Towards the Automation of Last Planner System ®

Michael Awe[1], Avleen Malhi[1], Nicholas Mavengere[1], Marcin Budka[1] and Bhargav Dave[2]

[1] Department of Computing and Informatics, Bournemouth University, UK
[2]VisiLean, UK

mawe@bournemouth.ac.uk, amalhi@bournemouth.ac.uk, nmavengere@bournemouth.ac.uk
mbudka@bournemouth.ac.uk, bhargav@visilean.com

Abstract -

The increasing complexity and uncertainty in construction projects necessitate more efficient and adaptive construction management systems. This research studies the automation of the Last Planner System (LPS), a collaborative lean scheduling tool, by integrating predictive strategies to enhance scheduling accuracy, constraint resolution, and insights for decision-making. A conceptual framework is proposed for LPS automation while also representing the automation of LPS as an optimisation problem and determining the core aspects of LPS, such as communication management, constraint management, and planning and scheduling. This framework provides a foundation to which integration of predictive strategies in digitised Last Planner Systems is possible, focusing on process improvement, delay mitigation, and performance optimisation. A model that determines the constraint resolution duration was presented. This model used feature extraction techniques to determine constraint complexity score and preprocessed other features such as priority, location, and category to predict constraint resolution duration. The results indicated a Mean absolute error of 21.38 days (MAE) from the actual constraint resolution duration. Although the outcome and model can help in constraint management towards an automated LPS, the research identifies key limitations, including challenges in dataset quality, unfavourable sample sizes, and challenges with parsing complex textual and tabular data, highlighting areas for improvement. Future research should prioritise exploring emerging techniques, such as Table parsing (TAPAS) and Retrieval-augmented generation (RAG), to enhance the integration of data-driven strategies into Last Planner System workflows.

Keywords -

Automation, Predictive Strategies, Last Planner System (LPS)

1 Introduction

In the face of increased complexities and uncertainties of projects in the construction sector, more efficient and intuitive project controls, planning, and scheduling systems are highly sought. Pressing factors contributing to inefficiencies in existing systems include inadequate management of constraints, communication, and lack of adaptability in scheduling and project control systems [1]. However, automatically classifying constraints, intuitively classifying project communication, instantly generating insights, and detecting deviations in these systems are crucial for navigating complex and high-risk projects such that predictability is enhanced and project outcomes are favourable [1, 2, 3]. To achieve this, a more adaptive project management system (Last planner system) is considered as prototyped by Ballard [4]. While existing project control and scheduling systems have indicated significant improvement in addressing inefficiency in project implementation, there remains a substantial gap for improvement in ascertaining project status, instant generation of insights from project trends, and ensuring predictable project outcomes [5, 6]. From the first principle, a last planner system (LPS) can, therefore, be represented as a combination of constraint management system (CN), collaboration/communication system (CB), and Planning and scheduling systems (PS), where x..n is the increasing project instances of elements contained that make each system functional;

$$LPS(x..n) \rightarrow CN(x..n)+CB(x..n)+PS(x..n) \qquad (1)$$

Improved efficiency in existing project control and scheduling systems depends on the integration of digitised project control and scheduling systems with other technologies and methods such as lean systems, visual simulation, BIM, artificial intelligence, AR VR, etc. [1, 5]. These integrations have been proposed as solutions to identified fundamental limitations, including unreliable schedules, unreliable data, manual processes leading to errors, and overarching challenges of delay, cost overrun, and reworks [7]. These challenges are expected for most projects, especially in complex and high-risk projects where project managers faced with trade-offs in cost, quality, time, resources, changes, etc., have to rely on tacit knowledge, experience, and intuition to make accurate decisions quickly rather than relying on insights from analysed operation data to draw conclusions and make data-driven decisions. Studies have shown that although experience and tacit knowledge are essential in managing projects, it is more reliable for businesses to find data-driven paths to making operational decisions to control current and future project outcomes [8, 9].

While predictive strategies can improve most of the existing digitised project control and scheduling systems, this study aims explicitly to automate digital last planner systems. LPS is a lean and collaborative tool that provides short-term schedules and lookahead information that enhances planning efficiency, improves the reliability and accuracy of plans, enables the selection of alternatives, and reduces time-space conflicts on construction sites. The system pulls information from all forms of communication regarding constraints, deviations, changes, etc., to determine tasks to be scheduled within specific lookahead periods. The benefits of LPS are evident in LPS-related research sources, and an automated Last Planner System (LPS) has been proposed to exert an even greater impact on construction planning and implementation processes. [10, 11, 7, 12, 13].

As part of the contribution of this research, this study presents a streamlined flow for automating the digitised Last planner system, considering all factors which enable it to be fully functional. In achieving this aim, the study follows a procedure that allows the identification of all relevant elements and features of a last planner system, represents the system as an optimisation problem and integrate predictive strategies in one of the identified elements (constraint management). Section 1 presents a summary of the conceptualisation of the topic, and Section 2 analyses existing literature and discusses the gaps and limitations. Section 3 presents the flow for the proposed automated last planner system. Section 4 initialises a basis for integration of artificial intelligence and presents a representation of the problem mathematically. Section 5 showcases the predictive strategies and offers implementation and discussion of results. finally, Section 6 and 7 discusses the research outcomes, limitations and future works.

2 Literature Review

The last planner system (LPS) is a project control and scheduling system created in the early 90s by Glen Ballard [4]. Research highlights that LPS emerged by applying lean construction (LC) principles in construction projects. It is a system based on the need to enhance workflow reliability by field workers implementing the task specified in planning processes. Key elements of LPS identified from existing research include 1) collaborative planning, 2) constraint management, 3) hierarchy planning systems and levels, 4) pull planning, and 5) continuous improvement [1, 3, 2].

