Dynamic immune cooperative scheduling of agricultural machineries

Xiaoyan Liu^{1,2} · Xinmeng Zhu^{1,2} · Kuangrong Hao^{1,2}

Received: 9 April 2021 / Accepted: 13 June 2021 / Published online: 25 July 2021
© The Author(s) 2021

Abstract

Considering the low flexibility and efficiency of the scheduling problem, an improved multi-objective immune algorithm with non-dominated neighbor-based selection and Tabu search (NNITSA) is proposed. A novel Tabu search algorithm (TSA)-based operator is introduced in both the local search and mutation stage, which improves the climbing performance of the NNITSA. Special local search strategies can prevent the algorithm from being caught in the optimal solution. In addition, considering the time costs of the TSA, an adapted mutation strategy is proposed to operate the TSA mutation according to the scale of Pareto solutions. Random mutations may be applied to other conditions. Then, a robust evaluation is adopted to choose an appropriate solution from the obtained Pareto solutions set. NNITSA is used to solve the problems of static partitioning optimization and dynamic cross-regional co-operative scheduling of agricultural machinery. The simulation results show that NNITSA outperforms the other two algorithms, NNIA and NSGA-II. The performance indicator C-metric also shows significant improvements in the efficiency of optimizing search.

Keywords Scheduling of agricultural machineries · Multi-objective immune algorithm · Non-dominated neighbor-based selection · Tabu search algorithm · Self-adapted mutation operator

Introduction

The multi-objective optimization algorithm (MOA) has been a hot issue in recent years because of its balance on each aspects of the problem, which mainly consists of the particle swarm optimization (PSO) algorithms [1–3], the immune clonal algorithms (ICA) [4], the evolutionary algorithms (EA) [5–7], the differential evolution (DE) algorithms [8,9], and other hybrid heuristic algorithms [10–14]. These biological heuristics algorithms obtain better solutions through a continuous iterative process, in which a set of search rules are proposed. To obtain a satisfactory solution, it is necessary to constantly absorb new information and improve

the performance of initial solutions through some type of search strategy during the entire iterative process [15–19]. Besides the heuristic mechanism, when it comes to solving multi-objective problems, there are also other special strategies. In 1994, Srinivas et al. presented the non-dominated sorting genetic algorithm (NSGA) [20], the non-dominant layered approach of which enabled good individuals to have greater opportunities to pass on to the next generation the ability to produce a new genetic model of the same species. In 2002, the NSGA-II based on NSGA was proposed, which used the elitism strategy [21]. In addition, the crowded distance sort was also an important aspect in obtaining a set of nice solutions with distribution. Algorithms which also used elitism strategies were SPEA and its improved version SPEA-II [22,23]. In 2008, Gong et al. proposed a multi-objective immune algorithm with non-dominant neighbor-based selection (NNIA) [24]. The NNIA simulated the phenomenon of symbiosis with multifarious antibodies and activation of minority antibodies in immune responses. However, each of the above related algorithms can only focus on the global search and the local search of the solutions, respectively, without performing deep local search for a partial solution simultaneously. In this paper, we propose to design an improved multi-objective immune

✉ Kuangrong Hao
krhao@dhu.edu.cn
Xiaoyan Liu
liuxy@dhu.edu.cn
Xinmeng Zhu
247298335@qq.com

¹ College of Information Science and Technology, Donghua University, Shanghai 201620, People's Republic of China

² The Engineering Research Center of Digitized Textile and Apparel Technology, Ministry of Education, Shanghai, People's Republic of China

algorithm which combines the advantages of the NNIA and the Tabu search algorithm (TSA) (NNITSA) to improve the climbing performance of the algorithm. The NNITSA is used to solve the static partitioning optimization and dynamic cross-regional co-operative scheduling problems of agricultural machinery. Agricultural machinery services are usually performed through fixed political districts. Under the existing management system of agricultural machinery operations, machinery services are overall designed by professional agricultural specialists and organized implementations such as ploughing, planting, harvesting and field management. NSGA-II alleviates all the above three difficulties. Specifically, a fast non-dominated sorting approach with O(MN²) computational complexity is presented. Also, a selection operator is presented that creates a mating pool by combining the parent and offspring populations and selecting the best solutions [25,26].

This involves multiple field units of different districts that need to be managed uniformly and assigned evenly. However, the political distributions of fields are not balanced, which in turn leads to poor efficiency and a high distribution cost. Therefore, it is necessary to break the fixed partitioning, while optimizing the distribution of fields to improve the efficiency of agricultural machinery and services.

Districting problem is defined as re-dividing geographical districts into new ones to conform to the current system. It is a kind of typical multi-objective optimization problem (MOP) which involves several criteria [27–30]. The effect of partitioning is said to be even greater than the effect of partitioning other partitions. So, we aim to find all the best solutions that constitute a balanced set. There is no ideal solution to this set, but a better solution can be found in many different of conditions. In addition to the actual operation of agricultural production, several problems with the operation of fields affect the efficiency of field service and completion times. Thus, the scheduling system must to be adapted to give the rational scheduling decision, and to performs cooperative scheduling to maintain the overall of the operating system while maintaining the overall benefits of the scheduler and operating system in the same manner as the operating system.

There are a few of researches on districting and dynamic cooperative scheduling problems so far, mainly focusing on the evolutionary algorithm (EA) [31–33], co-evolutionary algorithm (CEA) [34], goal programming [35], and the Tabu search algorithm (TSA) [36], which can basically solve the corresponding problems. What is more, there have been some researches on scheduling of agricultural machineries [37–40]. Based on most of the algorithms mentioned above, multi-objective optimization algorithm (MOA) plays an important role in solving districting problems. What is more, we already have related work in clonal selection algorithm [41], robust optimization of multi-objective approach [42], dynamic immune clustering for event-driven

wireless sensor networks [43] and immune system-inspired rescheduling algorithm [44].

The main contributions of this paper are as follows:

- (1) A TSA operator is designed to perform parallel mutations with a specific mutation to boost the performance of the algorithm.
- (2) A local search strategy was introduced to improve the performance of the immune algorithm in local search results, and to improve the accuracy of local search results.
- (3) A robust evaluation mechanism is introduced, with which, the proposed algorithm solves the dynamic cross-regional cooperative scheduling problem of agricultural machinery.

