CONSTRUCTION REQUIREMENT-DRIVEN PLANNING AND SCHEDULING WITH SPATIAL-TEMPORAL CONSTRAINTS USING AN ARTIFICIAL INTELLIGENCE APPROACH

YEOH KER-WEI

(B.Eng (Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF CIVIL AND ENVIRONMENTAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2012

Acknowledgements

Writing this acknowledgement has led me to reflect upon the people who have impacted my academic journey over the years. And I cannot help but feel the utmost gratitude for their presence, their guidance, and most importantly, their friendship.

I can name no other person who has made such a great influence on my academic life than my supervisor, *Prof David Chua*. I thank him for the frequent discussions, continuous unwavering support, and valuable advice he has dispensed to me over the years. To me, he has been supervisor, teacher, occasional career guidance counsellor, habitual preacher, and the ever-present friend.

I also wish to thank my friends and colleagues in the construction industry who have taught me some of the practical aspects of construction. Also I have greatly benefitted from their help and insight in the various case studies carried out as part of this dissertation. In no particular order:

1. From **Hock Lian Seng Infrastructure Pte Ltd** (HLS), *Mr Lim Peng Kiat, Mr Kee Guan Chia, Mr Daniel Tay, Ms Ang Kwee Hong, Mr Yunos Karim, Ms Cecilia Loh, Mr Lim Shau Chin, Mr Chai Chee Kean, Mr Steven Ng, Mr Ong Hong Keat, Ms Zhai Lifang, Mr M. Sufian, Mr Fong Kam Wai, Ms Sally Yong, Mr Choo Ket Weng,* and *Mr Saw Yew Bok.*
2. From **Construction Project Integrations Pte Ltd** and **JGC Singapore Pte Ltd**, *Mr Peter Ho, Mr Steven Lee, Mr Neo Chee Keong, Mr Wong Pinyan, Mr Eric Har*, and *Mr Sum Yuwei*.
3. *Mr Lee Chunkit* of **Learners Hub Pte Ltd**, *Mr Ivan Chew Hock Seng* of **Robin Village Development Pte Ltd,** *Mr Huang Yongliang* of **Tiong Aik Construction Pte Ltd**, *Mr Derrick Seah* of **VSL Singapore Pte Ltd**, *Mr Lim Wee Kiat* of **As-Built Pte Ltd** and *Mr Poh Zhihui*.

I also wish to extend my warmest thanks to my lab mates in NUS, namely *Md. Aslam Hossain, Bernard How, Wah Yi Feng, Yap Kim Thow, Cui Rongxin, Cao Jinxin, Chen Jianghang, Qi Jin, Shen Lijun, Liu Zhuo, Yousuf, Simon Falser, Guo Huiling, Ooi Waikeong,* and *Bai Jianhao*. They have been a source of emotional support, and I will miss the occasional midnight “supper” sessions while we slogged through the nights in the lab. Later additions to our lab fraternity also have my thanks: *Kittikun, Harif, Han Ting, Jiexin, Abraham, Zhu Lei, Alireza, Meghdad,* and especially *Trinh Dieu Huong*. Also, special thanks to *Nguyen Thi Qui* and *Ernest Abbott* for helping to proofread parts of this thesis. Lastly, a special mention is made to *Dr Song Yuanbin* who first set me on the path of academic research.

My sincere appreciation also goes to members of my PhD committee *Prof Jerry Fuh* and *Prof Meng Qiang* for their valuable comments during the qualification examination.

Table of Contents

Summary

This dissertation presents a framework on the incorporation of spatial and temporal attributes of Construction Requirements in construction workflow planning and scheduling using artificial intelligence techniques. The term "Construction Requirements Driven Planning and Scheduling" is coined to emphasize the importance of early construction input in planning the construction sequence. Construction Requirements represent the key preconditions for construction and forms the basis for representing critical information and construction knowledge; construction requirements driven planning becomes a key tool in constructability analysis of construction schedules via the early incorporation of construction requirements to drive construction planning.

The knowledge embodied in the construction requirement serves as a sequencing rationale, as well as a tool for analysis of the construction requirement. This knowledge is formally represented as a primitive knowledge construct with the temporal, spatial and abstract attributes, and the interactions between them. Construction Requirements Driven Planning is the planning paradigm where the requirement is defined as the primitive basic knowledge construct, with the temporal and spatial attributes, and their interactions coming into play. A core taxonomy for describing the important aspects of construction requirements is proposed, in which the spatial, temporal and abstract attributes are modelled. This allows the spatial and temporal impact of requirements to be represented for further analysis.

This research further develops the models proposed by prior research in the field of workspace conflict using four-dimensional computer-aided design. The approach developed here analyses spatial demand and supply from the perspective of construction operators, and a modelling methodology based on spatiotemporal utilization is proposed. The utilization factor model is developed to show that the criticality of the operator's spatiotemporal demand leads to worksite congestion and that congestion is a form of worksite conflict. The interference of other space entities increases the space demand, and this increment is quantified with a "dynamic space interference" index. This indicator is developed to identify activity spaces which suffer congestion. A decision making tool, the "congestion penalty indicator," is developed which obtains a schedule-level value for analysis, evaluation, and comparison.

Despite the importance of construction requirements, little attention has been given to the impact of construction requirements on a project schedule, possibly because of the lack of an adequate tool for representing these requirements. Construction requirements are distinguished into static and dynamic types, according to changes in the need of the requirement during its life cycle. A modelling framework, PDM++, is proposed to deal with schedule constraints arising from both static and dynamic construction requirements, provide greater semantic expression to capture schedule constraints unambiguously, and facilitate the representation of interdependent conditional relationships giving rise to alternative schedules. The concept of meta-intervals is also devised to represent complex requirements involving several activities and schedule constraints, and it facilitates modelling at higher levels of plan abstractions. Finally, an evolutionary approach to resolve both spatial and temporal aspects of the construction requirement is introduced.

Keywords: Construction Requirements; Knowledge Representation; Constructability Analysis; PDM++; Alternative Schedules; Artificial Intelligence

List of Tables

List of Figures

Nomenclature

Symbol	***Description***
U_s	Spatial Utilization
U_t	Temporal Utilization
ρ_i	Utilization Factor of Space Entity *i*
$S_{i,A}$	Overlapping Volume between Entity *A* and *i*
S_A	Spatial Volume of Entity *A*
$t_{i,A}$	Temporal Overlap between Entity *A* and *i*
t_A	Temporal Duration of Entity *A*
DSI_A	Dynamic Space Interference of Entity *A*
C	User determined Critical Value of congestion
α	Congestion Tolerance Factor
CPI_A	Congestion Penalty Indicator of Entity *A*
CPI_{Total}	Total Congestion Penalty Indicator for Schedule Comparison
CPI_{Avg}	Average Congestion Penalty Indicator for Schedule Comparison
L	Arbitrary Low value of DSI
m	Arbitrary Moderate value of DSI
N	Number of Space Entities
S_{AB}	Overlapping Area of Entity *A* and *B*
T_{ABC}	Temporal overlap of Entity *A*, *B* and *C*
S_{AC}	Overlapping Area of Entity *A* and *C*
p_{min}	Time Constant for Binary Minimal Literal
p_{max}	Time Constant for Binary Maximal Literal

Symbol	***Description***
q_{min}	Time Constant for Unary Minimal Literal
q_{max}	Time Constant for Unary Maximal Literal
ent	Entity
$Start_i$	Start of Activity *i*
Dur_i	Duration of Activity *i*
Sa	Start Activity
Ea	End Activity
{Name:(Sa,…,Ea),Sa,Ea}	Meta-Interval Definition
{C}	Constraint Set
{CS}	Conjunctive Constraint Set
{DS}	Disjunctive Constraint Set
m	Lag/Lead Value
$x_{l,mode}$	Constraint Validity Boolean Variable (for a specific Mode)

Chapter 1. Introduction

1.1. Research Motivation and Background

This dissertation presents a framework on the incorporation of spatial and temporal attributes of Construction Requirements in construction workflow planning and scheduling using Artificial Intelligence (A.I.) techniques. The term "Construction Requirements Driven Planning and Scheduling" is coined to emphasize the importance of early construction input in planning the construction sequence. Construction Requirements are the capabilities and conditions which the construction process system and the in-progress facility product must conform to. If not, the construction processes may be delayed or temporary stability of the in-progress structure may not be sustained during construction (Song and Chua, 2006). In other words, these construction requirements represent the key preconditions for construction (Chua and Yeoh, 2011). This then forms the basis for representing critical information and construction knowledge; construction requirements driven planning becomes a key tool for constructability analysis.

Every construction project is unique with its own peculiar set of constraints in the form of the above mentioned construction requirements. To represent, and subsequently resolve these constraints and requirements presents the main aim of this research. As will be explored further in greater detail subsequently in Chapter 3, the nature and characteristics of Construction Requirements are varied and wide-ranging covering several important domains in construction like safety, regulatory conformance and construction process. Of these characteristics, the spatial and temporal aspects of the construction requirement on the construction sequence/plan and schedule will be studied in the form of construction spaces and temporal relationships.

Construction Space is often modelled as a construction resource which affects almost every construction activity (Thabet and Beliveau, 1994b). Space Planning and Management plays a vital role in construction project management by identifying and analysing construction space requirements for workspace clashes within the AEC community. Examples of such Space Planning practices include early consideration of various space utilizations in planning site layout, programming high-level construction sequences, and selecting suitable construction methods (Song and Chua, 2005). However, this has often been overlooked in the project management process leading to schedule conflicts and a decrease in productivity due to congestion in the construction space (Zouein and Tommelein, 2001). Reasons for this oversight include the lack of available tools to capture and represent the spatial and temporal components of a Construction Requirement as a project constraint, as well as the lack of an analysis technique to resolve such issues properly.

The consideration of Space Planning and Management in project management is often a critical component in the design and planning process to achieve efficiency and effectiveness in construction. Incorporating these spatial requirements has been shown to give added benefits such as improved safety, decreased conflicts among workers, reduced crew waiting and work stoppage, better quality as well as reduced project delays (Mahoney and Tatum, 1994, Heesom and Mahdjoubi, 2004). Space Planning and Management thus, is a vital component of Constructability Analysis.

As stated previously, some construction requirements have both spatial and temporal attributes. Hence, modelling the spatial attributes only may not be adequate during analysis, and the temporal information needs to be included as well. However, the representation of temporal information for construction space requirements has not been fully explored. Currently, most of the temporal information (logics) for construction space requirements is gathered from schedules provided by project managers through the representations of Critical Path Methods (CPM), of which Precedence Diagramming Method (PDM) is currently the most popular (Wiest, 1981). However, certain limitations are presently known to exist with the PDM which will be further explored in Section 1.2.3. In addition, analysis of the impact of spatial attributes of construction requirements on construction schedules has also not been fully addressed by the research community.

This dissertation provides an overarching framework for conducting construction requirements driven planning and scheduling as part of the Constructability Analysis process. The framework will aid in sequencing construction processes via A.I. techniques (Constraint Logic Programming and Evolutionary Algorithms) with the aid of Four-Dimensional Computer Aided Design (4D CAD). 4D CAD refers to the addition of time as an additional dimension to traditional 3D CAD systems. The use of 4D CAD has become increasingly important to the AEC community, providing a vehicle on which to perform Space Planning and Management (Mahalingam, *et al.*, 2010). 4D CAD provides an excellent platform for communication between the different AEC project participants, allowing for analysis and refinement of work strategies and schedules, particularly in planning, site utilization and pre-construction (Chau, *et al.*, 2004).

1.2. What is Construction Requirements Driven Planning

This research puts forward the idea that construction requirements should be incorporated to drive the construction plan. The knowledge embodied in the construction requirement serves as a sequencing rationale, as well as a tool for analysis of the construction requirement. The key idea behind Construction Requirements Driven Planning is that the requirement should be defined as the basic knowledge construct, with the temporal and spatial attributes, and their interactions coming into play as the requirement is defined. This is opposed to using activities as the primitive knowledge construct in traditional planning frameworks. The framework in this research then projects the temporal attributes for generating and evaluating the schedule. By treating the construction requirement as the basic unit for analysis, this framework transfers the Planner's attention from simply managing the activity to managing *the constraints* of the activity. If carried out during preconstruction as part of the constructability analysis process, this will lead to a more constructible plan/schedule with the identification of key requirements which may potentially impede the progress of construction if overlooked.

1.3. Challenges of Incorporating Spatial-Temporal Requirements in Construction Planning and Scheduling

Despite the advantages of early elicitation of construction requirements for constructability analysis, this is still not being carried out by the AEC community. The reasons for this will be presented in the following section as research challenges for incorporating the spatial and temporal construction requirements.

1.3.1 Challenge 1: Inadequacies of Current Knowledge Representation Approaches for Construction Requirements

Construction Requirements are an acknowledged form of the overall project requirements. Research in this area has tended to focus on the upstream Client requirements through Quality Function Deployment approaches rather than the lower level construction requirements which are required for constructability analysis (Kamara, *et al.*, 1999). Most construction requirements are seldom formally captured due to the ambiguity which arises from using natural language to represent them. Hence, the knowledge embedded inside these construction requirements are not explicitly represented, and cannot be explicitly reused in knowledge-based frameworks. This forms the first major obstacle for representing construction requirements: the need to have a flexible and extendible framework which can then be used to define a suitable taxonomy for construction requirements. This implies a need for a formal method of treating construction requirements to achieve a consistent representation for use as a knowledge representation construct.

The second major challenge lies in developing a domain independent taxonomy. Often construction requirements may come from many varied domains, such as engineering, construction, safety and legal conformance requirements. The traditional methods of representing the knowledge from these have mainly focused on creating a single independent domain. Since the nature of construction requirements are so varied, there is a need to create an upper level ontology for construction requirements, which forms the basis for new and valid taxonomical terms to be added when necessary.

1.3.2 Challenge 2: Inadequacies of Current Spatial Modelling and Analysis Techniques

Current spatial modelling methodologies do not address several issues. Firstly, there is a uniform lack of integration of both temporal and geometric attributes of CAD entities during analysis. Common analysis approaches ignore the temporal attribute or treat temporal and geometric analysis separately. Doing so, could lead to overly conservative decisions. Treated independently, if the geometry of one entity overlaps with the geometry of another, and if a schedule overlap between the two activities associated with the geometries is present, a Planner might be inclined to classify this as a workspace conflict. However, this may be overly conservative if the schedule overlap is not significant. Hence, the interaction between space and time is not captured if treated independently, and this may not capture the reality that human operators may react to obstacles in a flexible manner.

Secondly, workspaces are depicted as "solid" representations in current methodologies. This belies a missing relation between the actual working spaces of operators and the designated activity work spaces. Some workspaces may be large, but the operator's working space is actually very small. This would allow them to accommodate infringements into their workspaces, which present methodologies usually do not consider fully.

Lastly, most analysis methods focus on pair-wise interactions of space entities, and do not extend to multiple overlapping scenarios where several entities overlap amongst themselves simultaneously.

McKinney and Fischer (1998) also highlighted the difficulties of using mental models and present scheduling methods to keep track of project information changes. Project information is often recorded on separate documents and tools, making it difficult for Planners to mentally visualise changes to the construction sequence (Koo and Fischer, 2000). 4D CAD overcomes this difficulty by incorporating the temporal element in 3D models. This has the advantage of visually conceptualizing construction plans/schedules and facilitating communication between project participants, thus promoting the constructability of a project.

Despite the advantages 4D CAD offers to Space Planning and Management, the main challenge lies in the difficulty of detecting and explaining workspace conflicts during the analysis of a project's construction space requirements. This is because several competing space requirements do not necessarily lead to conflict.

1.3.3 Challenge 3: Inadequacies of Current Temporal Modelling Techniques for Construction Requirements

In the AEC community, traditional methods of activity based project planning and scheduling consist of Linear Scheduling Methods (LSM) and Critical Path Methods (CPM). These methods provide Planners with tools to plan project sequences through varied descriptions (semantics) of the interdependencies between activities. Additionally, the representation of these plans has enabled different project participants, from owners to planners, and contractors to suppliers and subcontractors to communicate via a common platform.

In PDM, these semantics include the relationships defined as Finish-Start (FS), Finish-Finish (FF), Start-Finish (SF) and Start-Start (SS), as well as additional lead-lag factors which indicate the minimum amounts by which the start or finish of one activity leads (or lags) the start or finish of another (Moder, *et al.*, 1983).

However, the above methods do not adequately capture many of the temporal aspects of construction requirements, such as work/resource continuity and process concurrency/overlap (Jaafari, 1984, El-Rayes and Moselhi, 2001). Additionally, CPM dictates a specific work sequence although other sequences exist that equally fulfil the construction requirements. These inadequacies of CPM limit the semantic representation of the temporal aspects of construction requirements.

The inability to model construction requirements properly can frequently lead to misinterpretation and even lack of consideration of these construction requirements between the project parties. Consequently, project delays, cost overruns, productivity lapses and inefficiency set in a project. Hence, the expressiveness of present methods has to be increased via a richer semantic vocabulary to better describe the temporal impacts of these requirements. This enriched vocabulary subsequently allows for more detailed construction knowledge and planning considerations to be described within the planning model, consequently enhancing the constructability of a project. With this enriched vocabulary, there is also a need for a method to sequence activities to satisfy the construction requirements, which will be addressed in this research.

1.4. Objectives of Research

This dissertation aims to provide the framework, concepts and procedures to incorporate spatial and temporal aspects of construction requirements into construction planning/scheduling. This construction knowledge driven framework is referred to as Construction Requirements Driven Planning. The primary purpose of this dissertation is to advance the idea of using construction requirements for early stage planning and scheduling in constructability analysis, and demonstrate how the consideration of construction requirements can lead to more constructible schedules, particularly space scheduling.

In particular, the specific research objectives include:

1. Propose an ontological framework for formally describing the spatial, temporal and abstract nature of construction requirements. The objective is to develop a flexible and extendible taxonomical schema for varied

construction domains. This research then describes how the framework can be used to establish various types of construction requirements, particularly workspace resource requirements.

2. Develop a conflict identification and space congestion quantification methodology from 4D CAD to support the analysis of construction workspaces.
3. Develop a suitable representation framework for supporting the temporal attributes of construction requirements in formulating construction schedules.
4. The above mentioned representation framework will provide the basis for further evaluation of the schedule. A prototype solver will be developed for rapid generation of alternative construction schedules under the temporal constraints of construction requirements.
5. Develop a meta-heuristic optimization technique using Genetic Algorithms, which resolves the spatial and temporal interactions on a construction schedule; pertinent resolution strategies for resolving workspace congestion issues will be evaluated.

1.5. Scope of Research

The scope of this research will cover five main areas:

1. An ontological framework for describing construction knowledge, and representing the spatial, temporal and purposive aspects of this knowledge as construction requirements

2. Spatial modelling and analysis methodology for detecting conflict and congestion in construction requirements

3. Temporal modelling and analysis methodology for construction requirements

4. System architecture for the evaluation of the temporal model arising from the construction requirements

5. Meta-heuristic optimization technique for incorporating Construction Requirements into Construction Schedules

In the first area, the ontological framework first introduced by Song and Chua (2006) will be extended. This framework broached the idea that construction requirement is a formalised representation of some aspects of construction knowledge. In particular, the intermediate functional requirement was introduced to capture knowledge relating to the transient functionality of the temporary structures to support the construction product. The extensions in this research include a flexible and formalised representation for various types of construction requirements including functional and non-functional, as well as defining workspace resource requirements. This ontological framework will serve to tighten the integration between the product and process perspectives through the consideration of construction requirements. Also the study of the ontological framework will determine the core characteristics of construction requirement entities and subsequently develop a suitable taxonomy for describing construction requirements.

In the second area, this research will develop a more robust spatial modelling methodology which will mathematically incorporate the temporal and geometric attributes, allowing work process flexibility to be modelled. The methodology developed will also enable multiple overlapping activities to be quantified, while mitigating the “solid” nature of the space entity implied by previous models. Two indexes will be developed to provide a measure of congestion. This will allow several alternatives to be evaluated and consequently resolve any potential workspace congestion.

In the third and fourth areas, the temporal attributes of the construction requirement will be represented and described using a semantic logic specially developed for representing construction requirements. The semantic logic will be translated into an equivalent end-points formulation for mathematical representation and evaluation. Subsequently, an analysis of the criticality of the requirements will enable Planners to better control the construction schedule. Sometimes the construction requirement may be complex or conditional on other requirements, requiring increased expressiveness of the model which is beyond the capabilities of traditional Critical Path Methods. This research forwards the hypothesis that traditional activity-oriented planning and scheduling in construction is not adequate, and that construction requirements form the basic knowledge for constructing a construction schedule. A solver prototype will be discussed as part of the evaluation mechanism for temporal constraints in construction requirements.

In the final area, the resolution of spatial-temporal conflicts arising from conflicting construction requirements will be developed through a meta-heuristic optimization technique. The resolution methodology will allow various schedules to be compared, enabling the effects of schedule compression to be studied on the overall schedule. Treating space as a type of resource, some trade-off between the schedule and the amount of spatial-temporal conflict can be expected. This will be demonstrated in the case study provided, where several alternative scenarios are compared.

1.6. Research Methodology

The research methodology is presented in Figure 1.1. The research methodology is made up of four main steps: (a) Developing the Research Objectives, (b) Gathering of Research Data, (c) Generating Research Outputs, and (d) Analysing and Validating Research Outputs through Illustrative Industrial Case Studies.

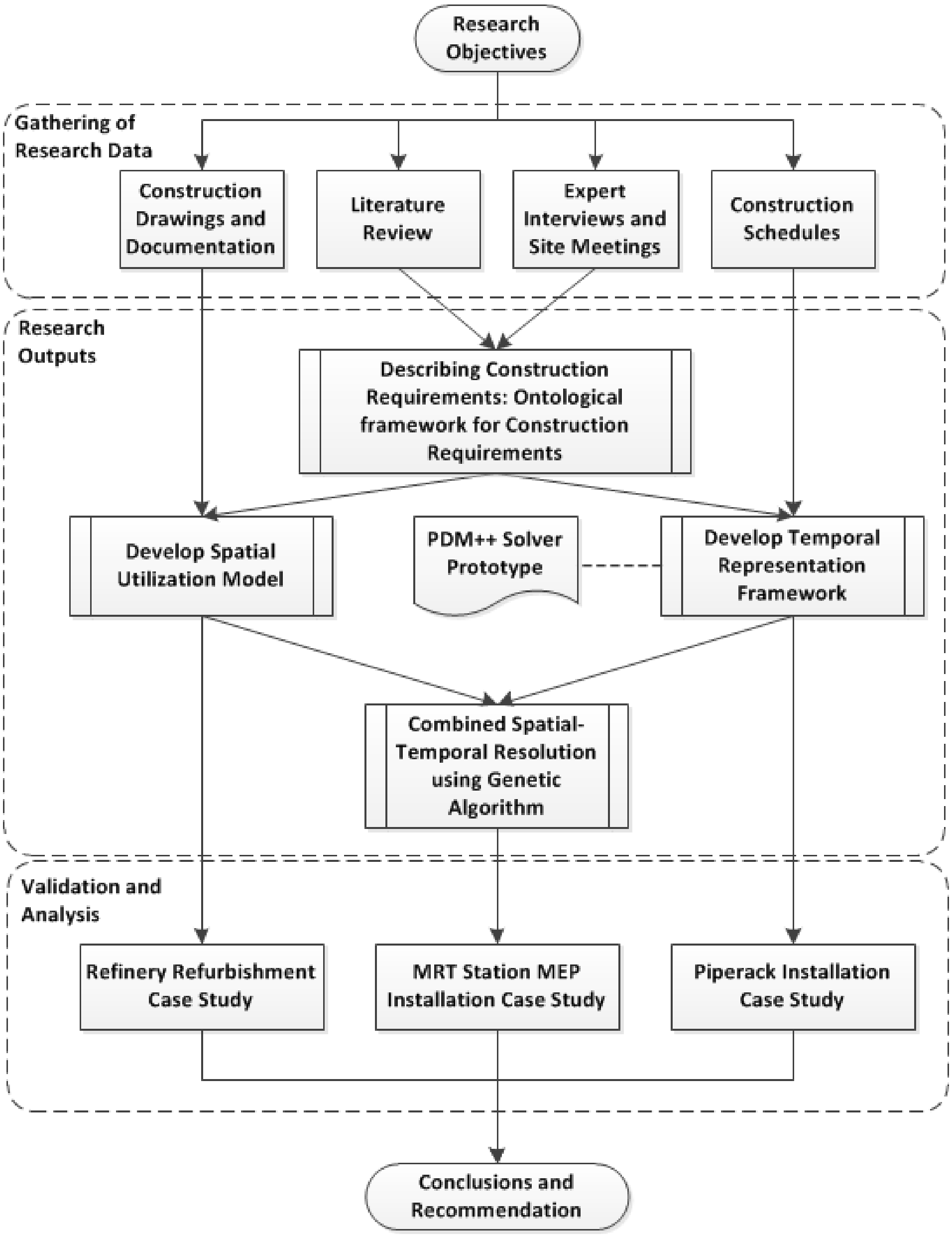


Figure 1.1. Logical Dependencies within Research Methodology

The Research Objectives were iteratively developed through various modes of data collection. These Data Collection modes were generally from Construction Drawings and Documentation, Academic Literature Review, Interviews and Site Meetings with Experts and Construction Schedules. Construction Drawings and Documentation gathered through the course of this research included the various site documents like Project Quality Plan, Project Management Plan, Conditions of Contract, Safety Management System, etc. Literature Review of relevant academic materials was also conducted to determine the state-of-the-art. Expert Interviews and Attending of Site meetings were also carried out with various Construction Managers and Project Managers of several companies, including JGC (Singapore) Pte Ltd, Construction Project Integrations Pte Ltd and HLS Infrastructure Pte Ltd. The various construction schedules for respective projects were also consolidated for analysis and validation of the case studies.

From the Research Outputs generated, an ontological framework for describing construction requirements was generated which frames the direction for this research work in terms of the spatial and temporal perspectives. Spatial and temporal representation models were then developed to capture the aforementioned spatial and temporal perspectives and interactions independently. Each representation model was then validated with an industrial case study and analysed. Finally, the combined model was also validated with an industrial case study.

1.7. Organization of Thesis

This dissertation is organized into nine chapters including this introduction as shown in Figure 1.2. Chapter 2 broadly covers the relevant concepts and topics related to this dissertation. Its purpose is to review computer aided constructability tools in present use within the AEC community, as well as cover some background concepts in artificial intelligence tools used in this research. More detailed reviews will be recorded in specific chapters to improve readability. Chapter 3 commences with a more detailed survey of ontological frameworks for construction requirements together with a discussion on the definition, nature and evolution of Construction Requirements.

