Updating Linear Schedules with Lowest Cost: a Linear Programming Model

Sławomir Biruk¹, Piotr Jaśkowski¹, Agata Czarnigowska¹
¹ Lublin University of Technology, Faculty of Civil Engineering and Architecture, Department of Construction Methods and Management, ul. Nadbystrzycka 40, 20-618 Lublin, Poland
a.czarnigowska@pollub.pl

Abstract. Many civil engineering projects involve sets of tasks repeated in a predefined sequence in a number of work areas along a particular route. A useful graphical representation of schedules of such projects is time-distance diagrams that clearly show what process is conducted at a particular point of time and in particular location. With repetitive tasks, the quality of project performance is conditioned by the ability of the planner to optimize workflow by synchronizing the works and resources, which usually means that resources are planned to be continuously utilized. However, construction processes are prone to risks, and a fully synchronized schedule may expire if a disturbance (bad weather, machine failure etc.) affects even one task. In such cases, works need to be rescheduled, and another optimal schedule should be built for the changed circumstances. This typically means that, to meet the fixed completion date, durations of operations have to be reduced. A number of measures are possible to achieve such reduction: working overtime, employing more resources or relocating resources from less to more critical tasks, but they all come at a considerable cost and affect the whole project. The paper investigates the problem of selecting the measures that reduce durations of tasks of a linear project so that the cost of these measures is kept to the minimum and proposes an algorithm that could be applied to find optimal solutions as the need to reschedule arises. Considering that civil engineering projects, such as road building, usually involve less process types than construction projects, the complexity of scheduling problems is lower, and precise optimization algorithms can be applied. Therefore, the authors put forward a linear programming model of the problem and illustrate its principle of operation with an example.

1. Introduction

During the construction stage of a project, the contractor’s operative planning involves mitigation of effects of unforeseen occurrences. If the actual performance strongly deviates from the baseline schedule, threatening project milestone dates, the schedule is to be updated. As delays are the most frequent type of deviations, updating implies that durations of some activities are to be reduced to meet the project completion date. Bakry, Moselhi and Zayed [1] list project crashing measures: these include introducing overtime work, weekend work, shift work, and replacing current resources with resources of higher capacities. Crashing particular processes, or whole process chains, may be achieved by moving resources from less critical tasks (or, in accordance with the linear scheduling terminology, from tasks that do not belong to the controlling activity paths [2]) to reinforce resources devoted to tasks that are critical for the project’s being delivered on time.
The paper investigates the problem of selecting optimal crashing measures for a linear project. Optimization is considered in terms of cost and scale of duration reduction. The literature on the subject [1, 3] presents iterative methods that do not guarantee finding optimal solutions.

2. Linear projects and linear scheduling

Proper selection of planning and control methods increases chances of completing the project on time, within budget and with expected quality [4]. Particularity of linear projects, such as construction of roads that consist in conducting repeatable processes in consecutive sections gave rise to specific scheduling techniques. Tools widely used in general construction (critical path method, bar charts) are claimed to be insufficient as they do not directly account for continuous resource employment even if resource levelling techniques are applied [5]. Though Russel and Wong [6] criticize continuous resource utilization as the main scheduling criterion, it is important due to losses generated by interruptions of work flow and resource idle time.

Linear projects in civil engineering, such as construction of tunnels, roads, or laying pipelines, are naturally broken down into sections, with the dividing points being interchanges, revision shafts, etc. [7,8]. Process durations may differ section to section due to different operating conditions (such as soil type), design differences (e.g. road section is wider at curve than at tangent). Resulting variability of resource output per unit of length of the structure may result in discontinuity of work, reduced plant productivity, higher cost and longer project duration.

A convenient method of presenting work flow in a linear project is the time-location diagram. Figure 1 is an example of such diagram, with its interpretation presented below.

