Article

Linear Project-Scheduling Optimization Considering a Reverse Construction Scenario

Ze Yu 1, Chuxin Wang 2, Yuanyuan Zhao 2, Zhiyuan Hu 2 and Yuanjie Tang 3,*

1 DUT School of Economics and Management, Dalian University of Technology, Dalian 116024, China; linglan_619@126.com
2 School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China; 20120901@bjtu.edu.cn (C.W.); 22121019@bjtu.edu.cn (Y.Z.); 20114074@bjtu.edu.cn (Z.H.)
3 Key Laboratory of Transport Industry, Big Data Application Technologies for Comprehensive Transport, Beijing Jiaotong University, Beijing 100044, China
* Correspondence: tangyj@bjtu.edu.cn

Abstract: The linear scheduling method (LSM) for optimization in linear projects has been the focus of numerous academic studies over the years. However, research on incorporating reverse construction activities and other practical scenarios, such as flexible activity–section–crew assignment demands, in linear project-scheduling optimization remains insufficient. This has further spurred research on practical scenario-based linear project-scheduling optimization. We performed an analysis of a description method system within the LSM framework, outlining the spatiotemporal logical relationship in scenarios involving reverse activities. A scheduling optimization model incorporating a flexible constraint system, capable of handling multisection–multicrew, multicrew–multimode, and multicrew–multiconstruction direction scenarios is developed to enhance the practicability of linear project-scheduling optimization. Additionally, an advanced grey wolf optimization (GWO) algorithm is devised and validated through a real-world case study. The case study provides insights into the impact of reverse construction activities on the linear project-scheduling problem, specifically in the dimension of project duration.

Keywords: railway infrastructure; linear scheduling method; scheduling optimization; construction project management; grey wolf optimizer

1. Introduction

Throughout the lifecycle of a construction project, several complex engineering and management challenges can arise [1], particularly in large-size infrastructure, such as railways and expressways, owing to their extensive construction period. Among the myriad issues these projects face, scheduling optimization is a significant concern [1,2]. This problem has attracted substantial attention from the academic community in recent years and is considered one of the most formidable challenges in construction project management [3,4]. Given this, the scheduling optimization of construction projects is particularly important. Typically, scheduling forms the backbone of project management decision making [5] and ensures successful management, notably in the area of resource management [4,6]. The fundamental aspect of construction project management can be viewed as the organization and distribution of activities and resources [7].

Construction projects can be classified into linear–repetitive and nonrepetitive projects [8]. Linear projects are characterized by numerous repetitive activities with the same type [9] and are continuously pushed forward on the mileage line during construction [10,11]. Activities of nonrepetitive projects, such as hospitals and libraries, do not have obvious repetitive characteristics. Detailed comparison for such two types of projects is referred to in [12,13]. In such a context, resource wastage due to construction delays must be eliminated by ensuring as much continuity of resources as possible [10]. Discontinuous
resource allocation typically leads to decreased staff morale, reduced productivity, increased costs, and extended construction periods [14–16]. These negative effects are magnified in the linear project scenario owing to the continuous repetition of construction. Conversely, the continuous use of resources enhances the learning curve utility, minimizes resource idle time, and improves productivity and cost efficiency [17–19]. Additionally, it enhances the accuracy of project duration and resource usage predictions [20]. In addition, the linear project’s repetitive as well as large-span characteristics inevitably lead to the demand for multi-crew strategies, construction continuity, and various special constraints such as spatiotemporal constraints, crew assignment constraints, construction sequence constraints, etc., which are important distinctions from nonrepetitive construction scheduling. These constraints are difficult to describe by traditional network scheduling methods. Overall, owing to the repetitive nature of linear activities, traditional network scheduling methods prove ineffective. Over the past decade, numerous academic studies have highlighted defects in these methods when applied to linear projects. Among these, the two most critical issues are that they fail to guarantee resource continuity and lack the representation of spatial information. Traditional “activity-based” scheduling methods, such as network planning, fall short in these areas. Consequently, the linear scheduling method (LSM), a new spatial location-based [16] and resource-driven [21,22] scheduling technique, tailored for linear projects, has been garnering increased attention from both academia and industry. A comparison between LSM and CPM is referred to in [23].

In recent years, the scheduling optimization of linear projects has been extensively researched and discussed. However, certain research gaps still exist, especially research on incorporating reverse construction activities and other practical scenarios, such as flexible activity–section–crew assignment demands, in linear project-scheduling optimization where research remains insufficient. This has further spurred research on practical scenario-based linear project-scheduling optimization. In response to these gaps, this study developed a scheduling optimization model under the LSM framework to incorporate reverse construction activities and a more flexible crew application mode. Additionally, we verified and analyzed the model’s performance.

The main novelty and contribution of this study are as follows:

1. Reverse construction activity scenarios were introduced into the field of linear scheduling optimization, while a systematic logical constraint system consisting of 48 scenarios was proposed. To the best of our knowledge, this is the first study on the logical relationship description method and constraint system for reverse construction activities under LSM;

2. A more practical scheduling optimization model based on LSM was proposed. The model is compatible with reverse construction logic constraints, proposed in this study, and various other classical constraints, and is capable of handling flexible activity–section–crew assignment requirements;

3. An improved GWO algorithm was designed and used for problem solving. To the best of our knowledge, this is the first application of a GWO-based algorithm in linear scheduling optimization.

2. Literature Review

Based on optimization objectives, these studies can be categorized into resource allocation [11,24–26], resource leveling [27–29], time–cost trade-off [30–32], and others, such as crew routing optimization [33], minimization of duration, and crew work interruptions optimization [11].

2.1. Resource Allocation Problem

The resource allocation research aims to schedule a set of activities subject to resource and precedence constraints to minimize the project makespan [34]. Kong et al. [35] considered resource constraints and various possible combinations of precedence relations, and constructed an optimization model with the objective of minimum duration; the model was
solved using constraint programming. Wang et al. [36] solved the RCPSP with a column generation method. J.D. Garcia-Nieves et al. [26] constructed a mixed integer programming model based on the original problem for project-scheduling optimization with consideration of multi-mode, controlled acceleration routine, and four types of precedence relations. The model was solved by Gurobi. Additional factors, such as construction interruption and multi-crew, were introduced in their following research [37]. With the consideration of potential delay by risks, C.M. Katunagawa et al. [24] proposed a fuzzy linear and repetitive heuristic scheduling model.

2.2. Resource Leveling Problem

Resources are critical aspects influencing the success of a project, while the aim of resource leveling is to guarantee the efficient use of resources. Li et al. [38] researched the resource leveling problem with uncertain time lags and activity durations, and compared the performance of an evolutionary algorithm and a Bat algorithm. Gwak and Lee [39] also conducted research on the resource leveling problem under uncertainty. The proposed mode can minimize resource fluctuations while maximizing project completion probability. A solution method that combines simulation technology and GA was designed for problem solving. A two-stage optimization method and a QPSO-based algorithm were used by Wang et al. [40] for resource leveling, while a simulated annealing algorithm was used by Piriyonesi et al. [41] for resource leveling optimization considering resource constraints and activity splitting. The fixed project duration constraint was relaxed by T. Atan and E. Eren [42] to build a mixed integer linear programming model. The model was used to find the best leveled schedule with the greedy heuristic method.

2.3. Time–Cost Trade-off Problem

The most common method to shorten the construction period is to increase resource allocation, but this inevitably comes with an increase in direct costs. The time–cost trade-off problem (TCTP) aims to find an optimum balance between time and cost [43]. S.M.R. Alavipour and D. Arditi [44] integrated financing optimization with TCTP to design a genetic and linear programming integrated algorithm for problem solving. T. Salama and O. Moselhi [45] proposed a multi-objective optimization model for repetitive construction projects in uncertain environments based on fuzzy set theory, which can achieve synchronous optimization of construction period, cost, and interruption time. GA was designed to solve the model. D. Liu et al. [46] designed a discrete symbiotic organism search method for the large-scale time–cost trade-off problem. A. Panwar and K.N. Jha [47] studied TCTP with construction quality and safety with NSGA-III for problem solving. Wang et al. [48] proposed a modified streamlined optimization algorithm for TCTP with a continuous curvilinear activity time–cost relationship.

Based on the above-mentioned classic optimization problems, various factors that may exist during construction are considered, resulting in mixed optimization problems with different optimization objectives, such as overtime usage [49] and unit delivery delays [50]. Meanwhile, various new algorithms, such as cloud computing [51,52] and the cuckoo search algorithm [52], are gradually being applied to project-scheduling problems.

2.4. Knowledge Gaps

A summary of related literature is described in Table 1. It could be seen that although this body of research has added considerable knowledge, existing studies on scheduling optimization based on LSM still have shortcomings in three key areas.
Table 1. Summary of related literature.