Chapter 4 starts with an in-depth survey of present spatial modelling methodologies, followed by the identification and quantification methodology for construction space conflicts from a requirements perspective.

Chapter 5 initially reviews present temporal representations and other planning paradigms in management science and computer science. The developed model PDM++ is covered with its representation methodology. Chapter 6 documents the system architecture of the PDM++ solver prototype using $ECL^{i}PS^{e}$ Constraint Logic Programming system with a discussion of the underlying mathematical concepts to show how the solver prototype handles the representation methodology and evaluates the model.

Chapter 7 provides some background information on Multi-mode Resource Constrained Project Scheduling Problem with Generalised Precedence Relations (mmRCPSP/max). The combination of the representation methodologies from Chapters 4 and 5 is discussed, and a multi objective genetic algorithm to resolve the spatial-temporal interaction issues which arise from construction requirements in a construction schedule is proposed.

Chapter 8 consolidates the various industrial case studies employing the models in Chapters 4, 5 and 6. Each case study is analysed with management implications presented herein.

Chapter 9 concludes the thesis, summarizing the research contributions of this dissertation. Further suggestions for future research and development directions are covered within this chapter.

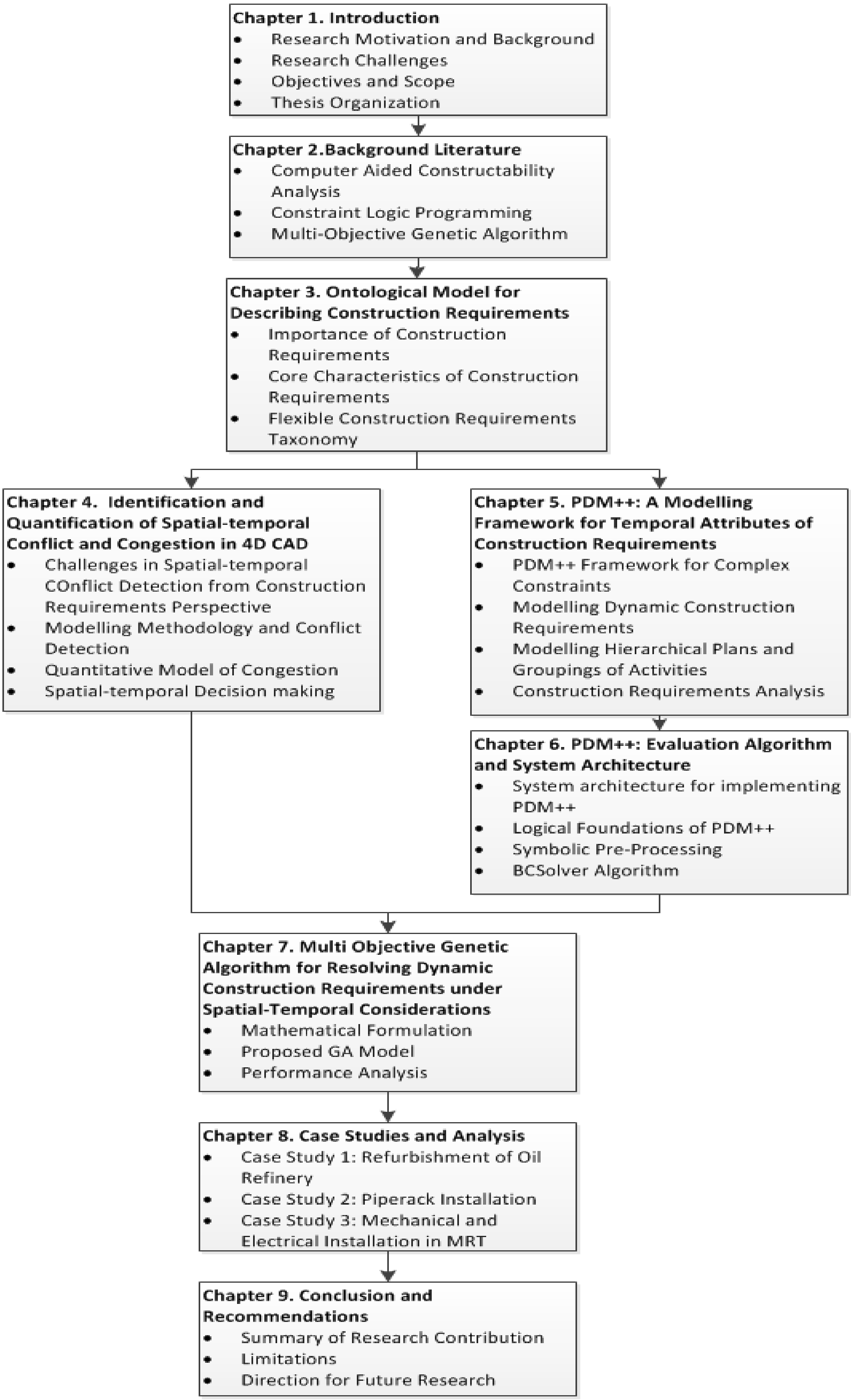


Figure 1.2. Organisation of Thesis

Chapter 2. Review of Background Literature

2.1. Introduction

This chapter presents a broad overview of the background literature relevant to comprehending future chapters of this dissertation. The purpose of this review is to allow readers to understand underlying concepts employed in existing CAD-based constructability analysis methodologies, and assess how these methodologies incorporate constructability knowledge for feasible planning and scheduling. Future chapters will present more detailed aspects of literature relevant to the content of the chapters.

The benefits of constructability input in the early stages of the project plan has been studied by Tatum, *et al.* (1986), and some of these have been identified:

1. Early constructability input has great early cost influence, as during the feasibility study and preliminary design phase, the level of expenditure is low in relation to total project cost, but the influence on the project outcome may be very large.
2. Reduction of work scope to meet minimum client requirements may be achieved by analysing the client's intent to make sure that the design is not over-built and consistent with engineering principles.
3. Reduce construction difficulty leading to increased quality and enhanced safety. The early consideration of site specific considerations on erection sequence and construction methodology, like storage, access and space limitations can ensure that a design is more constructible, ensuring better chances of project success.

Griffith and Sidwell (1993) have gone beyond the Construction Industry Institute's (CII) definition of constructability as "a system for achieving optimum integration of construction knowledge in the building process and balancing the various project and environmental constraints to achieve maximization of project goals and building performance". Their extension to the definition focuses on the identification of balancing constraints under the maximization of the project objectives. The implication of the statement is that such a system is often complex with large sources of information to conceptualize as construction knowledge. Hence, the need for information technology to aid in the knowledge management aspect within construction through acquiring, representing and utilizing the construction knowledge and information (Skibniewski, *et al.*, 1997). Additionally, Fisher, *et al.* (2000) surveyed and identified computer based tools like lessons learnt databases, geographic information systems and CAD as potential tools that support the constructability review process. The review in this chapter will primarily focus on CAD based tools for constructability analysis and review instead.

2.2. Review of Computer-Aided Constructability Analysis Methodologies

This section presents a review of computer-aided constructability analysis approaches which improve the decision making process for Planners by providing a mechanism which codifies construction knowledge for use in the early implementation of constructability in the construction plan/schedule. These approaches also provide some mechanism to identify potential constraints and conflicts early in the planning or design phase.

2.2.1 CAD-Integrated Knowledge Based Planning Systems

The first of two flavours of these computer aided tools falls under the category of CAD-Integrated Knowledge Based Planners (KBPs). These KBPs generally comprise a knowledge representation/acquisition facility, inference engine and a knowledge base of domain rules and facts to determine the construction sequence. While these systems do not claim to be used for the purpose of constructability analysis, they use construction knowledge to derive the construction schedule, and this knowledge is intimately linked to CAD representations of the physical products. Various CAD-Integrated KBPs exist in the AEC industry, and only the more established KBPs reviewed will be reviewed to demonstrate the main idea behind interfacing process knowledge with product models for construction planning.

The focus of these systems tends to be on its own proprietary domain-specific rules, which is often difficult to share between other domains and KBPs. The need for such KBPs is obvious: there exists a large amount of data and knowledge in the AEC community which could aid the construction process if available. The difficulty is that such knowledge is often unstructured and thus difficult to capture and disseminate.

Additionally some KBPS may not support the function of correcting problems with existing plans, or conduct further analysis to determine if additional discrepancies exist within an existing model, or whether additional optimization is possible (Lee and Soh, 1993).

2.2.1.1 Construction PLANEX

PLANEX (Zozaya-Gorostiza, *et al.*, 1990) was initially designed to plan and schedule construction excavation, but was later extended to other construction domains. PLANEX takes in a description of elementary components using the

MASTERFORMAT system, which is an industry standard of cost codes. Each of these components is a representation of the project at its lowest level, and includes its geometric information with other pertinent attributes.

PLANEX further distinguishes element activities and project activities. Element activities are associated with each component, while project activities aggregate the element activities together at a higher level for project planning. The element activities form the basis for rules which determine the sequence for construction through its attributes. Simple precedence relationships for common activities are stored as part of the knowledge base, and reused to generate the successors for activities.

PLANEX demonstrated the feasibility of knowledge based systems for construction planning in specific domains, and introduced the idea of having the construction product drive the construction schedule through its element activity representation. This framework enabled the product knowledge to be integrated directly into the project activity.

The construction knowledge in PLANEX is implicitly stored in its knowledge database, and the knowledge is evidenced by the generation of the precedence relationships between the different elements. This made it highly domain specific and restricted its reuse on similar domains. Additionally, PLANEX does not explicitly capture workspaces requirements for activities, which could be a vital component for constructability analysis.

2.2.1.2 OARPLAN

OARPLAN (Object-Action-Resource Planner) is a model based planner from Stanford University's Centre for Integrated Facility Engineering (CIFE), which uses the physical description from CAD entities as its basic knowledge construct (Darwiche, *et al.*, 1989, Winstanley, *et al.*, 1993). Similar to PLANEX, each of these entities in OARPLAN is able to store information such as its material type, strength etc. OARPLAN hierarchically stores the entities as components in component systems, with additional component geometric and topological relationships.

Activities in OARPLAN are defined as an action that is applied to a component, and requires resources. These activities may be subactivities in a hierarchy of tasks, which allows greater granularity for control over the plan. The activity dependencies embody the construction knowledge within the system. These dependencies are inferred from the relationship between the subactivities, other activities, and other component entities. The dependencies are stored and recalled from the knowledge base when needed. OARPLAN demonstrates true causal reasoning capabilities through the IF-THEN rules used in its knowledge representation and thus features a more robust inference engine in comparison to PLANEX.

Construction Methods are not explicitly considered in OARPLAN. However, CIFE introduced another planner which explicitly represented and reasoned about construction methods through a similar architecture (Fischer and Aalami, 1996). Their approach treated construction methods as the basic knowledge concept for transforming design to a feasible construction schedule. Similar to PLANEX, OARPLAN does not explicitly handle workspace requirements.

2.2.1.3 KNOW-PLAN and KBS

KNOW-PLAN (Morad and Beliveau, 1994) utilizes an object oriented representation where the building component is the primitive object with geometric data attributes. In addition to the geometric data, the type of connection with other objects around it is also stored. KNOW-PLAN focused on formalizing the dynamic sequencing knowledge/rationale in its Dynamic Sequencer module as rules, taking into account the geometric data and zone allocation of components. The activity relationships are declaratively obtained as sequence facts. KNOW-PLAN's close integration with the CAD model allowed it the functionality of visualisation as well, which will be discussed in the next section.

Echeverry, *et al.* (1991a) proposed Knowledge-Based-Systems (KBS) which specifically dealt with the factors determining activity sequences and precedence identified: Physical relationships, Trade interaction, Resource limitation and Code regulations. Based on the geometric relationships between the objects and the relationships between the classes to which the object is related to, the sequencing rationale can be constructed. If objects belong to the same class, then direction of installation is used to assert the sequence. Otherwise, the connection type and the geometries of the objects involved determine this sequence. Also impacting the sequencing rationale is the construction space required by the trade crews and equipment which are explicitly represented and embedded in the objects within KBS. KBS specifically accounted for the trade interactions and the associated workspaces within its knowledge base. However, the sequencing knowledge was still derived as basic precedent relationships.

2.2.2 Visualisation Tools for Constructability Analysis

Often, constructability is affected by site restrictions or space requirements, and some of the analysis tools have incorporated this in their mechanisms. The second category of computer aided constructability analysis tools emphasises on visualisation to detect potential construction problems. Visualisation of the construction process plays a vital role in the constructability analysis of construction workspace.

Visualisation is achieved through linking the 3D CAD model with the construction schedule, and detected conflicts using visualisation are then resolved through reconfiguration of the workspace either spatially (rearranging the work or storage areas) or temporally (rescheduling activities in the construction plan).

These visualisation tools also enable communication and generation of the design, sequencing and scheduling knowledge between constructability experts and owners/end-users. This enables experts to best apply their knowledge to balance the expectations of the owners/end-users (Hartmann and Fischer, 2007). The review of the following visualisation tools reveal how 4D CAD models are able to computationally support constructability reviews, providing a more exact (although not always necessarily better) form of analysis over other paper-based management techniques.

One of the major disadvantages of present commercial visualisation tools is their inability to conduct numerical analysis on a given problem. This arises from the lack of an interface for the tool to retrieve data from the 4D CAD model. Often the analysis carried out using visualisation tools is by human inspection, and this may limit the insight Planners might gain from the model (Jongeling, *et al.*, 2008). Also, this means that some form of schedule optimization may not be possible without the data from the 4D CAD model, which inhibits the usefulness of 4D CAD to the industry.

Visualisation tools for constructability may be differentiated into two forms: Deterministic visualisation of construction schedule, and stochastic visualisation. Depending on the intention of the Planner, physical aspects of key resources, temporary structures, materials and labour may be added to the model to enhance the constructability analysis. These intentions may encompass site utilization, temporary structure usage and safety (Park, *et al.*, 2011).

2.2.2.1 Deterministic Visualisation Techniques

Deterministic visualisation techniques may additionally depict spaces for movements, transformation and interactions between these entities. Deterministic techniques have the advantage of being easier to use: Commercial software (Autodesk Navisworks, Bentley Navigator, Tekla BIMSight, Synchro etc) are readily available that are founded on these deterministic techniques. In general, a deterministic visualisation may be accomplished by directly linking the 3D component with the activity in the schedule. Also deterministic schedules may be better suited to comparing actual with baseline schedules during project control.

The disadvantage of deterministic techniques is that the interactions of the construction resources may not be adequately captured as they are modelled from the activities. This means that we can see the evolution of the CAD entity with respect to the activity along a timeline, but this is in no way dependent on the resources that drive the actual construction (Kamat, *et al.*, 2011).

Dawood and Mallasi (2006) attempted to bridge the interaction of resource within the activity scope of the deterministic technique by incorporating patterns of workspace execution with critical analysis of site spaces (PECASO Model). They also incorporated production behaviour for activities to better simulate the movement of resource operations. Other researchers looked at different methods of schedule representation to represent the temporal dimension, where the representation of the resource operation is more closely integrated. Jongeling and Olofsson (2007) proposed a location based scheduling / line-of-balance tool with 4D CAD to plan work flows.

2.2.2.2 Stochastic Visualisation Techniques

Stochastic visualisation techniques differ from Deterministic visualisation techniques in the use of a probabilistic simulation engine to model various possible construction scenarios (Kamat and Martinez, 2001). These stochastic simulations are a combination of a discrete event simulation engine with the 3D visualisation capability. Due to the underlying model, stochastic visualisation techniques readily avail themselves to risk analysis and management during the early phases of the project to depict "what-if" scenarios. Often, these stochastic visualisation techniques have been employed to model low-level construction resources, equipment, and operations (Kamat and Martinez, 2005).

One of the challenges with stochastic visualisation techniques is the handling of extreme probability events or "black swan events". These "black swan events" refer to events which have small probabilities of occurring, but have a major impact on the construction system. Such critical issues may not be captured adequately during scenario analysis, and may even be missed completely (Klein and Herskovitz, 2005).

Tantisevi and Akinci (2007) identified a related time granularity issue for stochastic techniques in their paper on generating workspaces which encapsulate the working envelope of mobile crane operations. The discrete event simulations usually employed in stochastic visualisation techniques may not detect conflicts which occur during the step size in the simulation engine.

2.2.3 Other Computer-based Constructability Analysis Tools

The above tools focus primarily on the impact of constructability knowledge and its impact to planning and scheduling. This does not belie the full scope that constructability analysis can offer to the industry. In addition to the knowledge bases and visualisation tools for planning and scheduling purposes, other computer based constructability tools include Constructability Knowledge Expert (COKE) and Automated Rebar Constructability Diagnosis for determining specific constructible structural design (Fischer, 1993, Navon, *et al.*, 2000). Bansal (2011) also used Geographical Information Systems (GIS) to represent topographical and geospatial features which affect workspace planning on the construction site to complement existing 4D Constructability Analysis.

Constructability lessons learnt can be kept as records of previous constructability issues in databases. Kartam and Flood (1997) propose an interactive computerized method called the Constructability Lessons Learnt Database which facilitates the storage and retrieval of previous project cases, documented constructability knowledge and lessons for future projects. Understanding that the knowledge acquisition phase in building a knowledge based system is the most tedious and important part of the knowledge engineering process, Skibniewski, *et al.* (1997) present a machine learning approach to elicit constructability knowledge as rules in a data system. Hanlon and Sanvido (1995) also present an information architectural approach to store and subsequently retrieve this information in database systems. Soibelman, *et al.* (2003) introduced a system (jointly with the U.S. Army Construction Engineering Research Laboratory, CERL) and a framework to complement the existing design review process. This framework was closely linked to a lessons learnt database which is able to collect individual experiences during the design review process for sharing and reuse in future projects.

While not a CAD-integrated Knowledge Based Planner, M-RAM (Multi-Reasoning Model) evaluates the constructible feasibility of a proposed model via a distributed multi-reasoning mechanism during the conceptual design phase (Soibelman and Pena-Mora, 2000). At the knowledge representation level, the M-RAM object model explicitly captures the purpose and the intent of the design via geometric properties like height, stories and wind velocity, and a ranking/satisfaction metric respectively. The intent was also further decomposed into the objective, constraints and function attributes, providing the performance criteria and measures needed for ascertaining the adequacy of the design. These attributes of intent are the precursor to the Construction Requirement taxonomy which will be discussed later.

2.3. Summary of Specific Literature Reviews

The above state-of-the-art review of computer based constructability tools demonstrates the wealth and depth of construction knowledge available. However, the construction knowledge is highly fragmented, with none of the reviewed systems particularly suited to support the concept of construction requirement-driven planning and scheduling. The individual chapters in this thesis look at the different facets of construction knowledge, encompassing the ontological, spatial, temporal, and evaluation mechanism using artificial intelligence.

Current ontological approaches for modelling construction knowledge are domain specific and scaling the taxonomies to fit other domains represents a key difficulty. Construction requirements may span several construction domains, and require a consistent modelling methodology to represent the implicit construction knowledge. Moreover, construction requirements differ from traditional manufacturing or software requirements modelling which tend to focus on functional and non-functional types of requirements respectively.

The spatial modelling methodologies which will be reviewed later study the quantitative aspects of workspace interactions. These interactions lead to conflict, and congestion. Present methodologies may be categorised into project-level, activity-level and operator-level perspectives. The project and activity level perspectives allow a global handle to quantify workspace congestion, while the operator's position is intuitively more useful to develop management strategies for congestion. However, segregating the different levels of perspective prevents the transference of knowledge, limiting the use of existing 4D CAD methodologies for spatial modelling and analysis.

Temporally, representing activity information is also restricted to present CPM methodologies. Such traditional methodologies restrict the representation of activities to only one possible schedule. As such, there is a need to generalise the representation of activities to allow conditions and alternatives to be embedded as part of the construction requirements knowledge. Also, CPM is limited in its expressiveness, and may not be fully suitable to capture construction intentions like work continuity, resource overlaps etc.

Combing both the spatial and temporal modelling perspectives to provide a more comprehensive picture of construction requirements poses a new research problem of determining how to evaluate a schedule which fulfils the construction requirements, considering the effects of congestion. A review of the literature reveals that the mmRCPSP/max problem comes closest to representing the new research problem, but there are very few evolutionary methods available. Moreover, the spatial resource behaviour differs from the traditional resources often found in Operations Research.

In summary, the in-depth survey of the literature reveals several inadequacies which need to be addressed so that the construction requirements representation can be an effective modelling and analysis tool.

2.4. Overview of Relevant Artificial Intelligence Tools

2.4.1 Constraint Logic Programming in Planning and Scheduling

Constraint Logic Programming (CLP) is a declarative programming language, combining the paradigms of logic programming and constraint programming. The CLP methodology extends the initial Prolog language by incorporating several types of constraint solvers, where each constraint solver is particularly suited for a specific domain. Interested readers may wish to refer to the following references for a more in-depth discussion on the workings of CLP: (Jaffar and Maher, Apt and Wallace, 2007).

Chan and Paulson (1987) proposed the use of constraint satisfaction to drive the engineering design. Their approach was able to heuristically explore alternatives and automatically propagate changes so that the changes are accommodated by changing other design parameters. Their system was implemented using CLP, showing the application of constraint propagation in efficiently searching for alternatives. This is an important consideration in this dissertation where multiple alternatives have to be efficiently handled.

CLP has also been successfully used as a planning and scheduling tool, due to its ability to use constraints actively to reduce the computational effort to solve the combinatorial nature of scheduling problems (Caseau and Laburthe, 1994). This is mainly achieved through domain reduction mechanisms via constraint propagation algorithms. Baptiste, *et al.* (2001) noted that the performance of CLP schedulers is comparable to traditional operational research approaches, if not better for most problem instances, while offering greater model flexibility. The proposed methodology in this dissertation takes advantage of this flexibility to develop the modelling framework for analysing more complex planning considerations.

Parts of this dissertation assume that the reader has a certain level of familiarity with some of the basic programming concepts in Prolog/*ECL^{i}PS^e*. The reader may refer to Apt and Wallace (2007) for a summarized introduction to Prolog, and its extension to CLP.

2.4.1.1 Overview of ECL^{i}PS^e Constraint Logic Programming System

The *ECL^{i}PS^e* Constraint Logic Programming System is a dialect of the Prolog programming language, and is used in this research for modelling the temporal constraints. The fundamental mechanisms of Prolog are unification, tree-based data structures and automatic backtracking. Simply, Unification is the mechanism of matching various variables, while the tree-based data structure facilitates the automatic backtracking mechanism. This allows a declarative approach to programming, where the user queries rather than dictates the procedures in a system.

Prolog is closely related to mathematical logic, and has a syntax which follows that of first-order predicate logic formulas. It is possible to think of Prolog as a list of goals, where a goal is posted by the user which the system attempts to verify if it is true. This validity occurs if the user has defined such a clause (or an instance of such a clause) in the program, such that the head is identical to the goal, or if the goals in the body are true.

ECL^{i}PS^e extends the Prolog syntax by allowing back-end custom-built solvers to work closely with the Prolog mechanisms. These solvers are built to reason efficiently on particular domains, like real numbers, rational numbers or boolean domains. In this research, an interval, discrete, and finite based domain is identified and used to model the problems.

2.4.2 Multi-Objective Genetic Algorithm in Planning and Scheduling

Multi-Objective Genetic Algorithms (MOGAs) are population based meta-heuristic algorithms developed to find the Pareto frontier of a problem. MOGAs differ from ordinary GAs in that a set of non-dominated solutions is found to represent this Pareto frontier.

The application of MOGAs to planning and scheduling is widespread in the literature. This is probably due to the efficiency in which MOGA is able to find near-optimal solutions for traditionally NP-complete problems. This is particularly pertinent to construction engineering, where often optimality is not necessary but speed of obtaining a solution is. Also, MOGAs accord some level of control within the algorithm, allowing users to determine termination criteria while still obtaining better answers than at the start of the algorithm. In general, there are two common types of Pareto-based MOGAs prevalent: Non-Dominated Sorting Algorithm 2, NSGA-II (Deb, *et al.*, 2000) and Strength Pareto Evolutionary Algorithm 2, SPEA2 (Zitzler, *et al.*, 2001).

Variants of both algorithms have found widespread usage in construction planning and scheduling. Zheng, *et al.* (2004) have studied the usage of MOGAs on time cost optimization, where they concluded that MOGAs can assist Planners to concurrently arrive at optimal project durations and cost. Hyari and El-Rayes (2006) also propose a similar formulation with work crew scheduling. Jakowski and Sobotka (2006) also show that MOGAs is versatile and highly applicable to construction scheduling, where resource constraints are dynamic and changeable in time.

2.5. Concluding Remarks

This chapter provides a review of the computer aided approaches in constructability analysis. Of these approaches, two in particular were reviewed in-depth: CAD-integrated KBPs and Visualisation tools. In general, KBPs are construction knowledge expert systems that generate construction schedules, while visualisation tools incorporate the CAD entities with the construction sequence for analysis.

This chapter looks at traditional KBPs and contrast the capabilities of these systems, identifying two potential gaps: Their lack of representation of construction workspace, and their focus on specific domains of construction knowledge. The visualisation tools are also compared, with two flavours of the tools identified: deterministic techniques and stochastic techniques. Analysing these two techniques reveals that deterministic techniques suffer from the lack of incorporating resource operation specific information, while stochastic techniques could be susceptible to time granularity problems during analysis.

The objective of this survey of the present technology is to give a sense of where the AEC industry is with respect to the applicability of computer based constructability tools by covering the broad gaps in the present technology. Specific gaps will be addressed in the individual chapters of this dissertation. Through the identification of these gaps, the work presented in this dissertation will seek to complement and enhance the capabilities of Planners in reasoning about constructability issues.

The broad gaps identified frame the direction of this dissertation: Firstly, provide Planners with the ability to incorporate knowledge via the identification of construction requirements within the construction plan. The knowledge inside the construction requirements requires the development of a taxonomy to adequately describe the spatial and temporal characteristics of its associated workspaces.

Secondly, the visualisation tools reviewed allow for the spatial aspects of the construction knowledge to be evaluated and analysed, and will continue to play a key role in determining plan feasibility. A deterministic approach to analysing the spatial aspects of construction requirements will be chosen due to its ease of applicability to optimization methods. However, the incorporation of resource specific information needs to be addressed to allow for more detailed analysis to be conducted. This will be achieved in the next chapter by abstracting the utilization of a space in discrete time to a continuous space-time-volume, providing a 4D tool which gets around the time granularity issue with the incorporation of resource.

Also, this chapter gives an overview of two artificial intelligence tools employed in this research: Constraint Logic Programming and Multi-Objective Genetic Algorithms, and introduces their application in planning and scheduling.