The remaining content of this paper is organized as follows. In “The improved multi-objective immune algorithm combining NNIA and TSA”, the proposed NNITSA is developed based on NNIA and TSA operators, respectively. In “NNITSA for partitioning optimization of agricultural machinery”, the NNITSA is presented to solve the static partitioning optimization problem and the dynamic cross-regional cooperative problem of agricultural machinery. What is more, the performance of NNITSA is analyzed and proved. In “Conclusions”, a summary of the work and the perspectives considered are given.

The improved multi-objective immune algorithm combining NNIA and TSA

Using the strategy of selecting the no-dominant neighborhood individuals, with the proportional cloning strategy according to the crowding distance which strengthens the search of sparse areas of the current Pareto frontier surface, NNIA obtains good solutions for cross-over and mutation processing, so as to ensure that high quality solutions can be generated. Thus, NNIA can obtain a uniformly distributed Pareto optimal solution. Moreover, compared with the existing multi-objective evolutionary optimization methods, NNIA does not rank the non-dominated solution, which reduces the computational complexity. The implementation procedure of NNIA is depicted as follows [24].

Step 1: Initialization. At iteration $i = 1$, generate an initial antibody population POP with size NM .

Step 2: Update dominant population. Select dominant antibodies from POP.

Step 3: Termination. When reaching the stopping criteria (iteration $i = IMAX$), stop further iteration and output the dominant population. Otherwise, go on.

Step 4: No-dominated neighbor-based selection: update active population. Calculate the crowding distance of all indi-

viduals in NEPOP, sort NEPOP in the descending order of crowding distance, then extract the first NM (maximum size of active population) antibodies to constitute the active population in the genetic stage denoted by APOP.

Step 5: Genetic operation. Apply proportional cloning to the active population. The probability of which is calculated related to the crowding distance. Apply crossover and mutation to the cloned population. Denote the population as genetic population.

Step 6: Get the result antibody population by combining genetic population and dominated population; go to Step 2.

However, the result of non-dominated neighbor-based selection is implemented only in clonal selection stage. Due to the randomness of mutation and crossover, fewer searches conducted within an individual's neighborhood while expanding the global search scope. So it is important to adopt the neighborhood search strategy, which can not only ensure the diversity of the population, but also speeds up the search process and improves the efficiency of the search.

This neighborhood solution is a key point for the TSA. To get into the optimal solution set, TSA uses a journalist to record the several movements conducted during the several iterations of the program. Any movement in the Tabu list is forbidden in the current iteration. Thus, the same solutions will not appear twice in the latest iteration, which can prevent the evolution of time-consuming circulation and is helpful in finding the local optimum. What is more, TSA also has aspiration criterion in order not to miss the movement of the best solution; the TSA also has aspiration criteria to ensure that it does not miss out on the best solution. The procedure of Tabu search strategy used in this paper is as shown in Algorithm 1.

As the Tabu search strategy uses two kinds of movements: the migration and swap of units, a boundary algorithm is needed.

- (1) Given two unit sets A and B of two districts.
- (2) Calculate the distances between each two units belonging to A and B, denoted by BD. Sort BD in the ascending order of distances.
- (3) Select the first k pairs of units to constitute the boundary unit sets of A and B.

Considering the high quality of local search by the TSA, we use the TSA operator to enhance the local searching capability in the procedure of iterations. In the mutation stage of NNIA, we set the TSA operator as a mutation operator and propose NNITSA. The overall procedures of NNITSA are illustrated in Fig. 1.

Initialize all the units: Load the input parameters, as well as the necessary parameters of NNITSA, such as the dominant population size NM , active population size NA , cloning population size CS , the current iteration number G , the maxi-

Algorithm 1 Pseudocode of multi-objective TSA

```

1: Start with the initial solution  $x$ 
2: Let  $BSF = x$ 
3: While not reaching the stopping criteria (iteration  $i = IMAX$ ) do
4:   Generate neighbor solutions of  $BSF$ 
5:   Movements
6:   Divide all the neighbor solutions into 3 subsets:
7:     1) one is equal to the current point Pareto ( $\Omega_{equal}$ ), which cannot
       decide dominating relations;
8:     2) the second is superior to the current point ( $\Omega_{dominating}$ );
9:     3) the third is inferior to the current point ( $\Omega_{dominated}$ ).
10:  If  $size(\Omega_{dominating}) == 1$ 
11:     $s = \Omega_{dominating}$ 
12:  Else if  $size(\Omega_{dominating}) > 1$ 
13:    Obtain the non-inferior solution set  $\Omega_{dominating}'$  of
        $\Omega_{dominating}$ , then randomly pick up a point  $s$  which is not in
       Tabu list from  $\Omega_{dominating}'$ .
14:  Add the rest of points to the intensification memory list (IM)
15:  Else if  $\Omega_{dominating}$  is null and  $\Omega_{pareto-equivalence}$  is not
16:    Obtain the non-inferior solution set  $\Omega_{pareto-equivalence}'$  of
        $\Omega_{pareto-equivalence}$ , and randomly pick up a point  $s$  which is not
       in the Tabu list from  $\Omega_{pareto-equivalence}'$ .
17:  Else if IM is not null
18:    Randomly pick up a point  $s$  as the next current point.
19:  Else
20:    Break
21:  Let  $s$  be the next current point,  $BSF = s$ 
22:  Update the tabu list
23: End while

```

imum number of iterations $GMAX$, the crossover probability pc and the mutation probability pm .

Initialize the antibodies: At iteration $i=1$, an initial antibody population POP with size NM is generated.

Update dominant population: Calculate the objective function value of all individuals in POP. Update dominant antibodies to form the temporary dominant population, denoted by MEPOP.

No-dominated neighbor-based selection: Calculate the crowding distance of all individuals in MEPOP and sort MEPOP in the descending order of crowding distance, then extract the first NA antibodies as the active population.

In most of the multi-objective algorithms, the dominant antibodies are ranked according to crowding distance, which represents how much they contribute to the diversity of objective function values [45].