Challenges of LPS were first discussed expansively by Aslam et al., [7]. The shortcomings identified include the inability to manage project variability correctly, lack of visualisation of LPS capabilities, incorrect lookahead planning and many more [14]. However, The conceptual framework for integrated LPS included necessary features such as collaboration, constraint identification and communication methods, which led to increased collaboration, improved visualisation, risk analysis, continuous improvement, etc. The implemented digitised LPS also has shortcomings, even though it is digitised. These shortcomings include limited support and integration, lack of consistency in planning and scheduling, inadequate real-

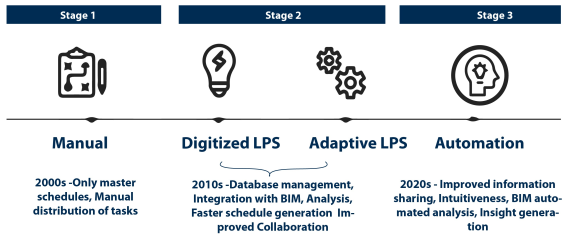

Figure 1. Last planner transformation stages

time information, data management challenges, and limited constraint management [2]. A comprehensively automated digital LPS is expected to improve efficiency in all processes and activities, enhance constraint management, allow real-time data analysis and visualisation, allow knowledge reuse and to improve collaboration and communication through streamlining processes and enhancement of information sharing.

Developing a workable automated planning system is predicated on identifying all required base elements that make the system automated, collaborative, and manage constraints. Automation, as identified in existing research, is indicated in 3 levels; the first level is research based on the digitisation of LPS systems as against the manual forms [4]; the second level is the research instances which developed linkages and union nodes for BIM and digitised LPS as an upgrade to overcome data homogeneity and enhance information sharing [15, 16, 12]; the third level is the point where predictive strategies and AI techniques are being integrated into digitised LPS platforms to aid automation and streamline processes. [2]

A review study by Agrawal et al. [1] identified several existing research where automated planning processes were categorised into master and phase scheduling, lookahead scheduling, weekly work planning and learning stage. Although the study excluded actual automation of planning and control in LPS, it systematically reviewed existing works. The research identified research works which presented automation in all three identified levels in Figure 1. Further reviews and classification of related works are summarised in Table 1.

Overall, the early research by Ballard [4] has since been transformed to become more encompassing from the manual linear scheduling methods to the digitised methods. Then, the digitised systems that link LPS with BIM are often integrated for automation, allowing for improved scheduling of activities and streamlining processes. Existing related research has focused on other specific applications and stages of projects but neglected the three core aspects of a functional LPS. This is the gap which this study seeks to cover. More specifically, automation in the Last Planner system has been applied mainly in performance evaluation. A framework or prototype of an integrated LPS system that considers all the factors and

Table 1. An overview of existing literature

Sections	Overview	Stage	Citations
Last planner sytems	LPS concept application manually using pilot projects	Ist	[4]
Schdeuling Automation Review	A systematic review of existing works.		[1]
Master and phase scheduling	Optimisation of buffer projects used varying optimisation techniques to achieve maximised schedule allocations	2nd	[17, 18, 19]
Safety, risk and logistics constraints	Constraints that may impact projects using BIM as a source of information	2nd	[20, 21]
Lookahead planning	Automation-driven research which used transformer models to predict lookahead and generate plans quickly	2rd	[22, 3]
Constraints management	Identification and removal of constraints by first achieving integration between different data sources	3rd	[23, 24, 25, 26]
Last planner system (performance)	Used machine learning techniques to forecast and predict performance using identified LPS-based performance indicators such as the Percent plan complete (PPC) and other indicators.	3rd	[2]
Integrated last planner system ILPS	helps smoothen the implementation process and overcome the shortcomings identified in the LPS.	2nd	[12, 7, 13]

features of an LPS system appears non-existent, motivating the need for this study.

3 Last planner system Automation

Project control and scheduling systems ensure the flow of work and control the progress and outcomes of the projects. As shown in Figure 2, a typical project will be guided by a developed master schedule, which serves as input detail to other scheduling stages, including phase scheduling, lookahead, weekly work scheduling and daily scheduling. Once the master schedule exists, LPS principles can be integrated into processes, including short-term planning and lookahead. Figure 2 contains the flow chart for logical automation of Last Planner systems.

3.1 Lookaheads

In lookahead plans or short-term plans, determining what tasks will be ready to start in a lookahead period is very important. It can be based on the probability that all tasks before this period are completed [27]. The probability of task completion can be calculated using historical productivity rates. Measuring the percentage of work completed (PPC) relative to the time spent on a similar previous task provides a more accurate estimate of productivity rates than relying on generalised project or organisational productivity rates, as suggested by Salama et al., [27]. Their method involves determining performance based on forecasted productivity rates (using linear regression) and the remaining quantity of work. However, this approach assumes that the same organisation's employees consistently perform the tasks across all projects, which is often untrue. In reality, various contractors frequently complete tasks, each contributing to different aspects of the work. This variability complicates accurately determining productivity rates, especially on-site, where numerous contractors are involved in diverse activities. Consequently, there is a pressing need to maintain detailed performance data so that individual contractors can evaluate their work completion speeds better.

