applied sciences

Article

A Project Scheduling Game Equilibrium Problem Based on Dynamic Resource Supply

Cuiying Feng [1,2,3], Shengsheng Hu [1,*], Yanfang Ma [4] and Zongmin Li [2]

1 School of Management, Zhejiang University of Technology, Hangzhou 310023, China
2 Business School, Sichuan University, Chengdu 610064, China
3 Zhejiang Keyou Information Engineering Co., Ltd., Yiwu 322001, China
4 School of Economics and Management, Hebei University of Technology, Tianjin 300401, China
* Correspondence: zjuthss@163.com

Abstract: In a resource-constrained project scheduling problem, most studies ignore that resource supply is a separate optimization problem, which is not in line with the actual situation. In this study, the project scheduling problem and the resource supply problem are regarded as a dynamic game system, with interactive influences and constraints. This study proposes a Stackelberg dynamic game model based on the engineering supply chain perspective. In this model, the inherent conflicts and complex interactions between the Multi-mode Resource-Constrained Project Scheduling Problem (MRCPSP) and the Multi-Period Supply Chain Problem (MPSCP) are studied to determine the optimal equilibrium strategy. A two-level multi-objective programming method is used to solve the problem. The MRCPSP is the upper-level planning used to optimize project scheduling and activity mode selection to minimize project cost and duration; MPSCP is a lower-level planning method that seeks to make resource transportation decisions at a lower cost. A two-layer hybrid algorithm, consisting of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), is proposed to determine the optimal equilibrium strategy. Finally, the applicability and effectiveness of the proposed optimization method are evaluated through a case study of a large hydropower construction project, and management suggestions for related departments are provided.

Keywords: multi-cycle supply chain; project scheduling; Stackelberg dynamic game; two-level multi-objective programming; GA with double strings; particle swarm optimization

Citation: Feng, C.; Hu, S.; Ma, Y.; Li, Z. A Project Scheduling Game Equilibrium Problem Based on Dynamic Resource Supply. *Appl. Sci.* **2022**, *12*, 9062. https://doi.org/10.3390/app12189062

Academic Editor: Antonio J. Nebro

Received: 20 August 2022
Accepted: 6 September 2022
Published: 9 September 2022

1. Introduction

Managing work in the form of projects has become a common practice to improve work efficiency. Currently, approximately 20% of the world's economic activity is in the form of projects, generating an annual economic value of roughly $12 trillion [1]. Project scheduling refers to the scientific and reasonable arrangement of the beginning and execution times of each activity in a project in order to achieve the established goal [2]. The Resource-Constrained Project Scheduling Problem (RCPSP) is a form of planning based on constraining the resources required by project activities. The Classic RCPSP scheduling decision must satisfy the temporal and resource constraints, and its solution is a scheduling plan that optimizes the management objective under these constraints. [3].

Many scholars have studied the RCPSP, and extension problems have been developed. Liu et al. [4] designed an RCPSP model based on the time window delay from the perspective of owner-contractor interaction. Kim et al. [5] considered the delay penalty on the basis of minimizing the total project time. Cheng et al. [6] considered the problem of night shifts in construction projects and minimized the project duration, cost, and utilization of night shifts while meeting the constraints of operational logic and labor availability. In the study of Demeulemeester and Herroelen [7] as well as Debels and Vanhoucke [8], an activity can be interrupted after every integer unit of its activity time. Muritala Adebayo

Isah and Byung-Soo Kim [9] presented a stochastic multiskilled resource scheduling model for RCPSP, which considers the impacts of risk and uncertainty on activity durations. The standard RCPSP assumes that an activity can only be executed in one mode, with a fixed duration and resource requirements. On this basis, Elmaghraby [10] proposed a new concept; in practice, management departments can flexibly arrange appropriate execution modes for project activities to achieve corresponding goals, and each mode has different durations and resource demands, i.e., the Multi-mode Resource-Constrained Project Scheduling Problem (MRCPSP). Varma et al. [11] discussed a multi-mode problem without the use of non-renewable resources. Zhu et al. [12] considered the MRCPSP with generalized resource constraints. Bellenguez and Emmanuel [13] discussed a special case: in an MRCPSP, each activity requires specific skills, while resources are employees with fixed skills, and employees must be selected according to their skills when arranging activities.

MRCPSP is a critical issue in engineering supply chain management, especially in large-scale engineering construction projects. The resource supply is complex and changeable, and the resource transportation policy is updated according to the different ordering schedules of project scheduling [14]. At this point, a Multi-Period Supply Chain Problem (MPSCP) arises, directly affecting both the cost and schedule of the project. If the project schedule is made without considering the constraints of upstream resource supply capacity, the supply delay or interruption of suppliers will delay the construction period and increase both the project cost and risk, among other factors. Similarly, resource supply driven by non-engineering schedule planning will lead to a lower resource utilization rate and a higher inventory cost. In this case, resource constraint is not only a constraint condition of MRCPSP, but also an optimization problem closely related to MRCPSP with the characteristics of a dynamic game. However, in most studies, project scheduling and resource supply are considered as two independent optimization problems, ignoring the interaction and conflict between them, possibly leading to a suboptimal solution for resource supply and project delay. Therefore, it is more realistic to consider project scheduling and resource supply as an integrated system for dynamic game optimization.

Relevant research by Sarker [15] demonstrated that the simultaneous optimization of project scheduling and resource supply can improve the efficiency of project scheduling and reduce the overall cost. Xie et al. [16] took the project duration and cost as the optimization objectives, considered the variable resource availability and expressed it by interval variables, and established a dual-objective optimization model of the MRCPSP under the constraint of variable resource availability. Lv et al. [17] further expanded renewable resources into flexible resources with capacity differences, and established a problem model considering capacity differences in which the capacity level affects activity duration. Schwindt and Trautmann [18] considered the time-dependent resource capacity and divided the aggregate demand of intermediate and final products into batches in the batch production mode. Shu-Shun Liu et al. [19] proposed a two-stage optimization model based on constrained programming to address the bridge maintenance scheduling problem.

Many scholars have proposed rich algorithms to solve the integrated system optimization problem of project scheduling and resource supply chains. Asta et al. [20] designed a hybrid algorithm that combines Monte Carlo and hyper-heuristic methods to solve this problem. Xie et al. [21] studied MRCPSP under the condition of uncertain activity duration and designed an approximate dynamic programming algorithm based on the rollout to solve it. Peteghem et al. [22] studied MRCPSP with resource preemption characteristics, introduced an extended serial scheduling generation scheme to improve mode selection, and designed a two-population genetic algorithm. Furthermore, many studies have proven that GA and PSO are more effective and have different advantages in solving such problems [23–27].

GA was first proposed by J. Holland in 1975. It is a random search algorithm that draws on natural selection and genetic mechanisms in the biological world and follows the principle of "survival of the fittest" [28]. Its basic idea is to imitate the natural evolution process through genetic manipulation of individuals with certain structural forms in the population, so as to generate a new population and gradually approach the optimal solution. PSO was proposed by J. Kennedy and R. C. Eberhart in 1995 [29]. It is a random search algorithm based on group cooperation, developed by simulating the foraging behavior of birds. It finds the global optimum by following the currently searched optimum.

The existing research has made important achievements in project scheduling problems and algorithm designs. However, when constructing the model, the interaction between decision makers is ignored. Secondly, the project scheduling problem from the perspective of the engineering supply chain is a multi-objective and multi-stage complex decision problem; previous studies [26,27] have shown that using the bi-level programming method can generate better results.

The innovation of this paper is that the project scheduling problem and resource supply problem are regarded as an integrated system of a dynamic game, involving interactive influences and constraints. Moreover, a two-level multi-objective programming method is adopted, which organizes the whole process of "objective—modeling—algorithm—optimization—decision." A large hydropower construction project is taken as an example to prove the scientificity and feasibility of the method.

The rest of the paper is organized as follows: Section 2 gives the key problem statement of MRCPSP-MPSCP integrated system and research methods; Section 3 details the modeling method and hypothesis of establishing the two-level dynamic game model; Section 4 proposes the two-level GADS/DIWPSO hybrid algorithm to solve the established model; Section 5 gives a practical case to emphasize the practicability and effectiveness of the optimization method, and proposes forward management suggestions to related departments; and finally, Section 6 provides conclusions and future research directions.

2. Research Overview

2.1. Problem Description

Project scheduling has always been considered the core of engineering supply chains, as the construction and operation of the supply chain are driven by the development of the project schedule. The engineering construction department first defines the resource demand of each demand point in each time period by forming a project schedule plan; next, the resource supplier attempts to meet the resource demand. However, as the resource supplier is also a decision-making subject with its own constraints, it optimizes its cost and time goals by formulating a resource transportation strategy and sends this information back to the engineering construction department, thus affecting the formulation of the project's schedule. Conflict and cooperation coexist in the engineering supply chain. The engineering construction department has higher decision-making power (i.e., the leader), whereas the resource supplier is subordinate (i.e., the follower). This "leader–follower" behavior is, in its essence, a Stackelberg game, with the characteristics of multi-periodicity in practice. Therefore, project scheduling and resource supply comprise an inseparable integrated system, which is the game analysis and dynamic coordination problem of the integrated system of "project scheduling–resource supply" from the perspective of the engineering supply chain. The successful operation of this system helps reduce project costs, shorten construction periods, and improve project quality and resource utilization.

The research object of this project is a large hydropower construction project located in southeast China. A concrete double-curvature arch dam is the main project, with many construction activities with priority relationships and shared resources; each activity has several alternative modes, and each mode has a certain duration and resource demand. To meet the requirements of shared resources, it is necessary to specify the ordering time and quantity in each time period when making the project scheduling scheme, and the resource supplier further formulates the resource transportation strategy. These constitute the dynamic game decision-making system of the MRCPSP-MPSCP integrated system, and the structural model is shown in Figure 1.