The definition of crowding distance of is given by:

$$Cd(d, D) = \sum_{i=1}^k \frac{Cd_i(d, D)}{f_i^{\max} - f_i^{\min}}, \quad (1)$$

$$Cd_i(d, D) = \begin{cases} \infty, & \text{if } f_i(d) = \min\{f_i(d') | d' \in D\} \\ & \text{or } f_i(d) = \max\{f_i(d') | d' \in D\} \\ \text{otherwise,} & \min_{d'' \in D : f_i(d'') < f_i(d) < f_i(d')} |f_i(d') - f_i(d'')| \end{cases}, \quad (2)$$

Fig. 1 The overall procedures of NNITSA

where d and D are each dominant antibody and the corresponding population, k is the size of the population, $f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum values of the i th objective function, respectively.

The value of the crowding distance shows the density of the distribution of antibodies in one solution, based on which we can extract active population from dominant population.

Local search: Comparing the current active antibody population with those of the previous iteration, determine whether they are the same, if not, go to Step 6, else, implement local search.

Randomly choose one antibody in the current active antibody population, implement the TSA operator. Replace the chosen antibody with the result of neighbor search; go to Step 6.

Termination: When reaching the stopping criteria (iteration $G = G_{\max}$), stop further iteration. Otherwise, go on.

Immunization operations: Special immunization operations, including cross-over and mutation operations, are used to generate the subsequent population.

Proportional cloning: Applying proportional cloning to the selected active population MEPOP. The probability of which is calculated related to the crowding distance.

Select one antibody a_i from the active population to illustrate, a_i will be implemented proportional cloning following the equation below:

$$CS_i = CS \times \frac{Cd(a_i, \text{MEPOP})}{\sum_{i=1}^{|A|} Cd(a_j, \text{MEPOP})}, \quad (3)$$

where CS is the cloning population size and CS_i is calculated by the crowding distance of each active antibody in MEPOP.

Crossover: Denote the antibodies generated by cloning procedure described above as CPOP. Two antibodies are randomly selected from CPOP and MEPOP. Then select two gene bits within the range of antibody length n and cross all gene bits in range of those two. After NM cycles, the crossed population is obtained.

Mutation (TSA): We introduce a TSA operator to improve the climbing performance of NNIA, by taking the dominant local solution of the TSA as the mutated antibody. Only when the solutions found by the TSA are superior to the current antibodies, they will be taken as the antibodies mutated.

Combination: Get the result antibody population by combining immunization population and dominated population; go to Step 3.

The TSA operator is utilized in two aspects:

(1) **Mutation:** When the Pareto solutions of the algorithm keep the same for several iterations, TSA operator would replace traditional mutation to obtain a better solution instead of randomness mutation. The probability of TSA mutation changes by the increasing of scale of Pareto solutions. Under other circumstance, random mutation is operated to ensure the searching scope of the algorithm.

The adaptive parameter is defined as below:

$$c_a = \sqrt{|MEPOP|}, \quad (4)$$

where $|MEPOP|$ is the scale of MEPOP, when MEPOP remains the same for c_a iterations, we adopt TSA mutation.

(2) **Local search:** As long as the Pareto solutions changes, which means a better solution set is obtained, a local search strategy would be performed. Based on TSA operator, choosing one of the solution set, a better solution would be found. Then if the solutions did not change, the local search would not go deeper, so that the algorithm can be prevented from getting into the local best solution.

NNITSA for partitioning optimization of agricultural machinery

The problem of static partitioning optimization can be described as: there is a randomly distribution of n fields in a certain range, the required service of each field is different. The center of agricultural machinery is located in the center of the range. A redistrict scheme is needed to break the limits of historical partitioning. This is obviously a multi-objective optimization problem because of several conditions that need

to be considered for partitioning. We have identified the target of the problem as below.

The goal is to obtain a dominant set of the partitioning of the given field units. Two objectives are considered in this model.

A balanced partitioning scheme must ensure the optimization result of total path. It is assumed that all machines are managed uniformly by the agricultural machinery center and distributed to each district. The travel distance of each district is denoted as: starting from the agricultural machinery center, traversing all the fields of a district and returning to the center of the agricultural machinery center.

$$\begin{aligned} \text{Minimize } f_1 = L(D) &= \sum_{D_i \in D} L(D_i) \\ &= \sum_{D_i \in D} \sum_{u_j^i \in D_i} L(u_j^i), \end{aligned} \quad (5)$$

where D is the fields set of all the districts, u_j^i is the j -th field unit in D_i , L means the length of total path.

We wish that all the districts can complete their work at the same time, so it is necessary to keep the balance of the service load of each district. Thus, the other objective function is shown in Eq. (6).

$$\begin{aligned} \text{Minimize } f_2 &= \text{Std}(Q(D_i)) \\ &= \text{Std}_{D_i \in D}(Q(D_i)) \\ &= \text{Std}_{D_i \in D}(\sum Q(u_j^i)). \end{aligned} \quad (6)$$

This objective function is to minimize the variance of service time among all the districts, where Q means the service time. For this problem, there are some particular implementation issues below.

In order to make it convenient to manage and operate, each district should be relatively aggregated and less overlapping. For improving the efficiency of the algorithm, an initial partitioning algorithm is designed to generate the optimized initial solutions, which combines indivisible field units into feasible districts. The procedure of generating initial scheme is as shown in Algorithm 2.

When it comes to the calculation of the distance between any two fields, the actual traveling paths of the machineries are taken into account. The specific path of the agricultural machine from a field to another field is not likely to be a simple link between the central point of those two fields, so it is optimized to be the sum of distance of the two right-angled sides of the right-angled triangle corresponding to the line connecting the fields center, as shown in Fig. 2, where $d_{ij} = d_{AB} + d_{BC}$. Furthermore, the traveling sequence of the fields is also optimized to avoid the intersection of the connected graph.