Chapter 3. An Ontological Model for Describing Construction Requirements

3.1. Introduction

The objective of this chapter is to present a generalised, flexible and formal representation to define construction requirements. For this purpose, a review of ontological approaches for defining and modelling requirements from mechanical design and software engineering will be conducted. This research will also stress the importance of considering construction requirements in construction planning and scheduling. To better understand the complexities of construction requirements, the evolutionary cycle of project requirements is also discussed, allowing readers to better understand the underlying implications of misrepresented and omitted construction requirements on the construction project.

Further, an ontological model for describing construction requirements will be proposed. This model will define the attributes of the construction requirements ontology, which will aid in presenting a uniform representation mechanism for construction requirements. The use of the representation mechanism for construction requirements will be demonstrated, with particular emphasis placed on representing workspace requirements.

3.2. Review of Ontological Approaches to Define Construction Requirements

Construction requirements represent the key pre-conditions for construction (Chua and Yeoh, 2011). Hence, it is necessary to identify requirements so that feasible construction planning is achieved. Despite this, little attention has been accorded to the impact of construction requirements on project schedules through associated schedule (temporal) constraints. As a result, the literature reviewed in this section will look at the ontological approaches for requirements modelling from the mechanical design domain, as well as ontological approaches to model products and processes within the AEC research community.

3.2.1 Review of Approaches for Requirements Modelling

Traditional modelling frameworks from the domains of software engineering and mechanical design typically segregate the requirements model into functional and non-functional types. Functional requirements describe the capabilities of the system from the perspective of the users of the system (Deng, 2002). Non-functional requirements describe the performance constraints on the system, and include items which limit the capabilities of the system.

In engineering design, functional requirements are captured in the function modelling process to elicit, express and evaluate the design intentions. Reasoning on the engineering design rationale is then used to derive the necessary product features (Chandrasekaran and Josephson, 2000). Moreover, further analysis in the form of identifying functional redundancy can be carried out. This form of analysis exploits the availability of functional redundancies in design components to increase reliability in the overall product (Umeda, *et al.*, 1996). In the context of this research, functional requirements implicitly link the features of the product model to the process model, and thus is an indispensable constituent in the construction requirements ontological model which follows.

From the ontological perspective, the functional requirement may be further analysed by reviewing its function, behaviour and state (or equivalently, structure).

The function refers to the design intentions for the product. The behaviour indicates the inherent properties or characteristics of the product. The behaviour is also implicitly expected to fulfil the function in a successful product. The state is the set of entities, relations and attributes which define the product at a point in time (Umeda, *et al.*, 1990). Gero and Kannengiesser (2004) further this concept by introducing the idea of "Situatedness" to tie the concept of function closer to the design intentions. This ontological model formalises a representation method to describe functional requirements from the intention to the behaviour. This is advantageous for reasoning about conflicts in design, and will subsequently be important in the analysis of construction requirements.

A review of the mechanical design literature demonstrates a trend of focusing on the product model, and its functionality. The non-functional requirements are seldom addressed, nor are the processes deemed impactful on the design. This is not the case for construction. The non-functional construction requirements like information availability may directly affect the completion of the product. Moreover, construction methods, often captured with process models, are key considerations in any construction project and should not be ignored.

In software engineering, Non-functional Requirements Modelling is known to be difficult due to its ambiguity and difficulty in representation. Often there is a lack of consensus regarding the meaning of these requirements. Also, there is no formal method of defining a non-functional requirement universally, and it is implied that these non-functional requirements are highly dependent on the domain (Mylopoulos, *et al.*, 1992). Cysneiros, *et al.* (2001) proposed a lexicographic approach to deal with the representation of non-functional requirements. Their representation approach focused on representing these non-functional requirements as goals to be satisficed in a AND/OR graph. This lexicographic approach is flexible and extendable, depending upon the domain of the model.

Describing and representing construction requirements as compared to manufacturing requirements or software requirements presents an added dimension of difficulty. The approaches reviewed above display a division in the requirements modelling literature between the functional and non-functional approaches. While manufacturing requirements research traditionally focus primarily on the functionality of the product, software requirements research have placed considerably more emphasis on the abstract non-functional aspects. Construction requirements lie at the intersection of the two fields, and hence must address this division. Hence, there is a need to amalgamate both functional and non-functional aspects together for a more complete representation, presenting a generalised formulation which is able to encapsulate both types.

3.2.2 Review of Requirements Analysis in Construction

In the AEC community, the focus of requirements elicitation, modelling and analysis has mainly been upstream, in the form of client requirements. Kamara, *et al.* (2002) introduced the Client Requirements Processing Model (CRPM) to describe and capture client requirements using a basic taxonomy of the client's need, facility process and project characteristics. Mitrovic, *et al.* (1999) also described the importance of analysing client requirements from the perspective of the interfaces between business processes and project processes. Their work lays the foundation for understanding the derivation of construction requirements from the upstream client and design requirements.

Unfortunately, there is little in the AEC research community on representing and analysing construction requirements *per se*. This is probably due to the lack of a formal representation of construction knowledge, which allows the explicit modelling and analysis necessary for various construction disciplines (Fischer, 2006). Song (2006) first proposed the concept of construction requirements as a constructability analysis tool. In particular, he focused on the Intermediate Function Requirement (Intermediate Requirements which support construction of the in-progress facility/intermediate products) where he proposed a knowledge representation methodology to capture these intermediate function requirements for analysis. From the perspective of constructability, some of the construction requirements were established as constructability rules. Fischer (1993) and Ugwu, *et al.* (2005) approached the representation of construction requirements as knowledge-based rules from ontological descriptions of the problem domain. Yurchyshyna and Zarli (2009) incorporate Industry Foundation Classes (IFC) with an ontological approach for depicting construction conformance requirements. Their work implied that an ontological approach should be employed to clearly represent construction requirements.

Other competing construction knowledge representation approaches focused on specific aspects of construction: Product, Process and Resource (Information). These representation approaches attempted to define construction knowledge from the integration of these aspects: Integrated Product-Process Modelling (Yamazaki, 1995, Stumpf, *et al.*, 1996, Bouchlaghem, *et al.*, 2004), Feature-based Process and Product Modelling for Cost Estimation (Staub-French, *et al.*, 2003a, Staub-French, *et al.*, 2003b), Construction Information Flow (Bo-Christer, 1992b, Kartam, *et al.*, 1997), and Construction Inspection Planning (Gordon, *et al.*, 2007).

The result of these approaches was that the knowledge representation was often problem-specific, and not scalable to other types of construction. This inflexibility led to the more recent research efforts directed at using the ontological approach to describe these aspects (El-Diraby and Kashif, 2005, El-Gohary and El-Diraby, 2010). Often, such an endeavour involved defining a set of higher-level core ontology with logical rules to draw additional inferences in these specific aspects. Despite these efforts, the modelling of construction knowledge was still centred about the product, process, and resource domain.

The aforementioned ontology based approaches to describe construction products and processes have generally focused on providing a common representation for the concepts within the construction domain. Ontologies provide a framework for representing, sharing and managing domain knowledge through concept taxonomies and ontological relationships through logical axioms that allow reasoning (El-Diraby, *et al.*, 2005). However, developing the domain taxonomy is both the strength and the limitation of the above approaches. The strength of the domain taxonomy approach lies in its immediate applicability to represent knowledge arising from the domain. The weakness, however is that in focusing on the domain knowledge, it does not provide enough flexibility to model across several interrelated domains, which is often required in construction.

Despite the limitation of the above approaches documented in this section, they lay the foundational ideas for the proposed ontological model of construction requirements. One of the key issues which this research will address is how a Planner or Knowledge engineer can flexibly enhance the taxonomy of construction requirements through identifying the “requirements” of construction requirements, which is termed in this research as the core characteristics. These characteristics will be defined particularly along the spatial and temporal dimensions, influencing the direction of research along the definition of particular workspace requirements, which may arise from several competing domains or construction trades. Hence, constructability issues arising from workspace congestion can be formally captured as construction knowledge.

3.3. Establishing the Importance of Construction Requirements in Construction Planning and Scheduling

This research argues that Construction Requirements should be an important part of the Construction Planning and Scheduling process, and that Construction requirements management and analysis should be carried out at all project levels and phases. In essence, the planning considerations in a construction schedule can be thought of as one form of representing construction requirements, and should have an indispensable role in improving the constructability of a project.

However, the construction community presently lacks the analysis tools for analysing and managing these construction requirements. As shown in the previous section, these tools are focused on the upstream activities for clients, architects and designers. This translates into a lack of detail and transparency in the modelling of construction requirements, which disrupt the transfer of plans from managers to supervisors and from main contractors to their subcontractors. Consequently, this lack also results in disruption of information between different trades. Some trades are thus ignorant of their responsibility to fulfil a construction requirement necessary for other trades. More seriously, misinterpretation of the constraints could amount to rework and contribute additional waste in the project lifecycle.

At the pre-planning and planning phase, construction requirements are the basic components of the oft tacit construction knowledge and experience. Hence, proper, clear and unambiguous elicitation of these construction requirements and its representation as knowledge constructs is necessary to prevent its omission and misinterpretation. Furthermore, this formal documentation process allows construction requirements to be passed on through the project phases, enhancing the traceability of changes for better project management.

Meanwhile, construction plans and schedules during the construction phase should also be examined from the perspective of construction requirements (during internal constructability improvement programs and constructability review procedures) to ensure that the functionalities required for the processes are provided for, preventing costly delays and abortive work. One way in which the construction requirement analysis can impact the construction plan is through the identification of several alternatives for resolving the potential requirement clashes.

In conclusion, construction requirements management should be a key process in the project management lifecycle. This chapter will demonstrate that the consideration of construction requirements is not only important during the pre-planning and planning phases, but is also vital for change management during the construction phases.

3.4. The Evolution of Construction Requirements

Constructability is defined as the optimum use of construction knowledge and experience in conceptual planning and design, engineering and procurement, and field operations to achieve overall project objectives (CII, July 1986). Construction knowledge should thus be incorporated early in a project to maximize its impact. Construction requirements are conditions that support/enable construction forming part of the construction knowledge. The fulfilment of these requirements at various project stages is vital for successful project completion. Hence, it is necessary for proper elicitation and analysis to adequately capture and define the construction knowledge in the form of requirements.

A project's schedule constraints are governed by its requirements. Kamara, *et al.* (2000) differentiated project requirements into three types: Client, Design and Construction. Client requirements refer to the business needs of project stakeholders, while Design requirements include design specifications and regulatory codes of practice. In particular, the construction requirements are the concerns and constraints that must be fulfilled for procurement, construction and logistic processes (Song and Chua, 2006).

The evolution of construction requirements may differ from project to project, with the general progression as shown in Figure 3.1. This typically starts with the elicitation of client requirements as a necessary first step during the conceptual phase, allowing for the definition of the project's value to the client, and the various stakeholders. The client requirements are then translated into design requirements, which are also subjected to design regulatory standards. Violation of design standards may be feedback to review the client's business needs. The design requirements are further translated to construction requirements; one form being shop drawings. At this stage, the initial design requirements may require review or redesign to facilitate practical construction methods, often subjected to site/environmental conditions. Client requirements may also directly impact the construction requirements by contractually specifying deadlines and specific construction methods and/or materials. Inversely, the inability to satisfy construction requirements may also be feedback to the clients, possibly causing a redefinition of the client's business needs. The conventional evolution of requirements is represented using unbroken arrows in Figure 3.1. The construction activities and their corresponding relationships may be inferred from the construction requirements arising from the following perspectives (shown in Figure 3.1): Topological Precedence, IF requirements, Space, Key Resources, Safety, Contracts, Site/Environment and Logistics/Procurement.

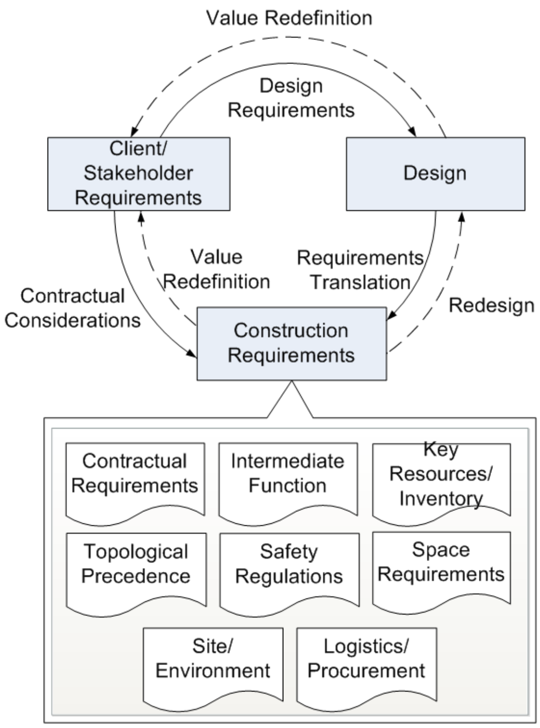


Figure 3.1. Evolution of Requirements

In summary, construction requirements exist as a form of derived requirements and are an abstraction of the client's intentions and design specifications. The evolution process is often improperly tracked, leading to poor traceability of requirement evolution. This could lead to misinterpreted requirements, or even missing construction requirements.

3.5. An Ontological model of Construction Requirements

Ontology is the study of the existence of things. Ontological models largely comprise a taxonomical schema which conceptualizes the terms in the domain, and a library of axioms which establish some facts about the terms in the domain (Yurchyshyna and Zarli, 2009). In this section, the taxonomy for construction requirements will be presented, which will allow new construction requirements to be formally formulated. This enables knowledge capture and sharing between the various trades, and can pre-empt potential interface issues.

In the course of introducing the ontological model, a case example is used to better illustrate the concepts proposed. A simple steel frame is erected as shown in Figure 3.2. Each beam (B1 to B4) and column (C1 to C4) is 3m in length, and held together by welded connections (W1 to W4). To model the changes occurring to the product, the component state concept (Chua and Song, 2003) is used to define the various intermediate stages through which the product undergoes change. The typical component state lifecycle for the column includes Column Erection (C1.S1) → Welding at Column (C1.S2) → Column under Weld Test (C1.S3) → Completed Column (C1.S4). Similarly, a component lifecycle for the beam would include Beam Lifting (B1.S1) → Welding on Beam (B1.S2) → Beam under Weld Test (B1.S3) → Completed Beam (B1.S4). Other components include the truck mounted crane (Crane) with its associated workspace envelope (WS_Crane), and the work spaces related to the welding and erection of the truss (WS1 to WS4) in Figure 3.3. The scaffolds (Scaffold) are not explicitly modelled for clarity.

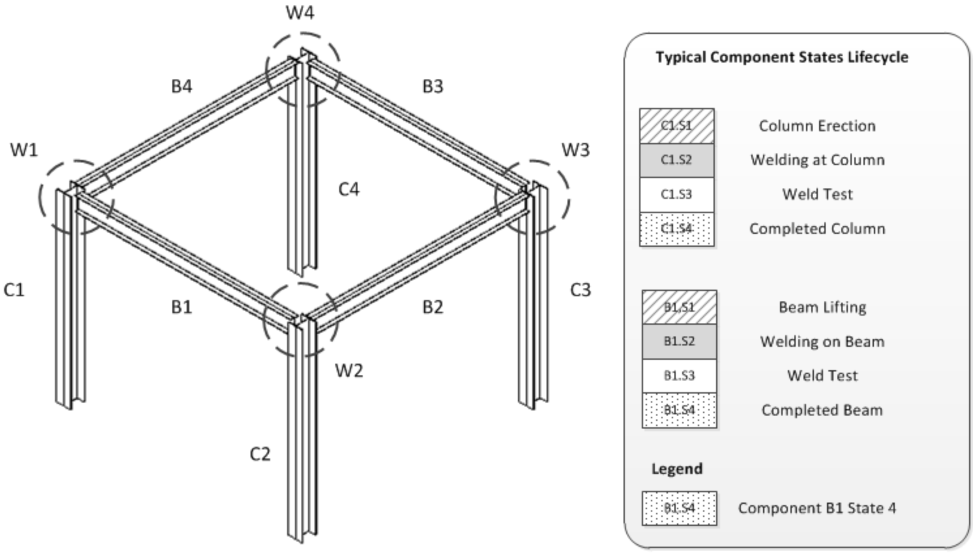


Figure 3.2. Steel Frame Case Example

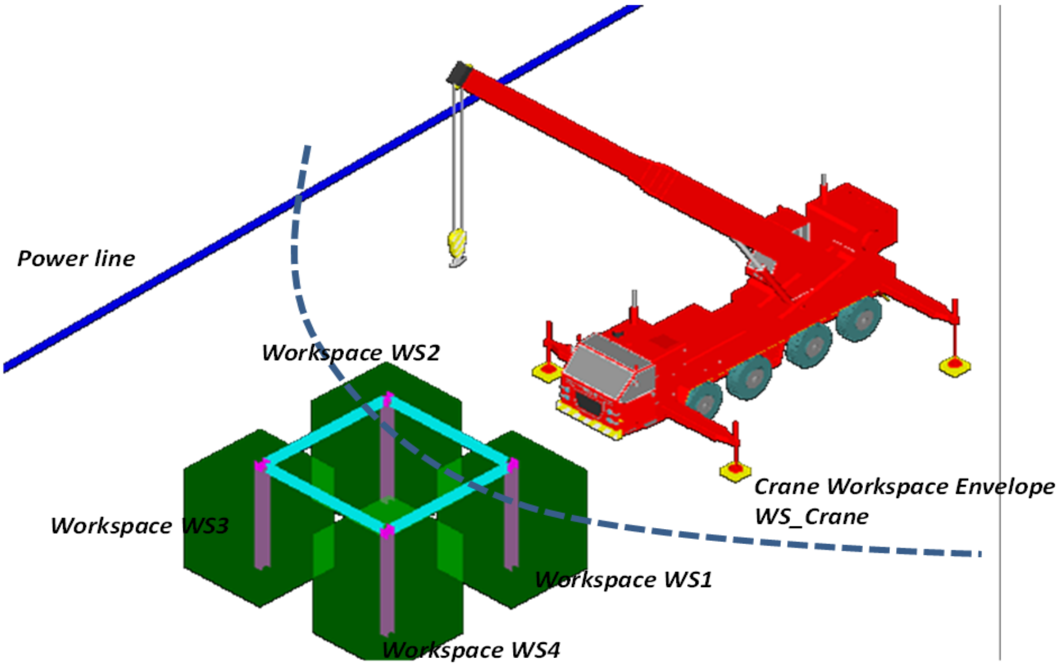


Figure 3.3 3D Perspective of Workspaces in Steel Frame Case Example

One possible construction method for the aforementioned frame is to erect C1 to C4, before erecting scaffolds around each column. The truck mounted crane is used to lift the beams in place before commencing welding at locations W1 to W4. Weld tests (WT_W1 to WT_W4) are conducted on each of the welds before the scaffolds are subsequently removed. Twelve tasks are identified for this construction method as shown in the Gantt Chart of Figure 3.4.

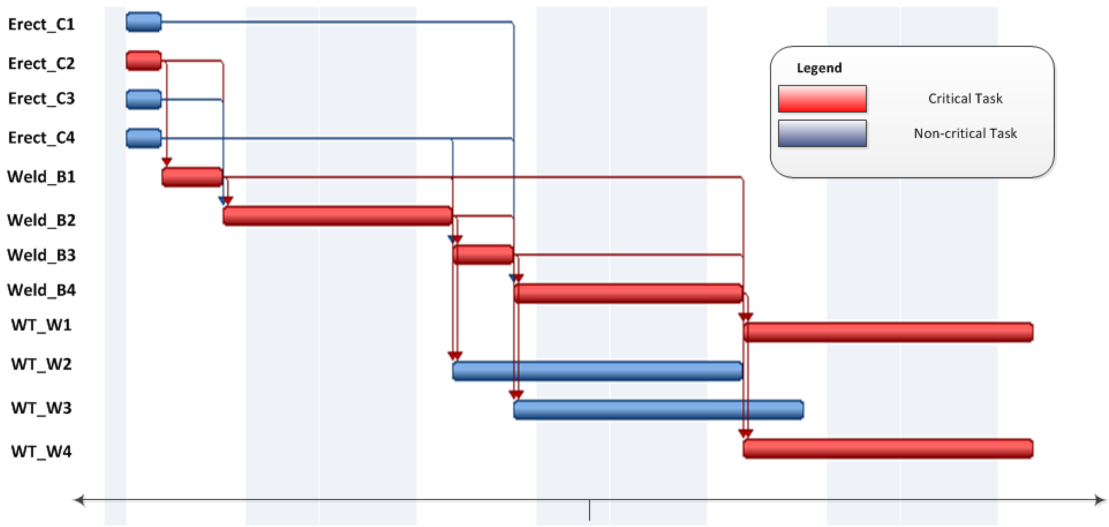


Figure 3.4. Gantt Chart of Steel Frame Case Example

3.5.1 Proposed Approach to Defining Construction Requirements

The proposed approach undertaken in this research lies in defining a general and immutable taxonomical schema to represent the core attributes of the entities which constitute a construction requirement. This taxonomical schema is inspired by the approach adopted in the upper level core ontology, DOLCE (Masolo, *et al.*, 2003). DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) is a foundational ontology, which attempts to address generic domains, allowing it the advantage of generality. This is particularly useful for describing natural language and human cognitive notions, which occur in the Semantic Web. Jureta, *et al.* (2009) demonstrated the use of DOLCE as the underlying framework for describing a core language for representing software requirements. While the approach adopted in this research is similar, the taxonomy developed herein differs significantly by simplifying the varied taxonomical categories introduced in DOLCE to suit the spatial, temporal and abstract nature of construction requirements.

The upper level core ontology is thus important to establish a medium for mutual understanding and interoperability between agents from different domains, and this is especially relevant from the perspective of construction requirements. As construction requirements may be expressed in many forms, from contractual conformance requirements, to site requirements expressed in natural language between contractors, it is vital to enable a common flexible and extendable language for encapsulating the knowledge represented in the requirement. This can be achieved via the proposed taxonomical schema defined on the entities constituting the construction requirement.

The approach adopted (shown in Figure 3.5) starts with the characterisation of a core immutable set of attributes. These core attributes are then instrumental in defining the entities making up the construction requirement, by providing a fundamental basis for describing their key characteristics. Basic entities like “work package” entities and “abstract” entities are introduced, which in turn form the knowledge constructs of a requirement. New taxonomies for construction requirements from different domains may then be built using this schema, allowing a flexibility and generality which is typically not found in current domain-centric ontological frameworks. By treating construction requirements as the fundamental knowledge constructs via this approach, we distinguish this research from other knowledge based approaches, which tend to focus on singular knowledge domains.

Figure 3.5. Approach Adopted for defining Construction Requirements

3.5.2 Core Characteristics of a Construction Requirement Entity

In this research, a construction requirement is defined by its constituent entities. An entity of a construction requirement may be defined by establishing one or more of the following characteristics:

- ***Tangibility***: Tangible entities or Concrete entities are entities which can be perceived, and inherently have spatial attributes. A construction requirement usually requires entities with these attributes to represent spatial relationships arising from the construction product. Conversely, abstract entities are the antithesis of concrete entities, and thus do not have any perceivable spatial attributes.

- ***Measurability***: Measurability refers to the perceptible measure of some entities related to the construction requirement. This measure may refer to concepts like distance, clearance, etc. Other construction requirement entities may not have a measurable attribute, and these could relate to non-measurable concepts like colour or smell.

- ***Perdurant* or *Endurant*:** Perdurance and Endurance describe the temporal behaviour of the entity in a construction requirement. Endurant entities are static at all times, while Perdurant entities are dynamic and may change as time passes. Often perdurant entities change due to the fulfilment or non-fulfilment of other requirements as pre-conditions.

The above establishes a requirement of requirements, and describes the main characteristics of the entities which are of relevance to modelling construction requirements; these characteristics are observed to cover the spatial, temporal and abstract aspects of the construction requirement.

To model these characteristics within the construction requirement entity, three attributes are defined: Spatial, Temporal, and Abstract. Each attribute is representative of the corresponding characteristic of tangibility, perdurance and measurability. The spatial attribute describes as its sub-attributes, the physical geometric attributes, its location and features. Also the type of the entity is introduced to define the space utilization characteristics: Dead space, Interdiction space etc. These utilization characteristics will be further elaborated in Chapter 4.

The temporal characterisation may be modelled using the temporal attribute. A temporal attribute refers to the time in which the entity occurs. The sub-attributes include the start, finish and duration, and are inferred from the associated construction tasks. For example, consider the steel column C1 in the steel truss example where C1.S1 represents the component state of C1 during the erection phase. C1.S1 then starts only when the “Erect_C1” task starts, and terminates when the task ends. The start, finish and duration of the entity may be inferred from the construction task, or its lifecycle in cases where the entity is used to represent a component state (Song and

Chua, 2007). The type of the temporal entity refers to whether the entity is perdurant or endurant, and the temporal behaviour will be reflected in the relationships between the entities, which will be demonstrated in a later section.

The abstract attribute is used to define abstract notions which impact the construction requirement. The abstract attribute may be used to define intangible features like the clearance between objects, weight of loads or cost. It may also be used to define abstractions of key resources. For example, in the steel truss example, the number of cranes could be represented as an abstract attribute, e.g. the value of "5" is stored within the abstract attribute to represent the number of cranes. In this research, only measurable abstract attributes are of importance, and these are used to define various metric quantities like "goal" and "soft goal". A discussion of these quantities will take place in a later section.

3.5.3 Basic Construction Requirements Entities

The prior section has established the characteristics of a construction requirement entity, and proposed three attributes to represent these characteristics within the entity. In general, a construction requirement entity can encompass any combination of these attributes. Of specific interest to this research is the definition of:

1) Work Package Entity,

2) Task Interval Entity,

3) Abstract Entity.

Other types of entities may be defined at the Planner's discretion to describe various aspects of the construction requirement, such as space entities with spatial and abstract attributes only.

3.5.3.1 Work Package Entity

A work package was proposed by Song and Chua (2006) as a construct which served as a link between a task interval and a component state. This research extends that definition by recognizing a work package as a type of construction requirement entity, with spatial, temporal and even abstract dimensions. The following class diagram in Figure 3.6 describes the relationships between the space attributes, temporal attributes and abstract attributes. The work package typically has one space attribute and one temporal attribute. Each space and temporal attribute may also be related to an abstract attribute. An example of an abstract attribute related to a space could be clearance, while an abstract attribute related to time could be resource based, e.g. number of workers. Each attribute has a set of inter-relationships defined as taxonomy.

