Dynamic Scheduling of Robotic Mildew Treatment by UV-c in Horticulture

Merouane Mazar, Belgacem Bettayeb, Nathalie Klement, M'hammed Sahnoun, Anne Louis

Dynamic scheduling of robotic mildew treatment by UV-c in horticulture

Merouane Mazar, Belgacem Bettayeb, Nathalie Klement, M'hammed Sahnoun, Anne Louis

Abstract Thanks to new technologies, it is possible to make an automatic robotic treatment of plants for the mildew in greenhouses. The optimization of the scheduling of this robotic treatment presents a real challenge due to the continues evolution of disease level. The conventional optimization methods can not provide an accurate scheduling able to eliminate the disease from the greenhouse. This paper proposes a solution to provide a dynamic scheduling problem of evolutionary tasks in horticulture. We first developed a genetic algorithm (GA) for a static model. Then we improved it for the dynamic case where a dynamic genetic algorithm (DGA) based on the prediction of the task amount is developed. To test the performance of our algorithms, especially for the dynamic case, we integrated our algorithms in a simulator.

Keywords Dynamic scheduling, Simulation, Optimization, Multi-agent system, Mildew.

Merouane Mazar
LINEACT - CESI. 80, rue Edmund Halley, Rouen Madrillet Innovation, 76800 Saint-Étienne-du-Rouvray, France e-mail: mmazar@cesi.fr

Belgacem Bettayeb
LINEACT - CESI. 8, boulevard Louis XIV, 59046 Lille, France e-mail: bbettayeb@cesi.fr

Nathalie Klement
Arts et Métiers Institute of Technology, LISPEN, HESAM Université, F-59000 Lille, France e-mail: Nathalie.Klement@ensam.eu

M'hammed Sahnoun
LINEACT - CESI. 80, rue Edmund Halley, Rouen Madrillet Innovation, 76800 Saint-Étienne-du-Rouvray, France e-mail: Nathalie.Klement@ensam.eu

Anne Louis
LINEACT - CESI. 80, rue Edmund Halley, Rouen Madrillet Innovation, 76800 Saint-Étienne-du-Rouvray, France e-mail: alouis@cesi.fr

1 Introduction

Robotics knows a huge evolution and starts to be used in several fields such as industry, rehabilitation, and agriculture. In horticulture, researchers are developing several types of robots to cultivate or treat plants [15]. In the literature, several works on harvesting robots can be found, like the robot presented in [16], which has an artificial vision to move easily between the cauliflower plants. In [14], another harvesting robot that collects watermelons is presented, which shows the ability of some robots to harvest heavy crops. Moreover, there are also robot equipped with sprayers that are used for plant treatments. For instance, [11] describes the design of a robot able to detect the powdery mildew and spray the diseased plants in order to reduce the quantity of spray and avoid the treatment of safe plants. The work presented in this paper is part of an European project called UV-Robot. In our case, we have a robot that treats the mildew fungus in horticulture using type C ultraviolet radiation (UV-c). UV-Robot replaces the chemical spray treatment with UV-c treatment. In [17], the authors have shown that UV-c treatment improves the production of strawberries.

The UV-Robot must treat the rows of plants that are affected by mildew during their growth. The energy supply of the robot is based on a battery that can ensure a continuous functioning during 3 hours in average and needs 2.5 hours to be fully charged. Mildew disease has an evolutionary and spreading behavior that follows a stochastic process. To achieve an optimal treatment schedule, decision support tools are needed. Simulation-optimization approach allows to solve the dynamic scheduling of UV-Robot systems. This approach has showed good performance in several works, and it has the advantage of allowing the simulator to learn and adapt over time with the best behavior [20]. For instance, [12] works on the problem of air transport of military aircraft of the United States using approximate dynamic programming. This approach is also used within the simulators of [8] and [21] which are developed to optimize the vehicles routing for the collection of conditioned biowaste.

Since few decades, Multi-Agent Systems (MAS) paradigm has emerged as an effective approach for complex systems modelling and simulation. [18] describes the MAS environment as the context in which the agents will evolve. For [4], a MAS is a set of entities called agents, sharing a common environment, which they are able to perceive and on which they can act. The simulator developed within this work represents the environment for testing our UV-Robot processing system. We will run optimization algorithms inside the simulator in order to plan the robot tasks.