Algorithm 2 Pseudocode of generating initial scheme

```

1: Start
2: Select a seed unit  $x$  for district  $D_1$  randomly
3: Select 3 adjacent basic units, from which, randomly select one unit for  $D_1$ 
4: While total service time  $Q$  within interval  $[a, b]$ 
5:   Calculate the center of  $D_1$  and generate the next 3 adjacent basic units of the center of  $D_1$ 
6:   Merge the basic units
7: End while
8: Generate the district  $D_2$  to  $D_{N-1}$  from the left units as step 2.3.4.5
9: Merge the left-over basic units as the last district  $D_N$ 
10: Obtain seed solution,  $s$ 

```

Fig. 2 Example of antibody representation

For this problem, we design two special movements. The neighborhood of a solution can be obtained by changing any unit in a district, so the movements in TSA operator are as shown in Algorithm 3.

Algorithm 3 Pseudocode of movements in TSA operator

```

1: For movement1
2:   Select border units of  $BSF$ 
3:   Move to another district
4: End for
5: For movement2
6:   Select border units of  $BSF$ 
7:   Swap two border units
8: End for

```

The description of the antibody population raises some issues which should be illustrated. We use a structure variable to represent a partitioning scheme. This structure variable has four data members: the first three members are arrays of fields number of the three districts, denoted as D_1, D_2, D_3 . Another member is a two-dimensional array which represents the fitness of two objective functions.

At the mutation and crossover stage, each antibody is encoded into a district vector $\mathbf{X} = (x^1, x^2, \dots, x^n)$, where n is the total number of fields to be partitioned. The value of each bit of $\mathbf{X}(x^i)$ is the number of district, which the field i is partitioned to, ranging from 1 to 3, represent D_1, D_2, D_3 , respectively (Fig. 3).

D1=[5 7 8 9 11 12 13 14 15 16]
D2=[2 3 4 6 10 20 24 25 27 29]
D3=[1 17 18 19 21 22 23 26 28 30]
Cost=[cost1, cost2]

Fig. 3 Example of antibody representation

Table 1 Example of solution encoding

i	1	2	3	4	5	6	7	8	9	10
x_i	3	2	2	2	1	2	1	1	1	2
i	11	12	13	14	15	16	17	18	19	20
x_i	1	1	1	1	1	1	3	3	3	2
i	21	22	23	24	25	26	27	28	29	30
x_i	3	3	3	2	2	3	2	3	2	3

Fig. 4 Illustration of immunization operations

What is more, to enhance the effect of crossover stage, we encode the partitioning scheme into a ternary sequence with the length of n , as show in Table 1.

The illustration of immunization operations is as shown in Fig. 4.

After balanced partitioning, choosing one relatively modest solution of “NNITSA for partitioning optimization of agricultural machinery” as shown in Fig. 5, and allocating unified task sequences for agricultural machines in all area. For example, given 12 machines and they are numbered from 1 to 12 where 1–4 belong to D1, 5–8 belong to D2 and 9–12 belong to D3. In Fig. 5, different colors of fields belong to different districts.

Assume that the machines are homogeneous, but some of them would not be able to continue providing services due to the failure. When a dynamic event occurs, a machinery breaks down and cannot continue to provide services. Other agricultural machines need to be scheduled to continue the work of the broken machine. What is more, in order to ensure the overall efficiency of three districts, the rest of the service may not be done by machineries belonging to the cor-

Fig. 5 Partitioning result of “NNITSA for partitioning optimization of agricultural machinery”

responding district only, which is the dynamic cross-regional co-operative scheduling problem.

A number of schedule parameters are required involving the number of available machineries and so on (Table 2).

In this experiment, the robustness of different schedule is analyzed. Trigger different amount of dynamic machineries at the same time. Dynamic machineries broken at various time are listed in Table 3. RoubstnessC1–RoubstnessC12 denote the experiments of different machines broken at various time, respectively.

The total travel path of all machines needs to be as small as possible. On the one hand, it is to reduce the energy consumption of the agricultural machinery, and on the other hand, the agricultural machinery passing through the fields will have an impact on the compactness of the soil [46].

$$\text{Minimize } f_1 = L(M)$$

$$= \sum_{m \in ML(m)} \sum_{k \in H} \sum_{j \neq k, j \in H} x_{jkm} l_{jk}. \quad (7)$$

We wish that all the districts can complete their work at the same time, so the expected end time of each district is ought to be as balanced as possible.

$$\text{Minimize } f_2 = \sigma(T(D)) = \sigma\left(\sum_{d_i \in D} T_m y_{mi}\right) = \sigma. \quad (8)$$

For dynamic cross-regional co-operative scheduling problem, there are some constraints as follows:

Constraint1: We assume that the speed of each machinery is the same.

Table 2 Schedule parameters and input parameters

Variable	Description
M	The set of available machines
M_m	Each available machine, $m \in \{1, 2, \dots, K, \dots, 12\}$
D	The set of existing districts
D_i	Each existing district, $i \in \{1, 2, 3\}$
H	The total number of fields
l_{jk}	The distance from field j to field k
x_{jkm}	1 if machine m travels from field j to field k , otherwise 0
y_{mi}	1 if machine m belongs to district d_i , otherwise 0
T_m	Service and travel time of machine m
v	Constant speed of machines (40 kilometers per hour)
z_{jm}	1 if machine m serves field j , otherwise 0
S_{vi}	Service which field i need, the corresponding value of the depot is 0
Mp_m	Service efficiency of machine m

Table 3 Dynamic machineries appeared at various time

No.	t	Dynamic machines appeared
RoubstnessC1	3	1
RoubstnessC2	3	1,2
RoubstnessC3	3	1,2,5
RoubstnessC4	3	1,2,5,6
RoubstnessC5	3	1,2,5,6,9
RoubstnessC6	3	1,2,5,6,9,10
RoubstnessC7	3	1
RoubstnessC8	3	1,2
RoubstnessC9	3	1,2,5
RoubstnessC10	3	1,2,5,6
RoubstnessC11	3	1,2,5,6,9
RoubstnessC12	3	1,2,5,6,9,10

Constraint2: Each field is served by one machine at one time.

Constraint3: It is allowed that a machine can leave the current field to serve another field before the service of the current field is completed and a field can be served by two or more machineries one after another.

Constraint4: The traveling sequence of each machine always begins at the depot, by way of several fields, and finally comes back to the depot.

