Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-10 (2022) 538–543

Sim-optimization hybrid approach for scheduling randomly deteriorating treatment tasks in horticulture

Merouane Mazar [*,***,****] **Belgacem Bettayeb** [**,†]
Nathalie Klement [***] **Mhammed Sahnoun** [*] **Anne Louis** [*]

[*] *LINEACT CESI, F-76800 Saint-Étienne-du-Rouvray, France*
[**] *LINEACT CESI, F-59000 Lille, France*
[***] *Arts et Métiers Institute of Technology, LISPEN, HESAM Université, F-59000 Lille, France*
[****] *Datalab MODIS, F-92400 Courbevoie, France*
[†] *Corresponding author. Email: bbettayeb@cesi.fr*

Abstract:
In this paper, we study the problem of scheduling robotized tasks in the context of Agriculture 4.0. The objective is to optimize the treatment tasks of plants against an evolving disease (mildew) within a greenhouse. The treatment is performed using a type-C ultraviolet radiation (UV-C) by an UV-Robot. We propose a semi-dynamic simulation-optimization approach based on a Markovian model of the disease behavior in the greenhouse. Two variants of simulation-optimization hybridization are tested and analyzed.

Keywords: Agriculture 4.0, simulation and optimization, Semi-dynamic scheduling.

1. INTRODUCTION

Long before, and then in parallel with the successive revolutions of the industrial sector, agriculture has evolved in a gradual and long-term process, often summarized by 4 eras, named “Agriculture 1.0” to “Agriculture 4.0”, where each is characterized by a significant technological advance (Zhai et al., 2020). Agriculture 1.0 corresponds to the era of traditional agriculture, which relies mainly on human labor and animal power.

During the 19-th century, steam engines were invented, improved and became widely used to provide new resources in all areas of life and industry, including agriculture. In this era of Agriculture 2.0, various agricultural machines were manually operated by farmers and many chemicals were used. Obviously, Agriculture 2.0 has greatly increased the efficiency and productivity of agricultural work.

In the 20-th century, Agriculture 3.0 was born with the rapid development of computers and electronics. Some also call it “biotechnology revolution”, and some others “green revolution”. In particular, a reasonable distribution of labor between agricultural machines has reduced the use of chemicals, improved the precision of irrigation, etc. Today, agriculture is experiencing its fourth revolution, thanks to the use of current new technologies, such as the Internet of Things, Big Data, artificial intelligence, cloud computing, remote sensing, etc. Applications of these technologies can significantly improve the efficiency of agricultural activities in terms of production, yield, food quality, and environmental and social impacts. In parallel to what is happening in industry, Agriculture 4.0 is characterized by the “massive” use of robotics.

Several types of robotic systems have been developed, both in the research field and on the market, such as harvesting robots, processing robots, monitoring robots and spraying robots. However, these systems need to be optimized, not only during their design phase, but also during their operation and maintenance phase to reach the expected objectives in terms of performance and return on investment. In particular, the complexity of robotic systems has emerged several research problems related to their supervision and control and the integrated optimization of their activities as an element of a more global productive system.

In this paper, we study the problem of scheduling robotized tasks in the context of Agriculture 4.0. The objective is to optimize the treatment tasks of plants against an evolving disease (mildew) within a greenhouse. The treatment is performed using a type-C ultraviolet radiation (UV-C) by a UV-Robot. We propose a semi-dynamic simulation-optimization approach based on a Markovian model of the disease behavior in the greenhouse.

The rest of the article is composed as follows. The following section (2) describes the system, states the studied problem, and gives a brief overview of the most relevant literature related to this work. Section 3 details the proposed approach and the models implemented to solve the problem of dynamically scheduling treatment tasks that are evolving stochastically over time. Some experiments are presented and discussed in section 4 and a summary of this work and its perspectives are given in section 5.

