A Cloud-Based In-Field Fleet Coordination System for Multiple Operations

Caicong Wu [1,2], Zhibo Chen [1], Dongxu Wang [1], Bingbing Song [1], Yajie Liang [1], Lili Yang [1,*] and Dionysis D. Bochtis [3,*]

[1] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China; wucc@cau.edu.cn (C.W.); chenzb@cau.edu.cn (Z.C.); Dongxu_Wang@cau.edu.cn (D.W.); songbing@cau.edu.cn (B.S.); liangyajie@cau.edu.cn (Y.L.)
[2] Key Laboratory of Remote Sensing for Agri-Hazards, Ministry of Agriculture, Beijing 100083, China
[3] Institute for Bio-economy and Agri-technology (IBO), Centre for Research and Technology—Hellas (CERTH), 6th km Charilaou-Thermi Rd, GR 57001 Thermi, Thessaloniki, Greece
* Correspondence: llyang@cau.edu.cn (L.Y.); d.bochtis@certh.gr (D.D.B.)

check for updates

Abstract: In large-scale arable farming, multiple sequential operations involving multiple machines must be carried out simultaneously due to restrictions of short time windows. However, the coordination and planning of multiple sequential operations is a nontrivial task for farmers, since each operation may have its own set of operational features, e.g., operating width and turning radius. Taking the two sequential operations—hoeing cultivation and seeding—as an example, the seeder has double the width of the hoeing cultivator, and the seeder must remain idle while waiting for the hoeing cultivator to finish two rows before it can commence its seeding operation. A flow-shop working mode can coordinate multiple machines in multiple operations within a field when different operations have different implement widths. To this end, an auto-steering-based collaborative operating system for fleet management (FMCOS) was developed to realize an in-field flow-shop working mode, which is often adopted by the scaled agricultural machinery cooperatives. This paper proposes the structure and composition of the FMCOS, the method of operating strip segmenting, and a new algorithm for strip state updating between successive field operations under an optimal strategy for waiting time conditioning between sequential operations. A simulation model was developed to verify the state-updating algorithm. Then, the prototype system of FMCOS was combined with auto-steering systems on tractors, and the collaborative operating system for the server was integrated. Three field experiments of one operation, two operations, and three operations were carried out to verify the functionality and performance of FMCOS. The results of the experiment showed that the FMCOS could coordinate in-field fleet operations while improving both the job quality and the efficiency of fleet management by adopting the flow-shop working mode.

Keywords: agricultural machinery; fleet management; auto-steering system; collaborative operating system; flow-shop; simulation; field experiment

1. Introduction

Agricultural fleet management is an important research domain that concerns resource allocation, operation scheduling, routing, communication, and real-time data acquisition [1]. Seamless communications and collaboration across spatial, temporal, operational, and machinery domains are a particular direction toward fleet management in scaled agricultural machinery service organization [2]. Sørensen and Bochtis [3] proposed that the total supply chain of large-scale harvesting has four levels of collaboration, i.e., in-field logistics, inter-field logistics, inter-sector logistics, and inter-regional logistics.

In terms of machinery collaboration, Wang et al. [2] proposed that in-field logistics can be subdivided into three levels, i.e., multi-operation coordination, multi-machinery coordination, and multi-factor coordination [4]. Typical cases of multi-operation collaboration include the master–slave approach [5–7] and leader–follower approach [8–10], which illustrate multiple machines working in parallel for different but related operating tasks. These systems refer to identical operations. Other approaches have been proposed for sequential operations. As an example, in [11], a multi-operation collaboration system was proposed for six sequential operations in a potato production system—namely, bed forming, stone separation, planting, reloading unit, spraying, and harvesting. In parallel, algorithms for optimizing in-field route planning have been developed in various studies [12–15]. Furthermore, the flow-shop working mode of agricultural machinery operation is somewhat more complex [16–19] when multiple units of machinery and multiple operations exist within one field during the same period [20].

For the implementation of all the approaches mentioned above, Real-Time Kinematic–Global Navigation Satellite System (RTK-GNSS)-based automated steering systems can provide automatic guidance for single agricultural machines with one-inch repeatability [21,22]. Auto-steering technology can reduce overlaps and skips and can reduce the dependence upon skilled operators [23]. Therefore, fusion of the efficient flow-shop working mode and advanced auto-steering technologies can improve the field operation efficiency of agricultural machinery. To realize this fusion, the guidance lines followed by the auto-steering systems and the operating progress should be shared among the agricultural machinery units of the fleet in real time, so as to allow the operators to choose the appropriate operating strips for working. It is worth mentioning that different agricultural machinery operations have differing operating strips (in terms of number and size) and, therefore, different guidance lines, since different agricultural machines probably carry different implements with different operating widths. Most auto-steering systems can share guidance lines between different agricultural machinery units for the same operation through U disk copying, radio transfer, and/or server exchange. For instance, Cao et al. [24] built a Web-GIS (Geographic Information System) to realize collaboration among multiple agricultural machinery units, and the John Deere company utilizes an MTG (Modular Telematics Gateway) intelligent gateway and JDLink software to share guidance lines between different agricultural machinery units [25]. To support multi-operation collaboration, Wang et al. [2] designed a cloud-terminal-based coordinative operating system for in-field fleet management. The authors gave the definition of an operating strip to replace the path [26], made use of intervals for strip identification, and proposed an algorithm for strip state updating between adjacent machinery operations. The research demonstrated how to calculate the waiting time or flow step accurately to reduce the delay between adjacent machinery operations [2,19]. However, the validation of the system took place through simulation, while the strip state-updating algorithm provided a feasible—yet not optimized—procedure.

In order to optimize the state-updating algorithm and to validate the functionality of the work presented in Wang et al. [2] under real field experimental conditions, a new collaborative operating system for fleet management (FMCOS) was redesigned and developed, which can support all the operating stages of a field crop and incorporates an upgraded algorithmic process for strip state updating that simplifies the calculation process. The algorithm was tested through simulations using Matlab (MathWorks®) before the development of FMCOS. Finally, three field experiments were carried out to validate FMCOS and to test its functionality and performance.