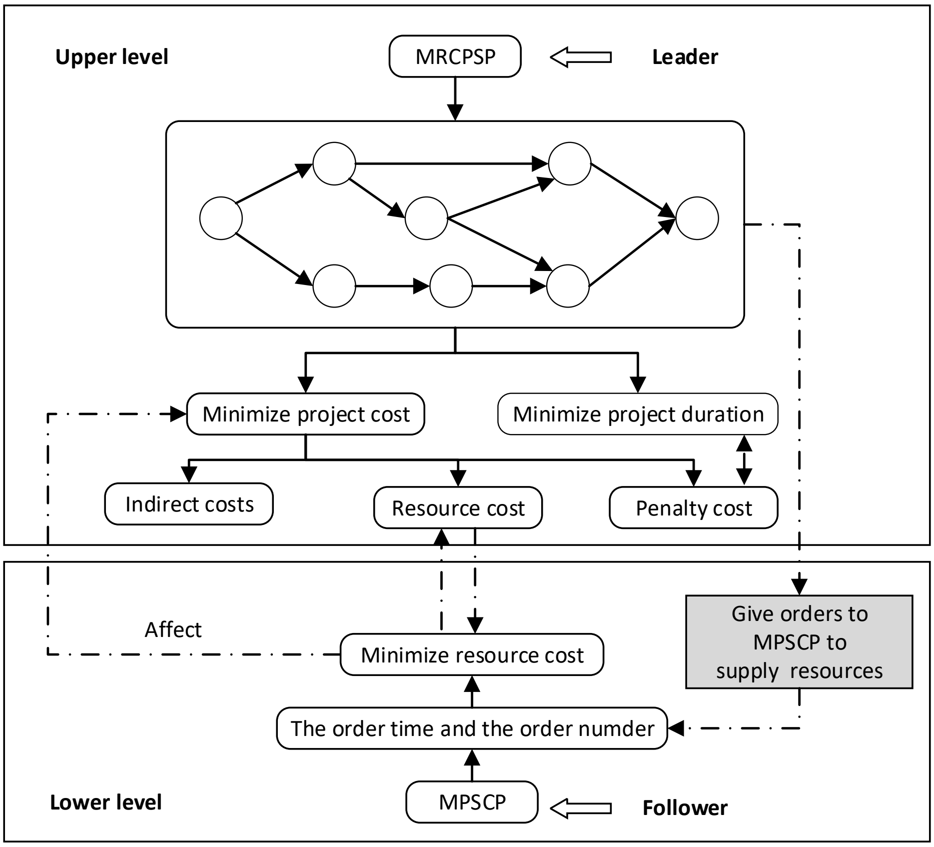

Figure 1. Structural model of the MRCPSP-MPSCP integrated system.

2.2. Research Methodology

In this paper, we adopt a two-level multi-objective mode of programming which informs the whole process of "objective—modeling—algorithm—optimization—decision." According to the characteristics of the dynamic game of this problem in the engineering supply chain, we adopt a two-level modeling method to express the interaction between MRCPSP and MPSCP. To determine the optimal equilibrium strategy of the model, a two-layer hybrid algorithm, composed of a GA with double strings and an improved PSO, is proposed. Considering the existence of many uncertainties in the engineering supply chain, for example, the project activity time is a typical uncertain variable; Bidot et al. [30] considered a project schedule with a random activity duration. In addition, factors such as weather conditions, labor efficiency, and transportation environment make the decision-making process more complicated. Therefore, random variables are used in this study to describe various variables in an uncertain environment. Finally, the applicability and effectiveness of the proposed optimization method are evaluated through a case of a large hydropower construction project.

3. Model Establishment

To properly express the dynamic game characteristics of the MRCPSP-MPSCP integrated system, a two-level multi-objective programming model is established, which includes the upper and lower models.

3.1. Symbols and Assumptions

3.1.1. Indicators

j: Project activity index, $j \in J = \{1, 2, \ldots, J\}$
k: Material type index, $k \in K = \{1, 2, \ldots, K\}$
t: Time period index, $t \in \{1, 2, \ldots\}$
m: Activity mode index, $m \in M = \{1, 2, \ldots, M\}$

i: Activity mode index, $i \in I = \{1, 2, \ldots, I\}$
s: Demand point index, $s \in S = \{1, 2, \ldots, S\}$

3.1.2. Parameters Related to Project Scheduling

B: Total available budget
D: Project planning cycle
IC_s: Inventory capacity at the demand point s
P_j: Set of predecessors of activity j
c_{jm}: Direct cost of activity j in mode m
d_{jm}: Operation time of activity j in m mode
r_{jmk}: The demand of m mode of activity j for resource k in each time period
r_k: Maximum supply capacity of resource k in each time period
c_0: Overhead cost per time period
EF_j: The earliest completion time of activity j
LF_j: The latest completion time of activity j
R_s: The set of activities for which demand point s is responsible
Pc_k: Unit purchase cost of resource k
Oc_k: Each order cost of resource k
Ic_k: Storage cost of resource k in each time period

3.1.3. Parameters Related to Resource Supply

$T(t)$: Delivery date of time period t
R_k: Maximum amount of resource k transported each time
P_{ik}: Supply capacity of resource k at supply point i
c_{isk}: Unit transportation cost of resource k on the transportation path (i, s)
t_{isk}: Unit transportation time of resource k on the transportation path (i, s)

3.1.4. Decision Variables

$v_{isk}(t)$: The allocation of resource k on transportation path (i, s) in time period t
$x_{jmt} = \begin{cases} 1, \text{ If activity } j \text{ executes mode } m \text{ in time period } t \\ 0, \text{Otherwise} \end{cases}$, represents the mode selection of activity j
$z_{kt} = \begin{cases} 1, \text{ If resource } k \text{ transported at the beginning of time period } t \\ 0, \text{Otherwise} \end{cases}$, represents whether resource k is transported during time period t

3.1.5. Intermediate Variables

ST_j: Starting time of activity j
FT_j: Completion time of activity j
A_t: Activity set of ongoing jobs in time period t
S_{kst}: The remaining amount of resource k at demand point s at the end of the time period t

3.1.6. Assumptions

① The project contains j activities and two virtual activities, in which the two virtual activities represent the initial and final activities of the project, denoted as $j = 0$ and $j = J + 1$, respectively.

② Only when all the predecessor activities of the activity are completed can the activity begin.

③ Each activity can only execute one mode without interruption.

④ The supply capacity of the supply point and the inventory capacity of the demand point are limited and cannot be increased.

⑤ The loading and unloading costs and time of the transport vehicles were included in the corresponding transport costs and time.

⑥ Uncertain parameters, such as resource demand, project activity time, and unit transportation cost, are random variables.

⑦ Resources are consumed evenly in each time period.

3.2. Project Scheduling

Project scheduling occupies a dominant position in an engineering supply chain with the contractor as the core. In view of the project scheduling problem, under the condition of ensuring the quality of project, duration and cost are its three major objectives.

3.2.1. Schedule Objective

One of the most important goals in project scheduling is to minimize the project duration and complete the project as early as possible under all constraints. In this study, the completion time of the last activity $(J+1)$ can be used to describe the duration of the project; that is, the duration F_t can be expressed as Equation (1).

$$F_t = \sum_{t=EF_{(J+1)}}^{LF_{(J+1)}} \sum_{m=1}^{M_{(J+1)}} t x_{(J+1)mt} \tag{1}$$

3.2.2. Cost Objective

Cost is another important goal in project scheduling. Project costs are generally divided into direct and indirect portions. Among them, ordering, purchasing, and storage costs belong to direct costs; indirect costs belong to fixed costs in any time period and are related to project duration. In summary, the cost function F_c can be expressed by Equation (2):

$$\begin{aligned} F_{\mathrm{c}} = {} & \sum_{j=1}^{J} \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_{\mathrm{j}}} d_{jm} x_{jmt} \sum_{k=1}^{K} r_{jmk} Pc_k + \sum_{j=1}^{J} \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_{\mathrm{j}}} c_{jm} x_{jmt} + \sum_{k=1}^{K} \sum_{t=1}^{Ft} Oc_k z_{kt} \\ & + \frac{1}{2} \sum_{j=1}^{J} \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_{\mathrm{j}}} d_{jm} x_{jmt} \sum_{k=1}^{K} r_{jmk} Ic_k + \sum_{k=1}^{K} \sum_{t=1}^{Ft} Ic_k S_{kst} + c_0 Ft \end{aligned} \tag{2}$$

3.2.3. Constraints

In full consideration of the actual situation of the engineering supply chain, the constraints are listed in this section. This will make the model more realistic.

$$\sum_{j \in R_s} \sum_{m=1}^{M_j} r_{jmk} \sum_{t'=t}^{t+d_{jm}-1} x_{jmt} \leq S_{ks(t-1)} + \sum_{i=1}^{I} v_{isk}(t), \forall k \in K, s \in S, t \in FT \tag{3}$$

$$\sum_{j \in A_t} \sum_{m=1}^{M_j} r_{jmk} \leq r_k, \forall k \in K, t \in FT \tag{4}$$

$$\sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} t x_{jmt} \leq \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} (t - d_{jm}) x_{jmt}, \forall j \in P_j, j \in J \tag{5}$$

$$S_{kst} = S_{ks(t-1)} + \sum_{i=1}^{I} v_{isk}(t) - \sum_{j \in R_s} \sum_{m=1}^{M_j} r_{jmk} \sum_{t'=t}^{t+d_{jm}-1} x_{jmt'}, \forall k \in K, s \in S, t \in FT. \tag{6}$$

$$S_{ks0} = 0, S_{ksFt} = 0, \forall k \in K, s \in S \tag{7}$$

$$S_{kst} \geq 0, \sum_{k=1}^{K} S_{kst} \leq IC_s, \forall k \in K, s \in S, t \in FT \tag{8}$$

$$F_t \leq D, F_c \leq B \tag{9}$$

$$FT_j = \sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} tx_{jmt}, ST_j = FT_j - d_{jm}, \forall j \in J \tag{10}$$

$$\sum_{m=1}^{M_j} \sum_{t=EF_j}^{LF_j} x_{jmt} = 1, \forall j \in J \tag{11}$$

$$x_{jmt} \in \{0, 1\}, \forall j \in J, t \in FT \tag{12}$$

$$z_{kt} = \begin{cases} 1, \sum\limits_{i=1}^{I} \sum\limits_{s=1}^{S} v_{isk}(t) > 0 \\ 0, \sum\limits_{i=1}^{I} \sum\limits_{s=1}^{S} v_{isk}(t) = 0 \end{cases} \tag{13}$$

Constraint condition Formula (3) represents the resource constraint. Equation (4) represents that the total consumption of resource k in each time period cannot exceed its maximum supply capacity. Equation (5) is the predecessor constraint. Equation (6) represents the remaining amount of available resources at the end of each time period, which can be regarded as a state transition variable. Equation (7) indicates that, to maximize the utilization of resources, the resource surplus should be zero at the beginning and end of the project. Equation (8) indicates that the resource surplus at the end of each period is greater than or equal to zero, and cannot exceed the inventory capacity. Equation (9) indicates the construction period and budget constraint. Equation (10) represents the start and end times of each activity. Equations (11)–(13) are logical constraints: each activity should be executed within the range of the earliest and latest completion times, and only one activity mode can be executed. Meanwhile, there are also characteristic constraints among the decision variables.

