

Construction Schedule Optimisation

Optimisation of BIM-based, component-level
construction schedule
for building structural and MEP systems
considering parallel working zones

Student Name	Student Number
--------------	----------------

Xinzhi Jiang	5349230
--------------	---------

Construction Schedule Optimisation

Optimisation of BIM-based, component-level
construction schedule
for building structural and MEP systems
considering parallel working zones

by

Xinzhi Jiang

in partial fulfillment of the degree of Master of Science
at the Delft University of Technology,
to be defended publicly at 15:30 on 20 December 2022.

Student number:	5349230	
Project duration:	April - December 2022	
Thesis committee:	Prof. dr. ir. J.W.F. Wamelink,	TU Delft, Chair
	Dr. ir. M. Nogal Macho,	TU Delft, Main Supervisor
	Dr. ir. O. Kammouh,	TU Delft, Secondary Supervisor
	Ir. Y. Shang,	TU Delft, Additional Supervisor
	D. Arts,	Hercuton b.v., Company Supervisor

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

This report is written in partial fulfillment to the MSc degree in Construction Management and Engineering in TU Delft. It also marks the end of my 2.3-year MSc study here.

My days in Delft has been full of learning, wonder and joy. I enjoyed many courses, projects and internship; I am surrounded by a group of like-minded and enthusiastic people, and I am amazed with the opportunities here to connect, develop, and thrive; I had the luck to re-start and be supported throughout a journey of self-discovery, and came to know and accept myself more at the end of the day. Today, I feel confident to say that I have developed myself academically, professionally, and as a person.

There are many people I would like to thank here.

First, I would like to thank the members of the graduation committee. My immediate and deepest gratitude goes to Dr. Maria Nogal Macho – one of the best I have had the privilege to meet. I have been very impressed, as well as influenced, by her devotion, dedication, guidance, and attention to detail and perfection, ever since one of her courses one and a half year ago, all the way to the honours project and now the graduation thesis. You have not only made this project better, but me as a better person.

I would also like to thank Prof. dr. ir. Hans Wamelink, who has given invaluable feedbacks both on the work itself, from a practical and overarching angle, and on the presentation skills as well. And to Dr. Omar Kammouh, and Ir. Yue Shang – I was always amazed by the detailedness and the amount of effort whenever I received their feedback documents, and their comments during the meetings always made the work better. Lots of my thanks go to the last member of the committee – Daan Arts, my company supervisor, for offering the opportunity of this graduation internship, sharing his perspective from the industry and constant feedbacks, and especially for his help with the preparation of the final presentation on how to think from the perspective of the audience.

Till this point in my life I have had the fortune of meeting a group of wonderful people. To my friends – I enjoyed our walks, meals in Aula, voice calls, study sessions at each other's places, or chats which can start literally anywhere inside library; I like our sharings on our own works, interests, personal experiences and the future. I wish you all the best in your future which I am sure is going to be a bright one. There are also lots of people without whose support this work would not have been possible, but it is not possible to thank them here one by one.

This whole journey started in a phone call one night over two years ago, when my parents backed my choice with enthusiasm, love, and support – the same love and support I have been receiving for the past 25 years. They have given me an “internal pot of gold” (Baron-Cohen, 2012), from which I have for many years drawn strength to overcome, recover, care, trust, and love. So, to my parents, and grandma – lots, lots, lots of love.

Xinzhi Jiang

December 2022, in the beautiful city of Delft

Executive Summary

Abstract

Construction schedule optimisation problems have been explored extensively, including activity sequencing rules and work packaging. Yet knowledge is still lacking in the sequencing of mechanical, electrical and plumbing (MEP) components with geometric complexity, and how to handle conflicting precedence between MEP work packages arising from the geometric complexity. Another concept of interest is working zones, which are spaces a building may be divided into to enable parallel working: they have the potential to reduce idle working space and project duration, but its integration with the scheduling of MEP systems and effect on schedule optimisation are under-investigated. This work studies the optimisation of construction schedules for building structural and MEP systems considering working zones. First, a conceptual framework is developed, on: (1) activity sequencing rules, featuring preferences on spatial proximity and component size for MEP components; (2) clustering and cluster-splitting method, to resolve the conflicts among MEP packages; and (3) mathematical formulation of schedule optimisation problems as mixed-integer linear programming (MILP) problems. Next, a software tool consisting of an Excel add-in, a MATLAB executable programme and an Excel macro is developed to implement the framework. Two case studies are carried out. The results of case studies and further analysis demonstrate the large potential of zones in reducing project duration, the effect of the amount of resource available, and strategies for future scheduling practices. Applicability to general construction projects, limitations and future directions are also discussed.

Research problem

Scheduling of construction activities is crucial for a construction project and has direct impact on project duration, cost and quality. For human planners, to produce an element-wise schedule can be tedious and vulnerable to human errors, and it is unknown how close the result is to optimal ones. In contrast to building structural systems, the scheduling of mechanical, electrical and plumbing (MEP) systems is under-investigated despite its large implication on project cost and duration, and the research is lacking on its sequence identification, grouping into packages of manageable size (or work packages), and how to deal with conflicts in the precedence between packages. Building information modelling (BIM) and the abundant information that can be extracted from a BIM model may serve as a starting point for sequence identification and automation of schedule generation.

Another concept of interest in construction scheduling is “zones”, which are spaces a building may be divided into, to enable teams working simultaneously in different spaces and consequently reduce project duration. The effect of the design of working zones, and how to schedule individual building components considering working zones, is to be investigated in this study.

The main research question is then proposed to be:

How to automatically generate and optimise a BIM-based, component-level schedule for structural and MEP systems, considering parallel working zones?

Methodology

To answer the research question, a conceptual framework is developed, including:

1. Activity sequencing rules for building structural and MEP systems. Two levels of sequencing preferences are proposed for MEP systems, considering preferences on special spatial relations and component size relations (length and cross-sectional area).
2. Clustering of individual components, and cluster-splitting using matrices to deal with the conflicts in precedence relations.
3. Mathematical formulation of the multi-objective optimisation (MOO) problem as a mixed-integer linear problem (MILP), with total direct cost and duration as objectives.

Software implementation

A software tool is developed to implement the conceptual framework. It takes BIM models as the main inputs and generates (near-)optimised schedule(s). The tool contains three parts: (1) an Excel add-in, to extract building component information from BIM models in .ifc; (2) a MATLAB standalone executable programme, to set up and solve the optimisation problem; and (3) an Excel macro, to write the output schedule into a Microsoft Project .mpp file.

The conceptual framework and software implementation are then verified and validated using a simple case and two case studies.

Results and recommendations

Observations are made from the case studies and further analysed. It is then discussed:

1. Effect of rectangular working zones and resource availability: the total duration is constrained by both the number of zones and maximum resource availability, and they show different magnitudes of effect on project cost and duration.
2. Selection of activity execution modes: for those activities which have more than one ways of execution associated with different productivity and cost, or, “execution modes”, some require a lower cost increase to achieve relatively high time saving, and are suggested to be selected first when compressing the schedule.
3. The effect of the unit of activity durations, computational capacity of the tool, and the extension to general construction project.

Based on the observations, recommendations are made to the industry on: (1) how to select activity execution modes, or increase the resource availability, to shorten project duration while maintaining optimality of the solution; (2) how to adjust problem settings to apply the tool in a general project. Directions for future research are also suggested, on more sophisticated activity precedence identification, integration with logistic and site planning, and more flexibility with the design of zones.

This work contributes to scientific research with: (1) a set of sequencing principles for MEP components, including two-level preference checking to account for spatial proximity and component sizing; (2) a method for work packaging of MEP components, which addresses the conflict in precedence relations at cluster- and component-level. Contributions to the industry include the open-source software tool and the insights for practice. The limitations of this work yielded from its key assumptions are also discussed.

Contents

Preface	i
Summary	ii
1 Introduction	1
1.1 Challenges in industrial practice	1
1.2 Challenges in research	2
1.3 Overview of this work	2
1.3.1 Research questions	2
1.3.2 Research methods	2
1.3.3 Outputs	2
1.3.4 Structure of this report	3
2 Related Works	4
2.1 Construction schedule optimisation problem	4
2.1.1 Overview	4
2.1.2 Activity sequencing rules	4
2.1.3 Sequencing of MEP components	5
2.2 Work packaging and lean construction	5
2.2.1 Work packaging	5
2.2.2 Lean construction and zones	6
2.3 Research gaps	6
3 Methodology and Conceptual Framework	8
3.1 Terminology and key assumptions	8
3.2 Activity sequencing rules	10
3.2.1 Sequencing of structural clusters	11
3.2.2 Sequencing of MEP components	11
3.3 Clustering and cluster-splitting	15
3.4 Mathematical formulation of the optimisation problem	20
4 Software Implementation	24
4.1 Functional requirements	24
4.2 The software tool	24
4.2.1 The Excel add-in: extraction of component information from BIM	25
4.2.2 The MATLAB programme: setting up the optimisation problem	26
4.2.3 The Excel macro: preparing the schedule for 4D visualisation	28
4.3 Verification and validation	28
5 Case Studies	31
5.1 Case 1: HoutWerk, Utrecht	31
5.1.1 Introduction of the case and problem setting	31
5.1.2 Results	31
5.1.3 Discussion	32
5.2 Case 2: a simplified distribution centre	35
5.2.1 Introduction of the case and problem setting	35
5.2.2 Results	38
5.2.3 Discussion	38
6 Discussion	40
6.1 Discussion on the results of the case studies	40
6.1.1 Effect of working zones and resource availability	40

6.1.2	Selection of activity modes	42
6.1.3	Computational capabilities	43
6.2	Sensitivity analysis	45
6.2.1	Activity duration	45
6.3	Extension to general projects	46
6.3.1	Schedule in portions and combine	46
7	Conclusion and recommendation	49
7.1	Answers to the research questions	49
7.2	Contribution of this work	50
7.3	Limitations	50
7.4	Recommendations to industrial practice	51
7.4.1	Recommendations on schedule optimisation	51
7.4.2	Use of the software tool	52
7.4.3	Design of working zones	52
7.5	Recommendations for future research	52
	References	54
A	Matrix transformation in Step 5 of cluster splitting	56
B	Generation of initial solution in the MILP problem	60
C	Case study 1: data and scheduling result	62
D	The software tool: list of customisable settings	64
E	The software tool: instructions on how to customise sequencing rules	66
E.1	Identification of floor level	66
E.2	How to customise sequencing rules	68

1

Introduction

In current practices, a construction schedule is mostly created by hand based on experience, and an element-wise schedule is labour-intensive and prone to error. Many research works have been done in this field, and yet the research is lacking in some aspects. Those will be the challenges this work aims to tackle.

This chapter gives an overview of the challenges in current industrial practices and research, which then leads to the research set-ups of this work – research questions, methodology, and expected outputs. Finally, an overview of the structure of the report is given.

1.1. Challenges in industrial practice

Scheduling of construction activities is crucial for a construction project and has direct impact on project duration, cost and quality. For human planners, to produce an element-wise schedule can be tedious and vulnerable to human errors, and it is unknown how close the result is to the optimal ones. The current BIM-based schedule management tools mostly focus on simulation and visualization functions, and require time-consuming manual preparation (Moon et al., 2015; Sigalov and König, 2017; Z. Wang and Azar, 2019); most widely used commercial software applications, such as Navisworks by Autodesk or Synchro by Bentley, do not support schedule generation or optimisation, except for a recent advancement in Bexel Manager by Bexel Consulting, which enables automated construction schedule generation based on pre-set construction methodology. Overall, automatic construction schedule generation or optimisation seems still under-developed in the industry.

Mechanical, electrical and plumbing (MEP) systems can contribute to 40-60% of the total construction cost of commercial buildings (Khanzode, 2010); but its scheduling is trickier than that of the structural system, due to the complex geometrical relations and interrelations with other activities (Andersson, n.d.). Currently, many conflicts in MEP schedules are only detected during installation, resulting in reworks, increased cost, and delays (Singh et al., 2021).

In practice, to reduce idle working space, some planners may divide a floor plan into a few working spaces, or “zones”: two teams can work simultaneously in two different zones without spatial collision, so that the project duration can be reduced. The scheduling of individual building components considering parallel working zones, and the optimal number or design of zones for optimal schedule await investigation.

1.2. Challenges in research

Numerous research works have been done in the field of construction schedule optimisation. Sequencing rules have been developed for building structural systems, based on criteria such as physical relationships, disciplines, or even structural analytical model (Echeverry et al., 1991; H. Liu et al., 2015; Y. Wang et al., 2018); works on the scheduling of non-structural systems are much more limited. Some conceptual rules and sequencing techniques were discussed in literature, but a systematic investigation is still lacking on: (1) the sequencing of MEP components, considering both complex spatial relations and practical preferences; and (2) the work packaging of MEP components and how to deal with the conflicts among packages.

1.3. Overview of this work

1.3.1. Research questions

Main question:

How to automatically generate and optimise a BIM-based, component-level schedule for building structural and MEP systems, considering parallel working zones?

Sub-questions:

1. **How to sequence structural and MEP components in one integrated schedule?**
2. **How to use the “zoning” concept in the schedule optimisation problem?**
 - 2-1: How does it affect the time-cost trade-off in a construction project?
 - 2-2: Can it help produce more optimal schedules?

1.3.2. Research methods

The main activities in this study and corresponding methods are listed as following:

- Problem analysis: the main activities are: (1) literature study in construction scheduling, with focus on the trade-off problem, scheduling of MEP systems, and how parallel crews are considered in literature; (2) analysis of past construction projects, and conceptual development of the optimisation model.
- Data collection and processing: the data needed and its sources includes: (1) building information modelling (BIM) models, schedule and cost information; and (2) practical knowledge from project managers and planners.
- Development of the conceptual framework.
- Implementation of the conceptual framework: this refers to the development of a software tool which realises the conceptual framework.
- Verification and validation: two case studies will be used to test if the developed tool and model meet expectation.

1.3.3. Outputs

- An approach to conduct integrated scheduling of structural and MEP systems;
- A software tool which implements the conceptual framework;
- Insights into how the design of working zones affect the time and cost performance of a construction project, and recommendations to the industry and future research.

1.3.4. Structure of this report

This report is organised as follows: Chapter 2 summarises the existing works in literature in the field of construction schedule optimisation and work packaging, and presents the research and development gaps targeted in this work; Chapter 3 presents three main components of the framework: activity sequencing rules for structural and MEP systems, a method for component clustering and cluster splitting, and mathematical formulation of the optimisation problem. To implement and test that conceptual framework, a software tool is developed, which is introduced in Chapter 4. The conceptual framework and the tool is then verified and validated using case studies in Chapter 5. Finally, the results, observations and limitation of the study are discussed in Chapter 6 and 7, followed by recommendations to the industry and future research.

2

Related Works

This study involves knowledge from several domains, including construction activity sequencing, schedule optimisation, work packaging, and lean construction. The past works on activity sequencing and scheduling lay the theoretical foundation of this work, and some concepts covered in work packaging are used in this work with adaptations; meanwhile, the lack of investigation into the sequencing and scheduling of MEP tasks with zones is also identified in this chapter.

2.1. Construction schedule optimisation problem

2.1.1. Overview

Over the past decades, construction schedule generation or optimisation problem has been investigated rather extensively from various angles. The topics covered include:

1. Sequencing of construction activities: rules have been developed based on physical, supporting, and topology relations
2. Schedule optimisation, using time as the sole objective, or, the trade-off between multi-objectives (MOO) (e.g. time, cost, quality, etc.), e.g. works by Menesi et al. (2015), Elbeltagi et al. (2016)
3. Scheduling with resource constraints and resource levelling, e.g. works by Tang et al. (2018)

BIM is used in many research works over the past decade, as a potential source of abundant data and centre of collaboration: Kim et al. (2013) proposed a framework for automatic data extraction from BIM (ifcXML) file and generation of a preliminary schedule; Moon et al. (2015) optimised project schedules by using genetic algorithm (GA) and self-developed 4D engine, using activity overlap as the sole objective; H. Liu et al. (2015) developed a workflow for component-level schedule generation by using BIM, user-specified work breakdown structure and Autodesk Revit add-ins; more recently, Nguyen et al. (2022) also studied the scheduling problem for major project tasks as a MOO problem, using quantity take-offs from BIM instead of the building components as model inputs. In these studies, the data from BIM lays a good basis for schedule generation, especially for retrieving information at component level or quantity take-offs.

2.1.2. Activity sequencing rules

The earliest works on construction activity sequencing date back to the last century, when sequencing rules of construction activities were proposed based on physical relationships, construction trades, space on site and code regulations such as safety (Echeverry et al., 1991). Similar physical relationships based on or modified from them, such as “supported by”, “covered by”, “embedded in”, have

been used in numerous works thereafter (Kim et al., 2013). In addition to physical relationships, construction sequence can also be derived from structural analytical model (and consequently supporting relationships), or other parameters such as component weight, occupied space and inter-component interference (H. Liu et al., 2015; Y. Wang et al., 2018). However, all those sequencing rules are for building structural components, which has so far been the primary focus of works in construction sequencing.

2.1.3. Sequencing of MEP components

Mechanical, electrical and plumbing (MEP) systems can contribute to 40-60% of the total construction cost of commercial buildings (Khanzode, 2010); the percentage of time taken up by MEP installation and commissioning can be even larger when the building structure components are prefabricated. However, unlike the structure system, not much work has been done in MEP-related installation scheduling problem. The works by Korman et al. (2003) is one of the first works on conceptual sequencing knowledge, mentioning the effect of installability, relative size, length on scheduling; Andersson (n.d.) touched upon the difficulty to forecast the MEP installation workload, due to close interrelation with other building activities and consequently constant adjustment. Two recent works by Singh et al. (2021) and Isaac and Shimanovich (2021) proposed approaches to automate pipe system installation scheduling or MEP scheduling, both using topology analysis of component pairs, namely, horizontal and vertical proximity, by calculating the distance between the faces of components.

But aside from the identification of vertical and horizontal proximities, systematic discussion on the spatial relationship between components is limited, nor is the effect of their surroundings on the spatial constraint discussed. In addition to the vertical and horizontal proximities between each pair of components, the spatial relationship with a third component which imposes extra spatial constraints, namely a ceiling, wall or column, should also be taken into account when determining the installation sequence of MEP components.

2.2. Work packaging and lean construction

2.2.1. Work packaging

A work package is “the lowest level of the work breakdown structure for which cost and duration can be estimated and managed” (PMI, 2004). It usually breaks down construction processes into activities of manageable sizes, executed by the same (sub)contractor, with minimal overlap or interruptions among packages (Gardner, 2006; Isaac et al., 2017). Work packaging can facilitate contracting and subcontracting, and is found beneficial in reducing interfaces between (sub)contractors, and controlling project schedule and cost performances at task level (Li et al., 2022).

When creating work packages, criteria frequently used by contractors are element types, work section (e.g. concrete, masonry), location (e.g. floor level) and construction aids (e.g. tools, scaffolding, formwork), as long as interdependent activities are assigned to the same package as much as possible (Ibrahim et al., 2009; Isaac et al., 2017). The work packages can be defined manually based on work breakdown structures (WBS), where the tasks contributing to the final deliverables are broken down by discipline (Gardner, 2006); they can also be created in a semi-automatic or automatic manner (Ibrahim et al., 2009; Li et al., 2022). Many of the criteria can be easily captured from project BIM model, which makes BIM a good source of information when packaging.

There can be inter-dependency or interruptions among work packages, in various dimensions such as shared resources, organisational responsibility, or starting conditions (Raz and Globerson, 1998). To deal with the interruptions, a widely adopted method is Design Structure Matrix (DSM) (Bashir et al., 2022). A DSM is a square matrix representing the dependencies and interdependencies between project activities, or the precedence relationship between a series of tasks in a complex system (Yassine and Braha, 2003; Bashir et al., 2022). A few recent works on work packaging adopt hierarchical clustering algorithm, with or without DSM: a bottom-up clustering algorithm starts with assigning each

data points into one cluster, before iteratively grouping the clusters which are most similar to each other until the desired number of clusters is reached; while a top-down one starts with one single cluster containing all data points, before iteratively splitting it into clusters which have the largest smallest similarity in between (Al-Jabery et al., 2019; Pezoulas et al., 2020). Bottom-up clustering is used by Li et al. (2022) with adaptation; also by Isaac et al. (2017), who first assigned the components into several zones according to their elevations relative to the floor levels, then used DSM to sequence the components within the same storey. The approach is found to be efficient and effective in minimising the interruptions within activities.

2.2.2. Lean construction and zones

In practice, one type of construction activity in one floor, or even in the whole building, is sometimes scheduled as one task: for example, the task “installation of MEP system in the first floor” may start only after “installation of structure system in the first floor” finishes. While possibly advantageous in terms of resources coordination and safety, when the floor area is large, treating one floor as a unit may lead to idle working space where some works could have been executed in parallel.

To enable parallel working, some planners may divide a floor plan into a few working spaces, or “zones”: two teams can work simultaneously in two different zones without spatial collision. In case of multi-disciplinary teams, by bringing one discipline to site earlier, after the work of their predecessors have been cleared in one portion of the floor, idleness in working space can be reduced; by leveling the amount of resources and creating a steady rate of progress, idleness in resources (e.g. working teams) can be reduced. This resonates with the lean construction techniques for waste reduction and efficient performance (Sacks et al., 2009; Binninger et al., 2016). Some works have been done where zones are designed by leveling the “work density”, or the time needed by each discipline per zone, by iterative trial-and-errors or formulated as an optimisation problem, with manual identification of work densities (Tommelein, 2017; Jabbari et al., 2020).

Another potential advantage of zoning is the reduction in the computational time of a schedule optimisation problem, as the elements are grouped and the total number of activities is reduced, which can otherwise easily exceed a few hundreds in a medium scale project (Menesi et al., 2015).

2.3. Research gaps

So far, no schedule MOO study is found which satisfactorily considers the following aspects at the same time:

1. Integration of building structural and MEP systems in one schedule: most research works focus on only scheduling the building structure system, and very few considers MEP system, despite that the latter can be even more time consuming than the structure system.
2. Precedence relations between MEP components, considering (1) spatial conflicts, both within MEP systems and with its adjacent structural components; and (2) component size differences.
3. When the packaging of construction activities takes into account the physical locations, such as which “zone” a building component falls into, topology relations among components can create extra interfaces or interruptions among work packages. How the topology relations affect the work packaging and how to handle the interfaces resulted is under-investigated, especially for MEP systems.

Aside from the main research gaps described above, the following gaps will also be considered in this project:

1. The target level of activities: many studies, including some BIM-based ones, target major project tasks only and require the planner to specify them. This means extra manual work which is to be repeated every time when the building model is updated. It also means that the schedule produced is not connected to building components or BIM, which limits the real-life applicability.

2. Interoperability: the BIM-based schedules in the works above are derived from different model formats, and are mostly output in a separate file format independent of the BIM model. Such extra file format and software usage is not helping with the interoperability issue in BIM-based collaboration. Instead, it would be more preferable to use one well-recognised, open and shareable file exchange format, such as Industry Foundation Classes (IFC), as both the input and output format or what an output can be easily connected to.

3

Methodology and Conceptual Framework

The research questions ask how to sequence activities and the effect of zones in scheduling. To answer the questions, a conceptual framework is developed, attempting to find proper sequencing principles and to formulate the schedule optimisation problem. This chapter first defines the main concepts and terms used (Ch3.1). Three main components of the framework are then presented:

1. Activity sequencing rules for structural and non-structural systems, and, in particular, two-level sequencing preferences for MEP components with special spatial relations (SSR) and size relations (CSR) (Ch3.2);
2. A method for component clustering and cluster splitting, to handle the conflicts in precedence resulted from the sequencing preferences for MEP components (Ch3.3);
3. Mathematical formulation of the multi-objective optimisation (MOO) problem as a mixed-integer linear problem (MILP) (Ch3.4).

3.1. Terminology and key assumptions

The key concepts to be used in this project are defined in this section.