10.1016/j.ifacol.2022.09.449

2. PROBLEM STATEMENT AND RELATED WORKS

We consider a system composed of a greenhouse with n rows of I plants each. The plants are subject to a disease that appears and evolves stochastically following a predefined Markov process characterized by a transition matrix P_i proper to each plant $i \in \{1, .., n \times I\}$ in the greenhouse. The treatment is performed using a type-C ultraviolet radiation (UV-C) by a robot carrying type-C UV lamps to expose plants to a certain dose of UV-C. To ensure better operation of the robot, an optimized scheduling is needed to manage dynamically the treatment tasks of the greenhouse with the robot. The difficulty of this system lies in the behavior of the disease that the robot must treat. A solution of task scheduling in a static case, where the disease does not evolve in the greenhouse during the treatment, has already been studied by Mazar et al. (2020).

Thus, to optimize the scheduling of treatment tasks, it is necessary to simulate the dynamic behavior of the disease in the greenhouse environment (Mazar et al., 2018). The use of simulation allows to exploit the space of time and to follow the behavior of the systems on large horizons, but simulation has a myopia in the space of state. Optimization allows to exploit the state space and helps to find the best solution for various problems, but optimization has a myopia in time. The proposed approach studied in this article is to merge simulation and optimization. This coupling allows to exploit the time space and the state space.

The simulation allows to study the real environment of the system, especially if the behavior of its entities are well integrated in the simulator. The evolution of the disease behavior in the greenhouse is a stochastic process, this behavior makes the solution of the treatment problem difficult. The disease behavior is modeled with Markov chains that allow to find the transition probabilities between the disease states on each plant (Mazar et al., 2021). The objective of the optimization is to make the scheduling of the best missions of treatment of plants with the UV robot.

2.1 Coupling simulation and optimization

A simulator is a tool that allows to reproduce the behavior and interactions of the components of real systems in order to see their evolution over time. When using a simulator integrating a certain number of decisions during simulation, these decisions are based on one or several parameters of the system. The simulation-optimization allows to optimize these parameters to improve the functioning of the simulated system. This method is used by several researchers to represent, analyze and improve a complex system: it is a system where it is impossible to define exactly its next state (because of the randomness, the number of agents, the internal dynamics, ...), based on different techniques. Generally, a meta-heuristic adapted to the system is chosen, such as Genetic Algorithm, Ant Colony or Simulated Annealing.

The approach of coupling simulation and optimization appeared for the first time in the 90s' (Carson and Maria, 1997). It takes several forms, which will be detailed later. In Farzanegan and Vahidipour (2009), the authors studied the integration of the genetic algorithm with a pre-existing grinding circuit simulator, called ball milling circuits simulator (BMCS), in the MATLAB environment.

We will study the importance of the decision time in a specific state for optimization algorithms. We distinguish three types of coupling between simulation and optimization. In the first type of coupling, the simulator is considered as the evaluation function for the optimization algorithm. The optimization allows to modify the input parameters of the simulator until the parameters of the optimized solution are obtained (Sahnoun et al., 2016). In the second type of coupling, at each decision time during the simulation, the optimization improves the behavior of a state of the system (Fu, 2002). In the third type of coupling, the optimization makes a decision at time T by considering the probable impact at time T+1 (Campi and Garatti, 2018). Sim-optimization allows to estimate the next state according to the current decision based on the history of the system (Powell, 2005), (Powell, 2008) and (Wu et al., 2003).

In this paper there is a study on two optimization algorithms that are integrated in the simulator, the first one is the GA (Genetic Algorithm), and the second one is a HA (Heuristic Algorithm) based on the glutton. GA makes a decision at the beginning of processing for all missions, while HA makes a decision for each mission.

2.2 Dynamic Scheduling

In our work, it will be a matter of determining the tasks of the robot, including which rows should be visited to treat the disease. Scheduling is the process of finding the best combination to assign tasks to resources, resources that can be robots. Generally, the objective is to optimize the execution time of all tasks. The use of robots, which are autonomous entities, with automatic supervision systems, requires to be adopted scheduling models and resolving methods. The level of the disease is constantly changing, which makes it difficult to change the processing plan between the time of decision-making and its implementation. It is therefore important to study dynamic scheduling.