3.3. Resource Supply

After the project scheduling scheme is determined, the resource supplier seeks to minimize the total operational cost and transportation time by optimizing the transportation volume between the supply and demand points. The transportation model can be expressed as follows.

3.3.1. Operating Cost Target

The resource supplier transports the corresponding amount of resources to the demand point of the project. The total operating cost (i.e., Z_c) of the resource transport model is the transportation cost from the supply point to the demand point. Therefore, the total operational cost of this model can be expressed by Equation (14):

$$Z_c = \sum_{i=1}^{I} \sum_{s=1}^{S} \sum_{t=1}^{T} \sum_{k=1}^{K} c_{isk} v_{isk}(t) \tag{14}$$

3.3.2. Transport Time Target

Minimizing transportation time is an important goal. The transportation time on the transportation path (i, s) in the time period t can be expressed as $T_{is}(t) = \sum_{k=1}^{K} t_{isk} v_{isk}(t)$. Therefore, the total transportation time in this model can be expressed by Equation (15):

$$Z_t = \sum_{t=1}^{T} \max_{i,s} T_{is}(t) \tag{15}$$

3.3.3. Constraints

Equation (16) is the equation of state variable $B_{ik}(t)$, which represents the amount of resources k remaining at each resource supply point at the end of each time period t. Equation (17) demonstrates that the quantity of resources at each supply point is the maximum supply capacity of the supply point at the beginning, and that the quantity of resources is non-negative throughout the entire process. Equation (18) indicates that the quantity of resources transported to each demand point must satisfy the demand level of project scheduling in terms of the total quantity. Equation (19) indicates that the quantity of transported resources cannot exceed the maximum supply. Equation (20) represents the delivery-date constraint. Equation (21) is a logical constraint.

$$B_{ik}(t) = B_{ik}(t-1) - \sum_{s=1}^{S} v_{isk}(t), \forall i \in I \tag{16}$$

$$B_{ik}(0) = P_{ik}, B_{ik}(t) \geq 0, \forall i \in I \tag{17}$$

$$\sum_{i=1}^{I} v_{isk}(t) \geq \sum_{t=1}^{Ft} \sum_{j=1}^{J} \sum_{m=1}^{M_j} r_{jmk} d_{jm} x_{jmt}, \forall s \in S, \forall k \in K \tag{18}$$

$$\sum_{s=1}^{S} v_{isk}(t) \leq P_{ik}, \forall i \in I, s \in S \tag{19}$$

$$\max \sum\nolimits_{k=1}^{K} t_{isk} v_{isk}(t) \leq T(t) \tag{20}$$

$$0 \leq v_{isk}(t) \leq R_k, \forall k \in K, t \in FT \tag{21}$$

3.4. Global Dynamic Game Optimization Model

After analyzing project scheduling and resource supply, the objective function and constraints are integrated into a dynamic game optimization model, which is more consistent with the coexistence of cooperation and conflict among supply chain members. This provides a theoretical basis for the sustainable operation of the engineering supply chain to improve technological innovation ability, cooperation, and management abilities among the upstream and downstream members.

When all constraints on project scheduling are set to A and resource supply constraints are set to B, then the overall dynamic game optimization model is as follows.

$$\begin{array}{l} \min\{F_t, F_c\} \\ s.t. \begin{cases} A \\ \min\{Z_c, Z_t\} \\ s.t.\{B \end{cases} \end{array} \tag{22}$$

4. Algorithm Design

The MRCPSP-MPSCP integrated system is an NP-hard problem. GA and PSO have been mentioned as the most practical methods to solve this kind of problem. For the problem with the 0–1 decision variable, Sakawa et al. [31] proved that a GA with Double Strings (GADS) shows superior convergence to the simple GA. Therefore, this study draws on several excellent algorithm ideas and proposes a hybrid GAPSO algorithm to solve the dynamic game optimization problem in the engineering supply chain. Specifically, GADS is used to solve the upper MRCPSP, and a Dynamically adjusted Inertial Weight PSO (DIWPSO) is used to solve the underlying MPSCP.

4.1. GADS

In this section, GADS is used to analyze and solve project scheduling. Its primary objective is to determine the execution priority of each activity and arrange the activities. Appropriate encoding methods and decoding rules were selected according to the characteristics of the problem, and the corresponding selection, crossover, mutation, and evolution termination conditions were designed.

4.1.1. Coding Design

To express the execution order of each activity and the characteristics of multiple models in the MRCPSP more reasonably, the algorithm uses the activity-linked list and the corresponding activity-mode-linked list as the code and composes the chromosome. To improve the efficiency of the algorithm, activity J was first stratified according to its priority. The level of each activity is determined as follows: the smaller the tier, the higher the priority of the activities within that tier. In the process of coding, the activities of the small level are always arranged before the activities of the large level, so that the chromosome can ensure the precedence constraint in the subsequent genetic operation and avoid the generation of infeasible solutions. As demonstrated in Table 1, there are nine activities on this chromosome, divided into four levels. The priority of the three activities in level one is higher than those of the other three levels, and the priority of the two activities in level two is higher than those of the three and four levels.

Table 1. Coding design.

Level		1		2			3		4
Activity J	1	3	2	5	4	6	8	7	9
Modes m_j	1	2	1	2	1	2	1	2	1

The priority of the project job is then encoded by numerical coding; that is, the length of the code is equal to the number of project activities, the position of the code represents the priority of activity J in this chromosome, and the number on this position represents the activity number. The higher the order of the activity J, the higher the priority. As indicated in Table 1, the priority of Activity 1 is $J_1 = 1$, which has the highest priority. Activity 9 has the lowest priority.

The job modes of an activity are encoded in a linked list of modes. m_j represents a set of modes of activity J.

4.1.2. Decoding Rules

Herein, a hybrid schedule generation scheme (HSGS) [32] was used as the decoding rule. The earliest start time of an activity can be determined when the predecessors of the activity have been completed and resource requirements have been met. HSGS is used to determine the completion time of each activity in turn and then calculate the total duration of the entire project.

Step 1. Let A_n be the set of activities that have been scheduled, and let U_n be the set of activities that have not been scheduled. When initialized, $A_n = \varnothing$ and $U_n = \{1, 2, \ldots, N\}$. First, the priority of each activity in U_n is sorted in descending order, and the activity with the highest priority is selected for the arrangement.

Step 2. Continue to select the highest priority activity from U_n and conduct a timing constraint judgment. If satisfied, proceed to the next step. If not, the next activity is selected for judgment until the activity that meets the conditions is determined.

Step 3. Conduct resource constraint judgment on the activity to determine whether it can be scheduled in parallel with scheduled activities. If so, proceed to the next step and arrange the activity into A_n; if not, go to Step 2.

Step 4. Update A_n and U_n, then repeat from Step 1 until all activities are scheduled, i.e., $A_n = \{1, 2, \ldots, N\}$ and $U_n = \varnothing$.

4.1.3. Fitness Function

Because there are two objective functions of duration and cost in the upper planning, the fitness function is constructed using the weighted aggregation method, to maintain the effectiveness of the multiple objectives. Let μ_1 and μ_2 represent the weights of the two objective functions; the fitness function can then be represented by Equation (23). After making the changes, the maximum fitness value is required.

$$Fitness(F) = \mu_1 \frac{F_t^{\max} - F_t}{F_t^{\max} - F_t^{\min}} + \mu_2 \frac{F_c^{\max} - F_c}{F_c^{\max} - F_c^{\min}} \tag{23}$$

4.1.4. Genetic Manipulation

Step 1. Set the parameters in the GADS: size L_1, maximum number of iterations T_1, crossover probability p_c, and mutation probability p_m.

Step 2. Initialize L_1 individuals as a group, set the initial iteration $\tau_1 = 0$, and use the coding program to generate the initial individuals $S_l(0)$.

Step 3. Through the elite roulette method to select individuals, according to the size of fitness, develop roulette with slots, and use the roulette to generate the next generation of individuals $(\tau_1 + 1)$. If the fitness function of the l chromosome in the population is $f(S_l)$, then the probability of chromosome S_l being selected is

$$P_l = \frac{f(S_l)}{\sum_{l=1}^{n} f(S_l)} \tag{24}$$

Step 4. Since the chromosome of the algorithm consists of an activity list and a mode list, it is necessary to cross these two lists in steps.

Step 4.1. The activity list is crossed using the alternating crossing method. First, the first gene from parent A is added to offspring A. Then, we select the first gene from parent B and judge whether it is duplicated with genes in offspring A. If it is duplicated or does not conform to the hierarchical order, it is discarded; if it is not repeated but conforms to the hierarchical order, it is added to offspring A.

Step 4.2. The second gene is selected from parent A to judge whether it is duplicated. Finally, the genes in the two parents are selected in turn to form offspring A. Similarly, the genes in parents B and A are selected to obtain child B.

Step 4.3. Then, the mode list was crossed by a single-point operation. Let i be the position of the gene, let N be the total length of the chromosome, and randomly select integer $n_1 < N$. If $1 \le i \le n_1$, then the mode of gene i of offspring A is equal to that of parent A; if $n_1 \le i \le N$, then the mode of child A is equal to that of parent B.

Step 4.4. Similarly, randomly select integer $n_2 < N$. If $1 \le i \le n_2$, then the mode of gene i of offspring B is equal to that of parent A; if $n_2 \le i \le N$, then the mode of child B is equal to that of parent B.

Assuming that $n_1 = 4, n_2 = 5$, a schematic diagram of the chromosome crossover operation is shown in Figure 2.

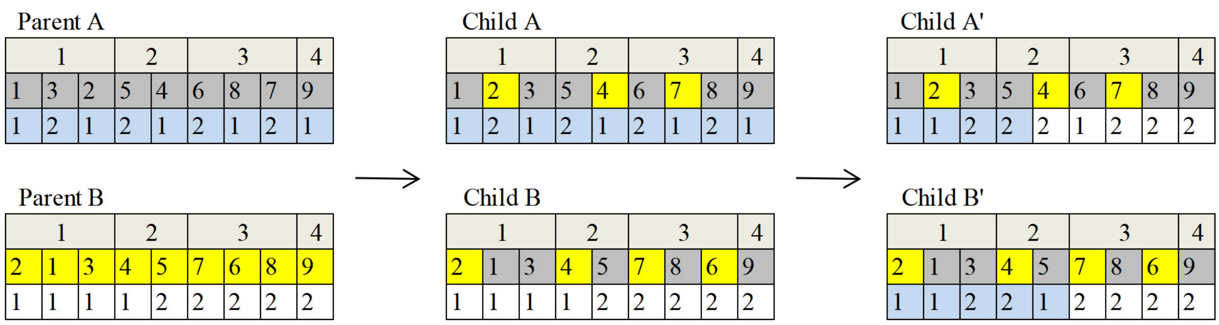

Figure 2. Schematic diagram of crossover operation.