<table>
<thead>
<tr>
<th>Authors (Year)</th>
<th>Methodology</th>
<th>Multi-Mode</th>
<th>Multi-Crew</th>
<th>Multi-Section</th>
<th>Flexible Section Selection</th>
<th>Flexible Activity–Section–Crew Assignment</th>
<th>Reverse Construction</th>
<th>Solution Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>F. Kong, D. Dou (2020) [35]</td>
<td>CPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CP</td>
</tr>
<tr>
<td>Q. Wang et al. (2019) [36]</td>
<td>CPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CG</td>
</tr>
<tr>
<td>J.D. Garcia-Nieves et al. (2018) [26]</td>
<td>LSM</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Gurobi</td>
</tr>
<tr>
<td>J.D. Garcia-Nieves et al. (2019) [37]</td>
<td>LSM</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Gurobi</td>
</tr>
<tr>
<td>C.M. Katsuragawa et al. (2021) [24]</td>
<td>LSM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Heuristic</td>
</tr>
<tr>
<td>H.B. Li (2019) [38]</td>
<td>CPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>EA</td>
</tr>
<tr>
<td>Wang et al. (2020) [40]</td>
<td>LSM</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>QPSO</td>
</tr>
<tr>
<td>Piriyonesi et al. (2019) [41]</td>
<td>CPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SA</td>
</tr>
<tr>
<td>D. Liu et al. (2020) [46]</td>
<td>CPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SOS</td>
</tr>
<tr>
<td>A. Panwar, K.N. Jha (2021) [47]</td>
<td>CPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>NSGA-III</td>
</tr>
<tr>
<td>J. Wang et al. (2023) [48]</td>
<td>CPM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SO</td>
</tr>
<tr>
<td>A. Altuwaim, K. El-Rayes (2021) [49]</td>
<td>LSM</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GA</td>
</tr>
<tr>
<td>M.S. Eid et al. (2021) [50]</td>
<td>LSM</td>
<td>√</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GA</td>
</tr>
<tr>
<td>This study</td>
<td>LSM</td>
<td>√</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>GWO</td>
</tr>
</tbody>
</table>

First, linear project naturally has a bi-directional construction scenario. On the one hand, some special construction activities, such as tunnel engineering, require bi-directional construction to overcome the insufficient workspace to accelerate construction progress. On the other hand, the introduction of bi-directional construction brings more spatiotemporal flexibility to linear project-scheduling optimization, which may further lead to better construction schedules that are potentially unknown. However, no description system has been proposed for the spatiotemporal logical relationship between activities based on LSM in a reverse construction scenario. This not only limits the model building based on reverse construction scenarios, but also means research on the effect of incorporating reverse construction activities in linear project-scheduling optimization remains insufficient.

Second, the linear characteristics and the typically large scale of linear projects necessitate multisection and multicrew scenarios. However, most existing studies employ a unified perspective in dividing sections, essentially overlooking the unique requirements of activities, which may reduce the practicality of the models. The consideration of personalized section division will increase the complexity of spatiotemporal constraints between activities, which brings necessity for a spatiotemporal logical constraint system considering actual scenarios.

Further, the application of a more flexible model for activity–section–crew assignment demands additional exploration.

3. Development of the LSM-CBPS Model

Given the limitations in current research, this paper introduces reverse construction activities and realistic scenarios, i.e., multisegment–multicrew, multicrew–multimode, and multicrew–multidirection flexible matching within the framework of LSM. Subsequently, a linear scheduling model considering bidirectional construction and other practical scenarios (LSM-CBPS) is established.

3.1. Assumptions and Notations

The following assumptions should be followed in the proposed model.

(1) The deterministic scenario is considered, i.e., related parameters are determined;
(2) Crew transition time between sections is not considered;
(3) Construction direction and mode are not allowed to be changed during construction;
(4) Other optimization objectives were not considered except for the minimum duration;
(5) Forward construction is defined as construction from a small mileage scale to a large mileage scale, while reverse construction is the opposite.

The parameters used in the proposed model are described in Tables 2–5, as follows.

Table 2. Index of LSM-CBPS.

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>index of activities</td>
</tr>
<tr>
<td>k</td>
<td>index of segment/section</td>
</tr>
<tr>
<td>c</td>
<td>index of crews</td>
</tr>
<tr>
<td>m</td>
<td>index of construction mode</td>
</tr>
</tbody>
</table>

Table 3. Constant of LSM-CBPS.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>number of activities</td>
</tr>
<tr>
<td>SD&lt;sub&gt;i&lt;/sub&gt;</td>
<td>starting distance of activity i</td>
</tr>
<tr>
<td>SD&lt;sub&gt;i,k&lt;/sub&gt;</td>
<td>starting distance of section k in activity i</td>
</tr>
<tr>
<td>ED&lt;sub&gt;i&lt;/sub&gt;</td>
<td>end distance of activity i</td>
</tr>
<tr>
<td>ED&lt;sub&gt;i,k&lt;/sub&gt;</td>
<td>end distance of section k in activity i</td>
</tr>
<tr>
<td>S&lt;sub&gt;i&lt;/sub&gt;</td>
<td>number of sections in activity i</td>
</tr>
</tbody>
</table>
Table 3. Cont.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>crews&lt;sub&gt;i&lt;/sub&gt;</td>
<td>maximum number of crews available for activity &lt;i&gt;i&lt;/i&gt;</td>
</tr>
<tr>
<td>&lt;i&gt;i&lt;/i&gt;</td>
<td>section &lt;i&gt;k&lt;/i&gt; in activity &lt;i&gt;i&lt;/i&gt; (subactivity)</td>
</tr>
<tr>
<td>&lt;i&gt;sc&lt;/i&gt;&lt;sub&gt;i&lt;/sub&gt;</td>
<td>set of crews chosen for activity &lt;i&gt;i&lt;/i&gt;</td>
</tr>
<tr>
<td>&lt;i&gt;nsc&lt;/i&gt;&lt;sub&gt;i&lt;/sub&gt;</td>
<td>number of crews chosen for activity &lt;i&gt;i&lt;/i&gt;</td>
</tr>
<tr>
<td>seq&lt;sub&gt;&lt;i&gt;c&lt;/i&gt;&lt;/sub&gt;</td>
<td>construction sequence set for crew &lt;i&gt;c&lt;/i&gt; of activity &lt;i&gt;i&lt;/i&gt;</td>
</tr>
<tr>
<td>modes&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;&lt;/sub&gt;</td>
<td>number of construction mode for activity &lt;i&gt;i&lt;/i&gt;</td>
</tr>
<tr>
<td>&lt;i&gt;rs&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
<td>required resources for subactivity &lt;i&gt;i&lt;/i&gt;&lt;sub&gt;&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
</tr>
<tr>
<td>&lt;i&gt;pr&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
<td>construction rate for subactivity &lt;i&gt;i&lt;/i&gt;&lt;sub&gt;&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
</tr>
<tr>
<td>&lt;i&gt;dur&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
<td>duration for subactivity &lt;i&gt;i&lt;/i&gt;&lt;sub&gt;&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
</tr>
<tr>
<td>&lt;i&gt;cs&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
<td>direct cost per mile for subactivity &lt;i&gt;i&lt;/i&gt;&lt;sub&gt;&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
</tr>
<tr>
<td>&lt;i&gt;tp&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;&lt;/sub&gt;</td>
<td>type of activity &lt;i&gt;i&lt;/i&gt;</td>
</tr>
<tr>
<td>Tmin&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;j&lt;/i&gt;&lt;/sub&gt;</td>
<td>minimum time buffer between activity &lt;i&gt;i&lt;/i&gt; and &lt;i&gt;j&lt;/i&gt;</td>
</tr>
<tr>
<td>Tmax&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;j&lt;/i&gt;&lt;/sub&gt;</td>
<td>maximum time buffer between activity &lt;i&gt;i&lt;/i&gt; and &lt;i&gt;j&lt;/i&gt;</td>
</tr>
<tr>
<td>Dmin&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;j&lt;/i&gt;&lt;/sub&gt;</td>
<td>minimum distance buffer between activity &lt;i&gt;i&lt;/i&gt; and &lt;i&gt;j&lt;/i&gt;</td>
</tr>
<tr>
<td>Dmax&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;j&lt;/i&gt;&lt;/sub&gt;</td>
<td>maximum distance buffer between activity &lt;i&gt;i&lt;/i&gt; and &lt;i&gt;j&lt;/i&gt;</td>
</tr>
<tr>
<td>R&lt;style&gt;.max&lt;/style&gt;</td>
<td>daily resource supply</td>
</tr>
<tr>
<td>RW&lt;style&gt;.max&lt;/style&gt;</td>
<td>expected resource usage fluctuation</td>
</tr>
<tr>
<td>TC&lt;style&gt;.max&lt;/style&gt;</td>
<td>expected total project cost</td>
</tr>
</tbody>
</table>

Table 4. Decision variable of LSM-CBPS.