In the literature, there are two different approaches used to solve dynamic scheduling problems: a planning-based dynamic approach (used in manufacturing flow lines), and a best-effort dynamic approach (Suresh and Chaudhuri, 1993), (Ouelhadj and Petrovic, 2009) and (Barenji et al., 2017). The authors made their study on the dynamic behavior related to the customers' demands. They defined the negative points of the manufacturing machine operating system as: static scheduling (Huff et al., 2021), lack of machine autonomy (Abosuliman and Almagrabi, 2021), lack of real-time scheduling to ensure the flexibility of the system in the face of dynamic customer demands (Karimi et al., 2021), and the scheduling system that does not react to internal system disturbances (Shi et al., 2021). In (Ren et al., 2021), the authors proposed a proactive-reactive scheduling methodology that adapts to dynamic changes in work environments in the case of joint scheduling of machine tools and transportation resources. A mixed integer programming model taking into account production efficiency and transportation constraints is proposed along with a particle swarm optimization algorithm to respond to dynamic events and generate the rescheduling plan. Tubilla and Gershwin (2021) studied production scheduling in a failure-prone multi-product machine with setup times. The objective is to minimize the average inventory and backlog costs.

3. PROPOSED APPROACH AND MODELS

3.1 Simulation

In order to understand the behavior of the system studying and to evaluate various UV-C treatment strategies, we chose a simulation approach based on multiagent systems (MAS).

An MAS is a system composed of a set of agents (a process, a robot, a human being, a plant, etc.), active in a certain environment and interacting according to certain rules. An agent is an entity characterized by the fact that it is, at least partially, autonomous, which excludes a centralized control of the global system.

The field of MAS is currently a research area that is attracting a lot of interest. This field was born at the end of the 70's and beginning of the 80's, from the idea of distributing knowledge and control in Artificial Intelligence systems. This idea emerged on the one hand from the need to face the increasing complexity of systems and was favored on the other hand by the emergence of models and parallel machines, making possible the operational implementation of distribution (Hassas, 2003).

The interest of the MAS in this work is in the framework of the simulation which makes it possible to easily represent the behavior of the populations such as it is the case of the plants of a same greenhouse. They also allow to separate the entities which intervene in the system and to give the level of intelligence and autonomy necessary to each agent. In the studied system, the robot and the supervisor are presented as active entities with two different levels of intelligence. We will try to evoke the different concepts related to the domain before considering the MAS as a simulation tool.

The simulation model based on MAS is divided on 7 agents: robot, plants, greenhouse, horticulturist, UV-C lamps, charging station and supervisor (Mazar et al., 2020). There are several interactions between the agents, which are the following:

- Farmer - Supervisor: the farmer can define and control the execution of the missions by interacting with the Supervisor agent.
- Supervisor - Robot: the supervisor (Supervisor agent) plans missions for the robot.
- Robot - Supervisor: the robot sends information (battery status, location, plant disease level) to the supervision.
- Robot - UV-C lamps: according to the presence, or not, of the disease in the plants, the robot turns on and off the lamps.
- UV-C lamps - Robot: each lamp is installed on a robot.
- Robot - Plants: the robot treats the diseases on the plants by using the UV-C lamps.
- Robot - Greenhouse: the robot moves in the greenhouse.
- Robot - Charging station: after each mission, the robot returns to the charging station to recharge.
- Plants - Greenhouse: the plants are in the greenhouse, where they evolve and are treated.
- Charging station - Greenhouse: the charging station is in the greenhouse.

The evolution of the infection level of the plant by mildew directly influences the UV-C dose to be applied, i.e. the duration of the treatment. To adjust the UV-C treatment doses, the robot changes its speed according to the infection level of the plant. When the infection level is high, the robot treats the plant at low speed. This ensures that the plant receives a sufficient dose of UV-C radiation. Therefore, the energy consumption of the robot is proportional to the applied treatment dose. Since the UV-C lamps account for most of the robot's energy consumption, when the robot moves slowly with lamps activated, they consume more energy while the movement consumption is much lower.