![Figure 1. Time-location diagram of a linear project](image)

Let’s assume that the schedule presented in Figure 1 represents processes related with construction of a road. The road is divided into five sections of the same length, and the works involve 10 processes to be conducted by separate crews/machine sets. Processes 1 and 2 are scheduled to be completed continuously along consecutive sections of the road with the same productivity, which is reflected by the steady slope of each line. Process 3 is also completed continuously at a steady rate, but the rate is lower than in the case of processes 1 and 2. By modification of resource composition (crew and plant), these three processes can be fully harmonized so that they take exactly the same time. In the case of Process 4, production rates differ section to section, which may be due to different working conditions or quantity of work in consecutive sections – thus the time for completing this process in each section is different. Process 5 is being conducted by two separate crews that start at the same date from the opposite ends of the road and work along the route towards each other; the crews work with different productivity (the time for completing two sections by one crew is longer than the time for completing three sections by the other crew). Process 6 is completed at one particular location (at the point dividing Section 3 and Section 4) – this process may consist in e.g. construction of a culvert. Process 7 is to be conducted along all sections but Section 3, whereas Process 8 is necessary only in Section 3. Process 9
consists in works that are conducted at the same time in two sections. Process 10 is scheduled to be interrupted for some time as Section 2 is completed, and after a break it starts again along the remaining sections.

Process start and finish dates usually need to fulfil certain conditions. Figure 2 depicts a project that involves three processes (\(A, B, C\)). Assuming that, for reasons related with e.g. construction methods, Process \(B\) (e.g. placing base course on a sub-base) cannot start earlier than \(f_{ab}\) days after completing Process \(A\) (e.g. cement bound sub-base that needs time to gain enough strength). Similarly, Process \(C\) cannot start earlier than \(f_{bc}\) after completion of \(B\). The processes are scheduled with different productivity rates. \(B\), scheduled originally at a quite high production rate, which might be costly, can as well be scheduled as \(B_1\) – so started earlier, but requiring less resources (or resources that are less productive and thus cheaper) than in the case of \(B\). Such slowing down Process \(B\) may have no negative effect on project duration, but have a positive effect on the budget. In Figure 2, a shortest lag between starts of Process \(B\) and Process \(C\) is \(f_{bc}\). If Process \(B\) is scheduled in its “earlier but slower” option of \(B_1\), and if Process \(C\) for some reasons has a fixed production rate depicted by the slope of the line \(C\), then Process \(C\) can be shifted to an earlier start date (represented by the dotted line \(C_1\)) \(f_{bc}\) after start of \(B_1\) in Section 1 – with an obvious benefit of earlier project completion. This way, in certain circumstances, slowing down some processes may result in reducing project cost and duration, with no harm to the logical constraints.

![Figure 2. Compressing schedule by reducing productivity rate of a process](image)

A different case is presented in Figure 3. The smallest acceptable lag between process \(A\) and \(B\) is \(f_{ab}\). The initial assumption on resources available for processes \(A\) and \(B\) result in certain productivity rates represented by the slopes of lines depicting these processes. To meet the lag constraint with the predefined productivity rates, the project requires long completion time. If the first process may be accelerated by employing more resources (\(A_1\)), then process \(B\) may be shifted to an earlier start (\(B_1\)) with no change to its resources’ productivity rate (\(B_1\)), the whole project may be completed earlier.

![Figure 3. Compressing schedule by increasing productivity rate of a process](image)
A classic approach to scheduling linear projects is mainly associated with graphical representation of schedules by means of time-location diagrams – very concise compared with traditional bar charts. However, going beyond graphics is necessary to make the approach useful. Many authors strive to develop the way of mathematical formulation for linear scheduling problems.

Harmelink and Rowings [2] proposed a method for defining project controlling paths. By reference to critical paths in CPM, the controlling paths are sequences of processes that need to be completed strictly according to the schedule if the project is to meet its deadlines. The project may involve processes that are linear in character (they are to proceed along a route), as well as block activities (conducted over larger areas for some time) and bar activities (that occur at particular locations). In contrast to CPM, the control chain may encompass only fragments of processes, what better corresponds to the character of a linear project. If CPM were to be applied, continuous processes would have to be split into arbitrarily divided sections that become separate CPM tasks – and the actual criticality of such arbitrarily defined tasks can be questioned. In contrast, the contents of the controlling path of a linear schedule is unambiguous, and is a good basis for analysing process floats and optimization of the schedule in the function of resources. A similar way of identifying “critical processes” in the case of sections of variable size (workload) was proposed by Harris and Ioannou [9].

