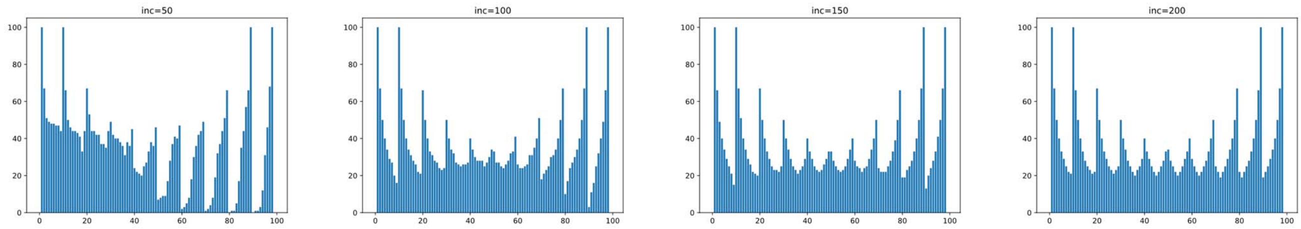
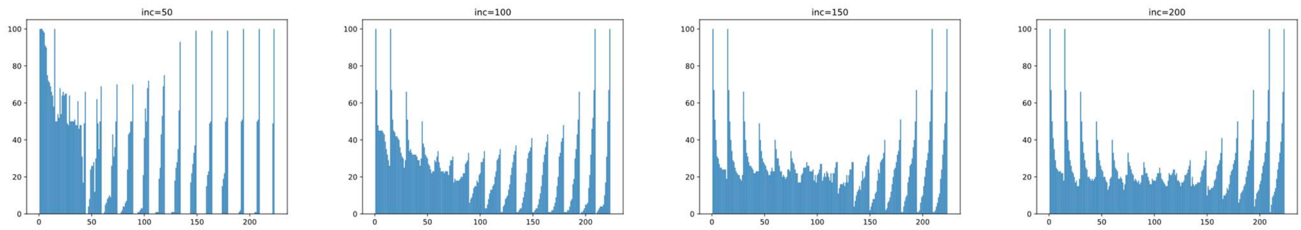


Fig. 5. Results for a 20x20 map for N={0;5;10;15;20;25;30;35;40;45;50;55;60;65;70;75;80;85}, with INC=800.



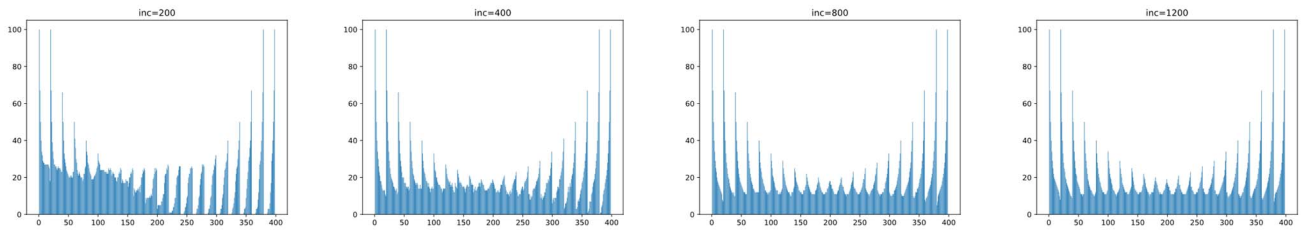
(a) **(b)** **(c)** **(d)**

Fig. 6. Histogram showing the number of times each node was inspected for a 10x10 map with an INC value of 50 (a), 100 (b), 150 (c), and 200 (d).



(a) **(b)** **(c)** **(d)**

Fig. 7. Histogram showing the number of times each node was inspected for a 15x15 map with an INC value of 50 (a), 100 (b), 150 (c), and 200 (d).



(a) **(b)** **(c)** **(d)**

Fig. 8. Histogram showing the number of times each node was inspected for a 20x20 map with an INC value of 200 (a), 400 (b), 800 (c), and 1200 (d).