2. Description of the Fleet Management System

2.1. System Composition

Typically, a complete cropping cycle consists of water management, land cultivation, seeding, nutrient management, pest management, and harvesting, as illustrated in Figure 1 [27]. For scaled agricultural machinery cooperatives, adoption of the flow-shop working mode in the phase of land cultivation is needed in order to improve the operating efficiency in a single field. Therefore, FMCOS

should not only service the whole production cycle of the field crop, but also support the in-field flow-shop working mode for some specific operating phases. For this purpose, the auto-steering systems, a fixed GNSS base station, and a database that can manage the parameters of the field area and implements were deployed as a system. The baseline in each field can be used to generate guidance lines for all related machinery based on the width of the implement. FMCOS mainly includes two components: the GNSS-based auto-steering systems installed on the agricultural machinery and the cloud-based operation coordination system.
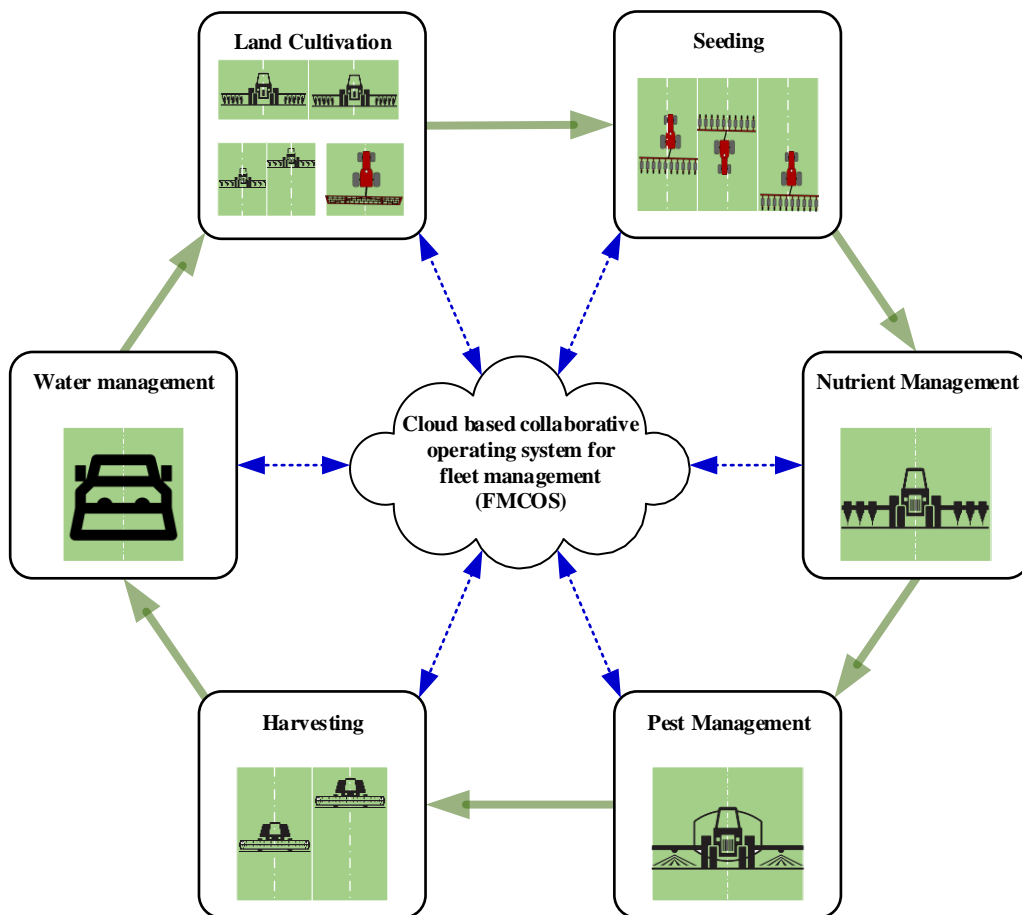
Figure 1. Collaborative application for the production cycle of field crop planting and management.

2.2. Operating Strip Segmentation

As shown in Figure 2, the field is segmented into strips for each machine. The width of the strip equals the width of the implement. Here, S_i^k refers to the ith strip of the kth operation, O^k. Strips are numbered from left to right. U_j^k are used to describe the machinery units.

A state shows the process progress in a strip by a machine and has four classes: *unready* (0), *ready* (1), *doing* (2), and *done* (3) (Table 1). When the task begins, the state of the strips that belong to the first operation is *ready*, and the machine arranged for the first operation can enter each of the strips for the first operation.

While the fleet is operating in the field, the strip states are updated in real time from the first operation to the last operation in turn. For a single operation, the strip states are updated from the left to the right. When the server gets the updated information of the operating strip from the Android application, the algorithm successively updates the strip state of the current operation according to the prior operating strip state.
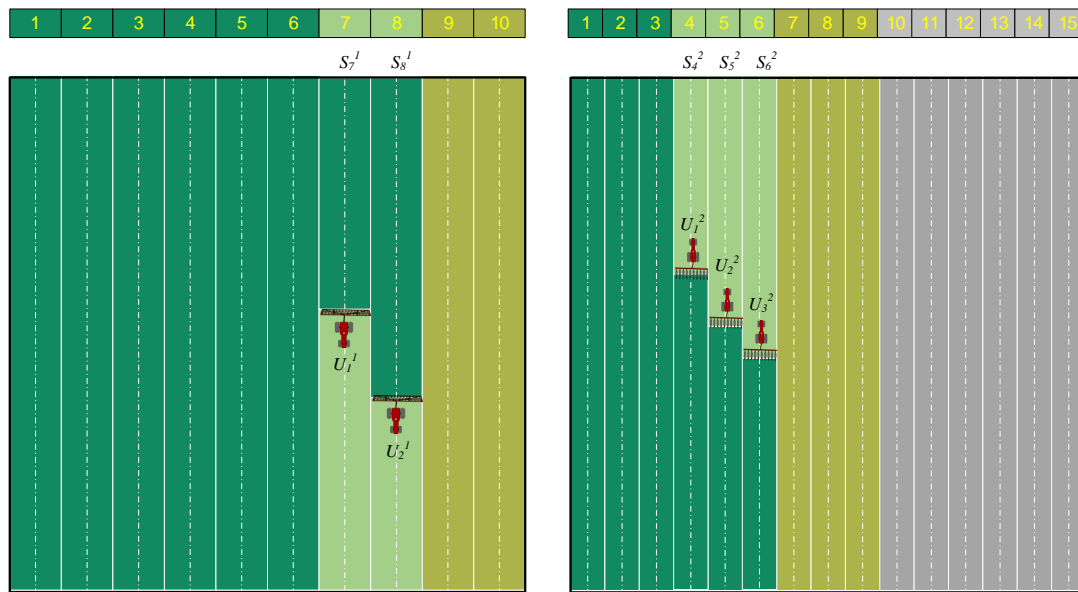
Figure 2. Expressions of operation tasks, operating strips, and machinery units.

Table 1. Definitions of operating strip states.

State	Color	Value	Definition
unready		0	The strip is not ready for operation.
ready		1	The strip is ready for operation.
doing		2	The strip is being processed by a machine.
done		3	The strip has been processed by a machine.

Figure 3 shows two operations, O^{k+1} and O^k. O^{k+1} is the operation after O^k, and the state of S_3^{k+1} of O^{k+1} is determined by three relevant strips of O^k, i.e., S_4^k, S_5^k, and S_6^k. This is because at the same location in the field, only when the related strips of O^k have been finished can the machinery for O^{k+1} enter the location. Therefore, the state (*ready* or *unready*) of O^{k+1} is determined by that of O^k. In other words, if, and only if, the state of the three strips becomes *done*, the state of S_3^{k+1} can be updated from *unready* to *ready*.
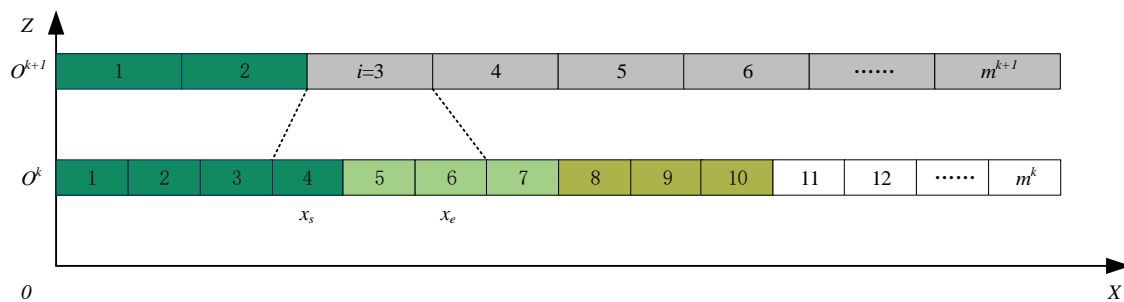

Figure 3. Operating strip state updating.