Step 5. This step is concerned with mutation. For the variation of the activity list, on the premise of satisfying the hierarchy order, the mutation operation is carried out by the exchange mutation method, in which two mutation points are randomly selected from parents and genes are swapped at those two locations. However, activity modes do not change, as demonstrated in Figure 3. In Figure 3, activities 6 and 8 in level 3 are exchanged and mutated to obtain new individuals. For the variation in the mode list, an activity is randomly selected, and its activity mode is changed, as demonstrated in Figure 3.

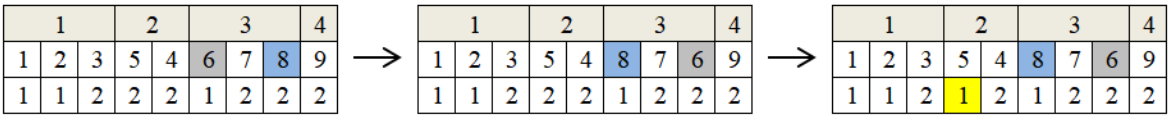

Figure 3. Variation operation diagram.

Step 6. Let the number of iterations be $\tau_1 = \tau_1 + 1$ and enter the next round of iterations until the maximum is reached.

4.2. DIWPSO

As a follower, the resource supplier must solve the problem of determining the resource allocation amount on each transportation path in each time period. Based on the characteristics of this problem, this section uses DIWPSO to solve the resource transportation policy.

4.2.1. Initial Code

In the existing research results, when solving the problem of resource transportation, the coding method mostly adopts the integer representation method, in which the customer (demand point) and the virtual distribution center are arranged together. In this study, the resource allocation quantity on the transportation path is adopted as the real number coding. Let $Y = v_{isk}(t)$ represent the position vector of each particle, initialize i particles as a population, and generate the ith particle with d-dimensional position vector Y_i; let its initial velocity $V_i = 0$, then the initial individual optimal is $P_i = Y_i^1$. The initial population is generated randomly so that it is distributed uniformly in the entire solution space as much as possible.

4.2.2. Fitness Function

The objective of resource supply is to minimize the running cost and transportation time, and the dimensions used are not the same. Therefore, the fitness function of the lower resource transportation model was constructed using the weighted aggregation method used in upper planning. Let β_1 and β_2 represent the weights of the two objective functions. The fitness function is shown in Equation (25), and the maximum fitness value is required.

$$Fitness(F_2) = \beta_1 \frac{z_c^{\max} - z_c}{z_c^{\max} - z_c^{\min}} + \beta_2 \frac{z_t^{\max} - z_t}{z_t^{\max} - z_t^{\min}} \tag{25}$$

4.2.3. Updating Policies

Step 1. Before the update operation, individuals are selected based on the elite strategy to increase the running speed of the algorithm. That is, the fitness of individuals generated in the population is first sorted from largest to smallest, and the top 50% of individuals are retained.

Step 2. DIWPSO is used for updating. Although the standard PSO has a fast convergence speed in the early stage, it is slow in later stages and easily converges locally. Therefore, this algorithm is improved from the perspective of the inertia weight. The inertia weight ω indicates the extent to which the original speed is retained; if ω is larger, the global search ability is stronger, and if ω is small, the local search ability is strong.

The update strategy is as follows: in the position vector, Equations (26) and (27) are used to update the particle velocity and position for the continuous factor $v_{isk}(t)$:

$$v_{id}(\tau + 1) = v_{id}(\tau)\omega + c_1 r_1(\tau)[p_{id}(\tau) - x_{id}(\tau)] + c_2 r_2(\tau)[g_d(\tau) - x_{id}(\tau)] \tag{26}$$

$$x_{id}(\tau+1) = x_{id}(\tau) + v_{id}(\tau+1) \tag{27}$$

$$\omega = \omega_{min} + (\omega_{max} - \omega_{min}) * e^{-\frac{\tau}{\tau_{max}}} + \sigma * betarnd(p,q) \tag{28}$$

where τ represents the current iteration number; τ_{max} represents the maximum number of iterations; ω_{max} represents the maximum inertia weight, which is set to 0.9; ω_{min} represents the minimum inertia weight, which is 0.1; σ is the inertia adjustment factor, which is 0.1; $p = 1, q = 3$; c_1 and c_2 are learning factors; r_1 and r_2 are uniform random numbers between [0,1]; $x_{id}(\tau)$ and $v_{id}(\tau)$ represent the position and velocity of the d dimension elements, respectively, of the i particle after the τ iteration; $p_{id}(\tau)$ represents the individual optimal position of the i particle in the d dimension; and $g_d(\tau)$ represents the global optimal position of all particles in the d dimension.

ω in the update strategy is an improved strategy for the dynamic adjustment of inertia weight [33], and the exponential function is used to control the change in inertia weight ω. With an increase in the number of iterations, $e^{-\tau/\tau_{max}}$ decreases nonlinearly; thus, ω can ensure the breadth of global search in the early stage and gradually decrease in the later stage to improve the ability of the local search and ensure its accuracy. Betarnd is a random number generator in MATLAB that can generate random numbers in line with the beta distribution. In addition, an inertia adjustment factor σ was added to control the deviation of the inertia weight, to make the adjustment more reasonable.

Step 3. Particle evaluation. To avoid generating infeasible particle positions and excessive velocities during the iteration, they must be within the corresponding limits.

$$v_{id} = \begin{cases} v_{\max}, v_{id} > v_{\max} \\ v_{\min}, v_{id} < v_{\min} \end{cases}, x_{id} = \begin{cases} x_{\max}, x_{id} > x_{\max} \\ x_{\min}, x_{id} < x_{\min} \end{cases} \tag{29}$$

Step 4. Particle adjustment. Since the fitness function is designed with the belief that larger is better, the individual $P_i(\tau)$ and the global $G(\tau)$ optimums are updated by calculating the fitness of the particles.

Step 4.1. For the individual optimum $P_i(\tau)$, if $Fitness[Y_i(\tau)] > Fitness[P_i(\tau-1)]$, update $P_i(\tau) = Y_i(\tau)$; otherwise, maintain the original value.

Step 4.2. For the global optimum $G(\tau)$, if $Fitness[P_i(\tau)] > Fitness[G(\tau-1)]$, update $G(\tau) = P_i(\tau)$; otherwise, maintain the original value, namely $G(\tau) = G(\tau-1)$.

Step 5. Premature particle determination. To judge the convergence degree of the particles, the population fitness variance [34] was introduced as the judgment mechanism of particle prematurity.

$$\delta^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{f_i - f_{avg}}{f}\right)^2 \tag{30}$$

$$f = \begin{cases} \max|f_i - f_{avg}|, \max|f_i - f_{avg}| > 1 \\ 1, \text{ otherwise} \end{cases} \tag{31}$$

where δ^2 is the variance of population fitness; the larger δ^2 is, the better the population diversity, and vice versa. f_i is the fitness of the i particle; f_{avg} is the average fitness of the population, and f is the normalization factor, which limits the size of δ^2.

A population fitness judgment threshold δ_T^2 is selected for premature judgment: when $\delta^2 < \delta_T^2$, the particle enters premature convergence. δ_T^2 is generally much smaller than the fitness variance of the initial population; $\delta_T^2 = 0.001$ is taken here.

Step 6. The mutation operation exists to improve the ability of the algorithm to jump out of premature convergence, ensure the diversity of the population, and keep the algorithm from falling into local convergence in the later stage to stop searching for a better solution. The mutation mechanism of the differential evolution algorithm is used to mutate the identified premature particles.

$$V_i(\tau+1) = x_{r1}(\tau) + \eta[x_{r2}(\tau) - x_{r3}(\tau)] \tag{32}$$

$r1, r2, r3 \in (1, 2, \ldots, N)$ is a random number and $r1 \neq r2 \neq r3 \neq i$, and η is a scaling factor adjusted by adaptive strategy:

$$\eta = \eta_{\max} - \tau(\eta_{\max} - \eta_{\min})/\tau_{\max} \tag{33}$$

where $\eta_{\max}$ and $\eta_{\min}$ are the upper and lower limits of the scaling factor, respectively.

4.3. Overall Process Framework of the Algorithm

The algorithm designed in this study is a two-layer GADS/DIWPSO hybrid algorithm. In the project scheduling problem, GADS is first used to initialize the feasible strategy and introduce it into lower-level planning. Then, DIWPSO is used to find the corresponding optimal solution of resource provisioning and the input to the upper planning is returned. Then, GADS is used to decode and generate the current optimal solution. This process is repeated until the upper optimal solution satisfies the stop condition. Through this dynamic interaction, the Stackelberg-Nash equilibrium strategy of the MRCPSP-MPMSP ensemble system is finally obtained.

The flow chart of this hybrid algorithm is shown in Figure 4, where the left part is the flow of GADS solving the upper-level project scheduling problem and the right part is the flow of DIWPSO solving the lower-level resource supply problem.