The reproduction of the disease behavior in the simulator is the most important point of the UV-Robot system environment. Because to study the stochastic process of the evolution of blight in greenhouses, the simulation of its behavior must be close to reality. A Markov model was developed to have a local behavior of the disease specific to each plant, allowing to reproduce a global behavior target of the greenhouse (Kemeny and Snell, 1976). Simulation tests of the disease behavior with Markovian transition matrices have given good results. This study is done in (Mazar et al., 2021), where the disease behavior reproduced in the simulator is close to the real behavior in the greenhouse.

In this paper, we will study the system in a semi-dynamic case, where the disease behavior is stable for 24 hours. In order to respect the evolution of the disease behavior developed by the Markov model, a temporary variable has been added in the plant agent that allows to compute the dynamic evolution of the disease but without touching the real plant level in the simulator. After each 24 hours, the temporary variable replaces the real disease level variable of each plant, to allow a disease evolution that respects the real behavior of blight.

The figure 3 shows the graphic interface of the simulator. It is developed with NetLogo software, which is a multiagent programmable modeling environment. The user can observe the evolution of the robot treatment and can control it using all the green button and can observe some indicators on the yellow ones. The level of mildew is represented by different color of plants. The user can define the size of the greenhouse by choosing the total number of plants, and the number of plants in each row. He can also define the horizon of the simulation, the rate of diseased plants at the beginning of the simulation, and he can choose an optimization algorithm. If the chosen algorithm is the genetic one, the user can modify the parameters of this algorithm. There are other choices, like the treatment type scenario. If the chosen treatment type is preventive, he can choose the treatment period and the speed of the robot during the treatment. Finally, he can define the

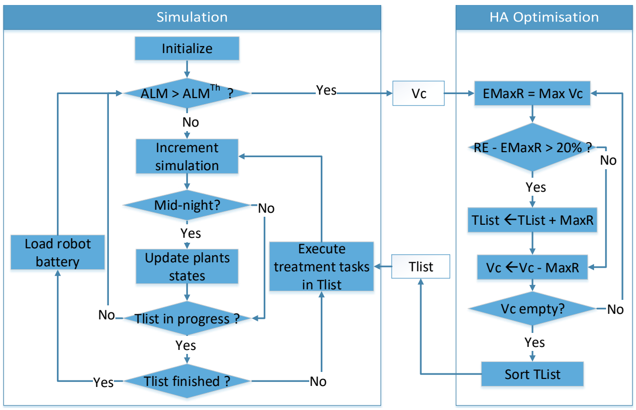


Fig. 1. Sim-optimization using heuristic algorithm

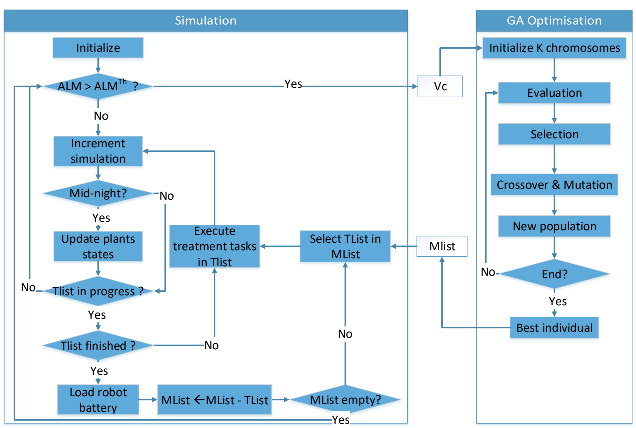


Fig. 2. Sim-optimization using genetic algorithm

days and time of treatment. NetLogo is very useful for our application because of its flexibility in developing algorithms and accepting the external language. In the case of this study, we have developed optimization algorithms and integrated them into the code of the monitoring agent. These algorithms will be presented in the next section.