Mattila and Abraham [4] presented a method for resource levelling dedicated for linear projects based on the idea of controlling paths. They used binary linear programming technique. Levelling was based on modification of productivity rates of crews that complete processes outside the controlling paths. The same problem was analysed by Gregory [10], who applied genetic algorithm to find optimal solutions.

El-Rayes and Moselhi [5] developed a two-stage iterative algorithm for scheduling linear projects to minimize resource idle time. Their method accounts for limited resource availability that is fixed within time windows. Resource availability constraints are of key importance in practical situations, as resources (crews, subcontractors) may be employed to serve a number of projects. This algorithm allows also for selecting one of available resources to complete a particular task: these resources differ in productivity, which means that duration of works related with a particular process in the same section are different according to the resource employed.

Methods of optimization in linear scheduling under a variety of constraints are vigorously developed in Russia [11] and Poland [12, 13]. The approach is known under “stream scheduling” and “time-couplings”. Among others, Rogalska, Bożejko and Hejducki [14] combine linear scheduling and theory of constraints and propose a hybrid evolutionary algorithm to find best time/cost tradeoff.

3. Mathematical formulation of the crashing problem

The authors assume a deterministic approach to problem of finding optimal crashing measures for a linear project. Optimization is considered in terms cost and scale of duration reduction.

Let us assume that repetitive processes of \( i \)-type that belong to the set of \( I = \{1, 2, \ldots, n\} \) are to be conducted in each section \( j \), \( j \in J \), \( J = \{1, 2, \ldots, m\} \) of a linear project. Each process is to be executed by its unique crew/machine set. The sequence of processes at each section is defined by a directed graph \( G_j = (I, A_j) \), with a single start node and a single end node. \( I \) represents the set of graph nodes (so the set of processes), and \( A_j \subseteq I \times I \) is a set of arches that connect the nodes; arches represent relationships between processes. The type of all relationships is “start process \( b \) no earlier than \( f_{ab} \) days after its direct predecessor, process \( a \), has been completed”. While defining the lag, \( f_{ab} \), one is to account for the time for completing \( a \) in a working area (fragment of a section) physically occupied by a crew that conducts
process \(a\) and the area physically occupied by the crew that conducts process \(b\). This way, one avoids the situation when two crews work in the same location disturbing each other.

The sequence of sections for conducting a process \(i\) is defined by a permutation \(\pi_i(j) = c_{i,j}\).
Processes run according to an initially defined pattern: along the route from the beginning of a section towards its end, or in the opposite direction. Each section has to be represented in \(\pi_i(j) = c_{i,j}\) even if a particular process is not to be conducted in this section; this has no effect on the optimization results.

For each process \(i\), a set \(W_i\) of optional methods of crashing is defined. The sets \(W_i\) include also the initially assumed way of conducting the processes. The selection of options is modelled by means of a binary variable \(x_{i,j,w} \in \{0,1\}\). This variable equals 1 if the process \(i\) in section \(j\) is to be conducted according to the option \(w\), \(w \in W_i\). The variable equals 0 if an option other than \(w\) is selected.

The time \(t_{i,j,w}\) and cost \(k_{i,j,w}\) of each process \(i\) to be conducted in the section \(j\) according to each option \(w\) is to be calculated on the basis of quantity take-offs, information on resource unit consumption rates, and resource prices. If a particular process is not to be conducted in a particular section, its time and cost would be 0.

The optimal options of process execution, and resulting start dates of processes in particular sections, \(s_{i,j}\), with a predefined time for completion for the whole project, \(T\), can be defined by solving the following model:

\[
\min z: \quad z = \sum_{i \in I} \sum_{j \in J} \sum_{w \in W_i} k_{i,j,w} \cdot x_{i,j,w}, \quad (1)
\]

\[
s_{i,j} = 0, \quad j : c_{i,j} = 1, \quad (2)
\]

\[
t_{i,j} = \sum_{w \in W_i} t_{i,j,w} \cdot x_{i,j,w}, \quad \forall i \in I, \quad \forall j \in J, \quad (3)
\]

\[
s_{i,j} + t_{i,j} \leq T, \quad \forall i \in I, \quad \forall j \in J, \quad (4)
\]