Working zones

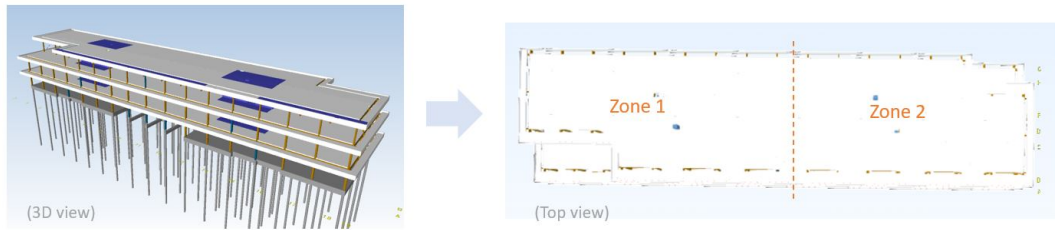
As mentioned earlier, working zones, or, zones, refer to working spaces a building may be divided into during construction, to enable simultaneous working in different spaces. Teams can work in parallel in two different zones without spatial collision between their works, so that the idleness of resources and space and consequently the total project duration is reduced.

In practice, zones can be of difference dimensions and irregular shape, to be adjusted based on the experience and preferences of the planner. But in this study, they are assumed to be mostly identical rectangular area (with possible exceptions of those near the boundary of the building), created by overlaying rectangular grids over the floor plan, with one corner of the building as pivot point.

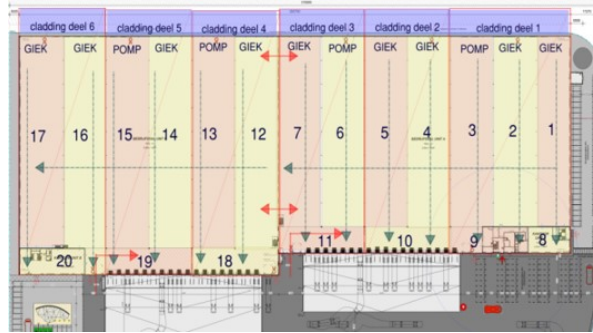
Atom activity

An atom activity refers to the placement or installation work of one building component, e.g. “mount the column of GUID 00001”. Specifically for this project, one atom activity associates with one and only one building component, and vice versa. Sub-activities, or rework on the same component, are not considered in this project due to its complexity.

Activity cluster



(a) Dividing a building into two zones



(b) A real-life planning with zones, where a large one-storey building is divided into 20 zones to be constructed from the right to the left

Figure 3.1: Examples of working zones

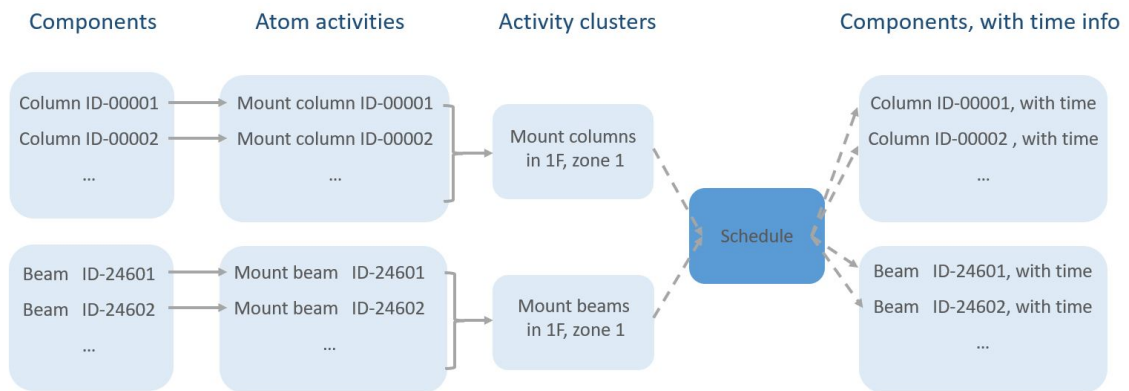


Figure 3.2: Relationship between individual building components, atom activities, and activity clusters

An activity cluster is a group containing multiple atom activities, which usually share the same component class, resource type and working space: for example, the atom activity “mount the column of GUID 00001” may fall into the cluster “mount all columns in first floor, zone 1”. Such cluster has the duration and cost equal to the sum of the duration and cost of all its atom activities, assuming all atom activities within the same cluster are carried out sequentially by the same team. Intra-cluster sequence — in which sequence the atom activities *within* one cluster should be executed — is not discussed in this work.

In this project, though the output schedule will list only activity clusters, it is produced based on component-level information, and each task in it can be traced back to its components. The cluster-level is used when producing the schedule instead of the atom- or component-level, for the following reasons: (1) it is assumed that an atom activity cannot start before all atom activities in the predecessor cluster are finished, e.g. a pipe cannot be installed in G floor, zone 1 before all the slabs there are placed, and therefore it does not affect the final result whether the slabs in G floor, zone 1 are considered individually or as a cluster; (2) by clustering the atom activities, the number of variables in the optimisation problem is reduced drastically, so that higher computational efficiency can be achieved.

In this report, the terms “activity clusters” and “work packages” (described in Ch2.2.1) are inter-changeable:

the former captures the process of clustering and cluster-splitting in later sections while the latter is used more in literature. From this section on, all “activities” refer to activity clusters unless otherwise specified.

Dummy finish activity

To facilitate the optimisation problem formulation, one project finish activity are added to the activity list with activity duration of zero. It will be the successor of all other activities, and the project finish day equals to the finish date of the dummy finish activity.

Activity execution mode

To execute the same activity using one types of resource, there may be more than one possible mode of execution which leads to different productivity and cost. For example, for the activity “mount all steel columns in Zone 1, 1F”, one resource may consist of “one crane + one operator + four worker”, or, “one crane + one operator + six workers”; the former may have less unit cost per hour of team utilisation, while the latter may complete the activity in less time (due to less idle time within the team). Those modes available to choose from can be recorded in the database, for each activity and resource.

	Duration	Cost
	d	c
	$0.8 * d$	$1.3 * c$

Figure 3.3: Example of two execution modes, with which the same activity can be carried out at different cost and duration

Main assumptions

Assumptions are made due to the complexity of the topic, or for a more focused and in-depth research within the time frame of the project. Some important assumptions of this project are listed as following.

1. Activity modelling level: one atom activity correspond to one and only one building components in the building model, and no sub-activities are considered.
2. Though planning in practice includes and is not limited to long-term plans for project milestones, and plans in shorter term such as detailed weekly work plans, in this project the scope is limited to the scheduling of building components extracted from building model.
3. Only duration and cost directly associated with carrying out an atom activity is considered.

3.2. Activity sequencing rules

As explained in the last section, activity clusters are used in the schedule optimisation rather than the atom activities. The execution sequence of these clusters are different for structural and non-structural systems: for most structural components, the sequencing may be strictly dependent on their classes and locations, e.g. a beam cannot be placed before the columns underneath are placed; for non-structural systems, however, the classes may be less of a concern, whereas the spatial proximity or the size of the components adds to the complexity.

In this section, sequencing rules are proposed and scenarios which have implications on the activity precedence are listed, for structural and MEP systems respectively.

3.2.1. Sequencing of structural clusters

The sequencing of activity clusters in structural systems is determined by:

1. component class, further categorised into essential and non-essential ones to which different rules apply;
2. zone number;
3. floor level.

Common component classes in the structural systems include foundation pile, column, beam, stairs, slab, wall, façade: among them, column, beam, slab are considered essential structural classes, and it is due to safety and feasibility reasons that they should start only after all the essential structure classes in the floor and zone directly below are finished; for non-essential ones, it is not compulsory to finish them before the essential classes in the floor above are installed.

All possible scenarios are listed in the form of event tree in Figure 3.4. The key four scenarios which have implications on activity sequencing are extracted below:

- Same zone, same floor, different classes: follow normal construction sequence (e.g. clusters of column -> beam -> slab -> facade);
- Same zone, different floors, both in essential structure classes: lower floor first;
- Same zone, different floors, both in non-essential structure classes: lower floor first.
- (Optional) Different zone, same floor, same class: smaller zone number first

3.2.2. Sequencing of MEP components

The sequencing of activity clusters in MEP systems basically follows the same criteria as those of structural system. However, what is unique here is the geometry-related preferences between individual components: the larger a component is, and the closer and more spatially “awkward” it is to install a component, the more preferable it is to install it before other components in the neighbourhood. That means, the sequence of MEP clusters is affected by:

1. component class;
2. zone number;
3. floor level;
4. geometry-related preferences, between pairs of individual components.

The presence of the fourth criterion also means that, while precedence relation between two MEP clusters can be found based on criterion 1 to 3, some individual components within those clusters may warrant a different precedence relation. Therefore, it should be investigated: (1) what are the precedence relations at component level and how to identify them, and (2) how to address the conflict between precedence relations at cluster level and at component level. The former will be discussed here, and the latter in Ch3.3.

The main geometry-related preferences are based on component spatial relation and size. They can however be conflicting: a thin water pipe may be located within a rather confined space between the ceiling and a large air duct; the air duct would be chosen as the predecessor based on component size, while the water pipe would be chosen based on spatial relation.

To address such conflicts, it is proposed to separate “feasibility” and “convenience”: the former has a higher priority, while the latter is taken into account only when it does not produce a conflicting precedence relation to the former. The preferences are categorised into two levels with different priorities.

Figure 3.5 summarises how two-level preferences are checked for each pair of MEP components. These preferences are further elaborated below.

Level-1 preferences: special spatial relations (SSR)

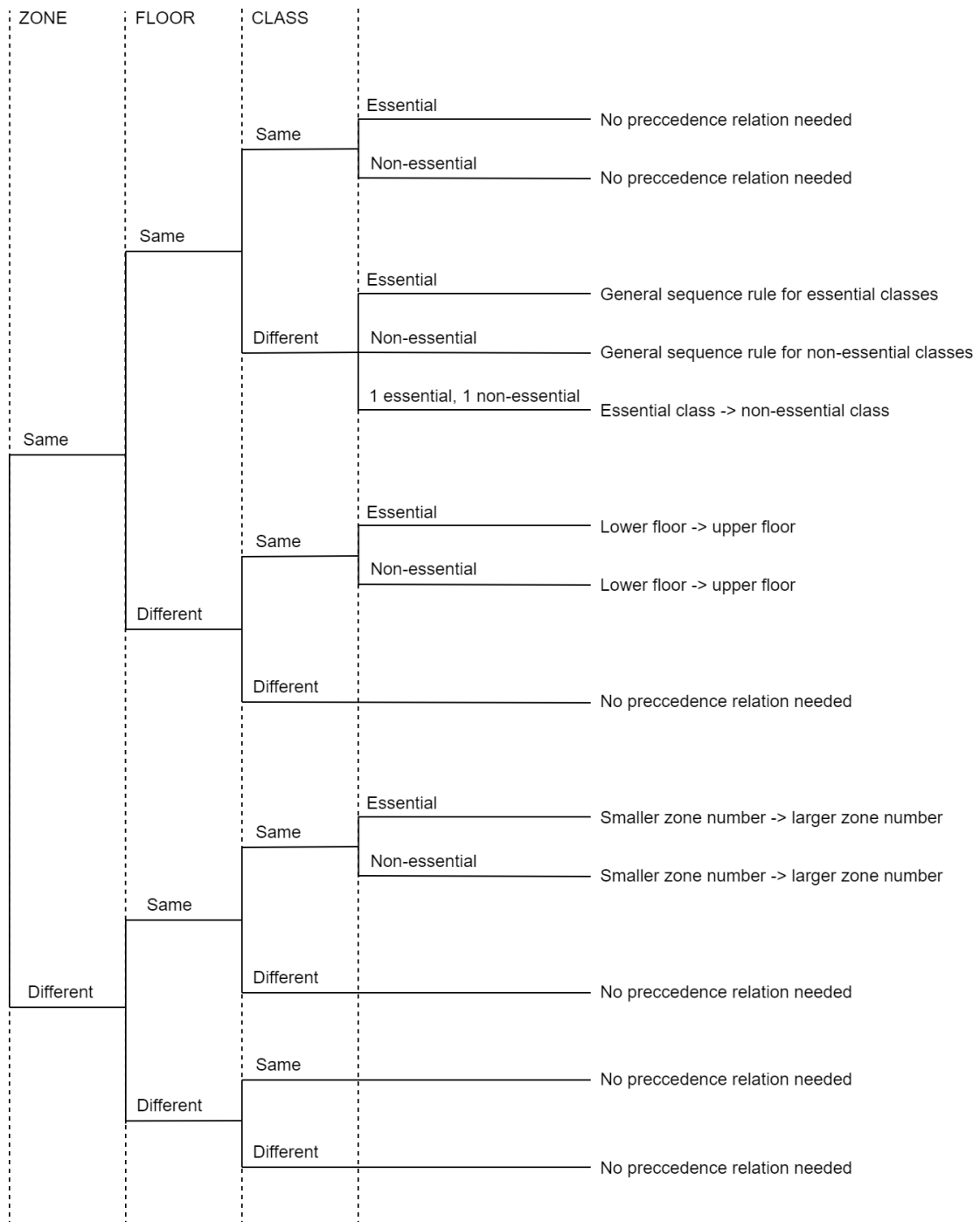


Figure 3.4: Sequencing rules in different scenarios, for the building structural system

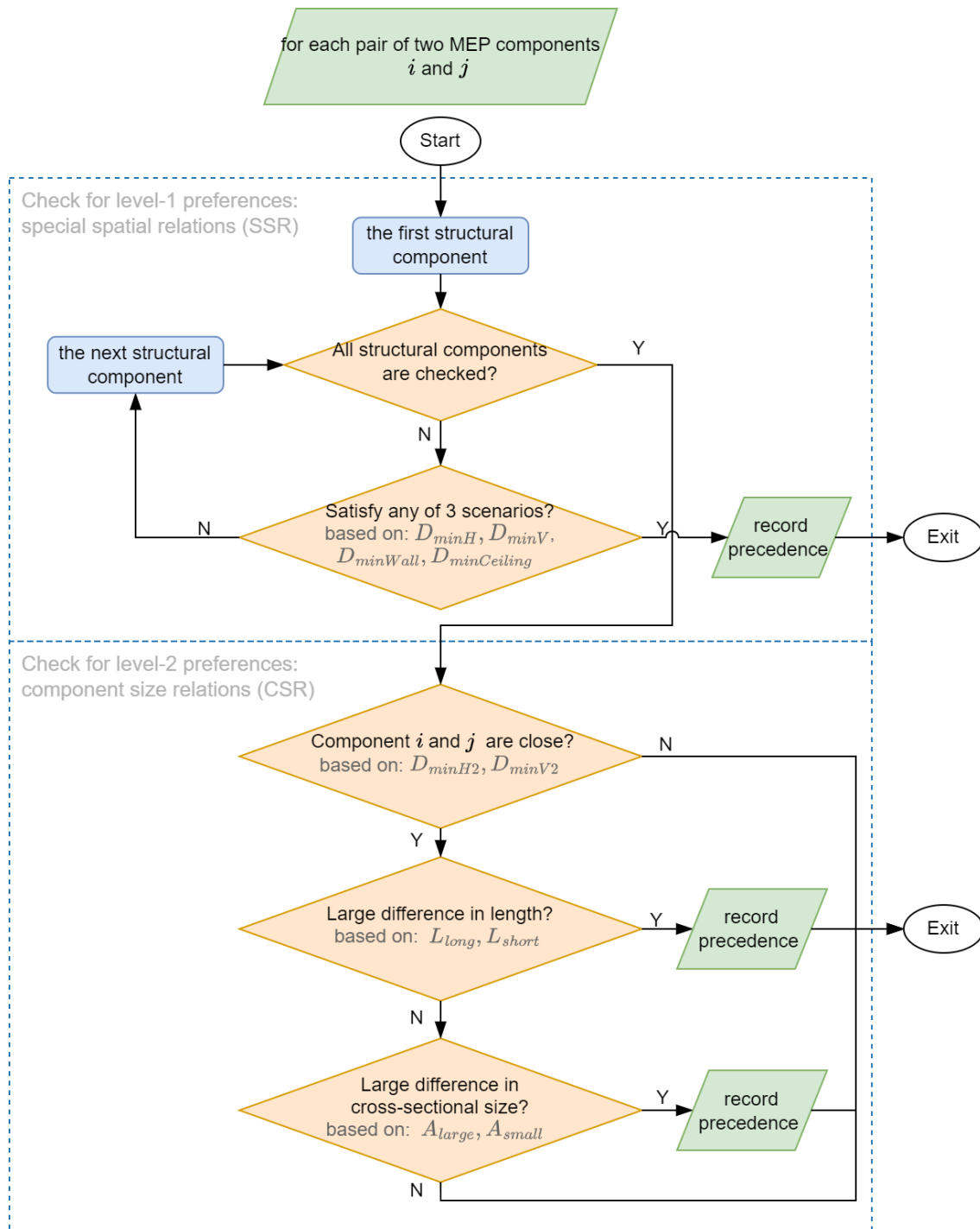


Figure 3.5: 2-Level preference checking for each pair of MEP components

Level-1 preferences takes priority and should be satisfied first, for its direct impact on constructability. Physical proximity between components, referred to as special spatial relations (SSR) hereafter, is considered to be Level-1.

For a pair of MEP components to satisfy SSR:

- these two MEP components should be very close to each other, **and**
- both of them should be very close to one common structural component.

Three such scenarios are presented in Figure 3.6 and Table 3.1. As an example, in scenario 1, (1) two MEP components overlap in y and z directions (represented by $D_y < 0$ and $D_z < 0$ in Table 3.1); (2) their distance in x direction is smaller than D_{minH} , which the minimum horizontal distance between two MEP components for them to be considered “horizontally close” (e.g. 300 mm); (3) both of them overlap with a wall nearby in y and z directions and their distances to the wall are both smaller than $D_{minWall}$, which is the minimum distance between a MEP component and a wall for them to be considered “horizontally close” (e.g. 500 mm). D_{minH} , D_{minV} , $D_{minWall}$ and $D_{minCeiling}$ in Table 3.1 are all user-specified values.

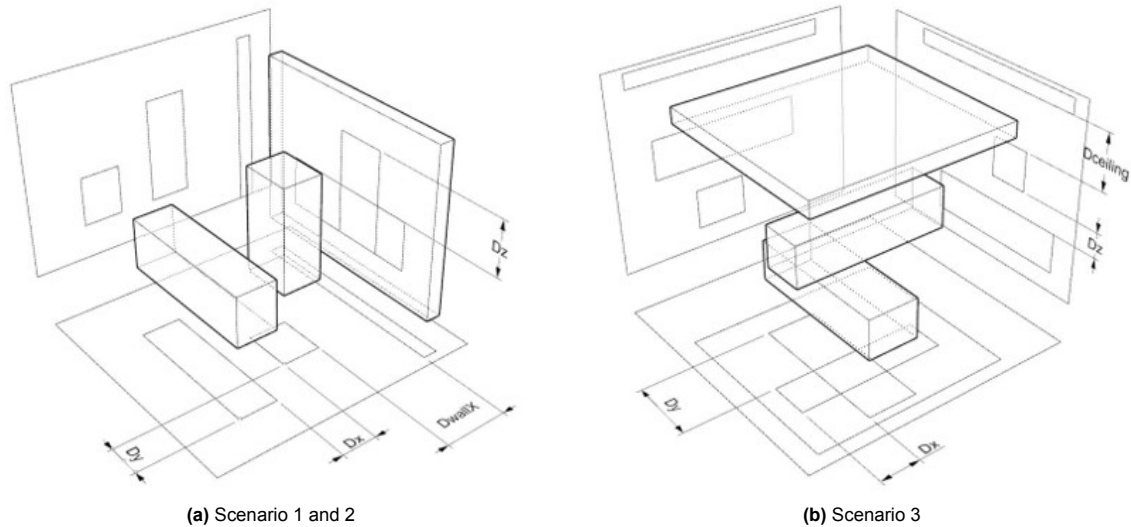


Figure 3.6: Three scenarios of MEP SSR relations, considering adjacent structural components. (Projections of the MEP components are drawn to show the spatial relations in 3D.)

	Scenario 1	Scenario 2	Scenario 3
D_x	$< D_{minH}$	< 0	< 0
D_y	< 0	$< D_{minH}$	< 0
D_z	< 0	< 0	$< D_{minV}$
D_{wallX}	$< D_{minWall}$	< 0	< 0
D_{wallY}	< 0	$< D_{minWall}$	< 0
$D_{ceiling}$	< 0	< 0	$< D_{minCeiling}$

Table 3.1: Three scenarios of MEP SSR relations, where D_{minH} , D_{minV} are the minimum horizontal and vertical distance between two MEP components; $D_{minWall}$, and $D_{minCeiling}$ are the minimum horizontal distance between a MEP component and a wall, and the vertical distance between a MEP component and a ceiling.

Level-2 preferences: component size relations (CSR)

Only after all component pairs satisfying level-1 preferences are identified, will level-2 preferences be considered, so that the preferences in different levels would not yield conflicting precedence relations. Component size relations (CSR) fall into level-2 category.

For a pair of MEP components to satisfy CSR:

- these two MEP components should be very close to each other, **and**
- one of the conditions below should be satisfied:

- large difference in length: one component is very long (i.e. $> L_{long}$), while the other is very short (i.e. $< L_{short}$), **or**
- large difference in cross-sectional area: one component has a very large cross-sectional area (i.e. $> A_{large}$), while the other has a very small one (i.e. $< A_{small}$).

3.3. Clustering and cluster-splitting

Atom activities should be “clustered”, or, “packed” into activity clusters, before the sequencing rules described above can apply: that is basically to assign individual components into several groups, based on their floor level and zone number.

For structural systems, the procedure is quite straightforward: first, for each set of zone dimensions, the boundary of zones can be found easily. Then the structural elements are assigned to zones by comparing the coordinates of the corners of their bounding box with the boundary of zones (Figure 3.7):

- If all corners falls into zone n -> component assigned to zone n ;
- If falls into more than one zone -> component assigned to the zone to be completed last. This to ensure all its predecessors are finished, e.g. a horizontal component is not placed before the component directly underneath is placed.

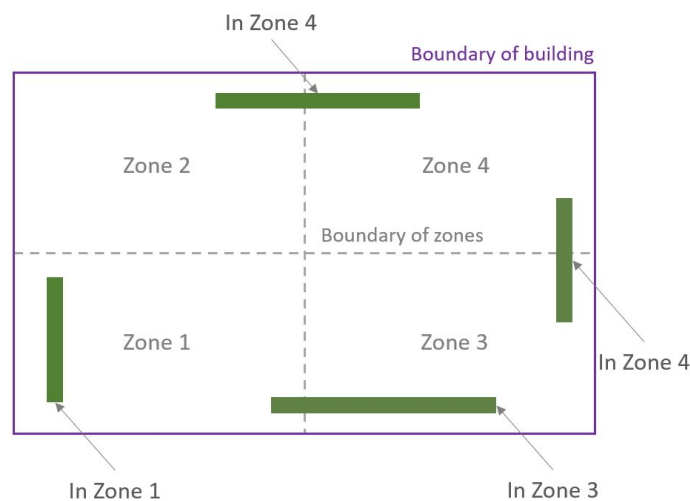


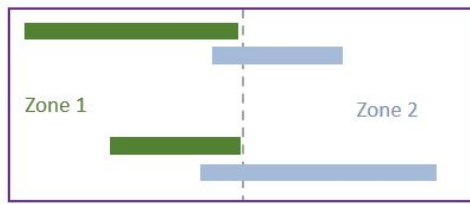
Figure 3.7: Assigning building components into zones, when their corners fall into one or multiple zones. In this example, the building is constructed from Zone 1 to 4 sequentially.

For MEP components, the assignment of components into zones follows the same rules above; however, the sequencing of MEP clusters can be much more complicated. Geometry-related preferences yield extra precedence relations between MEP component pairs; when several component pairs fall into two different clusters, conflicts between clusters can be introduced. A hypothetical scenario is illustrated in Figure 3.8 (a) and (b), where two pairs of components in Zone 1 and 2 have the opposite precedence relations. Such situations are not likely to be found when sequencing the structural system which has less spatial complications.

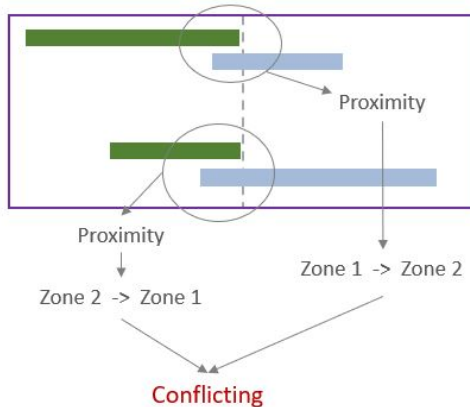
To address such conflicts, a cluster-splitting method is introduced specially for MEP clusters.