In general, to update the current strip state, the starting ID (x_s^k) and ending ID (x_e^k) of the prior operating strips should be determined. The two IDs and the total number (n) of related strips can be calculated by Formula (1). The aim of Formula (1) is to find all the strips of O^k that are related to O^{k+1}; this is equivalent to finding the IDs of the far-left strip and the far-right strip of O^k, which are related to O^{k+1}.

$$\begin{cases} x_s^k = int(\frac{w^{k+1}\times(i-1)}{w^k}) + 1 \\ x_e^k = int(\frac{w^{k+1}\times i}{w^k}) + 1 \\ n = x_e^k - x_s^k + 1 \end{cases} \qquad (1)$$

With the above parameters, the average state value of the objective operating strip can be calculated by Formula (2), where $\overline{V}$ is the mean of the state values of all the related strips of O^k:

$$\overline{V} = \sum_{j=x_s^k}^{x_e^k} V_j^k \div n \qquad (2)$$

Since 3 is the maximum value of the operating strip state, when $\overline{V} = 3$, that means that the state values of all the related strips are 3 (*done*). This is the condition required to change the strip state of the later operation (from *unready* to *ready*, or from 0 to 1), and the state of S_i^{k+1} can then be updated from *unready* to *ready*.

2.3. Sequential Operations Connection

2.3.1. Field Partitioning

For single-operation tasks, the field can be divided into partitions (sub-fields) according to the number of machinery units, and no optimization is needed within a partition. For multi-operation tasks, a partitioning strategy is also used for the flow-shop working mode. For instance, for the first operation task, all of the machines should work together from one side of the field to the other side, and each machine should occupy several operating strips as a temporary partition (Figure 4). When the machines finish a temporary partition, they move to the next temporary partition. Therefore, in each temporary partition, there will be no collisions between the multiple machinery units.
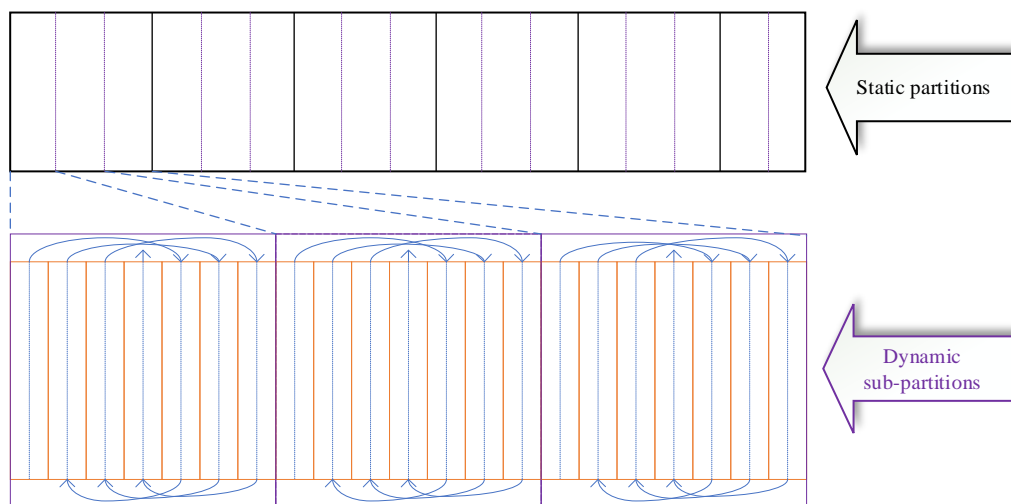

Figure 4. Dynamic partitioning for several sets of machinery units.

In practical work, the partitions for each operation are related to the number of machines, which is different in each operation. Different operations also have different work widths; thus, the partitioning for each operation is dynamic. Figure 4 shows an example of a fieldwork pattern with three sets of agriculture machinery. The entire field area is divided into six static partitions, and each partition is divided into three dynamic sub-partitions (excluding the last partition on the right). Each sub-partition is allocated to a single machine unit.

This study adopted the "first turn skip pattern" (FSP) [28] as the fieldwork pattern, which provides the highest savings in terms of in-field traversing distance among the most common pre-defined fieldwork patterns [29].

2.3.2. Operations Scheduling

The next step of the method includes the calculation of the optimal waiting time between two adjacent operations with a final goal to reduce the overall operating time. As shown in Figure 5, T^5 is the overall operating time of the fleet, and for its minimization, it is necessary to shorten the waiting times (T^k_{wait}) between any two adjacent operations.
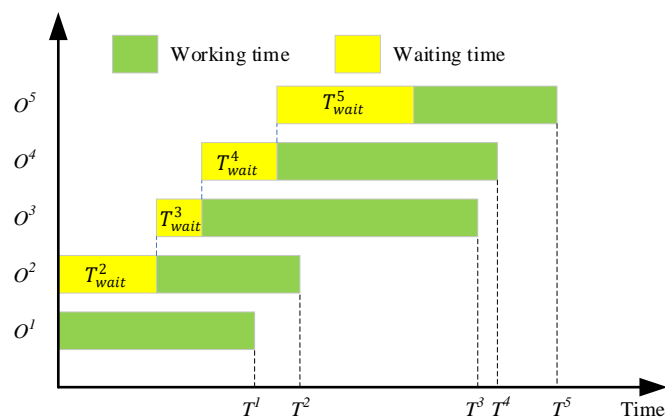
Figure 5. Waiting time and working time of each operation.

To calculate the waiting time, a new parameter, c^k, was defined to represent the operating capacity per unit time of the kth operation:

$$c^k = n^k \cdot v^k \cdot w^k \tag{3}$$

where n^k, v^k, and w^k are the number, velocity, and width, respectively, of the machinery unit of the kth operation.

Figure 6 shows the optimal waiting time between two adjacent operations. In Figure 6, each color represents an operation, and the horizontal line segment represents the duration within a single static partition. The left oblique line and the right oblique line go through the starting points and ending points of the horizontal line segments. To avoid collisions during operation, the left oblique line of the current operation should not intersect the right oblique line of the previous operation. Therefore, T^k_{wait} is determined by the relative relationship of the two slopes of the above two lines and can be calculated using Formula (4):

$$T^k_{wait} = \begin{cases} T^{k-1}_B, & c^{k-1} > c^k \\ \left(\frac{A}{c^{k-1}}\right) - \left(\frac{A}{c^k}\right) - T^{k-1}_B, & c^{k-1} < c^k \end{cases} \tag{4}$$

where T^i_B is the duration of a single static partition.