<table>
<thead>
<tr>
<th>Decision Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;i&gt;st&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
<td>start time of construction in the &lt;i&gt;k&lt;/i&gt;th section of activity &lt;i&gt;i&lt;/i&gt;</td>
</tr>
<tr>
<td>&lt;i&gt;xc&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;,&lt;i&gt;c&lt;/i&gt;&lt;/sub&gt;</td>
<td>denotes whether the &lt;i&gt;k&lt;/i&gt;th section of activity &lt;i&gt;i&lt;/i&gt; is constructed by construction crew &lt;i&gt;c&lt;/i&gt;: if yes, it equals 1; if not, it equals 0.</td>
</tr>
<tr>
<td>&lt;i&gt;xm&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;,&lt;i&gt;m&lt;/i&gt;&lt;/sub&gt;</td>
<td>indicates whether the &lt;i&gt;k&lt;/i&gt;th section of activity &lt;i&gt;i&lt;/i&gt; is constructed according to construction mode &lt;i&gt;m&lt;/i&gt;: if yes, it equals 1; if not, it equals 0.</td>
</tr>
<tr>
<td>dir&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
<td>construction direction of activity &lt;i&gt;i&lt;/i&gt; in the &lt;i&gt;k&lt;/i&gt;th section, where forward is denoted by 1 and reverse by 0.</td>
</tr>
</tbody>
</table>

Table 5. Variable of LSM-CBPS.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>D&lt;style&gt;.max&lt;/style&gt;</td>
<td>shortest construction period</td>
</tr>
<tr>
<td>&lt;i&gt;et&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;&lt;/sub&gt;</td>
<td>end time of construction in the &lt;i&gt;k&lt;/i&gt;th section of activity &lt;i&gt;i&lt;/i&gt;</td>
</tr>
<tr>
<td>&lt;i&gt;xl&lt;/i&gt;&lt;sub&gt;&lt;i&gt;i&lt;/i&gt;,&lt;i&gt;k&lt;/i&gt;,&lt;i&gt;t&lt;/i&gt;&lt;/sub&gt;</td>
<td>whether the &lt;i&gt;k&lt;/i&gt;th section of activity &lt;i&gt;i&lt;/i&gt; is constructed on the &lt;i&gt;t&lt;/i&gt;th day: if yes, it is 1; otherwise, it is 0.</td>
</tr>
</tbody>
</table>

3.2. Problem Description

Consider a linear project comprising <i>N</i> activities denoted as <i>i</i>, <i>i</i> = 1, 2, ..., <i>N</i>, such that each activity has a starting distance denoted as <i>SD</i><sub><i>i</i></sub> and an end distance denoted as <i>ED</i><sub><i>i</i></sub>. Each linear activity is further divided into subactivities across multiple construction sections, and for linear activity <i>i</i>, the number of sections is denoted as <i>S</i><sub><i>i</i></sub>. For the subactivity <i>i</i><sub><i>k</i></sub> of activity <i>i</i> in the <i>k</i>th segment, <i>k</i> = 1, 2, ..., <i>S</i><sub><i>i</i></sub>, the starting and ending distances
are represented by $SD_{i,k}$ and $ED_{i,k}$, respectively. Notably, block-type activities are not segmented.

1. Activity $i$ can be completed by $crews_i$ crews. For each section, only one crew can be selected, and that crew is always used for construction within the mileage range. For subactivity $i_k$, the chosen crew sequence number is $c$, $c = 1, 2, \ldots, crews_i$. The crew number selected for all subactivities forms the set $sc_i$, and the number of crews chosen for activity $i$ is $nsc_i (nsc_i \leq crews_i)$. Additionally, the construction sequence set $seq_{i,c}$ for crew $c$ can be derived, in which the set’s elements are sequence numbers of the subactivities that will be constructed by crew $c$. These are arranged in ascending order. Notably, crew transition time is not considered.

2. Activity $i$ has $modes_i$ construction modes. A certain crew can select only one construction mode, $m$, $m = 1, 2, \ldots, modes_i$, and it will be maintained until the crew completes all its tasks. Each construction mode is a triplet. With the construction mode, the following information for subactivity $i_k$ can be obtained: the required resources, $rs_{i,k}$; rate, $pr_{i,k}$; or duration, $dur_{i,k}$, of construction and the direct cost per mile, $cs_{i,k}$.

3. For linear activity $i$, each crew has a choice between forward/reverse construction directions. Forward construction progresses from small mileage to large mileage, while reverse construction proceeds from large mileage to small mileage. Each crew can select only one construction direction, and it will be maintained until the crew completes all its tasks. The construction direction also determines the construction sequence. For example, if $seq_{1,1} = [1,2,3]$ and crew 1 of activity 1 selects a forward direction, then the construction sequence is $[1,2,3]$. If crew 1 selects the reverse direction, the construction sequence becomes $[3,2,1]$, as shown in Figure 1.

Figure 1. Illustration of the different construction directions.

Linear project-scheduling optimization considering various practical scenarios such as bidirectional construction is depicted in Figure 2. The aim of this problem was to determine the construction crew, start time, construction mode, and construction direction for each subactivity. Consequently, the project schedule can fulfill various constraints and achieve an optimal objective function.

### 3.3. Decision Variable

The decision variables of this model encompass the following four types:

- $st_{i,k}$: start time of construction in the $k$th section of activity $i$;
- $xc_{i,k,c}$: denotes whether the $k$th section of activity $i$ is constructed by construction crew $c$: if yes, it equals 1; if not, it equals 0;
- $xm_{i,k,m}$: indicates whether the $k$th section of activity $i$ is constructed according to construction mode $m$: if yes, it equals 1; if not, it equals 0;
- $dir_{i,k}$: construction direction of activity $i$ in the $k$th section, where forward is denoted by 1 and reverse by 0.
3.4. Objective Function

In this study, we focused on construction duration optimization; therefore, the objective function for the shortest construction period was used, which is formulated as follows:

\[ \min T = D_{\text{max}}, \]  
(1)

\[ D_{\text{max}} = \max t_{i,k}, \quad i = 1, 2, \ldots, N, \quad k = 1, 2, \ldots, s_i, \]  
(2)

where \( t_{i,k} \) is the end time of construction in the \( k \)th section of activity \( i \), which can be calculated as follows:

\[ t_{i,k} = s_{i,k} + d_{i,k}, \]  
(3)

\[ d_{i,k} = \begin{cases} \frac{ED_{i,k} - SD_{i,k}}{t_{p_i} = \text{linear}}, & \text{if } \frac{t_{p_i}}{dur_{i,k}} \leq \text{block} \\ \text{linear}, & \text{otherwise} \end{cases} \]  
(4)

where \( t_{p_i} \) represents the type of activity \( i \).

3.5. Constraints

3.5.1. Spatiotemporal Constraints in the Forward Construction Scenario

In this study, spatiotemporal constraints were utilized to describe the logical relationships between activities. Within the classical LSM framework, the logical constraints between activities are mostly limited to scenarios where activities adopt forward construction. Existing research has thoroughly investigated the 24 types of spatiotemporal constraints in this scenario [25,53]. This paper will not repeat this description.

3.5.2. Spatiotemporal Constraints in Reverse Construction Scenarios

To address the practical scenarios of reverse construction in linear project construction and explore its impact for scheduling optimization in terms of project duration, this section describes the 48 types of spatiotemporal logic constraints involved in reverse construction scenarios.

(1) Time constraints

1) If a minimum time constraint exists between subactivity \( j_l \) and subactivity \( i_k \), scenarios are categorized and discussed based on activity type, construction direction, construction rate, and activity mileage. The minimum time constraint between \( j_l \) and \( i_k \) is subdivided into the following 12 cases, which are shown in Figure 3:

\[ \text{Case 1: } s_{l,k} + T_{i,k,SD_{l,j},ED_{l,k}} + T_{\text{min}_{i,j}} \leq s_{l,j}; \]  
(5)

\[ \text{Case 2: } s_{l,k} + T_{i,k,SD_{l,j},ED_{l,k}} + T_{\text{min}_{i,j}} \leq s_{l,j} + T_{j,l,SD_{l,i},SD_{l,k}}; \]  
(6)

\[ \text{Case 3: } s_{l,k} + T_{i,k,SD_{l,j},ED_{l,k}} + T_{\text{min}_{i,j}} \leq s_{l,j} + T_{j,l,ED_{l,k},ED_{l,j}}; \]  
(7)
Case 4: \[ st_{i,k} + T_{i,k,SD_{i,k},ED_{j}}, + T_{min_{i,j}} \leq st_{j,l}; \]  

Case 5: \[ st_{i,k} + T_{i,k,SD_{i,k},ED_{j}}, + T_{min_{i,j}} \leq st_{j,l} + T_{j,l,SD_{j,l},ED_{j},}; \]  

Case 6: \[ st_{i,k} + T_{i,k,SD_{i,k},ED_{j}}, + T_{min_{i,j}} \leq st_{j,l} + T_{j,l,SD_{j,l},ED_{j},}; \]  

Case 7: \[ st_{i,k} + T_{min_{i,j}} \leq st_{j,l} + T_{j,l,ED_{j},ED_{j},}; \]  

Case 8: \[ st_{i,k} + T_{i,k,ED_{j},ED_{j},} + T_{min_{i,j}} \leq st_{j,l}; \]  

Case 9: \[ st_{i,k} + T_{i,k,SD_{j,l},ED_{j},} + T_{min_{i,j}} \leq st_{j,l}; \]  

Case 10: \[ st_{i,k} + T_{i,k,SD_{i,k},ED_{j},} + T_{min_{i,j}} \leq st_{j,l}; \]  

Case 11: \[ st_{i,k} + d_{i,1} + T_{min_{i,j}} \leq st_{j,l} + T_{j,l,ED_{j},1,ED_{j},}; \]  

Case 12: \[ st_{i,k} + d_{i,1} + T_{min_{i,j}} \leq st_{j,l}. \]  

2) Similarly, the maximum time constraint between \( j_l \) and \( i_k \) is subdivided into the following 12 cases, which are shown in Figure 4.