3.2 Optimization

In order to optimize and analyze the UV-Robot system in the semi-dynamic case, we have developed a greedy heuristic and a meta-heuristic, which is the genetic algorithm. These algorithms have been evaluated in a previous work in a static case by comparing their results with the exact solution obtained by the commercial solver "FICO® Xpress" in Mazar et al. (2020).

Heuristic Algorithm (HA):
This heuristic is a greedy algorithm close to Best-Fit, where it first selects rows that have maximum energy consumption. When a row cannot be selected in a mission, the algorithm looks for another row that requires a lower amount of energy. It goes through all the rows until it has finished completing the mission. When a mission is completed, it moves on to the next one.

The proposed heuristic (cf. figure 1) is a greedy algorithm that assigns processing tasks to robot missions iteratively. At the beginning of each iteration, the algorithm initializes the vector of energy consumption of tasks Vc, which corresponds to the diagonal of the matrix W (line 2), the battery charge E and the task list `TASKS=[]` are initialized as empty (line 3 and 4). To be sure that the robot can return to the charging station, a safety power $(Max_k w_{k0})$ will be removed, it corresponds to the power needed to travel the maximum distance between the charging station and the furthest row (line 6). The assignment rule for selecting tasks for a mission is as follows: the first task with the highest energy consumption that can be added (lines 8 and 9), it must be less than or equal to the remaining energy capacity minus the power needed to travel to its row from the row of the last added task. If the power is sufficient, the processing of the selected row is added to the task (line 11) and the corresponding energy is removed from E (line 12), and this process continues until all remaining rows are tested.

Genetic algorithm (GA):
The chromosome of the genetic algorithm is coded as a matrix containing several missions. Each row of the matrix is a mission and each column represents a row of the greenhouse. If the robot has to process the row j during the mission i, the value of the element (gene) (i, j) is equal to one, otherwise it is equal to zero.
The flowchart of the genetic algorithm is given in the (cf. figure 2). The algorithm starts by generating S_P matrices of individuals which constitute the initial population. As long as the stopping condition (number of generations) is not satisfied, a new generation is created. In each generation, ordinary genetic operators are used to constitute each population. After each crossing of the two parents, the algorithm takes the obtained children to perform a mutation. After the mutation, each child is evaluated through the calculation of its fitness function. If it does not satisfy the constraints, it will not be selected in the new generation. At the end, the algorithm returns the best individual, which contains the minimum number of missions.

4. EXPERIMENTS AND DISCUSSIONS

GA and HA optimization algorithms are tested in a semi-dynamic case, to see which one is more efficient when the disease evolves every 24 hours. The simulation tests show the difference in the approach of coupling simulation and optimization with HA and GA, i.e. the difference between the use of decision-making at each state (the case of HA), and a global decision-making in a time T in order to improve the system state in T+1 (the case of GA). The experimental design is defined according to the following parameters:

- Optimization algorithm: GA and HA
- Disease evolution: semi-dynamic (evolution every 24 hours)
- Number of robots: 1
- Robot autonomy: 3 hours of treatment for 2 hours and 30 min of loading
- Robot speed: the robot adapts its speed for each disease level
- Greenhouse size: 100 rows
- Initial infection rate (IR^0): 50% and 100%.
- Number of scenarios: 5
- 100 plants per row
- Simulation horizon: 8 days.