In fact, c^k is the slope of the left or right oblique line of the kth operation.

Due to each operation needing to cover the same area of the field, the total duration (T^k_{work}) of the kth operation is calculated as

$$T^k_{work} = A / c^k \tag{5}$$

where A is the area of the field.

Finally, the total operating duration of the kth operation is then calculated by

$$T^k = \sum_{2}^{k-1} T_{wait}^{k-1} + T_{work}^k.$$
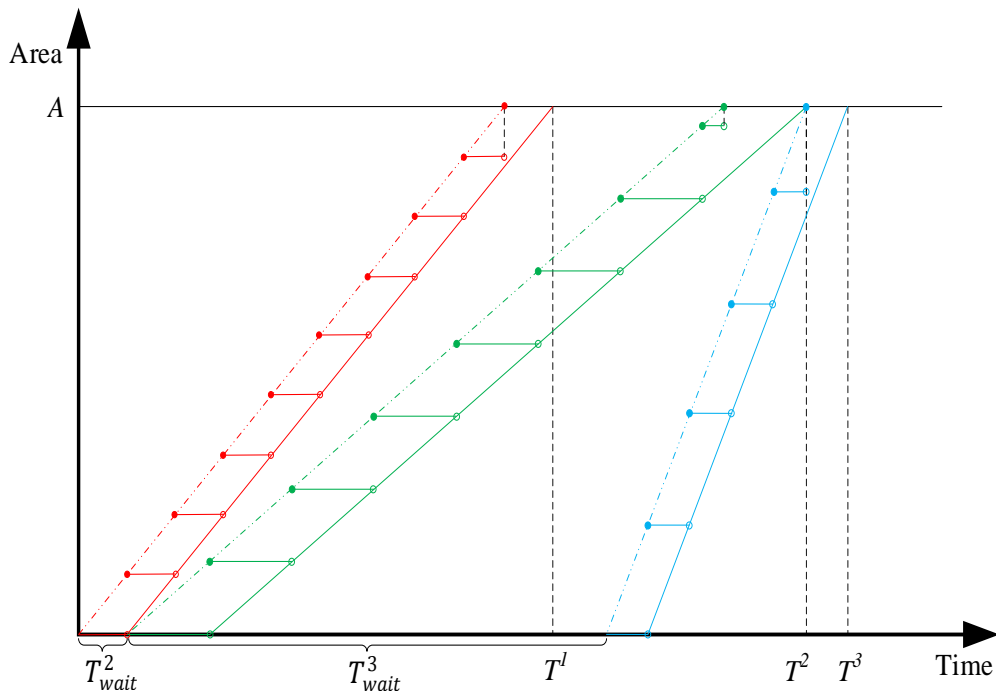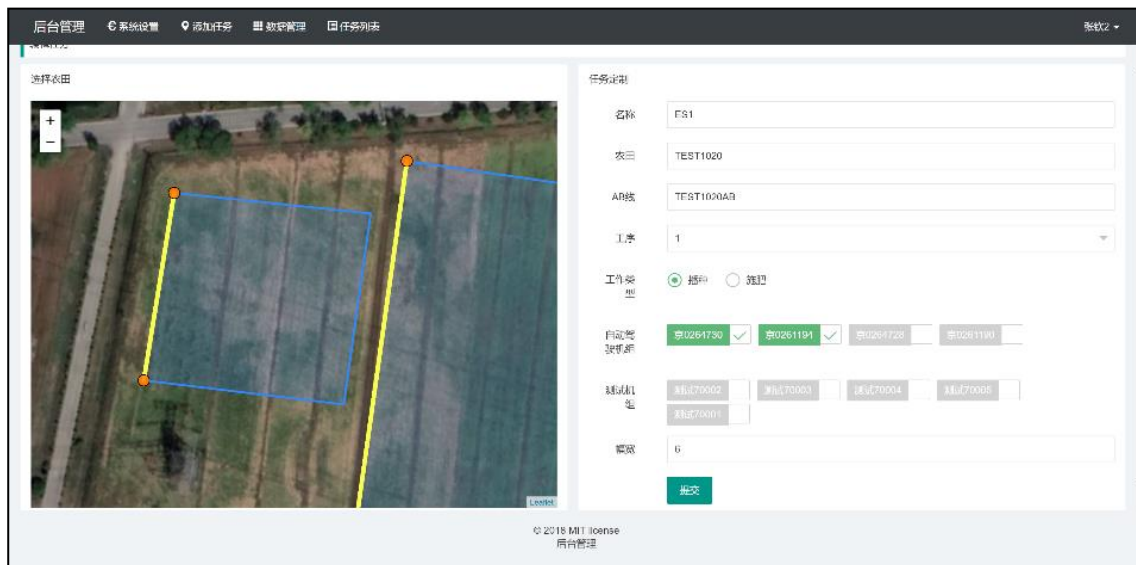
(6)

Figure 6. The relationship between the waiting time and the operating capacity per unit time.
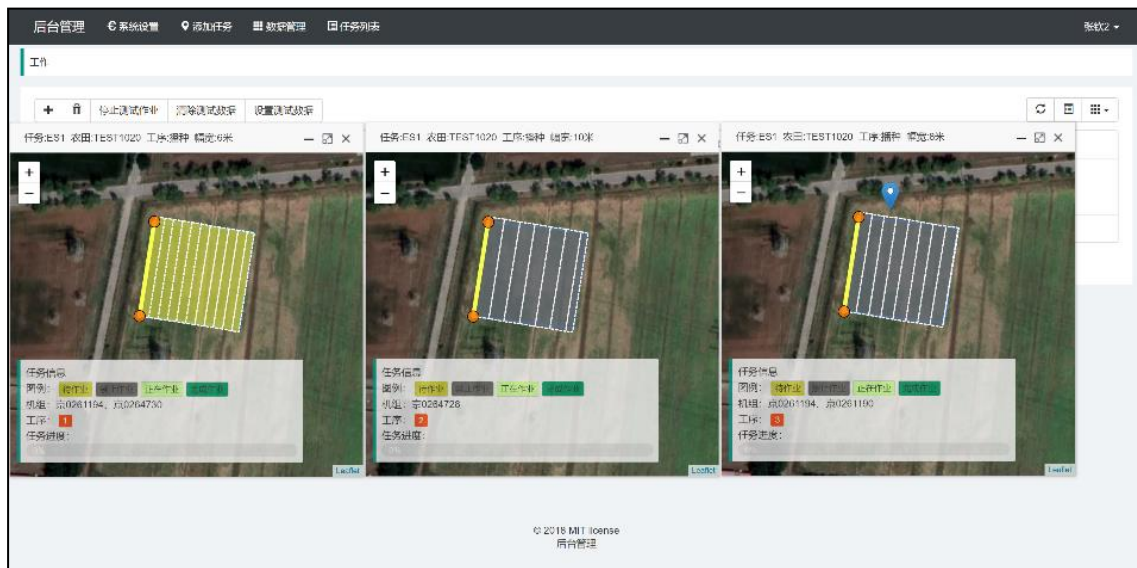
2.4. Collaborative Operating System

As shown in Figure 7, the cloud-based coordination system has functionalities to manage the field boundaries, operation tasks, agricultural implements, guidance lines, etc. In this study, only A-B guidance lines were used.