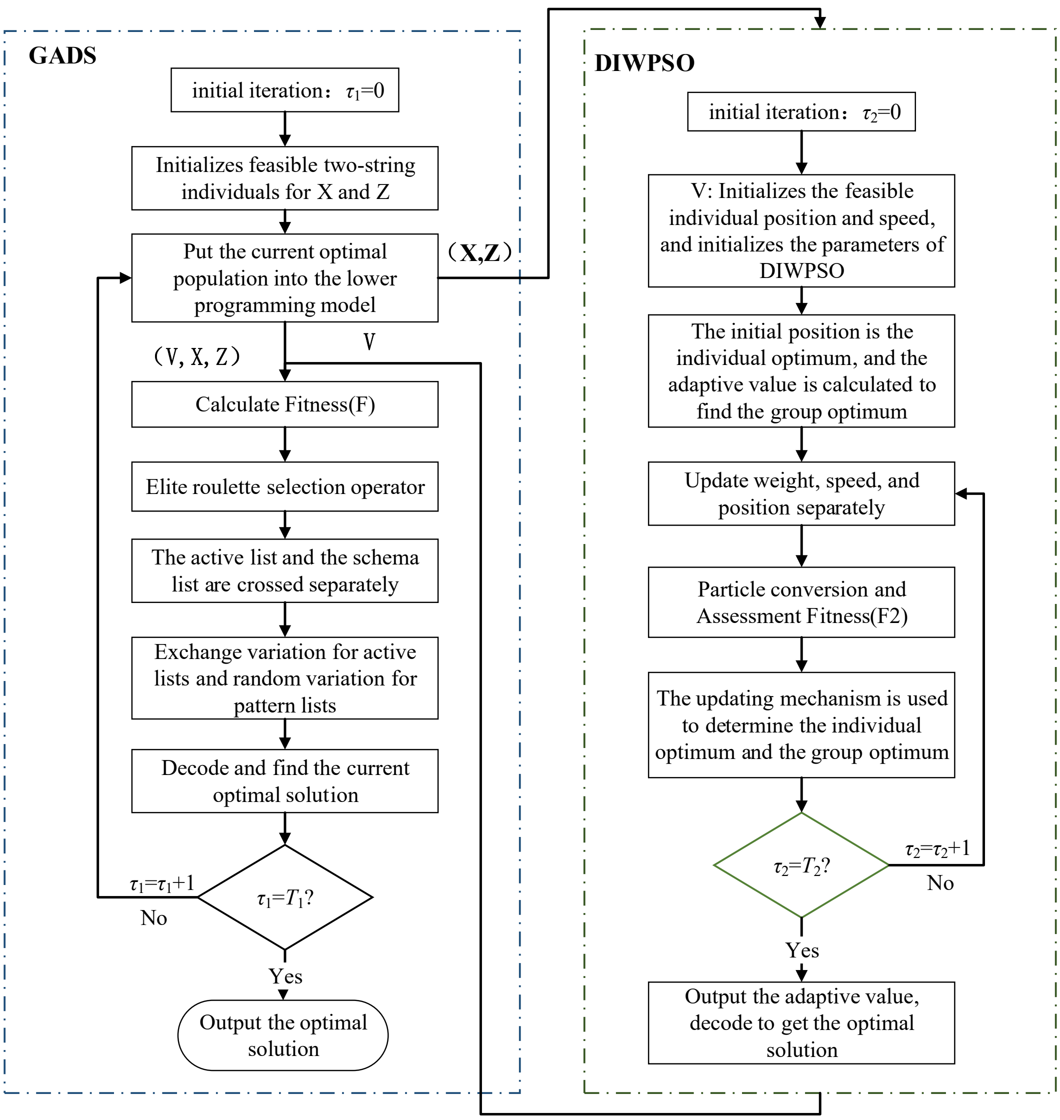

Figure 4. Flow chart of the hybrid algorithm.

5. Practical Application

The practical application and calculation test of a dam project verified the practicability and effectiveness of the proposed optimization method and provided decision-making guidance.

5.1. Project Description

In this study, a large hydropower project located in southeast China was considered as an application example. The project had a variety of hydraulic structures such as river dams, flood discharge structures, and hydraulic power generation systems. The river dam was a concrete double-curvature arch dam with a height of 610 m.

The concrete double-curvature arch dam construction project, which consists of 17 engineering activities, is the most important part. A flowchart is shown in Figure 5. Each activity has several optional modes, and each mode has a certain duration and resource demand. At the construction site, there are two large-scale resource demand points to allocate resources for each activity within the project, and the three resources required by the demand points are supplied by an external resource supplier with four resource supply points.

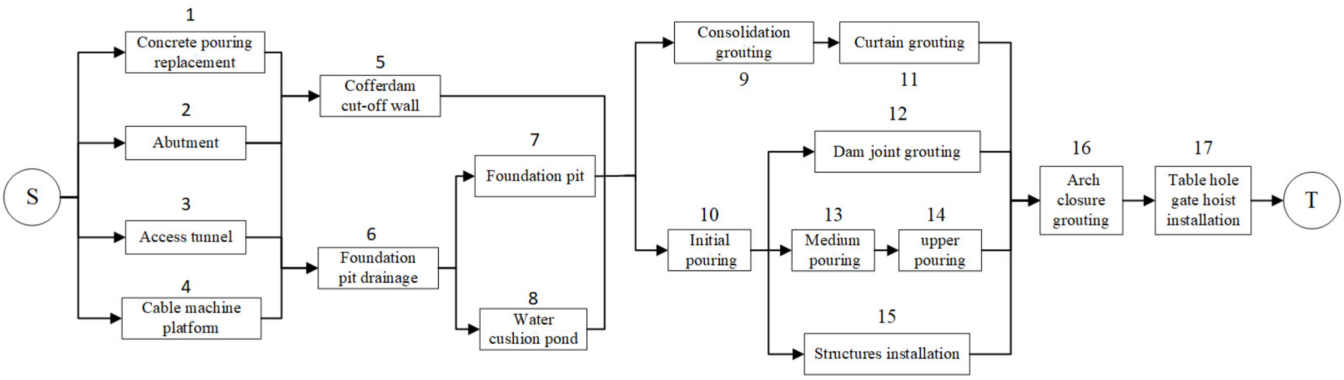

Figure 5. Construction flow chart of a concrete double-curvature arch dam.

5.2. Data Collection and Setting

5.2.1. Project Scheduling Data Processing

To collect relevant data for this practical application, we conducted interviews and surveys with relevant construction companies. The construction process of a concrete double-curvature arch dam can be divided into 17 activities, among which there are three types of common resources. Table 2 shows the activities in which each demand point is responsible for providing resources, and the other necessary data are shown in Table 3.

Table 2. Demand point-project activity mapping table.

Demand Points	*Rs*
1	1, 2, 3, 4, 6, 7, 8, 10
2	5, 9, 11, 12, 13, 14, 15, 16, 17

Table 3. Details on other parameters.

Resources	P_{ck}	I_{ck}	O_{ck}	r_k	R_k
$k = 1$	2.1	0.02	13.5	25	138
$k = 2$	3.6	0.03	21.6	18	105
$k = 3$	1.8	0.01	14.8	20	110

According to the preliminary data collected, the data of each activity in the project were processed in detail; specifically, uncertain variables were expressed in the form of random variables. The detailed processing data are shown in Table 4. In addition, the project planning period and available budget are $D = 52$ and $B = 8510$, respectively, the indirect cost of each time period is $c_0 = 5.8$, and the storage capacity of each period is $IC = 300$. The weights of the objective functions in the upper model were set to $\mu_1 = \mu_2 = 0.5$.

Table 4. Concrete double-curvature arch dam project activity details.

Activity	Mode	Resources r_{jmk}			Duration	Cost	Predecessors
j	m	$k = 1$	$k = 2$	$k = 3$	d_{jm}	c_{jm}	p_j
S	1	0	0	0	0	0	0
1	1	N(4.0,0.15^2)	N(4.1,0.20^2)	N(5.0,0.31^2)	N(3.1,0.15^2)	N(21.8,1.05^2)	S
	2	N(4.3,0.12^2)	N(4.6,0.20^2)	N(5.9,0.30^2)	N(2.8,0.15^2)	N(24.7,1.3^2)	S
2	1	N(12.8,0.40^2)	N(7.4,0.28^2)	N(6.7,0.21^2)	N(13.2,0.30^2)	N(84.6,2.0^2)	S
	2	N(13.6,0.45^2)	N(7.9,0.35^2)	N(7.1,0.20^2)	N(12.5,0.42^2)	N(87.8,1.8^2)	S
	3	N(14.8,0.30^2)	N(8.6,0.42^2)	N(7.6,0.13^2)	N(11.6,0.30^2)	N(91.5,1.7^2)	S
3	1	N(9.2,0.20^2)	N(8.2,0.28^2)	N(11.3,0.60^2)	N(5.8,0.32^2)	N(35.7,1.3^2)	S
	2	N(10.2,0.38^2)	N(9.1,0.35^2)	N(12.6,0.56^2)	N(5.2,0.20^2)	N(38.2,1.2^2)	S
4	1	N(7.3,0.10^2)	N(5.9,0.30^2)	N(7.3,0.23^2)	N(9.0,0.42^2)	N(29.5,1.5^2)	S
	2	N(8.0,0.15^2)	N(6.5,0.36^2)	N(8.1,0.35^2)	N(8.2,0.30^2)	N(32.3,1.0^2)	S
5	1	N(12.3,0.32^2)	N(7.8,0.45^2)	N(4.5,0.10^2)	N(9.3,0.25^2)	N(42.6,1.8^2)	1, 2, 3, 4
	2	N(13.1,0.32^2)	N(8.3,0.25^2)	N(4.8,0.20^2)	N(8.7,0.20^2)	N(46.5,1.7^2)	1, 2, 3, 4
6	1	N(3.7,0.08^2)	N(3.2,0.15^2)	N(8.7,0.20^2)	N(2.1,0.06^2)	N(15.7,1.08^2)	1, 2, 3, 4
7	1	N(7.0,0.35^2)	N(5.1,0.20^2)	N(10.7,0.40^2)	N(5.2,0.17^2)	N(38.0,1.0^2)	6
	2	N(7.5,0.16^2)	N(5.4,0.30^2)	N(11.6,0.40^2)	N(4.8,0.17^2)	N(39.2,1.3^2)	6
8	1	N(10.7,0.50^2)	N(8.6,0.32^2)	N(6.8,0.10^2)	N(4.0,0.07^2)	N(43.0,1.2^2)	6
	2	N(12.0,0.40^2)	N(9.8,0.42^2)	N(7.3,0.18^2)	N(3.5,0.10^2)	N(45.7,1.6^2)	6
9	1	N(6.8,0.20^2)	N(8.5,0.38^2)	N(9.1,0.28^2)	N(8.4,0.22^2)	N(62.5,1.7^2)	5, 7, 8
	2	N(7.2,0.15^2)	N(8.9,0.41^2)	N(9.7,0.30^2)	N(8.0,0.16^2)	N(65.0,1.7^2)	5, 7, 8
10	1	N(14.5,0.37^2)	N(8.4,0.20^2)	N(4.1,0.28^2)	N(4.3,0.11^2)	N(55.8,1.14^2)	5, 7, 8
	2	N(15.5,0.60^2)	N(9.0,0.18^2)	N(4.4,0.10^2)	N(4.0,0.06^2)	N(57.5,1.0^2)	5, 7, 8
11	1	N(6.0,0.18^2)	N(8.2,0.20^2)	N(9.3,0.28^2)	N(8.0,0.18^2)	N(48.3,1.4^2)	9
	2	N(6.4,0.16^2)	N(8.7,0.40^2)	N(10.0,0.30^2)	N(7.5,0.18^2)	N(50.8,1.6^2)	9
12	1	N(4.6,0.20^2)	N(3.4,0.15^2)	N(6.0,0.15^2)	N(15.5,0.26^2)	N(51.4,0.9^2)	10
	2	N(4.7,0.15^2)	N(3.5,0.10^2)	N(6.4,0.30^2)	N(15.0,0.37^2)	N(53.5,1.0^2)	10
	3	N(5.0,0.10^2)	N(3.7,0.12^2)	N(6.8,0.25^2)	N(14.2,0.35^2)	N(55.2,1.0^2)	10
13	1	N(10.1,0.20^2)	N(4.9,0.18^2)	N(3.5,0.10^2)	N(9.3,0.10^2)	N(72.4,1.2^2)	10
	2	N(10.8,0.5^2)	N(5.1,0.20^2)	N(3.9,0.10^2)	N(8.8,0.15^2)	N(74.8,1.8^2)	10
14	1	N(8.9,0.25^2)	N(4.9,0.10^2)	N(3.3,0.12^2)	N(3.0,0.05^2)	N(41.8,0.8^2)	13
	2	N(10.2,0.30^2)	N(6.0,0.30^2)	N(3.8,0.10^2)	N(2.6,0.06^2)	N(43.0,1.0^2)	1, 2, 3, 4
15	1	N(5.0,0.10^2)	N(2.9,0.05^2)	N(3.5,0.10^2)	N(2.8,0.05^2)	N(25.6,0.8$^{2)}$	10
	2	N(6.0,0.23^2)	N(3.5,0.20^2)	N(4.3,0.13^2)	N(2.5,0.06^2)	N(27.4,1.1^2)	10
16	1	N(9.2,0.25^2)	N(7.5,0.30^2)	N(8.7,0.32^2)	N(3.0,0.07^2)	N(36.2,1.2^2)	11, 12, 14, 15
	2	N(9.8,0.25^2)	N(8.0,0.30^2)	N(9.3,0.32^2)	N(2.8,0.07^2)	N(37.2,1.0^2)	11, 12, 14, 15
17	1	N(7.2,0.18^2)	N(5.3,0.10^2)	N(2.4,0.08^2)	N(4.2,0.08^2)	N(36.7,1.2^2)	16
	2	N(8.0,0.20^2)	N(5.9,0.20^2)	N(2.7,0.05^2)	N(3.8,0.05^2)	N(38.1,1.0^2)	16
T	1	0	0	0	0	0	17