Case 1: \[ st_{i,k} + T_{max_{i,j}} \geq st_{j,l} + T_{j,l,SD_{j,l},ED_{j},}; \]  

Case 2: \[ st_{i,k} + T_{i,k,ED_{j},ED_{j},} + T_{max_{i,j}} \geq st_{j,l} + T_{j,l,SD_{j,l},ED_{j},}; \]  

Case 3: \[ st_{i,k} + T_{i,k,SD_{j,l},SD_{j,l}} + T_{max_{i,j}} \geq st_{j,l} + T_{j,l,SD_{j,l},ED_{j},}; \]  

Case 4: \[ st_{i,k} + T_{max_{i,j}} \geq st_{j,l} + T_{j,l,SD_{j,l},ED_{j},}; \]  

Case 5: \[ st_{i,k} + T_{max_{i,j}} \geq st_{j,l} + T_{i,k,ED_{j},ED_{j},}; \]  

Case 6: \[ st_{i,k} + T_{j,l,ED_{j},ED_{j},} + T_{max_{i,j}} \geq st_{j,l}; \]  

Case 7: \[ st_{i,k} + T_{i,k,SD_{j,l},ED_{j},} + T_{max_{i,j}} \geq st_{j,l} + T_{j,l,SD_{j,l},ED_{j},}; \]  

Case 8: \[ st_{i,k} + T_{i,k,SD_{i,k},ED_{j},} + T_{max_{i,j}} \geq st_{j,l} + T_{j,l,SD_{i,k},ED_{j},}; \]  

Case 9: \[ st_{i,k} + T_{min_{i,j}} \geq st_{j,l}; \]  

Case 10: \[ st_{i,k} + T_{i,k,ED_{j},1,ED_{j},} + T_{min_{i,j}} \geq st_{j,l}; \]  

Case 11: \[ st_{i,k} + d_{i,1} + T_{max_{i,j}} \geq st_{j,l} + T_{j,l,SD_{j,l},ED_{j},}; \]  

Case 12: \[ st_{i,k} + d_{i,1} + T_{max_{i,j}} \geq st_{j,l} + T_{j,l,SD_{j,l},ED_{j},}. \]
Minimum time buffer constraints between the activities under reverse direction.

Figure 3.
Figure 4. Maximum time buffer constraints between activities under reverse direction.
(2) Distance constraints

1) If a minimum distance constraint exists between subactivity \( j_i \) and subactivity \( i_k \), scenarios are categorized and discussed based on activity type, construction direction, construction rate, and activity mileage. The minimum distance constraint between \( j_i \) and \( i_k \) is subdivided into the following 12 cases, which are shown in Figure 5.

Case 1: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\min,i,j} \leq SD_{j,i} \) \hspace{1cm} (29)

Case 2: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\min,i,j} \leq SD_{j,l} + D_{j,l,t,et,i_k} \) \hspace{1cm} (30)

Case 3: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\min,i,j} \leq SD_{j,l} + D_{j,l,t,et,i_j} \) \hspace{1cm} (31)

Case 4: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\min,i,j} \leq SD_{j,l} + D_{j,l,t,et,i_j} \) \hspace{1cm} (32)

Case 5: \( SD_{i,k} + D_{\min,i,j} \leq SD_{j,l} + D_{j,l,t,et,i_j} \) \hspace{1cm} (33)

Case 6: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\min,i,j} \leq SD_{j,l} \) \hspace{1cm} (34)

Case 7: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\min,i,j} \leq SD_{j,l} + D_{j,l,t,et,i_j} \) \hspace{1cm} (35)

Case 8: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\min,i,j} \leq SD_{j,l} + D_{j,l,t,et,i_j} \) \hspace{1cm} (36)

Case 9: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\min,i,j} \leq SD_{j,l} \) \hspace{1cm} (37)

Case 10: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\min,i,j} \leq SD_{j,l} \) \hspace{1cm} (38)

Case 11: \( ED_{i,l} + D_{\min,i,j} \leq SD_{j,l} + D_{j,l,t,et,i_j} \) \hspace{1cm} (39)

Case 12: \( ED_{i,l} + D_{\min,i,j} \leq SD_{j,l} \) \hspace{1cm} (40)

2) Similarly, the maximum distance constraint between \( j_i \) and \( i_k \) is subdivided into the following 12 cases, which are shown in Figure 6.

Case 1: \( SD_{i,k} + D_{\max,i,j} \geq SD_{j,l} + D_{j,l,t,et,i_k} \) \hspace{1cm} (41)

Case 2: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\max,i,j} \geq SD_{j,l} + D_{j,l,t,et,i_k} \) \hspace{1cm} (42)

Case 3: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\max,i,j} \geq SD_{j,l} + D_{j,l,t,et,i_k} \) \hspace{1cm} (43)

Case 4: \( SD_{i,k} + D_{\max,i,j} \geq SD_{j,l} + D_{j,l,t,et,i_k} \) \hspace{1cm} (44)

Case 5: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\max,i,j} \geq SD_{j,l} + D_{j,l,t,et,i_k} \) \hspace{1cm} (45)

Case 6: \( SD_{i,k} + D_{i,k},t,et,i_k + D_{\max,i,j} \geq SD_{j,l} + D_{j,l,t,et,i_k} \) \hspace{1cm} (46)

Case 7: \( SD_{i,k} + D_{\max,i,j} \geq SD_{j,l} + D_{j,l,t,et,i_j} \) \hspace{1cm} (47)
Case 8: \[ SD_{jk} + D_{jk,et_{i,k}} + D_{max,i,j} \geq SD_{ij} \]  

Case 9: \[ SD_{jk} + D_{max,i,j} \geq SD_{ij} \]  

Case 10: \[ SD_{jk} + D_{et_{i,j,k}} + D_{max,i,j} \geq SD_{ij} \]  

Case 11: \[ ED_{i,1} + D_{max,i,j} \geq SD_{j,l} + D_{st_{i,j,l}} \]  

Case 12: \[ ED_{i,1} + D_{max,i,j} \geq SD_{j,l} + D_{st_{i,j,l}} \]  

3.5.3. Resource Constraints

(1) Daily resource usage constraints

\[ R_{max} - R_t \geq 0, 1 \leq t \leq D_{max}, \]  

where \( R_{max} \) represents the daily resource supply; \( D_{max} \) denotes the total project duration; and \( R_t \) indicates the resource usage on the \( t \)th day. \( R_t \) can be calculated as follows:

\[ R_t = \sum_{i=1}^{N} \sum_{k=1}^{S_i} x_{t,i,k} \cdot r_{i,k}, \]  

where \( x_{t,i,k} \) refers to whether the \( k \)th section of activity \( i \) is constructed on the \( t \)th day. If yes, it is 1; otherwise, it is 0.

\[ x_{t,i,k} = \begin{cases} 1, & st_{i,k} < t + 1 \text{ and } et_{i,k} > t \\ 0, & \text{otherwise} \end{cases}, \]  

(2) Resource leveling constraints

\[ \sum_{t=1}^{D_{max}} |R_t - R_{t-1}| \leq RW_{max} \]  

where \( RW_{max} \) represents the expected resource usage fluctuation.

3.5.4. Construction Crew Constraints

Considering the influence of multicrew construction scenarios, the following two types of construction crew constraints were introduced into the model.

(1) Construction crew constraints within section

This constraint ensures that each section can be assigned to only one construction crew.

\[ \sum_{c=1}^{\text{crews}_i} x_{c,i,k} = 1, \ i = 1, 2, \ldots, N, \ k = 1, 2, \ldots, S_i. \]
Figure 5. Minimum distance buffer constraints between activities under reverse direction.
Figure 6. Maximum distance buffer constraints between activities under reverse direction.
(2) Cross-sectional construction continuity constraints

The construction continuity constraint refers to the uninterrupted execution of construction activities, ensuring efficient resource utilization and avoiding wastage. In the classical LSM, where each activity is treated as a whole, construction continuity is naturally achieved. However, in the multisection scenario, where activities are divided into subactivities, additional constraints are required to ensure construction continuity between these subactivities. As this study focused on multicrew construction, the construction continuity constraints were applied at the crew level. Furthermore, considering the flexible matching relationship of multisection–multicrew, multicrew–multimode, and multicrew–multidirection, the construction continuity constraints were extended to include time continuity, direction consistency, and mode consistency, as follows:

\[
dir_{i,k} = dir_{i,l}, \quad i = 1, 2, \ldots, N, \quad 1 \leq k < l \leq S_i
\]

\[
xm_{i,k,m} = xm_{i,l,m}, \quad i = 1, 2, \ldots, N, \quad m = 1, 2, \ldots, \text{modes}_i, \quad 1 \leq k < l \leq S_i
\]

\[
et_{i,k} = st_{i,l}, \quad i = 1, 2, \ldots, N, \quad 1 \leq k < l \leq S_i, \quad dir_{i,k} = dir_{i,l} = 1,
\]

\[
et_{i,k} = et_{i,l}, \quad i = 1, 2, \ldots, N, \quad 1 \leq k < l \leq S_i, \quad dir_{i,k} = dir_{i,l} = 0,
\]