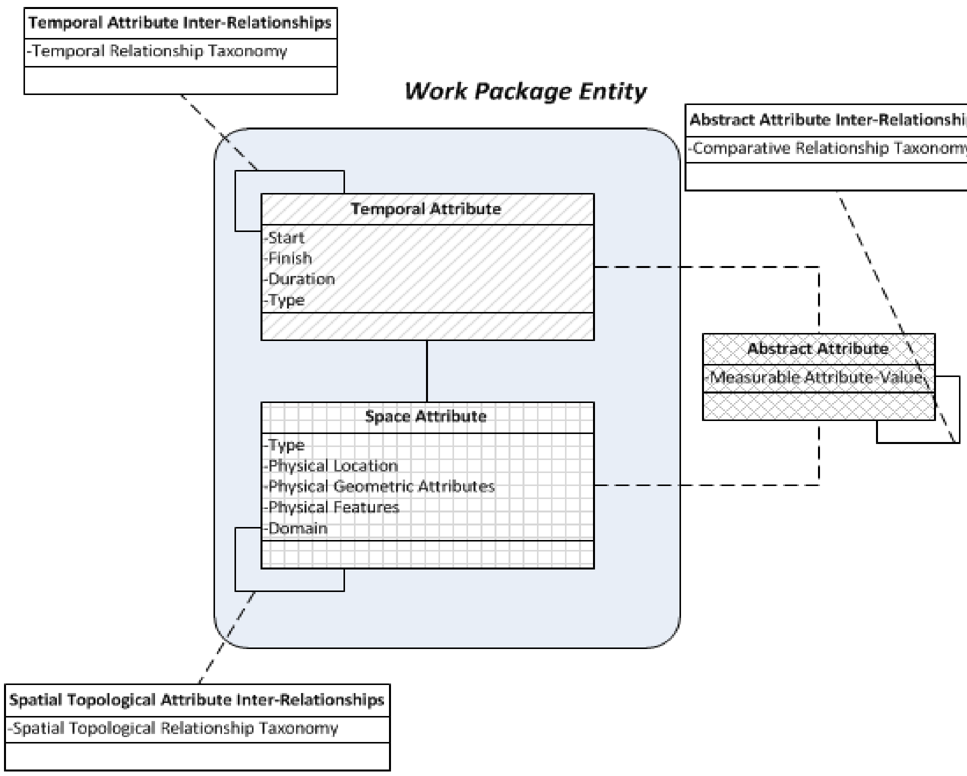


Figure 3.6 Components of the Work Package Entity

3.5.3.2 Task Interval Entity

Within the context of this research, the task interval entity is defined as a construction requirement entity which has only temporal and abstract attributes. It is termed a task interval entity as it is normally used to model the time intervals represented by a task or an activity. Various abstract attributes may be referenced by the task interval entity as well to better define its impact on the construction requirement.

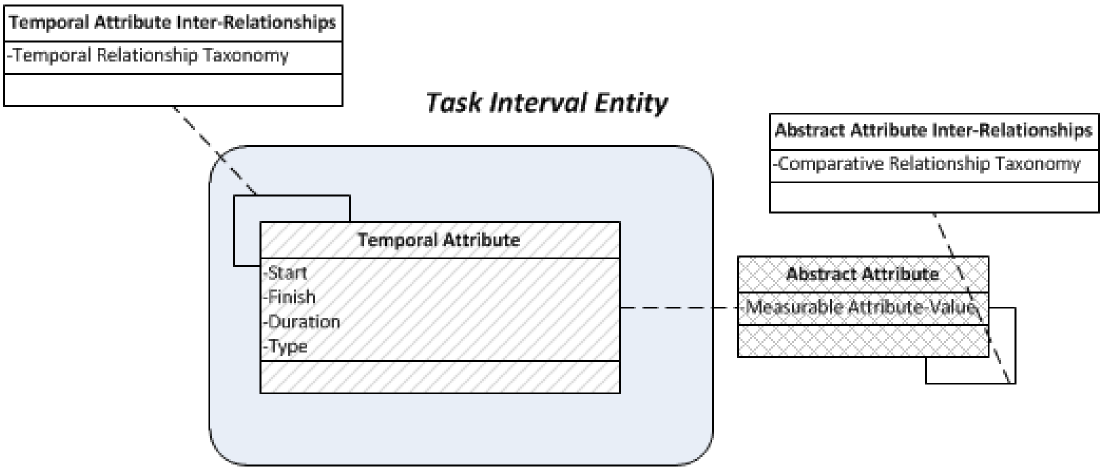


Figure 3.7. Conceptual Schema of Entities in a Requirement

3.5.3.3 Abstract Entity

The final entity proposed in this research, the Abstract Entity has no space or temporal attributes, and exists as a consistent representation of intangible measures for the construction requirement. These abstract entities will be used to define the goals of the construction requirement, and will be demonstrated in the next section.

3.5.3.4 Proposed Representation of a Construction Requirement Entity

The following predicate (Equation 3.1) is used to generically define any construction requirement entity using the three attributes proposed previously:

$$\boldsymbol{entity_E(spatial\ attribute, temporal\ attribute, abstract\ attribute)} \quad \textbf{(3.1)}$$

In particular, the following predicates (Equation 3.2 to 3.4) may be used to define the proposed construction requirement entities defined in this section.

$$work\_package_S(spatial\ attribute, temporal\ attribute, \emptyset) \quad \textbf{(3.2)}$$

$$task\_interval_T(\ \emptyset, temporal\ attribute, abstract\ attribute) \quad \textbf{(3.3)}$$

$$abstract_{AB}(\ \emptyset, \emptyset, abstract\ attribute) \quad \textbf{(3.4)}$$

For example, the workspace needed for workspace (Welding_Workspace) in the case example may be defined as a work package entity using Equation 3.2 as shown in Equation 3.5. Welding_Workspace references the geometric workspaceWS1, and task "Erect_C1".

$$work\_package_{Welding\_Workspace}(WS1, Erect\_C1, \emptyset) \quad \textbf{(3.5)}$$

3.5.4 Inter-Entity Relationships

The construction requirement entities are related to one another through the spatial attribute inter-relationships, temporal attribute inter-relationships and the abstract attribute inter-relationships (as shown in Figure 3.6). The spatial attributes are related by the spatial topological relationships, the temporal attributes by temporal relationships and the abstract attributes by comparative relationships.

3.5.4.1 Spatial Attribute Inter-Relationships

For spatial topological relationships, the taxonomical approach between any two component entities by Nguyen and Oloufa (2001) has been adapted for use in this research. The spatial interactions are classified according to five categories as shown in Figure 3.8: Adjacency, Containment, Separation, Intersection, and Connectivity. One focus of this research is to examine the workspace conflict and congestion on the worksite; hence, one area of spatial interactions is of particular importance: Intersection. Intersection of construction requirement entities may give rise to conflict, and a more detailed account of the taxonomy will be presented in Chapter 4.

For completeness, the other spatial interactions have also been included in the figure to show that the modelling of construction requirements may involve these other spatial relationships. For example, the following spatial relationships could be defined for the steel frame in the case example: “B1 supported_by C1”, or “B2 adjacent to B3”.

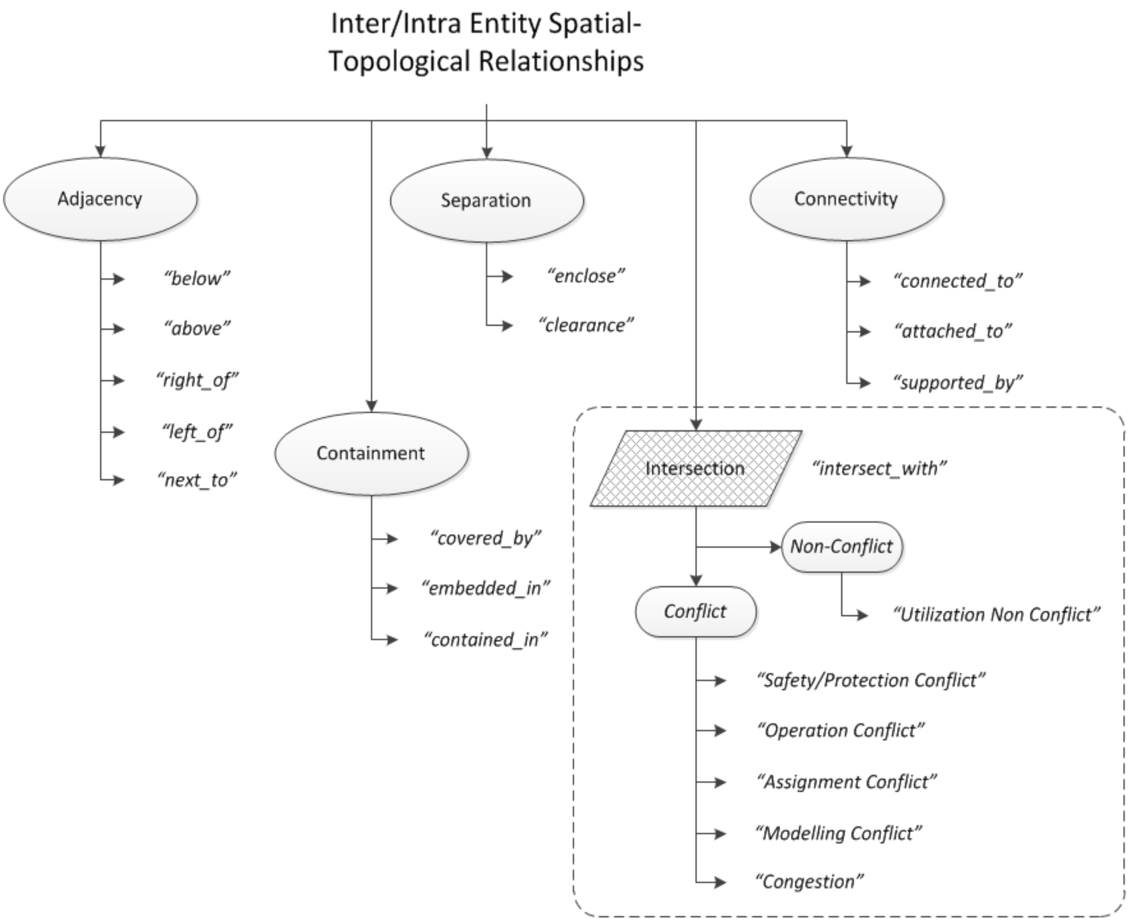


Figure 3.8. Spatial Topological Attribute Relationships

3.5.4.2 Temporal Attribute Inter-Relationships

The temporal attribute inter-relationships are modelled by the temporal relations. These temporal relations capture the process considerations and the activity sequences. This research will establish a case for more detailed modelling of temporal relations in Chapter 5, due to the need for representing construction requirements in construction plans. PDM++ is thus introduced to model some of these complex relationships (Figure 3.9). Further elaboration on the temporal relationships between the temporal dimensions of the entities will be detailed in Chapters 5 and 6.

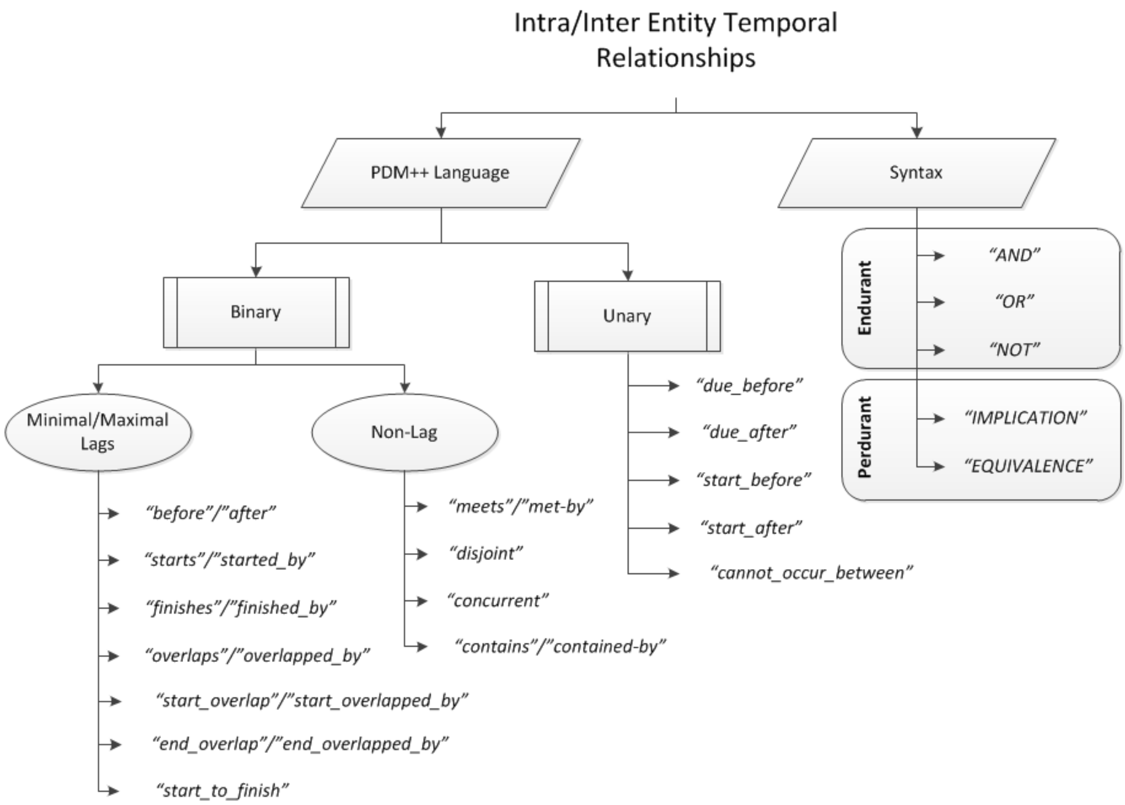


Figure 3.9. Temporal Attribute Relationships

The temporal relationships are defined according to the semantic of the PDM++ modelling language proposed in Chapter 5. To better represent the perdurant and endurant temporal behaviour of the construction requirement, the syntax relationships are introduced to enhance the capabilities of the representation framework to capture the complexities arising from construction requirements. The endurant relations are related to the static construction requirements defined in Chapter 5. Similarly, the perdurant relations are related to the dynamic construction requirements.

Endurant temporal characteristics imply that the need to satisfy static construction requirements remains invariant and necessary. This typically represents the set of construction requirements that must be realised, regardless. These may include safety regulations like "Painting should not be done simultaneously with hotwork". The temporal attributes of static construction requirement entities would be endurant.

In contrast, dynamic construction requirements are those whose need to be satisfied may change due to the inherent interdependencies between the different requirements. This means the existence of dynamic construction requirements is conditional upon the fulfilment of other requirements. Some Intermediate Function (IF) Requirements fall into this classification due to their transient nature. IF requirements are the intermediate functionalities provided by the in-progress facility product for supporting construction (Song and Chua, 2006). One IF requirement which commonly exhibits characteristics of dynamic requirements is temporary works. For example, "It is expedient to carry out wall painting after the adjacent pipes are installed, otherwise, additional temporary protective staging for the finished paintwork must also be provided". Hence, such requirements facilitate the representation of conditional pre-requisites. The temporal attributes of dynamic construction requirement entities would typically be perdurant if its existence is conditional upon other constraints.

3.5.4.3 Abstract Attribute Inter-Relationships

The abstract attribute relationships introduced here will not be covered in detail in this research, but are included for the purpose of completeness of the subject matter. Generally, abstract attribute inter-relationships describe the comparison of the abstract metric entities between one another, and the taxonomy of these relationships is shown in Figure 3.10. These relationships relate an abstract entity with another, and establish a mechanism for comparing between entities.

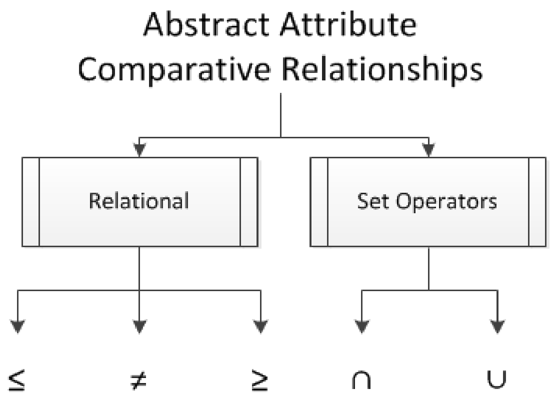


Figure 3.10. Abstract Attribute Comparative Relationships

The importance of the abstract attribute relationships are that it allows a mathematical description of the intangible entities to be established within the construction requirement. The abstract attributes may be used to represent goals of a construction requirement and the comparative relationships will enable the conditions of fulfilling the goal to be stated. For example, a goal of the requirement may be that the weight of beam B1 must not exceed the combined capacity of C1 and C2. Abstract entities may be defined for the weight of B1, the capacity of C1 and C2 as $Weight_{B1}$, $Capacity_{C1}$ and $Capacity_{C2}$ respectively. The fulfilment of this goal may then be stated as:

$$\leq \left(Weight_{B1}, \cup \left(Capacity_{C1}, Capacity_{C2}\right)\right) \quad (3.6)$$

3.5.5 Flexible Construction Requirements Taxonomy

A basic taxonomy for describing construction requirements is now established using the concepts in the prior sections. A construction requirement can be observed to be an interaction between its purpose and operation. Hence, the taxonomy of a construction requirement can be defined as: Purposive, Operational and the necessary conditions defining the interaction between the two. This taxonomy is not meant to be comprehensive, and may be flexibly extended at any time depending on the needs of the Planner.

3.5.5.1 Purposive

The purpose for a construction requirement may be defined as fulfilling the desired intention. This intention takes the form of a function, a goal, or a soft goal. A function is the action of performing an intention to do something. It is possible to infer that the function is physical, and thus directly involves both spatial and temporal dimensions (entities). In the steel frame example, a function, R1 could be “R1: Column C1 Supports Beam B1”. The “Support” indicates a physical function by C1, to be used by B1.

The goal and soft goal describes the aim of a performance of action. In this case, the goal and soft goal are entities with abstract attributes which may or may not involve other work package and task interval entities. More specifically, the goal is considered as an obligation to be satisfied. Soft goals are considered as subjective preferences whose fulfilment is desired, but which may not be achieved. An example of a goal in the steel frame example is “R2: Weld W1 has a labour requirement of 3 men for optimal productivity”. An example of a soft goal is “R3: Provision of aesthetically pleasing hoarding such that on-going construction activities are not apparent to the public”.

The purpose of the requirement is intimately tied to the entities of the system. For functions, the work package entities may be categorized into function users and function providers. Users are the requesters of the function, while providers provide the function. Using the example "R1: C1 supports B1", C1 is the function provider, while B1 is the function user.

For goals and softgoals, such a classification of users and providers is not necessary. However, goals and softgoals may still reference work package or task interval entities for a full description of the requirement. For example, "Provision of aesthetically pleasing hoarding" would entail the hoarding work package entities.

3.5.5.2 Operational

The operational aspect of the requirement depicts the exhibited characteristics of the construction requirement entities. For work package entities, the behaviour captures the manner in which the entities act under specified conditions, circumstances or in relation to other entities. Using the example R1, a useful behaviour of the column C1 in the case example is the material load bearing capacity leading to the structural strength necessary for supporting B1.

For other abstract entities, the operational aspect may be measured using performance metrics. Such performance metrics are usually suggested by the actor (originator) of the requirement.

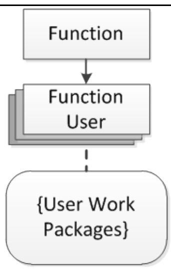
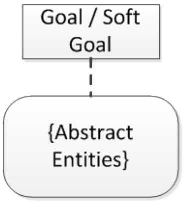
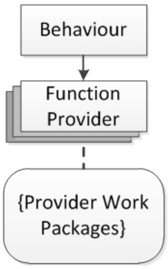
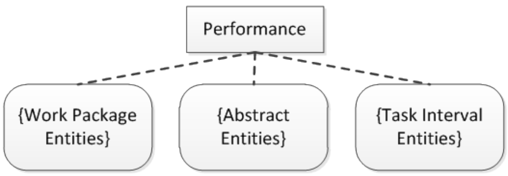
3.5.5.3 Necessary Conditions

The necessary conditions are the conditions which must be fulfilled before the requirement is available for proceeding. These conditions may be modelled using the spatial, temporal and abstract interrelationship taxonomies proposed earlier.

3.5.5.4 Functional and Non-functional Requirements

Extending the above taxonomy to implementation, a construction requirement may be distinguished into functional and non-functional requirements, incorporating the roles defined below (Table 3.1). In general, the functional requirements are the construction intentions for supporting a process or for sustaining the in-progress structure, while the latter refer to performance constraints like capacity and productivity. Other types of requirements are treated as derivatives of these two classes.

Table 3.1. Purposive and Operational Roles in Construction Requirements

	Functional Requirements	***Non-Functional Requirements***
Purposive		
Operational		

Functional requirements are defined as having two sets of work packages: a set of user work packages, and a set of provider work packages. The user work packages define the purpose and demand a function, which has to be fulfilled by the operational behaviour of the provider work packages. To demonstrate the representation of functional construction requirements, an instance of the example functional requirement R1 is modelled using the following schema in Figure 3.11:

Functional Requirement R1:

Purpose: Function: B1 requires Support

Function User: B1

User Work Package: Work_package$_{B1}$(B1, "Weld_B1", Ø)

Operational: Function: C1 provides Support

Function Provider: C1

Provider Work Package: Work_package$_{C1}$(C1, "After Erect_C1", Ø)

Necessary Conditions:

Spatial: Supported_by (B1, C1)

Temporal: before("Erect_ C1", "Weld_B1", 0)

Abstract: Ø

Figure 3.11. Functional Requirement Example

In Figure 3.11, the purposive and operational aspects of the requirement are identified with the user and provider work packages defined. B1 requires a support, and this support is to be provided by C1. The work packages B1 and C1 are identified, with the spatial and temporal attributes shown. The temporal intervals associated with the temporal attributes are denoted in quotation marks. The necessary conditions for the fulfilment of R1 are that the spatial topology between B1 and C1 are met by a "supported_by" relationship, and there is a precedent temporal relationship between the "Erect_C1" and "Weld_B1" relationship.

Non-functional requirements have goals/softgoals which are represented using abstract entities. Performance Metrics are then defined to evaluate the satisfaction of the goals/softgoals. The performance metrics may reference other work package entities, task_interval entities or abstract entities to derive the necessary information. The following Figure 3.12 shows the representation of a client's regulatory conformance requirement R2 which provisions for weld tests on Weld W1 at the intersection of B1 and C1 to conform to the regulatory standards/criteria.

Non-Functional Requirement R2:

Purpose: Conduct Weld Test on W1

Goal: Abstract$_{W1_test}$(Ø, Ø, Weld Hydraulic Test Criteria)

Operational: Weld Strength on W1

Performance: Abstract$_{W1_strength}$(Ø, Ø, Weld Strength)

Work Package: {Work_package$_{B1}$(B1,"WT_W1", Abstract$_{W1_strength}$), Work_package$_{C1}$(C1,"WT_W1", Abstract$_{W1_strength}$)}

Necessary Conditions:

Spatial: Supported_by (B1, C1)

Temporal: Ø

Abstract: ≤ (Abstract$_{W1_test}$, Abstract $_{W1\_strength}$)

Figure 3.12. Non-Functional Requirement Example

In this example, the purpose of the requirement is to fulfil the goal of adequately conducting the weld test on W1. This goal (purpose) is represented by the abstract entity W1_test, which references the weld hydraulic test criteria. The performance (operational) of the requirement is dependent on the exhibited weld strength represented by another abstract entity W1_strength. This abstract entity is inferred from two work package entities B1 and C1. The necessary condition between the goal and the performance is the comparison between the two abstract entities to ensure that W1_test is less than the exhibited strength W1_strength.

3.6. Modelling Various Types of Construction Requirements

This section demonstrates the use of the proposed schema to model two common construction requirements which are important to the context of this research: safety requirements and workspace resource requirements. These were chosen as they reflect the common requirements experienced on the worksite. Also, the workspace resource requirement is representative of the type of spatial temporal problem which this dissertation is addressing.

3.6.1 Safety Construction Requirements

In construction, safety and resource requirements represent some of the more common engineering requisites onsite. As an example some types of safety requirements such as R3: “Truck mounted crane should not operate in a position/location within 3m of a live power line” can be modelled using the ontological model presented as follows (the power line is visible in Figure 3.3).

Safety Requirement R3:

Purpose: Cranes should not be within 3m of power lines

Goal: $Abstract_{Safety\_goal}$(Ø, Ø, Minimum Clearance Distance)

Operational: Minimum clearance of 3m

Performance: $Abstract_{Safety}$(Ø, Ø, Crane Clearance)

Work Package: {$Work\_package_{Crane}$(Crane,”Weld_B1”, $Abstract_{Safety}$),
$Work\_package_{Crane}$(Crane,”Weld_B2”, $Abstract_{Safety}$),
$Work\_package_{Crane}$(Crane,”Weld_B3”, $Abstract_{Safety}$),
$Work\_package_{Crane}$(Crane,”Weld_B4”, $Abstract_{Safety}$)}

Necessary Conditions:

Spatial: Ø

Temporal: Ø

Abstract: ≤ ($Abstract_{Safety\_goal,}$ $Abstract_{Safety}$)

Figure 3.13 Safety Requirement Example

In this safety requirements example (Figure 3.13), four work packages for modelling the crane are identified. Each work package is related to the location of the workspace envelope of the crane, and each has a temporal attribute related to the task in which the crane is employed. The goal (purpose) is represented as an abstract entity to denote the minimum clearance. The performance metric (operational) is another abstract entity created to denote the distance between the crane at its various positions and the power line. The necessary condition for the safety requirement is then the comparison of the safety performance metric in relation to its goal.

Some safety requirements which do not have spatial dimensions may also be similarly represented. For example, a permit-to-work may be needed before the commencement of hotwork (Weld_B1) as shown in Figure 3.14. The goal may be represented as an abstract (Boolean) entity to denote the issuance of the approved permit, while the performance metric is a similar abstract entity denoting the acquirement of the permit. Task interval entities are identified for the tasks of "Approval_permit" and "Weld_B1". The necessary conditions of the safety requirement are then the temporal precedence of "Check_permit" before "Weld_B1", as well as the abstract relationship depicting the matching of the goal and performance metric.