Similar to the concept of Design Structure Matrix (DSM), which is a square matrix containing the precedence relationship between a series of tasks in a complex system (Yassine and Braha, 2003), matrices are adopted here to represent the relationship between components for its elegant representation and ability to deal with a large amount of data. Unlike the works of Isaac et al. (2017) which takes a bottom-up approach in assigning components to working packages, a top-down approach is chosen here. In

(a) Find the original cluster each compo belongs to



(b) Identify conflicting precedence relations



(c) One option is to split the cluster in Zone 2 into two subclusters



New precedence relations:
Zone2-2 -> Zone1 -> Zone2-1

Figure 3.8: Conflicting precedence relations between MEP clusters, resulted from geometry-related preferences

the top-down approach all components are assigned to clusters first, and while splitting the clusters, the interruption to the work flow is considered, which is not considered in the bottom-up approach.

The guiding principle for splitting is to minimise the interrupts to the work flow, represented by (1) the total number of splits after resolving all the conflicts, and (2) how concentrated the splits are: it would be preferable in practice, if the total number of (sub)clusters are as small as possible, and if only a few clusters are split.

The relevant six steps in clustering and cluster-splitting for MEP systems are described below, with emphasis on Step 5 which is the key of this cluster-splitting process.

Step 1

Find all MEP component pairs with geometry-related preferences, i.e. SSR and CSR.

Step 2

Create initial clusters and assign all MEP components to them, following the same rules described above for the structural system.

Step 3

Find precedence between original clusters, based on preferred construction sequence, e.g. a cluster in a lower floor before one in a higher floor, or, a cluster with a smaller zone number before one with larger zone number in the same floor.

e.g. ground floor Zone 1 -> ground floor zone 2, or, ground floor Zone 1 -> first floor zone 1.

Step 4

Find all "inter-cluster" component pairs (i.e. one component in the pair falls into a cluster while the other falls into a different cluster), and record the number of such inter-cluster pairs in a matrix.

A hypothetical matrix (Figure 3.9) is created to aid explanation of Step 4 and 5. In this matrix, a to g are the names of the original clusters, and the numbers within represents the number of inter-cluster component pairs which satisfy a certain sequence.

	a	b	c	d	e	f	g
a		10	9			20	
b				7			
c	2	25			4		
d		5			6	3	
e			6	8			
f			15				30
g				4			

Figure 3.9: The original matrix $M_{cluster}$, before splitting. The locations where no value is shown has the value of zero.

For example, number 5 in row d , column b , or location (d, b) , means that, there exist 5 component pairs and each one of them:

- has one component in cluster d and the other in cluster b ; and
- the component in cluster d should be installed earlier than the one in b .

Similarly, number 7 at location (b, d) means that there are 7 component pairs warranting an opposite sequence, i.e. the component in cluster b should be earlier than the one in d .

When the numbers at (i, j) and (j, i) are both nontrivial, a conflict occurs. Those nontrivial numbers leading to such conflicts are marked in red in Figure 3.9.

Step 5

After the clusters containing conflicting sequencing results are identified in Step 4, some of them will be split into subclusters in Step 5. The cluster-splitting method is designed in such a manner that the interruptions resulted is minimal.

Step 5 is an iterative process: in each round, it is first identified which cluster should be split, and then new columns and rows are added to the matrix. This “identification-transformation” is done in several rounds until all the conflicts are resolved. More detailed process is described at the end of this section with an example.

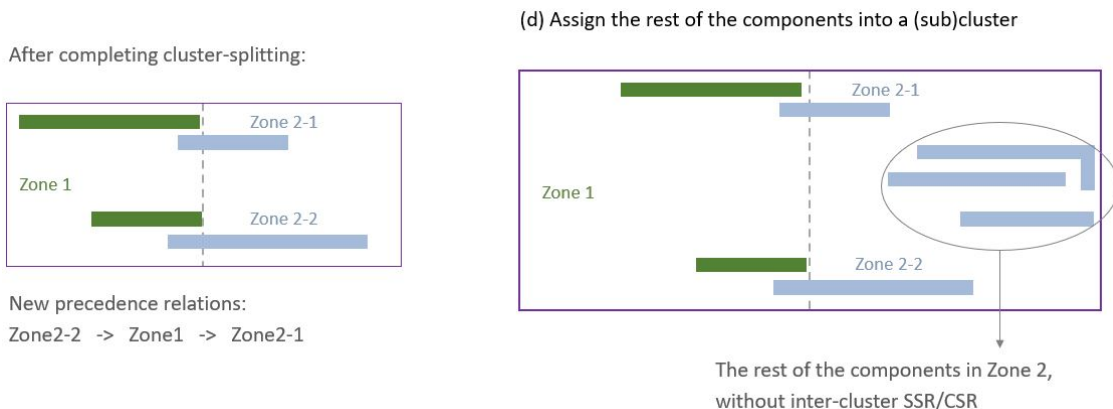


Figure 3.10: Inter-cluster and non-inter-cluster pairs: the former is handled in Step 4 and 5, while the latter is to be assigned into (sub)clusters afterwards

Step 6

Step 4 and 5 deal with inter-cluster pairs with SSR and CSR relations. Among all the MEP components, only some components appear in these pairs and many others do not, e.g. many components have no proximity to other components, or have SSR or CSR relations with only the components within the same zone (Figure 3.10(d)). As mentioned in Ch3.1, atom activities *within* one cluster are assumed to be executed in sequence and intra-cluster scheduling is not discussed in this work; but, those components not in inter-cluster pairs should still be assigned to clusters to be scheduled as a whole. The assignment is done in this final step.

There are two options on how to assign those components not in inter-cluster pairs:

- **Option 1:** Assign them into the original cluster they fall into, or a random one of the subclusters if the original cluster is split in Step 5.
This does not change the total number of (sub)clusters, which is still the lowest possible.
- **Option 2:** Follow the preferences on construction sequence identified in Step 3.
In this option, some new subclusters will be created and thus the total number of clusters is increased, but the preferred construction sequence will be largely followed (except for some components in inter-cluster pairs), which is closer to the reality. The new rules in option 2 are briefly introduced below.

Step 6, option 2 – step 1: For each pair of clusters M_m and M_n : if there is preferred construction sequence (CS) between two clusters M_m and M_n , then compare the precedence yielded from SSR/CSR and CS, and follow the rules in Table 3.2.

If: seq. from construction	If: seq. from SSR/CSR	Then: how to assign the compo
$M_m \rightarrow M_n$	If there exists: $M_{m-e} \rightarrow M_{n-g}$, and $M_{m-f} \leftarrow M_{n-h}$	Assign compo not in inter-cluster pairs to M_{m-e} and M_{n-g}
$M_m \rightarrow M_n$	If there exists: $M_{m-f} \leftarrow M_{n-h}$, and no $M_{m-e} \rightarrow M_{n-g}$	Create new subclusters M_{m-e} and M_{n-g} , and assign compo to M_{m-e} and M_{n-g}

Table 3.2: Scenarios in MEP clustering & cluster-splitting, option 2: how to assign MEP compo without inter-cluster SSR/CSR relations into (sub)clusters. Notation: M_m : cluster m ; M_{m-e} : subcluster $m-e$ created out of cluster m ; $M_m \rightarrow M_n$: M_m is predecessor of M_n

Step 6, option 2 – step 2: For the components not in inter-cluster pairs, check if and when the cluster they fall into is split:

1. If the cluster is split in Step 5, then add the component to a subcluster which is not associated with conflicting precedence yielded from SSR/CSR and CS.
e.g. if $M_{m-2} \rightarrow M_n$ and $M_n \rightarrow M_{m-3}$ from SSR/CSR, and $M_m \rightarrow M_n$ from CS, then add the component to M_{m-2} and not M_{m-3} .
2. If the cluster is split in Step 6, option 2 – step 1, then add the component to any one of the subclusters.

Detailed Step 5 in MEP cluster-splitting

The hypothetical case from Step 4 (Figure 3.9) is used here to explain how the conflicts are resolved by transforming the matrix.

At the start of the first round (Step 5-1), the cluster(s) which has the most conflicts with other clusters are identified, which in this example is cluster d , with three conflicts with other clusters (b , e and g). Cluster d is then chosen to be split in this round, making the subclusters as concentrated as possible.

Six subclusters will then be created out of cluster d : subclusters $d1$ and $d2$ are to accommodate the conflict with cluster b , and similarly, $d3$ and $d4$ for e , and $d5$ and $d6$ for g . Using the matrix representation, that means the column and row d in Figure 3.9 will be replaced by six new columns and rows, which are marked in blue in Figure 3.11a. In the newly created area there is no nontrivial numbers symmetrical to the diagonal line, meaning there is no conflicts between subclusters $d1$ to $d6$ and the rest of the clusters. Step 5-1 is then finished.

	a	b	c	d1	d2	d3	d4	d5	d6	e	f	g
a		10	9									20
b				7								
c	2	25								4		
d1												
d2		5										
d3										6		
d4												
d5												3
d6												
e			6			8						
f			15									30
g									4			

(a) Sub-step 5-1: split cluster d

	a	b	c1	c2	c3	c4	c5	c6	d1	d2	d3	d4	d5	d6	e	f	g
a		10			9												20
b									7								
c1			25														
c2																	
c3																	
c4	2																
c5															4		
c6																	
d1																	
d2		5															
d3															6		
d4																	
d5																	3
d6																	
e								6			8						
f				15													30
g														4			

(b) Sub-step 5-2: split cluster c

Figure 3.11: Splitting of matrix $M_{cluster}$

Similarly, in the next sub-step (Step 5-2), cluster c is found to be the one with most conflicts (2 conflicts) with the rest (a and e), resulting in four new subclusters. But, what is different from Step 5-1 is that row c and column c also contain two numbers in green colour: number 25 at (c, b) , and number 15 at (f, c) ; these two numbers do not cause conflicts, but should also be included inside a subcluster. Therefore, cluster c will be split into $(4 + 2 =) 6$ subclusters: subcluster $c1$ and $c2$ to accommodate the relations with b and f , and subcluster $c3$ to $c6$ for the conflicts with a and e .

At the end of Step 5-2, it is observed that there is no nontrivial numbers symmetrical to the diagonal

line, which means all the conflicts are resolved. The resulted matrix has 17 (sub)clusters, out of which 12 are split from 2 original clusters. For comparison, an example of result by using an alternative and less desirable splitting method is shown in Appendix A, and that results in 23 (sub)clusters.

During the matrix transformation, when new columns and rows are created, the numbers within the original column and row are distributed in a specific way to four new areas, shaded in green and blue in Figures 3.11a and 3.11b: top, bottom, left, and right. The mathematical representation of the matrix transformation – the change in the indices of the nontrivial numbers are presented in Appendix A.

3.4. Mathematical formulation of the optimisation problem

A schedule optimisation problem can be formulated as different types of problems, e.g. linear and nonlinear. In this study, mixed-integer linear problem (MILP) is chosen, for its rigorousness, flexibility and extensive modeling capability, and the availability of efficient commercial solvers (Floudas and Lin, 2005; Urbanucci, 2018). The parameters of the optimisation problem, including decision variables, data, objectives, and constraints, are listed below.

The rest of this section will use the following symbols and subscripts:

Activity a	$a = 1, 2, \dots, A$
Resource r	$r = 1, 2, \dots, R$
Day d	$d = 1, 2, \dots, D$
Mode m	$m = 1, 2, \dots, M$

Decision variables

Decision variables are the unknown variables whose values are to be found through the optimisation; in this study, the starting and finishing days of activities and mobilisation of resources are treated as discrete decision variables. In addition, extra binary variables are added to facilitate logical modelling and maintain the linearity of the objectives and constraints. All variables are integers; some are binary (0 or 1).

ES_a, EF_a	Start and finish day of Activity a
ES_{ar}, EF_{ar}	Start and finish day of Resource r mobilised in Activity a
M_{arm}	Binary variables showing the mode of Resource r mobilised in Activity a : for example, if mode 1 is chosen, then $M_{ar1} = 1$ and $M_{arm} = 0, m = 2, \dots, M$
U_{ard}	Binary variables showing the utilisation of Resource r in Activity a on Day r ; equals to 1 if it is utilised, 0 otherwise
Z_{ard}, Z'_{ard}	Binary indicators to facilitate logical modelling

Data

The following data is treated as known inputs which define the optimisation problem.

Q_{rd}	Number of Resource r available on Day d
C_{arm}	Unit cost of Resource r utilised on Activity a in Mode m [euro/hr]
E_{arm}	Efficiency of Resource r utilised on Activity a in Mode m [hr/element]
N_a	Number of elements in Activity a (assumed to be the same for different resources and modes)
Z_x, Z_y	Dimensions of rectangular zones (which decides N_a)
$p_i, i = 1, 2, \dots, I$	List of activities which are followed by other activities
$s_i, i = 1, 2, \dots, I$	List of activities which are following other activities
$b_i, i = 1, 2, \dots, I$	Buffer time between the finish day of activity p_i and the start day of activity s_i

Objectives

Two common goals in construction scheduling are considered in this multi-objective optimisation (MOO) problem: to minimise the total direct cost, and to minimise the total duration. These two objectives generally move in the opposite directions: with a higher direct cost, e.g. by allocating more resources, a shorter activity and project duration may be achieved (Feng et al., 1997; Alavipour and Arditi, 2019).

Objective 1: to minimise project total direct cost:

$$\sum_{a=1}^A \sum_{r=1}^R \sum_{m=1}^M C_{arm} * (E_{arm} * N_a * M_{arm}) \quad (3.1)$$

Only construction activity-related labour and equipment cost, or direct cost, is accounted for in the objective function (3.1).

Objective 2: to minimise project total duration

$$EF_A \quad (3.2)$$

The project total duration equals to the finishing day of the dummy finish activity. However, this term together with the constraints in the next section do not guarantee:

1. that ES_a and EF_a take the minimum value possible when it is solely based on the constraints (3.10) to (3.14), when it is not on the critical path;
2. that ES_a take the maximum value of ES_{ar} based on constraints (3.5), e.g. while the earliest date of resource utilisation on Activity 3 is day 5, ES_3 may take the value of 4.

To help push the activities forward to as early as possible while keeping an accurate starting date, four additional terms are added to the objective function (3.3), changing the objective to:

$$EF_A + \rho_1 * \sum_{a=1}^A EF_a + \rho_2 * \sum_{a=1}^A \sum_{r=1}^R EF_{ar} + \rho_3 * \sum_{a=1}^A ES_a + \rho_4 * \sum_{a=1}^A \sum_{r=1}^R ES_{ar} \quad (3.3)$$

where ρ_1 to ρ_4 is the coefficient to adjust the weighting of the additional terms: $\rho_1 = 1/(A * D/2)$, $\rho_2 = 1/(A * D * R/2)$, $\rho_3 = -1/(A * D/4)$, $\rho_4 = -1/(A * D * R/4)$.

A posteriori method and Pareto fronts

For insights into the trade-off between two objectives, a posteriori method is used: a series of different weights are given to the objective functions (3.1) and (3.3) to obtain the weighted sum, transforming the problem into a series of single-objective ones. They yield a series of (optimal) solutions, with each solution entailing one value for each of the two objectives.

A collection of optimal solutions forms a Pareto front, which, in this study, is expected to be a curve spanning from top-left to bottom-right in a “time-cost” plot. For decision makers, Pareto fronts can offer important insights into the trade-off between the objectives; all points on the Pareto front are optimal solutions from which decision makers can choose based on their preferences.

But before the summation, both values are first normalised to a comparable scale, by using the following conversion:

$$O_{normalised} = \frac{O_{original}}{O_{max} - O_{min}} \quad (3.4)$$

where O_{min} and O_{max} are the theoretical minimum and maximum value of the objective before normalisation.

Constraints

The optimisation is subject to a series of conditions which limit the solution space. A feasible solution of the optimisation problem should satisfy all the constraints. The constraints in this formulation are listed below.

The starting and finishing dates of activities and resources are connected by Eqs. (3.5) to (3.7). The activity precedence relationship is implemented by Eq. (3.8). The maximum quantity of resources on each day is constrained by Eq. (3.9). Eqs. (3.10) to (3.14) are one set of equations to make sure the binary variables $U_{ard} = 1$ when $ES_{ar} \leq d \leq EF_{ar}$, and $U_{ard} = 0$ otherwise, using binary indicators Z_{ard} and Z'_{ard} as the connection in between. Eq. (3.15) is based on the definition of the modes.

$$ES_a \leq ES_{ar}, \quad \forall a = 1, 2, \dots, A, \quad \forall r = 1, 2, \dots, R \quad (3.5)$$

$$EF_a \geq EF_{ar}, \quad \forall a = 1, 2, \dots, A, \quad \forall r = 1, 2, \dots, R \quad (3.6)$$

$$EF_{ar} = ES_{ar} + \sum_{m=1}^M M_{arm} * E_{arm} * N_a - 1, \quad (3.7)$$

$$\forall a = 1, 2, \dots, A, \quad \forall r = 1, 2, \dots, R$$

$$EF_{p(i)} + b(i) + 1 \leq ES_{s(i)}, \quad \forall i = 1, 2, \dots, i \quad (3.8)$$

$$\sum_{a=1}^A U_{ard} \leq Q_{rd}, \quad \forall r = 1, 2, \dots, R, \quad \forall d = 1, 2, \dots, D \quad (3.9)$$

$$ES_{ar} = D + 1 - \sum_{d=1}^D Z_{ard}, \quad \forall a = 1, 2, \dots, A, \quad \forall r = 1, 2, \dots, R \quad (3.10)$$

$$Z_{ard} \geq Z_{ar(d-1)}, \quad \forall a = 1, 2, \dots, A, \quad \forall r = 1, 2, \dots, R, \quad \forall d = 1, 2, \dots, D \quad (3.11)$$

$$EF_{ar} = \sum_{d=1}^D Z'_{ard}, \quad \forall a = 1, 2, \dots, A, \quad \forall r = 1, 2, \dots, R \quad (3.12)$$

$$Z'_{ard} \leq Z'_{ar(d-1)}, \quad \forall a = 1, 2, \dots, A, \quad \forall r = 1, 2, \dots, R, \quad \forall d = 1, 2, \dots, D \quad (3.13)$$

$$Z'_{ard} + Z'_{ar(d-1)} \leq U_{ard} + 1, \quad \forall a = 1, 2, \dots, A, \quad \forall r = 1, 2, \dots, R, \quad \forall d = 1, 2, \dots, D \quad (3.14)$$

$$\sum_{m=1}^M M_{arm} = 1, \quad \forall a = 1, 2, \dots, A, \quad \forall r = 1, 2, \dots, R \quad (3.15)$$

Initial solutions

A MILP problem is NP-hard and even a small problem can take a long time. Initial solution(s) can be provided when solving a MILP problem, which may change the amount of time *intlinprog* takes to converge with certain heuristics.

In this work, initial solutions are also useful if a feasible solution cannot be found within a reasonable time frame when the optimisation problem is large. The procedures to generate an initial solution is shown in Appendix B.

4

Software Implementation

To implement and test the conceptual framework described in Chapter 3, a software tool is developed. This chapter presents the workflow of using the tool, what each component of the tool does, and shows the user interface (UI) of the tool (Ch4.2). The plans for verification and validation are also introduced (Ch4.3).

4.1. Functional requirements

To implement the conceptual framework, the following functionalities should be met:

1. Map individual building components with their activity, duration, cost and resource data in the database;
2. Formulate and solve the optimisation problem;
3. Output the optimised schedule.

In addition, two extra desirable features are added:

1. Integration with BIM: though the output schedule will list only activity clusters, it is produced based on component-level information, and each task in it can be traced back to its components. The granularity and large amount of information (at component-level) can be facilitated by automatic extraction of information from BIM. Furthermore, the output schedule should be able to be automatically coupled with the building model for the advantages of 4D BIM.
2. It is desirable that the tool has a wide applicability to a general construction project, which means that some problem settings should allow customisation and project-specific preferences from user.

4.2. The software tool

A software tool is developed specifically for this study. It consists of three parts: an Excel add-in, a MATLAB standalone executable file, and an Excel macro. The overall workflow is shown in Figure 4.1, and the functions of each part in Table 4.1.

More details are explained next on how the major functions are implemented and how each part of the tool works.

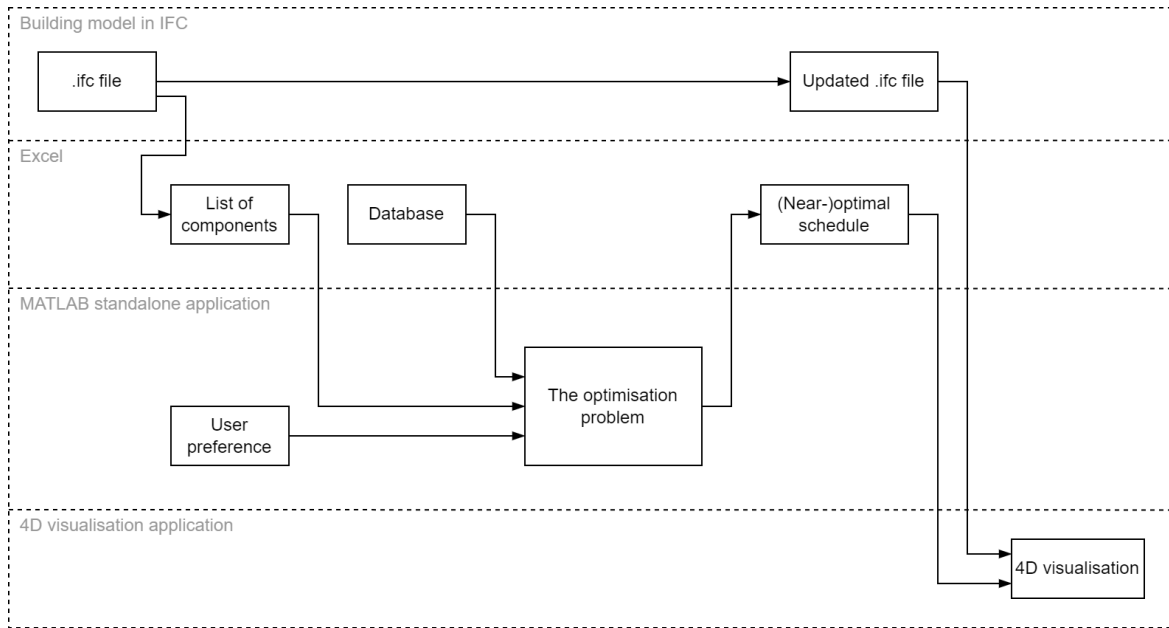


Figure 4.1: Workflow of the tool

An Excel add-in	Extract component information from BIM (.ifc) files, and write into Microsoft Excel spreadsheet (.xlsx)
A MATLAB standalone executable programme	<ul style="list-style-type: none"> • Read component information from .xlsx file • Map the components with their activity duration, cost and resource data • Take project-specific preferences from user • Formulate, solve and display the result of optimisation • Write the schedule into .xlsx file
An Excel macro	Write the schedule from .xlsx into Microsoft Project (.mpp) file

Table 4.1: Three parts of the software tool and their functions

4.2.1. The Excel add-in: extraction of component information from BIM

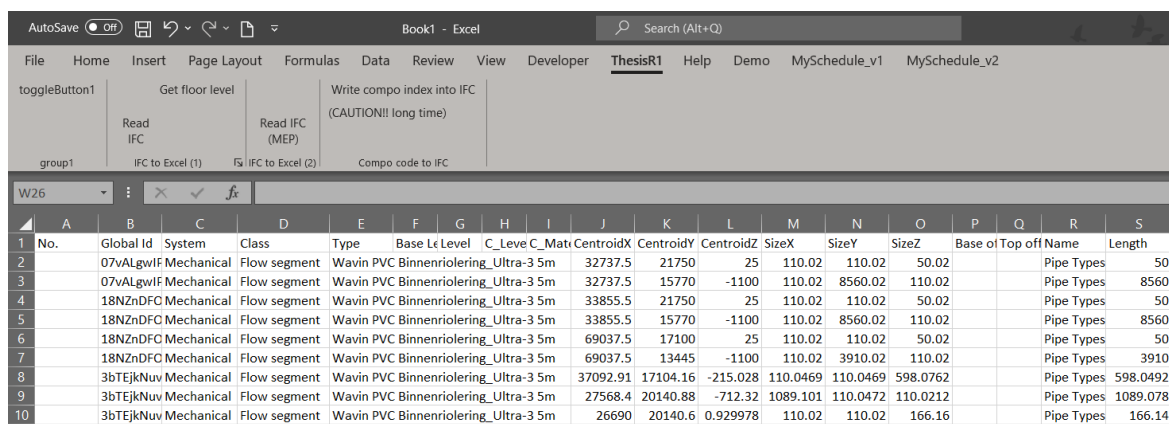


Figure 4.2: UI of the Excel add-in and sample output

As the first step, building model(s) in IFC4 file format are used to produce a list of all building components to be included when scheduling, together with their properties, including Global Unique ID (or GUID), class, floor level if applicable, and the location and size of their bounding box. This information

extraction process is done via an Excel add-in (a screenshot of the UI and sample output in Figure 4.2), written in C# using Microsoft Visual Studio 2019. To read information from .ifc files, two .NET open-source software development BIM toolkits, Xbim Essentials and Xbim Geometry, are used (Lockley et al., 2017).