5.2.2. Resource Supply Data Processing

All detailed engineering data on the resource supply were obtained from a hydropower project construction company in the watershed project. In a transportation network, the transportation of various resources is accompanied by the entire construction cycle. The entire transportation network can be divided into four supply and two demand points, and three shared resources can be transported from any supply to any demand point.

The maximum resource capacities of the four supply points were 723.4×10^4 m^3, 581.7×10^4 m^3, 528.3×10^4 m^3, and 790.2×10^4 m^3. The maximum resource capacity of the two demand points was 15×10^4 m^3. The project used dump trucks to transport three resources along different routes between different supply and demand points. The unit transport cost and time data for each resource are presented in Table 5.

Table 5. Unit transportation cost and time of resources.

Cost Parameters		Resource Types			Time Parameters		Resource Types		
		k_1	k_2	k_3			k_1	k_2	k_3
c_{isk}	c_{11k}	N(5.20,3.1)	N(6.00,4.2)	N(5.82,3.8)	t_{isk}	t_{11k}	N(0.34,0.21)	N(0.37,0.22)	N(0.32,0.18)
	c_{21k}	N(3.25,2.1)	N(3.66,2.2)	N(3.72,1.8)		t_{21k}	N(0.25,0.1)	N(0.26,0.15)	N(0.22,0.18)
	c_{31k}	N(4.23,1.7)	N(4.43,2.1)	N(4.59,2.4)		t_{31k}	N(0.23,0.12)	N(0.21,0.1)	N(0.19,0.13)
	c_{41k}	N(6.12,3.8)	N(6.44,4.2)	N(6.40,4.0)		t_{41k}	N(0.42,0.21)	N(0.44,0.32)	N(0.40,0.28)
	c_{12k}	N(5.57,2.8)	N(5.41,3.0)	N(5.77,4.2)		t_{12k}	N(0.27,0.11)	N(0.30,0.22)	N(0.24,0.12)
	c_{22k}	N(6.21,4.1)	N(6.33,4.2)	N(6.50,3.8)		t_{22k}	N(0.21,0.08)	N(0.23,0.12)	N(0.20,0.12)
	c_{32k}	N(5.60,3.1)	N(5.41,3.0)	N(5.77,4.2)		t_{32k}	N(0.28,0.37)	N(0.41,0.22)	N(0.37,0.24)
	c_{42k}	N(3.63,2.1)	N(3.84,2.2)	N(4.00,2.0)		t_{42k}	N(0.33,0.20)	N(0.34,0.22)	N(0.29,0.16)

5.3. Selection of Algorithm Parameters

These parameters are controllable factors that affect the convergence, effectiveness, and efficiency of the algorithm. To determine the most appropriate parameters, preliminary experiments and comparisons must be performed under different parameter settings. Herein, a fuzzy logic controller is used to automatically adjust the mutation rate of each generation, and the initial mutation rate is set as $p_m(0) = 0.1$. The inertia weight is adjusted with iteration according to equation (28), and previous studies [35] reveal that $\omega(1) = 0.9$ and $\omega(T) = 0.1$ are the most appropriate. The Taguchi method [36] was used to adjust the other parameters. Finally, the corresponding algorithm parameters were selected, as listed in Table 6.

Table 6. Hybrid algorithm parameter setting.

Parameters	GADS				DIWPSO								
	L_1	T_1	p_c	$p_m(0)$	L_2	T_2	c_1	c_2	σ	η_{max}	η_{min}	$\omega(1)$	$\omega(T_2)$
Values	100	300	0.7	0.1	100	200	2	2	0.1	0.6	0.2	0.9	0.1

5.4. Calculation Results

The designed hybrid algorithm was run in MATLAB(R2018b) on the collected data. After running the program 30 times, an optimal solution was obtained. The total project scheduling time and cost were 48.9 and 8326.54, respectively. The upper planning MRCPSP calculation results are listed in Table 7, showing the start-end time and mode selection of each activity; the corresponding Gantt chart is shown in Figure 6. The calculation result of the MPSCP of the lower planning is shown in Table 8, which defines the transportation volume of the three resources on each transportation route in each time period. The total transportation cost and time were 1144.38 and 13.73, respectively. The convergence iteration is 146 times, and the computation time is 956.3 s.

Table 7. MRCPSP calculation results.

Result	Project Activities																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
ST_j	0.0	0.0	8.1	0.0	13.2	13.2	18.8	15.3	24.1	24.1	32.1	28.1	30.9	39.7	28.1	42.3	45.1
LT_j	3.1	13.2	13.2	8.1	22.7	15.3	24.1	18.8	32.1	28.1	40.1	42.2	39.7	42.3	30.9	45.1	48.9
m	1	1	2	2	1	1	1	2	2	2	1	3	2	2	1	2	2

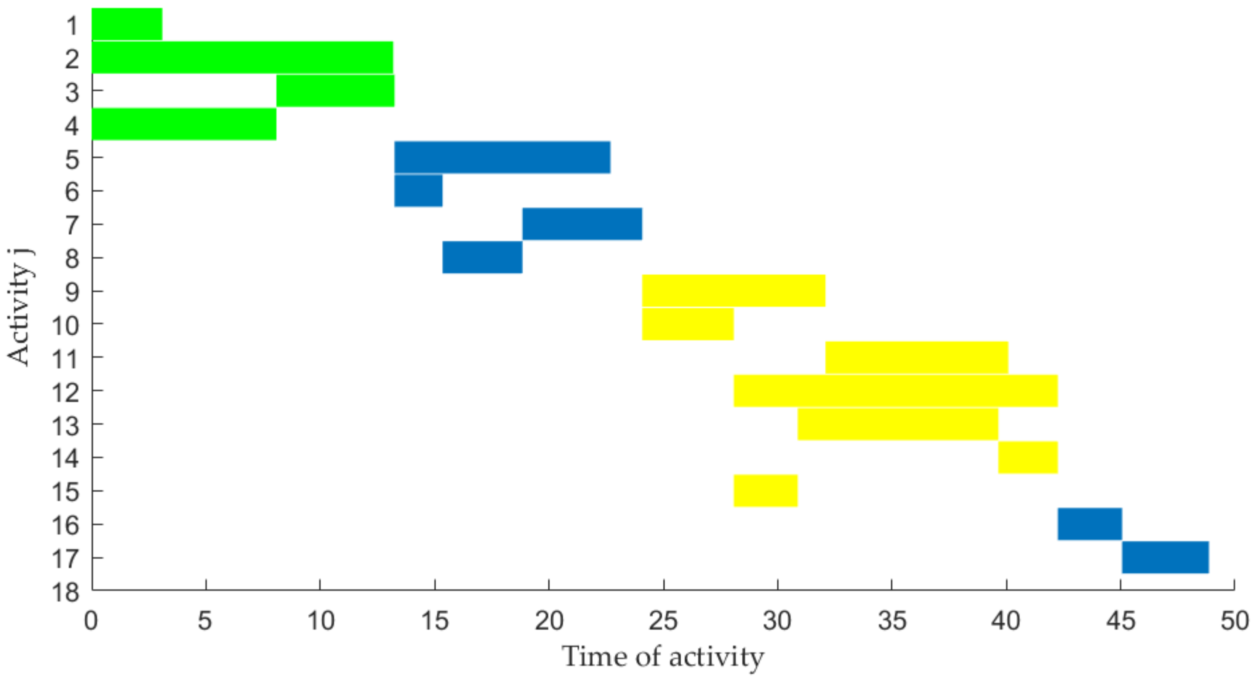

Figure 6. Gantt chart of MRCPSP.