Here, the subactivities \(i_k\) and \(i_l\) belong to the construction sequence set \(\text{seq}_{i,c}\) of crew \(c\) for activity \(i\), and they are adjacent subactivities. The following constraints should be satisfied:

\[
\sum_{s=k+1}^{l-1} xc_{i,s,c} = 0, \quad c = 1, 2, \ldots, \text{crews}_i, \quad 1 \leq k < l \leq S_i,
\]

\[
xc_{i,k,c} = xc_{i,l,c} = 1, \quad c = 1, 2, \ldots, \text{crews}_i, \quad 1 \leq k < l \leq S_i.
\]

3.5.5. Construction Period Constraints

To ensure the practical significance of defining the construction period, active construction must be initiated on day 0 after the project begins. Therefore, first-day construction constraints must be introduced to restrict the start time of each activity.

\[
st_{i,k} = st'_{i,k} - st'_{i,min},
\]

where \(st'_{i,k}\) represents the start time of each subactivity, and \(st'_{i,min}\) represents the minimum value among them.

\[
st'_{i,min} = \min(st'_{1,k}, st'_{2,k}, \ldots, st'_{N,S_i}).
\]

3.5.6. Cost Constraints

The expected total project cost, which includes both direct and indirect costs, is denoted by \(TC_{\text{max}}\), which is expressed as follows:

\[
TC \leq TC_{\text{max}},
\]

\[
TC = \sum_{i=1}^{N} \sum_{k=1}^{S_i} cs_{i,k} \cdot (BD_{i,k} - SD_{i,k}) + D_{\text{max}} \cdot icr,
\]

where \(icr\) is the daily indirect cost of the project.
3.5.7. Construction Mode Constraints

For subactivities within any section of an activity, only one construction mode can be selected. This ensures that a consistent construction approach is adopted for each subactivity in a given section.

\[ \sum_{m=1}^{\text{modes}_i} x_{i,k,m} = 1, \quad i = 1, 2, \ldots, N, \quad k = 1, 2, \ldots, S_i. \]  

(68)

4. Design of a Solving Algorithm for the LSM-CBPS Model

4.1. Grey Wolf Optimization Algorithm

The grey wolf optimizer (GWO), initially proposed by Mirjalili et al. in 2014 [54], is a meta-heuristic optimization algorithm inspired by the hunting behavior of grey wolves. It has been successfully applied in various fields, such as power flow optimization [55], vehicle path planning [56], and feature selection [57]. In the GWO, the solution with the best fitness is considered to be \( \alpha \) wolf, the second and third solutions with the best fitness are \( \beta \) wolf and \( \delta \) wolf, respectively, and the remaining solutions are \( \omega \) wolves. The hunting (optimization) process is guided by the top three grades of wolves, followed by \( \omega \) wolves. Assuming the population size is \( N \) and the search space is \( D \)-dimensional, the position of the \( i \)th grey wolf in \( D \)-dimensional space can be represented as \( X_i = (x_{1i}, x_{2i}, \ldots, x_{Di}) \), \( i = 1, 2, \ldots, N \).

The hunting behavior of grey wolves, where they surround their prey, is modeled using the following equations.

\[ D = \left| C \cdot X_p^d(t) - X_i^d(t) \right| \]  

(69)

\[ X_i^d(t + 1) = X_p^d(t) - A \cdot D \]  

(70)

where \( t \) is the current iteration number; \( X_p = (x_{1p}, x_{2p}, \ldots, x_{Dp}) \) is the prey position; and \( A \) and \( C \) are the coefficients, which can be defined as:

\[ A = 2a \cdot r_1 - a \]  

(71)

\[ C = 2 \cdot r_2 \]  

(72)

where \( r_1 \) and \( r_2 \) are random numbers between [0,1]; and \( a \) is a distance control parameter that linearly decreases from 2 to 0 as the number of iterations increases.

Grey wolves can identify the location of the prey and surround it. The \( \beta \) and \( \delta \) wolves, led by wolf \( \alpha \), guide the wolf pack during the hunting process. The algorithm keeps track of the three solutions with the best fitness obtained so far, whereas the other candidate wolves within the population update their positions accordingly. The positions of the \( \beta \), \( \delta \), and \( \alpha \) wolves estimate the position of the prey, and the positions of the other wolves are randomly updated around the estimated prey position. The equation for updating the positions is as follows:

\[
\begin{align*}
D_\alpha &= \left| C \cdot X_\alpha^d(t) - X_i^d(t) \right| \\
D_\beta &= \left| C \cdot X_\beta^d(t) - X_i^d(t) \right| \\
D_\delta &= \left| C \cdot X_\delta^d(t) - X_i^d(t) \right| \\
X_{i,\alpha}^d(t+1) &= X_{\alpha}^d(t) - A \cdot D_\alpha \\
X_{i,\beta}^d(t+1) &= X_{\beta}^d(t) - A \cdot D_\beta \\
X_{i,\delta}^d(t+1) &= X_{\delta}^d(t) - A \cdot D_\delta
\end{align*}
\]  

(73)

(74)
When the prey stops moving, the wolves attack it to complete the hunt. This attacking behavior is simulated by reducing the value of $a$. As $A$ is a random value during the interval $[a, -a]$, when $a$ decreases, the range of $A$ decreases as well. When $|A| < 1$, the wolves are forced to attack their prey, representing the local search phase of the algorithm. The searching behavior of prey corresponds to the global search. When $|A| > 1$, it forces the wolves to spread out and search for prey independently. The linear decrease in $a$ balances the local search and global search.

4.2. Improved Grey Wolf Optimization Algorithm Design

In this study, the algorithm was designed based on the framework of GWO, and the solution of the LSM-CBPS model was obtained. The original GWO was enhanced in the following three aspects.

(1) Discretization: The original GWO was designed for continuous optimization problems; however, the model contains discrete variables. To make the algorithm applicable to the model, a discretization strategy was applied. Therefore, in the improved GWO algorithm, a discrete position update operator was proposed to update the positions of the individual;

(2) Constraint handling: The GWO itself does not inherently handle constraints. Given the presence of multiple complex constraints in the proposed model, directly using the GWO may not yield satisfactory solutions. Therefore, a dynamic constraint handling mechanism and heuristic rules tailored to the optimization problem characteristics were introduced. These enhancements aimed to improve the algorithm’s capability to handle constrained optimization problems;

(3) Search ability improvement: GWO is prone to getting trapped in local optima and may have low solution accuracy. To address these issues, additional improvement strategies were incorporated. In particular, a cross-mutation strategy was introduced to enhance both the global and local search abilities of GWO.

The pseudo-code for the improvement GWO is depicted in Algorithm 1. The algorithm design includes seven components: coding rules, population initialization method, decoding rules, discrete position update operators, dynamic constraint handling methods, cross-mutation strategies, and heuristic rules.

**Algorithm 1. Pseudo-code of improved GWO.**

**Input:** number of activity $N$, construction mode $mode_i$, construction direction $dir_{ik}$, construction crew $crew_i$, population size $npop$, maximum number of iterations $itr_{max}$, and distance control parameter $a$.

**Output:** project schedule.

1. For each individual in $npop$.
2. Determine the value of decision variables $st_{i,k}$, $xc_{i,k,c}$, $xm_{i,k,m}$, and $dir_{i,k}$ through random generation.
3. Modify $st_{i,k}$ according to heuristic rules.
4. Generate initial population and set initial parameters of the algorithm.
5. Select the best three individual according to dynamic constraint processing rules.
6. While the number of iterations is less than $itr_{max}$ do
7. Update the individual position through location update operator.
8. Use cross-mutation operator to enhance population diversity.
9. Modify $st_{i,k}$ for each individual according to heuristic rules.
10. Update the best three individual according to dynamic constraint processing rules.
11. Output optimal individual.
4.2.1. Encoding Mode

The scheduling optimization model involves four decision variables. To represent these variables, a four-layer coding rule was constructed, as illustrated in Figure 7.

![Figure 7. Coding method.](image)

1. The first layer represents the selected construction crew for each section. When $c_{r,k} = c$, it indicates that the construction crew selected for the subactivity of activity $i$ on the $k$th section is $c$. Subactivities that share the same construction crew $c$ constitute set $\text{seq}_c$ of the construction crew $c$. All selected crews for activity $i$ constitute set $\text{scr}_i$.
2. The second layer represents the start time of each crew. When $c_{t,c} = t$, the construction start time for the $c$th crew of activity $i$ is $t$;
3. The third layer represents the construction direction for each crew. When $c_{d,c} = 1$, the construction direction for the $c$th crew of activity $i$ is positive. When $c_{d,c} = 0$, the direction is reversed;
4. The fourth layer represents the construction mode for each crew. When $c_{m,c} = h$, the construction mode chosen by the $c$th construction crew representing activity $i$ is $h$.

4.2.2. Population Initialization

Once the encoding mode of the grey wolf location information is determined, the next step is generating the initial population. To simplify the process of generating feasible solutions and enhance the efficiency of the algorithm, the crew, mode, and construction direction were randomly initialized. The start time of the construction crew was determined using heuristic rules, as described in Section 4.2.7.

4.2.3. Decode

After obtaining the population information of each generation, the grey wolf location information must be decoded and mapped onto a schedule to calculate the objective function value and constraint violation value, followed by population assessment and updating. The decoding process consists of the following steps.