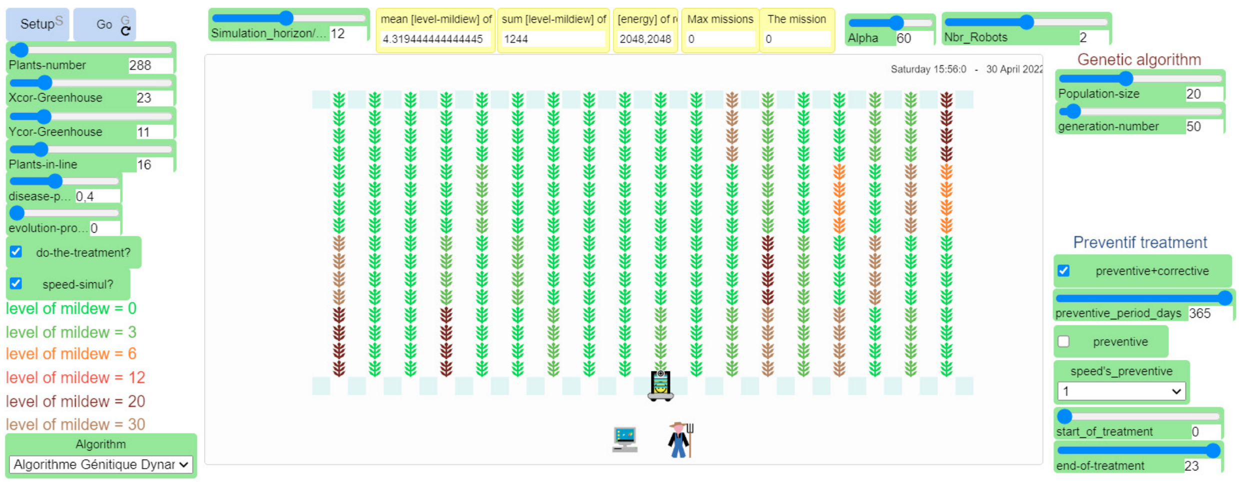


Fig. 3. Simulator interface

- Treatment threshold $\geq$ average blight in the greenhouse = 3
- Time of treatment: 5.00 pm to 7.00 am

The simulation tests are launched on 10 different greenhouses: 5 greenhouses with an initial average disease rate of 50% and 5 others with a rate of 100% (all plants are infected). The simulator is run for a period of 8 days for each test. The disease is updated every 24 hours. Treatments are performed only during the night because we assume that UV-C doses are more effective during the darkness (Janisiewicz et al., 2016).

Figure 4 shows the evolution over time of the average level of mildew in the greenhouse and the robot's battery level with the two optimization algorithms HA and GA with two values of initial infection rate of the green house. We can observe that, when looking to the average level of mildew is the greenhouse, the GA outperforms the HA. In fact, for both initial infection rates $IR^0 = 0.5$ and $IR^0 = 1$, the treatment tasks scheduling using the GA allows to decrease the average level of mildew more importantly and more rapidly than the HA. As it can be observed in sub-figures 4 (a) and (c), the average level of mildew reaches zero at the both during the third day. Whereas, we can see in sub-figures 4 (b) and (d) that it is not the same effect with the HA. However, the HA makes it possible to better manage energy consumption by respecting the minimum battery level of 20%, which should not be exceeded in order to prolong its life and avoid the risk of the robot "running out of power" in the middle of the greenhouse.

5. CONCLUSION

We studied the problem of scheduling robotized tasks in the context of Agriculture 4.0 with the objective of optimizing the treatment tasks of plants against an evolving disease (mildew) within a greenhouse. The treatments are performed using a robot to expose the infected plants to a specific dose of type-C ultraviolet radiation (UV-C). We proposed a hybrid semi-dynamic simulation-optimization approach based on a Markovian model of the disease behavior in the greenhouse. Two variants of hybridization approach are tested, and the results are analyzed to show the effectiveness of the approach and to understand the effect of some parameters of the model.

ACKNOWLEDGMENT

This research was made possible thanks to € 1.35 million financial support from the European Regional Development Fund provided by the Interreg North-West Europe Program in context of UV-Robot project.

REFERENCES

Abosuliman, S.S. and Almagrabi, A.O. (2021). Routing and scheduling of intelligent autonomous vehicles in industrial logistics systems. *Soft Computing*, 1–14.