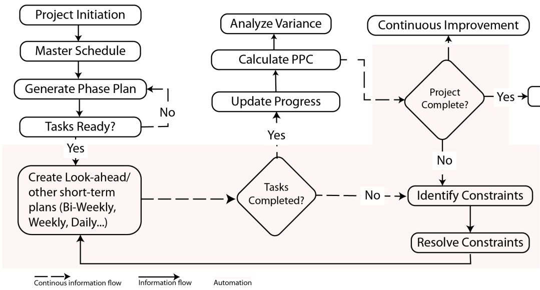

Figure 2. Proposed automated Last planner systems

4 LPS Aspects For AI Integration

As described already in equation 1, an LPS, on a fundamental basis, is a combination of three elements, which include collaboration, constraints and planning systems. This entities are further elaborated below;

AS in figure 3, the Project schedule contains a list of tasks (LT) detailing the following features: Task Name TN_I .. TN_n, a unique identifier for each task. Location L_i .. L_n, the location identifier for each task. Trade: CT_i .. CT_n, the construction trade responsible for each task. Scheduled start and end dates: $t_{s,i}$ and $t_{e,i}$ for tasks TN_i, Duration: $D_{t,i} = t_{e,i} - t_{s,i}$, the duration of each task.

Constraint Table (CN) contains project constraints and their features: Constraint Name C_i .. C_n, a unique identifier for each constraint. Location L_i .. L_n, location identifier where the constraint exists. Trade CT_i .. CT_n, the trade associated with the constraint. Commitment start and end dates cts_i and cte_i. Constraint resolution time $D_{c,i} = cte_i - cts_i$, the time to resolve the constraint. finally, the tables LT and CN are linked through the linked task names, which associates constraints with corresponding tasks (TN_i).

Problem Definition

Given the tables LT and CN, we aims to optimize the inclusion of tasks ($\text{TL} \subset \text{LT}$) into a specified Lookahead Period (LAP). The goal is to dynamically determine an optimal subset of tasks while resolving associated constraints and efficiently allocating resources, subject to the following conditions:

- Binary Decision Variables (x_i); $x_i = 1$ if task i is included in the Lookahead Task List (TL), and $x_i = 0$ otherwise.
- Tasks left with status (ready) are well-defined for scheduling.
- All constraints are valid and removed for tasks ready for scheduling.
- The Lookahead Period is defined by the Lookahead Start Time ($\text{LAT}_{\text{start}}$) and Lookahead End Time (LAT_{end}).
- Tasks and constraints are resolved and aligned with resource availability and stakeholder collaboration.

Schedule (List of tasks)_LT

Task Name	Labels	Responsi	Status	Location	Trade	Quantity	Start	End	Duration	Task_Type	Priority	Notes
TN_i	.	.	.	L_i	CT_i	.	ts_i	te_i	Dt_i	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
TN_n	.	.	.	L_n	CT_n	.	ts_n	te_n	Dt_n	.	.	.

Constraints_CN

Constraint	Trade	Responsible	Location	Priority	Category	Commitment Date	Completion Date	Duration	Notes	Linked TN
C_i	CT_i	.	L_i	.	.	cts_i	cte_i	Dc_i	.	TN_i
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
C_n	CT_n	.	L_n	.	.	cts_n	cte_n	Dc_n	.	TN_n

Figure 3. Standard Data-frame of the schedules and constraints entities.

Objective Function

As given in the problem definition, the objective function Z is to find the tasks TL in LT, which maximises Z.

amax Z =

$$\sum_{i \in \mathrm{TL}} w_i \cdot x_i \cdot D_{t,i} - \lambda_1 \cdot \mathrm{Pen}_{\mathrm{cons}} - \lambda_2 \cdot \mathrm{Pen}_{\mathrm{colab}} \tag{2}$$

Where w_i is priority weight for all tasks i (i.e., high = 3, medium = 2, low = 1), $D_{t,i}$ is duration of all tasks x_i, $\mathrm{Pen}_{\mathrm{cons}}$ is penalty for unresolved constraints and $\mathrm{Pen}_{\mathrm{colab}}$ is penalty for unresolved collaboration conflicts. Z is subject to the PS constraints, CN constraints, CB constraints and dynamic updates.

Schedule PS: This entity entails planning for different projects at all levels. Here, the critical dependencies, sequences, lookahead and buffers are represented accordingly.

- Task Dependency Constraint: tasks must satisfy their dependencies before being included in TL.

$$ts_i \geq te_j, \quad \forall (j \rightarrow i) \text{in dependencies.} \tag{3}$$

- Lookahead Time Period Constraint; Only tasks whose durations overlap with the Lookahead Period ($\mathrm{LAT}_{\mathrm{start}}$ to $\mathrm{LAT}_{\mathrm{end}}$) are considered

$$t_{s,i} \geq \mathrm{LAT}_{\mathrm{start}}, \quad t_{e,i} \leq \mathrm{LAT}_{\mathrm{end}} \tag{4}$$

- Lookahead Duration Constraint; The total duration of selected tasks must not exceed the Lookahead Duration (LAD)

$$\sum_{i \in \mathrm{TL}} x_i \cdot D_{t,i} \leq \mathrm{LAD} \tag{5}$$

- Buffers: This justifies the necessary lack of periods needed for selected tasks.

$$t_{e,i} + \mathrm{buffer}_i \leq t_{s,j}, \quad \forall (i \rightarrow j) \tag{6}$$

Constraints CN: This entity manages issues and emerging risks to enhance quicker constraint resolutions by teams and domains for different projects. Sometimes, tasks are usually hard-constrained. However, this component is primarily controlled by responsible persons. Sometimes, the impediment may be forced removed or managed to get the task completed. In the event of force removal or manoeuvres, the system should be updated to ensure the continuous correctness of scheduled tasks.