Dynamic scheduling is a problem studied in several systems, such as on robots, manufacturing machines or distribution chains. Several works have studied this problem from which we can cite the article by [5], where a machine processes jobs that arrive continuously. A case on several resources (multiprocessor in a computer) is presented in [13], which executes a set of tasks that arrive dynamically. In dynamic scheduling, it is generally the time of arrival or departure (duration) of tasks

which is dynamic. In our case, both the occurrence and spread of mildew disease on plants are dynamic.

In the case of dynamic duration , we explored the lead of scheduling problems with deteriorating jobs. In this problem, the tasks duration follow a degradation process, as in [7], where the temperature of the ingots drops after they come out of the oven. In [6] and [1], the tasks are degraded in the same way starting from T_0 according to a linear degradation equation. The processing of tasks is done by job batch in [7], [6] and [1]. These batches can be processed in parallel, which is not possible in our case with a single robot.

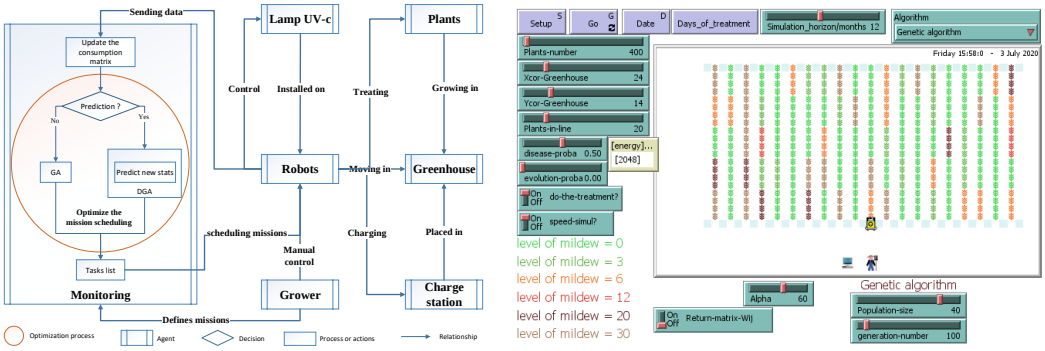
This paper proposes to resolve a dynamic scheduling problem with evolving task duration due to the evolution of the mildew level. A sim-optimization approach based on a genetic algorithm and a multi-agent simulation is used to implement the dynamic solution. NetLogo [19] simulation software is used to implement the optimization and simulation algorithms. In this study, the agent based architecture is used for its capacity to present and simulate complex systems with centralized GA based decision making (only one active agent). To the best of our knowledge, there is no work proposing the resolution of dynamic scheduling of robot in horticulture.

The article is organized as follows. Section 2 presents the simulation model with MAS. Section 3 details the disease behavior model and its estimation. Then, two optimization algorithms are proposed in Section 4. Section 5 describes and discusses the results obtained. Finally, the conclusion and the perspectives of this work are given in Section 6.

2 Simulation model

This section presents the simulation model of the robotized mildew treatment process, which is based on MAS, and explains the role of agents and the interactions between them. Before building the simulation model, we carefully studied our system to define all the agents. In the UV-Robot treatment system, a robot equipped with UV-lamps performs treatment missions of infected plant rows in the greenhouse. The robot moves to the battery charging station after each mission. The robot is also equipped with a smart e-nose to inspect the level of plant disease. The smart e-nose absorbs chemicals substances around the plants, then calculates the level of mildew for each plant section. The robot performs a measurement of the entire greenhouse using e-nose and then begins treatment knowing that the disease progresses during treatment. However, the robot cannot launch a mission before being given the authorisation from the monitoring agent. The monitoring system is composed of a central computer able to control the robot missions, collect data about the greenhouse and represent the state of the system on a dashboard for the grower. It allows the grower to manually control the robot, plan its missions and update the mildew level.