Step 1: Start with activity 1 and read the selected crews for each section. Obtain the set of selected crews and the construction sequence for each crew;

Step 2: For each crew $c$ in the selected crew set, read the encoded information to determine the construction mode $c_{m,c} = m$, construction direction $c_{d,c} = dr$, and construction start time $c_{t,c} = t$;

If $dr$ is 1, that is, the construction direction is positive, then the crew continues constructing from small miles to large miles during the construction process of activity 1; hence, the construction sequence is traversed from left to right. For the first subactivity $i_k$ in the
sequence, \( \text{dir}_{i,k} = 1 \), \( s_{i,k} = t \), \( x_{m_{i,k,m}} = 1 \) and \( e_{i,k} = s_{i,k} + d_{i,k} \). For the second subactivity \( i_2 \) in the sequence, \( \text{dir}_{i_2,k} = 1 \), \( s_{i_2,k} = e_{i_2,k} \), \( x_{m_{i_2,h,m}} = 1 \), and \( e_{i_2,h} = s_{i_2,h} + d_{i_2,h} \), and so on. If \( dr \) is 0, that is, the construction direction is inverse, then the crew keeps working from large to small miles during the construction process of activity 1; thus, the same calculation was performed by traversing the construction sequence from right to left. Finally, the coding method can cover the information of all decision variables in the model.

Step 3: According to the method in the second step, the construction situation of each crew in each section of the collection is determined in turn;

Step 4: The construction situation for each activity in the project is determined to obtain the overall construction process.

4.2.4. Discrete Location Update Operator

As discussed above, discrete processing strategies were introduced in the improved algorithm. The following discrete location update operators were proposed.

\[
X^d_i(t+1) = \begin{cases} 
X^d_i(t), & \text{if } r1 < \frac{1}{3} \text{ and } r2 > c \\
X^d_i(t), & \text{if } r1 < \frac{2}{3} \text{ and } r1 > \frac{1}{3} \text{ and } r2 > c \\
X^d_i(t), & \text{if } r1 < 1 \text{ and } r1 > \frac{2}{3} \text{ and } r2 > c \\
X^d_i(t), & \text{if } r2 \leq c
\end{cases}, \tag{76}
\]

where \( c \) is the control factor that decreases linearly from 1 to 0 as the number of iterations increases; and \( r1 \) and \( r2 \) are random numbers in the range \([0,1]\). If \( r2 \) is greater than \( c \), the \( d \)-th dimension position information \( X^d_i(t) \) of the \( i \)-th grey wolf individual is updated based on the guidance of the best three grey wolves. In particular, the probability that \( X_i \) is updated to the value of one of the best grey wolves is \( 1/3 \). As the number of iterations increases, the guidance from the best three grey wolves to the group deepens, and the population gradually converges.

4.2.5. Dynamic Constraint Processing

In addition, herein, a dynamic constrained processing method \([58]\) was introduced to effectively solve the constrained optimization problem. This method transforms the single-objective constrained optimization problem into a double-objective unconstrained optimization problem. For grey wolves outside the feasible domain, their primary goal is to enter the feasible domain rather than searching for the optimal solution. When a grey wolf enters the feasible region, the goal is to approach the optimal solution. To compare any two grey wolves, the following rules are applied:

1. When grey wolves \( x_1 \) and \( x_2 \) are not feasible, the one with a small constraint violation value is better;
2. When both the grey wolves \( x_1 \) and \( x_2 \) are feasible, the one with a small objective function value is better;
3. When the grey wolf \( x_1 \) is feasible and \( x_2 \) is not, \( x_1 \) is superior.

4.2.6. Cross-Mutation Strategy

In the original GWO, the position update of each individual is only influenced by the three optimal grey wolves. This restriction limits the information sharing among the population, thereby leading to limited exploration ability and a higher likelihood of getting trapped in local optimal solutions. As the number of iterations increases, the population diversity weakens; consequently, escaping the local optima becomes challenging for the algorithm. To address this, the cross-mutation operator was introduced.
(1) Cross

Two parent grey wolves are randomly selected, and their construction crew, construction start time, construction mode, and construction direction are crossed using the uniform crossover method. For each location, a random number is generated in the range [0,1]. If the random number is less than the crossover probability, the values at that location are exchanged. The process is iterated for all four parts, resulting in two new offspring grey wolves.

(2) Variation

Herein, a variation method based on the activity level is adopted. For each activity, a section is randomly selected, and the selected construction crew for that section undergoes mutation. If the current value is \( c_1 \), it is randomly changed to another value \( c_2 (c_2 \neq c_1) \) in the set of optional crews for that activity. Similarly, the construction mode \( m_1 \) corresponding to the selected crew \( c_2 \) is randomly changed to another value \( m_2 (m_2 \neq m_1) \) from the set of optional modes for that activity. As for the construction direction \( d_{r1} \) corresponding to crew \( c_2 \), it was changed to the opposite direction. This variation process is applied to each activity, ensuring individual variations in the multisegment–multicrew, multicrew–multimode, and multicrew–multidirection matching scenarios, as shown in Figure 8.

![Figure 8](image_url)

**Figure 8.** Illustration of mutation.

4.2.7. Heuristic Rule

In the process of evolution, because of the complexity of the search space and the presence of constraints, there may be many infeasible solutions in the population, which can significantly impact the search performance and effectiveness of the algorithm. The constraint condition-based correction was introduced, which involves adjusting the position of an individual that does not meet the constraint conditions, to enable meeting the constraint conditions and reducing its constraint violation value. The specific process is as follows.

Step 1: For activity \( j \), if a minimum time constraint exists between it and the preceding activity \( i \), according to Section 2, the earliest start time \( Pst_{j,h} \) and end time \( Pet_{j,h} \) for each section of activity \( j \) can be calculated.

For any construction crew \( m \) in the set of selected crews for activity \( j \), the construction sequence \( seq_{j,m} \) is traversed, the difference \( pval_{j,h} \) between the earliest start time \( Pst_{j,h} \) and the current actual start time \( st_{j,h} \) in each section is determined, and the maximum difference is \( mpval_{h,m} \):

\[
pval_{j,h} = Pst_{j,h} - st_{j,h},
\]

\[
mpval_{h,m} = \max \{ pval_{j,h} \}.
\]

If \( mpval_{h,m} \) is greater than 0, it indicates that the start time of some subactivities does not meet the minimum time constraint. In this case, the start time of all the subactivities in
the seq\textsubscript{j,m} is adjusted upward by adjust\textsubscript{j,m}. If a maximum time constraint between activity \textit{j} and \textit{i} does not exist, the final start time \textit{st}_{\textit{j},h} for each activity can be obtained, and the equation is shown in (82). Otherwise, proceed to step 2.

\[
\text{adjust}_{j,m} = \max \{0, \text{mpval}_{h,m} \} \quad (80)
\]

\[
\text{st}'_{j,h} = \text{st}_{j,h} - \text{adjust}_{h,m}, \quad (81)
\]

\[
\text{st}_{j,h} = \text{st}'_{j,h}. \quad (82)
\]

Step 2: If a maximum time constraint between activity \textit{j}\textit{ and } \textit{i} exists, the latest start time Lst\textsubscript{j,h} and end time Lst\textsubscript{j,h} could be calculated by the maximum time constraint.

For each construction crew \textit{m} in the set of selected crews for activity \textit{j}, the construction sequence seq\textsubscript{j,m} is traversed to determine the difference lval\textsubscript{j,h} between the latest start time Lst\textsubscript{j,h} and adjusted start time st\textsubscript{j,h}, and the minimum difference is mlval\textsubscript{h,m}.

\[
lval_{j,h} = \text{st}_{j,h} - \text{Lst}_{j,h}, \quad (83)
\]

\[
mlval_{h,m} = \max \{lval_{j,h}\}. \quad (84)
\]

If the minimum difference value mlval\textsubscript{h,m} is greater than 0, it indicates that the start time of some subactivities does not meet the maximum time constraint. In this case, the start time of all the subactivities in the seq\textsubscript{j,m} is adjusted downward by mlval\textsubscript{h,m}.

\[
\text{adjust}_{j,m} = \max \{0, \text{mlval}_{h,m} \}, \quad (85)
\]

\[
\text{st}''_{j,h} = \text{st}'_{j,h} - \text{adjust}_{h,m}. \quad (86)
\]

The construction sequence seq\textsubscript{j,m} is traversed again to obtain the difference pval\textsubscript{j,h} between the earliest start time Pst\textsubscript{j,h} and the adjusted start time st\textsubscript{j,h} for each section, and the maximum difference is mpval\textsubscript{h,m}.

\[
pval_{j,h} = \text{Pst}_{j,h} - \text{st}''_{j,h}, \quad (87)
\]

\[
mpval_{h,m} = \max \{pval_{j,h}\}. \quad (88)
\]

If the maximum difference mpval\textsubscript{h,m} is greater than 0, it indicates that the start time of certain subactivity after adjustment no longer meets the minimum time constraint; that is, the construction sequence cannot simultaneously meet the minimum and maximum time constraints. In this case, the minimum time constraint is satisfied first, and the final activity start time \textit{st}_{\textit{j},h} is obtained.

\[
\text{st}_{j,h} = \text{st}'_{j,h}. \quad (89)
\]

If the maximum difference mpval\textsubscript{h,m} is less than or equal to 0, it indicates that the adjusted start time of each subactivity satisfies both the minimum and maximum time constraints simultaneously, and the final start time \textit{st}_{\textit{j},h} is obtained.

\[
\text{st}_{j,h} = \text{st}''_{j,h}. \quad (90)
\]

5. Case Study

5.1. Case Description

In this study, the subgrade project from Xiaohuangshan to Wucaiwan of the Wuzhun Railway in China was used. This case is a classic example in the field of linear project-
scheduling optimization. This railway project spans 7 km and consists of eight activities. Detailed information regarding the starting and finishing distances, construction modes, and time and space constraints for each activity can be found in the literature [25,53]. To comprehensively test the effectiveness of the model and algorithm, the following supplements were made.