- Status constraint; The status constrain represents the readiness indicator for each task x_i, we introduce a binary variable R_i:

$$R_i = \begin{cases} 1, & \text{if all constraints for task } x_i \text{ are resolved} \\ 0, & \text{otherwise} \end{cases}$$

- Resource Constraint; To ensure that the selected tasks do not exceed resource limits for materials M and workers W.

$$\sum_{i \in \mathrm{TL}} x_i \cdot m_i \leq M_{\mathrm{available}} \tag{7}$$

$$\sum_{i \in \mathrm{TL}} x_i \cdot w_i \leq W_{\mathrm{available}} \tag{8}$$

- Space Constraint (S): Only tasks with available space for the job are marked ready to be scheduled. i.e available workspace $S(avail)$ must be sufficient for the selected tasks.

$$\sum_{i \in \mathrm{TL}} x_i \cdot S_i \leq S_{\mathrm{avail}}, \quad \forall t \in [\mathrm{LAT}_{\mathrm{start}}, \mathrm{LAT}_{\mathrm{end}}] \tag{9}$$

Collaboration CB: This entity represents communication and cooperation among teams (planners, quality, design, engineering, supply chain, etc.) and domains to pre-empt and resolve potential conflicts and plan for projects to withstand emerging risks.

For a given project schedule PS, $N_i k$: Notes in LT and CN tables are associated with task x_i and constraint vector k. $CN_i k$: The k-th constraint vector in the constraint table.

The collaborative conflict for the Lookahead task list (TL) is defined as:

$$\mathrm{Pen}_{\mathrm{colab}} = \sum_{i \in \mathrm{TL}} \sum_{s} x_i \cdot (CN) \cdot \alpha(N_{i,k}) \tag{10}$$

Other than constraint CN, Each constraint type (e.g., safety, design, quality) is explicitly modelled as a separate binary

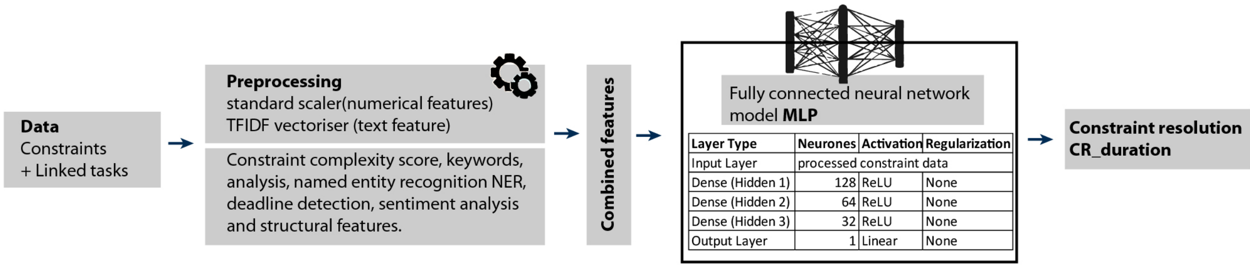

Figure 4. Model Architecture - Fully connected neural network

variable $Sf_{i,k}$ safety for task x_i:

$$Sf_{i,k} = \begin{cases} 1, & \text{if constraint of type } k \text{ for task } i \text{ is resolved} \\ 0, & \text{otherwise} \end{cases}$$

Dynamic Updates

At each time change t of a project, there are consequent changes in the amount of work completed, PPC, constraints, and communication, as these entities are not static but usually evolving. Based on this, the updated Lookahead plan must evolve. Hence, we perform rolling updates for every new iteration as the final constraint to the objective function:

$$\text{LAT}^{(t+1)}_{\text{start}} = \text{LAT}^{(t)}_{\text{end}} + \Delta T \tag{11}$$

Recompute lookahead plan task list TL dynamically. TL accuracy lies in the accuracy of the data that is fed into the process.

$$TL_{accuracy} \simeq \begin{cases} Pr(y_1) = 1 & \text{if upcoming tasks;} \\ CN_{x_i} = 0 & \text{no constraint exist;} \\ Pr(x_n) = 1 & \text{if previous tasks;} \end{cases}$$

where x_n are work tasks x_i before specific lookahead interval and y_1 are WTs within the lookahead or short-term interval. The performance indication will then be the range of deviation of actuals from planned, estimated or predicted values.

5 Implementation

As illustrated in Figure 5, advanced deep learning techniques, including Table Parsing (TaPas), deep implicit layers, self-querying mechanisms, and large language model (LLM) frameworks such as Retrieval-Augmented Generation (RAG), represent pivotal architectures for enabling the digital integration and automation of the Last Planner System (LPS) [28, 29, 30, 31].

5.1 Deep learning for ALPS

Deep learning DL, a subset of machine learning, utilises advanced architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to process and generalise complex data with high accuracy [32]. These models, known for scalability and pattern recognition, are well-suited for enhancing project control and scheduling within frameworks like the Last Planner System as indicated in Figure 5. Also, deep implicit layers, which are an ordinary differential equation technique that is able to represent constant changes and understand multi-level reasoning as needed in an automated last planner system (ALPS), would be crucial for automation. It is noted that complex problems such as one represented in Section 4, can be crunched into a single fixed iteration formula, which can then guide LLMs to make feasibility and multilevel reasoning solutions [33].

5.2 RAG for ALPS

Retrieval-augmented generation (RAG) enhances LLMs by integrating data from external databases, enabling more accurate and context-aware outputs. RAG operates through three interconnected modules: the retriever, retriever fusion, and generator[34]. RAG techniques are particularly effective for automating text summarisation, question answering, information extraction, text classification, etc. In the context of ALPS, RAG can significantly enhance processes by enabling constraint classification, automating notifications and communication categorisation, and summarising extensive text data. This integration offers a powerful tool for streamlining complex workflows and improving project management efficiency.