Safety Requirement R3a:

Purpose: Permit-to-work approved before commencement of hotwork

Goal: $Abstract_{Safety\_goal}$(Ø, Ø, Permit Approved)

Operational: Obtain Permit-to-work before commencing hotwork

Performance: $Abstract_{Safety}$(Ø, Ø, Permit Approved)

Work Package: {$Task\_interval_{A}$(Ø,"Weld_B1", $Abstract_{Safety}$),
$Task\_interval_{B}$(Ø,"Approval_Permit", $Abstract_{Safety}$),

Necessary Conditions:

Spatial: Ø

Temporal: Before("Approval_Permit", "Weld_B1")

Abstract: = ($Abstract_{Safety\_goal}$, $Abstract_{Safety}$)

Figure 3.14 Non Spatial Safety Requirement Example

3.6.2 Workspace Resource Requirements

Space requirements for work processes may also be represented for space conflict analysis. Various prior researches have proposed the modelling of spatial requirements to be similar to other resource requirements (Guo, 2002, Dawood and Sriprasert, 2006). Such space requirements may similarly be classified as non-functional requirements with the goal being the fulfilment of the Dynamic Space Indicator (*DSI*) or Congestion Penalty Indicator (*CPI*). These metrics will be elaborated further in the next chapter.

Using the steel truss case example, each workspace WS1 to WS4 has a workspace resource requirement which must not exceed a critical value, *C*. This statement may now be represented as the purposive goal of the requirement. Operationally, the performance measure *CPI* is modelled as an abstract entity. In particular, the work spaces are modelled as work package entities with time and space dimensions. Three such work packages are identified: The Crane work package, the WS1 work package and WS2 work package. The Crane work package references the crane workspace envelope WS_Crane, and occurs during the "Crane Lift" task. WS1 work package and WS2 work package directly reference the Workspaces 1 and 2 during the "Weld_B1" task.

The necessary conditions check that when there is an intersection of the workspaces WS_Crane with either WS1 or WS2, and this intersection takes place with a temporal overlap between the tasks of "Weld_B1" and "Crane Lift", then the safety performance metric defined by *CPI* is less than the work space critical value.

Workspace Requirement R4:

Purpose: Adequate Workspace to be provided during Welding of B1

Goal: $Abstract_{Workspace}$(Ø, Ø, Workspace Critical Value)

Operational: Amount of Workspace Intersections

Performance: $Abstract_{Workspace}$(Ø, Ø, CPI)

Work Package: {$Work\_package_{Crane}$(WS_Crane,"Crane Lift", CPI),
$Work\_package_{WS1}$(WS1,"Weld_B1", CPI),
$Work\_package_{WS2}$(WS2,"Weld_B1", CPI)}

Necessary Conditions:

Spatial: {Intersection(WS_Crane, WS1), Intersection (WS_Crane,WS2)}

Temporal: Overlap("Weld_B1", "Crane Lift")

Abstract: ≤ (CPI, Workspace Critical Value)

Figure 3.15. Workspace Requirement Example

3.7. Concluding Remarks

This chapter has formalised the definition of construction requirements from an ontological perspective. The key advantage of the proposed model is in its ability to flexibly represent various types of construction requirements using a consistent schema.

In summary, three characteristics of the construction requirements were identified which encompassed the spatial, abstract and temporal attributes. This ontological perspective then allowed the basic entities of the construction requirement to be formulated from these characteristics. In this way, the chapter established the construction requirements taxonomy from these characteristics, and demonstrated how various requirements like functional, non-functional, safety and workspace resource can be built using the taxonomy proposed.

The concepts in Chapter 4 and 5 build upon the spatial and temporal characteristics identified as the requirements of construction requirements within this chapter. In particular, the interactions of the spatial attributes of the construction requirement entities lead to various spatial conflicts and congestion, and these will be examined in detail in the next chapter. Also, the temporal attributes of the construction requirement entities exhibit two modes of behaviour which is termed perdurant and endurant. The modelling framework in Chapter 5 deals with this, and proposes logical extensions to handle the identified behaviour of requirements.

Chapter 4. Identification and Quantification of Spatial-Temporal Conflict and Congestion in 4D CAD

4.1. Introduction

This chapter will discuss the analysis of the spatial attribute of construction requirements into the construction plan. From the previous chapter, it was shown that the spatial attributes of entities are important characteristics of the construction requirement representation. Where the previous chapter focused on representing the construction requirement entities, this chapter further focusses on the workspaces which may also be modelled as spatial entities with an added abstract metric quality called a "Utilization Factor". This quantity is important for identifying and quantifying conflict and congestion in 4D CAD.

A review of current space planning and modelling methodologies is conducted to investigate any current gaps in the present techniques. An ontological model of space utilization is proposed in this chapter to better abstract the construction workspace representation in 4D CAD. Based on this model of space utilization, a conflict detection methodology is proposed, and due to its abstract nature, attention will be paid to the congestion phenomenon.

Various measures will be introduced to quantify the amount of space utilization from a spatial demand and supply perspective. From these measures, two indicators are also proposed to quantify this workspace congestion from an activity level, and from a higher level abstraction of time like a time window or a project.

4.2. Review of Spatial Representation and Planning Analysis Methodologies in Construction

A large library on Workspace Conflict Detection using *n*D CAD exists. Various methodologies for modelling space utilization requirements have been proposed for the analysis of spatial conflicts, while the idea of interference between workspaces is fundamental to some of the present literature (Thabet and Beliveau, 1994b, Riley and Sanvido, 1995, Akinci, *et al.*, 2002b, Akinci, *et al.*, 2002c, Guo, 2002). The proposed models provide various means of modelling and analysing spatial requirements. Previous research has recognized the importance of space as a construction resource and has subsequently incorporated it as an integral part of planning constraints (Thabet and Beliveau, 1994b, Zouein and Tommelein, 2001, Winch and North, 2006).

Thabet and Beliveau (1994a) noted that construction sequences were often constrained by the sequential occupation of workspaces. The utilization of space associated with these sequences is then analysed from a comparison of space supply and demand. Winch and North (2006) further refined this idea by defining and analysing the criticality of space in a manner analogous to Critical Path Method.

Akinci, *et al.* (2002c) introduced a taxonomy of space conflicts which correctly defines conflict as a high level knowledge construct, encompassing various forms including congestion, unavailability of access, safety hazards, damage of finished products and design conflict. From this taxonomy, a distinction is made between conflict and congestion, indicating the difference between the two phenomena. This distinction is important from a semantic perspective, as congestion is just one of the many forms of conflict that is evident from spatial construction requirements.

Mallasi (2006) realised that prior research was defined according to two different paradigms: Identifying conflict at an activity level, and identifying conflict at a high-level project scale. A quantification method was consequently proposed, which bridged the two paradigms by assigning user-defined weightages and pegging the value to the Space Criticality concept proposed by Winch and North (2006) .

Guo (2002) analysed spatial conflict and temporal conflict separately, introducing two independent interference indicators called the Interference Space Percentage (ISP) and the Interference Duration Percentage (IDP). Additionally, the spatial requirements of movement paths (pathspaces) for workers, equipment and materials on-site have not been adequately modelled. The inclusion of pathspaces, which are abstracted as pathspace requirements (minimum path height and minimum path width) could facilitate the verification of the availability of access to work faces.

Additionally, other research used graphical methods to explain potential congestions in collided areas, and detection of interferences among trades. Riley and Sanvido (1997) argued that abstracting workspaces in 'solid' CAD models was not truly representative of on-site construction. Instead, they focused on patterns of workflow to characterize their research. In similar vein, Bo-Christer (1992a) described a space-centric abstraction of construction entities, and defined an ontological schema for construction space encompassing spaces, boundaries and enclosing structures. In the proposed space abstraction, the use of imaginary space boundaries interacting with physical space boundaries is included to describe construction workspaces. (Ekholm and Fridqvist, 2000) formalise this idea by introducing a spatial perspective of the construction workspace where the spatial attributes of the process user is modelled within the imaginary boundary of the construction workspace. Maher, *et al.* (1997) tie the abstraction of space entities closer to the requirements of activities, by introducing an Activity/Space Model to provide meaning to the spatial envelope associated with the activity. To this end, the term "soft spaces" is introduced to incorporate the idea that the activity's spatial envelopes may overlap within a common space.

The above methodologies aid the visualisation of space utilization among the different construction trades. Such visualisation helps engineers to identify possible conflicts (and congestion) arising from the detected space collisions. Most of these approaches are applied to the analysis process through the use of discrete event analysis. This means that discrete windows of time are analysed independently for spatial conflicts. A discrete methodology may lead to granularity issues, which means that conflicts which have a short time frame, may not be detected if the time window of analysis is too large. Additionally, due to the unnecessary constraint implied by the temporal borders of time frames, this implies that work operators do not have the necessary flexibility to repackage their work in the future knowledge that another trade will be impinging on their workspace.

In summary, the different approaches adopted in previous studies were defined on three levels: Project, Activity and Operative. The project-level analysis looked at conflict on the construction schedule (Thabet and Beliveau, 1994b, Winch and North, 2006), while the activity-level analysis evaluated conflict through pair-wise comparisons of activity processes in a 3D CAD model (Akinci, *et al.*, 2002a, Akinci, *et al.*, 2002c, Guo, 2002). Finally, the operative-level approach studied the movement and workflows of individual workspace users, who are commonly the construction operators (Riley and Sanvido, 1995, Riley and Sanvido, 1997). The current research seeks to abstract space utilization from the operative-level perspective for space planning at activity and project levels.

4.3. Modelling Methodology and Conflict Detection for Spatial Attributes of Construction Requirements

The modelling methodology used is this research is based on an ontological description of the actual utilization of universal site space. The universal site space comprises the entire space in a construction site that is relevant for the modelling whether empty or utilized by construction products, processes or resources. The ontological representation serves to define the nature of worksite conflicts and forms the basis for quantifying spatiotemporal congestion or conflict of a schedule. The ontological constructs are the space elements within the universal site space which can be mapped onto 3D CAD/BIM elements, referred to in this chapter as 'Space Entities'. These space entities are the embodiment of the concrete spatial attributes which is a key element of a construction requirement. The benefit of explicitly modelling the concrete perspective of the construction requirement lies in being able to infer conflicts affecting the construction process by distinguishing the "state of utilization" of the space entity.

The proposed ontological model is termed the Space Utilization Hierarchy Model (shown in Figure 4.1) and is presented as a binary tree, with the space element types as leaf nodes. The space element types are characterized in terms of spatial utilization from two perspectives: usability and operator accessibility. From the perspective of usability, Interdiction Spaces are spaces where no product, process or resource is allowed to occupy, and typically specified for reasons of hazards or protection. On the other hand, Usable Spaces can be further characterized from the perspective of operator accessibility. Dead Spaces are generally occupied by a "permanent" physical product component such as slabs and walls, whereas Accessible Spaces are transiently occupied, often depicting human or operator occupation. A further distinction of Accessible Spaces between Activity Workspaces and Pathspaces is made. Workspaces are defined as space entities where processes are carried out, and are typically adjacent to workfaces, while Pathspaces are defined as entities where movement of workers, equipment and/or physical materials from an initial designated origin to the final destination takes place.

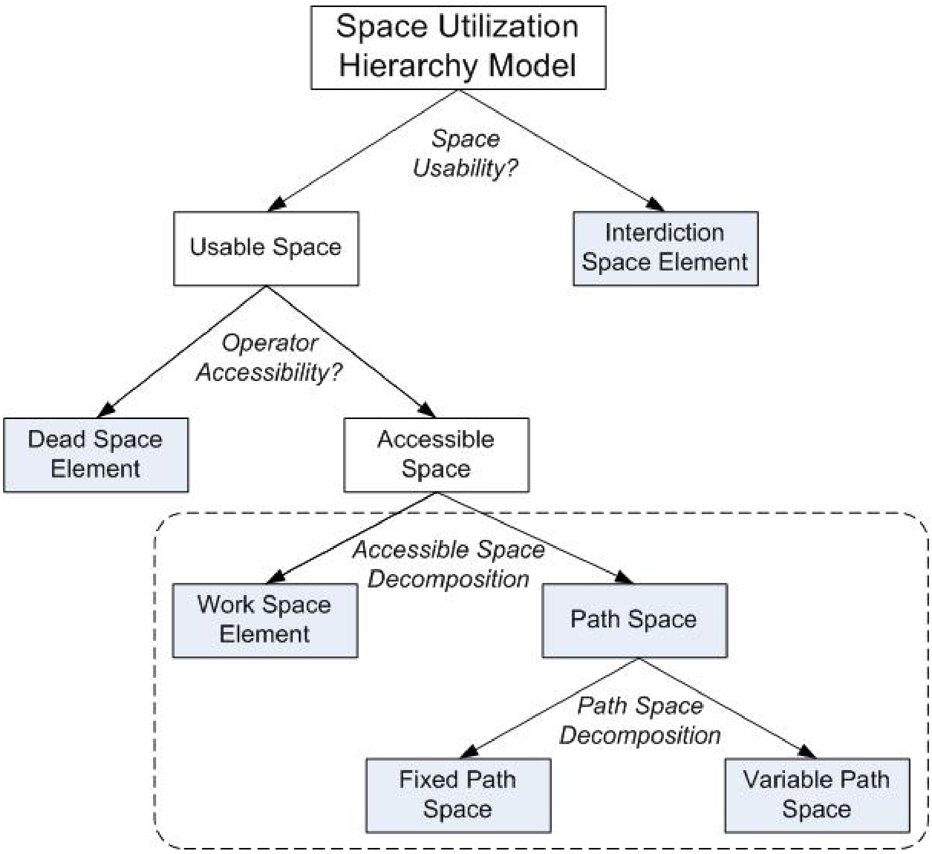


Figure 4.1. Space Utilization Hierarchy Model

The current research explicitly models pathspaces as space entities, rather than abstract requirements as proposed by Guo (2002). This research further distinguishes the pathspace entity into two types: Fixed Pathspaces and Variable Pathspaces. Fixed Pathspaces may be prescribed for resources of certain characteristics which will require confined routes. Variable Pathspaces represent Pathspace entities that define various permissible routes for movement. The spatial modelling of Variable Pathspaces is then the union over the boundaries of all possible Pathspace entities. The availability of multiple paths lessens the impact of interference on the encroached path entities.

A taxonomy based on pairwise comparison between different space elements of the Space Utilization Hierarchy Model is presented in Figure 4.2 to distinguish conflict and congestion scenarios. The spatial demand derived from multiple project perspectives of product, process and resource, gives rise to the above space entities depicted in Figure 4.2. Products and resource holding areas are generally characterized by a dead space representing its existence in the universal site space and an interdiction space for protection. On the other hand, processes are generally characterized by workspaces, pathspaces and corresponding interdiction spaces for safety. Unlike previous models which distinguished conflicts from a functional and/or semantic perspective (Akinci, *et al.*, 2002b), this research defines the taxonomy for conflict from the perspective of actual space utilization.

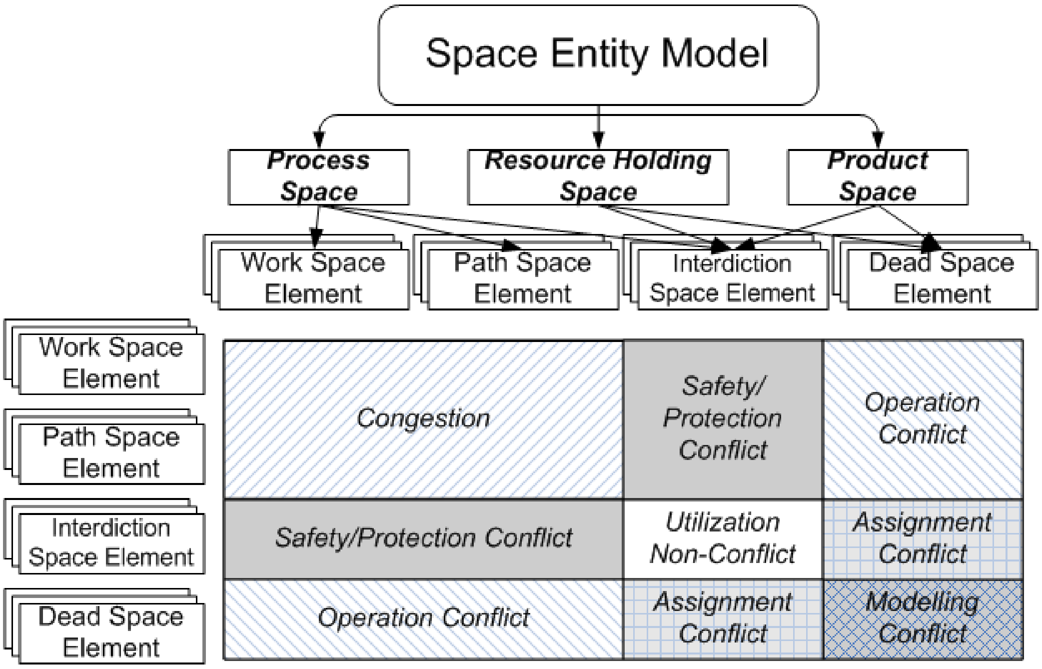


Figure 4.2. Detection of Conflict and Congestion

The interactions between different space entities yield potential conflicts on the worksite. Five forms of conflict are identified in this taxonomy. Modelling Conflict exists in Dead Space-to-Dead Space interactions, depicting the scenario where physical construction products and/or resources occupy the same time and space. Safety/Protection Conflict exists in interactions between Interdiction-to-Workspace and Interdiction-to-Pathspace. Such interactions describe situations where operators enter hazardous or protected zones. Operation Conflict is the scenario where work operators are obstructed due to the presence of interfering construction products or resources. Assignment Conflict occurs when assigned Interdiction Spaces interact with each other or with Dead Spaces. Such interactions may not give rise to actual real-world conflict, for example, two protection spaces may overlap without causing any potential hazard or damage, but two safety spaces could overlap to present a potential hazard. Lastly, Congestion, as a form of conflict, is observed only between various work performers through the interactions of pathspaces and workspaces.

Utilization Non-conflict occurs when interactions between interdiction spaces occur. These interactions do not represent a conflict as both do not have space entities within. For example, two work protection spaces may co-exist within the same space without interfering with each other.

From the above discussion, the first four forms of conflicts identified can be evaluated immediately through inspection of the spatial entities. An overlap of spaces indicates a conflict. However, with congestion there is a degree of crowding which may not necessarily constitute a conflict and is often the most difficult to detect; the methodology is developed in the remaining sections of the chapter.

4.4. A Quantitative Model of Congestion

The above discussion illustrates that models to analyse conflict and congestion can be formulated with the inception of the idea of utilization. In fact, this research defines utilization from a perspective of space and time or space-time-volume, and proposes a framework for quantifying worksite congestion from this concept of utilization.

4.4.1 Quantification of Utilization by a Space Entity

Present methods rely on visualisation of changes to construction sequences using 4D CAD, which relies on the experience of Planners to elicit conflicts. The proposed indicators in this section complement this visualisation aspect of 4D CAD, by allowing the construction sequencing and its corresponding activities to be quantitatively elicited from construction requirement and subsequently analysed.

The concept of utilization can simply be described as a measure of how much a resource is put in use through the concept of space demand and supply so that worksite conflict is evaluated as a function of such space economics. The operative-level utilization underlies one main thrust of this research, and provides a vital link, bridging operation space with activity space.

Understanding the concept of operative-level utilization provides a richer and hence, more accurate depiction of congestion and conflict. For example, consider two operators, *A* and *B* occupying two separate workspaces which overlap each other. Figure 4.3 shows a simplified pictorial summary of three separate scenarios involving *A* and *B*. Scenario 1 shows the case of little overlap (interference) of workspace, with little space utilization by the operators. The operators could easily work around to ensure that they do not simultaneously occupy the "interference regions". However, for the same interference, a higher utilization of the workspaces would result in severe congestion, as illustrated in Scenario 2. Conversely, where there is a high degree of overlap in workspace, a low utilization can still create a situation of low congestion, as depicted in Scenario 3. The above example emphasizes the importance of considering both aspects of workspace interference and space utilization (at operative level) in analysing congestion.

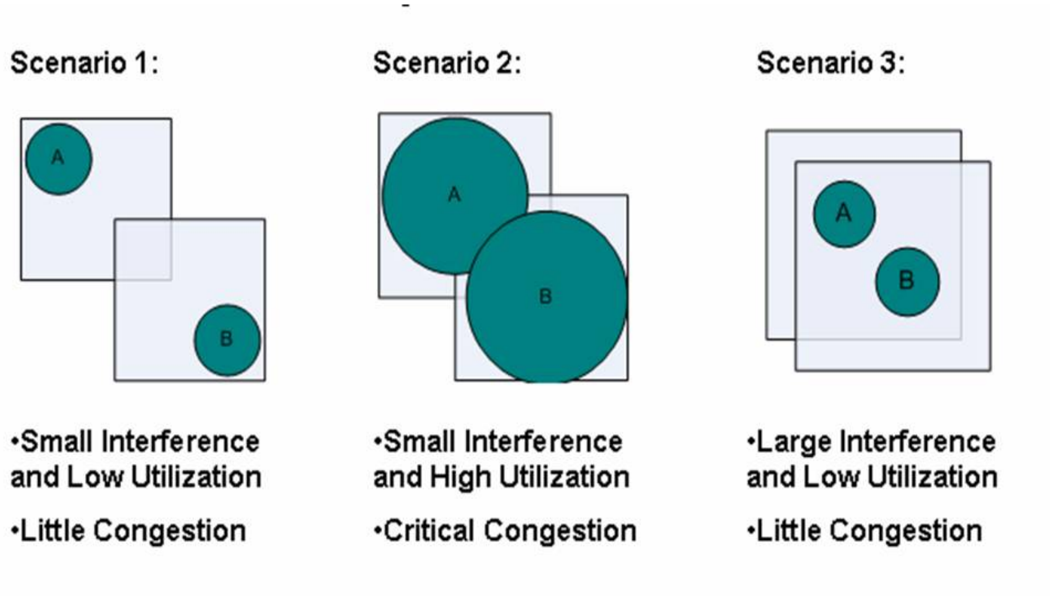


Figure 4.3. Relationship between Utilization and Spatial Interference

A new abstract metric attribute, Utilization Factor, ρ is introduced which quantitatively measures the level of usage for a given space entity from two perspectives: spatial and temporal. Spatial Utilization, U_S is the ratio index of the space required by the operator/equipment to the total available space allocated to an activity; the Operator Space being the amount of space necessary for the operator to perform the activity. Multiple crews may be considered by summing up the total operator spaces needed. The Total Boundary Space refers to the amount of space depicting the activity space. U_S is the intensity of a space imposed by an activity determined as follows:

$$U_S = \frac{\sum \text{Operator Space}}{\text{Total Boundary Space}} \quad \textbf{(4.1)}$$

Spatial Utilization can be conceived from two perspectives. Firstly, U_s can be considered as the probability of finding the construction operator entity in the entire workspace or path space. Hence, the greater the Utilization factor, the greater is the probability of encountering the operator. Secondly, U_s can be described from the perspective of space economics. Here, the Operator Space is the demand on the space entity, while the Total Boundary Space is the supply available. Hence U_s is the ratio of the space demand to the space supply. These two perspectives make Spatial Utilization an intuitive measure of the spatial requirements of an activity, allowing the effect of increasing crew sizes within a single activity to be modelled.

In effect, the concept of spatial utilization can also be extended to the other space entities introduced in the Space Utilization Hierarchy Model. Since a physical product or resource can be expected to fully occupy its allocated space, we can assign a value of 1 to the U_s of Dead Spaces. Similarly, by definition, a hazard or protection space is not expected to have any occupation, so that $U_S = 0$ is used for Interdiction spaces. By definition, Variable Pathspace entities are aggregations of all possible paths so that U_s is based on the union of the total boundary spaces of all possible paths.

Temporal Utilization, U_t recognizes that space entities may not always be utilized throughout the activity's operation time and may be used to describe the intermittent nature of continuous activities. This is especially evident in pathspace entities where the actual usage (utilization) of the space is a fraction of the activity's duration. The temporal utilization may then be expressed as a ratio depicted in Equation 4.2. If time is considered as a resource, temporal utilization may be viewed from an economic perspective of time required (or temporal demand) by the operator and the time available (or temporal supply).

$$U_S = \frac{\text{Actual Time Utilized}}{\text{Total time of activity operation}} \tag{4.2}$$

The resultant Utilization Factor (ρ) is then defined as the geometric mean of both U_s and U_t which provides a representation of the consequences of spatial and temporal demands as it depicts the “average” product of the two utilization factors, and given by

$$\rho = \sqrt[a+b]{U_s^a \times U_t^b} \tag{4.3}$$

where *a* and *b* are user-defined weights, which allow for unequal emphasis to be allotted to either the spatial or temporal utilization of a single entity. This unequal emphasis could arise from the Planner's judgment/priorities. The weights *a* and *b* are meant to allow a "choice" mechanism between the spatial or temporal utilization. For example, an activity may have low spatial utilization, but high temporal utilization. Moreover, this activity lies along the critical path, making it an important activity to manage. The planner may decide that the utilization value is too low, and unreflective of his assessment of the spatial-temporal demand-supply. He may then decide to penalise the spatial utilization by increasing the weight *b*. A discussion regarding the selection of values for *a* and *b*, and its subsequent effects on ρ will be carried out in Appendix A.

The mathematical definition of U_s and U_t causes ρ to be bounded between 0 and 1. It follows that space entities with $\rho = 1$ are fully utilized in terms of both time and space. Economically, the spatiotemporal supply is fully taken up by the spatiotemporal demand. Space entities with $\rho = 0$ are unutilized, as no demand exists.

Quantifying utilization is necessary for the study of worksite conflict and congestion as Utilization provides a low-level abstraction of space demand and supply from the operative level perspective. It provides a value to aggregate and quantify workflow patterns so that it may be incorporated into high-level space planning. More uniquely, ρ implicitly considers both spatial and temporal perspectives in a single ratio.

4.4.2 Quantifying Spatial-Temporal Interference of Functional Requirements

Worksite conflict and congestion occur due to the interferences between space entities. This section extends the concept of utilization to that of activity workspace interference, and quantifies the effects of the interferences from the utilization viewpoint. This will result in an index useful for decision making, allowing project managers to identify congested workspaces.

An index measure called "Dynamic Space Interference" (*DSI*) is introduced here which quantifies the utilization when interference with other activities is experienced. The measure characterizes the obstruction to the ability to work around time and space constraints imposed by other activities when interference occurs. Another way of conceptualizing *DSI* is the measure of the extent that a work operator can accommodate the interferences due to other activity workspaces. Equation 4.4 formulates the *DSI* for the primary space entity *A*, where ρ_i is the Utilization Factor of *i* which is an element of a set of interfering space entities, S_{iA} the overlapping volume between *A* and *i*, S_A the spatial volume of *A*, t_{iA} the time interval over which *A* and *i* overlap and t_A the activity duration of *A*.

$$DSI_A = \rho_A + \sum_i \left(\rho_i \cdot \frac{S_{iA}}{S_A} \cdot \frac{t_{iA}}{t_A} \right) \qquad \forall i \in \text{Interfering Space Entities} \tag{4.4}$$

DSI_A comprises the utilization of the primary space entity (ρ_A) and an increment component which is a function of the utilization of interfering activity space entities *i*, ρ_i, as well as the spatial and temporal infringement into the primary space entity given by $\frac{s_{iA}}{s_A}$ and $\frac{t_{iA}}{t_A}$, respectively. DSI_A can be abstracted as a space-time-volume of space entity *A* with an inherent spatiotemporal demand-supply ratio (ρ_A). When an infringement occurs, there is an added demand on the same spatiotemporal supply imposed by the interfering entities given by the second term in the equation. Detailed derivation is provided in the next section together with an example of its use for multiple space interference.