One additional and optional step is taken in one of the case studies in this project: each building component is assigned a unique component ID of 5 digits, as custom property in a new .ifc file. The new ID's serve the same purpose of 22-character-length GUID, except for that they are shorter and makes it more convenient when coupling the schedule and components in a later step.

4.2.2. The MATLAB programme: setting up the optimisation problem

As the key part of the software tool, a MATLAB executable programme is developed to meet most of the functional requirements, taking as major input the component information from the previous step, and as output the optimised schedule. A UI is designed with several consecutive tabs for the user to follow step by step. The functions covered are explained below.

Further processing of component information

The component information in .xlsx from the previous step is likely not ready for direct usage, due to inconsistency in units, class names, etc. In this step the user has the options to:

- Convert unit of length;
- Filter the components;
- Modify material and class names into recognisable ones.

An example is shown in Figure 4.3, on the filtering of components. More explanations on the processing of component information can be found in Appendix E.

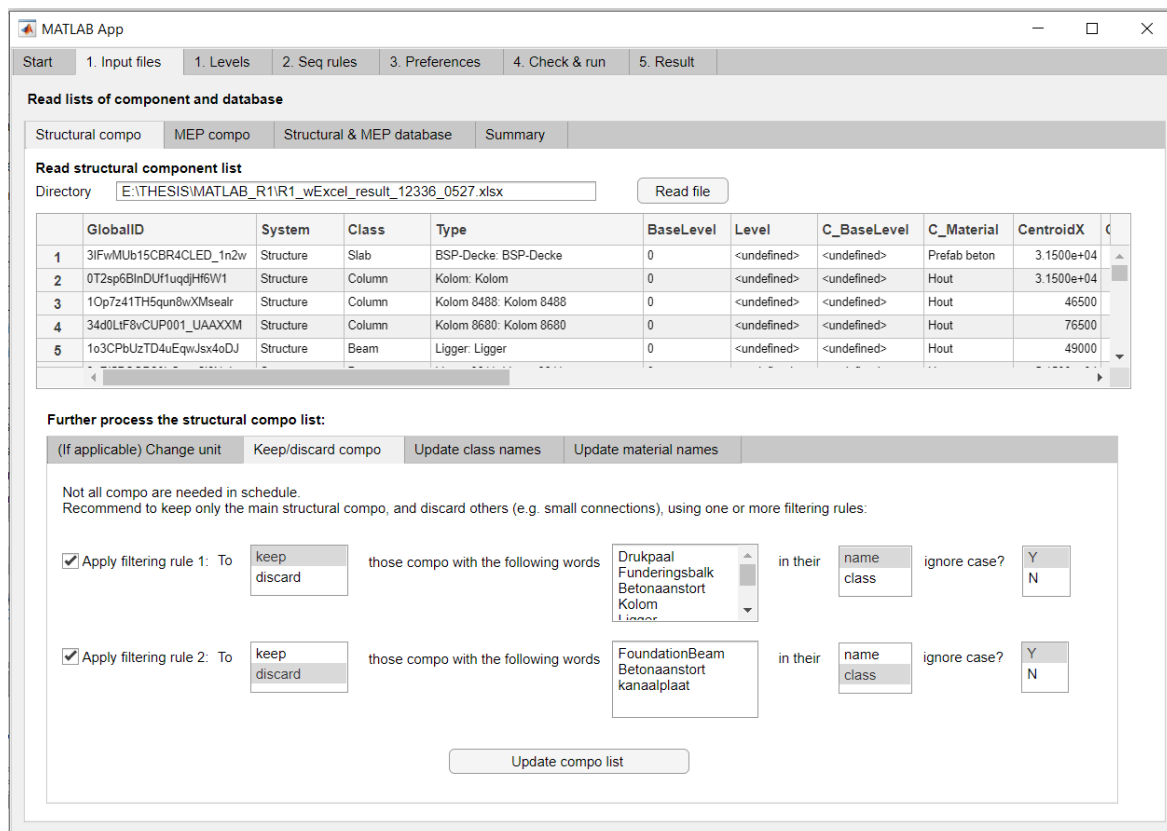


Figure 4.3: UI of the MATLAB executable programme: processing building component spreadsheet

Database and mapping

In this project, each building element is associated with one and only one atom activity, and each atom activity is associated with a cost and duration value. A database is compiled for this purpose, based on practical knowledge and experience. A series of element properties are included in the database, and it is specified which combinations of properties are associated with which cost and duration data.

A sample database file in .xlsx is shown in Figure 4.4.

No.	System	Class	Type	Material	Level	Size			Resource	Res1Mode1	Efficiency		Cost_r1m		Efficiency_r1m2	Resource_2	Res2Mode1
						Max Size_1	Max Size_3	Category			Cost_r1m1 (euro/hr)	Efficiency_r1m1 (hr/ele)	Cost_r1m2 (euro/hr)	Efficiency_r1m2 (hr/ele)			
1btp_of_S	structure	Column	Column	Prefab betoi OF		500	7000	S	Mountage	1crane+1operator	295	0.84	1crane+1operato	500	0.63	NA	NA
1btp_of_M	structure	Column	Column	Prefab betoi OF		700	9000	M	Mountage	1crane+1operator	295	1.04	1crane+1operato	500	0.78	NA	NA
1btp_of_L	structure	Column	Column	Prefab betoi OF		1000	12000	L	Mountage	1crane+1operator	295	1.25	1crane+1operato	500	0.94	NA	NA
1btp_of_M	structure	Column	Column	Prefab betoi OF		700	16000	XL	Mountage	1crane+1operator	295	1.46	1crane+1operato	500	1.10	NA	NA
1btp_of_L	structure	Column	Column	Prefab betoi OF		1000	20000	XXL	Mountage	1crane+1operator	295	1.88	1crane+1operato	500	1.41	NA	NA
1btp_of_S	structure	Column	Column	Prefab betoi Above DF		500	7000	S	Mountage	1crane+1operator	295	0.67	1crane+1operato	500	0.50	NA	NA
1btp_of_M	structure	Column	Column	Prefab betoi Above DF		700	9000	M	Mountage	1crane+1operator	295	0.83	1crane+1operato	500	0.62	NA	NA
1btp_of_L	structure	Column	Column	Prefab betoi Above DF		1000	12000	L	Mountage	1crane+1operator	295	1	1crane+1operato	500	0.75	NA	NA
1btp_of_M	structure	Column	Column	Prefab betoi Above DF		700	16000	XL	Mountage	1crane+1operator	295	1.17	1crane+1operato	500	0.88	NA	NA
1btp_of_L	structure	Column	Column	Prefab betoi Above DF		1000	20000	XXL	Mountage	1crane+1operator	295	1.5	1crane+1operato	500	1.13	NA	NA
2btp_of_S	structure	Beam	Beam	Prefab betoi OF		500	10000	S	Mountage	1crane+1operator	295	0.2	1crane+1operato	500	0.15	NA	NA
2btp_of_M	structure	Beam	Beam	Prefab betoi OF		700	15000	M	Mountage	1crane+1operator	295	0.3	1crane+1operato	500	0.23	NA	NA
2btp_of_L	structure	Beam	Beam	Prefab betoi OF		1200	20000	L	Mountage	1crane+1operator	295	0.5	1crane+1operato	500	0.38	NA	NA

Figure 4.4: Sample database file containing resource, cost, duration information

Identification of floor level; customisation of sequencing rules

Floor level information is sometimes either missing or inaccurate for some components; that is dealt with by customisable rules for identification of floor level, based on coordinates in relation to floor elevations and user-specified tolerance. Then the user can adjust sequencing rules by ranking structural component classes, or, changing the parameters used in identifying special spatial relations (SSR) and component size relations (CSR) in MEP systems (only option 2 described in Ch3.3 is integrated).

Both the identification of floor level and customisation of sequencing rules allow for: (1) one set of general rules, and (2) one or two sets of optional rules to fine-tune for special cases. As an example, Figure 4.5 shows the section of UI where sequencing rules for the structural system can be adjusted. More explanations can be found in Appendix E.

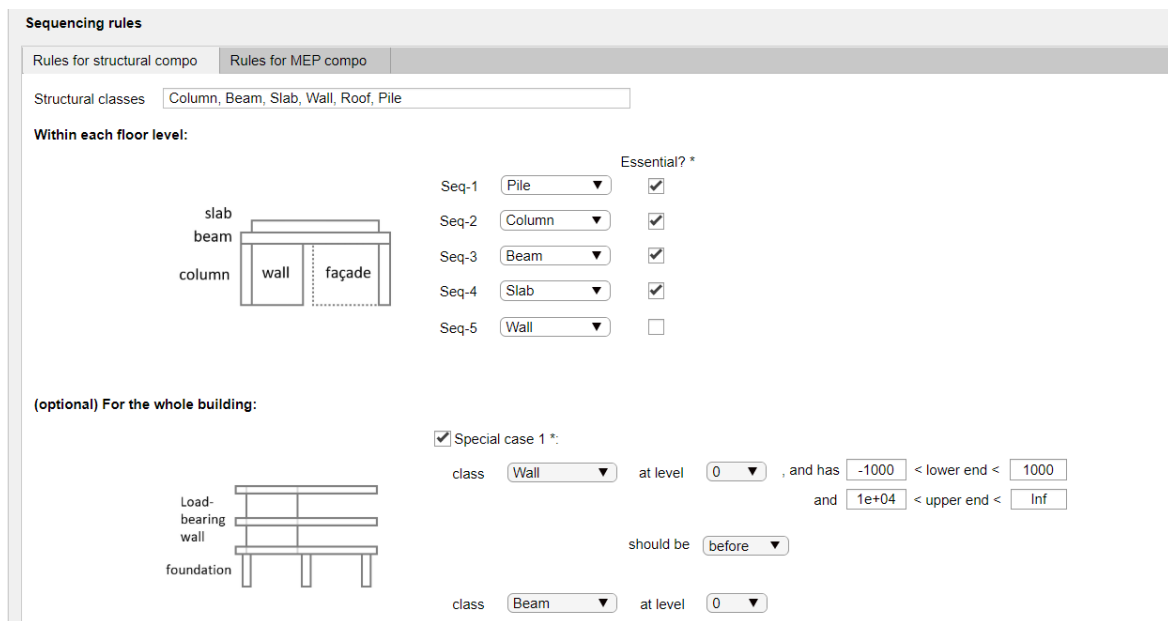


Figure 4.5: UI of the MATLAB executable programme: customisation of sequencing rules (structural system). In the upper half of this tab, general sequencing rules can be set, by selecting component classes from the dropdown lists in the desirable sequence; in the lower half, one special rule can be set if the general rules cannot account for all preferences.

Setting working zone and optimisation options

Here the user specifies:

1. Project-specific preferences, e.g. the minimum and maximum dimension of zone, maximum resource availability, preferred construction sequence (e.g. from which corner the construction starts);
2. Preferences on optimisation problem solving, e.g. maximum running time till termination.

Solving the optimisation problem

The multi-objective optimisation (MOO) problem is formulated as explained in Ch3.4 and solved by using *intlinprog* solver in MATLAB Optimisation Toolbox.

If satisfied with the scheduling result, the user can select a project starting date and export the schedule as an .xlsx file.

4.2.3. The Excel macro: preparing the schedule for 4D visualisation

This software tool does not entail an application for 4D visualisation, thus an external application should be used. In this project, a commercial application Synchro 4D by Bentley is chosen for its wide adoption in the industry.

A schedule in .xlsx from the last step may not be directly usable in 4D BIM applications; the file conversion is done by the Excel macro, which automatically writes the schedule from .xlsx to .mpp in Microsoft Project, which can then be saved in .xml format. The schedule and the building model can then be easily imported into Synchro 4D, and each task in the schedule can be automatically mapped with its corresponding individual components.

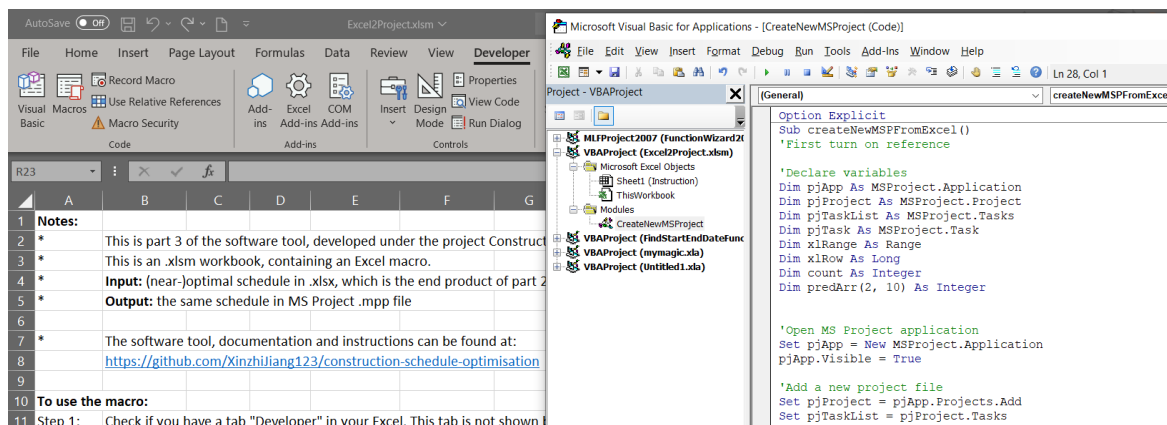


Figure 4.6: The Excel macro workbook

More detailed instructions on the identification of floor level, and customisable sequencing rules can be found in Appendix E. A complete list of customisable settings can be found in Appendix D.

The software tool, including all three parts, together with installation guide and instructions on how to use it, is publicly accessible at its GitHub page (link).

4.3. Verification and validation

Verification

Three cases are used to verify the methodology and software implementation:

- A simple one-storey structure. The building model is largely simplified from a real-life building so that the scale is reasonably small for manual checking.
- Two real-life projects as case studies.

Verification - part 1: a simple one-storey structure

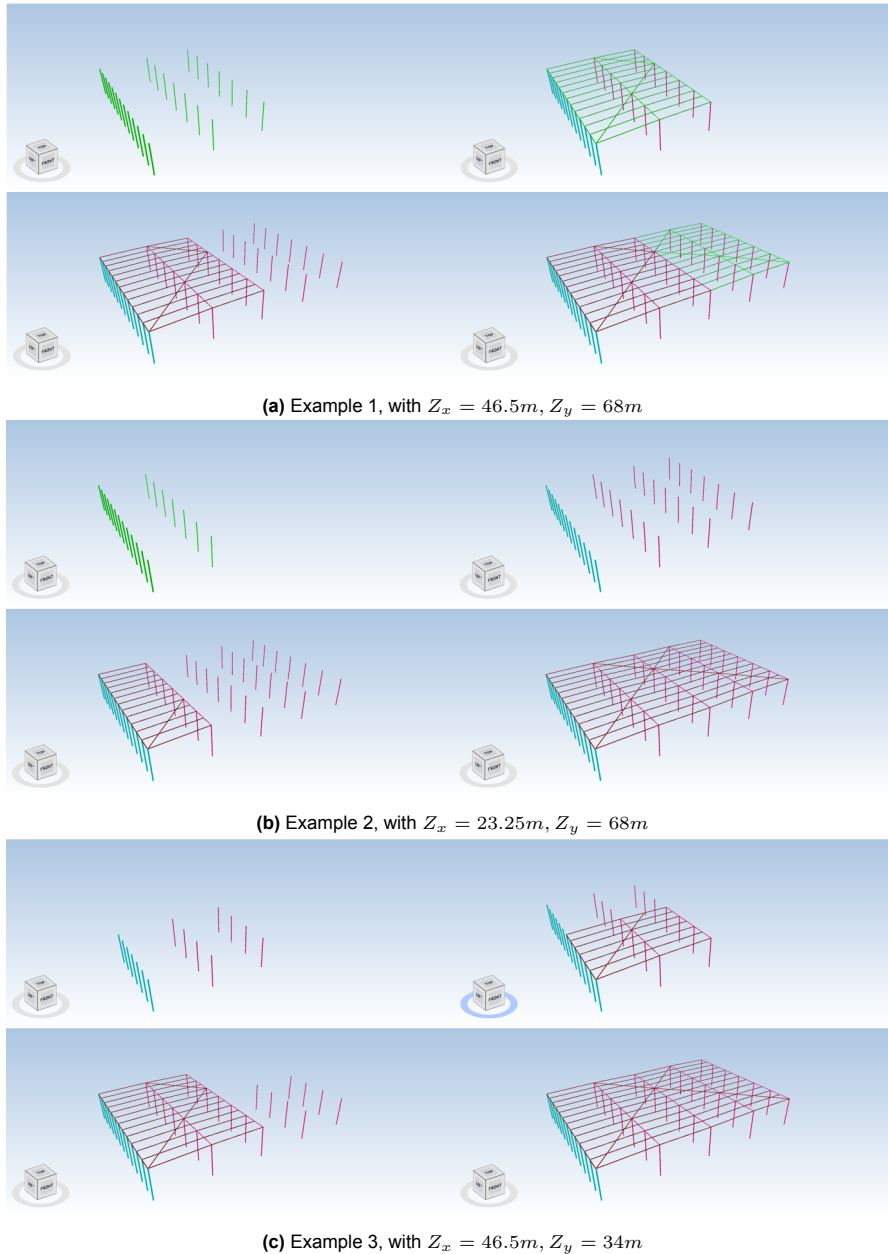


Figure 4.7: Scheduling results of a simplified structure for verification

Verification on date extraction, i.e. to check if the list of components and their properties extracted from the .ifc file is correct, is done by manually sampling components and comparing the information on the list and the information shown in 3D BIM software application (Revit and BIMsync in this study).

Verification on component zone assignment, clustering and cluster-splitting: this is checked by: (1) comparing the sum of the cost and duration value of all components in the component list, and the sum of the cost and duration value of all activity clusters, to check if there is duplicate or missing activities; (2) manually sampling component-level data and checking if the associated cost, duration data and zone number is correct.

Verification on the optimisation problem setting and solutions: this is checked by examining the 4D visualisation of the construction process produced by coupling the schedule generated and the original 3D BIM model, to see if the construction process is logical and follows the sequencing rules, and if there are components left out (Figure 4.7).

The results show that the tool can successfully perform the functions.

Verification - part 2: two case studies

Two real-life projects:

- Case 1: an industrial building, with 3-storey wood structure, and relatively complicated MEP system. This will be presented in Ch5.1.
- Case 2: a simplified distribution centre, with 1-storey steel structure, and large floor area of homogeneous (repetitive) design. This will be presented in Ch5.2.

In both case studies, checkings similar to the ones described above are conducted. The checkings in the case studies show that the outputs (scheduling results) of the developed tool are yielded correctly from the inputs (building model, database and set preferences). Furthermore, the MEP clustering and cluster-splitting is verified by checking the scheduling results of Case 1. With successful verification in all three cases, it is considered that the developed tool together with the methodology implemented in the tool are verified.

Validation

Validation on if the developed tool serves the research and development goal is done by using the case studies in the next chapter.

5

Case Studies

Two real-life projects are chosen, to verify and validate the conceptual framework and software implementation:

- Case 1: an industrial building, with 3-storey wood structure, and relatively complicated MEP system
- Case 2: a simplified distribution centre, with 1-storey steel structure, and large floor area of homogeneous (repetitive) design.

The first case is used to investigate the capacity of the developed tool in the context of a complicated building system, with multiple floors and element classes, and spatial complexity in a relatively congested space; the second case is chosen because of its large floor area and repetitive design which reduces the influence by the spatial complexity, and thus is suitable for investigating the cost-duration trade-off related to the design of working zones.

This chapter introduces the cases and records the results. Each section starts with an introduction on the project feature and problem setting, and then presents and discusses the results.

5.1. Case 1: HoutWerk, Utrecht

5.1.1. Introduction of the case and problem setting

Case 1: an industrial building, of 3-storey wood structure, with relatively complicated MEP system

This industrial building locates on Campus Werkspoor, Utrecht. It has three storeys and the gross floor area of approximately $3,400 \text{ m}^2$. Almost the entire supporting structure is made of wood (except for the foundation and ground floor), enabling the reduction of CO_2 emission by more than half comparing with that of an equivalent concrete structure. It is delivered in 2021 after a realisation period of nine months. Pictures and floor plans of the building are shown in Figure 5.1.

The component classes and their associated resources are recorded in Table 5.1. For MEP systems, only flow segments such as ducts and pipes in the mechanical system are considered in the case study.

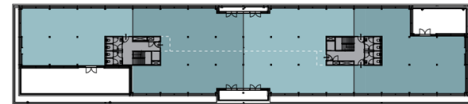
5.1.2. Results

The case is analysed under a series of different problem settings:

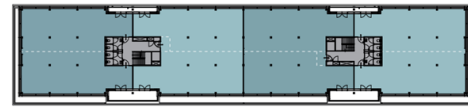
1. Different MEP component SSR and CSR preferences: this is determined by the value of D_{minH} , D_{minV} , $D_{minWall}$, etc. (details in Ch3.3). Seven sets of these values are tested; the clustering and cluster-splitting results are recorded in Figure 5.2a.



(a) As-built of the building



2e verdieping



1e verdieping



Begane grond

(b) Floor plan of the second, first and ground floor, from top to bottom



(c) During construction (1)



(d) During construction (2)

Figure 5.1: The HoutWerk: pictures during and after construction, and the floor plans.

System	Component class	Associated resource
Structural, essential	Foundation, column, beam, slab, load bearing wall	Mountage team
Structural, non-essential	Facade	Mountage team
MEP	Flow segment	Installation team

Table 5.1: Problem setting of case 1: components and resources.

2. Different zone dimensions: seven sets of zone dimensions are tested, under the same SSR and CSR preferences; the in- and output parameters are recorded in Figure 5.2b

One example of scheduling result is visualised in Synchro 4D (Figure 5.3); a full schedule is included in Appendix C.