Barenji, A.V., Barenji, R.V., Roudi, D., and Hashemipour, M. (2017). A dynamic multi-agent-based scheduling approach for smes. *The International Journal of Advanced Manufacturing Technology*, 89(9-12), 3123–3137.

Campi, M.C. and Garatti, S. (2018). Wait-and-judge scenario optimization. *Mathematical Programming*, 167(1), 155–189.

Carson, Y. and Maria, A. (1997). Simulation optimization: methods and applications. In *Proceedings of the 29th conference on Winter simulation*, 118–126.

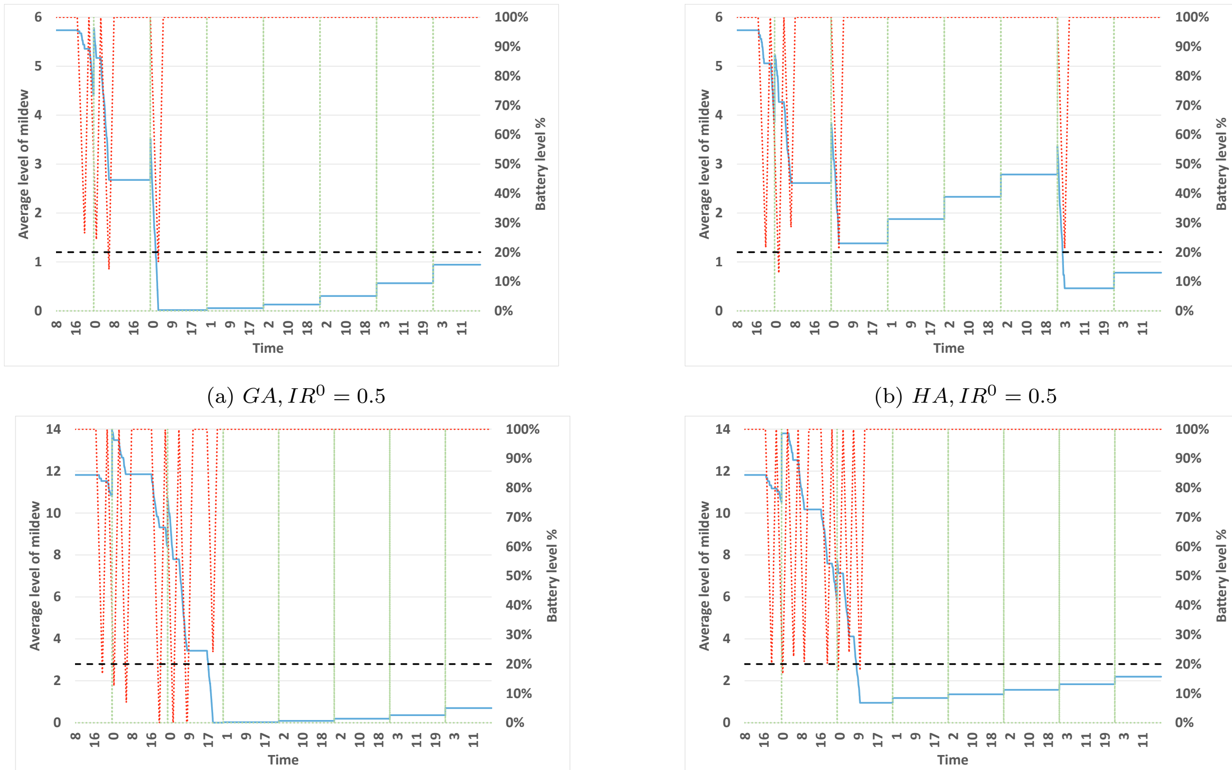
Farzanegan, A. and Vahidipour, S. (2009). Optimization of comminution circuit simulations based on genetic algorithms search method. *Minerals Engineering*, 22(7), 719–726.

Fu, M.C. (2002). Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3), 192–215.

Hassas, S. (2003). Systèmes complexes à base de multi-agents situés. *HDR, University Claude Bernard Lyon*.