DSI has no upper bound. However, from the semantic understanding of utilization, *DSI* values greater than 1 indicate that the space-time demand has exceeded its supply, and that worksite conflict has occurred. An important implication is that while the utilization of the primary activity (ρ_A) is low, the additional demands placed on the space by other interfering activities may cause the activity to experience worksite congestion. At the operative level, the operators of interfering space entities can be expected to accommodate each other's spatial and temporal demands on the same space, reaching a compromise through `local scheduling' to prevent incursions. From the perspective of space-time economics, a higher DSI_A indicates that *A*'s ability to perform such local scheduling becomes increasingly difficult.

In summary, *DSI* implicitly accounts for overlaps of multiple spaces. Moreover, it captures the idea that the amount of work done can be redistributed `locally' when interferences occur. By basing its foundation on the concept of utilization, graphical methods developed (Riley and Sanvido, 1995, Riley and Sanvido, 1997) through the considerations of workflow can now be aggregated and represented as a quantifiable variable. In essence, *DSI* offers a measure of utilization which serves to bridge the operator's space requirements with the activity's workspaces.

4.4.3 Deriving DSI from multiple spatial interferences

The derivation of *DSI* is based upon the abstraction of a space-time-volume of a space entity (in this case, *A*), with an inherent spatiotemporal utilization, ρ_A. A space-time-volume is defined as the product of space and time, such that time is treated as a spatial dimension. Imagine another space-time-volume from a set of interfering space entities, *i* with ρ_i infringing upon the space-time-volume of space entity *A*, as shown in Figure 4.4. This figure simplifies the explanation by considering only 2D abstractions, but 3D abstractions can be extended. The overlapping portion with Space S_{iA} and Time t_{iA} experiences two values of utilization, ρ_A and ρ_i. The demand from the additional utilization is then evenly distributed as shown in Figure 4.4.

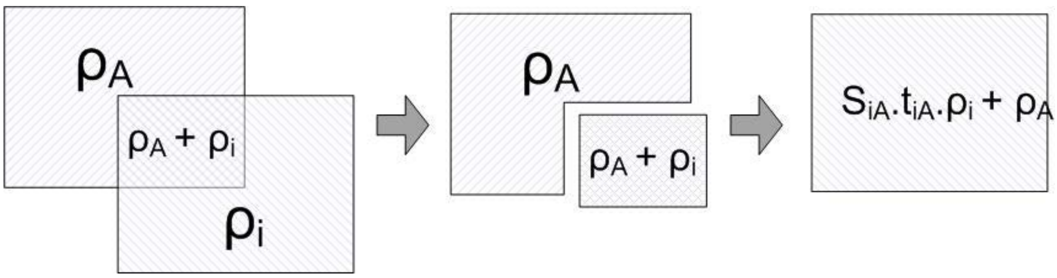


Figure 4.4. *DSI*$_A$ for Overlapping Entities

The additional demand from an interfering entity is given by:

$$S_{iA} \cdot t_{iA} \cdot \rho_i \qquad (4.5)$$

Hence, the total demand from all the interfering activities on the space entity *A* is given by:

$$\rho_A \cdot S_A \cdot t_A + \sum_i \rho_i \cdot S_{iA} \cdot t_{iA} \tag{4.6}$$

Therefore, for a given spatiotemporal supply $S_{iA} \cdot t_{iA} \cdot \rho_i$, the final form of the equation for DSI_A as in Equation 4 may be derived as follows:

$$\begin{aligned} DSI &= \frac{\text{Total Spatiotemporal Demand}}{\text{Spatiotemporal Supply}} \\ &= \frac{\rho_A \cdot S_A \cdot t_A + \sum_i (\rho_i \cdot S_{iA} \cdot t_{iA})}{S_A \cdot t_A} \\ &= \rho_\text{A} + \sum_i \left(\rho_i \cdot \frac{S_{iA}}{S_A} \cdot \frac{t_{iA}}{t_A} \right) \end{aligned} \tag{4.7}$$

Figure 4.5 demonstrates the above concept through a simplified 2D abstraction of multiple workspaces overlapping one another. Here, activity workspaces *A*, *B* and *C* overlap one another, and the overlapping areas S_{AB}, S_{AC} and S_{BC} shown indicate the interferences between *A*-*B*, *A*-*C*, and finally *B*-*C* respectively. Figure 4.6 shows the same activities on a Gantt Chart illustrating the temporal overlaps of the activities. Five temporally overlapped intervals are identified as shown in Figure 4.6, each representing a discrete time interval with a homogeneous activity configuration. Using Activity *A* with volume, S_A and duration, Dur_A in Figure 4.5 as an example, DSI_A can be evaluated using Equation 4.8 as follows:

$$DSI_A = \rho_A + \rho_B\left(\frac{S_{AB}}{S_A} \times \frac{T_{ABC}}{Dur_A}\right) + \rho_C\left(\frac{S_{AC}}{S_A} \times \frac{T_{AC}+T_{ABC}}{Dur_A}\right) \tag{4.8}$$

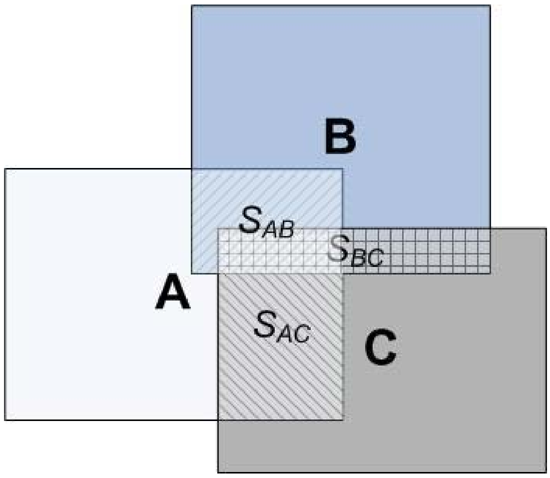


Figure 4.5. Spatial Illustration of DSI$_A$ for Multiple Overlapping Entities

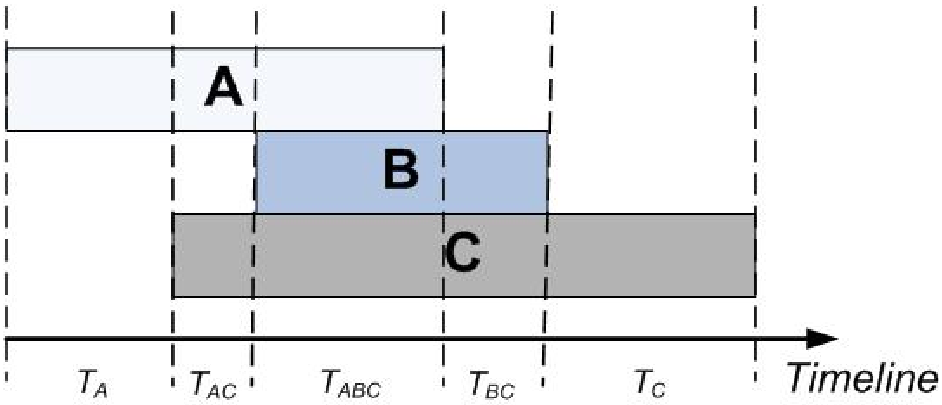


Figure 4.6. Gantt Chart Representation of Temporal Overlapping of Multiple Entities for Illustration of DSI_A

4.5. Spatial-Temporal Decision Making

4.5.1 Need for a High-level Indicator

The evaluation using *DSI* would lead to two outcomes for a schedule: “Feasible” or “Infeasible”. An Infeasible schedule indicates that some activities have *DSI* values more than 1, indicating that the activity's space demands exceed the supply available. This can consequently be identified as worksite conflict, and resolution through re-sequencing of activities may be necessary. A Feasible schedule is one where all the activities are not congested, with respective *DSI* values of less than 1. Here, the value of 1 is chosen as a convenience to represent an upper boundary or critical value. Other *DSI* values may also be chosen at the Planner's discretion to specify meaningful critical values, which indicate the level of utilization that constitutes worksite conflict.

A high-level indicator (Congestion Penalty Indicator, *CPI*) is devised to allow different feasible project schedules or critical time windows to be evaluated, analysed and compared. The indicator maps the *DSI* activity values generated earlier to a piecewise “disutility” scale. Equation 4.9 represents the *CPI* for space entity *A* where the congestion tolerance factor, α denotes the Planner's tolerance to worksite congestion. The function establishes two reference points at $DSI = C$, where C indicates the user-determined critical value and $DSI = 0$. The first reference point refers to the point of critical utilization of the space entity, while the second refers to the point of no utilization. Since the point of critical utilization can be deemed ‘Infeasible’, the value of *CPI* at $DSI = C$ and larger are evaluated as ∞.

$$CPI_A = \begin{cases} \dfrac{1 - \exp^{\frac{\mathrm{DSI_A} \cdot \alpha}{C}}}{1 - \exp^{\alpha}} & \text{if } \mathrm{DSI_A} < C \\ \infty & \text{otherwise} \end{cases} \qquad (4.9)$$

The motivation for the exponential function given in Equation 4.9 stems from needing a monotonic function between the *DSI* interval of *0* to critical value *C* to ensure a unique solution to the certainty equivalent of the preference trade-off problem shown later in Figure 4.7. Also, it is expected that the Planner will be "congestion" averse. Moreover, it is further assumed that the Planner will have constant absolute "congestion" aversion, which means that his/her assessment of the impact of congestion is independent of the pre-existing congestion already present. To fulfil these two criteria, the exponential utility function is modified to that of Equation 4.9 and used.

The composite congestion indicator CPI_{Total}, is then formulated as the sum total of all the *CPI* values of the activity space entities in the critical time window, as shown in Equation 4.10.

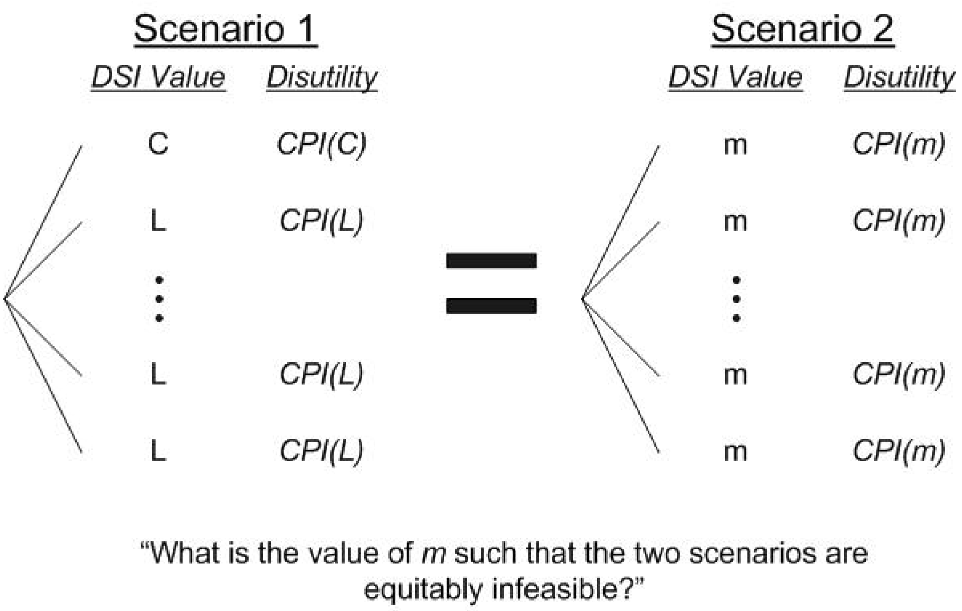
$$CPI_{Total} = \sum_{i \in N}^{N} CPI_i \qquad \text{where N is the set of interacting space entities} \qquad \textbf{(4.10)}$$

Hence, the schedule with lower congestion potential will be denoted by a lower composite CPI_{Total} value, representing a sense of the impact of activity congestion.

4.5.2 Eliciting the Planner's Congestion Tolerance

The value of congestion tolerance factor, α in Equation 4.9 may be elicited from the Planner in a similar manner employed in Utility Theory, by determining the trade-off between the Planner's preferences for two given scenarios. One scenario has a single activity with the critical *DSI* value *C*, while its interacting activities have low

DSI values, *L*. The other scenario has all interacting activities with a moderate *DSI* value of *m* which is involved in the trade-off. Since *C* is the Planner's critical utilization level, Scenario 1 is the lower bound of “infeasibility”. The trade-off question would then be: “What is value of *m* such that the two scenarios are equitably infeasible?” as depicted pictorially in Figure 4.7. *m* is thus the acceptable limit for which the Scenario 2 becomes infeasible like the first.



$$1+(N-1)\times\frac{1-\exp^{L\cdot\alpha/C}}{1-\exp^{\alpha}}=N\times\frac{1-\exp^{m\cdot\alpha/C}}{1-\exp^{\alpha}}$$

Figure 4.7. Preference Trade-off for Eliciting *CPI*

Equating the CPI_{Total} of both scenarios as shown in Figure 4.7, and rearranging the terms yield Equation 4.11 from which the Planner's congestion tolerance factor α, and thus the Planner’s inherent attitude towards congestion, can be determined.

$$\frac{1}{\exp^{\alpha}-1}\left[\exp^{\alpha}+(N-1)\exp^{\frac{L\alpha}{C}}-N\exp^{\frac{m\alpha}{C}}\right]=0 \qquad \textbf{(4.11)}$$

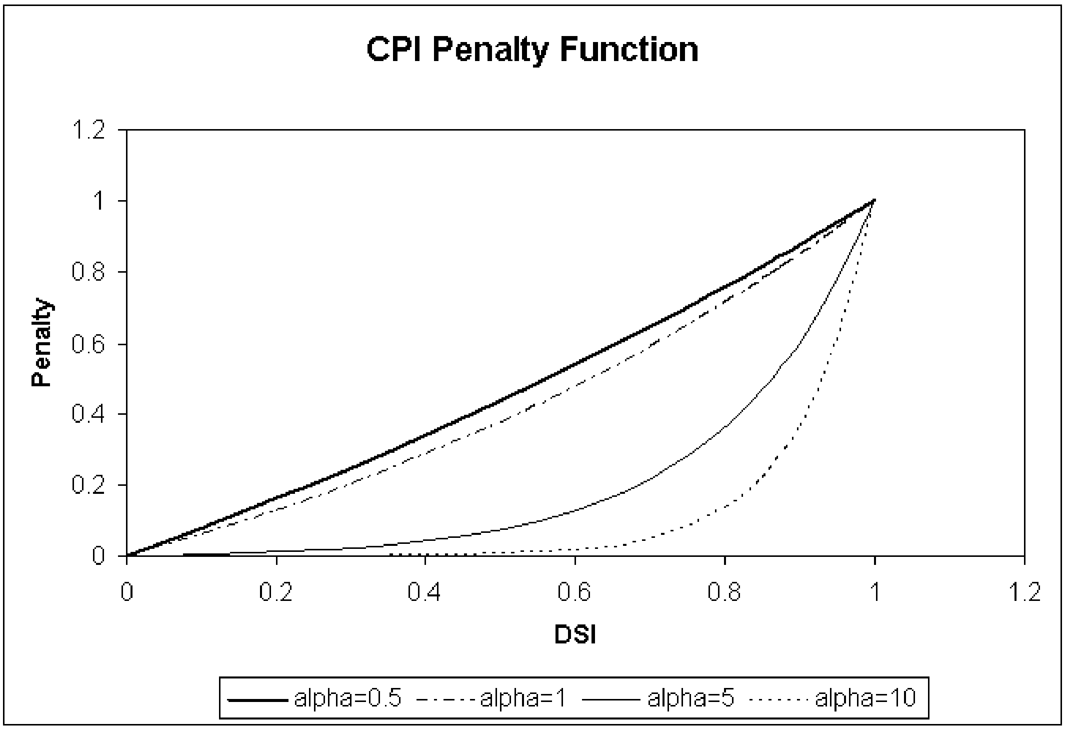


Figure 4.8. Effect of α on *CPI*

The greater the value of α, the greater is the curvature, depicting a more `tolerant' attitude as shown in Figure 4.8. For the same value of *DSI*, the corresponding *CPI* value of a more tolerant curve (larger α) is lower than that of a less tolerant curve. The value of α is affected primarily by the Planner's choice of *m*. Higher values of *m* lead to higher values of α. This is to be expected, as the higher the Planner's tolerance to congestion, the corresponding acceptable limit of *DSI* values for Scenario 2 is expected to be higher. *CPI* values less than 1 indicate that the tolerable congestion cut-off identified by the Planner has not been reached. From the derivation, it is evident that a schedule with CPI_{Total} more than 1 is infeasible as determined by the Planner's preference. By introducing the utility approach, the CPI_{Total} provides a consistent scale to evaluate congestion potential subjected to a congestion tolerance.

4.6. Illustrative Case Study

The following case study is used to show the applicability of the proposed quantification methodology for analysing congestion. Using the *CPI* and *DSI* indicators, the schedules could be improved to lower spatial congestion. The improvement of the schedule with respect to the quantified congestion provides a basis for optimization to be carried out, which will be demonstrated in Chapter 7.

The case study involved an overhaul of an existing oil refinery by a major refinery company. The works included the internal modification of a stripper column with an internal diameter of 3.6m. The column has a central core riser 1.2m in diameter. The process involved the removal of a series of 10 baffle plates inside the stripper column by plasma cutting, after which the internal walls of the column were revamped to allow for the installation of two internal grid structures. New metallic gauze packing comprising eight gauze layers would be loaded onto a grid structure at the bottom, and subsequently "held down" by a grid structure at the top. Simultaneously, a new steam ring below the removed baffle stripper plates was to be replaced. The works were supervised by a Site Engineer.

The waste from the plasma cutting was removed through a manhole at the bottom of the column, while the loading of the metallic gauze packing was done through a manhole located at the top of the stripper column. The potential interferences between the access paths through the manholes and the workspaces for installation and removal coupled with the narrow space between the column wall and the internal riser made site conditions extremely cramped with a potential for severe site congestion.

The pictorial representation of the works involved is shown in Figure 4.9, where Figure 4.9(a) depicts the external tower, Figure 4.9(b) the existing internal structures, and Figure 4.9(c) the refurbished internal structures.

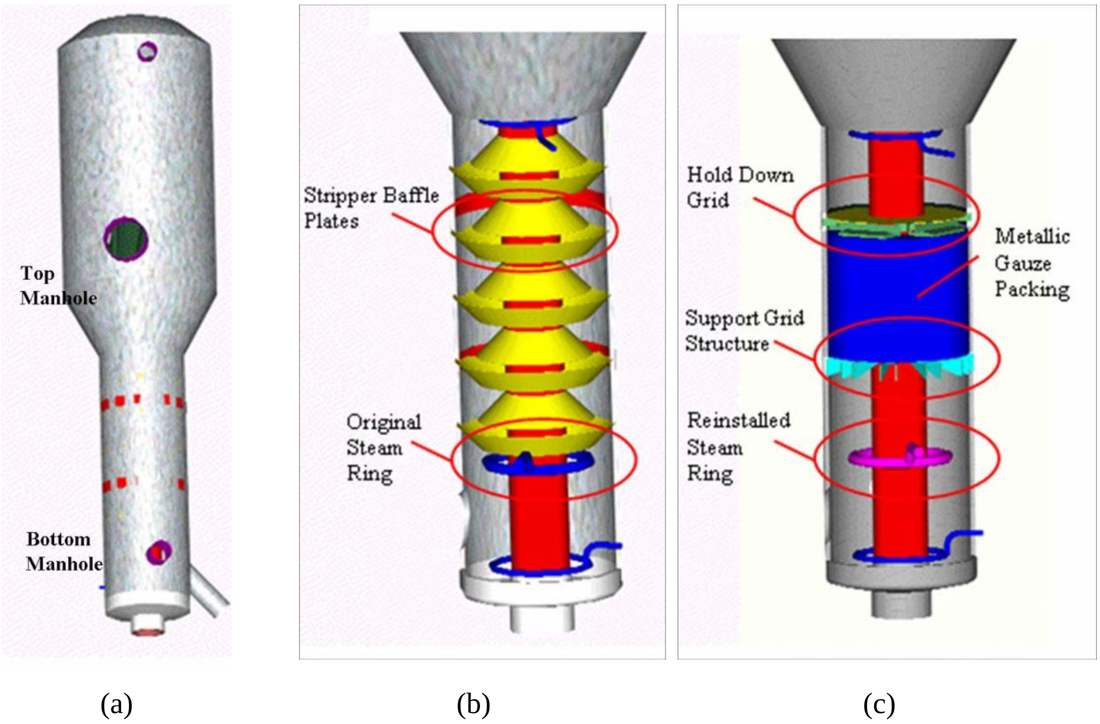


Figure 4.9. 3D Space Representation of Scope of Works

Figure 4.10 shows the critical time window of the original schedule for the refurbishment phase. Due to the tight time constraint, work proceeded on a 24-hour schedule throughout a seven-day work week. In total, the project was completed in 18 days.

Remove existing Steam Ring	8hrs
Trim existing Baffle Plates	48hrs
Hack Refactory for new Hold Down Grid Brackets (5 points)	16hrs
Hack Refactory for new Support Grid Brackets (20 points)	24hrs
Install and Wield new Support Grid and Hold Down Grid Brackets (24 points)	36hrs
Reinstate Refactory for Support Grid and Hold Down Grid Brackets	36hrs

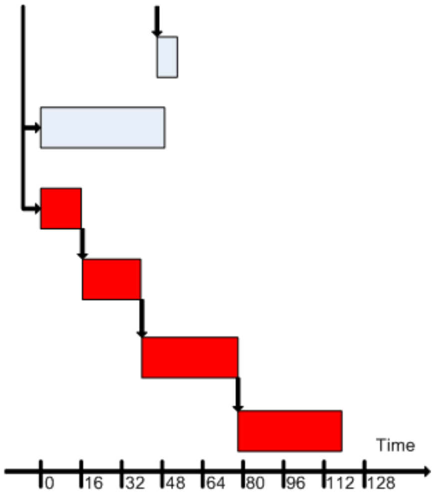


Figure 4.10. Schedule of Refurbishment Works

4.6.1 Analysis of Case Study

Figure 4.11 depicts the space entities of the activities in the critical time window of Figure 4.10. The Pathspace entities are also included in the analysis although they are not explicitly shown in for clarity. The Pathspaces are defined such that they originate from the top and bottom manholes shown in Figure 4.9(a) towards their respective workfaces. Various works were simultaneously being carried out: the trimming of the sharp edges of the steel baffle plates and the installation of both the Hold down Brackets and Support Brackets. The result of the simultaneous workflows is the existence of multiple interactions between Workspace and Pathspace components. These interactions lead to the phenomenon of congestion of workspace within the internal column.

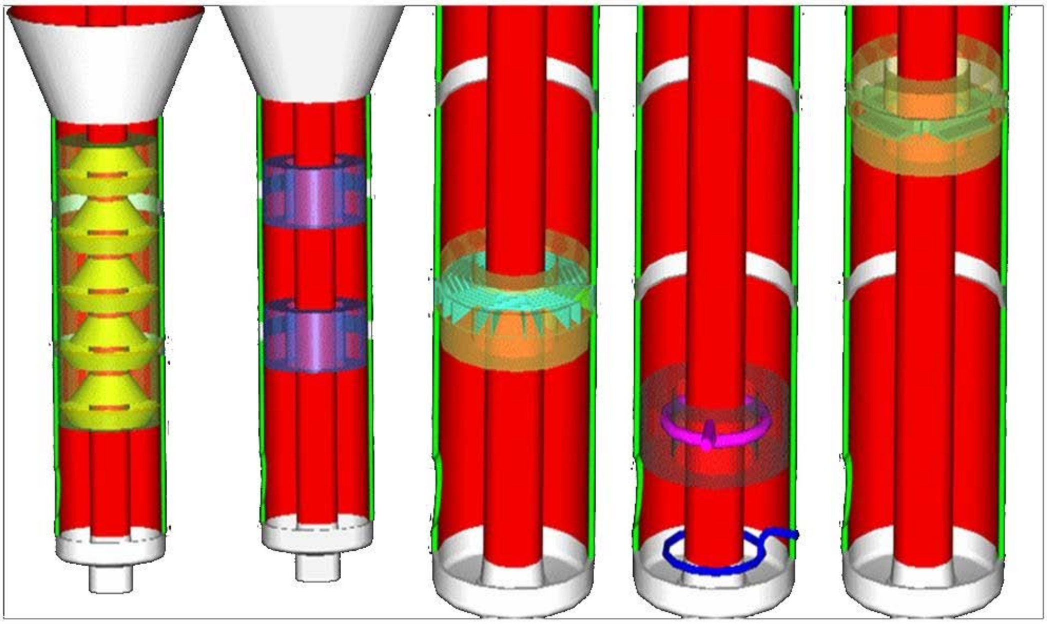


Figure 4.11. Layout of Relevant Workspaces

The crew size for each activity was four men with each man assumed to occupy an operator space of $0.6m^3$. Equal weightage between temporal and spatial utilizations was also assumed. From the estimation of the Site Engineer, the Pathspaces had 30% temporal utilization, while the workspaces had 100% temporal utilization. Table 4.1 shows the calculated *DSI* for the activities in Figure 4.10. The results indicated that the Workspace for the Removal of the Steam Ring (*SteamRing_Removal_WS*) was the activity with the highest *DSI* value (0.95 for workspace and 0.75 for pathspace) which was indeed perceived on site to be most congested by the Site Engineer. The analysis further indicated that this was due to the multiple paths from concurrent activities which infringed the activity space required by *SteamRing_Removal_WS*.