5.1.3. Discussion

Verification of the sequencing of the MEP system

The sequencing of MEP system is checked in this case study, to check if the method proposed in Ch3.2.2 and Ch3.3 can successfully identify geometry-related sequencing preferences (special spatial

		Seven different settings on SSR & CSR preferences						
		1	2	3	4	5	6	7
Setting - MEP component sequencing preferences								
No. structural components		679						
No. MEP components		2215						
Check special spatial relations (SSR) (mm)	D_{minH}	100	300	300	500	300		
	D_{minV}	100	300	300	500	300		
	D_{minWall}	800	800	1000	1000	800		
	D_{minCeiling}	800	800	1000	1000	800		
Check component size relations (CSR) (mm)	D_{minH2}	300				300		
	D_{minV2}	300				300		
	L_{lengthLong}	12000				20000	12000	20000
	L_{lengthShort}	1000				300	1000	300
	A_{secLarge} (mm2)	80000				80000	100000	100000
	A_{secSmall} (mm2)	1200				1200	500	500
Result - clustering & splitting								
No. compo pairs with SSR		172	547	585	784	547	547	547
No. compo pairs with CSR		154	154	154	154	133	71	50
No. compo pairs with SSR and non-conflicting CSR		311	646	684	883	629	601	584

(a) Case study 1: different MEP component SSR and CSR preferences and their clustering & splitting results

		Seven different zone dimensions						
		1	2	3	4	5	6	7
Input parameters								
Zx (m)		90	45	32	30	29	22.5	22
Zy (m)		35	35	35	35	35	35	35
No. zones along x		1	2	3	3	3	4	4
No. zones along y		1	1	1	1	1	1	1
No. act. clusters: structural		14	28	42	42	42	56	56
No. act. clusters: MEP (before)		4	8	12	12	12	16	16
Result - clustering & splitting								
No. compo pairs with SSR		547						
No. compo pairs with CSR		154						
No. compo pairs with SSR and non-conflicting CSR		646						
No. compo pairs with SSR and non-conflicting CSR, present at zone boundary		2	3	9	10	8	13	12
No. act. clusters: MEP (after)		4	8	15	17	14	21	19
Cluster(s) split		-	-	F0Z2 F1Z2	F0Z2 F1Z2	F1Z1	F0Z1 F0Z3 F1Z2 F1Z3	F0Z1 F1Z3
Result - scheduling								
Total duration (days) - option 1		50	52	57	57	57	52	54
Total duration (days) - option 2		50	51	51	51	57	52	52

(b) Case study 1: in- and output parameters with different zone dimensions (varying zone dimension along x, fixed zone size along y).

relations and component size relations), and address conflicting preferences at cluster- and component-level. Two places where such conflicts exists are shown next (Figure 5.4 and 5.6): the scheduling results generated by the tool will be checked against the theoretical analysis. In addition, example 1 is used to demonstrate the difference between option 1 and 2 in Step 6 of cluster-splitting (Ch3.3): one

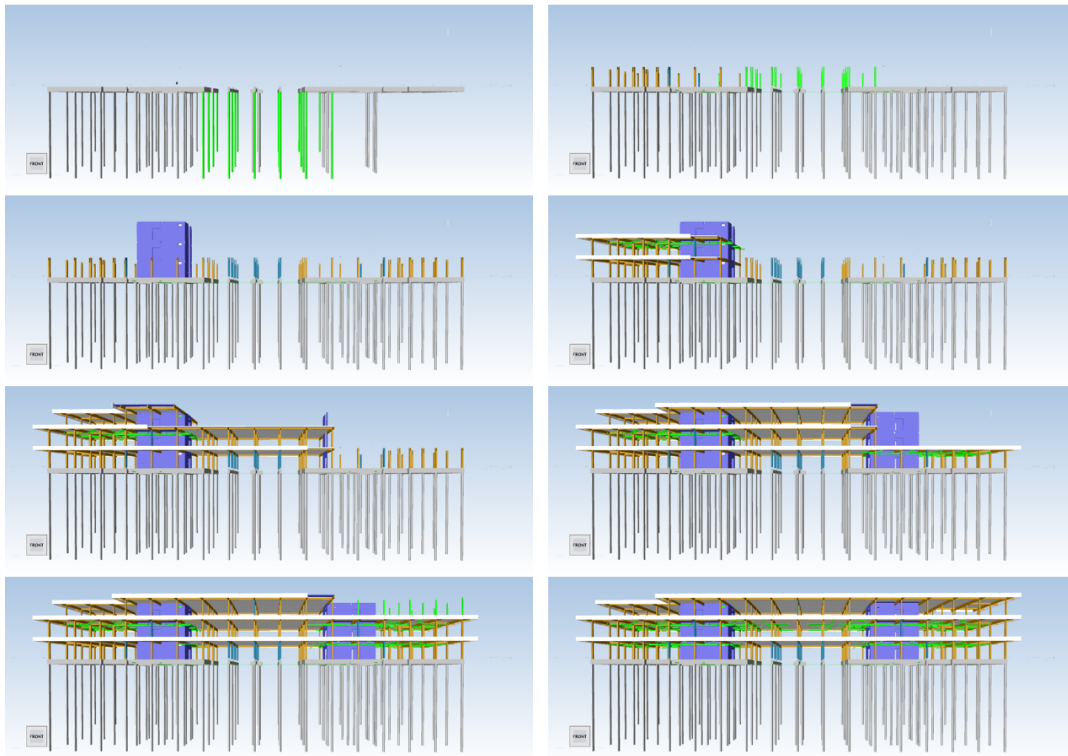


Figure 5.3: One example of the generated project schedule, for both structural and MEP systems, visualised in Synchro 4D

results in the minimal total number of subclusters, while the other considers preferences on construction sequence.

The first example (Figure 5.4a) shows where Zone 2 and 3 in Floor 1 adjoin. From visual inspection, several component SSR/CSR relations can be identified:

1. Based on spatial relations, component *0802* in Zone 2 is directly above, and thus is to be installed earlier than components *0144* and *0667* in Zone 3 (i.e. Zone 2 -> Zone 3).
2. Based on component lengths and cross-sectional sizes, component *0805* in Zone 3 is found to be long and much longer, and thus is to be installed earlier, than components *0857* and *0803* in Zone 2 (i.e. Zone 3 -> Zone 2).

The above precedence relations result in conflicts between Zone 2 and 3 in Floor 1.

Two schedules are generated using option 1 and 2 in Step 6 of cluster-splitting respectively, and the resulted schedules are visualised after coupling with the building model (Figures 5.4 and 5.5). Based on visual examination, all precedence relations identified above are captured and satisfied in the schedules.

Figures 5.4 and 5.5 also show the difference of option 1 and 2 on when the majority of components in Zone 2 and 3 are installed:

With option 1:

Zone 2-sub (compo *0802*)

-> Zone 3 (the majority/all of components)

-> Zone 2-sub (the majority of components)

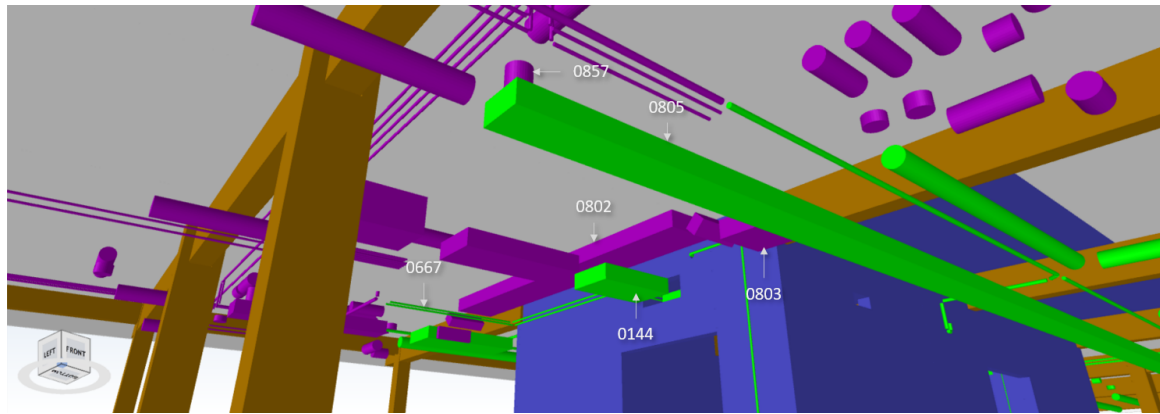
With option 2:

Zone 2-sub (the majority of components)

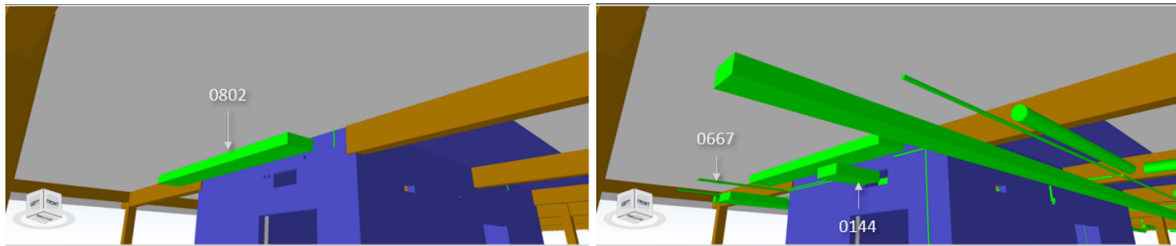
-> Zone 3 (the majority/all of components)

-> Zone 2-sub (compo *0857*, *0803*)

The preferences on construction sequence is set to be “smaller zone number -> larger zone number, in

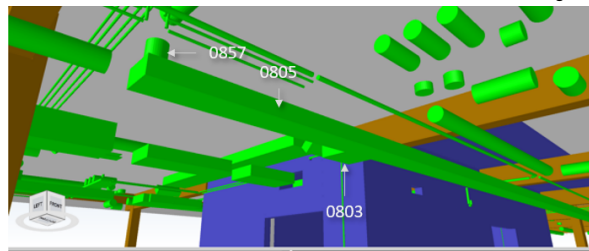


(a) The original building model. Components in purple: Floor 1, Zone 2; Components in green: Floor 1, Zone 3. Only component of class "pipe segments" are considered; pipe fittings, i.e. the connection between pipe segments, are not considered in the optimisation problem and thus not shown in the screenshot.



(b) First, component 0802 (which has SSR relation with 0144 and 0667) in Zone 2 is installed.

(c) Next, the majority/all of components in Zone 3 are installed, including 0144 and 0667.



(d) Finally, the majority of components in Zone 2 are installed, including 0857 and 0803 (which have CSR relation with 0805 in Zone 3).

Figure 5.4: Result of MEP sequencing in case study 1, example 1, using option 1 in cluster-splitting & assignment

the same floor" in this case study, which means most components in Zone 2 are to be installed before those in Zone 3. With option 1, this preference is disregarded; but it is satisfied with option 2, only except for a small amount of components with inter-cluster SSR/CSR relations.

The second example (Figure 5.6) can be checked in a similar manner.

Through this case study, the proposed method on MEP sequencing, i.e. clustering and cluster splitting method described in Ch3.3, is verified.

5.2. Case 2: a simplified distribution centre

5.2.1. Introduction of the case and problem setting

Case 2: a simplified distribution centre, with 1-storey steel structure, and large floor area of homogeneous (repetitive) design

The component classes and resources considered in the case study are listed in Table 5.2. After

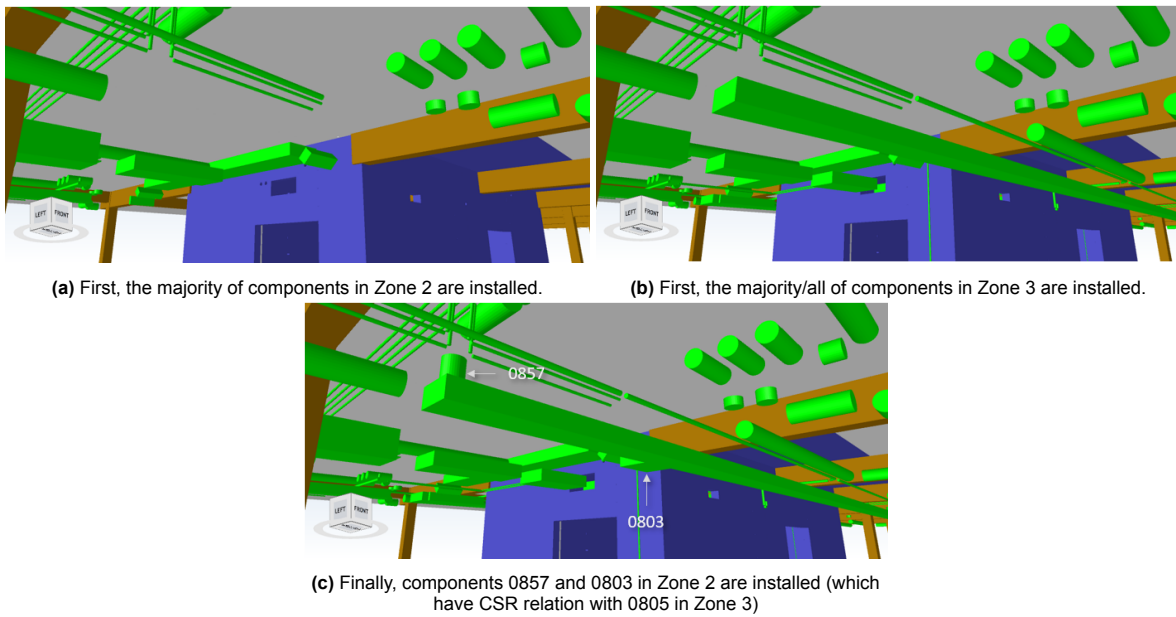


Figure 5.5: Result of MEP sequencing in case study 1, example 1, using option 2 in cluster-splitting & assignment

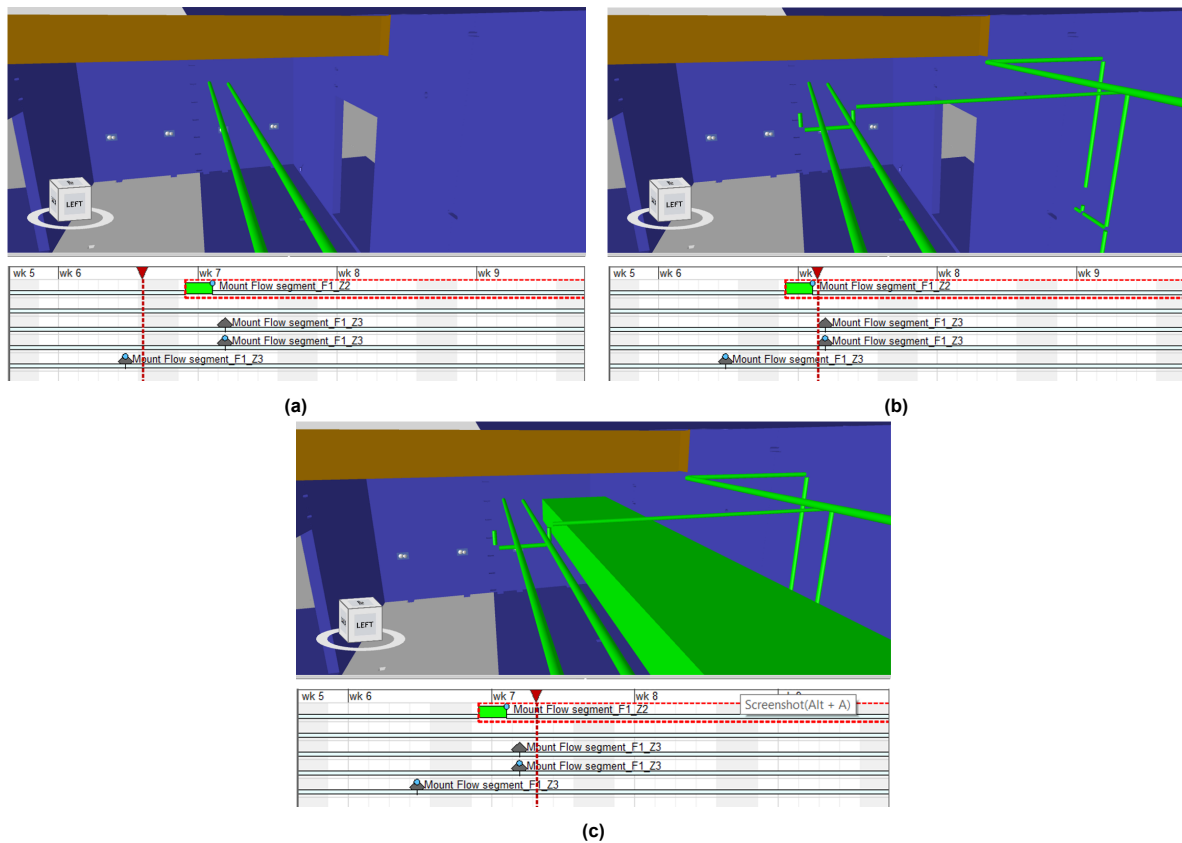


Figure 5.6: Result of MEP sequencing in case study 1, example 2

analysing the original project schedule, three types of activities are considered in the study: (1) columns and beams from the building model (Figure 5.7), combined under one type of activity; (2) roof; and (3) general MEP installations. Roof and MEP installations are not shown in the building model, therefore they are added to the problem, with cost and duration calculated by unit cost or duration per floor area times the floor area covered by one zone. Three types of resources are considered.

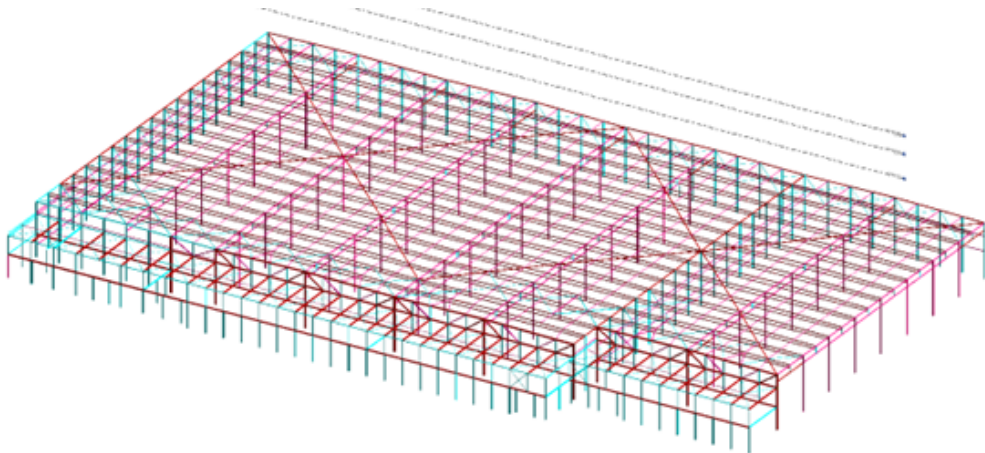


Figure 5.7: Simplified structural model of the distribution centre



(a) Rendered model of the building



(b) During construction (1)



(c) During construction (2)



(d) During construction (3)

Figure 5.8: Case study 2: building model and pictures during construction.

System	Component class	Associated resource
Structural, essential	Column and beam	Mountage team (structural)
Structural, non-essential	Roof	Mountage team (roof)
MEP	General MEP installation	Installation team

Table 5.2: Problem setting of case 2: components and resources.

The following simplifications also are made, to reduce the scale of the case:

- Only one portion (approximately one-third) of superstructure is considered; other parts of the structure or the premises are not included;
- The beams in this structure are in the form of a warren truss system in x and y directions. The struts and bracings are removed from the building file, to reduce the number of building components in the study.

Problem settings related to working zones:

- The minimum grid size: chosen by the planner, considering features of the structure, i.e. each zone should be self-standing and have enough space to safely work within.
- The increment of zone dimensions and zone number assignment: chosen by the planner. The structure features 8 almost identical sections of approximately 23.5 m width, and therefore 23.5 m is chosen as the minimum increment. When assigning the zone number to a structural component, the rules described in Ch3.3 make sure that a horizontal component is not placed before the component directly underneath is placed.

A posteriori method is used in this case study, with a series of different weights of two objectives after normalisation (described in Ch3.4). Multiple Pareto fronts (collection of optimal solutions) are plotted using different problem settings:

1. Different zone dimensions;
2. Different resource availabilities.

The Pareto fronts here and in upcoming sections are plotted by running the optimisation problem multiple times with different weightings of objectives, each time until a (near-)optimal solution satisfying the termination criteria is found or the maximum running time is reached, and ultimately, till an envelope can be plotted. Only the points at the Pareto fronts are plotted.

5.2.2. Results

Figure 5.9 shows Pareto fronts in six scenarios with different working zone dimensions and resource availabilities. Each point at the curves in Figure 5.9 represents one optimal solution which cannot be further optimised, i.e. every solution is associated with a total duration and a total cost value, and a planner cannot reduce the duration without increasing the cost, or reduce the cost without increasing the duration. All the points in the curve will be up to the planner to choose, depending on which objective (cost or duration) they find more important.

More results are included in the next chapter.

5.2.3. Discussion

The cost-duration trade-off

Observations from Figure 5.9:

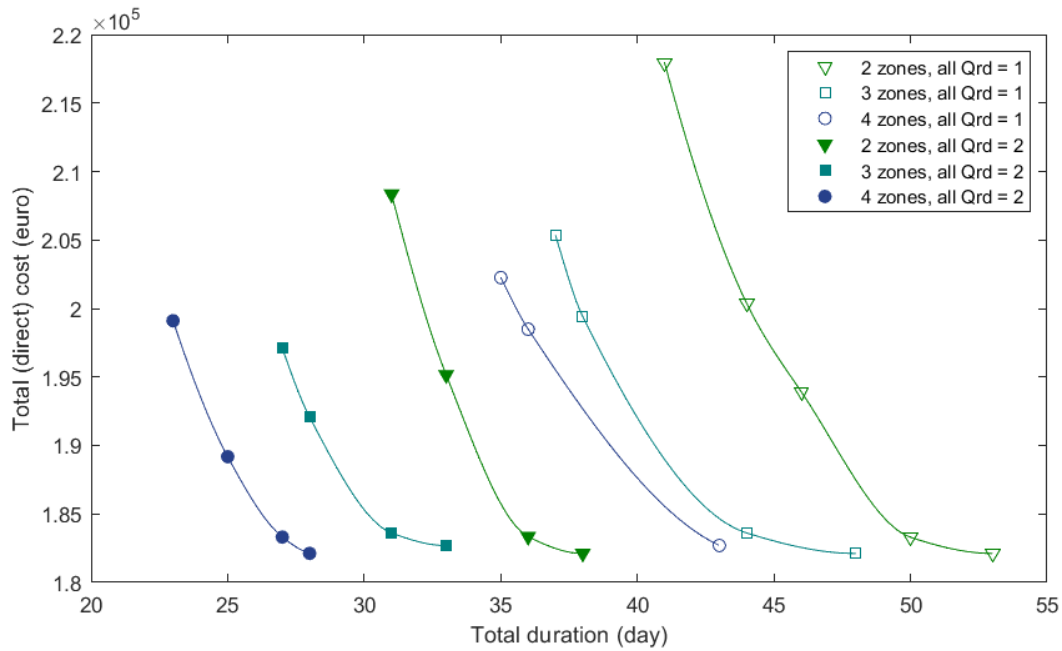


Figure 5.9: Time-cost trade-off, for 2, 3 or 4 working zones, maximum 1 or 2 teams for each working team type. (Q_{rd} : maximum number of resource r available on day d)

1. High direct cost when the duration is small, and the direct cost decreases as the duration increases. That is consistent with the literature.
2. The total duration is constrained by: (1) number of zones; (2) maximum resource availability. For the scenario with 2 zones and $Q_{rd} = 1$, they are both active constraints. In this case, the magnitude of effect when they are relaxed is different: if a horizontal line is drawn where the total cost of 2×10^5 euro and its intersections with the curves are found, it can be observed that the total duration reduces from around 44 days with 2 zones and 1 team (the first curve from the right), to 38 days by increasing the number of zones to 3 (the second curve from the right), or, to 33 days by increasing the number of teams to 2 (the fourth curve from the right).

Number of zones larger than 4 is not shown, due to:

1. Practical limitation on the dimension of zones (which is 8 zones of 23.5m width in this case), and
2. Computational limit of the software tool.

The latter is further investigated in Ch6.1.3.

The effect of zones

To observe the effect of increasing the number of zones, three curves on the right (with hollow markers and all $Q_{rd} = 1$) are compared:

1. The effect of the number of zones on the project duration can be significant. The original project duration is 58 days with only one zone (not shown in the figure); when the number of zones increases to 2, 3, and 4, the project duration reduces to 41, 37 and 35 respectively. That means 40% reduction in total duration, by simply dividing one floor plan into 4 identical zones.
2. The effect of increasing the number of zones weakens as the number becomes larger, e.g. the reduction in project duration is only 2 days when the number of zones increases from 3 to 4.

6

Discussion

This chapter discussed in-depth the results and insights from the case studies, applications and limitations of the tool. First, Ch6.1 discusses: (1) the effect of working zones on the schedule, (2) the effect of resource availability on the schedule and which resources should be targeted, and (3) how to switch between activity modes so that the solution remain optimal. Next, limitations of the developed tool – the computation capacity, and the approximation in the time unit used for activity duration – are analysed, and suggestions are made on how to account for the limitations (Ch6.2). Finally, the applicability to general projects is discussed (Ch6.3).