(**a**)

Figure 7. *Cont.*

(**b**)

Figure 7. Part of the software interface of the auto-steering-based collaborative operating system for fleet management (FMCOS): (**a**) Software interface for farmland management and (**b**) Software interface for three-operation coordination.

The more important function of the collaborative operating system is to realize a flow-shop working mode with multiple operations and multiple machinery units. Figure 8 shows the module structure of the collaborative operating system, in which the collaborative module for operations can coordinate all the agricultural machinery within the same operation, and the collaborative module for the fleet can coordinate the agricultural machinery between two sequential operations. Obviously, the fleet has $(N-1)$-many pairs of combinations of adjacent operations.
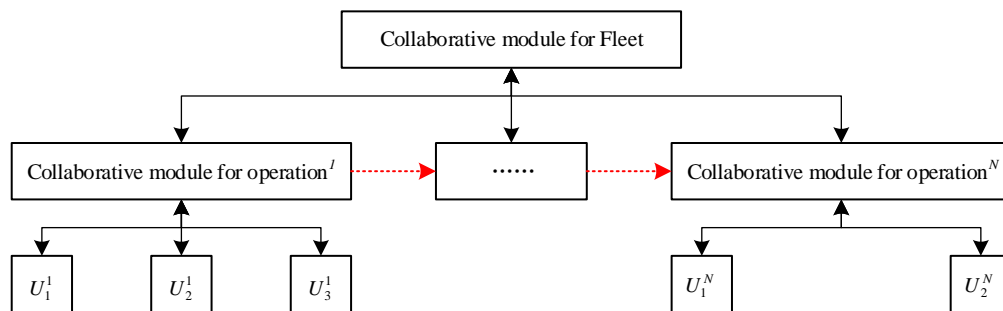

Figure 8. Module structure of the collaborative operating system for fleet management.

2.5. Auto-Steering System

As shown in Figure 9, a hydraulic auto-steering system (HuaCe NX200) was used for testing this system. This auto system has the functionality of sending GNSS positions to and downloading guidance lines from the cloud server. Dual-frequency differential signals with an accuracy of ±1.0 cm from the RTK-GNSS base station located at the experimental field were used.

However, NX200 is a commercial product and cannot be modified. For instance, the guidance line (A-B line) cannot be replaced by the operating strip, and the states also cannot be differentiated with colors. Therefore, a T715C tablet (see Figure 9) with an embedded GPS receiver of a single frequency was used, and an Android-based application was developed for better display. The application can download the position data of the NX200 on the same tractor from the server to replace the T715C's own position data to solve the problem of low accuracy. Although the data transform caused some 2 s

delay, the application was able to improve the positioning accuracy of the T715C to ±1.0 cm, which means that the tablet had the same accuracy as the NX200 auto-steering system.

Figure 9. The NX200 auto-steering system for a John Deere 1204 tractor and the Android-based application for a T715C tablet.

3. Experiments

3.1. Simulation Experiments

A simulation model (Matlab 2018a (MathWorks®)) was developed to validate the state-updating algorithm. The modules of the simulation program are shown in Figure 10. In the model, a strip was simulated as the "pushbutton" of UIControl (see Figure 11), while the operation IDs, the strip IDs, and the values of states were stored in the "UserData" field of the control as a three-dimensional array.
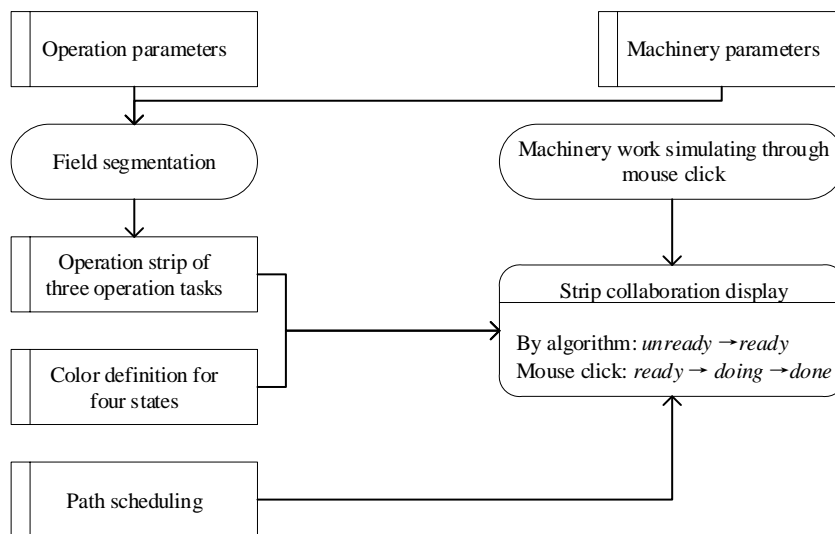

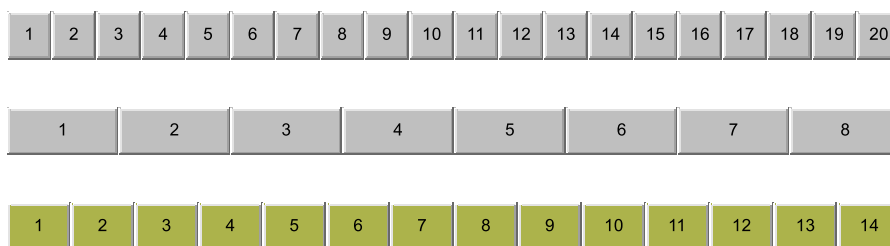**Figure 10.** Modules of the simulation model of the collaborative operation.

Figure 11. Simulation of the state-updating algorithm using Matlab (MathWorks®).

As shown in Figure 12, in the same operation, the MouseClick command was used to change the state from *ready* to *doing* (simulating the machinery operating on the strip), or from *doing* to *done* (simulating the machinery finishing and leaving the operating strip). When the state of *done* appears, the field operation subsequent to the current field operation will check its *unready* strips to determine the state changes by the algorithm.
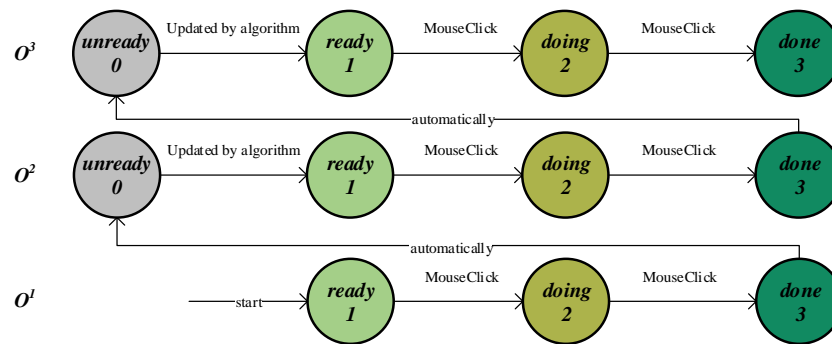

Figure 12. Deterministic finite state automata for the simulation of multi-operation collaboration.