Table 4.1. Results of Dynamic Space Interference Factors

Activity Space Entity	Description	DSI
Trim_Baffle_WS	Workspace for Trimming of Baffle Plates	0.33
Refactory_Removal_WS_1	Workspace for Removal of Top Refactory for Hold Down Grid	0.69
Refactory_Removal_WS_2	Workspace for Removal of Bottom Refactory for Support Grid	0.69
SupportBracket_WS	Workspace for Installation of Support Grid Brackets	0.53
SteamRing_Removal_WS	Workspace for Removal of Steam Ring	0.95
HoldDown_Bracket_WS	Workspace for Installation of Hold Down Grid Brackets	0.53
Trim_Baffle_PS	Pathspace for Trimming of Baffle Plates	0.38
SupportBracket_PS	Pathspace for Installation of Support Grid Brackets	0.32
SteamRing_Removal_PS	Pathspace for Removal of Steam Ring	0.75
Refactory_Removal_PS_1	Pathspace for Removal of Top Refactory for Hold Down Grid	0.38
Refactory_Removal_PS_2	Pathspace for Removal of Bottom Refactory for Support Grid	0.43
HoldDown_Bracket_PS	Pathspace for Installation of Hold Down Grid Brackets	0.28
Refactory_Install_PS_1	Pathspace for Installation of Top Refactory for Hold Down Grid	0.26
Refactory_Install_PS_2	Pathspace for Installation of Bottom Refactory for Support Grid	0.20
Refactory_Install_WS_1	Workspace for Installation of Top Refactory for Hold Down Grid	0.50
Refactory_Install_WS_2	Workspace for Installation of Bottom Refactory for Support Grid	0.50
Trim_Baffle_WS	Workspace for Trimming of Baffle Plates	0.33
Refactory_Removal_WS_1	Workspace for Removal of Top Refactory for Hold Down Grid	0.69
Refactory_Removal_WS_2	Workspace for Removal of Bottom Refactory for Support Grid	0.69
SupportBracket_WS	Workspace for Installation of Support Grid Brackets	0.53
SteamRing_Removal_WS	Workspace for Removal of Steam Ring	0.95
HoldDown_Bracket_WS	Workspace for Installation of Hold Down Grid Brackets	0.53

The activity with the longest free float, namely "Trim Baffle Existing Plates", was selected for rescheduling to demonstrate how a better schedule with respect to congestion could be generated. The graphs in Figure 4.12 show the effect of varying the start time of the *Trim_Baffle* activity between its early start and late start (from 0hrs to 65hrs) on the *DSI* values of the interfering activity workspaces and pathspaces.

The activity comprises two different space entities: the *Trim_Baffle* workspace and the *Trim_Baffle* pathspace. The delay of this activity changes the interaction pattern of these space entities with the space entities of the other activities; the other interaction patterns remain unchanged.

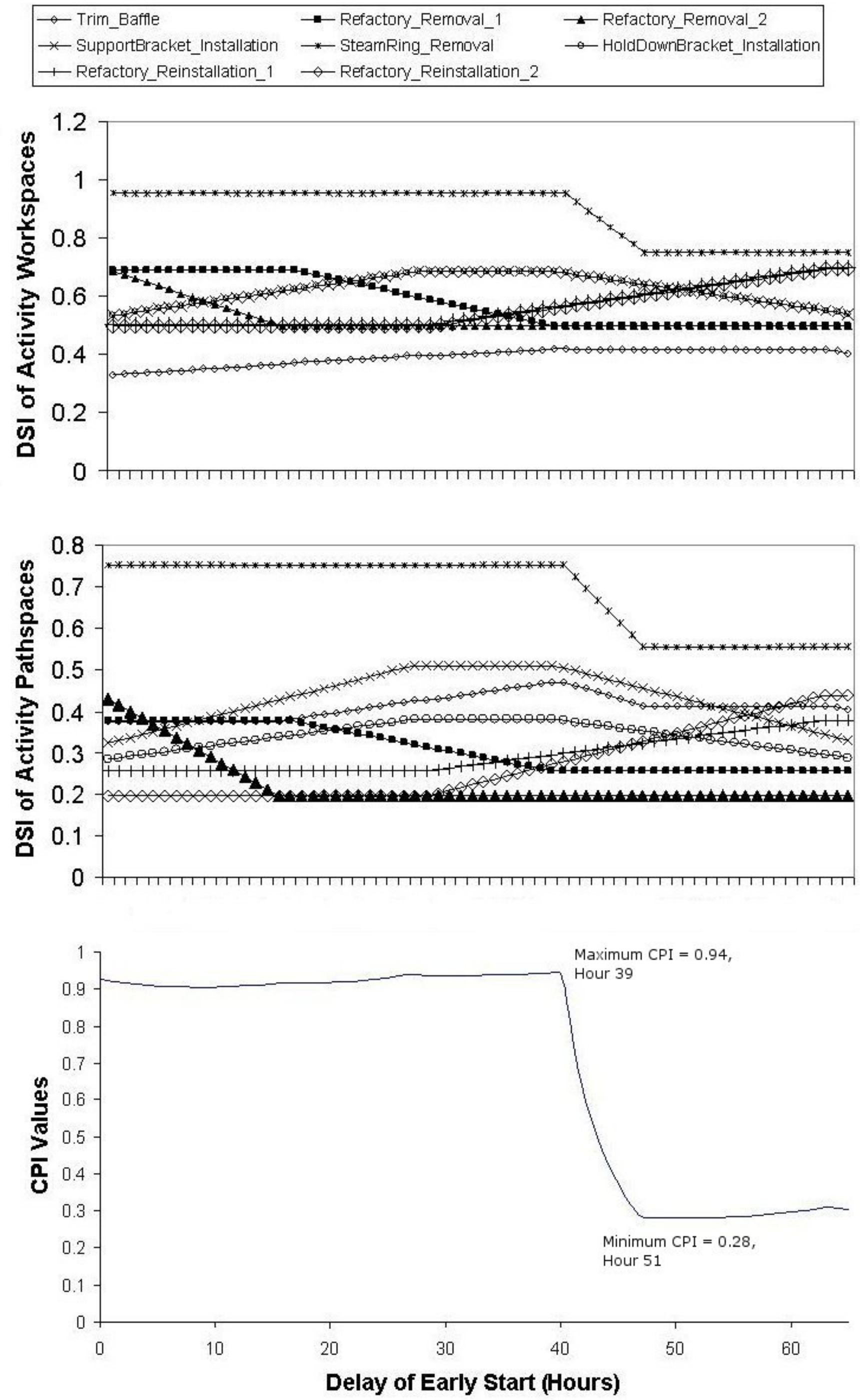


Figure 4.12. Effect of Delaying the ES of Trim_Baffle Workspace

For that critical time window with 16 interacting space entities, a critical value of 1 and tolerance limit of 0.7 yields a congestion tolerance value of $\alpha = 9.24$ that would represent the Site Engineer's preference. Varying the start time of the *Trim_Baffle* activity from its early start to late start, results in varying *CPI* values for the 16 interfering activity space entities. Figure 4.12 shows the composite CPI_{Total} for the critical time window. It is evident from Figure 4.12 that a minimum $CPI_{Total} = 0.28$ can be obtained if the *Trim_Baffle* activity is delayed 51 hours after its Early Start ('Hour 51') as compared to the maximum $CPI_{Total} = 0.94$ which occurs 39 hours after the Early Start ('Hour 39'). This can be justified by analysing the interactions of the activities at 'Hour 51' and 'Hour 39'. At 'Hour 39', 8 other activity space entities interfere with the *Trim_Baffle* activity, while at 'Hour 51', only the "Installation of Support Grid and Hold down Brackets" and the "Reinstate Refactory" activities are affected.

The *CPI* value quantifies the level of congestion as a composite index, and demonstrates the idea that rearranging the activity floats can minimize activity interactions, and consequently reduce congestion. *CPI* can be used as a decision making tool to evaluate projects from a congestion perspective. Moreover, it offers the opportunity for simple optimization techniques to be employed to generate improved schedules in critical time windows.

4.7. Concluding Remarks

For cases where work continuity is not adhered to, different locations within the total boundary space do not lead to significant degrees of interference. This is because operators can be expected to accommodate one another, reaching a "local compromise or workaround". The overall utilization factor (and subsequently the DSI value of Equation 4.4) achieves this by adopting a single representative value which is the "average scale" of U_S and U_T values.

For cases where work continuity is expected or where a singular workspace is not representative of the conditions on-site, the quantification methodology can still be applied by modelling smaller workspaces or by decomposing the singular workspace into its constituent workspaces (e.g. multiple workspaces or equipment spaces). This decomposition usually implies a segregation of the time frame allocated to the workspace as shown in the following Figure 4.13. The values of U_S and U_T for the interfered sub-workspaces are increased, and the subsequent DSI value also increases. This is because the available space and time are bound, limiting the "flexibility" for workarounds to be conducted, as the operators cannot continue to work in another workspace until the current workspace is completed. Hence, this reflects the increased congestion due to more restrictive space-time constraints. The division of workspaces into its sub-workspaces or constituent workspaces is left to the modeller's discretion.

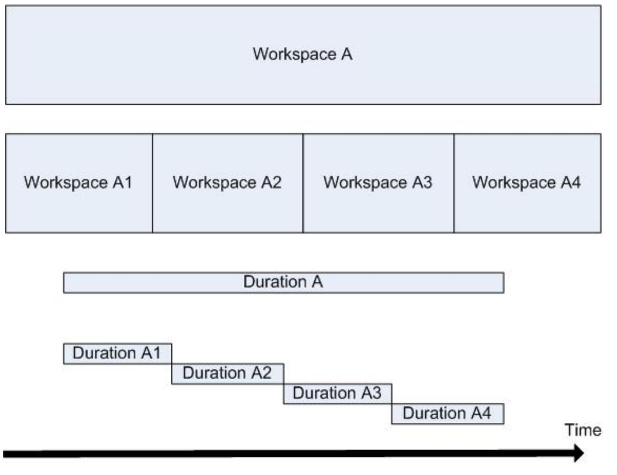


Figure 4.13. Division of Workspaces

This chapter presented an ontological model of space from a utilization perspective. This augments the previous deterministic space analysis models through the quantitative consideration of the operator space as a resource. From a construction requirements perspective, this operator space is a fundamental component of the spatial attribute in the knowledge representation of space entities within construction requirements. Consequently, this enables the redefinition of the space conflict taxonomy from the perspective of space utilization.

The identification and quantification mechanism of spatial temporal conflicts and congestion is also proposed as a contribution of this research. This mechanism comprises the DSI indicators, which is a geometric combination of the space and time utilization factors of a specific space entity. The conflict/congestion identification then centres about the analysis of the supply and demand of space-time for that space entity. This indicator is incorporated as an important metric quantity for space entities of the construction requirement.

Finally, this chapter also argues for the need of a high level indicator, to provide Planners with a single value to quantify the conflict/congestion in a construction schedule. This allows for optimization of the schedule to minimize conflict/congestion as shown in Chapter 8.

The workspace analysis framework proposed in this chapter contributes to the knowledge of constructability analysis, allowing for quantitative identification of worksite conflict and congestion. Planners can use this as a tool to automatically analyse site spaces for potential issues arising from congested work areas.

Chapter 5. PDM++: A Modelling Framework for the Temporal Attributes of Construction Requirements

5.1. Introduction

This chapter presents the fundamental modelling framework for incorporating the temporal attributes of Construction Requirements, which were represented as part of the temporal attribute inter-relationship taxonomy. Following the ontological introduction of construction requirements in Chapter 3, this chapter will focus on the temporal aspects of these requirements. As such, the spatial aspects are implicitly assumed to be present; hence, the term "work package" which was originally proposed in the previous chapter to imply the existence of both spatial and temporal aspects is used interchangeably with its temporal constituent "task" or "activity" for enhanced readability in this chapter.

A review of the current models for knowledge based sequencing in construction planning will be presented in the following section, and it will also demonstrate the inadequacies of present methodologies for representing temporal attributes of construction requirements.

Following the review, the logical foundation of PDM++ will then be introduced which will provide the reader with an insight into the underlying mathematical logic. These logical foundations will include the introduction of the basic semantic and syntax which make up the PDM++ modelling methodology.

Following the logical foundation of PDM++, this chapter will present the use of the modelling methodology in describing the various optional operation modes for a simple pipe laying example. This chapter will also include a simple constraint analysis and define new categories of criticality in the presence of other operation modes.

5.2. Review of Current Modelling Frameworks for Construction Planning

Traditional planning and scheduling models like Critical Path Method (CPM) and Linear Scheduling Method (LSM) cannot adequately capture many of the construction requirements, like work/resource continuity and process concurrency/overlap (Jaafari, 1984, El-Rayes and Moselhi, 2001). Additionally, CPM dictates a specific work sequence although alternative sequences exist which equally fulfil the construction requirements. These inadequacies of CPM, of which Precedence Diagramming Method (PDM) is currently most popular (Wiest, 1981), limit the semantic representation of requirements.

Previous research has identified the limitations and has attempted to extend the traditional PDM. Plotnick (2006) proposed Relationship Diagramming Method (RDM) which adds "reason/why" codes programmatically to give planners a clearer understanding into the reasons for the inclusion of a relationship or activity. Singular events allow important datum to be included in RDM's activity based representation, enhancing the semantic description of activities and relationships. The RDM framework handles alternative sequences through stochastic means, incorporating Graphical Evaluation and Review Techniques (GERT).

Koo, *et al.* (2007) proposed Constraint-Loaded CPM (CLCPM) which is focused on using a Constraint Ontology to identify the role of a constraint on the schedule, allowing for project planners to identify activities for re-sequencing. However, CLCPM only handles finish-start precedence relationships, and is at present unable to handle more complex and dynamic relationships which are characteristic of construction requirements. In a similar fashion, Echeverry, *et al.* (1991b) described the flexibility of constraints based on the knowledge of their relationship with physical components, trade interactions, path interference and code regulations. However, no solution methods were proposed to evaluate his classification methodology. Finally, El-Bibany (1997) also studied the role of knowledge in enhancing CPM schedules, and introduced a constraint programming / parametric framework to incorporate such knowledge into CPM.

Another approach uses Soft Logics to reason parallel paths of construction activities (Tamimi and Diekmann, 1988, Fan and Tserng, 2006). These soft logic models recognize that normal CPM methods only capture one prescribed work sequence. The SOFTCPM Algorithm was proposed to heuristically sequence the activities under the effects of soft and fixed logics. However, the scope of such soft logic models is still limited, and is unsuitable for construction requirements as they are not able to capture some of the more complex temporal relationships.

5.3. System Requirements of the Modelling Framework

This research identifies the following system requirements for modelling the temporal impact of Construction Requirements on project schedules, particularly using traditional methods in Section 1.3.3. These are labelled SR1 to SR5 as follows. Later sections will refer to these system requirements, showing how the PDM++ modelling framework is able to handle these requirements.

- **System Requirement 1 (SR1):** Firstly, it may be possible to choose between sets of temporal constraints while still satisfying a specific construction requirement, implying a disjunction between temporal constraints. Traditional planning models do not usually allow for planning considerations where activities are constrained such that they cannot be carried out simultaneously (disjointed activities). These planning considerations may be used for several purposes: to force activities with known safety hazards to be carried out non-concurrently, or allow for limited scheduling of key resources during planning. The framework must thus be able to adequately handle such "time windows of non-work". As an example, for safety considerations, hotwork and painting operations should not be carried out simultaneously; either the hotwork is scheduled before painting or vice versa. The availability of choice leads to diffferent schedules while satisfying the construction requirement.

- **System Requirement 2 (SR2):** Secondly, more complex temporal constraints arising from the construction requirements must be able to be modelled. For example, the framework should be able to adequately represent work or resource continuity (Vanhoucke, 2006). This sometimes arises in repetitive construction projects, where a resource which is required by a set of activities moves sequentially from one stage to another. Such considerations are sometimes necessary when it is vital to minimize the idle time of a resource.
- **System Requirement 3 (SR3):** Thirdly, overlapping and concurrent relationships between activities need to be adequately defined and handled by the system. Often, modelling these overlaps and concurrencies include the use of negative lags (Douglas, *et al.*, 2006) and/or logical loops (Callahan, *et al.*, 1992). Such logical loops are valid from a model perspective, but invalid algorithmically as it violates the acyclic network

assumption. A typical example is given (O'Brien and Plotnick, 2010) as shown in Figure 5.1: A Drywall activity, *X* with 7 days duration has a SS2 restraint with a Rough-in electrical activity, *Y* with a 5 day duration. However, *Y* must finish 2 days before *X* finishes (depicted with a FF2 relationship). The network logic is violated even though the activities make logical sense from an engineering perspective.

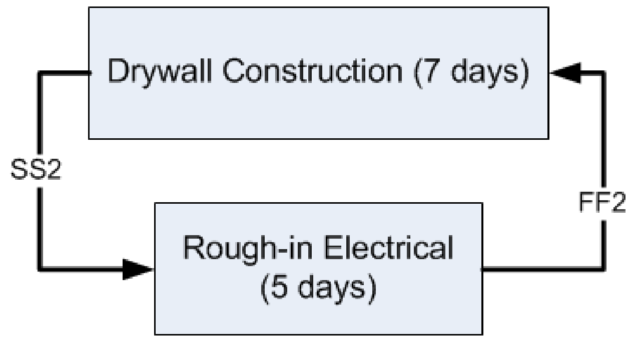


Figure 5.1. Example of Dry Wall Construction

- **System Requirement 4 (SR4):** Fourthly, some construction requirements are interdependent. Fulfilling a (set of) requirement(s) may be conditional on the satisfaction/non-satisfaction of another (set of) requirement(s). This inherent interdependency may cause additional difficulties, as the set of requirements to be fulfilled may change over time due to alternative construction methods and changing site conditions. For example, modules are usually hoisted in place by cranes to construct offshore platforms. However, schedule delays may lead to space constraints, requiring a temporary holding structure be built to receive the module. This new requirement arising from lack of space requires the module be hauled in place.

- **System Requirement 5 (SR5):** Lastly, some construction requirements may involve several intervals simultaneously. This may arise from a need to model hierarchical decompositions of activities into sub-tasks, or for representing key component states of resources and equipment (Chua and Yeoh, 2011). These high-level abstractions of groups of activities simplify the model representation, reducing confusion through minimizing the number of relationships specified within the model.

The use of logical operators is proposed to capture the complex descriptions of construction requirements as detailed in the system requirements SR1 to SR3. To address the interdependencies of requirements (SR4), Construction Requirements were classified in Chapter 3 as being static or dynamic; some advanced logical operators are introduced in this research to handle these interdependencies. Additionally, a meta-interval construct is proposed to facilitate the representation necessary in SR5. A framework called PDM++ is developed to provide clearer semantic mapping from construction requirements to schedule constraints. This presents an alternative to the traditional PDM formulation to deal with the complex schedule constraints arising from requirements, and provides planners with a tool to drive schedules from the perspective of construction requirements.

5.4. Using the PDM++ Framework for Temporal Constraints

The proposed PDM++ framework is inspired by Artificial Intelligence representation approaches developed by Allen (1983). Allen's representation comprises a set of 13 mutually exclusive binary temporal relations, which depicts a complete description of the possible relationships between any two time intervals

(Allen, 1984). These intervals are summarized in Table 5.1 and Table 5.2. Song and Chua (2007) have previously used Allen's relations to model the class of Intermediate Function requirements. The key contribution of this work is the extension of Allen's representation from a symbolic language to a numeric one consistent with current planning frameworks like PDM. This allows for numeric evaluation and analysis via float computation of the construction plan. From the complex symbolic representation of Allen's relations, a numerical equivalent for evaluating construction requirement temporal relationships is built.

Allen's relations between intervals are initially decomposed into Start points and End points of the activities, much like the approach by Vilain, *et al.* (1990). For example, Allen's relation "*X <(before) Y*" can be transformed mathematically involving the Start Point of *Y*, Y^- and the End Point of *X*, X^+ as follows

$$X^+ < Y^- \tag{5.1}$$

One critical assumption of PDM++ is the continuity of activities (non splittable), so that the end point, X^+ may be expressed as a simple linear function of the start point, X^- as follows

$$X^- + d_X = X^+ \tag{5.2}$$

where d_X is the duration of the activity. Hence Equation 5.1 may be expressed solely in terms of the start points of *X* and *Y* as follows

$$X^- + d_X < Y^- \tag{5.3}$$

Metric information in the form of lag time, m is added to the mathematical description of Allen's relations to facilitate project scheduling, enabling the proposed model to emulate PDM. In the above example, a lag of m days between activities X and Y may be introduced by incorporating the lag in Equation 5.3, resulting in the following relationship:

$$X^- + d_X + m < Y^- \quad \textbf{(5.4)}$$

With some embellishment to Equation 5.4, the Allen's relationship may be translated into the PDM++ relationship of "*X Before(n) Y*":

$$X^- + d_X + n \leq Y^- \quad \textbf{(5.5)}$$

This embellishment is applicable due to the change from Allen's symbolic domain over a continuous time interval to the discrete time interval commonly adopted in current planning frameworks, where $n = m - 1$ is used for this example.

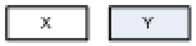
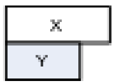
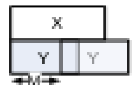
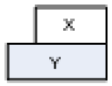
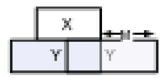
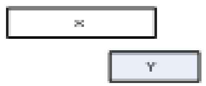
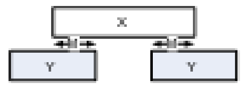
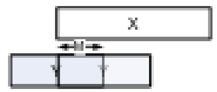
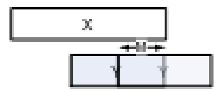
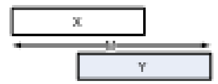
The two basic logical operators are introduced here to extend the syntax in PDM++: "Conjunction" and "Disjunction". The Conjunction operator ($\wedge$) allows for conjunctive constraints to be expressed, which is analogous to the "AND" Boolean operator. Meanwhile, the Disjunctive operator ($\vee$) allows for disjunctive sets of constraints to be depicted, which is analogous to the "OR" Boolean operator. These operators will be further developed in subsequent sections. These operators increase the expressiveness of PDM++ to capture construction requirements by allowing for more explicit representation of temporal relations arising from planning considerations through the perspective of construction requirements.

5.4.1 Representing Semantic Relationships as Constraints in PDM++

Table 5.1 shows Allen's Interval relationships with the corresponding semantically embellished binary constraints of PDM++. Three types of relationships are distinguished: Minimal-lag, Maximal-lag and Non-lag type. The Minimal-lag type is the usual start-and-end-point formulation adopted in the traditional PDM framework, where the constraint must *at least* satisfy the given lag time, *m* in the relationship. The Maximal-lag type is adapted from prior research (Neumann and Zhan, 1995, Hajdu, 1997), and these constraints must *at most* satisfy the given lag time, *m*. To distinguish the two, a tilde is added to the nomenclature of Maximal-lag types in Table 5.1. As a convention in this dissertation, binary relationships with minimal lags of zero duration omit the zero lag values. For example, "*Before(0)*" is simply represented as "*Before*".

Since Allen's Interval Representation identifies all the possible relationships for describing the temporal relationship between any two intervals or activities, Table 5.1 and Table 5.2 (shown later in this section) is sufficient to show the full semantic capability of PDM++ for describing any relationship between any two intervals (activities) arising from a construction requirement.

Table 5.1. Binary Relationships of PDM++

Allen's Interval Relationship	Pictorial Representation	PDM++ Framework				
		Minimal-lag	Mathematical Definition	Maximal-lag	Mathematical Definition	Pictorial Representation
$X < Y$ (before) $Y > X$ (after)		X Before(m) Y Y After(m) X	$X^- + d_X + m \le Y^-$	X Before(~m) Y Y After(~m) X	$X^- + d_X + m \ge Y^-$	
X s Y (starts) Y si X (started-by)		X Starts(m) Y Y Started-by(m) X	$X^- + m \le Y^-$	X Starts(~m) Y Y Started-by(~m) X	$X^- + m \ge Y^-$	
X f Y (finishes) Y fi X (finished-by)		X Finishes(m) Y Y Finished-by(m) X	$X^- + d_X + m \le Y^- + d_Y$	X Finishes(~m) Y Y Finished-by(~m) X	$X^- + d_X + m \ge Y^- + d_Y$	
X o Y (overlaps) Y oi X (overlapped-by)		X Overlaps(m) Y Y Overlapped-by(m) X	$(X^- + d_X \ge Y^- + m) \wedge (Y^- + d_Y \ge X^- + m)$	X Overlaps(~m) Y Y Overlapped-by(~m) X	$(X^- + d_X \le Y^- + m) \vee (Y^- + d_Y \le X^- + m)$	
		X Start-Overlap(m) Y Y Start-Overlapped-by(m) X	$(X^- + m \le Y^- + d_Y) \wedge (X^- \ge Y^-)$			
		X End-Overlap(m) Y Y End-Overlapped-by(m) X	$(X^- + d_X \le Y^- + d_Y) \wedge (X^- + d_X \ge Y^- + m)$			
		X Start-Finish(m) Y	$X^- + m \le Y^- + d_Y$	X Start-Finish(~m) Y	$X^- + m \ge Y^- + d_Y$	

To address SR3 the following relationships are introduced. The "*Before/After*", "*Starts/Started-by*" and "*Finishes/Finished-by*" relationships are analogous to FS, SS and FF precedence relationships in traditional PDM. The "*Overlaps/Overlapped-by*" relationship from Allen's Intervals is refined into four cases*: "Start-Overlap(m)", "End-Overlap(m)", Overlap(m)", and "Start-Finish(m)"*. The "*Start-Overlap(m)*" and "*End-Overlap(m)*" are intended to capture a similar semantic meaning as Allen's "*Overlaps/Overlapped-by*" while the "*Overlap(m)*" relationship is a more general case of depicting overlap between two intervals. The "*Start-Finish(m)*" relationship is often used in practice to show an overlapping of activities where the completion of one activity requires the inputs or contribution from its predecessor and is analogous to SF precedence relation of traditional PDM.

The "*Start-Overlap(m)*" and "*End-Overlap(m)*" relationships depict sequences of work where the start or end, respectively, of an activity may be important. For example, an activity X's start or end (of *m* lag days) may require "support" in the form of resources, materials or information from another activity, Y. Figure 5.2 illustrates the "*Start-Overlap(m)*" showing the possible Early Start and Late Start scenarios (denoted by subscript ES and LS respectively) of Y in relation to X on a Gantt Chart. This, for example, can be used to describe the relationship between "*Ducting and Cabling of Prestressing Tendons*" activity (X) and "*Rebar Cage Fabrication*" activity (Y) in the post-tensioning of a bridge segment. The placement of the cable ducts requires concurrent welding to the rebar cage, thus requiring the "support" of the "*Rebar Cage Fabrication*" activity at its start. The "*End-Overlap(m)*" can be similarly depicted.