6.1. Discussion on the results of the case studies

6.1.1. Effect of working zones and resource availability

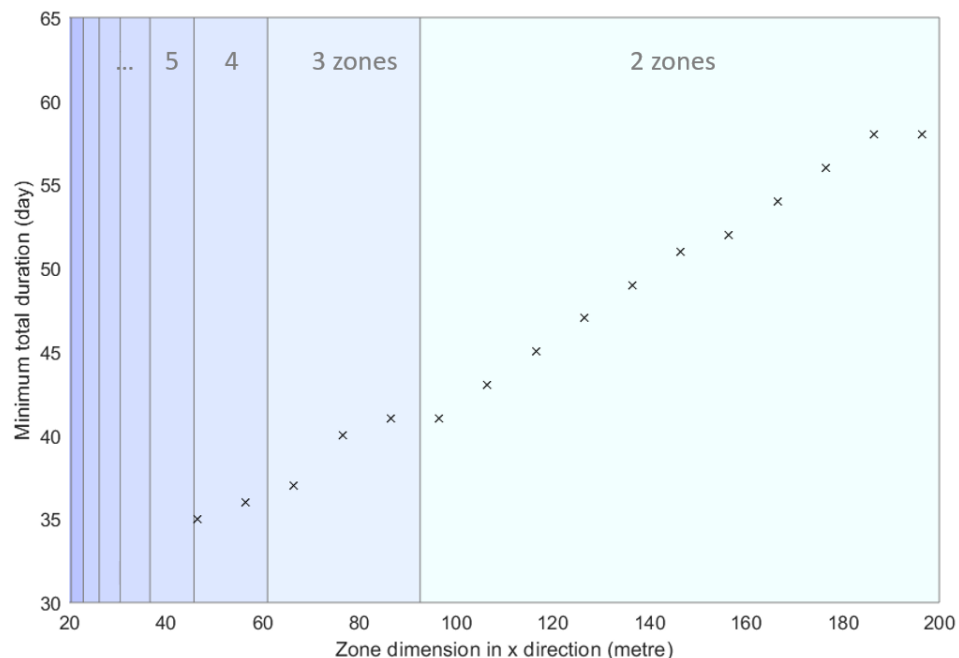


Figure 6.1: The minimum total duration when using different zone dimensions, without considering the cost. The dimension of the building in x direction is approximately 190 m.

In addition to the time-cost trade-off in Figure 5.9, further investigation is conducted on how the number

of zones and the maximum number of resource available affect the project duration.

1. The total duration is constrained by: (1) number of zones; (2) maximum resource availability. For the scenario with small number of zones and resource availability, they are both active constraints. To reduce the total duration, one can choose to lift either one, or both of them simultaneously.
2. The first choice may be to increase the number of zones, as it may not involve extra mobilisation cost and coordination. But the effect of dividing more zone diminishes when the number of zone gets larger (Figure 6.1). In real life the number of zones is also limited by practical and safety considerations (which are listed in Ch7.4.3).
3. When the number of zones approaches its limits, one can then work on the constraints on resources, which can lead to a large reduction in the total duration, as shown in Figure 5.9.

When increasing the maximum resource availability, it is however not always necessary to increase that of all resources. The effect is further investigated in the case of 2 zones in Case study 2: three types of teams (resources) are used in Case study 2, and different maximum number of teams are tested. The result is shown in Figure 6.2.

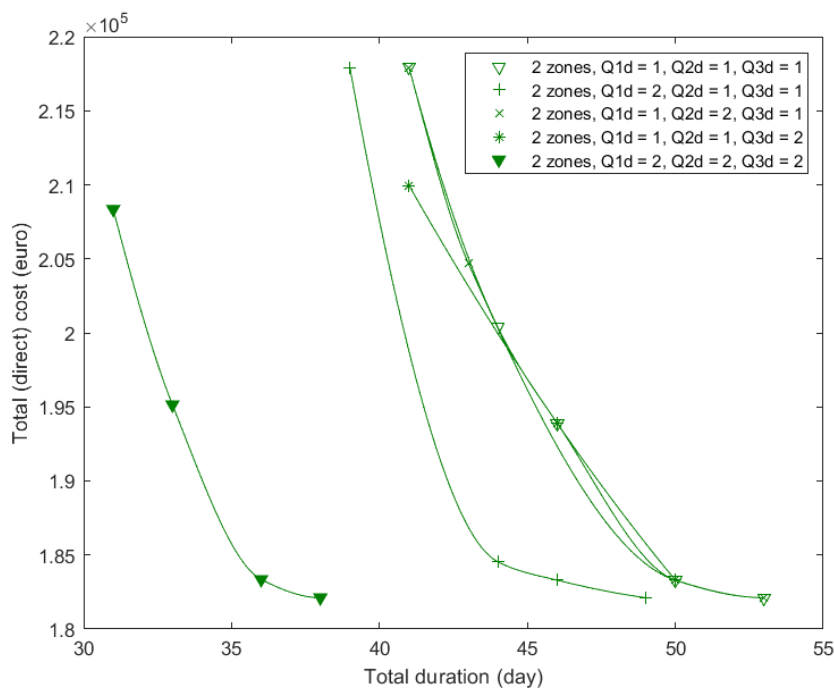


Figure 6.2: Time-cost trade-off, for 2 working zones and different numbers of teams. (Q_{rd} : maximum number of resource r available on day d)

From the figure, merely increasing the number of team 2 or team 3 does not have much influence on the Pareto front; by adding one extra team 1, the Pareto front is shifted slightly to the left; by increasing the number of all three teams, the Pareto front is significantly shifted and the minimum total duration possible is reduced from 41 to 31 days. Therefore, although increasing the availability of all resources may have a large influence on the schedule, it is not always necessary to increase that for all.

This can be explained by examining the first a few activities in Case 2 (Figure 6.3): Act1 to Act3 belong to the same zone, and so do Act4 to Act6. Act1 and Act4 have the longest duration in their zones respectively, and are relatively critical in the first scenario ($Q_{1d} = Q_{2d} = Q_{3d} = 1$), while other activities may have floats. In scenario 1, to add more teams to resource *res 2* or *res 3* will have no influence on the total duration; but if one extra *res 1* is added (i.e. $Q_{1d} = 2$, in the second plot), Act1 and Act4 can be executed in parallel and the end date is shifted.



Figure 6.3: Gantt chart of a part of case 2 schedule, with different numbers of teams (Q_{rd}).

From the case studies, it is also found that the shape of the zones does not affect the total duration, as long as the zones are practically feasible to construct and (almost) identical, e.g. the time-cost trade-off of 1-by-4 and 2-by-2 zone division is found identical in Case 2, both with 35 days as the minimum duration possible and the same Pareto front.

6.1.2. Selection of activity modes

As explained in Ch3.1, activity execution mode refers to how the same activity can be executed in different manners which lead to different productivity and cost, e.g. by “one crane + one operator + four worker” with lower cost and longer duration, or “one crane + one operator + six workers” with higher cost and shorter duration. The modes available to choose from are provided in the database. For all activities, mode 1 is set to be the one with lower cost and longer duration.

Figure 6.4 shows how an increasing number of mode 2 is selected, as the weighting of duration increases in posteriori method. In particular, some activities seem to have a tendency to be chosen first, such as activity no. 1, 4, 7 and 12.

This is analysed using hypothetical scenarios simplified from Case 1 (Figure 6.5), assuming identical activities in two zones; each of six scenarios represents the (near-)optimal solution under one objective weighting set (w_1 and w_2). As the weighting of total duration (w_2) increases gradually from 0.01 to 0.41, Act6 and Act1 are first compressed, as it is most economical to do so (extra 600 euro for 1 day, and then 2000 euro for 2 days); if w_2 further increases to 0.45, Act4 is compressed from 8 to 6 days, which leads

		Result 1	Result 2	Result 3	Result 4	Result 5	Result 6
Weight of objective 1 (total direct cost)		0.99	0.65	0.59	0.55	0.45	0.08
Weight of objective 2 (total duration)		0.01	0.35	0.41	0.45	0.55	0.92
Activity		Activity mode					
1	Mount column & beam_F0_Z1	1	1	2	2	2	2
2	Mount roof_F0_Z1	1	1	1	1	1	1
3	Install installation_F0_Z1	1	1	1	1	2	1
4	Mount column & beam_F0_Z2	1	1	1	2	2	2
5	Mount roof_F0_Z2	1	1	1	1	1	1
6	Install installation_F0_Z2	1	1	1	1	1	1
7	Mount column & beam_F0_Z3	1	1	1	2	2	2
8	Mount roof_F0_Z3	1	1	1	1	1	1
9	Install installation_F0_Z3	1	1	1	1	1	2
10	Mount column & beam_F0_Z4	1	1	1	1	1	1
11	Mount roof_F0_Z4	1	1	1	1	2	2
12	Install installation_F0_Z4	1	2	2	2	2	2
Total duration (days)		45	43	40	36	35	35
Total direct cost (e3 euro)		182.1	182.7	188.6	198.5	202.3	202.3

Figure 6.4: Several near-optimal results and the activity modes selected.

to reduction of 1 day in the total duration, as the critical path shifts and Act3 becomes the constraining one (extra 2000 euro for 1 day); if w_2 increases further, Act3 is then compressed (extra 600 euro for 1 day).

The analysis leads to the following two insights:

1. When adjusting activity durations, there is no one or several fixed critical paths;
2. As the weighting of total duration increases, to keep the optimality of the solution, activities which:
 - are in the current critical paths, and
 - have the lowest ratio of cost increase over project duration reduction
are selected sequentially.

In other words, some activities require a lower cost increase to achieve relatively high time saving. These are more “efficient” and “easier” changes, and tend to be selected first. As the weighting of duration increases, it becomes harder and harder to compress the project duration, and by compressing a small amount of duration the cost increase will then be high.

Therefore, each time the project duration is to be shortened, the planner is advised to review the critical path, and start with the activities on the path which are “easier” to change, i.e. with lower ratio of cost difference and duration difference between modes.

Another observation is that, if there are multiple disciplines in the same zone, more aligned working speed (i.e. similar duration to complete the work in one zone) would lead to less idle time and more compact schedule, which coincides with lean concept.

6.1.3. Computational capabilities

Case 1 is run in PC with Intel Core i7, 2.80 GHz processor, 64-bit operating system, 8GB RAM and 512 GB SSD; Case 2 is run on the same PC as well as DelftBlue (DHPC) (2022), the TU Delft supercomputer, using up to 4 CPUs and 20 GB memory limit per CPU. Mixed-integer linear programming is selected due to its efficiency in large problems, e.g. with thousands of variables which may potentially take much longer for nonlinear problems. The function *intlinprog* in MATLAB is used to solve the optimisation problem.

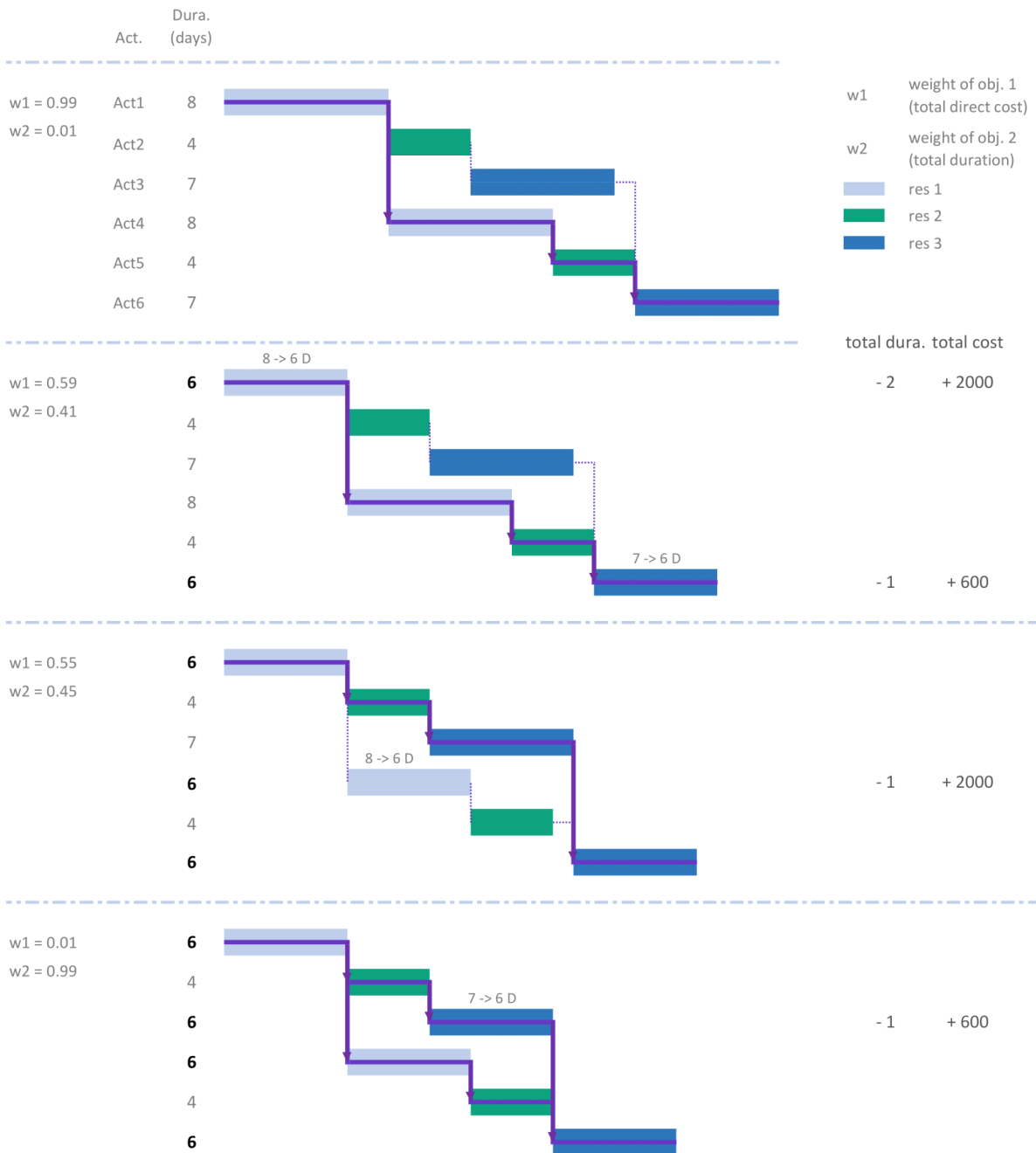


Figure 6.5: Hypothetical schedule using different weightings of objectives, for 4 working zones and maximum 1 team per resource.

Matrix size in MATLAB is constrained by the computer storage space available. In the formulation, the largest matrix is the linear inequality constraint matrix A , which is of size $nvars * nvars$ approximately. Sparse matrices is used to reduce the size of A so that the maximum $nvars$ the programme can process increases from around 28k to 100k (e.g. when number of activities $A = 60$, number of resource types $R = 2$, estimated maximum total duration $D = 270$) with 7.8 GB RAM. However, as $nvars$ increases, the performance of the solver – computational time and quality of the solution – becomes another limiting factor.

The factors that affect the performance and capacity of the tool are found to be:

1. Number of variables and constraints

2. Termination criteria of the solver, such as the maximum running time, maximum gap tolerance

No.	Num. zones	Num. activities	Num. time unit	Time unit (hrs)	Num. resources	Num. variables	Time to find solution(s) satisfying termination criteria (i.e. near-optimal solution)	Find feasible solution(s) within 3600s?	Find solution(s) satisfying termination criteria within 3600s?
1	3	10	48	8	3	4640	900	Y	Y
2	3	10	48	8	3	4640	1800	Y	N
3	4	13	45	8	3	5681	600	Y	Y
4	4	13	45	8	3	5681	1800	N	N
5	4	13	45	8	3	5681	-	N	N
6	5	16	45	8	3	6704	-	N	N
7	4	13	86	4	3	10478	5700	Y	N
8	4	13	86	4	3	10478	-	N	N

Figure 6.6: Tests on the computation capacity of the tool.

The capability of the software tool in dealing with large number of variables is tested, using the data from case study 2 (Figure 6.6). The efficiency of MILP is confirmed in this study with less than 6,000 variables (e.g. $A = 10$, $R = 2$, $D = 100$; or, $A = 30$, $R = 1$, $D = 70$) and similar number of constraints. However, as the number of variables rises above this level, it is considerably harder for the solver to find near-optimal solutions, or a feasible solution at all.

Suggestions to accommodate computational limits

The number of variables do not directly relate to the size (e.g. floor area) of the building. The planner is suggested to estimate the number of activities, hours and variables before running the optimisation problem. If the problem is too large, there are the following options:

1. Change the time unit: use a finer granularity, e.g. 2-, 4- or 8-hour, for projects with less variables; otherwise, use 8- or 16-hour.
2. Conduct optimisation on part of the project, before extending the result or insights to the whole project. This can be done by:
 - (a) Simplify the consideration, e.g. reduce the number of classes by mapping structural column, beam and slab into one class.
 - (b) Analysing part of the project, e.g. half of the whole floor area, or some floors.

The effect of method 1 (changing the time unit) is investigated in the next section; method 2(a) is explained in Ch6.3 for projects of large size.

6.2. Sensitivity analysis

6.2.1. Activity duration

When formulating the problem, one important simplification is the conversion of duration of activity clusters from hours to days: if the duration of activity “mount all columns in Floor 1, Zone 1” is 10 hours, it is divided by 8 (hours per day) and rounded to the nearest integer which is larger, i.e. 2 days. The reason for this approximation is the computational efficiency; otherwise, the number of variables will be multiplied by 8 if “hour” is the time unit, or, the problem may be formulated as a nonlinear one. Nevertheless, the rounding can impact the accuracy of the scheduling result.

Case study 2 is again used to investigate the effect of the approximation of activity duration on the total

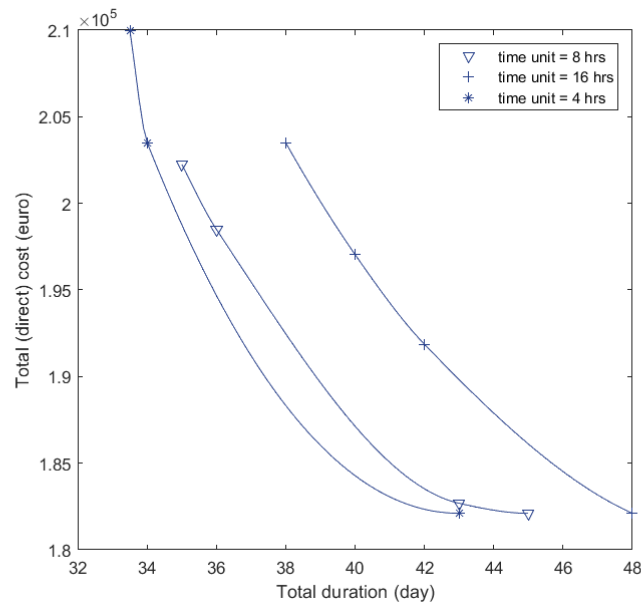


Figure 6.7: Effect of time units: time-cost trade-off (4 working zones, maximum resources of 1 team in case study 2).

duration (Figure 6.7). A range of time units are taken – 4-hour (half-day), 8-hour (1 day) and 16-hour (2 days) – and the total durations yielded are compared: the Pareto front moves to the left as the time unit becomes more precise; the result is found very inaccurate when time unit is 16-hour, and the accuracy improves using 8-hour and 4-hour.

6.3. Extension to general projects

Case 1 and 2 are selected as they represent two very different types of buildings: one has a homogeneous or repetitive form of geometry, typical for large structures spread over large floor area; the other has spatially complicated structural and non-structural systems.

In practice, a building can take a different form or size: a building may not be a well-fit to either of those types, which can still be processed by the same software tool without modifying the original script; or, a building can consist of different portions with different characteristics, e.g. a distribution centre may have both a large hall and offices with compacted components, and a planner can choose to run optimisation for those building portions separately and combine the schedules. This “divide and conquer” strategy is explained in Ch6.3.1.

6.3.1. Schedule in portions and combine

This applied to projects with distinctively separated areas, or large projects for which large computational times is expected.

For buildings which have portions of floor areas which are distinctively separated by their characteristics, e.g. office area and storage hall, it may be preferable in practice to include them in separate zones. That can be done by setting the zone dimensions in the software tool so that the zone boundary overlaps with the boundary of the portions; however, currently the software tool considers only identical rectangular zones and the user cannot freely draw zones of different dimensions.

To deal with such building types, the planner is suggested to run optimisation for those building portions separately, before aligning the progress of different portions.

Suggested steps on how to schedule in portions and combine

If a building is to be considered in several portions, schedules of individual portions can be concatenated to form the schedule of the whole project. Using the software tool, the proposed steps are described below.

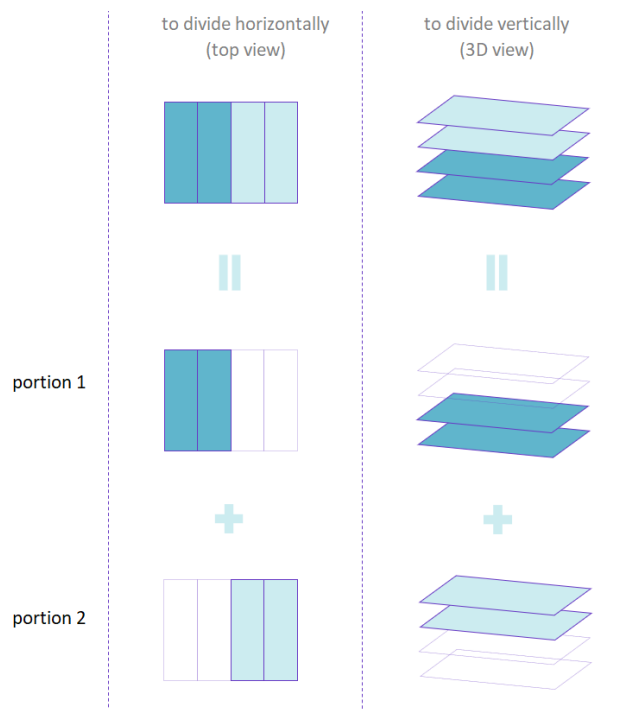


Figure 6.8: Extension to general projects – how to divide the project model into portions.

Step 1

Divide the building model into the portions desired (e.g. horizontally or vertically, as illustrated in Figure 6.8), and save into several files by using BIM software applications; alternatively, the user can filter for the building elements by coordinates in the original scripts of the tool (not through the UI).

Step 2

Run optimisation for each building portion using the tool. Observe the produced schedules and check: (1) durations of activities by each discipline, and the duration to complete each zone; and (2) suitable and (near-)optimal zone dimensions.

Step 3 (optional)

To minimise the idle time and reduce the total duration, it is advised to make the working speed of different disciplines as similar as possible. If that is desirable, the user can adjust the problem settings (range of zone dimensions, maximum resource availability, weightings of objectives) in the user interface and run the optimisation several times to produce more desirable schedules.

Step 4

Identify the “conjunction” dates of schedules. When the schedule of portion (n+1) can be connected to that of portion (n) is case-dependent. Two example scenarios are given in Figure 6.9, assuming identical activities and activity durations in both portions: in scenario 1 (Figure 6.9a), the activities with the longest duration a are most critical, and the earliest starting date of the second schedule is $(a + a)$; in scenario 2 (Figure 6.9b), the most critical activity is the ones with duration c , and the earliest starting date of the second schedule calculated by $((a + b + c) + c - (a + b))$.

Starting from the second portion, select the starting date to be the identified earliest starting date when saving the output schedule into Excel sheet.