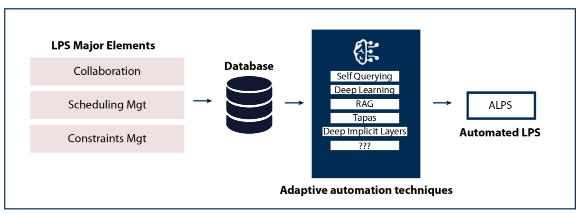

Figure 5. LPS - AI integration

Constraint Title	Notes	CR_ days
drainage and sewage pipes Qrac20 north	KUPR issued drawings which they are working	120
Conduit under Staircore to be coordinated and IFC'd and layed prior to staircore	Infrastructure to arrange	235
Kupr can't start works if not moved	Tank moved Friday 08/03/2024	1
Crane mat design		4

Figure 6. Sample of Constraints; Highlighting keywords, date and NER

5.3 Towards ALPS: Constraint Resolution prediction

The potential for improved augmentation can be significantly enhanced through the integration of predictive strategies in digitised LPS, and the aspects that this integration can impact are enormous. In terms of application, we only looked at ascertaining the resolution duration for constraints, which is an integral aspect of ALPS.

5.3.1 Data

VisiLean, a major cloud-based construction management software, supplied constraint and schedule data for major ongoing projects. Facts and details of the project are withdrawn for privacy reasons. A total of 173 constraints of a large multi-stage developmental project are contained in the dataset, and the data is framed as in Figure 3 and analysed. A sample of the constraint and notes can be seen in Figure 6.

5.3.2 Process

To build a model that predicts constraint resolution duration, we used the constraint dataset. This data was analysed using a cloud-based Python IDE called Collab. First, preliminary data preprocessing was carried out to understand the essential features needed for the model. The unnecessary columns in our dataset were dropped, and constraint resolution duration for all constraints was calculated by determining the duration between the date of constraint resolution and the date of constraint creation. As indicated in Table 2, complexity score, which depicts the difficulty level of a constraint, was determined using keywords, named entity recognition NER, deadline detection, sentiment analysis and structural features, including the length of constraints. After this, both textual and numerical columns were preprocessed. This step regularised the dataset for accurate model training purposes. The final set of features used for training these models includes TDIF vectorisation of combined textual data (Constraint title, description, notes and linked tasks), priority, and Trades. The model architecture is indicated in Figure 4. Afterwards, the created model was evaluated and accessed in the following section. The model is simplified with reduced layers, removed dropouts, and batch normalisation to allow the model to map the complexity of the small datasets fully. Epoch was set to 50, and batch size was reduced to 4 to optimise weight updates and better utilize limited data.

Table 2. Complexity score calculation

Feature	Summary
NER	This recognises named entities via SpaCy
Keywords	keywords are classified based on complexity -critical and mild
Sentiment Analysis	Scoring based on perceived negativity or positivity.
Structure	Words depicting dependencies signifies complexity eg and, but etc.
Length combined	Length of character of the combined textual data
Presence of dates	dependency, date or deadline count
Counts of keywords	Number of keywords, dependencies per constraints

6 Results

The results of the fully connected neural network model are shown in Table 3. The model predicts constraint resolution duration, and the output is explained here. The average duration error or variance for each of the test instances was determined as 23.18 days (MAE), and the standard deviations of each of the error instances in our datasets, indicating effectively the deviation of the actuals from the target variable is 30.12 days (RMSE).

Table 3. Fully connected neural network results

Model	MAE	MSE	RMSE
Neural Network (MLP)	23.18	907.22	30.12

Figure 7 indicates the results plot. These plots, the RMSE and MAE values, indicate that the model overall might not be fully capturing the patterns in the data. Although the target variable is a variable with increased variability and the dataset size is small, the model will keep improving as it is trained on more data. The model can help automate the last planner system as it will assist in pre-empting when constrained jobs can be rescheduled or included in certain lookahead

Table 4. Error comparison using other models

Model	MAE	MSE	RMSE
SVR	21.78	935.93	30.59
LightGBM	23.43	1048.46	32.38
Random Forest	24.06	1108.12	33.29

7 Discussions and Limitations

In further discussion and comparison of the results in Tables 3 and 4, it was discovered that the support vector regression SVR achieves a slightly lower MAE of 21.78 days but exhibits a comparable RMSE of 30.59 days, indicating that SVR has a lower deviation from the ideal when compared with neural network model. On the other hand, the lightGBM and random forest models display higher absolute error values (MAE: 23.43 and 24.06 days), suggesting that they are unable to map the patterns in the data as much as SVR or the Neural network model. This higher error variability is also replicated in the RMSE values, indicating that neural network models perform relatively better. Although SVR is not completely outperformed in terms of MAE by the neural network model, the neural network model maintains a competitive RMSE, signigfiying its ability to generalise despite the small dataset and variability in target