The model of our multi-agent system and a screenshot of the developed simulator are shown in Fig. 1. This system contains seven agents interconnected by some



(a) Agent-based conceptual model

(b) Screenshot of the Netlogo Simulator

Fig. 1: Agent-based conceptual model and Simulator

directed links that represent the interactions between them. In our model, we only use the capacity of MAS to reduce the complexity of systems in terms of modeling and simulation. The model have an active "Monitoring" agent, which receives all the information from the other agents and then makes decisions regarding plants treatment and missions planning. The "Plants" and "Robots" agents cannot make any decision and they are modeled as completely reactive agents. Fig. 1-(a) presents a more detailed view to show the operating diagrams of some agents. Each agent has its own behavior, and can interact or communicates with one or more agents.
The "Grower" agent defines missions for the monitoring, and can manually control the robot. The "Monitoring" agent receives information from the greenhouse and data from the robot, and runs the optimization algorithms to create the missions of the robot. The "Robots" agent has four main roles: (i) calculates the plant disease levels to send it to the monitoring; (ii) controls the UV-c lamps (turns them on or off); moves to the charging station when battery level is low; and (iv) moves between crop rows to treat infected plants. The "Plants" agent has a disease level that can increase and/or spread and that is reset to zero each time the plant is treated. The plants grow in the greenhouse and are treated by the robot. The agent "UV-c Lamp" is installed on the robot which control them. The "Charge station" agent is placed in the greenhouse and allows the robot to recharge its battery. The "Greenhouse" agent, which carries all the agents, is the environment that can influence the appearance of fungus (mildew).

3 Mildew's behavior model

The evolution of mildew infection level influences directly the UV-c dose to apply, i.e. the duration of 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 with a low speed, so the plant receives a sufficient dose of UV-c radiation. Moreover, the energy consumption of the robot is proportional to the applied treatment dose. As UV-c lamps represent the biggest part of the robot energy consumption, when the robot is moving slowly with lamps turned-on, it consumes more energy even if the consumption of motor is low.

In order to properly calibrate our resolution algorithms in our system, the behavior of mildew needs to be simulated to bring it closer to reality. We used data from [2], which represents the evolution over time of the level of mildew in vineyards in 2007. Fig. 2 shows the IGT2007 mildew behavior curve, and our estimation curve $\hat{f}(t)$. This data can be approached by the time series in discreet or continues representation.

$$\hat{f}(t) = \frac{c}{1 + be^{-at}} \tag{1}$$

The estimation function (1) is a three-parameter logistic function that represents the time series. Its parameters are the following.

- The numerator c is the limit of the function at infinity (the curve peaks under a horizontal asymptote).
- The function is symmetrical with respect to its inflection point of abscissa $\frac{ln(b)}{a}$ and ordinate $\frac{c}{2}$.
- $c = 30$ is the maximum level of disease.

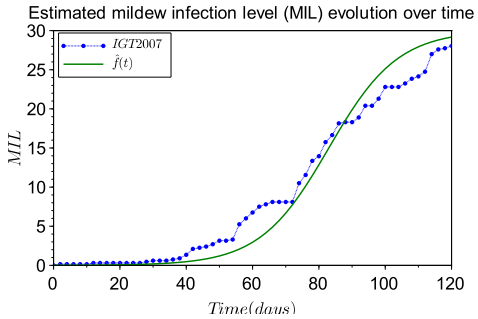


Fig. 2: Real mildew behavior IGT2007 [2] and estimated logistic function $\hat{f}(t)$

To compute a and b, we take a point in the graph from the IGT2007 curve $(40, 0.468)$ and construct two following equations (2) and (3):

$$\frac{c}{1 + be^{-40a}} = 0.468 \tag{2}$$

$$\frac{ln(b)}{a} = 83.054 \tag{3}$$

After solving both equations (2) and (3), we obtain $a \approx 0.096$ and $b \approx 2936$. Thus $\hat{f}(t)$ is given by equation (4).

$$\hat{f}(t) = \frac{30}{1 + 2936e^{-0.096t}} \tag{4}$$

To get a simulated disease behavior similar to the one represented by function (4), we proceed by trying empirically several evolution probabilities that steer the transition of plants infection rate from the current level to the superior one. We carried out several tests using the simulator. The retained probability function of disease level transition is given by equation (5).