As such, to model this problem using the proposed algorithm every time the agent passes through a location the heat value will be incremented by a certain value. Each time step, the entire map is cooled down by a value of 1.

The INC value has a significant effect on the results. This is shown in Fig 4 and Fig 5, where the results for the a 20x20 map, showing 85 map traversals (in steps of 5), with an INC value of 200 and 800, respectively. It can be seen that if the value of INC is too low (in the case of 200), the cool-down of the map is not sufficiently fast enough to for the agent to cover the bottom-left sector of the map before the upper-right one "cools down" (i.e, urgency of the inspection task there is reset again too fast before the agent has the chance to cover the full map.

This does not seem to be the case when INC is set to 800, as the agent manages to fully traverse the map, achieving the two main objectives: 1) go through the shortest path to minimize the energy consumption, and 2) achieve the mission of inspecting the full site with all areas being frequently visited. In order to investigate how the INC parameter can be selected, a sensitivity analysis is made with different map sizes for the same problem.

B. Sensitivity Analysis: Map Size Effect on INC Parameter

In Fig. 6, Fig. 7, and Fig. 8, results for a map of size 10x10, 15x15, and 20x20 are shown, respectively. In each case, a histogram is shown, representing the number of times each node in the map was visited for inspection by the mobile agent after 100 traversals of the map.

In the case of a 10x10 map, when the value of INC was 50, it is clear that some nodes were very frequently inspected by the agent, while some nodes were in fact not visited at all. This inequality is shown to decrease as the value of INC is increased, and uninspected nodes are completely eliminated.

This same pattern can be seen in the case of the slightly larger 15x15 map. However, an important thing to note is with an INC value of 50 in this case, there is a significantly larger number of uninspected nodes compared to the case of the 10x10 map. Although the same pattern is noticed such that the number of uninspected nodes decreases with larger INC values, for this map a larger INC value is needed compared to the 10x10 case to reach the same inspection uniformity.

This can also be seen in the results of the 20x20 map shown in Fig. 8. Here, even an INC value of 200 is not sufficient to avoid uninspected nodes, which only start to disappear after INC is increased to 400. By conducting this sensitivity analysis we observed that the value of INC can be heuristically determined as twice the size of the map (i.e., 200 for a 100-node map, 600 for a 300-node map, and 800 for a 400-node map).

V. Conclusions and Future Work

In this study, we propose, implement, and validate a novel algorithm for optimal task management by a limited number of mobile agents. The proposed algorithm is generic in nature, meaning it can be adapted to different problems in which a limited number of mobile agents are required to perform a set number of tasks, with the minimal movement cost (and thereby energy consumption). A generic tasking problem with mobile agents is modeled using graph theory to guarantee applicability to a wide range of modern problems (particularly in the industrial and transportation sectors). By using an extended Dijkstra SPF algorithm, the shortest path for mobile agents to conduct tasks spread across a generic site is determined for traversal of the map to/from charging locations. A generic case study based on an industrial site inspection problem was used to validate and test the proposed algorithm, and a sensitivity analysis was performed to find optimal tuning of the parameters. Given the generic nature of the algorithm, it is versatile and can be employed to a wide range of task optimization problems, specifically in the industrial and transport sectors. For future work, different case studies can be tested, such as: modern warehouse management, EV fleet routing, and optimal charging infrastructure placement.

References

[1] M. Lotfi, C. Monteiro, M. Shafie-Khah, and J. P. S. Catalao, “Evolution of Demand Response: A Historical Analysis of Legislation and Research Trends,” in 2018 20th International Middle East Power Systems Conference, MEPCON 2018 - Proceedings, 2018, pp. 968–973.

[2] J. A. P. Lopes et al., “The future of power systems: Challenges, trends, and upcoming paradigms,” WIREs Energy Environ., Dec. 2019.

[3] R. K. Jain, J. Qin, and R. Rajagopal, “Data-driven planning of distributed energy resources amidst socio-technical complexities,” Nat. Energy, vol. 2, no. 8, pp. 1–11, Aug. 2017.