1. To validate the model’s capability for multicrew and subsection construction and account for the nonuniform characteristics of section lengths in actual construction projects due to varying geological conditions and activity types, each activity was divided into sections based on the specifications provided in Table 6. The number of optional construction crews for each activity corresponds to the number of sections;
2. The constraint values for daily resource usage were set to 190 persons/day, resource leveling constraint value was set to 690, indirect cost was set to 2000 yuan/day, and cost constraint value was set to 9,197,389 yuan.

Table 6. Segments of activities.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Number of Sections</th>
<th>Starting Location of Each Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground shaping</td>
<td>3</td>
<td>DK53+000, DK56+000, DK58+000</td>
</tr>
<tr>
<td>Treatment of weak foundation</td>
<td>1</td>
<td>DK53+350</td>
</tr>
<tr>
<td>General embankment filling</td>
<td>4</td>
<td>DK53+000, DK54+500, DK56+000; DK58+350</td>
</tr>
<tr>
<td>Embankment filling for base of foundation</td>
<td>4</td>
<td>DK53+000, DK54+500, DK55+500, DK58+000</td>
</tr>
<tr>
<td>Embankment filling for foundation surface</td>
<td>3</td>
<td>DK53+000, DK55+500, DK57+500</td>
</tr>
<tr>
<td>Trim of subgrade</td>
<td>3</td>
<td>DK53+000, DK55+500, DK57+500</td>
</tr>
<tr>
<td>Ancillary works for subgrade</td>
<td>3</td>
<td>DK53+000, DK55+500, DK57+500</td>
</tr>
<tr>
<td>Trim and ending</td>
<td>3</td>
<td>DK53+000, DK55+500, DK57+500</td>
</tr>
</tbody>
</table>

Table 7 describes some key experiment designs of the case study.

Table 7. Key experimental design.

<table>
<thead>
<tr>
<th>Assessed Factors</th>
<th>Related Case</th>
<th>Related Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance of GWO</td>
<td>Initial-scale case described in Section 5.1</td>
<td>Section 5.2</td>
</tr>
<tr>
<td></td>
<td>Large-scale case (expanded five times by initial-scale case)</td>
<td>Section 5.2</td>
</tr>
<tr>
<td></td>
<td>Initial-scale case without reverse construction</td>
<td>Section 5.3</td>
</tr>
<tr>
<td></td>
<td>Initial-scale case with reverse construction</td>
<td>Section 5.3</td>
</tr>
<tr>
<td>Performance of LSM-CBPS</td>
<td>Large-scale case without reverse construction</td>
<td>Section 5.3</td>
</tr>
<tr>
<td></td>
<td>Large-scale case with reverse construction</td>
<td>Section 5.3</td>
</tr>
</tbody>
</table>

5.2. Algorithm Effectiveness Analysis

To assess the performance of the improved GWO algorithm, a comparative analysis with the original GWO, genetic, and particle swarm algorithms was conducted. These additions were made to ensure a comprehensive evaluation of the model’s and algorithm’s effectiveness in the real-world scenarios. All four algorithms were evaluated using the same population size and number of iterations (50 and 500, respectively). After reaching the number of iterations, the algorithm will terminate. Table 8 provides details of the parameter settings used.
Table 8. Other parameter settings of the algorithm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved GWO</td>
<td>The crossover probability is 0.2, the mutation probability is 0.2, and the control parameters linearly decrease from 2 to 0.</td>
</tr>
<tr>
<td>Original GWO</td>
<td>The control parameter decreases linearly from 2 to 0.</td>
</tr>
<tr>
<td>GA</td>
<td>Cross probability is 0.8 and mutation probability is 0.08.</td>
</tr>
<tr>
<td>PSO</td>
<td>The inertia weight is linearly reduced from 0.9 to 0.5 and the acceleration factor sum is 2.</td>
</tr>
<tr>
<td></td>
<td>The maximum speed is 30 in the start time dimension and 2 in the other dimensions</td>
</tr>
</tbody>
</table>

To ensure a fair comparison and highlight the differences in the algorithm frameworks, the personalized improvement strategies implemented in the improved GWO algorithm, including population initialization, heuristic rules, and dynamic constraint processing methods, were also applied to the original GWO, genetic algorithm, and particle swarm optimization algorithms. In the genetic algorithm, a uniform crossover mode was used, and the variation mode of the four decision variables aligned with that of the improved GWO algorithm. In the particle swarm optimization (PSO), the dynamic constraint processing compared the global optimal solution of the population with the historical optimal solution of each individual, thereby influencing the update speed and position of the individuals.

The algorithm programs were implemented using MATLAB and executed on a PC with an Intel Core i5 processor and 12 GB of RAM. To mitigate the impact of randomness on the optimization results, each algorithm was independently run 20 times. The optimal solutions, average values, and standard deviations obtained from the four algorithms are presented in Table 9. According to the results in Table 9, the improved GWO algorithm achieved lower average values and standard deviations of the optimal solution compared with the other three algorithms. This indicates its superior search ability, higher solution precision, and better stability. These findings confirm the effectiveness of the proposed algorithm improvement strategies.

Table 9. Results of the various algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Best Solution</th>
<th>Mean Value</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved GWO</td>
<td>91.29</td>
<td>106.01</td>
<td>6.39</td>
</tr>
<tr>
<td>Original GWO</td>
<td>110</td>
<td>117.02</td>
<td>7.12</td>
</tr>
<tr>
<td>GA</td>
<td>130.57</td>
<td>176.33</td>
<td>22.71</td>
</tr>
<tr>
<td>PSO</td>
<td>129.43</td>
<td>179.08</td>
<td>22.48</td>
</tr>
</tbody>
</table>

To validate the effectiveness of the heuristic rule in Section 3, it was removed from each algorithm, and the case was solved again. Each algorithm was independently run 20 times, and the results are shown in Table 10. A comparison of the results in Tables 9 and 10 reveals that the algorithm utilizing the heuristic rule easily obtained feasible and improved solutions. This further demonstrates that designing heuristic rules based on problem characteristics is an effective approach to enhance algorithm performance for optimization problems with complex constraints. Moreover, Table 10 reaffirms the advantages of the improved GWO algorithm over the other three algorithms, even without the heuristic rule.
Table 10. Results of algorithms without the heuristic rule.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Number of Feasible Solutions Obtained</th>
<th>Best Solution</th>
<th>Minimum Constraint Violation Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Improved GWO</td>
<td>9</td>
<td>138.67</td>
<td>—</td>
</tr>
<tr>
<td>Original GWO</td>
<td>3</td>
<td>192</td>
<td>—</td>
</tr>
<tr>
<td>GA</td>
<td>0</td>
<td>—</td>
<td>40.43</td>
</tr>
<tr>
<td>PSO</td>
<td>0</td>
<td>—</td>
<td>31.16</td>
</tr>
</tbody>
</table>

To evaluate the algorithm’s capability to handle large-scale problems, the case size was expanded five times, and both the improved and original GWO algorithms were applied to solve the problem. The results were obtained from three independent runs, as presented in Table 11. The findings demonstrate the superiority of the improved algorithm in tackling large-scale linear project-scheduling optimization problems.

Table 11. Results under the large-scale case of algorithms.

<table>
<thead>
<tr>
<th>Run Times</th>
<th>Improved GWO Algorithm</th>
<th>Original GWO Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>105.38</td>
<td>198.90</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>256.57</td>
</tr>
<tr>
<td>3</td>
<td>98.10</td>
<td>196.14</td>
</tr>
</tbody>
</table>

5.3. Model Validity Analysis

In the classical LSM, linear activities typically follow forward construction. However, this study aimed to develop a model that can optimize scheduling considering reverse construction activities. To validate the effectiveness of the model and analyze the effect of introducing reverse construction activities in the construction schedule, the following analysis scenario was designed. For the two scaled-up cases mentioned above, four scenarios were designed based on whether reverse construction activities were allowed, and the improved GWO algorithm was employed to solve them 20 times. The best solutions obtained are listed in Table 12, and the corresponding linear schedules for the four results are depicted in Figures 9–12.