$$P = 0.000005 * [level\_mildew] * [last\_treatment] \tag{5}$$

$[level\_mildew]$ is the plant's current level of mildew and $[last\_treatment]$ is the number of days elapsed since the last treatment. For the propagation of disease between neighbor plants, we check if there is an infected plant within a radius of 3 meters and, if it is the case, the probability P is increased by 0.01.
There are 6 disease levels (0, 3, 6, 12, 20 and 30) defined with our partners in the UV-Robot project. The speed of the disease evolution is not linear and can be assimilated to the function $\hat{f}(t)$. The speed of the disease increase is given by the derivative function $d\hat{f}(t)/dt$. For example, the speed is low at the beginning (between day 0 and day 40) and at the end (after day 120) whereas it is much higher around the infection point (day 83).

4 Dynamic Scheduling

Bin-packing is a known operational research problem that can be used for the modeling of robotic task planning. It consists in filling bins with items while respecting the size of the bins. The goal of this problem is to minimize the number of used bins. In our problem, bins correspond to missions, and items correspond to treatment tasks performed during missions. The size of each mission is limited by the capacity of the battery that provides the electric energy necessary for robot movement and for UV-lamps operation when performing treatment tasks. As the objective is to eradicate the disease from the greenhouse as soon as possible, the goal is to minimize the number of missions. Table 1 summarizes the analogy between our problem and the bin-packing problem. The mathematical model of this problem is detailed in [9]. Authors have already resolved it in the static case using a genetic algorithm and an exact method. The model will not be detailed in this paper.

In this section, two genetic algorithms are proposed to solve the problem of treatment missions planning for semi-static and dynamic cases. The dynamic case takes into account the evolution behavior of mildiou in the greenhouse.

Bin-packing problem	Our problem
Bins	Missions
Items	Infected plant rows treatment tasks
Size of an item	Needed energy for the treatment task
Capacity of the bin	Capacity of the battery
Minimize the number of bins	Minimize the number of missions

Table 1: Analogy between our problem and the bin-packing problem

4.1 Genetic algorithm in semi-static case

First, the genetic algorithm (GA) has been developed to resolve a semi-static case, where the disease behavior changes every 24 hours. This period allows a long time for the robot to execute its missions with a stable level of disease. We used the operations of a classic GA (selection, crossing and mutation) [3]. The charging station is in the middle of the greenhouse. In order to optimize displacements, the robot does all the treatments by visiting the rows selected within a mission in an ascendant order of their identification number (from the left to the right of the greenhouse).

The coding scheme of an individual in the GA is represented by a matrix as shown in Fig. 3, where lines represent the treatment missions and columns are the rows of the greenhouse. If a row j is to be treated during a mission i, the value of the element ij is equal to 1. The sum of power consumption of each mission (vector multiplication of "mission" line vector and "consumption" column vector) should not exceed the robot battery capacity, otherwise the individual is discarded. The evaluation of each chromosome takes into account the energy consumed for the displacements of the robot to reach the charging station. The initial population is created with a greedy heuristic, which takes the treatments of large size at the beginning and fulfill the missions while respecting the capacity of the robot. Since the goal is to minimize the number of missions, the chromosome with fewer missions is the best individual. Fig. 3 presents the crossing between two parents at a random point which gives two children as the output.

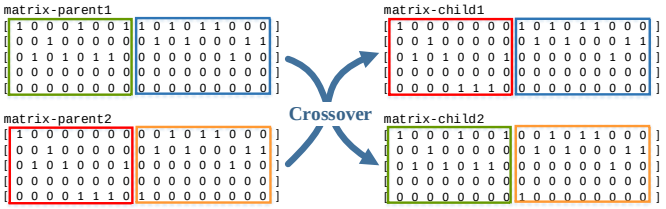


Fig. 3: Crossover operator

Fig. 4 shows how the mutation operation works in the proposed GA. Each child has a 60% of chance to be mutated. During the mutation, the algorithm randomly

selects 2 lines of the chromosome matrix. Then, each element equal to 1, in each line, has a 50% chance to switch with the corresponding element of the other line (element with the same j). After each iteration of GA, 10% of the best individuals are selected to be included in the new generation.