Table 8. Resource transportation decision.

t		1	7	13	19	20	21	26	27	28	33	34	39	Others
	k_1	38.04	36.82	14.76	7.39									0
v_{11}	k_2	26.50	26.92	13.64		4.86								0
	k_3	29.00	30.20	30.60			13.63							0
	k_1	30.21	29.24	28.01	29.16			30.49						0
v_{21}	k_2	21.05	21.38	23.20		20.11			18.88					0
	k_3	23.03	23.98	24.30			22.23			4.76				0
	k_1	27.44	26.56	25.44	26.49			15.76						0
v_{31}	k_2	19.11	19.42	21.07		18.26								0
	k_3	20.92	21.78	22.07			20.19							0
	k_1	41.04	39.72											0
v_{41}	k_2	28.59	29.04											0
	k_3	31.29	32.58	2.74										0
	k_1			20.51	29.33			38.39			36.73		37.99	0
v_{12}	k_2			15.58		20.46			28.40		28.50		28.46	0
	k_3						14.37			30.60		27.65	28.00	0
	k_1										29.17		30.18	0
v_{22}	k_2								3.68		22.63		22.60	0
	k_3									19.55		21.96	22.24	0
	k_1							11.94			26.50		27.41	0
v_{32}	k_2								20.49		20.56		20.53	0
	k_3									22.07		19.95	20.16	0
	k_1			38.05	39.62			41.42			39.63		40.99	0
v_{42}	k_2			31.52		27.31			30.64		30.75		30.70	0
	k_3			30.28			30.20			33.02		29.84	30.16	0

5.5. Analysis and Discussion

5.5.1. Weight Analysis

Different weight settings (i.e.,μ_1 and μ_2) represent different combinations of preferences for decision-makers. To further understand the influence of the weight setting in upper-level planning, a sensitivity analysis was carried out, and the corresponding results are presented in Table 9. Different weight settings led to different results in the upper and lower models, which indicates that the decisions of the two levels are greatly influenced by the upper weight settings and are closely related to each other.

Table 9. Weight sensitivity analysis.

Cases	Weight Values		Objective Function Values			
	μ_1	μ_2	F_t	F_c	Z_c	Z_t
case 1	0.7	0.3	47.35	8347.16	1150.97	13.51
case 2	0.6	0.4	48.04	8335.30	1147.75	13.58
case 3	0.5	0.5	48.86	8326.54	1144.38	13.73
case 4	0.4	0.6	49.40	8320.65	1141.21	13.85
case 5	0.3	0.7	50.36	8315.23	1138.83	13.97

5.5.2. Model Comparison

To verify the effectiveness of the model and the superiority of obtaining the optimal and satisfactory solution, the game model was compared with the single-layer model of the MRCPSP and MPSCP, which ignores the conflict.

To establish the corresponding single-layer model, project scheduling and resource supply were combined into a separate optimization problem. The objective function is the duration and cost of project scheduling, F_t and F_c, the decision variables are also (v, x), and the constraints include all the constraints in the upper planning. To calculate the comparative rationality of the results, the GADS proposed in the upper planning was also applied to the single-layer model and run in MATLAB(R2018b). Subsequently, the decision results are substituted into Z_c and Z_t to calculate the function value, and the objective function value of the single-layer model in the ideal state is obtained.

However, in practice, the lower-level planning MPSCP also has its own optimization objectives and constraints, and there are decision conflicts between the construction department and the resource supplier. Therefore, the ideal optimal solution obtained by the single-level planning model may not be a satisfactory solution for the MPSCP and will usually deviate. Therefore, the results obtained using the ideal single-layer model must be modified as follows:

In the first step, the decision result of the ideal single-layer model was used as the decision result of the upper MRCPSP. In the second step, considering the sequence of decisions, the decision results of the MRCPSP were substituted into the MPSCP to obtain the optimal transportation decision under this situation, namely, the modified solution. In the third step, the result of the transportation decision is substituted into the objective function of the MRCPSP to obtain the objective function value in this case.

In the dynamic game model, considering the hierarchical decision structure and the existence of decision conflicts, the above correction method is repeatedly used to obtain a satisfactory Stackelberg-Nash equilibrium solution. The corresponding calculation results are listed in Table 10, and Table 11 lists the comparison results of the algorithms.

Table 10. Selection of algorithm parameters.

Algorithms	Parameters											
	L_1	T_1	p_c	$p_m(0)$	L_2	T_2	c_1	c_2	σ	$\eta_{\max}$	$\eta_{\min}$	$\omega(1)$	$\omega(T_2)$
GA/PSO	100	300	0.75	0.15	100	200	1.8	2	×	×	×	0.9	0.1
GADS/PSO	100	300	0.7	0.1	100	200	1.8	2	×	×	×	0.9	0.1
GA/DIWPSO	100	300	0.75	0.15	100	200	2	2	0.1	0.6	0.2	0.9	0.1
GADS/DIWPSO	100	300	0.7	0.1	100	200	2	2	0.1	0.6	0.2	0.9	0.1

Figure 7 demonstrates the iterative process of the algorithm. The results of algorithm comparison reveal that: ① All four algorithms can obtain the optimal fitness in 200 iterations, and the hybrid GADS/DIWPSO algorithm has a higher fitness. ② The computation time and convergence speed of the four algorithms are acceptable, among which GADS/DIWPSO hybrid algorithm is faster than GADS/PSO but slightly slower than GA/DIWPSO and GA/PSO. ③ The GADS/DIWPSO hybrid algorithm has better standard deviation corresponding to fitness, convergence iteration times, and computation time than other algorithms, showing stable performance, which also reveals that the algorithm can effectively avoid infeasible solutions and reduce the probability of premature convergence. Therefore, the GADS/DIWPSO hybrid algorithm proposed in this study performs better than other algorithms in an acceptable computation time.

Table 11. Algorithm comparison results.

Algorithms		Fitness			Convergence Iteration Number			Computation Time		
		Best	Average	Standard Deviation	Best	Average	Standard Deviation	Best	Average	Standard Deviation
GA/PSO	F F_2	0.957 0.905	0.940 0.890	0.0082 0.0065	126	142	7.0	927.6	961.5	14.8
GADS/PSO	F F_2	0.968 0.926	0.961 0.916	0.0043 0.0040	149	157	4.2	968.2	990.0	10.3
GA/DIWPSO	F F_2	0.963 0.935	0.954 0.929	0.0065 0.0032	122	131	5.3	912.5	940.4	13.4
GADS/DIWPSO	F F_2	0.976 0.947	0.971 0.943	0.0035 0.0020	146	152	3.8	956.3	975.6	9.5

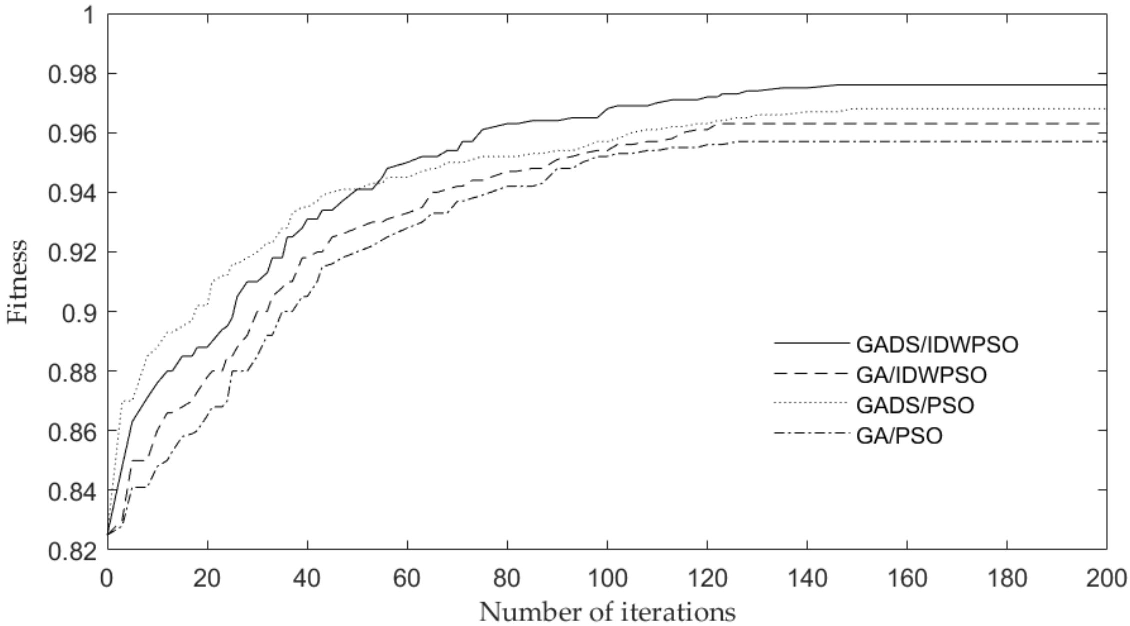

Figure 7. Algorithm iteration process.

5.6. Management Suggestions

Through the application of practical cases, some management suggestions are proposed for relevant departments from the perspective of the engineering supply chain:

① When making the project schedule, the decision maker of the engineering project shall ensure that the project schedule and resource supply are within a reasonable range so that the construction schedule based on materials, equipment, and labor force can meet the expected requirements. At the same time, it must be considered that too much or too little resource supply cannot ensure the schedule advancement, because the process sequence and intermittent time in the construction process of the project determine that the actual construction progress cannot violate the internal law of the project. Once the construction progress based on the process is exceeded, quality problems are likely to occur.

② The engineering supply chain generally involves multiple stakeholders such as owners, contractors, resource suppliers, and transportation agents. Different stakeholders are responsible for various professional tasks. These tasks are often interrelated, and if considered separately and while ignoring the conflicts of various stakeholders, they can lead to suboptimal solutions, which in turn can cause economic losses, construction delays, and other problems. Therefore, in the actual implementation of engineering projects, inherent conflicts and complex interactions must be identified and resolved.

③ In engineering practice, project managers must consider all kinds of resources, such as the labor force, materials, and equipment as a whole. The disharmony between any type of resource and other resources may cause resource redundancy or project stagnation at a certain link in an engineering project.

④ Modeling the decision-making process helps to understand the complexity and conflicts involved in the supply chain and then conducts quantitative analysis to determine a satisfactory equilibrium strategy. For example, the new Stackelberg dynamic game model proposed for the MRCPSP-MPSCP integrated system is more suitable than the corresponding single-layer model. In addition, the preference setting of the multi-objective function is important, and different preference combinations lead to different results.