Table 12. The best solution of the different scenarios.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Case Selection</th>
<th>Whether Reverse Construction Is Allowed</th>
<th>Best Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial case</td>
<td>No</td>
<td>112</td>
</tr>
<tr>
<td>2</td>
<td>Initial case</td>
<td>Yes</td>
<td>91.29</td>
</tr>
<tr>
<td>3</td>
<td>Large-scale case</td>
<td>No</td>
<td>112.19</td>
</tr>
<tr>
<td>4</td>
<td>Large-scale case</td>
<td>Yes</td>
<td>98.10</td>
</tr>
</tbody>
</table>

In initial-scale case, scenario 1 only allows forward construction of activities, and according to Algorithm 1, its minimum project duration is 112 days. The construction schedule is shown in Figure 9. Scenario 2 allows activities to adopt reverse construction methods. As shown in Figure 10, due to the cancellation of construction direction restrictions, all linear activities have some subactivities (A2, C3, D3, E2, F2, G2, and H2) that adopt reverse construction methods, which successfully shorten the project duration to 91.29 (see Table 12), achieved a 18.5% reduction. In large-scale case, similar conclusions could be obtained. The project duration was shortened from 112.19 days to 98.1 days due to the introduction of reverse construction method. Evidently, in both scaled-up cases, the inclusion of reverse construction scenarios led to further optimization of the results.

The reason for the improvement of project duration is that the introduction of reverse construction brought greater spatiotemporal flexibility, such that activities could be arranged more flexibly without violating logical constraints.
construction activities. To validate the effectiveness of the model and analyze the effect of introducing reverse construction activities in the construction schedule, the following analysis scenario was designed. For the two scaled-up cases mentioned above, four scenarios were designed based on whether reverse construction activities were allowed, and the improved GWO algorithm was employed to solve them 20 times. The best solutions obtained are listed in Table 1, and the corresponding linear schedules for the four results are depicted in Figures 9–12.

In initial-scale case, scenario 1 only allows forward construction of activities, and according to Algorithm 1, its minimum project duration is 112 days. The construction schedule is shown in Figure 9. Scenario 2 allows activities to adopt reverse construction methods. As shown in Figure 10, due to the cancellation of construction direction restrictions, all linear activities have some subactivities (A2, C3, D3, E2, F2, G2, and H2) that adopt reverse construction methods, which successfully shorten the project duration to 91.29 (see Table 1), achieved a 18.5% reduction. In large-scale case, similar conclusions could be obtained. The project duration was shortened from 112.19 days to 98.1 days due to the introduction of reverse construction method. Evidently, in both scaled-up cases, the inclusion of reverse construction scenarios led to further optimization of the results.

The reason for the improvement of project duration is that the introduction of reverse construction brought greater spatiotemporal flexibility, such that activities could be arranged more flexibly without violating logical constraints.

Table 1. The best solution of the different scenarios.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Case</th>
<th>Whether Reverse Construction is Allowed</th>
<th>Best Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initial case</td>
<td>No</td>
<td>112</td>
</tr>
<tr>
<td>2</td>
<td>Initial case</td>
<td>Yes</td>
<td>91.29</td>
</tr>
<tr>
<td>3</td>
<td>Large-scale case</td>
<td>No</td>
<td>112.19</td>
</tr>
<tr>
<td>4</td>
<td>Large-scale case</td>
<td>Yes</td>
<td>98.10</td>
</tr>
</tbody>
</table>

Figure 9. LSM diagram of the best solution of Scenario 1.

Figure 10. LSM diagram of the best solution of Scenario 2.
5.4. Managerial Insights

Further analysis on potential positive impact of introducing reverse construction on scheduling in terms of project duration is conducted in this section. Consider the subactivity $j_l$ and its preceding subactivity $i_h$. Site managers may consider the following methods for shortening the duration.

1) If $tp_i = \text{line}, tp_j = \text{line}, SD_{l,k} < SD_{j,l}$, $ED_{l,k} < ED_{j,l}$, $ED_{l,k} > SD_{j,l}$, and $pr_{l,k} < pr_{j,l}$.
   1. When both $i_h$ and $j_l$ choose reverse direction, activity $j_l$ achieves an earlier start time compared with when both constructions are forward.
   2. When $(SD_{l,k} - SD_{l,k})/pr_{l,k} > (ED_{l,k} - SD_{j,l})/pr_{j,l}$, $i_h$ selects reverse direction and $j_l$ selects forward direction, $j_l$ achieves an earlier start time compared with when both constructions are forward.
   3. When $BD_{l,k} > BD_{l,k} - SD_{l,k}$, $i_h$ selects forward direction and $j_l$ selects reverse direction, $j_l$ achieves an earlier start time compared with when both constructions are forward.
(2) If \( tp_i = \text{line}, \) \( tp_j = \text{line}, \) \( SD_{ij,k} \geq SD_{ij,l}, \) \( ED_{ij,k} \leq ED_{ij,l}, \) and \( pr_{ij,k} < pr_{ij,l}. \)

1) When \( ED_{ij,l} - SD_{ij,k} > ED_{ij,k} - SD_{ij,l}, \) \( i_k \) and \( j_l \) both choose reverse direction, \( j_i \) achieves an earlier start time compared with when both constructions are forward.

2) When \( ED_{ij,k} - ED_{ij,l} > ED_{ij,l} - SD_{ij,l}, \) \( i_k \) selects forward direction and \( j_l \) selects reverse direction, \( j_i \) achieves an earlier start time compared with when both constructions are forward.

3) If \( tp_i = \text{line}, \) \( tp_j = \text{line}, \) \( SD_{ij,k} < SD_{ij,l}, \) \( ED_{ij,k} < ED_{ij,l}, \) and \( ED_{ij,k} > SD_{ij,l}. \)

1) When \( i_k \) and \( j_l \) both choose reverse, \( j_i \) achieves an earlier start time compared with when both constructions are forward.

2) When \( SD_{ij,l} - SD_{ij,k} > ED_{ij,k} - SD_{ij,l}, \) \( i_k \) selects reverse, and \( j_l \) selects forward, \( j_i \) achieves an earlier start time compared with when both constructions are forward.

3) When \( (SD_{ij,l} - ED_{ij,k})/ pr_{ij,k} > (ED_{ij,k} - ED_{ij,l})/ pr_{ij,l}, \) \( i_k \) selects forward construction and \( j_l \) selects reverse construction, \( j_i \) achieves an earlier start time compared with when both constructions are forward.

We may not be able to discuss all the scenarios here, but these interesting findings are meaningful for on-site scheduling. Moreover, by eliminating the ideas provided in this section, more situations and conclusions can be further analyzed and obtained.

6. Conclusions

In summary, this study focused on scheduling optimization in construction projects under the LSM framework and made the following contributions.

(1) For linear scheduling optimization, scenarios with reverse construction activities were considered. A systematic spatiotemporal logical constraint system consisting of 48 scenarios was proposed to fully describe the temporal and spatial logical relations faced by various reverse construction scenarios in linear project construction. To the best of our knowledge, this is the first study on the spatiotemporal logical relationship description method and constraint portrayal system for reverse construction activities under LSM. The proposed constraint system may serve as a general foundation for future research in this field;

(2) An LSM-based scheduling optimization model named LSM-CBPS was proposed. This model is compatible with the 48 types of reverse construction spatiotemporal logic constraints mentioned above, as well as other classic constraints such as crew and mode constraints, and achieves the flexible matching ability of multisection–multicrew, multicrew–multimode, and multicrew–multiconstruction directions. These practical scenario-oriented designs enhance the practicality of the model;

(3) The application of the GWO algorithm in scheduling optimization based on the LSM framework was explored. An improved GWO algorithm with four-layer coding rules, discrete position update operators, dynamic constraint processing, cross-mutation operators, and heuristic rules was designed. This algorithm demonstrated superior performance compared with other algorithms, i.e., original GWO, GA, and PSO, in terms of solution quality, stability, and efficiency. To the best of our knowledge, this is the first application of GWO-based algorithm in linear scheduling optimization.

The effectiveness of the proposed model and algorithm was verified through a practical railway construction project, thereby highlighting its advantages in handling large-scale optimization problems and generating construction schedules with optimized durations.

However, this study still has some limitations. It considers only the optimization of construction duration but does not address other objectives such as resource leveling. Multiobjective optimization has received widespread attention in various fields [39], however, considering the characteristics of the GWO algorithm, this study focuses on single objective optimization and does not address the impact of reverse construction activities scenarios on multiple objectives synchronous optimization. Furthermore, the construction process of the construction crew is simplified, thereby neglecting various uncertainties that occur in real construction scenarios. Future research should address these limitations and explore additional factors to further improve construction scheduling optimization.
Author Contributions: Conceptualization, Y.T. and Z.Y.; methodology, Y.T. and Z.Y.; validation, Z.Y., C.W. and Z.H.; formal analysis, Z.Y., Y.Z. and Z.H.; data curation, C.W. and Y.Z.; writing—original draft preparation, Z.Y.; writing—review and editing, Z.Y. and C.W.; visualization, Z.Y. and Y.Z.; supervision, Y.T.; project administration, Y.T.; funding acquisition, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fundamental Research Funds for Central Universities (grant number 2023JBMCO07), the National Natural Science Foundation of China (grant numbers 71801010 and 62132003) and the Science and Technology Research and Development Plan Project of China National Railway Group Co., Ltd. [Grant Number N2022G025].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created.

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

References

2. Terzis, D. Monitoring innovation metrics in construction and civil engineering: Trends, drivers and laggards. Dev. Built Environ. 2022, 9, 100064. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.