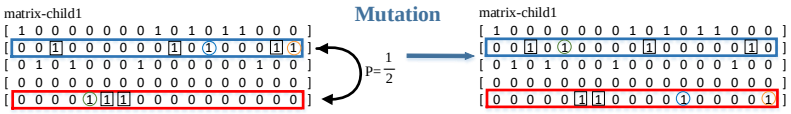


Fig. 4: Mutation operator

4.2 Dynamic genetic algorithm

The proposed dynamic genetic algorithm (DGA) is considered as an improvement of the GA in the dynamic case. Since the previous GA is limited when disease behavior changes over and over in the greenhouse, we improved it while keeping certain operations. The added value of the DGA is the use of the function $\hat{f}(t)$ to predict the evolution of mildew. When the algorithm fulfills the missions by the processing tasks, it takes into account an additional energy consumption. Fig. 5 shows a diagram of the operation of the DGA. The predicted disease level increases over time because there is a waiting period before the execution of each mission. The waiting time of a mission is the execution time of the previous mission plus the battery charging time. Fig. 6 shows the difference between DGA and GA regarding

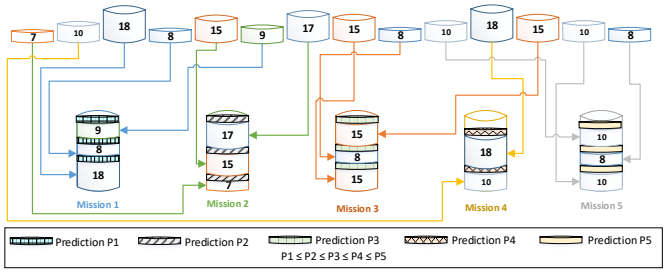


Fig. 5: Representation of the dynamic fulfilment of treatment tasks in missions

energy consumption. DGA always allocates a part of the battery capacity to the expected additional consumption, due to the evolution of infection level, for each mission. On the opposite, GA does not take into account the evolution of the disease. Fig. 6 also shows the execution times for missions with a treatment time of 3 hours

and a charging time of 2 hours and 30 minutes. We used the estimation function $\hat{f}(t)$ for the prediction calculation P_i as shown in equation (6). In fact, we need to estimate the amount of the additional increase of infection level P_i at the execution moment of the i^{th} mission. For that, we first estimate the global level of disease using the function $\hat{f}(t)$ and then we remove the measured value recorded at the last measurement action (using e-nose or human estimation).

$$Pi = \frac{\hat{f}\left(3+5.5(i-1)+t_m\right)\alpha i}{Nbr_{plants}/4} - \mathcal{M} \tag{6}$$

α is an empirical parameter defined thanks to the simulation, $\mathcal{M}$ is the last measured mildew level for the concerned plant, and t_m is the estimated time corresponding to the value of the last measured mildew level ($\mathcal{M}$).

(a) GA (b) DGA

Fig. 6: Comparison of the dynamic and static attribution of treatment

5 Experimentation

We consider a greenhouse containing 100 rows of strawberry plants, each row has 100 meters of length and contains 100 plants. A UV-C robot with an autonomy of 3 hours and a charging time of 2.5 hours is used to treat the mildew. We assume that in the initial conditions, 50% of plants are infected with different levels of disease.

In the first step of experiment, we launched several simulations with DGA to calibrate its parameter α to comply the model developed for vineyards to the evolution of mildew for strawberry. Then, we launched 5 simulations of each algorithm (DGA and GA) in a dynamic environment where the disease evolves at each simulation step. The time horizon of each simulation is 20 hours, which is enough for the treatment of all infected plants in the greenhouse. In both the GA and the DGA, the population size is 20 and the limit number of generations is 50. These parameters were determined empirically through several trials. Fig. 7 draws the evolution of the level of the robot battery as a function of time. It compares three curves with different values of the parameter α (36, 50 and 60). The decreasing segment in the curves is linked to the time of treatment, and the increasing segment is relative to the time of battery charging. In order to increase the lifespan of the battery, we limit the consumption to 80% of battery's capacity, as it is recommended by experts [10]. To compare the curves in Fig. 7 we have added the line E_L which represents 20% of the battery charge. So we will choose DGA with $\alpha = 60$ (DAG60) for the next simulations because its curve respects the minimum level of energy.