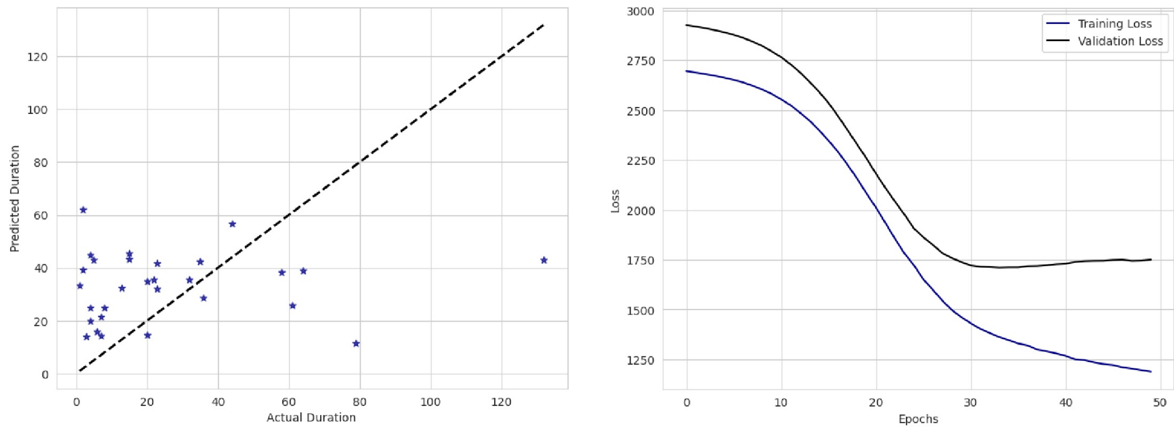

Figure 7. Plot of constraint-resolution-duration actuals and prediction compared with ideal prediction line. Validation loss and loss plot indicates the efficiency at which the model learns.

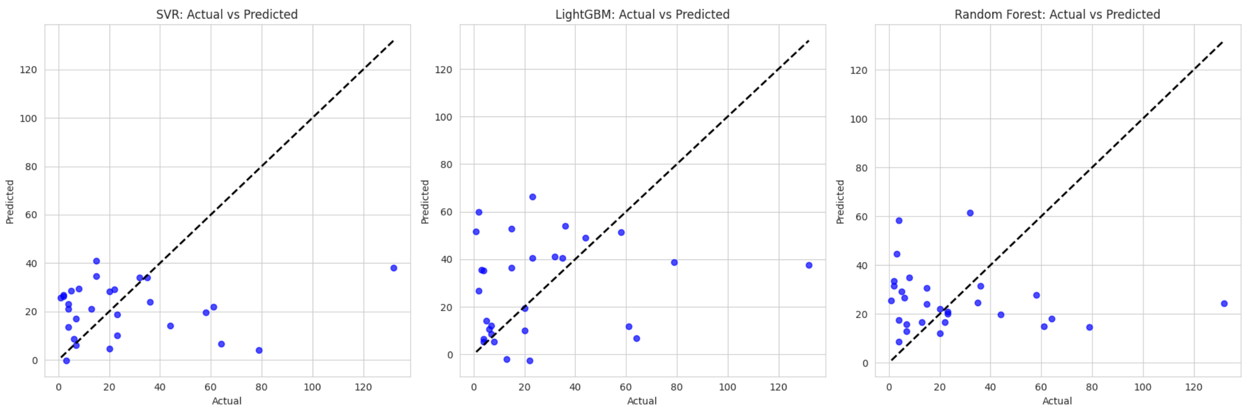

Figure 8. support vector regression SVR, LightGBM, and Randomforest Plot of constraint-resolution-duration actuals and prediction compared with ideal prediction line.

variable. Furthermore, as the dataset expands, the neural network has greater potential for improvement through training and hyper-parameter tuning. The plots in Figures 7a and 8a showcases the prediction of each constraint instance to the ideal with specific improvement in predictions especially when plots of both neural network and SVR are weighed side by side.

While this work highlighted the core aspects of LPS where automation can be applied, the potential for automated constraint management systems, automated collaborative systems, and automated scheduling systems all integrated into one system will increase the efficiency and accuracy of the Last Planner system. To initiate this ambitious research concept, the study conceptualized the system as an optimization problem, which can be reformulated into a fixed iterative equation, such as a gradient descent operation. This equation can subsequently be integrated as a constraint within a large language model (LLM) to facilitate the generation of accurate semantic queries and the determination of feasible solutions for lookahead scheduled tasks. Secondly, an analysis of constraint data of major ongoing projects to determine constraint or conflict resolution time will be handy in quick insight generation and accuracy of Lookahead tasks. However, certain limitations were evident throughout the study.

- The dataset revealed several instances where the creation_date and committed_date occurred after the completion date, presenting a distorted representation of the actual outcomes. This discrepancy primarily arises from the fact that actual data inputs are often recorded not for process improvement or performance evaluation but rather merely to fulfil reporting requirements. Such limitations hinder the model's ability to generalise effectively over the training dataset, as many data inputs exhibited either zero or negative values for constraint resolution time.

- The available dataset was relatively small in size. In cases involving larger datasets, the model would likely demonstrate improved predictive capability regarding constraint resolution durations, thereby minimising associated errors.
- The textual data utilised in the analysis, which includes fields such as Constraint_title, Description, Notes, and Linked_tasks, exhibited a variety of named entities and structural inconsistencies. These variations complicated the analysis of constraint complexity and hindered the appropriate feature engineering necessary for enhancing model accuracy.

7.1 Recommendations and Implications for Practitioners

While the ideas presented in this research are still evolving, they can be viewed as an augmentation of the Last Planner System (LPS) rather than a complete automation of last planners. This is reflected in the title: Towards Automation of the Last Planner System. The first concept focuses on optimizing the scheduling management system to categorize the most suitable list of tasks for specific look-ahead periods while considering existing constraints and communication dynamics. Although the optimization was evaluated mathematically, the proposed LPS-AI framework suggests employing methods such as Retrieval-Augmented Generation (RAGs), TAPAS, Deep Implicit Layers, Deep Learning, and self-querying to automate the entire Last Planner System.