[4] M. Chowdhury and M. Maier, “Local and nonlocal human-to-robot task allocation in fiber-wireless multi-robot networks,” IEEE Syst. J., vol. 12, no. 3, pp. 2250–2260, Sep. 2018.

[5] V. Ortenzi et al., “Robotic manipulation and the role of the task in the metric of success,” Nat. Mach. Intell., vol. 1, no. 8, pp. 340–346, Aug. 2019.

[6] A. Ovalle, A. Hably, S. Bacha, G. Ramos, and J. M. Hossain, “Escort Evolutionary Game Dynamics Approach for Integral Load Management of Electric Vehicle Fleets,” IEEE Trans. Ind. Electron., vol. 64, no. 2, pp. 1358–1369, Feb. 2017.

[7] X. Hu, C. Zou, X. Tang, T. Liu, and L. Hu, “Cost-optimal energy management of hybrid electric vehicles using fuel cell/battery health-aware predictive control,” IEEE Trans. Power Electron., vol. 35, no. 1, pp. 1–1, 2019.

[8] X. Hou, J. Wang, T. Huang, T. Wang, and P. Wang, “Smart Home Energy Management Optimization Method Considering Energy Storage and Electric Vehicle,” IEEE Access, vol. 7, pp. 144010–144020, 2019.

[9] H. M. D. Espassandim, M. Lotfi, G. J. Osorio, M. Shafie-Khah, O. M. Shehata, and J. P. S. Catalao, “Optimal operation of electric vehicle parking lots with rooftop photovoltaics,” in 2019 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2019, 2019.

[10] M. Bhaskar Naik, P. Kumar, and S. Majhi, “Smart public transportation network expansion and its interaction with the grid,” Int. J. Electr. Power Energy Syst., vol. 105, no. December 2017, pp. 365–380, 2019.

[11] M. Rogge, E. Van Der Hurk, A. Larsen, and D. U. Sauer, “Electric bus fleet size and mix problem with optimization of charging infrastructure,” Appl. Energy, vol. 211, pp. 282–295, 2018.

[12] E. Yao, T. Liu, T. Lu, and Y. Yang, “Optimization of electric vehicle scheduling with multiple vehicle types in public transport,” Sustain. Cities Soc., vol. 52, no. August 2019, p. 101862, 2020.

[13] S. Wang, J. Wan, D. Zhang, D. Li, and C. Zhang, “Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination,” Comput. Networks, vol. 101, pp. 158–168, Jun. 2016.

[14] X. Gong, Y. Liu, N. Lohse, T. De Pessemier, L. Martens, and W. Joseph, “Energy- and Labor-Aware Production Scheduling for Industrial Demand Response Using Adaptive Multiobjective Memetic Algorithm,” IEEE Trans. Ind. Informatics, vol. 15, no. 2, pp. 942–953, Feb. 2019.

[15] B. Beirigo, F. Schulte, and R. Negenborn, “Dual-Mode Vehicle Routing in Mixed Autonomous and Non-Autonomous Zone Networks,” in IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 2018, vol. 2018-November, pp. 1325–1330.

[16] A. Khamis, A. Hussein, and A. Elmogy, “Multi-robot task allocation: A review of the state-of-the-art,” Stud. Comput. Intell., vol. 604, pp. 31–51, 2015.

[17] C. Liu, Multi-Robot Task Allocation for Inspection Problems with Cooperative Tasks Using Hybrid Genetic Algorithms. .

[18] C. Sarkar, H. S. Paul, and A. Pal, “A Scalable Multi-Robot Task Allocation Algorithm,” in Proceedings - IEEE International Conference on Robotics and Automation, 2018, pp. 5022–5027.

[19] M. Hermann, T. Pentek, and B. Otto, “Design principles for industrie 4.0 scenarios,” Proc. Annu. Hawaii Int. Conf. Syst. Sci., vol. 2016-March, pp. 3928–3937, 2016.

[20] E. W. Dijkstra, “A note on two problems in connexion with graphs,” Numer. Math., vol. 1, no. 1, pp. 269–271, Dec. 1959.