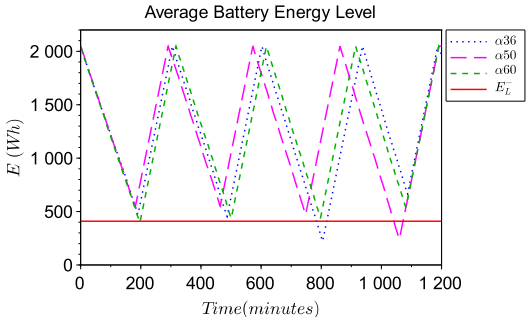
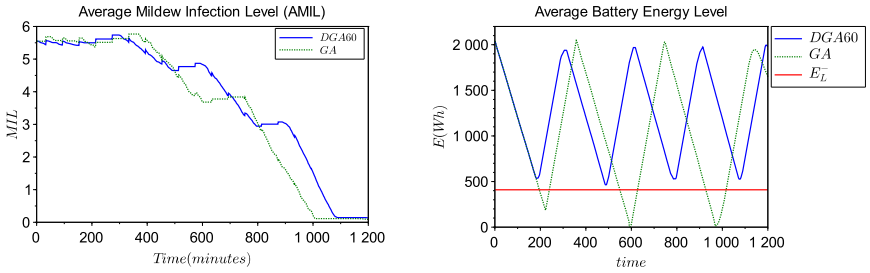


Fig. 7: Evolution of energy consumption for different values of α

Fig. 8a shows the evolution of level of disease over time. There are two curves in the figure: the blue solid curve which represents the level of mildew using DGA60 and the green pointed curve which is related to the use of GA. We can clearly observe that using GA allows the robot to finish the treatment of mildew in 1000 minutes, whereas it takes 1100 minutes using DGA. Moreover, we notice that using the GA is risky and does not allow using the robot in an autonomous way. In fact, during the execution of the scenario using GA, the grower restart manually the robot several times because it has not enough energy in the battery to return back to the charging station. In fact, an extra energy consumption may happen when the actual level of mildew is more than expected. We also notice that the level of disease increases from time to time in both graphs during the robot's charging period.



(a) Average Mildew Infection Level

(b) Energy consumption

Fig. 8: Comparison between *DGA*60 and *GA*

In Fig. 8b, there are also two curves relative to DGA60 and GA drawing the level of battery energy as function of time. Both curves correspond to the average of several simulations for each algorithm. The GA scenario does not respect the battery capacity constraint, where the robot uses more than 80% of its battery capacity. As

show on Fig. 8b, this level reaches a 100% of battery capacity several times, which necessitates the intervention of the grower to bring the robot to its charge station.

The average time for a simulation with GA is 16 minutes and 41 seconds, and that of DGA60 is 21 minutes and 36 seconds. This increase in computation time is due to the additional computation of the prediction of mildew level.

As a conclusion, we can say that even if the use of GA allows less treatments and computation times, it is still a risky scenario because it can not ensure the total autonomy of the robot and needs the intervention of the grower in several times. Moreover, the use of DGA60 can be considered more efficient because it respects the capacity constraint concerning the use of only 80% of the total battery capacity, which allows a total autonomy of the robotic treatment. In addition, the use of DGA60 gives realistic scenarios and allows possible real implementation.

6 Conclusion

This paper studies the dynamic task scheduling problem, applied to UV-c treatment of plants in horticulture. The difficulty was in scheduling tasks to treat the disease which has a dynamic evolutionary behavior. The use of simulator allows to test our algorithms in the dynamic case. We improved a Genetic Algorithm (GA), previously proposed, to Dynamic Genetic Algorithm (DGA) allowing the autonomous execution of treatment with respect to the battery capacity constraint in the dynamic environment. The results provided by DGA show better accuracy of the treatment with more respect of the technical battery constraints and give the possibility to launch real life horticultural tests.
In perspective, we plan to add a preventive treatment that allows to control the evolution and the propagation of disease. We will introduce the case of a multi-robots for several greenhouses, which will increase the number of active agents and allows the use of distributed intelligence such as Contact net protocol, or Potential Fields.

7 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

1. TCE Cheng, Liying Kang, and CT Ng. Due-date assignment and single machine scheduling with deteriorating jobs. *Journal of the Operational Research Society*, 55(2):198–203, 2004.