Constraint5: If there is only one element in the to-be-served list of an agricultural machinery, it should stay in the field after finishing all the service waiting for the next instruction instead of returning to the machine center. Otherwise, only when the machine gets to the next field, it would perform other dynamic instructions.

In Algorithm 4, movement1 generates $H(H-1)/2$ neighboring solutions, where H is the number of fields to be served. While each of the movements 2 and 3 generates a maximum

of H solutions. These movements result in a maximum of $H(H-1)/2 + 2H$ solutions in the neighborhood of a solution. It is revealed that the scale of local search expands, so NNITSA spends more time on local search.

Algorithm 4 Pseudo-code of Movements in TSA Operator

```

1: For movement1
2:   Select two field units of  $BSF$ 
3:   Swap two field units
4: End for
5: For movement2
6:   Select one field unit of  $BSF$ 
7:   Increase the number of machine serving the field unit by one,
   unless in doing so would exceed the maximum machines available
8: End for
9: For movement3
10:  Select one field unit of  $BSF$ 
11:  Decrease the number of machine serving the field unit by one,
   unless in doing so would result in zero.
12: End for

```

Experiment results and analysis

Experiment results and analysis of static partitioning optimization

To fully reflect the performance of the proposed algorithm, all test instances are solved by NSGA-II, NNIA and NNITSA respectively on MATLAB 2016a platform. For all the algorithms, the maximum number of iterations is set as 300, the probability of crossover is set as 0.8, the probability of mutation is set as 0.3, the maximum size of population (the dominant population of NNIA and NNITSA) is set as 100, the maximum size of active populations of NNIA and NNITSA

are both set as 20 and the maximum size of clone populations of NNIA and NNITSA are both set as 20.

The parameters of TSA operator in NNITSA are set as follows: The iterations of Tabu search are set as 3 and the maximum number of iterations is set as 5.

For traditional multi-objective algorithms, test functions are widely used to show performance of continuous problems based on true Pareto. However, as NNITSA is proposed to solve discrete problems of schedule of agricultural machines, it is hard to determine true Pareto of a problem. So it is not realistic to use classic test functions to invalidate its performance. We employ a performance indicator *C*-metric [34]. Given two Pareto approximation sets *A* and *B* obtained by two algorithms, the definition of *C*-metric is as follows:

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \leq b\}|}{|B|}, \quad (9)$$

C-metric is a coverage degree of two finite sets, measures how much *A* covers *B* or *B* covers *A*. For $b \in B$, if $\exists a \in A : a \leq b$, which means *b* is dominated by at least one solution in *A*. Thus, *C*(*A*, *B*) shows the percentage of solutions in *B* which are dominated by any solution in *A*. What is more, when *C*(*A*, *B*) = 1, all solutions in *B* are dominated by some solutions in *A*; when *C*(*A*, *B*) = 0, there is no solution in *B* dominated by solutions in *A*.

The location and amount of service required of each field are summarized in Table 4. All algorithms are performed 10 runs, and the comparative results are shown in Table 5 (A: NNITSA, B: NSGA-II, C: NNIA).

We observe that from the first and second columns, all values of *C*(*A*, *B*) are greater than *C*(*B*, *A*) for all runs. Taking the first run for example, *C*(*A*, *B*) = 1 and *C*(*B*, *A*) = 0 means all solutions obtained by NSGA-II are dominated by those obtained by NNITSA. Taking Table 6 into consideration, NSGA-II has a high time cost compared to two other algorithms. It is because NSGA-II performs dominant ranking on Pareto solutions during each iteration, which takes a lot of time. Experiments show that NNITSA outperforms NSGA-II not only in performance but also in computation time. From the third and fourth columns, similar results are observed for NNITSA and NNIA. There is a big difference between the value of *C*(*A*, *C*) and *C*(*C*, *A*) for all runs. Which means that NNITSA obviously outperforms the NNIA.

From Table 6, we can see that the time cost of NNITSA is a little greater than that of NNIA on average. This suggests that NNITSA improves search efficiency while cannot avoid the fact that it takes more time for neighborhood searches.

The Pareto front of the three algorithms for one run is shown in Fig. 6. We can see that NSGA-II is obvious not better than the other two algorithms. NNITSA has a better distribution and performance compared to NNIA. What is

Table 4 Parameters of fields

No. of fields	<i>x</i> / <i>y</i> -coordinate of field location (<i>km</i>)	Service
1	8.3469,41.7744	12.8
2	14.9865,33.9493	13.5
3	17.3388,30.1454	13.8
4	9.6730,10.6216	15.9
5	45.3797,42.2835	12.6
6	29.1984,3.26008	15.4
7	41.6799,12.8014	14.6
8	52.1649,16.7168	13.5
9	52.6875,17.8132	12.8
10	62.7973,17.1121	13.7
11	81.7547,19.8118	13.9
12	97.2974,5.26769	12.5
13	80.0330,26.9119	14.4
14	81.8148,37.2409	12.9
15	72.2439,48.9687	13.5
16	80.3364,53.9126	14.3
17	98.4063,66.9175	13
18	83.1379,66.6338	13.2
19	82.5313,94.2736	14.5
20	64.8991,73.7858	13.1
21	65.9605,95.1630	15.3
22	65.6859,99.9080	15.9
23	51.8594,92.0332	14.9
24	39.0937,70.1098	13.4
25	39.9257,66.6527	14.3
26	13.3171,98.3052	12.4
27	43.1651,56.1199	15.6
28	1.54871,88.1866	15.5
29	43.2391,54.7870	15.3
30	6.04711,69.8105	13

Table 5 Results of three algorithms

runs	<i>C</i> (<i>A</i> , <i>B</i>)	<i>C</i> (<i>B</i> , <i>A</i>)	<i>C</i> (<i>A</i> , <i>C</i>)	<i>C</i> (<i>C</i> , <i>A</i>)
1	1	0	0.71	0
2	1	0	0.70	0.21
3	1	0	0.97	0
4	1	0	1	0
5	1	0	1	0
6	1	0	0.910	0
7	1	0	0.74	0
8	1	0	0.81	0.06
9	1	0	0.86	0
10	1	0	1	0

more, we can take the extreme points of NNIA and NNITSA into consideration. As shown in Table 7, the extreme points