3.2. Field Experiments

Three experimental setups (see Table 2) were designed to test the FMCOS. These setups, ES1, ES2, and ES3, were designed to test the collaboration functions and performance with one operation, two operations, and three operations, respectively. Three sets of NX200 were mounted on three JD1204 tractors (denoted A, B, and C). The superscript of the tractors' symbols (e.g., A^1) represents the sequence of operations.

Table 2. Experimental scheme of FMCOS.

Experiment Scheme	Operating Tasks	Machinery Arrangement *	Width of Strip/m
ES1	One operation	A^1 & B^1	6
ES2	Two operations	A^1 & B^1 VS C^2	6, 8
ES3	Three operations	A^1 VS B^2 VS C^3	8, 5, 6

* In ES1, tractors A and B worked in the first operation. In ES2, tractors A and B worked in the first operation, and then tractor C worked in the second operation. In ES3, operations 1, 2, and 3 were worked by tractors A, B, and C, respectively.

Figure 13 shows the RTK-GNSS base station, which was the experimental field (located at 40.237789° N, 116.581026° E) where all experiments took place, and Figure 14 shows the tractors used for tests.

Figure 13. Real-Time Kinematic–Global Navigation Satellite System (RTK-GNSS) base station for field experiments and the experimental field.
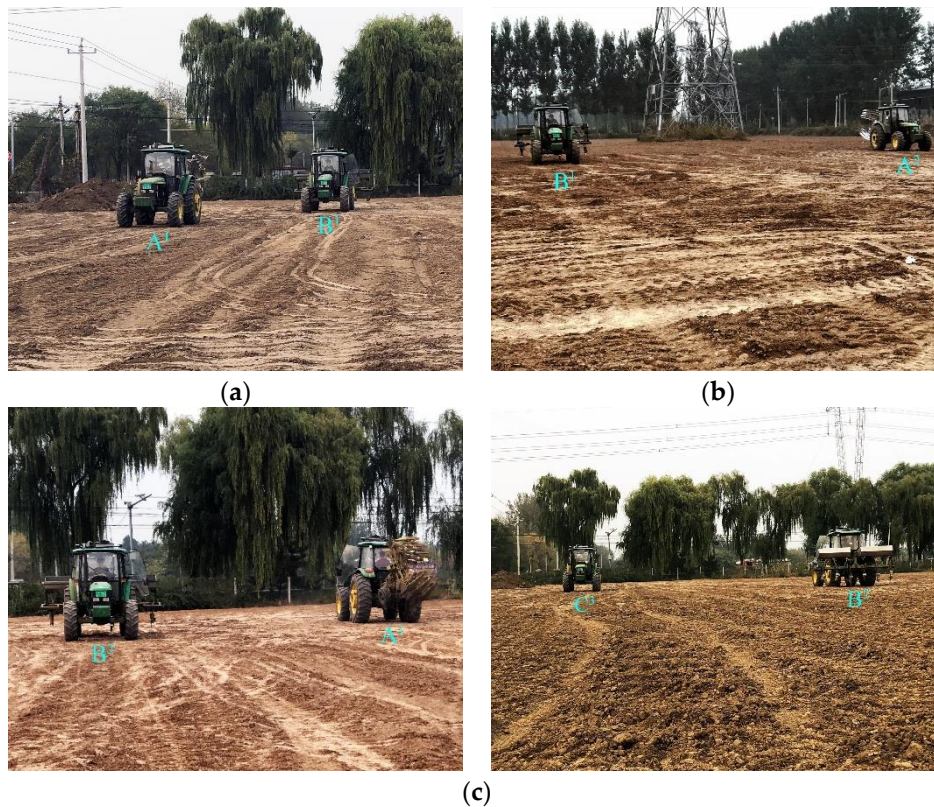
Figure 14. Field experiments scene of (**a**) ES1, (**b**) ES2, and (**c**) ES3.

3.3. Results and Discussion

3.3.1. Simulation Experiments

Figure 15a presents the scenario of two sets of agricultural machinery for the first operation (O^1)—which was worked from the middle of the field to both sides—and one set of agricultural machinery for the second operation (O^2). When clicking on the strip to change the state from *doing* to *ready*, such as $S_6^1 \sim S_{11}^1$, the state of strips related to the second operation, such as S_6^2, changes from *unready* to *ready*, which indicates the appropriate strip for the agricultural machinery of the second operation to work.
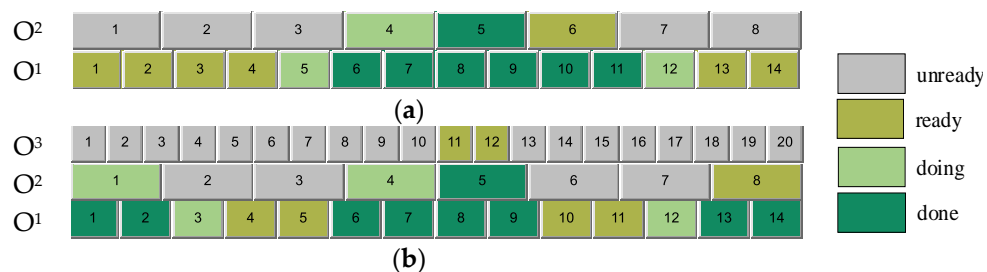

Figure 15. Simulation result in the flow-shop (**a**) Two-operation and (**b**) Three-operation working modes.

Figure 15b simulates the scene of three sets of machinery for the first operation working simultaneously. Two of them worked from the left or right side to the center of the field, while the other worked from the center to the left side of the field. Similarly, the two sets of machinery for the second operation were able to easily find the *ready* strip for working. The machinery for the third operation can choose the appropriate *ready* strips (such as S_{11}^3 and S_{12}^3), and so on.

The results of the above simulations show that the state-updating algorithm can not only well coordinate the state change for multiple machinery within the same operation, but it can also well coordinate two or more operations executed by a fleet of machines.