The second concept involves predicting constraint resolution durations. This predictive model assists in determining the period within which specific constraints will be resolved. As a result, it facilitates the scheduling of optimized look-ahead tasks by providing foresight into constraint resolution timelines. Traditionally, in the absence of such models, planners and foremen have had to replan and reschedule tasks at the beginning of each work period due to their inability to anticipate constraint resolution times, thereby losing valuable productive work hours.

Although the target variable exhibits high variability and the dataset is currently limited, the model is expected to improve as it is trained on more data. By preemptively identifying when constrained tasks can be rescheduled or incorporated into specific look-ahead plans, this model contributes to automating the Last Planner System and enhancing workflow efficiency

7.2 Future Work

Overall, the limitations are pointers to what must be done to achieve either an augmented system or a self-sufficient system that can rely on information at hand to generate accurate lookahead schedules for desired periods. The quality of the generated actual data is very important, and a structured list of possible constraints and tasks should be adhered to. This will help analysis and integration easily. While the problem is dynamic, as highlighted in the definition of the optimisation problem, constrained iterations with implicit and explicit functions would be coded into an LLM as constraints for outputs in a RAG structure to improve and augment the existing LPS. Also, improved research on table parsing (TAPAS) to map and understand complex linked tables with several rows would advance the possibility of augmentation of last planner systems.

8 Conclusion

This research highlights the potential for automating the Last Planner System (LPS) by integrating advanced AI techniques to enhance constraint management, scheduling, and collaboration. Rather than aiming for full automation, the proposed framework augments the Last Planner System by optimizing lookahead scheduling and predicting constraint resolution durations. By framing LPS as an optimization problem, this study explored the use of deep learning, Retrieval-Augmented Generation (RAG), Large Language Models (LLMs), and predictive modeling to improve decision-making and efficiency.

One key contribution of this research is the introduction of a predictive model that estimates constraint resolution durations, thereby enabling more proactive scheduling of lookahead tasks. Traditionally, foremen and planners face challenges in anticipating when constraints will be resolved, leading to frequent replanning and inefficiencies. By leveraging AI-driven predictions, this study lays the foundation for reducing uncertainty and minimizing lost productive hours.

Despite its promise, limitations such as dataset inconsistencies, small sample sizes, and challenges in complex table parsing highlight areas for further development. To overcome these challenges, future research should focus on refining the optimization representation into a constrained iterative framework, embedding implicit and explicit functions into LLM structures for better integration within RAG architectures. Additionally, improving structured data collection processes and enhancing table parsing techniques (e.g., TAPAS) will be critical in augmenting LPS capabilities.

By addressing these gaps, this research paves the way for an AI-augmented Last Planner System that enhances construction workflow efficiency, improves predictability, and ultimately transforms project planning in complex, dynamic environments.

9 Acknowledgments

This study was supported by Bournemouth University and VisiLean under a match-funded scholarship program in the UK. I acknowledge the contributions of my supervisors for their constructive comments and feedback throughout the preparation of this research work.

References

[1] Ajay Kumar Agrawal, Yang Zou, Long Chen, Mohammed Adel Abdelmegid, and Vicente González. Moving toward lean construction through automation of planning and control in last planner system: A systematic literature review. *Developments in the Built Environment*, page 100419, 2024. ISSN 2666-1659.

[2] Camilo I Lagos, Rodrigo F Herrera, Alejandro F Mac Cawley, and Luis F Alarcón. Predicting construction schedule performance with last planner system and machine learning. *Automation in Construction*, 167:105716, 2024. ISSN 0926-5805.

[3] Daniel Heigermoser, Borja Garcia de Soto, Ernest Leslie Sidney Abbott, and David Kim Huat Chua. BIM-based Last Planner System tool for improving construction project management. *Automation in Construction*, 104:246–254, 2019. ISSN 0926-5805.

[4] Glenn Ballard. The Last Planner System of Production Control. *University of Birmingham Research Archive*, 1 2000.

[5] Bhargav Dave, S Boddy, and Lauri Koskela. Visilean: Designing a production management system with lean and BIM. *19th Annual Conference of the International Group for Lean Construction 2011, IGLC 2011*, pages 477–487, 1 2011.

[6] Bhargav Dave. *Developing a construction management system based on lean construction and building information modelling*. PhD thesis, University of Salford, 2013. URL https://www.proquest.com/openview/5a51dcdb6d531ebc4a7a35c9b42869d6/1?cbl=2026366&diss=y&pq-origsite=gscholar.

[7] Mughees Aslam, Zhili Gao, and Gary Smith. Development of innovative integrated last planner system (ILPS). *International Journal of Civil Engineering*, 18:701–715, 2020. ISSN 1735-0522.

[8] Thomas H Davenport, Jeanne G Harris, David W De Long, and Alvin L Jacobson. Data to knowledge to results: building an analytic capability. *California management review*, 43(2): 117–138, 2001. ISSN 0008-1256.

[9] Nowshade Kabir and Elias Carayannis. Big data, tacit knowledge and organizational competitiveness. In *proceedings of the 10th international conference on intellectual capital, Knowledge Management and Organisational Learning: ICICKM*, page 220, 2013.

[10] Olli Seppänen, Glenn Ballard, and Sakari Pesonen. The combination of last planner system and location-based management system. *Lean construction journal*, pages 43–54, 2010.