Table 6 Time costs of three algorithms

Runs	NNITSA	NSGA-II	NNIA
1	39.94	215.76	14.50
2	41.14	218.62	14.83
3	31.62	220.15	14.42
4	34.91	218.18	14.53
5	61.27	217.36	13.73
6	38.47	216.32	14.47
7	30.85	216.06	14.52
8	32.80	220.68	14.59
9	29.02	220.35	14.89
10	46.47	218.98	14.47
Average	38.65	218.25	14.50

Fig. 6 Pareto front of three algorithms

and number of Pareto solutions are compared. First, although the number of Pareto solutions of NNITSA is less than that of NNIA in average, the diversity of solutions of NNIA is a little bad. In most of the runs, NNITSA can always find the best minimum of f_1 (3.3174, 136.1000), while NNIA can only find four times in ten runs. Taking the sixth run for example, NNITSA got a worse minimum of f_1 (3.5697, 28.3000), but it got a better minimum of f_2 (4.0588, 0.1000) in compensation. From the row of Best and Worst, we can see that NNITSA can get a better result than NNIA in the extreme points. So we can still come to the conclusion that NNITSA plays better than NNIA.

Experiments results and analysis of dynamic cross-regional cooperative scheduling

Each arrangement of serving scheme is denoted as the vector $\mathbf{X} = (x^1, x^2, \dots, x^H)$, where x^j is the number of machine serving the j -th field in according district.

The corresponding schedule of Table 8 is shown in Fig. 7.

To fully demonstrate the complexity of the problem, we have considered the situation of one depot, two depots and three depots. The parameters of fields and algorithms are the same with the static problem. All algorithms are also performed 10 runs, and the comparative results are shown in Table 9 (A: NNITSA, B: NSGA-II, C: NNIA). To illustrate the performance of the proposed algorithm in dynamic problem, taking the failure of the agricultural machine 1 in $t = 3$ as an example.

All the travel path and service finished by machines are shown with solid lines and the planned path and service shown with dotted lines in Figs. 8, 10 and 12, where the red triangle represents the location of the depot. The current locations of the machines are shown under Constraint5. By performing NNITSA, one of the cross-regional cooperative schedules is shown in Figs. 9, 11 and 13.

In Figs. 8 and 9, the axis x represent the value of x -coordinate of the field and the axis y represent the value of y -coordinate of the field. It can be seen that (1) M_1 is originally planned to serve field 10, 12, 11 and so on. When M_1 is broken down, the left service of M_1 is assigned to M_4 and M_5 . When M_5 finished its service of field 5 in D_2 , it will help to serve fields in D_1 . (2) Field 8 is in the adjacent of field 9 but they belong to different districts. Machine M_2 of D_1 would help to finish the work in field 7 in D_2 and Machine M_6 of D_2 would help to finish the work in field 10 in D_1 . (3) There is no work assigned to M_7 , so it can return to the agricultural machineries center.

Taking Tables 9 and 10 into consideration, all the values of $C(A, B) = 1$ and $C(B, A) = 0$ reflect that NNITSA fully triumphs over NSGA-II in terms of both performance and time. And most solutions obtained by NNITSA are better than NNIA, which means NNITSA outperforms NNIA. Although the running time of NNITSA is bigger than NNIA, it is worthwhile so that the searching performance can be proved.

From Tables 11 and 12, we can see that the improved algorithm outperforms the other algorithm in all situations of different number of depots.

We use the robustness to measure how much the schedule cost changes compared with that before the dynamic situation occurs [29]. The cost can be the energy of the machines consumed in the schedule. The robustness is denoted as below, where $i = 1, 2K, 6$ represents different conditions.

$$RT_i = \frac{\text{Energy_after}_i - \text{Energy_before}}{\text{Energy_before}}. \quad (10)$$

According to Eq. (10), $RT_i \in [0, +\infty)$, the smaller RT_i is, the more robust the schedule is.

The robustness of Pareto solutions obtained by NNITSA in different conditions are compared in Fig. 14. The numer-

Table 7 Comparison of the extreme points and number of Pareto solutions

No.	NNIA			NNITSA		
	minf1(f1,f2)	minf2(f1,f2)	NumPareto	minf1(f1,f2)	minf2(f1,f2)	NumPareto
1	3.3174,136.1	4.1000,0.2000	24	3.3174,136.1000	3.9004,0.3000	21
2	3.3174,136.1	4.2964,0.2000	23	3.3174,136.1000	4.1707,0.1000	19
3	3.4376,121.1000	4.8394,0.1000	30	3.3174,136.1000	4.0588,0.1000	20
4	3.6600,63.7000	4.5458,0.1000	21	3.3174,136.1000	4.2322,0.1000	20
5	3.6600,63.7000	4.1646,0.5000	20	3.3174,136.1000	3.9801,0.2000	20
6	3.5185,77.6000	4.7447,0.1000	23	3.5697,28.3000	4.0588,0.1000	13
7	3.3174,136.1000	4.2895,0.2000	27	3.3174,136.1000	4.2055,0.1000	23
8	3.4376,121.1000	5.0367,0.3000	26	3.4376,121.1000	4.3561,0.1000	18
9	3.3174,136.1000	4.4006,0.4000	23	3.3174,136.1000	4.3222,0.1000	17
10	3.4376,121.1000	4.3328,0.3000	21	3.3174,136.1000	4.34302,0.1000	23
Best	3.3174,136.1000	4.7447,0.1000	30	3.3174,136.1000	4.0588,0.1000	21
Worst	3.6600,63.7000	5.0367,0.3000	21	3.5697,28.3000	3.90044,0.3000	13

Table 8 Example of individual schedule

j	1	2	3	4	5	6	7	8	9	10
x^j	2	1	1	1	1	4	3	1	1	1
j	11	12	13	14	15	16	17	18	19	20
x^j	7	7	7	7	5	7	7	6	8	8
j	21	22	23	24	25	26	27	28	29	30
x^j	12	11	12	12	12	10	10	10	9	9

ical robustness in Fig. 14 is selected as the minimum value of each solution of the Pareto solution set, which represents the optimal robust solution of the set. Comparisons of the results of one depot show that the robustness improves with the increase of the number of dynamic machines appears, the robustness becomes better in general. What is more, when the number of machines is greater than three in the second condition ($t=5$), the robustness of the schedules drops to less than zero, which reveals that less machines may cost less. In the situation of two and three depots, the robustness basically tends to a consistent scope, which shows the number of depots affect the robustness of a scheduling scheme.