2. Magnien Claude. Mildiou de la vigne - bilan de la campagne 2007. In *Actualités Phytosanitaires*, pages 99–105. IFV, 2007.
3. Lawrence Davis. *Handbook of genetic algorithms*. CumInCAD, NY, 1991.
4. Salima Hassas. Systèmes complexes à base de multi-agents situés. *Mémoire d'Habilitation à Diriger les Recherches. University Claude Bernard Lyon*, 2003.
5. Jingzhu Li, Peng Wang, and Changxing Geng. The disease assessment of cucumber downy mildew based on image processing. In *Computer Network, Electronic and Automation (ICCNEA), 2017 International Conference on*, pages 480–485. IEEE, 2017.
6. Jun-Qing Li, Mei-Xian Song, Ling Wang, Pei-Yong Duan, Yu-Yan Han, Hong-Yan Sang, and Quan-Ke Pan. Hybrid artificial bee colony algorithm for a parallel batching distributed flowshop problem with deteriorating jobs. *IEEE transactions on cybernetics*, 2019.
7. Shisheng Li, CT Ng, TC Edwin Cheng, and Jinjiang Yuan. Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan. *European Journal of Operational Research*, 210(3):482–488, 2011.
8. Merouane Mazar, Vincent Constant-Meney, M'hammed Sahnoun, David Baudry, and Anne Louis. Simulation et optimisation de la tournée des véhicules pour la collecte de biodéchets conditionnés. 2017.
9. Merouane Mazar, M'hammed Sahnoun, Belgacem Bettayeb, Nathalie Klement, and Anne Louis. Simulation and optimization of robotic tasks for uv treatment of diseases in horticulture. *Operational Research*, pages 1–27, 2020.
10. Yongguo Mei, Yung-Hsiang Lu, Y Charlie Hu, and CS George Lee. A case study of mobile robot's energy consumption and conservation techniques. In *ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, pages 492–497. IEEE, 2005.
11. Roberto Oberti, Massimo Marchi, Paolo Tirelli, Aldo Calcante, Marcello Iriti, Emanuele Tona, Marko Hočevar, Joerg Baur, Julian Pfaff, Christoph Schütz, et al. Selective spraying of grapevines for disease control using a modular agricultural robot. *Biosystems Engineering*, 146:203–215, 2016.
12. Warren B Powell. Approximate dynamic programming: lessons from the field. In *Simulation Conference, 2008. WSC 2008. Winter*, pages 205–214. IEEE, 2008.
13. Jyoti Sahni and Deo Prakash Vidyarthi. A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment. *IEEE Transactions on Cloud Computing*, 6(1):2–18, 2015.
14. Satoru Sakai, Michihisa Iida, Koichi Osuka, and Mikio Umeda. Design and control of a heavy material handling manipulator for agricultural robots. *Autonomous Robots*, 25(3):189–204, 2008.
15. F Sistler. Robotics and intelligent machines in agriculture. *IEEE Journal on Robotics and Automation*, 3(1):3–6, 1987.
16. Ben Southall, Tony Hague, John A. Marchant, and Bernard F. Buxton. An autonomous crop treatment robot: Part i. a kalman filter model for localization and crop/weed classification. *The international journal of robotics research*, 21(1):61–74, 2002.
17. Fumiomi Takeda, WJ Janisiewicz, BJ Smith, and B Nichols. A new approach for strawberry disease control. *Eur J Hortic Sci*, 84(1):3–13, 2019.
18. John Tranier. *Vers une vision intégrale des systèmes multi-agents*. PhD thesis, Université Montpellier II, Montpellier, Thèse de doctorat, 2007.
19. Uri Wilensky and I Evanston. Netlogo: Center for connected learning and computer-based modeling. *Northwestern University, Evanston, IL*, pages 49–52, 1999.
20. Tongquiang Wu, Warren B Powell, and Alan Whisman. 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*, 2003.
21. Yiyi Xu, M'hammed Sahnoun, Merouane Mazar, Fouad Ben Abdelaziz, and Anne Louis. Packaged bio-waste management simulation model application: Normandy region, france. In *2019 8th ICMSAO*, pages 1–5. IEEE, 2019.