\[
s_{b,j} - s_{a,j} \geq f_{a,b}, \quad \forall (a, b) \in A_j, \quad \text{and of the same direction of work flow, } \forall j \in J \quad (5)
\]

\[
s_{b,j} - s_{a,j} - t_{a,j} \geq f_{a,b}, \quad \forall (a, b) \in A_j, \quad \text{and of the same direction of work flow, } \forall j \in J \quad (6)
\]

\[
s_{b,j} - s_{a,j} - t_{a,j} \geq f_{a,b}, \quad \forall (a, b) \in A_j, \quad \text{and of the opposite work flow, } \forall j \in J \quad (7)
\]

\[
s_{i,j} = s_{i,k} + t_{i,k}, \quad \forall i \in I, \quad \forall (k, l) : k, l \in J \land c_{i,j} = c_{i,k} + 1, \quad (8)
\]

\[
\sum_{w \in W_i} x_{i,j,w} = 1, \quad \forall i \in I, \quad \forall j \in J, \quad (9)
\]

\[
x_{i,j,w} \in \{0,1\}, \quad \forall i \in I, \quad \forall j \in J, \quad \forall w \in W_i, \quad (10)
\]

\[
s_{i,j} \geq 0, \quad \forall i \in I, \quad \forall j \in J. \quad (11)
\]

The objective function (1) minimizes the total cost of project completion. The first process starts at the date of 0 in the first section that requires it (2). Time for completing each process in each section is calculated according to (3). Completion of the project as a whole cannot be later than \(T\) (4). Start dates of the remaining processes are defined according to conditions (5)-(8), considering the logic of works defined by the graph \(G\), the predefined direction of work flow, the minimum lags between predecessors.
and direct successors, and the sequence of taking over the sections by crews defined by the permutation \( \pi_j(j) \) for each process. Condition (8) assures continuity of work of crews. Each process in each section can be conducted according to only one option – condition (9). The variables have to fulfil boundary conditions (10) and (11).

To allow for block and bar activities, conditions (2)-(9) need to be expanded by additional constraints. If a bar activity \( v \) is to be completed within time of \( t_v \), after the start of process \( u \), before the start of process \( t \), at the location in the border between section \( g \) and \( h \), than the start date \( s_v \) of this process is to meet the following conditions:

\[
\begin{align*}
    & s_v - \max \{s_{u,h}, s_{u,g}\} \geq f_{u,v}, \\
    & s_{t,h} - s_v - t_v \geq f_{a,v}, \quad \text{if process } t \text{ starts from the beginning of section } h \text{ towards its end}, \\
    & s_{t,g} - s_v - t_v \geq f_{a,v}, \quad \text{if process } t \text{ starts from the end of section } g \text{ towards its beginning},
\end{align*}
\]

(12), (13), (14)

\[
    t_v = \sum_{w \in W_v} t_{v,w} \cdot X_{v,w},
\]

(15)

where \( t_{v,w} \) stands for the completion time of bar process \( v \) conducted according to option \( w \), and \( X_{v,w} \) is a binary variable that models the decision on selection of a particular option. As for block activities (processes to be executed at the same time in an area covered by one or more sections), conditions similar to (12)-(15) should be defined for borders of these sections.

### 4. Illustrative example

To illustrate the proposed method of finding optimal way to compress linear schedules in deterministic conditions, it was applied to a notional case of a linear project. There were four sections and six processes to be considered; one of the latter was a bar activity (process 6). Figure 4 is the time-location diagram of the baseline schedule with 66 days of time for completion. Table 1 lists input values used to construct the schedule.