Conclusions

In this paper, a multi-objective optimization algorithm NNITSA is proposed for static partitioning optimization and dynamic cross-regional co-operative scheduling problem of agricultural machinery. To increase the searching efficiency, a local search TSA operator is introduced based on the Tabu search algorithm. In addition to solving a dynamic cross-regional co-operative scheduling problem, an evaluation is adopted to estimate different schedules and help to choose the

Fig. 7 Schedule of Table 8**Table 9** Results of three algorithms

Runs	$C(A, B)$	$C(B, A)$	$C(A, C)$	$C(C, A)$
1	1	0	1	0
2	1	0	1	0
3	1	0	0.8333	0.0909
4	1	0	1	0
5	1	0	0.3750	0.25
6	1	0	1	0
7	1	0	1	0
8	1	0	1	0
9	1	0	1	0
10	1	0	1	0

Fig. 8 The planned path of machines (one depot)

Fig. 11 The schedule after a machine is broken (two depots)

Fig. 9 The schedule after a machine is broken (one depot)

Fig. 12 The planned path of machines (three depots)

Fig. 10 The planned path of machines (two depots)

Fig. 13 The schedule after a machine is broken (three depots)

Table 10 Time costs of three algorithms

Runs	NNITSA	NSGA-II	NNIA
1	291.17	1476.56	18.85
2	367.41	1488.61	18.76
3	153.42	1499.45	17.99
4	173.43	1541.36	17.74
5	138.73	1468.29	18.15
6	128.95	1477.71	18.28
7	195.09	1482.79	18.24
8	220.81	1496.94	18.10
9	216.04	1491.42	18.06
10	282.29	1498.39	17.92

Table 11 Results of three algorithms

Runs	Two depots		Three depots	
	$C(A, C)$	$C(C, A)$	$C(A, C)$	$C(C, A)$
1	1	0	1	0
2	1	0	1	0
3	0.72	0	0.91	0.08
4	1	0	1	0
5	1	0	1	0
6	1	0	0.95	0.11
7	0.79	0.13	1	0
8	1	0	1	0
9	1	0	1	0
10	1	0	0.94	0

Table 12 Time costs of three algorithms

Runs	Two depots		Three depots	
	NNITSA	NNIA	NNITSA	NNIA
1	205.56	19.20	153.26	19.79
2	219.61	18.23	118.50	19.67
3	73.77	18.30	165.86	18.36
4	117.10	17.83	223.13	18.63
5	396.86	18.00	173.78	18.05
6	146.18	17.64	110.49	18.36
7	74.42	18.10	165.07	18.03
8	181.74	17.74	88.57	18.47
9	275.97	18.00	204.65	18.97
10	220.90	17.98	113.71	17.95

appropriate solution from the Pareto solutions. The route in practical production environment is also considered to refine of machineries. Finally, NNITSA is applied to one static case and one dynamic case. The results compared with NNIA and NSGA-II indicate that NNITSA outperforms the other two algorithms.

(a) One depot **(b)** Two depot **(c)** Three depot**Fig. 14** Robustness as different number of dynamic machines