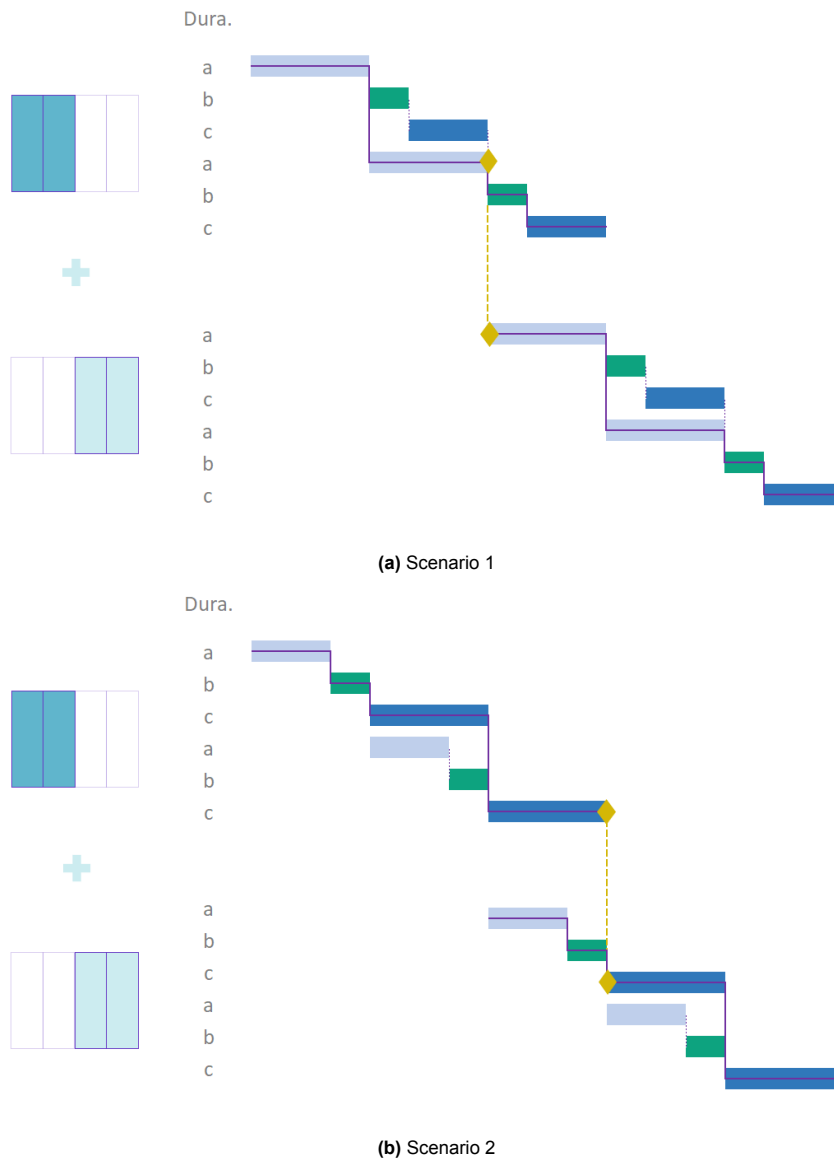


Figure 6.9: Extension to general projects – how to combine the schedules of the portions divided.

Step 5

Copy and paste all the schedules in one Excel sheet. That will be the schedule for the whole building.

Schedule updates during project execution

Execution of construction projects has large uncertainty, and one rigid schedule often does not work in practice. Ideally, a tool should have the ability to change and re-plan a schedule as the project goes.

Record of planned and actual work progress or direct modification of activity starting and finishing date via UI is not within the scope of this work, and such functionality can be one direction for future developments. But, it is still possible to update a schedule in execution phase using the current tool, in a more manual way: the planner can delete from the BIM model the components that are already completed as of date, then feed the new model into the tool and re-do the planning, so that the new schedule contains only the components and activities to be planned next.

7

Conclusion and recommendation

This chapter concludes the whole report. First, the research questions are re-visited and the answers provided. The study is then reviewed critically for both its contributions and limitations. Finally, recommendations are given, to the industry and future research.

7.1. Answers to the research questions

Answers to the main research question and sub-questions are already covered in previous chapters. In this section, more concise answers are given.

Sub-question 1: How to sequence the structural and MEP building components in one integrated schedule?

Individual building components can be grouped as activity clusters when scheduling. For structural systems, the precedence between activity clusters are identified using a set of sequencing principles based on component class, floor level, zone number, and if the component is in an essential structural class. Four scenarios where there are precedence relations are identified.

For MEP systems, two-level preferences on constructability are added, on component spatial relations and size. To resolve the subsequent conflicts in precedence at component- and cluster-level, those clusters can be split by using the method proposed in Ch3.3.

Sub-question 2: How to use the “zoning” concept in the schedule optimisation problem?

2-1: How does it affect the time-cost trade-off in a construction project?

2-2: Can it help produce more optimal schedules?

In this work, zones are created by overlaying rectangular grids over the floor plan, to facilitate parallel working and reduce idleness of resources and space. Through case studies and theoretical analysis, its effect on project total direct cost and duration is investigated. It is found that increasing the number of zones has large potential in reducing project total duration with the same total direct cost, represented by shifts of Pareto fronts to the left in time-cost plots, e.g. by up to 40% reduction in duration in one of the case studies, though its effect decreases when the number of zones is large.

Main question: How to automatically generate and optimise a BIM-based, component-level schedule for building structural and MEP systems, considering parallel working zones?

To answer the question, a conceptual framework is first developed to tackle three key aspects the question entails: how to sequence construction activities, how to sequence activities as clusters, and how to formulate a schedule optimisation problem.

A software tool is developed to implement the conceptual framework, with BIM models as the main input and optimised schedule as output; the latter can be coupled with individual building components eventually. The conceptual framework and the tool is then verified and validated by case studies. The results of case studies and further analysis of the results demonstrate the large potential of zones in reducing project duration, the effect of the amount of resource available, and suggest to the industry strategies on the compression of project schedule.

7.2. Contribution of this work

To scientific research

1. A set of sequencing principles for MEP components, including two-level preference checking to account for (1) spatial proximity with adjacent MEP and structural components, and (2) component sizing.
2. A method for work packaging of MEP components, which addresses the conflict in precedence relations at cluster- and component-level and produces the minimal number of subclusters, or, alternatively, more subclusters which follow preferred construction sequence.
3. Insights on: (1) the effects of working zones on construction project scheduling; (2) how to choose between activity modes to adjust the total duration or cost while maintaining optimality.

To the industry

1. A open-source software tool, to conduct BIM-based automatic schedule optimisation and obtain insights into time-cost trade-off in specific projects. A series of instructions is provided alongside the tool, on how to adjust the settings for a general project, customise sequencing rules, deal with the limitations and maximise its value.
2. Observations and recommendations for future project practices, on:
 - (a) effect of working zone division and resource availability on project duration;
 - (b) how to switch between activity execution modes while still obtaining (near-)optimal schedules.

7.3. Limitations

Limitation from the key assumptions and deviation from the practice

1. **Derivation of construction sequence:** In this work, the construction sequence is deduced from the component class, level and spatial relationship; however, as put forward by H. Liu et al. (2015), the derivation of supporting relationships and construction sequences cannot be fully rely on spatial relationships such as connection relationship. In their work, a simplified structural analytical model containing supporting information between components was extracted from Revit, before the construction sequence is determined. For a more realistic sequencing logic, a similar approach – extracting supporting information automatically from the building model, or, allowing more customisable sequencing rules to account for more project-specific rules – is recommended to enrich the current work.
2. **Project duration and cost items:** The project duration and cost essentially consists of that of construction component mounting activities, assuming all such activities can be seamlessly

executed without idleness. Real-life projects are usually much more complicated:

- Many activities are not directly pinned to a building components, such as temporary fencing, material unloading, or machine mobilisation.
 - The cost incurred when a crew is in idleness is not accounted for, e.g. if crew A works on predecessor activities of crew B, and they need 6 and 4 days respectively to complete work in one zone, B may have an idle time of 2 days each time before they move on to the next zone; in this project, it is assumed that there is no cost associated with B in those 2 days. If the cost during crew idle time is considered, the project total cost may increase (Eiris Pereira and Flood, 2017). This also means that the tool may have limited value for geometrically linear infrastructural projects, such as pavement or railways where the idleness of resources and unproductive costs are important considerations for planners (S.-S. Liu et al., 2021).
 - Other factors such as logistics also play a role, e.g. the supply of material to site and material transport within site can influence project planning, and they may even be the limiting factor in urban construction projects with congested site area or travel restrictions.
3. **Optimisation on partial project:** the problem formulation focuses on only a part of a project, while excluding many other activities on- and off-site. The result of such “sub-optimisation” does not necessarily lead to an optimised overall project schedule.
 4. **Extra practical considerations and limitations:** An example is the starting condition of “interior” activity. Practically, an interior activity may start after its predecessors not in that zone, but in the whole building, is finished, i.e. after the whole building is water- and wind-tight. To fast track the long-duration interior activities in a large building, it may be considered to divide the building into a few large zones where the water- and wind-tight requirement can be met within the zone, and then, if so preferred, divide those zones into smaller ones for other activities for further time-saving.

Other limitations

1. **Better database and mapping:** Currently, one value of duration is associated with each component based on their class, class, material and floor level. In reality there are more factors, such as the complexity of the project, or the complexity of the connections between components.
2. **Activity execution modes:** Now there is important theoretical assumption based on the insights from experts, which should be further investigated in practice.
3. **Computational performance:** Linear programming is selected due to its efficiency in large problems, e.g. with thousands of variables which may take much longer for other types of problem formulations. That is found true in this study when the number of variables is below around 6,000 and the number of constraints of a similar value. However, as the number of variables rises above this level, it is considerably harder for the solver to find a near-optimal solutions, or a feasible solution at all.

7.4. Recommendations to industrial practice

7.4.1. Recommendations on schedule optimisation

Recommendation 1: Use of zones and resource

To reduce the total duration, one can choose to lift the constraint on either (1) number of zones, or (2) maximum resource availability, or both simultaneously. First choice may be to increase the number of zones; when the number of zones approaches its limits, one can then work on the constraints on resources.

When increasing the maximum resource availability, it is not always necessary to increase that of all resources, but to identify the activities which affects the total duration the most; in case of nearly-identical activities for all zones, those activities are most likely the ones with the longest durations

(which causes idle time in other resources).

Recommendation 2: How to shorten the project duration, by changing activity execution mode

Step 1: Identify the current critical path.

Step 2: Change the mode of activities on the critical path which have the lowest ratio of cost increase over project duration reduction.

7.4.2. Use of the software tool

Users of the tool are recommended to carefully decide the set-ups of the optimisation problem described in Ch4.2.2, due to large diversity in construction projects; careful inspections of the schedule produced and several try-outs are recommended.

The accuracy and practicality of the schedule depends largely on the accuracy of the data used. It is suggested to establish a complete and accurate database, conduct analysis of historical data and identify the factors influencing the duration and cost. Furthermore, more duration and cost items, such as time for resource mobilisation and indirect cost, can be added to the database and the problem formulation; other performance indicators or objectives, such as safety or environmental impact, can be integrated into the formulation in future developments.

7.4.3. Design of working zones

The working zones are currently limited to rectangular and (almost) identical ones within one project. Its design is largely project-dependent and thus concrete numbers will not be given here. The following general considerations are however proposed regarding its usage:

1. Dividing zones is a good way to reduce the total duration without significantly increasing the resource availability or the cost. But it also has limit in how much time it can save, as:
 - its effect diminishes when the number of zones is large;
 - the size of zone cannot be too small, for safety and constructability reasons.
2. The following factors are suggested to be taken into consideration when deciding zone dimensions:
 - characteristics of the structure: an example is Case study 2, where the superstructure has 8 identical sections of 23.5 m span each;
 - practical considerations on safety and constructability, e.g. each team should have sufficient space within one zone to mobilise the equipment or carry out the works safely;
 - implications of zones, mentioned by Jabbari et al. (2020): hand-offs between trades, setups and mobilisation of resources between zones.
3. Shape of zones does not matter time-wise. But, if more than one zone dimension can be chosen (which is not the case within this study), to maximise the benefit, it is recommended to:
 - ensure that the activities by different disciplines (resources) within one zone have similar durations;
 - have similar durations to complete all activities in all zones.

7.5. Recommendations for future research

Three directions for future research are suggested:

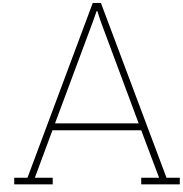
1. **Sequencing considering supporting & connections:** e.g. by extracting supporting information automatically from structural analytical models.

2. **More flexibility with the design of zones:** In this work, the zones are made by overlaying rectangular grids over the building floor plan, with one corner of the floor plan as the pivot point. Future extensions can be on:
 - (a) More flexibility in shapes, e.g. irregular ones;
 - (b) Treating zone dimensions as variables, e.g. by bi-level optimisation, instead of listing the dimensions brute force and treating them as data in each run;
 - (c) Multi-level zones, e.g. further divide one zone into multiple sub-zones for some activities, to achieve more uniform rate of progress.
3. **Integration with logistics & site planning**

References

- Alavipour, S. R., & Arditi, D. (2019). Time-cost tradeoff analysis with minimized project financing cost. *Automation in Construction*, 98, 110–121.
- Al-Jabery, K., Obafemi-Ajayi, T., Olbricht, G., & Wunsch, D. (2019). *Computational learning approaches to data analytics in biomedical applications*. Academic Press.
- Andersson, N. (n.d.). The single and multi project approach to planning and scheduling—contractors vs. subcontractors [CBS]. In *Managing the Construction of Buildings*.
- Baron-Cohen, S. (2012). *The science of evil: On empathy and the origins of cruelty*. Basic books.
- Bashir, H., Ojiako, U., Marshall, A., Chipulu, M., & Yousif, A. A. (2022). The analysis of information flow interdependencies within projects. *Production Planning & Control*, 33(1), 20–36.
- Binninger, M., Dlouhy, J., Oprach, S., & Haghsheno, S. (2016). Methods for production leveling—transfer from lean production to lean construction. *Proceedings of the 24th Annual Conference of the International Group for Lean Construction (IGLC 24), Boston, MA, USA*, 20–22.
- (DHPC), D. H. P. C. C. (2022). DelftBlue Supercomputer (Phase 1).
- Echeverry, D., Ibbs, C. W., & Kim, S. (1991). Sequencing knowledge for construction scheduling. *Journal of Construction Engineering and Management*, 117(1), 118–130.
- Eiris Pereira, R., & Flood, I. (2017). Impact of linear correlation on construction project performance using stochastic linear scheduling. *Visualization in Engineering*, 5(1), 1–12.
- Elbeltagi, E., Ammar, M., Sanad, H., & Kassab, M. (2016). Overall multiobjective optimization of construction projects scheduling using particle swarm. *Engineering, Construction and Architectural Management*.
- Feng, C.-W., Liu, L., & Burns, S. A. (1997). Using genetic algorithms to solve construction time-cost trade-off problems. *Journal of computing in civil engineering*, 11(3), 184–189.
- Floudas, C. A., & Lin, X. (2005). Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications. *Annals of Operations Research*, 139(1), 131–162.
- Gardner, G. R. (2006). Effective construction work packages. *AACE International Transactions*, S131.
- Ibrahim, Y., Lukins, T. C., Zhang, X., Trucco, E., & Kaka, A. (2009). Towards automated progress assessment of workpackage components in construction projects using computer vision. *Advanced Engineering Informatics*, 23(1), 93–103.
- Isaac, S., Curreli, M., & Stoliar, Y. (2017). Work packaging with bim. *Automation in Construction*, 83, 121–133.
- Isaac, S., & Shimanovich, M. (2021). Automated scheduling and control of mechanical and electrical works with bim. *Automation in Construction*, 124, 103600.
- Jabbari, A., Tommelein, I. D., & Kaminsky, P. M. (2020). Workload leveling based on work space zoning for takt planning. *Automation in Construction*, 118, 103223.
- Khanzode, A. (2010). An integrated, virtual design and construction and lean (ivl) method for coordination of mep. *Unpublished Technical Report*, 187.
- Kim, H., Anderson, K., Lee, S., & Hildreth, J. (2013). Generating construction schedules through automatic data extraction using open bim (building information modeling) technology. *Automation in Construction*, 35, 285–295.
- Korman, T. M., Fischer, M. A., & Tatum, C. (2003). Knowledge and reasoning for mep coordination. *Journal of Construction Engineering and Management*, 129(6), 627–634.
- Li, X., Wu, C., Xue, F., Yang, Z., Lou, J., & Lu, W. (2022). Ontology-based mapping approach for automatic work packaging in modular construction. *Automation in Construction*, 134, 104083.
- Liu, H., Al-Hussein, M., & Lu, M. (2015). Bim-based integrated approach for detailed construction scheduling under resource constraints. *Automation in Construction*, 53, 29–43.
- Liu, S.-S., Budiwirawan, A., & Arifin, M. F. A. (2021). Non-sequential linear construction project scheduling model for minimizing idle equipment using constraint programming (cp). *Mathematics*, 9(19), 2492.

- Lockley, S., Benghi, C., & Černý, M. (2017). Xbim.Essentials: A library for interoperable building information applications. *The Journal of Open Source Software*, 2(20), 473. <https://doi.org/10.21105/joss.00473>
- Menesi, W., Abdel-Monem, M., Hegazy, T., & Abuwarda, Z. (2015). Multi-objective schedule optimization using constraint programming.
- Moon, H., Kim, H., Kamat, V. R., & Kang, L. (2015). Bim-based construction scheduling method using optimization theory for reducing activity overlaps. *Journal of Computing in Civil Engineering*, 29(3), 04014048.
- Nguyen, D.-T., Chou, J.-S., & Tran, D.-H. (2022). Integrating a novel multiple-objective fbi with bim to determine tradeoff among resources in project scheduling. *Knowledge-Based Systems*, 235, 107640.
- Pezoulas, V., Exarchos, T., & Fotiadis, D. I. (2020). *Medical data sharing, harmonization and analytics*. Academic Press.
- PMI, A. (2004). Guide to the project management body of knowledge. *Project Management Institute*, 130.
- Raz, T., & Globerson, S. (1998). Effective sizing and content definition of work packages. *Project Management Journal*, 29(4), 17–23.
- Sacks, R., Treckmann, M., & Rozenfeld, O. (2009). Visualization of work flow to support lean construction. *Journal of construction engineering and management*, 135(12), 1307–1315.
- Sigalov, K., & König, M. (2017). Recognition of process patterns for bim-based construction schedules. *Advanced Engineering Informatics*, 33, 456–472.
- Singh, J., Cheng, J. C., & Anumba, C. J. (2021). Bim-based approach for automatic pipe systems installation coordination and schedule optimization. *Journal of Construction Engineering and Management*, 147(11), 04021143.
- Tang, Y., Liu, R., Wang, F., Sun, Q., & Kandil, A. A. (2018). Scheduling optimization of linear schedule with constraint programming. *Computer-Aided Civil and Infrastructure Engineering*, 33(2), 124–151.
- Tommelein, I. D. (2017). Collaborative takt time planning of non-repetitive work. *Annual Conference of the International Group for Lean Construction*.
- Urbanucci, L. (2018). Limits and potentials of mixed integer linear programming methods for optimization of polygeneration energy systems. *Energy Procedia*, 148, 1199–1205.
- Wang, Y., Yuan, Z., & Sun, C. (2018). Research on assembly sequence planning and optimization of precast concrete buildings. *Journal of Civil Engineering and Management*, 24(2), 106–115.
- Wang, Z., & Azar, E. R. (2019). Bim-based draft schedule generation in reinforced concrete-framed buildings. *Construction Innovation*.
- Yassine, A., & Braha, D. (2003). Complex concurrent engineering and the design structure matrix method. *Concurrent Engineering*, 11(3), 165–176.



Matrix transformation in Step 5 of cluster splitting

Notation and the terms (e.g. “red” and “green” numbers) to be used are introduced in Table A.1.

s_{orig}	Index of the cluster to split, e.g. in Figure A.2a, $s_{orig} = 3$ if cluster c is to be split
“Red” numbers	When the numbers at location (i, j) and (j, i) are both nontrivial, these numbers are marked “red”. They imply a conflict between cluster i and j .
“Green” numbers	When one of the numbers at location (i, j) and (j, i) is nontrivial and the other one is not, the nontrivial number is marked “green”. It implies a precedence relation between cluster i and j , but not a conflict.
$R(i), i = 1, 2, \dots, I$	List of the non- s_{orig} indices of red numbers, e.g. in Figure A.2a, if cluster c is to be split, $R(i) = 1, 10, I = 2$
$G(j), j = 1, 2, \dots, J$	List of the non- s_{orig} indices of green numbers, e.g. in Figure A.2a, if cluster c is to be split, $G(j) = 2, 11, J = 2$

Table A.1: Notation used in the matrix transformation in Step 5

	a	b	c	d	e	f	g
a		10	9			20	
b				7			
c	2	25			4		
d		5			6		3
e			6	8			
f			15				30
g				4			

Figure A.1: The original matrix $M_{cluster}$, before splitting. The locations where no value is shown has the value of zero.

Change in indices of red numbers

Row and column indices before the transformation: $(s_{orig}, R(i))$ and $(R(i), s_{orig})$.

After the transformation:

	a	b	c	d1	d2	d3	d4	d5	d6	e	f	g
a		10	9								20	
b				7								
c	2	25								4		
d1												
d2		5										
d3										6		
d4												
d5											3	
d6												
e			6			8						
f			15									30
g									4			

(a) Sub-step 5-1: split cluster d

	a	b	c1	c2	c3	c4	c5	c6	d1	d2	d3	d4	d5	d6	e	f	g
a		10			9											20	
b									7								
c1			25														
c2																	
c3																	
c4	2																
c5															4		
c6																	
d1																	
d2		5															
d3															6		
d4																	
d5																	3
d6																	
e								6			8						
f				15													30
g															4		

(b) Sub-step 5-2: split cluster c

Figure A.2: Splitting of matrix $M_{cluster}$, in the proposed manner

If $R(i) < s_{orig}$:

- Area on the top: $(R(i), s_{orig} + 2i - 2 + J)$
- Area on the left: $(s_{orig} + 2i - 1 + J, R(i))$

If $R(i) > s_{orig}$:

- Area on the bottom: $(R(i) + 2I - 1 + J, s_{orig} + 2i - 1 + J)$
- Area on the right: $(s_{orig} + 2i - 2 + J, R(i) + 2I - 1 + J)$

Change in indices of green numbers

Row and column indices before the transformation: $(s_{orig}, G(j))$ or $(G(j), s_{orig})$.

After the transformation:

If $G(j) < s_{orig}$:

- Area on the top: $(G(j), s_{orig} - 1 + j)$
- Area on the left: $(s_{orig} - 1 + j, G(j))$

If $G(j) > s_{orig}$:

- Area on the bottom: $(G(j) + 2I - 1 + J, s_{orig} - 1 + j)$
- Area on the right: $(s_{orig} - 1 + j, G(j) + 2I - 1 + J)$

Example: how the numbers in the third column and row c in Figure A.1, are moved to their locations in Figure A.2b.

At the beginning of Step 5-2 (Figure A.2a),

$s_{orig} = 3; g = 2; R(i) = 1, 10$ when $i = 1, 2; I = 2; G(j) = 2, 11$ when $j = 1, 2; J = 2$.