6. Conclusions and Future Research

This study investigated the integration of multimode project scheduling and resource supply in an engineering supply chain. Resource constraint is not only a constraint condition of the engineering supply chain, but is often a separate optimization problem. Therefore, integrating resource supply into project scheduling is an MRCPSP-MPSCP integrated system with multi-agent decision-making characteristics and a hierarchical decision-making structure. Resolving conflicts in this integrated system helps ensure that the project runs successfully at an acceptable cost and is completed on time. On this basis, a Stackelberg dynamic game model was established, and a two-level multi-objective programming method was designed to further solve internal conflicts. Subsequently, a two-layer GADS/DIWPSO hybrid algorithm with an interactive evolution mechanism was proposed to solve the new Stackelberg model, and a satisfactory Stackelberg-Nash equilibrium solution was determined through a repeated dynamic interaction process. This provides theoretical significance for solving related problems of engineering supply chain.

In the context of the global impact of COVID-19, coordinated optimization and sustainable operation of the engineering supply chain play an important role in the recovery of the industrial economy. This study provides a theoretical basis and algorithm support for how engineering and construction departments and resource suppliers in the supply chain promote the optimization of overall benefits. For the engineering construction department, considering the limitation of resource supply, more thought is devoted to the project scheduling problem to ensure the overall operation of the project. For resource suppliers, considering the characteristics of master-slave decision-making, this study provides a reference for the formulation of a resource transportation strategy, and finally promotes mutual benefit on both sides to achieve better cooperation results.

After discussion and analysis, it can be discovered that in the engineering supply chain, the multi-period resource supply problem does have an impact on the project scheduling. Therefore, the dynamic game model for the MRCPSP-MPSCP integrated system is more realistic, and the proposed two-level multi-objective programming method and GADS/DIWPSO hybrid algorithm can solve the conflicts between stakeholders, and finally realize the Stackelberg-Nash equilibrium strategy. In conclusion, when solving similar problems, researchers should start from reality, fully consider the conflicts of interest among participants, and make reasonable assumptions. Only in this way can a better decision plan be generated.

However, there are still some limitations in this study: ① he scheduling problem of multiple projects is not considered; ② the mixed transportation of multi-type vehicles is not considered in terms of resource transportation; and ③ more participants can be considered in a large engineering supply chain, such as material manufacturers and transportation agents. These limitations will form the basis for future research.

Author Contributions: Conceptualization, C.F.; methodology, C.F. and S.H.; data analysis, C.F., S.H. and Y.M.; writing—original draft preparation, C.F. and S.H.; writing—review and editing, Z.L.; supervision, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (No. 71702167 and No. 72202056).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All processed data used in the study have been shown in the article.

Acknowledgments: The authors gratefully acknowledge the funding and support provided by the National Natural Science Foundation of China (No. 71702167 and No. 72202056) and are grateful to the editors and the reviewers for their insightful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, C.-L.; Hall, N.G. Work Package Sizing and Project Performance. *Oper. Res.* **2019**, *67*, 123–142. [CrossRef]
2. Herroelen, W. Project Scheduling-Theory and Practice. *Prod. Oper. Manag.* **2005**, *14*, 413–432. [CrossRef]
3. Browning, T.R.; Yassine, A.A. Resource-constrained multi-project scheduling: Priority rule performance revisited. *Int. J. Prod. Econ.* **2010**, *126*, 212–228. [CrossRef]
4. Guoshan, L.; Min, W.; Zhuanxia, Z. A Bi-level Programming Problem Based on Time-window Delay for Resource-Constrained Project Scheduling. *Oper. Res. Manag. Sci.* **2021**, *30*, 6–12+27.
5. Kim, K.; Yun, Y.; Yoon, J.; Gen, M.; Yamazaki, G. Hybrid genetic algorithm with adaptive abilities for resource-constrained multiple project scheduling. *Comput. Ind.* **2005**, *56*, 143–160. [CrossRef]
6. Cheng, M.-Y.; Tran, D.-H. Opposition-based Multiple Objective Differential Evolution (OMODE) for optimizing work shift schedules. *Autom. Constr.* **2015**, *55*, 1–14. [CrossRef]
7. Demeulemeester, E.L.; Herroelen, W.S. An efficient optimal solution procedure for the preemptive resource-constrained project scheduling problem. *Eur. J. Oper. Res.* **1996**, *90*, 334–348. [CrossRef]
8. Vanhoucke, M.; Debels, D. The impact of various activity assumptions on the lead time and resource utilization of resource-constrained projects. *Comput. Ind. Eng.* **2008**, *54*, 140–154. [CrossRef]
9. Isah, M.A.; Kim, B.-S. Integrating Schedule Risk Analysis with Multi-Skilled Resource Scheduling to Improve Resource-Constrained Project Scheduling Problems. *Appl. Sci.* **2021**, *11*, 650. [CrossRef]
10. Schwarze, J. Activity networks: Project planning and control by network models. *Eur. J. Oper. Res.* **1979**, *3*, 167–168. [CrossRef]
11. Varma, V.A.; Uzsoy, R.; Pekny, J.; Blau, G. Lagrangian heuristics for scheduling new product development projects in the pharmaceutical industry. *J. Heuristics* **2007**, *13*, 403–433. [CrossRef]
12. Zhu, G.; Bard, J.F.; Yu, G. A Branch-and-Cut Procedure for the Multimode Resource-Constrained Project-Scheduling Problem. *INFORMS J. Comput.* **2006**, *18*, 377–390. [CrossRef]
13. Bellenguez-Morineau, O.; Neron, E. A Branch-and-Bound method for solving Multi-Skill Project Scheduling Problem. *RAIRO Rech. Opérationnelle* **2007**, *41*, 155–170. [CrossRef]
14. Sakawa, M.; Nishizaki, I. Interactive fuzzy programming for two-level nonconvex programming problems with fuzzy parameters through genetic algorithms. *Fuzzy Sets Syst.* **2002**, *127*, 185–197. [CrossRef]
15. Sarker, B.R.; Egbelu, P.J.; Liao, T.W.; Yu, J. Planning and design models for construction industry: A critical survey. *Autom. Constr.* **2012**, *22*, 123–134. [CrossRef]
16. Fang, X.; Zhe, X.; Jing, Y. Bi-objective optimization for the project scheduling problem with variable resource availability. *Syst. Eng. Theory Pract.* **2016**, *36*, 674–683.
17. Zhao, X.; Lü, X.; Mei, Q.; Feng, W. Project scheduling problem constrained by flexible resource with capability difference. *Ji Suan Ji Gong Cheng Yu Ying Yong* **2012**, *48*, 231–237. [CrossRef]
18. Schwindt, C.; Trautmann, N. Batch scheduling in process industries: An application of resource–constrained project scheduling. *OR Spektrum* **2000**, *22*, 501–524. [CrossRef]
19. Liu, S.-S.; Huang, H.-Y.; Risna Dyah Kumala, N. Two-Stage Optimization Model for Life Cycle Maintenance Scheduling of Bridge Infrastructure. *Appl. Sci.* **2020**, *10*, 8887. [CrossRef]
20. Asta, S.; Karapetyan, D.; Kheiri, A.; Özcan, E.; Parkes, A.J. Combining Monte-Carlo and hyper-heuristic methods for the multi-mode resource-constrained multi-project scheduling problem. *Inf. Sci.* **2016**, *373*, 476–498. [CrossRef]
21. Fang, X.; Hongbo, L.; Qingguo, B. Stochastic multi-mode resource constrained project scheduling. *Chin. J. Manag. Sci.* **2020**, 1–13. [CrossRef]
22. Peteghem, V.V.; Vanhoucke, M. A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. *Eur. J. Oper. Res.* **2010**, *201*, 409–418. [CrossRef]
23. Nusen, P.; Boonyung, W.; Nusen, S.; Panuwatwanich, K.; Champrasert, P.; Kaewmoracharoen, M. Construction Planning and Scheduling of a Renovation Project Using BIM-Based Multi-Objective Genetic Algorithm. *Appl. Sci.* **2021**, *11*, 4716. [CrossRef]
24. Xie, L.; Chen, Y.; Chang, R. Scheduling Optimization of Prefabricated Construction Projects by Genetic Algorithm. *Appl. Sci.* **2021**, *11*, 5531. [CrossRef]

25. Kuo, R.J.; Huang, C.C. Application of particle swarm optimization algorithm for solving bi-level linear programming problem. *Comput. Math. Appl.* **2009**, *58*, 678–685. [CrossRef]
26. Kuo, R.J.; Lee, Y.H.; Zulvia, F.E.; Tien, F.C. Solving bi-level linear programming problem through hybrid of immune genetic algorithm and particle swarm optimization algorithm. *Appl. Math. Comput.* **2015**, *266*, 1013–1026. [CrossRef]
27. Amirtaheri, O.; Zandieh, M.; Dorri, B.; Motameni, A.R. A bi-level programming approach for production-distribution supply chain problem. *Comput. Ind. Eng.* **2017**, *110*, 527–537. [CrossRef]
28. Singh, V.; Ganapathy, L.; Pundir, A.K. An Improved Genetic Algorithm for Solving Multi Depot Vehicle Routing Problems. *Int. J. Inf. Syst. Supply Chain. Manag.* **2019**, *12*, 1–26. [CrossRef]
29. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 1944, pp. 1942–1948.
30. Bidot, J.; Vidal, T.; Laborie, P.; Beck, J.C. A theoretic and practical framework for scheduling in a stochastic environment. *J. Sched.* **2008**, *12*, 315–344. [CrossRef]
31. Nishizaki, I.; Sakawa, M. *Cooperative and Noncooperative Multi-Level Programming*; Nishizaki, I., Sakawa, M., Eds.; Springer: New York, NY, USA, 2009; Volume 48.
32. Feng, C.; Ni, T.; Li, Z.; Cai, J.; Ma, Y. Conflict resolution towards an integrated project scheduling and material ordering system in a large-scale construction project. *Appl. Comput. Math.* **2019**, *18*, 202–217.
33. Tangqing, H.; Xuxiu, Z.; Xiaoyue, C. A hybrid particle swarm optimization with dynamic adjustment of inertial weight. *Electron. Opt. Control.* **2020**, *27*, 16–21.
34. Jianping, L. Hybrid particle swarm optimization algorithm based on chaos and differential evolution. *Comput. Simul.* **2012**, *29*, 208–212.
35. Feng, C.; Xu, J.; Yang, X.; Zeng, Z. Stackelberg-Nash Equilibrium for Integrated Gravelly Soil Excavation-Transportation-Distribution System in a Large-Scale Hydropower Construction Project. *J. Comput. Civ. Eng.* **2016**, *30*, 4016024. [CrossRef]
36. Taguchi, G.C.S.; Wu, Y. *Taguchi's Quality Engineering Handbook*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2005.