Huff, D., Dai, S., and Hanrahan, P. (2021). Clockwork: Resource-efficient static scheduling for multi-rate image processing applications on fpgas. In *2021 IEEE 29th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 186–194. IEEE.

Janisiewicz, W.J., Takeda, F., Nichols, B., Glenn, D.M., Jurick II, W.M., and Camp, M.J. (2016). Use of low-dose UV-c irradiation to control powdery mildew caused by podosphaera aphanis on strawberry plants. *Canadian journal of plant pathology*, 38(4), 430–439.



(a) $GA, IR^0 = 0.5$

(b) $HA, IR^0 = 0.5$

(c) $GA, IR^0 = 1$

(d) $HA, IR^0 = 1$

Fig. 4. Evolution of the average level of mildew and battery level

Karimi, H., Bahmani, R., and Jadid, S. (2021). Stochastic multi-objective optimization to design optimal transactive pricing for dynamic demand response programs: A bi-level fuzzy approach. *International Journal of Electrical Power & Energy Systems*, 125, 106487.

Kemeny, J.G. and Snell, J.L. (1976). *Markov chains*, volume 6. Springer-Verlag, New York.

Mazar, M., Bettayeb, B., Klement, N., Sahnoun, M., and Louis, A. (2021). Multi-agent simulation to predict global behavior of population based on elementary local markovian model. In *2021 1st International Conference On Cyber Management And Engineering (CyMaEn)*, 1–5. IEEE.

Mazar, M., Sahnoun, M., Bettayeb, B., and Klement, N. (2018). Optimization of robotized tasks for the UV-c treatment of diseases in horticulture. In *The 2018 International Conference of the African Federation of Operational Research Societies (AFROS 2018)*.

Mazar, M., Sahnoun, M., Bettayeb, B., Klement, N., and Louis, A. (2020). Simulation and optimization of robotic tasks for UV treatment of diseases in horticulture. *Operational Research*, 1–27.

Ouelhadj, D. and Petrovic, S. (2009). A survey of dynamic scheduling in manufacturing systems. *Journal of scheduling*, 12(4), 417–431.

Powell, W.B. (2005). The optimizing-simulator: Merging simulation and optimization using approximate dynamic programming. In *Proceedings of the 37th conference on Winter simulation*, 96–109. Winter Simulation Conference.

Powell, W.B. (2008). Approximate dynamic programming: lessons from the field. In *Winter Simulation Conference 2008*, 205–214. IEEE.

Ren, W., Yan, Y., Hu, Y., and Guan, Y. (2021). Joint optimisation for dynamic flexible job-shop scheduling problem with transportation time and resource constraints. *International Journal of Production Research*, 1–22.

Sahnoun, M., Bettayeb, B., Bassetto, S.J., and Tollenaere, M. (2016). Simulation-based optimization of sampling plans to reduce inspections while mastering the risk exposure in semiconductor manufacturing. *Journal of Intelligent Manufacturing*, 27(6), 1335–1349.

Shi, L., Guo, G., and Song, X. (2021). Multi-agent based dynamic scheduling optimisation of the sustainable hybrid flow shop in a ubiquitous environment. *International Journal of Production Research*, 59(2), 576–597.

Suresh, V. and Chaudhuri, D. (1993). Dynamic scheduling—a survey of research. *International journal of production economics*, 32(1), 53–63.

Tubilla, F. and Gershwin, S.B. (2021). Dynamic scheduling in make-to-stock production systems with setup times and random breakdowns: performance analysis and improved policies. *International Journal of Production Research*, 1–19.

Wu, T., Powell, W.B., and Whisman, A. (2003). The optimizing simulator: An intelligent analysis tool for the military airlift problem. *Unpublished Report. Department of Operations Research and Financial Engineering, Princeton University, Princeton NJ.*

Zhai, Z., Martínez, J.F., Beltran, V., and Martínez, N.L. (2020). Decision support systems for agriculture 4.0: Survey and challenges. *Computers and Electronics in Agriculture*, 170, 105256.