Example 1

When $i = 1, R(1) = 1 < s_{orig}$. Hence, number 9 at (1,3) and number 2 at (3,1) in the old matrix, will be moved to the area on the top and left in the new matrix respectively:

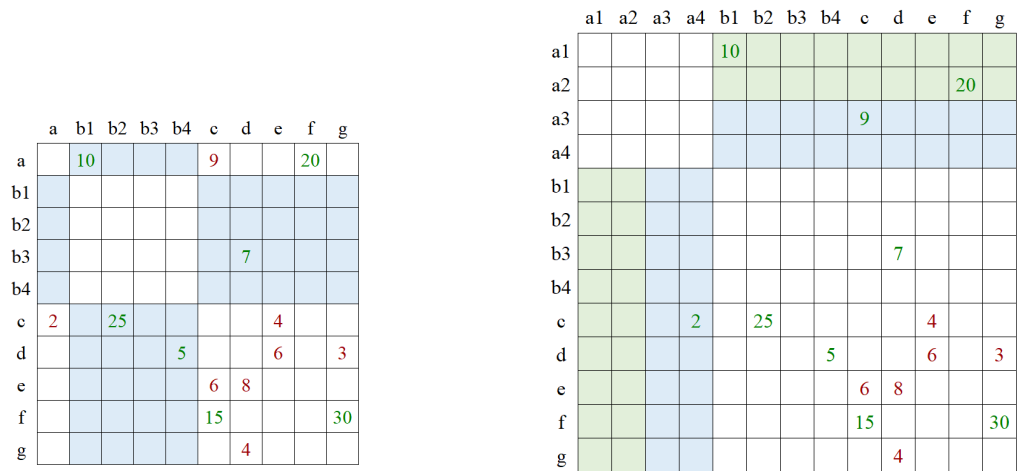
- Number 9 at (1,3) is to be moved to: $(R(i), s_{orig} + 2i - 2 + J)$, which is (1, 5) in Figure A.2b
- Number 2 at (3,1) is to be moved to: $(s_{orig} + 2i - 1 + J, R(i))$, which is (6, 1) in Figure A.2b

Example 2

When $j = 2, G(2) = 11 > s_{orig}$. Hence, number 15 at (11,3) in the old matrix, will be moved to the area on the bottom in the new matrix:

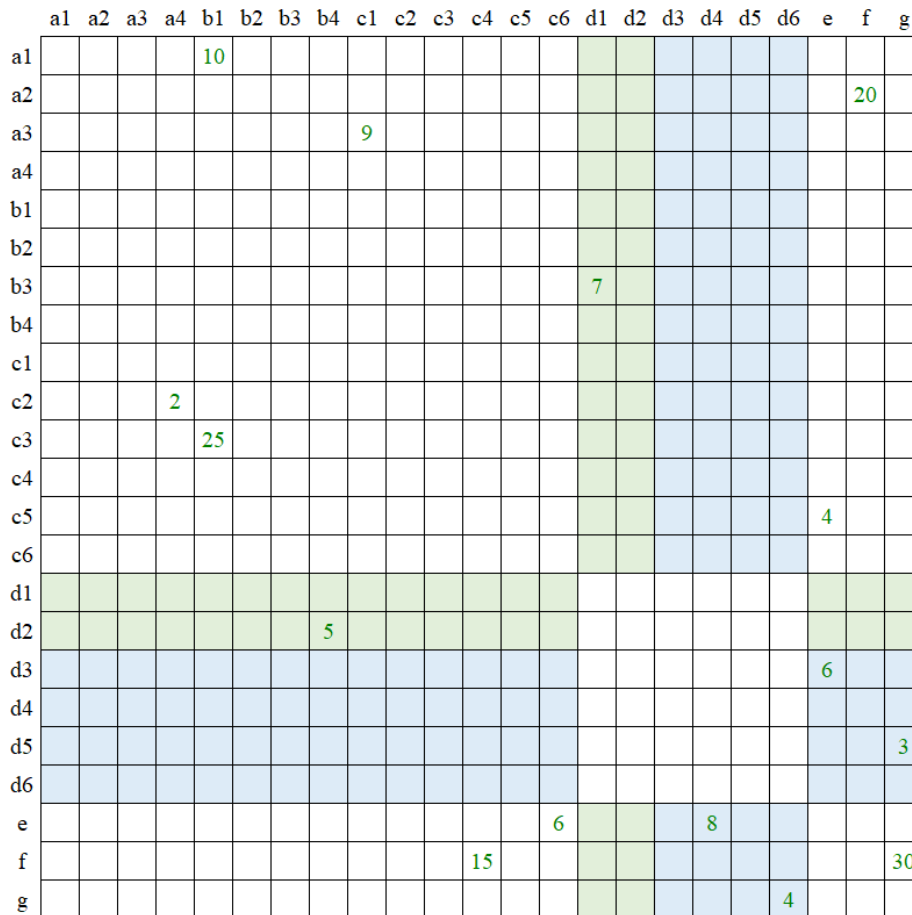
- Number 15 at (11,3) is to be moved to: $(G(j) + 2I - 1 + J, s_{orig} - 1 + j)$, which is (16, 4) in Figure A.2b

Figure A.3 show an alternative, and less desirable manner of splitting, resulting in a matrix of 23 (sub)clusters, out of which 20 are split from 4 original cluster.



(a) Sub-step 5-1 (alternative): split b

(b) Sub-step 5-2 (alternative): split cluster a



(c) Sub-step 5-final (alternative); middle steps omitted

Figure A.3: Splitting of matrix $M_{cluster}$, in an alternative manner

B

Generation of initial solution in the
MILP problem

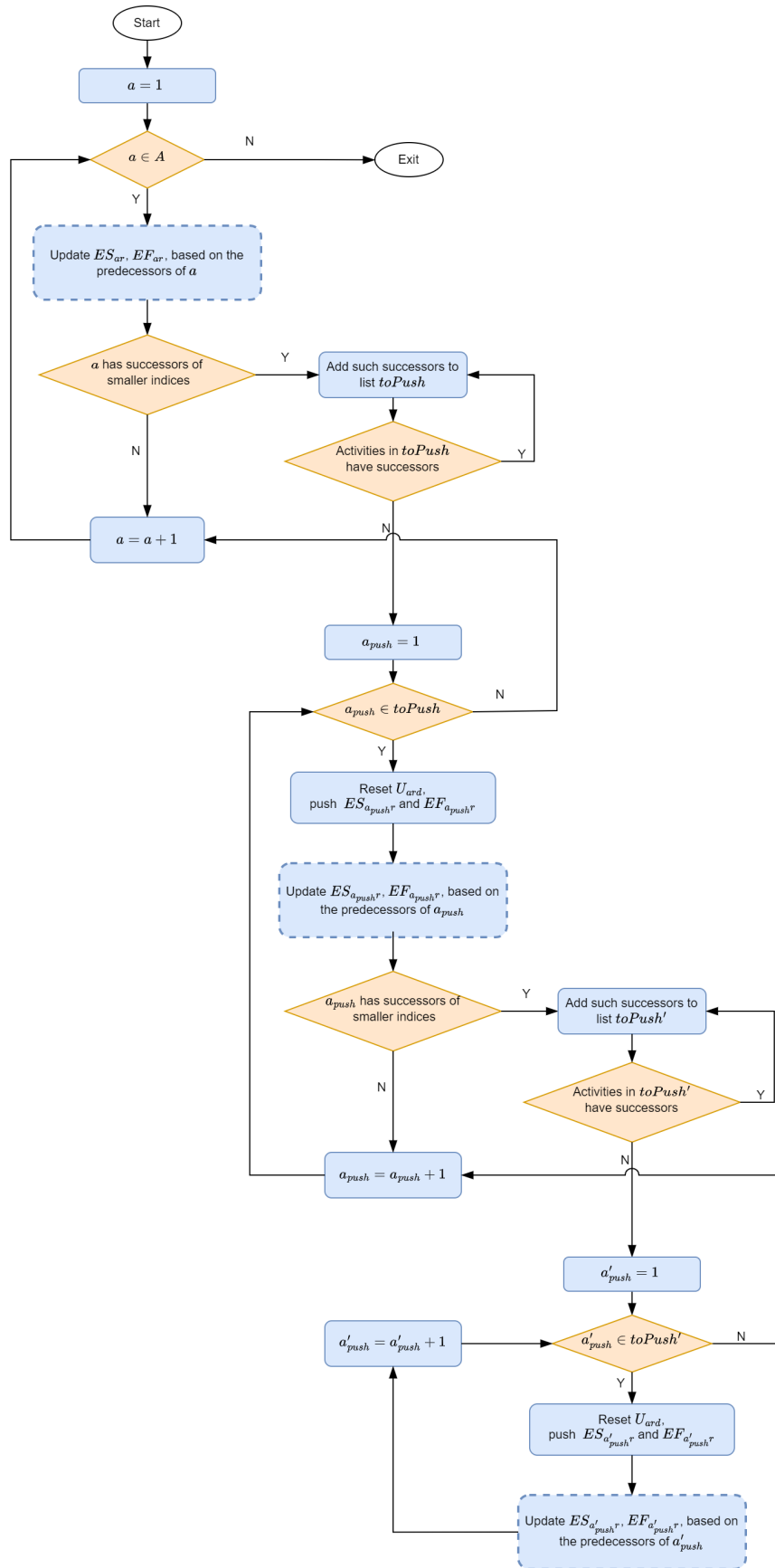


Figure B.1: Workflow of initial solution generation

C

Case study 1: data and scheduling
result

No.	Task Name	Duration	Start (ES_Act)	Finish (EF_Act)	Resources	Predecessor
1	Project start	1	04/05/2022	04/05/2022		
2	Foundation_Fug_Z1	3	04/05/2022	06/05/2022	1	
3	Foundation_Fug_Z2	3	09/05/2022	11/05/2022	1	
4	Foundation_Fug_Z3	3	12/05/2022	16/05/2022	1	
5	Mount Column_F0_Z1	2	17/05/2022	18/05/2022	2,1	
6	Mount Beam_F0_Z1	2	20/07/2022	21/07/2022	2,5,41,1	
7	Mount Slab_F0_Z1	1	22/07/2022	22/07/2022	2,5,6,1	
8	Mount Wall_F0_Z1	1	25/07/2022	25/07/2022	5,6,7,1	
9	Mount Column_F0_Z2	3	25/05/2022	27/05/2022	3,1	
10	Mount Beam_F0_Z2	1	30/05/2022	30/05/2022	3,9,42,1	
11	Mount Slab_F0_Z2	1	31/05/2022	31/05/2022	3,9,10,1	
12	Mount Wall_F0_Z2	1	01/06/2022	01/06/2022	9,10,11,1	
13	Mount Column_F0_Z3	2	23/05/2022	24/05/2022	4,1	
14	Mount Beam_F0_Z3	2	02/06/2022	03/06/2022	4,13,43,1	
15	Mount Slab_F0_Z3	1	06/06/2022	06/06/2022	4,13,14,1	
16	Mount Wall_F0_Z3	1	07/06/2022	07/06/2022	13,14,15,1	
17	Mount Column_F1_Z1	3	26/07/2022	28/07/2022	5,6,7,1	
18	Mount Beam_F1_Z1	2	29/07/2022	01/08/2022	5,6,7,17,1	
19	Mount Slab_F1_Z1	1	02/08/2022	02/08/2022	5,6,7,17,18,1	
20	Mount Wall_F1_Z1	1	03/08/2022	03/08/2022	8,17,18,19,1	
21	Mount Column_F1_Z2	4	08/06/2022	13/06/2022	9,10,11,1	
22	Mount Beam_F1_Z2	1	14/06/2022	14/06/2022	9,10,11,21,1	
23	Mount Slab_F1_Z2	1	15/06/2022	15/06/2022	9,10,11,21,22,1	
24	Mount Wall_F1_Z2	1	16/06/2022	16/06/2022	12,21,22,23,1	
25	Mount Column_F1_Z3	3	17/06/2022	21/06/2022	13,14,15,1	
26	Mount Beam_F1_Z3	2	22/06/2022	23/06/2022	13,14,15,25,1	
27	Mount Slab_F1_Z3	1	24/06/2022	24/06/2022	13,14,15,25,26,1	
28	Mount Wall_F1_Z3	1	27/06/2022	27/06/2022	16,25,26,27,1	
29	Mount Column_F2_Z1	3	04/08/2022	08/08/2022	17,18,19,1	
30	Mount Beam_F2_Z1	2	09/08/2022	10/08/2022	17,18,19,29,1	
31	Mount Slab_F2_Z1	1	11/08/2022	11/08/2022	17,18,19,29,30,1	
32	Mount Wall_F2_Z1	1	12/08/2022	12/08/2022	20,29,30,31,1	
33	Mount Column_F2_Z2	4	28/06/2022	01/07/2022	21,22,23,1	
34	Mount Beam_F2_Z2	2	04/07/2022	05/07/2022	21,22,23,33,1	
35	Mount Slab_F2_Z2	1	06/07/2022	06/07/2022	21,22,23,33,34,1	
36	Mount Wall_F2_Z2	1	07/07/2022	07/07/2022	24,33,34,35,1	
37	Mount Column_F2_Z3	3	08/07/2022	12/07/2022	25,26,27,1	
38	Mount Beam_F2_Z3	2	13/07/2022	14/07/2022	25,26,27,37,1	
39	Mount Slab_F2_Z3	1	15/07/2022	15/07/2022	25,26,27,37,38,1	
40	Mount Wall_F2_Z3	1	18/07/2022	18/07/2022	28,37,38,39,1	
41	load bearing wall_Z1	1	19/07/2022	19/07/2022	1	
42	load bearing wall_Z2	1	19/05/2022	19/05/2022	1	
43	load bearing wall_Z3	1	20/05/2022	20/05/2022	1	
44	Mount Flow segment_F-1_Z1	1	13/05/2022	13/05/2022	45,2,1	
45	Mount Flow segment_F-1_Z2	1	12/05/2022	12/05/2022	3,1	
46	Mount Flow segment_F-1_Z3	1	17/05/2022	17/05/2022	4,1	
47	Mount Flow segment_F0_Z1	6	27/07/2022	03/08/2022	44,48,5,6,7,1	
48	Mount Flow segment_F0_Z2	2	06/06/2022	07/06/2022	9,10,11,1	
49	Mount Flow segment_F0_Z2	1	08/06/2022	08/06/2022	9,10,11,1	
50	Mount Flow segment_F0_Z2	1	30/06/2022	30/06/2022	51,9,10,11,1	
51	Mount Flow segment_F0_Z3	5	23/06/2022	29/06/2022	49,13,14,15,1	
52	Mount Flow segment_F1_Z1	6	05/08/2022	12/08/2022	54,17,18,19,1	
53	Mount Flow segment_F1_Z2	4	24/08/2022	29/08/2022	52,21,22,23,1	
54	Mount Flow segment_F1_Z2	1	22/06/2022	22/06/2022	21,22,23,1	
55	Mount Flow segment_F1_Z2	1	01/07/2022	01/07/2022	21,22,23,1	
56	Mount Flow segment_F1_Z2	1	12/07/2022	12/07/2022	57,21,22,23,1	
57	Mount Flow segment_F1_Z3	5	04/07/2022	08/07/2022	55,25,26,27,1	
58	Mount Flow segment_F2_Z1	2	30/08/2022	31/08/2022	52,29,30,31,1	
59	Mount Flow segment_F2_Z2	1	11/07/2022	11/07/2022	33,34,35,1	
60	Mount Flow segment_F2_Z3	1	18/07/2022	18/07/2022	37,38,39,1	

Figure C.1: Duration data and scheduling result of case project 1.

D

The software tool: list of customisable settings

The software tool is designed in such a way that: (1) it is applicable to not only the case studies covered in this project, but more general cases; (2) the user is allowed to alter some problem settings according to their preferences. This implies customisable settings.

A list of customisable settings is given in the next page.

	Customisable items
Processing the information from input files (component spreadsheet)	Convert the units into the ones required in the problem, if different length-related units (i.e. mm or m) are used in different projects.
	Filter the components which are to be / not to be included when scheduling, by checking if their name or class contains user-specified terms, as not all components are needed in the final schedule. e.g. “keep/delete” components which contain the term “washer” in their “name/class”.
	Modify material and class names into ones that are recognisable to the tool (so that the components can be mapped properly with the database), if different names are used in different projects. e.g. change material name from “beton” (<i>concrete</i> in Dutch language) to “concrete”.
Sequencing rules	Adjust the sequence of structural component (clusters) by class, by ranking the classes per desirable sequence.
	Tighten or loosen the criteria for special spatial relations (SSR) and component size relations (CSR). e.g. change parameters such as $D_{wallmin}$, D_{minX} .
Options for optimisation problem	Choose between a priori and a posteriori. For a priori: set weights of objectives. For a posteriori: set the number of points to check
	Change the termination criteria. e.g. maximum running time.

Table D.1: List of customisable settings

E

The software tool: instructions on how to customise sequencing rules

It is mentioned in Appendix D that the software allows some room for customising sequencing rules:

- Adjust the sequence of structural component (clusters) by class, by ranking the classes per desirable sequence.
- Tighten or loosen the criteria for special spatial relations (SSR) and component size relations (CSR).

How to do that is explained here.

E.1. Identification of floor level

Floor level is one of the criteria when judging the precedence relations, e.g. columns in “floor 1 zone 1” should precede columns in “floor 2 zone 1”. Ideally, the BIM file should contain floor level information, but that is sometimes either missing or inaccurate for some components. That calls for identification of floor level by rules. In this study, the identification of floor level is based on the minimum and maximum value of coordinates, in relation to the elevations of floors.

To judge which floor a building component is categorised to, the software tool should know:

1. Elevations of all floor levels;
2. Rules to judge the floor level of components in relation to floor elevations;
3. If there is special cases where some components should be modified/judged separately.

Step 1: Elevations of all floor levels

Elevations of all floor levels are input in a list, starting with the ground floor level, as shown in Figure E.1. These elevations are generally the top level of floor slabs (without considering floor finishes), though it may be adjusted (as will be explained in Step 2).

All units in this section are millimetre.

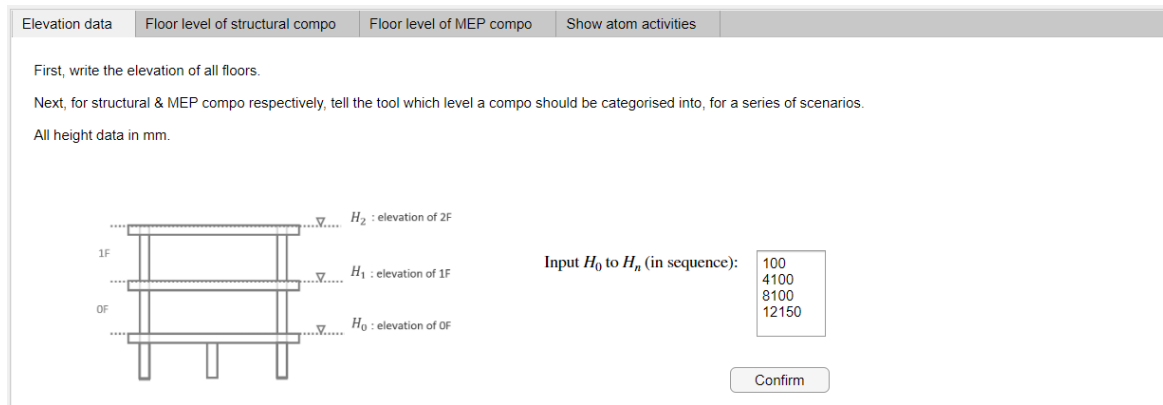


Figure E.1

Step 2: Rules to judge the floor level of components

Most components do not just fall within two adjacent floor elevations specified, e.g. while floor 1 is between the elevation of 4100 and 8100 mm from Figure E.1, a column may have a lower end at 4000 mm, yet it may still be a floor-1 column.

General rules

This is generally dealt with by a set of rules (Figure E.2). A new parameter ΔH is introduced, which acts as tolerance to the floor elevations H_n .

- The illustrations on the left shows nine scenarios where components can be located in relation to ΔH and H_n ;
- The button groups on the right is where the user tells the software tool which floor a component should be assigned to, for all nine scenarios.

Example: in Case study 1, when $n = 1$, $H_1 = 4100$ mm, $H_2 = 8100$ mm, $\Delta H = 300$ mm. A column has its lower and upper end at 3240 and 7760 mm, which falls under scenario 3 in Figure E.2. According to button group 3, this column is categorised into floor $(n + 1)$, that is, the second floor.

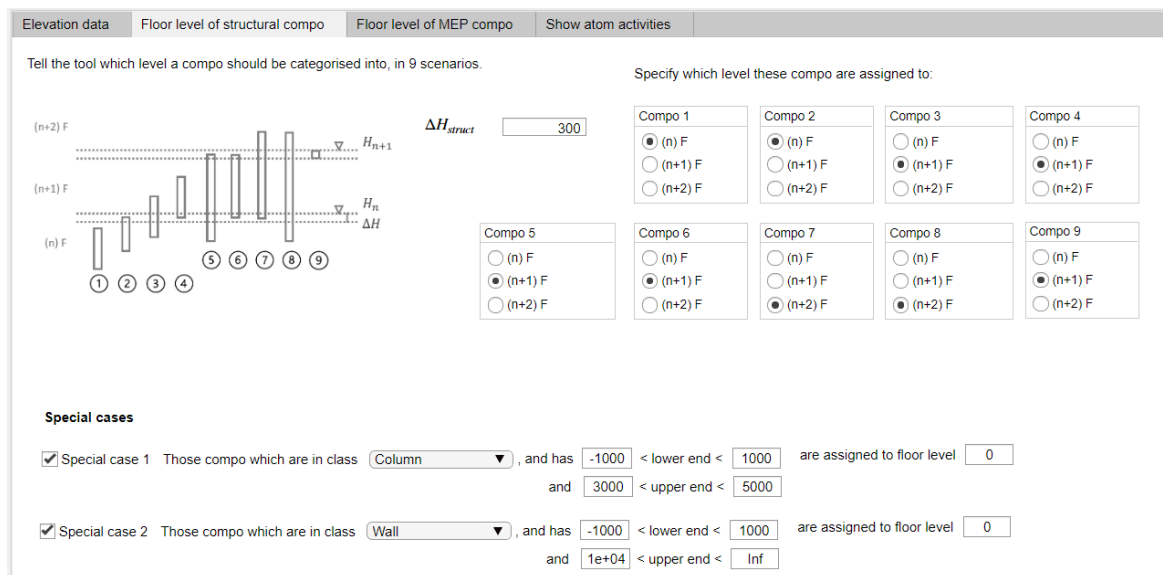


Figure E.2

Special cases

Some components are exceptions to the general rules, e.g. a very tall wall may extend from the ground floor to the second floor. It may be categorised into the second floor under the general rules, but actually we may want to categorise it into the ground floor (to build earlier than other components).

Such special cases can be specified in the lower half of Figure E.2. Two special cases can be specified, by choosing the component class (the available classes are automatically updated in the dropdown list) and the ranges of the elevations of the lower and upper ends.

Example: in Case study 1, some walls have the lower and upper ends at the elevation of -80 and 11800 mm. We would like to make it floor 0. Thus in the bottom of Figure E.2 we ask the tool to put “wall” components which have lower ends between [-1000, 1000] and upper ends between [10000, ∞) into floor level “0”.

E.2. How to customise sequencing rules

Similar to the floor level identification, no one single set of rules is enough to properly sequence all the components. Two sets of sequencing rules can be specified: one for components within one floor level, and one for special case on the whole building.

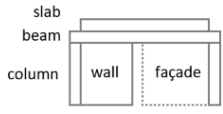
General rules - structural

The general sequencing rules are specified, by selecting component classes in desirable sequence from the dropdown lists in Figure E.3. For the structural system, the user should also tell the tool if a class is an “essential” structural class (more explanation in Chapter 3.2 on activity sequencing rules).

Example: the sequencing rules for the structural system in Case study 1 is shown in Figure E.3. It tells the tool:

- *Within the same floor level and zone, the sequence should be: column -> beam -> slab -> wall;*
- *Within the same vertical zone, all columns, beams, and slabs in floor n should be finished, before components in floor $(n + 1)$ can start;*
- *Within the same vertical zone, walls in floor n do not need to be finished, before components in floor $(n + 1)$ can start.*

Within each floor level:



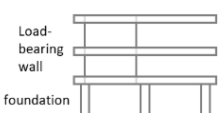
	Seq-1	Seq-2	Seq-3	Seq-4	Seq-5	Essential? *
	Pile	Column	Beam	Slab	Wall	<input checked="" type="checkbox"/>

Figure E.3

Special case - structural

One special sequencing rule can be set. The selection of the dropdown list and the value inputs are very similar to that in the special case for floor identification.

(optional) For the whole building:



Load-bearing wall
foundation

Special case 1*:

class Wall at level 0, and has -1000 < lower end < 1000
and 1e+04 < upper end < Inf

should be before

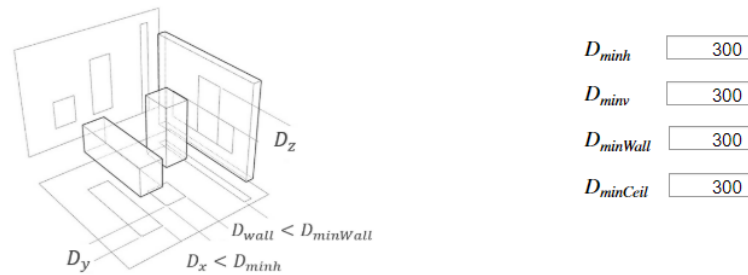
class Beam at level 0

Figure E.4

Rules - MEP

For the MEP system, the sequence among clusters depends on the floor level, class, zone number, special spatial relations (SSR) and component size relations (CSR). The user can easily tighten or loosen the criteria for SSR and CSR, by adjusting the values in Figure E.5: the larger the numbers, the more component pairs will be considered to have SSR and CSR.

Spatial relations



Size relations (length and cross-sectional size):

D_{minh2} <input type="text" value="300"/>	L_{long} <input type="text" value="2e+04"/>
D_{minv2} <input type="text" value="300"/>	L_{short} <input type="text" value="500"/>
	A_{large} <input type="text" value="1e+04"/>
<input checked="" type="checkbox"/> Check length before area?	A_{small} <input type="text" value="1000"/>

Figure E.5