3.3.2. Field Experiments

Figure 16 depicts screen captures taken during the field experiments of the monitoring interfaces of the collaborative operating system for ES1, ES2, and ES3. ES1 demonstrated the collaborative action of two sets of agricultural machinery (A^1 and B^1) within the same machinery operation (O^1). Figure 16a shows the collaborative interface and Figure 17a shows the trajectory of ES1. From the above referenced pictures, we can see that the tractors could get the same base line from the server and generate the same guidance lines in the NX200 terminal. When the tractor entered or left a strip, the strips of the server and the guidance lines of the terminal changed the state. Therefore, operators were able to choose the *ready* strip to work. ES2 demonstrated the collaborative action of three sets of agricultural machinery (A^1, B^1, and A^2) within two machinery operations (O^1 and O^2). Figure 16b shows the collaborative interface, and Figure 17b shows the trajectory of ES2. The experiment showed that the strip state of the second machinery operation could be updated with the operating progress of the first machinery operation. Therefore, the operators within the same operation are able to choose the *ready* strip for work, and the operators of different operations are also notified in order to select the appropriate strips to enter.
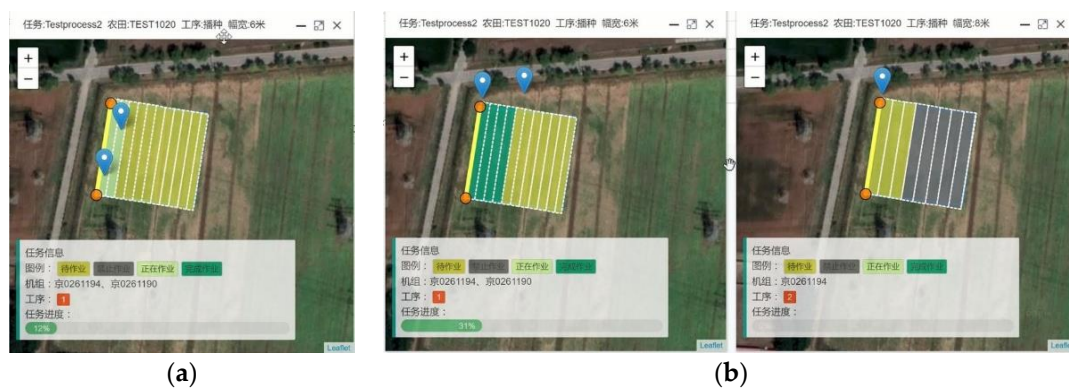

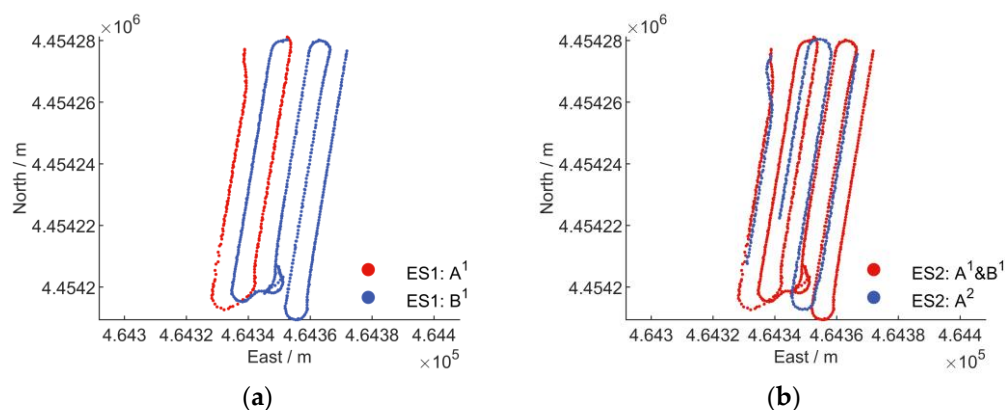**Figure 16.** Monitoring interface of (**a**) ES1 and (**b**) ES2.

Figure 17. Trajectory of (**a**) ES1 and (**b**) ES2.

In Figure 17b, in order to show the difference with the second machinery operation, the trajectories of the two tractors (A^1 and B^1) for the same machinery operation were fused.

As shown in Figure 18, ES3 demonstrated the collaborative action of three tractor operations. The experiment process verified that FMCOS could support the coordination of multiple operations. In fact,

three-operation collaboration consists of two two-operation collaborations. Therefore, the performance of ES2 has already proved the feasibility and usability of FMCOS.
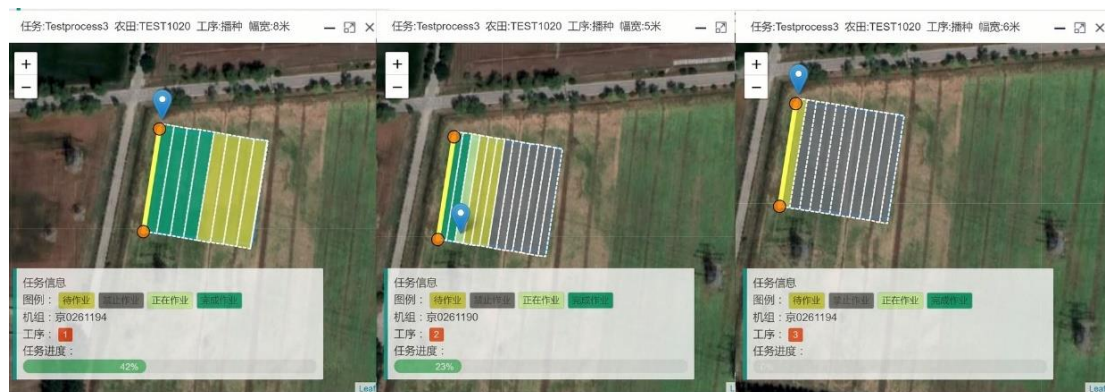

Figure 18. Monitoring interface of ES3.

The results of the three experiments are shown in Table 3. In ES1, tractors A and B worked in one operation. ES2 consisted of two operations, for which tractors A and B worked in the first operation, then tractor C worked in the second operation. ES3 consisted of three operations, worked by tractors A, B, and C, respectively. Table 3 provides the range of the efficiency improvement for different operating tasks, which have different operation combinations. The maximum efficiency improvement occurs in the experiment of ES3, while the minimum efficiency improvement occurs in the experiment of ES2. In general, the range of efficiency improvement was between 23.05% and 32.72%. It can be concluded that the FMCOS can significantly shorten the waiting time between adjacent operations so as to improve the whole work efficiency.

Table 3. Comparison between Non-FMCOS and FMCOS data in regard to efficiency improvement for ES1, ES2, and ES3.

Experiment	Machinery Arrangement		Operating Time [s]	Non-FMCOS [1] [s]	FMCOS [2] [s]	Efficiency Improvement [3] [%]
ES1	A^1&B^1	A^1 B^1	238 713	951	713	25.03
ES2	A^1&B^1 C^2		713 558	1271	978	23.05
ES3	A^1 B^2 C^3		485 685 563	1733	1166	32.72

[1] The sum of operating time; [2] Time from the start to the end of the experiment; [3] $\frac{[(\text{Non}-\text{FMCOS})-(\text{FMCOS})]}{\text{Non}-\text{FMCOS}} \times 100\%$.

4. Conclusions

A cloud-based collaborative operating system for fleet management (FMCOS) was developed by utilizing a flow-shop working mode and by implementing auto-steering systems. The core technology of the system lies in the state-updating algorithm for the operating strips between adjacent machinery operations and the strategy for adjacent operation connection. The system defined the strip states with four classes: *unready*, *ready*, *doing*, and *done*. The FMCOS was developed after testing the algorithm in a simulation model developed in Matlab (MathWorks$^{®}$). Three field experiments were carried out, namely, one-operation, two-operation, and three-operation experiments, to verify the functionality and performance of FMCOS. The assessment of the performance of FMCOS showed efficiency improvement up to 25.03% and 32.72% when compared with non-FMCOS. The results of the simulation and actual

experiments showed that FMCOS can support in-field fleet management, which can improve the performance of machines in a fleet operating in the same field.