[11] Roshan Poudel, Borja Garcia de Soto, and Eder Martinez. Last Planner System and Scrum: Comparative analysis and suggestions for adjustments. *Frontiers of engineering management*, 7:359–372, 2020. ISSN 2095-7513.

[12] Ankit Bhatla and Fernanda Leite. Integration framework of bim with the last planner systemtm. In *IGLC 2012-20th Conference of the International Group for Lean Construction*, 2012.

[13] M C Garrido, R Mendes Jr, S Scheer, and T F Campestrini. Using BIM for last planner system: case studies in Brazil. In *Computing in Civil Engineering 2015*, pages 604–611. 2015.

[14] Bhargav Dave, Juho-Pekka Hämäläinen, and Lauri Koskela. Exploring the Recurrent Problems in the Last Planner Implementation on Construction Projects. *Proceedings of the Indian Lean Construction Conference (ILCC 2015)*, 1 2015.

[15] Conrad Boton, Yaya Pitti, Daniel Forgues, and Ivanka Iordanova. Investigating the challenges related to combining BIM and Last Planner System on construction sites. *Frontiers of Engineering Management*, 8:172–182, 2021. ISSN 2095-7513.

[16] Rein Terje Thorstensen, Bo Terje Kalsaas, John Skaar, and Sigmund Jensen. Last planner system innovation efforts on requirements for digital management system. In *Annual conference of the international group for lean construction*, volume 21, 2013.

[17] Mani Poshdar, Vicente A González, Gary M Raftery, Francisco Orozco, and Guillermo G Cabrera-Guerrero. A multi-objective probabilistic-based method to determine optimum allocation of time buffer in construction schedules. *Automation in Construction*, 92:46–58, 2018. ISSN 0926-5805. doi:https://doi.org/10.1016/j.autcon.2018.03.025. URL https://www.sciencedirect.com/science/article/pii/S0926580517300432.

[18] Vicente González and Luis Fernando Alarcón. A methodology for integrated buffer design and management in repetitive construction projects. In *Proceedings of the 17th Annual Conference of the International Group for Lean Construction*, pages 393–406, 2009.

[19] David Jason Gerber, Burçin Becerik-Gerber, and Alex Kunz. BUILDING INFORMATION MODELING AND LEAN CONSTRUCTION: TECHNOLOGY, METHODOLOGY AND ADVANCES FROM PRACTICE. In *IGLC -Enabling Lean with IT*, Haifa, Israel, 7 2010. URL https://api.semanticscholar.org/CorpusID:13917181.

[20] Ophir Rozenfeld, Rafael Sacks, and Yehiel Rosenfeld. 'CHASTE': Construction hazard assessment with spatial and temporal exposure. *Construction Management & Economics*, 27:625–638, 7 2009. doi:10.1080/01446190903002771.

[21] Rafael Sacks, M Treckmann, and O Rozenfeld. Visualization of Work Flow to Support Lean Construction. *Journal of Construction Engineering and Management-asce - J CONSTR ENG MANAGE-ASCE*, 135, 12 2009. doi:10.1061/(ASCE)CO.1943-7862.0000102.

[22] Fouad Amer, Yoonhwa Jung, and Mani Golparvar-Fard. Transformer machine learning language model for auto-alignment of long-term and short-term plans in construction. *Automation in Construction*, 132, 9 2021. doi:10.1016/j.autcon.2021.103929.

[23] Luis F Alarcon and Rodrigo Calderon. A Production Planning Support System for Construction Projects. In *11th Annual Conference of the International Group for Lean Construction*, Virginia, USA, 2003. URL http://iglc.net/Papers/Details/222/pdfhttp://iglc.net/Papers/Details/222.

[24] Hyun Jeong Choo, Iris Tommelein, Glenn Ballard, and Todd Zabelle. WorkPlan: Constraint-Based Database for Work Package Scheduling. *Journal of Construction Engineering and Management-asce - J CONSTR ENG MANAGE-ASCE*, 125, 6 1999. doi:10.1061/(ASCE)0733-9364(1999)125:3(151).

[25] Hyun Jeong Choo and Iris D Tommelein. WorkMovePlan: Database for Distributed Planningand Coordination. In *Iglc*, 2000. URL https://api.semanticscholar.org/CorpusID:55674696.

[26] E Sriprasert and Nashwan Dawood. Multi-constraint information management and visualisation for collaborative planning and control in construction. *Electronic Journal of Information Technology in Construction*, 8:341–366, 10 2003.

[27] Tarek Salama, Ahmad Salah, and Osama Moselhi. Integrating critical chain project management with last planner system for linear scheduling of modular construction. *Construction innovation*, 21(4):525–554, 2021. ISSN 1471-4175.

[28] Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. Large language model for table processing: A survey. *arXiv preprint arXiv:2402.05121*, 2024.

[29] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. ISBN 3030585670.

[30] Malte Helmert. The fast downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006. ISSN 1076-9757.

[31] Weiye Xu, Min Wang, Wengang Zhou, and Houqiang Li. P-RAG: Progressive Retrieval Augmented Generation For Planning on Embodied Everyday Task. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6969–6978, 2024.

[32] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1):53, 2021. ISSN 2196-1115. doi:10.1186/s40537-021-00444-8. URL https://doi.org/10.1186/s40537-021-00444-8.

[33] Kolter Zico, Duvenaud David, and Johnson Matt. Deep Implicit Layers - Neural ODEs, Deep Equilibirum Models, and Beyond, 2023. URL https://implicit-layers-tutorial.org/.

[34] Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and Chun Jason Xue. Retrieval-Augmented Generation for Natural Language Processing: A Survey. 7 2024. URL http://arxiv.org/abs/2407.13193.