![Figure 4. Baseline schedule – time-location diagram of the illustrative case.](image_url)

**Table 1. Sequence of linear processes (illustrative case)**

| Process | Direct predecessor | \( f_{a,b} \) | Permutation of sections |
|---------|-------------------|--------------|------------------------|
| \( j=1 \) | \( j=2 \) | \( j=3 \) | \( j=4 \) |
| 1 | | | 1 | 2 | 3 | 4 |
| 2 | 1 | 5 | 1 | 2 | (3) | (4) |
| 3 | 1 | 5 | (4) | (3) | 2 | 1 |
| 4 | 2 | 10 | 1 | 2 | 3 | 4 |
| 5 | 4 | 5 | 1 | 2 | 3 | 4 |
Linear processes 1, 2, 4, 5 were scheduled to proceed along the route, from its beginning towards the end, with process 2 to be completed only in two sections. Process 3 is planned to be started at the end of the route and progress towards its beginning across sections 4 and 3. The bar activity (process 6), not listed in Table 1, was to be conducted at a location between sections 2 and 3, and it was scheduled to start as soon as process 2 had been completed in all sections.

The aim of optimization was to compress this schedule from 66 to no more than 55 days at the lowest cost, assuring that resources work in a continuous way. There existed three options of execution of each linear process. The first option was adopted in the baseline schedule. The remaining two consisted in working overtime, and hiring more resources to reinforce the crews, both meant to shorten the completion time, but coming at an additional cost. As for the bar activity, process 6, only one option was available; its duration was 10 days and its cost was fixed, though start date was negotiable (process 6 may represent subcontracted works with the contract already signed).

Tables 2 and 3 list values of duration and cost of linear processes at particular sections as to be conducted according to possible options of process execution.

**Table 2.** Durations $t_{i,j,w}$ of processes at consecutive sections for three considered options, values expressed in days (illustrative case).

| $i$ | Base option $w=1$ | Working overtime $w=2$ | Extra resources $w=3$ |
|-----|-------------------|------------------------|----------------------|
|     | $j=1$ | $j=2$ | $j=3$ | $j=4$ | $j=1$ | $j=2$ | $j=3$ | $j=4$ | $j=1$ | $j=2$ | $j=3$ | $j=4$ |
| 1   | 5     | 7     | 5     | 5     | 4     | 6     | 4     | 4     | 3     | 5     | 3     | 3     |
| 2   | 4     | 5     | 0     | 0     | 3     | 4     | 0     | 0     | 2     | 3     | 0     | 0     |
| 3   | 0     | 0     | 4     | 4     | 0     | 0     | 3     | 3     | 0     | 0     | 2     | 2     |
| 4   | 8     | 10    | 8     | 8     | 6     | 8     | 6     | 6     | 7     | 8     | 7     | 7     |
| 5   | 6     | 8     | 6     | 6     | 5     | 7     | 5     | 5     | 4     | 6     | 4     | 4     |

**Table 3.** Costs $k_{i,j,w}$ of processes at consecutive sections for three considered options, values expressed in monetary units (illustrative case).

| $i$ | Base option $w=1$ | Working overtime $w=2$ | Extra resources $w=3$ |
|-----|-------------------|------------------------|----------------------|
|     | $j=1$ | $j=2$ | $j=3$ | $j=4$ | $j=1$ | $j=2$ | $j=3$ | $j=4$ | $j=1$ | $j=2$ | $j=3$ | $j=4$ |
| 1   | 10    | 14    | 10    | 10    | 11    | 15    | 11    | 11    | 12    | 16    | 12    | 12    |
| 2   | 12    | 13    | 0     | 0     | 13    | 15    | 0     | 0     | 15    | 16    | 0     | 0     |
| 3   | 0     | 0     | 12    | 12    | 0     | 0     | 13    | 13    | 0     | 0     | 15    | 15    |
| 4   | 20    | 24    | 20    | 20    | 22    | 28    | 22    | 22    | 24    | 31    | 24    | 24    |
| 5   | 12    | 15    | 12    | 12    | 13    | 16    | 13    | 13    | 16    | 19    | 16    | 16    |

The mathematical model described by equations (1)-(15) was fed with the above data and solved by means of Lingo 14.0, a modelling language and solver for linear, non-linear, and integer programming. The solution – a compressed schedule – is presented in Figure 5 in the form of a time-location diagram. Reduction of the project duration to the required 55 days occurred to be possible at an additional cost of 11 monetary units. In this case, crashing consisted in 1) conducting process 1 according to the third option – in all sections, and 2) conducting processes 3 and 4 according to the second option in section 4. The remainder of processes 4 and whole process 5 was left to be conducted as in the baseline option, but shifted to earlier start dates thanks to speeding up their predecessors.
Figure 5. Crashed schedule – result of optimization

It can be observed that processes 4 and 5 in sections 1 and 2 do not belong to the controlling path of the project and might be completed at a slower pace if started earlier. This possibility would be worth exploring if related with reduction of cost. The option of “slowing down” was not considered in the illustrative example for the sake of brevity, but is certainly worth taking into account in real-life cases.