References

1. Bsaybes, S.; Quilliot, A.; Wagler, A.K. Fleet management for autonomous vehicles: online PDP under special constraints. *RAIRO Op. Res.* **2019**, *53*, 1007–1031. [CrossRef]
2. Wang, J.; Zhu, Y.; Chen, Z.; Yang, L.; Wu, C. Auto-steering based precise coordination method for in-field multi-operation of farm machinery. *Int. J. Agric. Biol. Eng.* **2018**, *11*, 174–181. [CrossRef]
3. Srensen, C.G.; Bochtis, D.D. Conceptual Model of Fleet Management in Agriculture. *Biosyst. Eng.* **2010**, *105*, 41–50. [CrossRef]
4. Wu, C.C.; Zhou, L.; Wang, J.; Cai, Y.P. Smartphone based precise monitoring method for farm operation. *Int. J. Agric. Biol. Eng.* **2016**, *9*, 111–121. [CrossRef]
5. Noguchi, N.; Will, J.; Reid, J.; Zhang, Q. Development of a master-slave robot system for farm operations. *Comput. Electron. Agric.* **2004**, *44*, 1–19. [CrossRef]
6. Haifeng, L.; Chaoyi, Z. Design and Simulation of a Master-slave Robot for Apple harvest. *Adv. Mater. Res.* **2013**, *816*, 1024–1027.
7. Zhang, X.; Geimer, M.; Noack, P.O.; Grandl, L. Development of an intelligent master-slave system between agricultural vehicles. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, San Diego, CA, USA, 24–24 June 2010; pp. 250–255.
8. Zhang, C.; Noguchi, N.; Yang, L. Leader-follower system using two robot tractors to improve work efficiency. *Comput. Electron. Agric.* **2016**, *121*, 269–281. [CrossRef]
9. Zhang, C.; Noguchi, N. Development of Leader-follower System for Field Work. In Proceedings of the 2015 IEEE/SICE International Symposium on System Integration (SII), Nagoya, Japan, 11–13 December 2015; pp. 364–368.
10. Bai, X.; Wang, Z.; Hu, J.; Gao, L.; Xiong, F. Harvester Group Corporative Navigation Method Based on Leader-Follower Structure. *Trans. Chin. Soc. Agric. Mach.* **2017**, *48*, 14–21. [CrossRef]
11. Zhou, K.; Jensen, A.L.; Bochtis, D.D.; Sorensen, C.G. Simulation model for the sequential in-field machinery operations in a potato production system. *Comput. Electron. Agric.* **2015**, *116*, 173–186. [CrossRef]
12. Rodias, E.; Berruto, R.; Busato, P.; Bochtis, D.; Sorensen, C.G.; Zhou, K. Energy Savings from Optimised In-Field Route Planning for Agricultural Machinery. *Sustainability* **2017**, *9*, 1956. [CrossRef]
13. Hameed, I.A.; Bochtis, D.; Sorensen, C.A. An Optimized Field Coverage Planning Approach for Navigation of Agricultural Robots in Fields Involving Obstacle Areas. *Int. J. Adv. Robot. Syst.* **2013**, *10*, 231. [CrossRef]
14. Jensen, M.A.F.; Bochtis, D.; Sørensen, C.G.; Blas, M.R.; Lykkegaard, K.L. In-field and inter-field path planning for agricultural transport units. *Comput. Ind. Eng.* **2012**, *63*, 1054–1061. [CrossRef]
15. Han, X.Z.; Kim, H.J.; Kim, J.Y.; Yi, S.Y.; Moon, H.C.; Kim, J.H.; Kim, Y.J. Path-tracking simulation and field tests for an auto-guidance tillage tractor for a paddy field. *Comput. Electron. Agric.* **2015**, *112*, 161–171. [CrossRef]
16. Bochtis, D.D.; Dogoulis, P.; Busato, P.; Sorensen, C.G.; Berruto, R.; Gemtos, T. A flow-shop problem formulation of biomass handling operations scheduling. *Comput. Electron. Agric.* **2013**, *91*, 49–56. [CrossRef]
17. Senlin, G.; Nakamura, M.; Shikanai, T.; Okazaki, T. Hybrid Petri Nets Modeling for Farm Work Flow. *Comput. Electron. Agric.* **2008**, *62*, 149–158. [CrossRef]
18. Bai, X.; Hu, J.; Gao, L.; Li, T.; Liu, X. Self-tuning model control method for farm machine navigation. *Trans. Chin. Soc. Agric. Mach.* **2015**, *46*, 1–7. [CrossRef]

19. Sadik, A.R.; Urban, B. Flow shop scheduling problem and solution in cooperative roboticscase-study: One cobot in cooperation with one worker. *Future Internet* **2017**, *9*, 48. [CrossRef]

20. Wu, C.; Cai, Y.; Hu, B.; Wang, J. Classification and evaluation of uncertain influence factors for farm machinery service. *Int. J. Agric. Biol. Eng.* **2017**, *10*, 164–174. [CrossRef]

21. Bechar, A.; Vigneault, C. Agricultural robots for field operations. Part 2: Operations and systems. *Biosyst. Eng.* **2017**, *153*, 110–128. [CrossRef]

22. des Dorides, C. *GNSS Market Report*; European GNSS Agency: Luxembourg, 2017.

23. Easterly, D.R.; Adamchuk, V.I.; Kocher, M.F.; Hoy, R.M. Using a vision sensor system for performance testing of satellite-based tractor auto-guidance. *Comput. Electron. Agric.* **2010**, *72*, 107–118. [CrossRef]

24. Cao, R.; Li, S.; Wei, S.; Ji, Y.; Zhang, M.; Li, H. Remote Monitoring Platform for Multi-machine Cooperation Based on Web-GIS. *Trans. Chin. Soc. Agric. Mach.* **2017**, *48*, 52–57. [CrossRef]

25. Deere, J. John Deere Machine Sync. Available online: https://www.deere.com/en_INT/products/equipment/agricultural_management_solutions/guidance_and_machine_control/machine_sync/machine_sync.page? (accessed on 24 November 2014).

26. Jensen, M.F.; Bochtis, D.; Sorensen, C.G. Coverage planning for capacitated field operations, part II: Optimisation. *Biosyst. Eng.* **2015**, *139*, 149–164. [CrossRef]

27. Wu, C.; Yuan, Y.; Han, Y. *Application of Beidou in Agricultural Production Process*; Song, M., Ed.; Publishing House of Electronics Industry: Beijing, China, 2016.

28. Hunt, D. *Farm Power and Machinery Management*, 11th ed.; Waveland Press: Lake County, IL, USA, 2016; pp. 9–17.

29. Zhou, K.; Jensen, A.L.; Bochtis, D.D.; Sorensen, C.G. Quantifying the benefits of alternative fieldwork patterns in a potato cultivation system. *Comput. Electron. Agric.* **2015**, *119*, 228–240. [CrossRef]