In the current complex agricultural production environment, such as biomass production, the demand for fleet management will increase and the cost of the production will be reduced [47]. In future, we will design a multi-operational scheduling algorithm which focuses on the optimization of task sequence of agricultural machineries.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (No. 61903078), the Fundamental Research Funds for the Central Universities (Nos. 2232021A-10, 2232020D-48) and Natural Science Foundation of Shanghai (20ZR1400400).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Jiang X, Yue Y, Min Y et al (2021) Particle Swarm optimization with multiple adaptive subswarms. *J Phys* 1757(1):012024
- Zhao L, Zeng Z, Wang Z et al (2021) PID control of vehicle active suspension based on particle Swarm optimization. *J Phys* 1748(3):028–032
- Wu Y, Sun X, Yang P et al (2021) Transformer fault diagnosis based on improved particle Swarm optimization to support Vector Machine. *J Phys* 1750(1):012074
- Shang RH, Jiao LC, Liu F et al (2012) A novel immune clonal algorithm for MO problems. *IEEE Trans Evol Comput* 16(1):35–50
- Li M, Yang S, Li K et al (2014) Evolutionary algorithms with segment-based search for multiobjective optimization problems. *IEEE Trans Cybernet* 44(8):1295–1313
- Zhang Q, Lu J, Jin Y (2021) Artificial intelligence in recommender systems. *Complex Intell Syst* 7(1):439–457
- Hua Y, Jin Y, Hao K et al (2020) Generating multiple reference vectors for a class of many-objective optimization problems with degenerate Pareto fronts. *Complex Intell Syst* 6:276–285
- Meng Z, Yang C, Li X (2020) Di-DE: depth information-based differential evolution with adaptive parameter control for numerical optimization. *IEEE Access* 8:40809–40827
- Wei Y, Feng Q, Yuan S (2020) Differential evolution algorithm based on ensemble of constraint handling techniques and multi-population framework. *Int J Intell Sci* 10(2):22–40
- Nie X, Luo J (2021) The hybrid intelligent optimization algorithm and multi-objective optimization based on big data. *J Phys* 1757(1):012132
- Fu L, Zhu H, Zhang C (2021) Hybrid harmony search differential evolution algorithm. *IEEE Access* 9:1–1
- Shaik T, Ravi V, Deb K (2021) Evolutionary Multi-Objective Optimization Algorithm for Community Detection in Complex Social Networks. *SN Computer Science* 2(1):1–25
- Wang F, Liao F, Li Y (2021) An ensemble learning based multi-objective evolutionary algorithm for the dynamic vehicle routing problem with time windows. *Comput Ind Eng* 154(3):107–131
- Shaw SS, Ahmed S, Malakar S et al (2021) Hybridization of ring theory-based evolutionary algorithm and particle swarm optimization to solve class imbalance problem. *Complex Intell Syst.* (2021). <https://doi.org/10.1007/s40747-021-00314-z>
- Zki A, Babalik A (2017) A novel metaheuristic for multi-objective optimization problems: the multi-objective vortex search algorithm. *Inf Sci* 402:124–148
- Xu LK, Yu G, Jiang Y (2015) Energy-efficient resource allocation in single-cell OFDMA systems: multi-objective approach. *IEEE Trans Wirel Commun* 14(10):5848–5858
- Liu BJ, Bi XJ (2021) Adaptive-constraint multi-objective evolutionary algorithm based on decomposition and differential evolution. *IEEE Access* 9:17596–17609
- Zille H, Ishibuchi H, Mostaghim S et al (2017) A framework for large-scale multi-objective optimization based on problem transformation. *IEEE Trans Evol Comput* 99:1–1
- Wu JH, Zhang J, Zhang XG et al (2011) Hierarchical co-evolution immune algorithm and its application on TSP. *Acta Electronica Sinica* 39(2):336–344
- Srinivas N, Deb K (1994) Multiobjective optimization using non-dominated sorting in genetic algorithms. *Evol Comput* 2(3):221–248
- Deb K, Agrawal S, Pratap A et al (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans Evolut Comput* 6(2):182–197
- Zitzler E, Thiele L (1999) Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
- Adham AM, Mohd-Ghazali N, Ahmad R (2015) Performance optimization of a microchannel heat sink using the Improved Strength Pareto Evolutionary Algorithm (SPEA2). *J Eng Thermophys* 24(1):86–100
- Gong M, Jiao L, Yang D (2008) Multiobjective immune algorithm with nondominated neighbor-based selection. *Evol Comput* 16(2):225–255
- Liu ZZ, Wang BC, Tang K (2021) Handling constrained multi-objective optimization problems via bidirectional coevolution[J]. *IEEE Trans Cybernet PP*(99):1–14
- Yang Z, Ding Y, Jin Y et al (2018) Immune-endocrine system inspired hierarchical coevolutionary multiobjective optimization algorithm for iot service[J]. *IEEE Trans Cybern PP*:1–14
- Tavarespereira F, Figueira J, Mousseau V et al (2007) Multiple criteria districting problems: the public transportation network pricing system of the Paris region. *Ann Oper Res* 154(1):69–92
- Rashid BANM, Choudhury T (2021) Cooperative co-evolution and Mapreduce: a review and new insights for large-scale optimisation. *Int J Inf Technol Project Manage (IJITPM)* 12(1):29–62
- Guo YN, Cheng J, Luo S et al (2018) Robust dynamic multi-objective vehicle routing optimization method. *IEEE/ACM Trans Comput Biol Bioinf* 15(6):1891–1903
- Hu ZH, Ding YS, Shao Q (2009) Immune co-evolutionary algorithm based partition balancing optimization for tobacco distribution system. *Expert Syst Appl* 36(3):5248–5255
- Tian Y, Peng S, Rodemann T (2019) Automated selection of evolutionary multi-objective optimization algorithms. *IEEE Sympos Ser Comput Intell (SSCI)* 2019:3225–3232
- Tan Z, Wang H (2020) A Kriging-assisted evolutionary algorithm using feature selection for expensive sparse multi-objective optimization. *IEEE Congress Evolut Comput (CEC)* 2020:1–8
- Guo J, Shao M, Jiang S (2020) An improved multiobjective optimization evolutionary algorithm based on decomposition with hybrid penalty scheme. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pp 165–166

34. Lei HT, Wang R, Gilbert L (2016) Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. *Comput Operat Res* 67:12–24
35. Marsiglio S, Privileggi F (2019) On the economic growth and environmental trade-off: a multi-objective analysis. *Ann Oper Res* 296:1–2
36. Martinez-Puras A, Pacheco J (2016) MOAMP-Tabu search and NSGA-II for a real bi-objective scheduling-routing problem. *Knowl-Based Syst* 112:92–104
37. Edwards G, Sorensen CG, Bochtis DD et al (2015) Optimised schedules for sequential agricultural operations using a Tabu Search method. *Comput Electron Agric* 117:102–113
38. Conesa-Muoz J, Pajares G, Ribeiro A (2016) Mix-opt: a new route operator for optimal coverage path planning for a fleet in an agricultural environment. *Expert Syst Appl* 54:364–378
39. Bochtis DD, Sorensen CG, Busato P (2014) Advances in agricultural machinery management: a review. *Biosys Eng* 126(39):69–81
40. Bochtis DD, Dogoulis P, Busato P, Sorensen CG (2013) A flow-shop problem formulation of biomass handling operations scheduling. *Comput Electron Agric* 91:49–56
41. Xu N, Ding Y, Ren L et al (2018) Degeneration recognizing clonal selection algorithm for multimodal optimization. *IEEE Trans Cybernet* 48:848–861
42. Huang YJ, Ding YS, Hao KR et al (2017) A multi-objective approach to robust optimization over time considering switching cost. *Inf Sci* 394:183–197
43. Ding YS, Chen R, Hao KR (2016) A rule-driven multi-path routing algorithm with dynamic immune clustering for event-driven wireless sensor networks. *Neurocomputing* 203:139–149
44. Yao GS, Ding YS, Ren LH et al (2016) An immune system-inspired rescheduling algorithm for workflow in cloud systems. *Knowl-Based Syst* 99:39–50
45. Bochtis DD, Sorensen CG, Vougioukas SG (2010) Path planning for in-field navigation-aiding of service units. *Comput Electron Agric* 74(1):80–90
46. Lei HT, Wang R, Laporte G (2016) Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. *Comput Operat Res* 62:12–24
47. Orfanou A, Busato P, Bochtis DD et al (2013) Scheduling for machinery fleets in biomass multiple-field operations. *Comput Electron Agric* 94(11):12–19

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.