5. Summary and Conclusions

Special character of linear (repetitive) projects calls for dedicated scheduling methods. Such methods are aimed at facilitating resource planning, in particular harmonizing work flows, making the most of resource capacities, and reducing cost by eliminating idle time and resource relocation. Linear projects in civil engineering (building roads, pipelines etc.), usually involve less process types compared with building projects, which reduces complexity of scheduling problems and, potentially, enables applying exact methods of optimization.

Any civil engineering and building projects are subject to risks. One expects delays due to bad weather, plant failure, etc. As contractors commit themselves to delivering projects in timely manner, if a delay occurs, they are forced to reschedule works. Methods that enable them to select most economical way to compress schedules are much in demand. Therefore, the authors propose an exact mathematical method for selecting best (cheapest) options for crashing a linear schedule under constraints of maintaining work flow continuity and meeting the deadline. If the number of processes and section is reasonably low, the model is simple enough to be solved by means of ready-made solvers.

Acknowledgment

This work was financially supported by the Ministry of Science and Higher Education in Poland under the statutory research number S/63/2017. The authors gratefully acknowledge the support.

References

[1] I. Bakry, O. Moselhi, and T. Zayed T., “Optimized acceleration of repetitive construction projects”, *Autom. Constr.*, vol. 39(1), pp.145-151, 2014.
[2] D.J. Harmelink, and J.E. Rowings, “Linear scheduling model: development of controlling activity path”, *J. Constr. Eng. M. ASCE*, vol. 124(4), pp. 263-268, 1998.
[3] A. Hassanein, and O. Moshelhi, “Accelerating linear projects”, *Constr. Manage. Econ.*, vol. 23(4), pp. 377-385, 2005.
[4] K.G. Mattila, and D.M. Abraham, “Resource leveling of linear schedules using integer linear programming”, *J. Constr. Eng. M. ASCE*, vol.124(3), pp. 232-234, 1998.
[5] K. El-Rayes, and O. Moselhi, “Resource-driven scheduling of repetitive activities”, *Constr. Manage. Econ.*, vol. 16(4), pp. 433-446,1998.
[6] A.D. Russell, and W.C.M. Wong, “New generation of planning structures, *J. Constr. Eng. M. ASCE*, vol. 119(2), pp. 196-214, 1993.
[7] J.D. Lutz, and A. Hijazi, “Planning repetitive construction: current practice”, *Constr. Manage.*
Econ, vol. 11, pp. 99-110, 1993

[8] O. Moselhi, and A. Hassanein, “Optimized scheduling of linear projects”, J. Constr. Eng. M. ASCE, vol. 129(6), pp. 664-673, 2002.

[9] R.B. Harris, and P.G. Ioannou, “Scheduling projects with repeating activities”, J. Constr. Eng. M. ASCE, vol. 124(4), pp. 269-278, 1998.

[10] M.E. Georgy, “Evolutionary resource scheduler for linear projects, Autom. Constr., vol. 17(5), pp. 573-587, 2008.

[11] V.A. Afanasev and A. V. Afanasev, Поточная организация работ в строительстве (Stream scheduling of works in civil engineering), С-ПГАСУ, St. Petersburg, 2000 (Russian)

[12] Z. Hejducki, J. Mrozowicz, “Stream methods of construction work organisation: an introduction to the problem”, Eng., Constr. Archit. Manag., vol.8(2), pp.80–89, 2001.

[13] Z. Hejducki, “Scheduling model of construction activity with time couplings”, J. Civ. Eng. Manage., vol. 9(4), pp. 284–291, 2003.

[14] M. Rogalska, W. Bożejko, Z. Hejducki, “Time/cost optimization using hybrid evolutionary algorithm in construction project scheduling”, Autom. Constr., vol. 18(1), pp. 24-31, 2008.