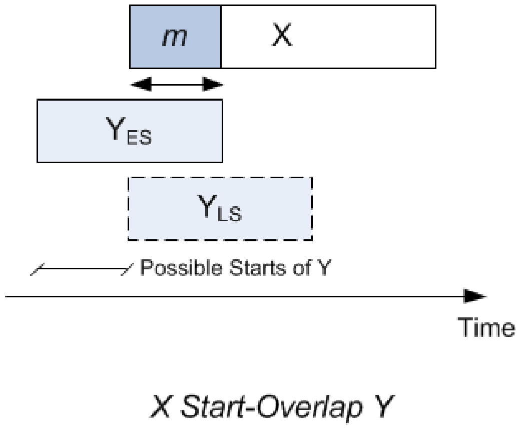


Figure 5.2. Start-Overlap(m) Relationships

The *"Overlap(m)"* is used when both activities X and Y must overlap by a minimum of *m* days (minimal-lag) or a maximum of *m* days (maximal-lag), but the order of the activities is immaterial, as depicted in Figure 5.3. This relationship may be used when an inspection crew needs to check the as-built dimensions of a series of beams (X). Some beams are inconveniently located, and require a scaffold (for *m* days), provided by a painting subcontractor (Y). The order in which the beams are checked is immaterial, so these inconveniently located beams may be checked at any point of time during X, with the scaffold from Y providing the necessary "support".

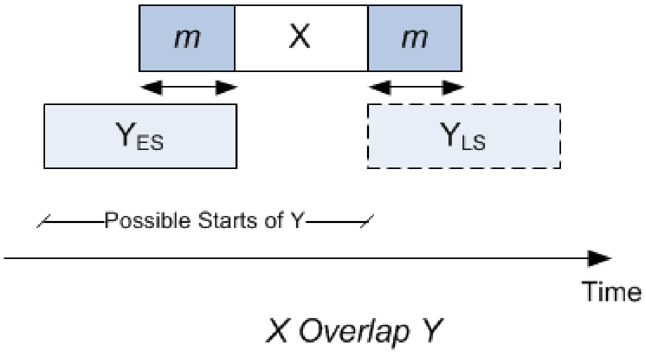
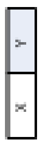
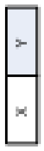
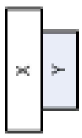
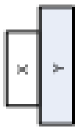
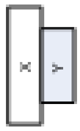
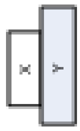
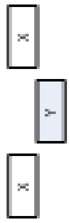


Figure 5.3. Overlap(m) Relationship

The third, Non-lag type relationships, in PDM++ is depicted in Table 5.2 with its corresponding Allen's Interval relationships. The Non-lag type, as the name suggests, is independent of any lag times, and provides greater descriptive capabilities to define the required relationship between two activities.

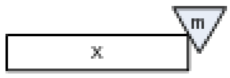
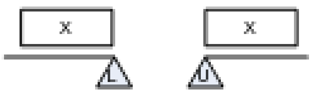
Table 5.2. Non-Lag type Binary PDM++ Relationships

Allen's Interval Relationship	Pictorial Representation	PDM++ Framework: Non-Lag	PDM++ Framework: Mathematical Definition	PDM++ Framework: Pictorial Representation
X m Y (Meet) X mi Y (Met-by)		X Meets Y Y Met-by X	$X^{-} + d_{x} = Y^{-}$	
X d Y (During) X di Y (During) $X = Y$ (Equal)		X Contains Y Y Contained-by X	$(X^{-} \geq Y^{-}) \wedge (X^{-} + d_{X} \leq Y^{-} + d_{Y})$	
		X Concurrent Y	**Case 1:** $d_X \geq d_Y$ $(X^{-} \leq Y^{-}) \wedge (X^{-} + d_{X} \geq Y^{-} + d_{Y})$	
			Case 2: $d_Y > d_X$ $(X^{-} \geq Y^{-}) \wedge (X^{-} + d_{X} \leq Y^{-} + d_{Y})$	
		X Disjoint Y	$(X^{-} + d_{X} \leq Y^{-}) \vee (Y^{-} + d_{Y} \leq X^{-})$	

SR2 may be handled using the following semantic relationships. The "*Meet/Met-by*" condition may be used to model strict resource or work continuity which the traditional PDM cannot enforce. The "*Contains*", "*Contained-by*" and "*Concurrent*" relationships can be used to define activities that must occur simultaneously. The *"Contains/Contained-by"* relationship carries a specific sense that a shorter activity is performed during a longer activity, while the *"Concurrent"* relationship implies a more general sense of concurrency. Similarly, to address SR1, the *"Disjoint"* relationship is introduced which may describe situations where activities sharing the same resources like equipment, labour, or conflicting workspace cannot be concurrent or overlap.

The Unary constraints of PDM++ shown in Table 5.3 are defined as constraints affecting a single activity. From a project perspective, unary constraints may arise from contractual issues such as milestones or operations issues like material, resource and information availability. For example, the constraint relationship *"Cannot-Occur"* can be used to describe a period where an activity cannot be carried out for various reasons such as contractual obligations, which is formulated as a disjunction of two constraints, *"Due-Before"* and *"Start-After"*.

Table 5.3. Unary PDM++ Relationships

Relationship	Mathematical Definition	Pictorial Representation
X Due-Before (m)	$X^- + d_X \leq m$	
X Due-After (m)	$X^- + d_X \geq m$	
X Start-Before (m)	$X^- \leq m$	
X Start-After (m)	$X^- \geq m$	
X Cannot-Occur $[L,U]$	$(X^- + d_X \leq L) \vee (X^- \geq U)$	

The above PDM++ relations bridge the semantic descriptions availed to by the use of interval-to-interval relationships between activities, with the mathematical manipulability of endpoint-to-endpoint relationships. The semantic descriptions of the interval-to-interval representation have the added advantage of closely following the natural language for specifying the schedule implications and temporal attributes arising from construction requirements. Consequently, the inclusion of logical operators ("Conjunction" and "Disjunction") extends the mathematical relations in Table 5.1 and Table 5.2 so that the interval-to-interval descriptions can be translated into the start-and-end-point relationships following the framework of PDM++. Similarly, Table 5.3 presents the unary semantic descriptions as start-and-end-point relationships, which may also be subjected to the operations of conjunction and disjunction.

5.4.2 Modelling Dynamic Construction Requirements

Dynamic construction requirements are complex requirements that are conditional upon the fulfilment of other requirements. This means that when a change occurs to a (set of) requirement(s), other requirements may also be affected. These dynamic construction requirements are implied by SR4, and their behaviour may be modelled through the following logical operators "IMPLICATION, $\rightarrow$", "EQUIVALENCE, $\leftrightarrow$" and "EXCLUSIVE-OR, $\otimes$".

"$I \rightarrow J$" means if I is true, then J is implied to be true. However, if I is false, then J may be either true or false. "IMPLICATION" may be used in the example, "If Activity A finishes before B starts, then C must start after A finishes", expressed as:

$$(\text{A } \textit{Before} \text{ B}) \rightarrow (\text{C } \textit{After} \text{ A}) \qquad \textbf{(5.6)}$$

Here, C may or may not start after A if A does not finish before B. Such conditional interdependency is used when activities become constrained under certain pre-conditions.

"I↔J" is used to model stronger pre-conditions, which semantically corresponds to "If and only If" statements. "EQUIVALENCE" operators are used to describe situations where the pre-condition, I if true implies that the post-condition J must also be true and vice versa. For example, "Activity C finishes before D starts if and only if A finishes before B starts", can be expressed as

$$(\text{A } \textit{Before} \text{ B}) \leftrightarrow (\text{C } \textit{Before} \text{ D}) \qquad \textbf{(5.7)}$$

This interdependency means either "C finishes before D starts" and "A before B", or else "C cannot finish before D starts" if "A does not finish before B". "EQUIVALENCE" corresponds to the "XNOR" Boolean operation.

One use of the above operators is to imply the existence of requirements when specific pre-conditions are met. An example of such a requirement may include safety requirements which are enacted when the conditions of the construction method require it to be present. Another example cited by Fan and Tserng (2006) which was not addressed by them, describes the requirements between the *"Wall Painting"* (WP) and *"Floor Carpeting"* (FC) activities. While it is possible to schedule either activity first, the FC would require an additional activity *"Carpet Protection"* (CP) if WP commenced after. This means that if FC commences first, then CP would not be necessary during FC which may be captured using the following "IMPLICATION":

$$(\text{WP } A\textit{fter} \text{ FC}) \rightarrow (\text{CP } \textit{Concurrent} \text{ FC}) \qquad \textbf{(5.8)}$$

"I⊗J" has been used previously under different contexts to describe alternative scheduling (Beck and Fox, 2000, Fan, *et al.*, 2003). The "EXCLUSIVE-OR" operator means that only one of two conditions I or J is true, but not both. For example, it may be possible for Activity B to start after A or for C to start after A, but not both. This may be used to model other construction methods which achieve the same outcomes.

Adding these conditional interdependencies to model dynamic construction requirements allows for a wider requirements representation with respect to schedule constraints. Some practical engineering applications of using the conditional interdependencies include the provision of safety equipment like ventilation fans or fall arrest systems subject to some high-risk work conditions, or catering for additional resources and equipment if certain contractual milestones are not met. Other logical operators may be added as necessary to better describe the conditional interdependencies of requirements. Some of these will be discussed in later sections of this research

5.4.3 Modelling Hierarchical Plans and Groupings of Activities

The final system requirement (SR5) arises because of the need to specify complex relationships in construction requirements. Often, requirements may be complex, involving several activities and schedule constraints. Hence, it may be necessary to model the constraints or requirements at higher levels of plan abstraction.

Such complex relationships often refer to constraints that involve three or more activities, or even to a system of constraints between two or more groups of activities. The representation of different sequences of work involving a group of activities described in the case study in Chapter 8 is an example.

Project Management Institute (2008) defines a similar temporal concept called the Hammock or Summary activity, where "a group of related schedule activities aggregated at some summary level, and displayed/reported as a single activity at that summary level". An algorithm is presented by Harhalakis, *et al.* (1987) , with its corresponding implementation described in Harhalakis (1990), where the hammock activity is made to comply with the behaviour of the other activities in the network. However, for describing the temporal attributes of construction requirements, the idea of the hammock activity is not expressive enough, and needs to be extended to better define the usable construction states as well. This is important, as it is the existence of the usable windows of the construction states which support the activity.

The system should allow for the new expressive semantics in PDM++ to be used on groups of activities. This may be used to model hierarchical decompositions of activities into sub-tasks, or for representing key component states of resources and equipment. These high-level abstractions of groups of activities simplify the model representation, reducing confusion through minimizing the number of relationships specified within the model.

Within the framework of PDM++, the Meta-interval construct is proposed to represent the aforementioned complex requirements and to address the requirement SR5. The meta-interval generalises the hammock activity, by incorporating the spatial component as component states within its ontological description. Hence, the meta-interval can be conceived as a higher-level work package entity of a construction requirement, being made up of several low-level work package entities.

These spatial and temporal characteristics give rise to the ability of the meta-interval to represent useful time periods within the group of intervals or activities which may represent these component states. Accordingly, more descriptive associations may then be made on these meta-intervals (and transitively, their component states) using the PDM++ framework. Additional implementation details involving the meta-interval and its generalisation of the hammock activity will be dealt with in later sections of this dissertation where the syntax is enhanced to incorporate the meta-interval construct. This section will focus on the description of the meta-interval, and its construction for modelling various construction scenarios.

The temporal aspects of the meta-interval construct can be defined as a time interval spanning across, and comprising, different activity durations. As such, unlike activity intervals with fixed durations, meta-intervals have variable duration, dependent upon the start and finish activities. The interval may be defined by this notation: *{Name.Interval: Comprised Activities, Start Activity, End Activity},* and its attributes characterized as follows:

1. *Name.* This provides an object identifier.

2. *Comprised Activities*. The activities that comprise the meta-interval.

3. *Start Activity, SA.* (optional) Activity characterizing start of meta-interval.

4. *End Activity, EA.* (optional) Activity characterizing end of meta-interval.

If the Start Activity (End Activity) is not specified, then the start (finish) of the meta-interval is tacitly the earliest start (finish) of all the activities in the set of its comprised activities. Four kinds of intervals may be used to specify the constraint relationships with other activities or meta-intervals, as characterized in Figure 5.4. The start-finish (SF) interval depicts the time interval between the start of SA to the finish of EA. The other time intervals may be similarly implied from their nomenclature. In Figure 5.4, the corresponding ECL^iPS^e clauses are given. Details of the implementation will be elaborated in a later chapter.

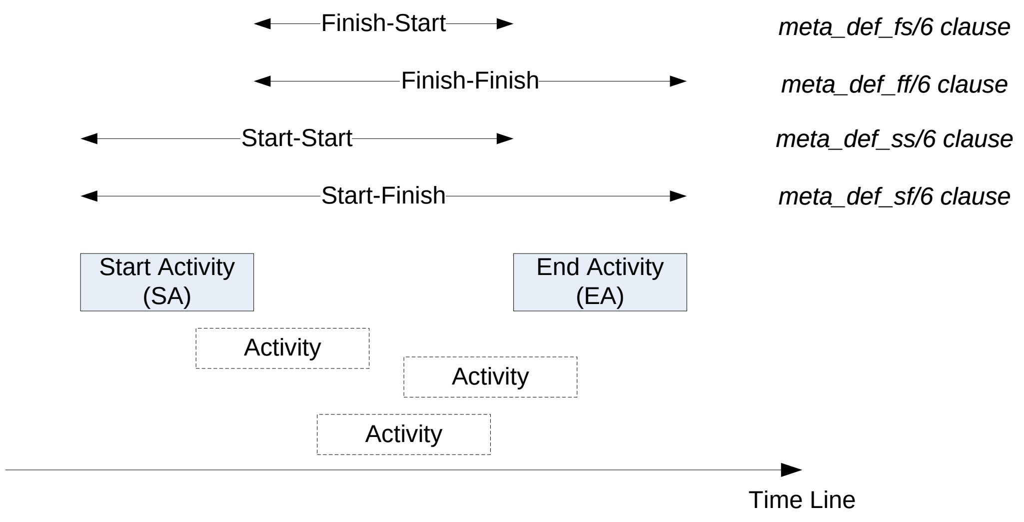


Figure 5.4. The Four Types of Meta-intervals

In the PDM++ framework, meta-intervals are treated like normal activity intervals. The same relationships and logical operators of "AND", "OR", "IMPLICATION", "EQUIVALENCE" and "EXCLUSIVE-OR" may be applied to meta-intervals.

The meta-interval may be used in the situations as shown in Figure 5.5 and Figure 5.7. Figure 5.5 depicts an example arising from hierarchical planning, involving groups of activities (A_1 to A_n), (B_1 to B_m) and (C_1 to C_p), represented as meta-intervals M_A, M_B and M_C respectively, with their corresponding constraints. The "*Contained-by*" relationship specifies that M_B must span within M_A while the "*Disjoint*" constraint requires that M_C cannot occur during the execution of activities in M_A. Meta-intervals M_A, M_B and M_C can serve as higher level abstractions of their constituent activities.

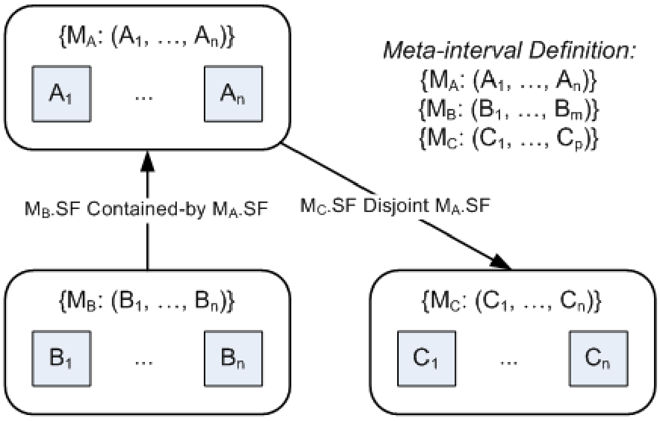


Figure 5.5. Meta-Intervals Implementation Example for Grouping of Activities

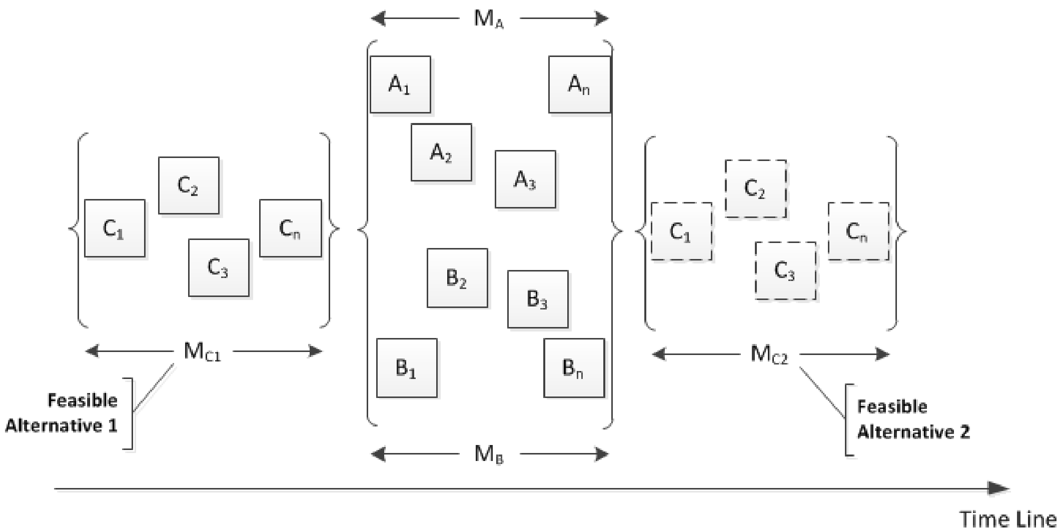


Figure 5.6. Timeline showing Feasible Alternatives for Meta-Intervals Implementation Example

Figure 5.6 shows the timeline for the same example given in Figure 5.5. Here, the location of the activities on the timeline indicates its possible starting time for this illustration, while the length of the box representing the activity denotes its duration. The brackets in this figure denote the meta-interval location along the timeline. Intervals M_A and M_B share the same period of time, with all the activities in M_B located within the interval demarcated by A_1 and A_n. Two possible locations for M_C exist (denoted by M_{C1} and M_{C2}) for this example, which shows the activities C_1 to C_n not being able to reside within the same period as M_A.

Figure 5.7 shows how the meta-interval is used to represent the state of some construction products or resources affecting the execution of an activity. The example shows the requirement where activity B needs the support of a scaffold. Its usable state is represented by the meta-interval {$M_{Scaffold}$.FS} which is governed by the interval defined by Finish of *"Erection"* (Activity A) and Start of *"Dismantling"* (Activity C). This requirement is depicted by the *"Contained-by"* relation as shown in the figure. Figure 5.8 shows the usable period available for using the scaffold needed by activity B along a timeline. This usable period is denoted by the meta-interval $M_{Scaffold}$. The available float for activity B within this meta-interval is also denoted to show that activity B may occur at any instance of time within this meta-interval.

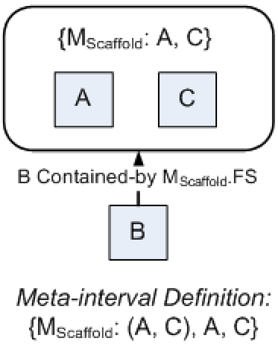


Figure 5.7. Meta-Interval Implementation Example of Describing Construction States

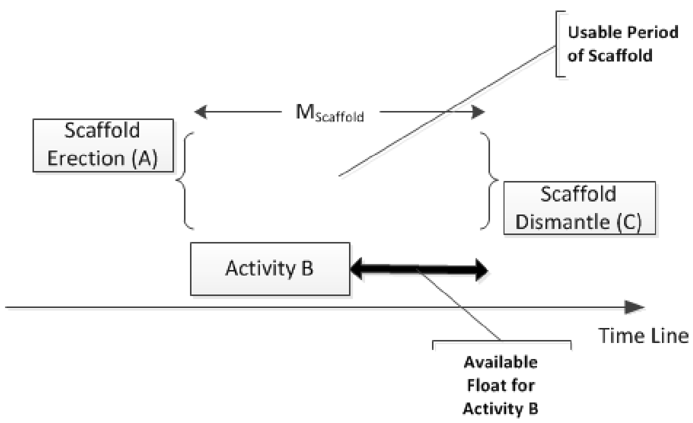


Figure 5.8. Timeline showing Implementation Example for Construction States

5.5. Construction Requirements Analysis

The PDM++ framework was developed with a generalised scheduler using *ECL^{i}PS^e*. Details of the development of this solver will be covered in Chapter 6. The output generated is either a set of feasible schedules which fulfil the mathematical representation of the temporal constraints arising from the construction requirements while also optimizing the project makespan[1], or no feasible schedule exists. For each activity in a feasible schedule, a domain of values is returned indicating the range of

[1] The makespan may also be added as a constraint within the modelling framework. This will allow the Planner to control the array of alternatives available, and reflect the effect of project end-date milestones on the generated alternative schedules. More details of the implementation will be given in Chapter 6.

possible start times for that particular activity. The Total Float is evaluated for each activity as the difference between the upper bound and lower bound of the range of possible start times. Critical activities are defined as per conventional methods as activities with zero total float, or as activities lying on the longest path. For the meta-interval, the scheduler returns not only the range of possible start times, but also a range of values representing possible durations.

From the perspective of analysing schedules, the available CPM methodologies do not allow for the analysis of alternatives and conditions. Traditional CPM quantifies the criticality of an activity, and consequently determines if a constraint is binding. Consequently, project indices like total float and free float are used to show the impact of delaying an activity on the project makespan and on the early start of the subsequent activity respectively. These indices of the project performance are usually specific to a particular schedule, and do not allow analysis over alternative schedules (Bowers, 2000). Furthermore, the traditional methodologies are not able to handle conditional and logical constraints.

This research emphasises on the importance of construction requirements on the schedule, and proposes that the requirements should be analysed directly to identify which of these vital requirements may hold up the schedule, as well as to identify the bottleneck constraints within the requirement. Since the construction requirements are recognised as sets of temporal constraints, the emphasis of identifying criticality from the perspective of activities for better project management should be shifted to studying and classifying the criticality from the perspective of constraints, and consequently its requirements. The proposed classification allows Planners to determine if a requirement is important by introducing the idea of constraint criticality under the effects of alternative constraints.

5.5.1 Definitions of Constraint Criticality

The above PDM++ model emphasizes the management of constraints through construction requirements rather than just solely managing activities on critical paths. This is because PDM++ may generate several alternative schedules, with differing critical paths. For the purpose of analysis, the framework presents three new definitions of criticality for helping Planners to monitor the requirements.

- **Definition (Critical Constraint):** A constraint is identified as being critical if it is a binding constraint, i.e. the activities affected by the constraint has a singleton value in its domain, and the constraint is exactly satisfied by the singleton values. These critical constraints can then be traced back to its construction requirement, allowing appropriate managerial action to be taken.

- **Definition (Super-Critical Constraint):** Further, a set of constraints which is identified as critical in all the alternative schedules may warrant greater attention from managers. These constraints are called "Super-critical". Delays or violations of any constraint in this "super-critical" set will invariably affect the project makespan.

- **Definition (Sub-Critical Constraint):** Another set of constraints are identified, which are critical in only some of the alternative schedules. These constraints are called "Sub-critical". This "sub-critical" set also requires attention from managers. Identifying this "sub-critical" set allows for plan flexibility when unforeseen circumstances occur which perturb the plan. Hence, when a “sub-critical” constraint is perturbed, a possible mitigation may be to proceed with an alternative schedule where the affected constraint is no longer critical.

The effective identification of “super-critical” and “sub-critical” constraints allows managers to identify the driving constraints of a project. As constraints and the requirements that govern them share a many-to-one relationship (i.e. many constraints belong to one requirement), the criticality of the requirement is thus dependent upon the criticality of the constraint. Hence, a requirement is the most critical of its constraints. From the sub-critical requirements, Planners can then identify “secondary” sets of requirements to fulfil, which if perturbed, could force alternative schedules to be considered.

5.6. Illustrative Case Study on Temporal Modelling of Requirements

A simplified example of installing a steel pipe rack in an oil refinery is used to demonstrate the application of PDM++ and its corresponding constraints analysis. The entire pipe rack is divided into three phases of construction, with Phase 1 and 3 spanning a length of 8m and a height of 2.5m, and Phase 2 spanning 5m by 4m. Figure 5.12 provides a 3D perspective.

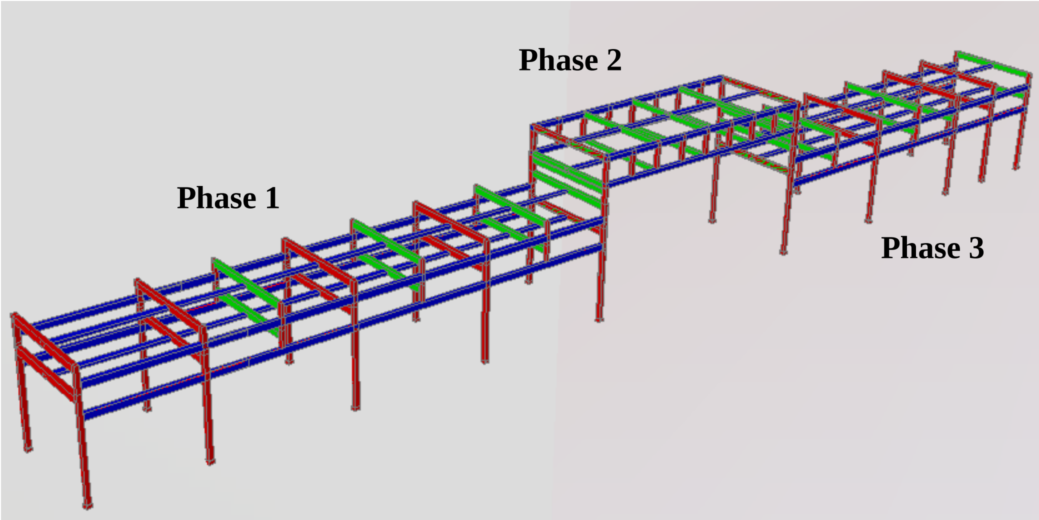


Figure 5.9. 3D Perspective of Pipe Rack Installation

Some of the pertinent project requirements are detailed as follows:

- Requirement 1: Both phases of shallow foundations are done concurrently.
- Requirement 2: Piperack columns are carried out by the same subcontractor.
- Requirement 3: For each phase, the scaffold erection can be done after one day of piperack column installation.
- Requirement 4: The first and second phase of scaffold erection must be done concurrently, before the start of the third phase.
- Requirement 5: The pipe laying must be carried out continuously.

Figure 5.10 shows the resulting project constraint network describing the above requirements graphically. The temporal constraints are indicated on the directed arcs. Directed arcs without any indications are assumed to depict the “before” constraint, which is analogous to the normal precedent constraint in PDM. The “super-critical” constraints are highlighted in bold, while the “sub-critical” constraints are marked by dashed bold lines.

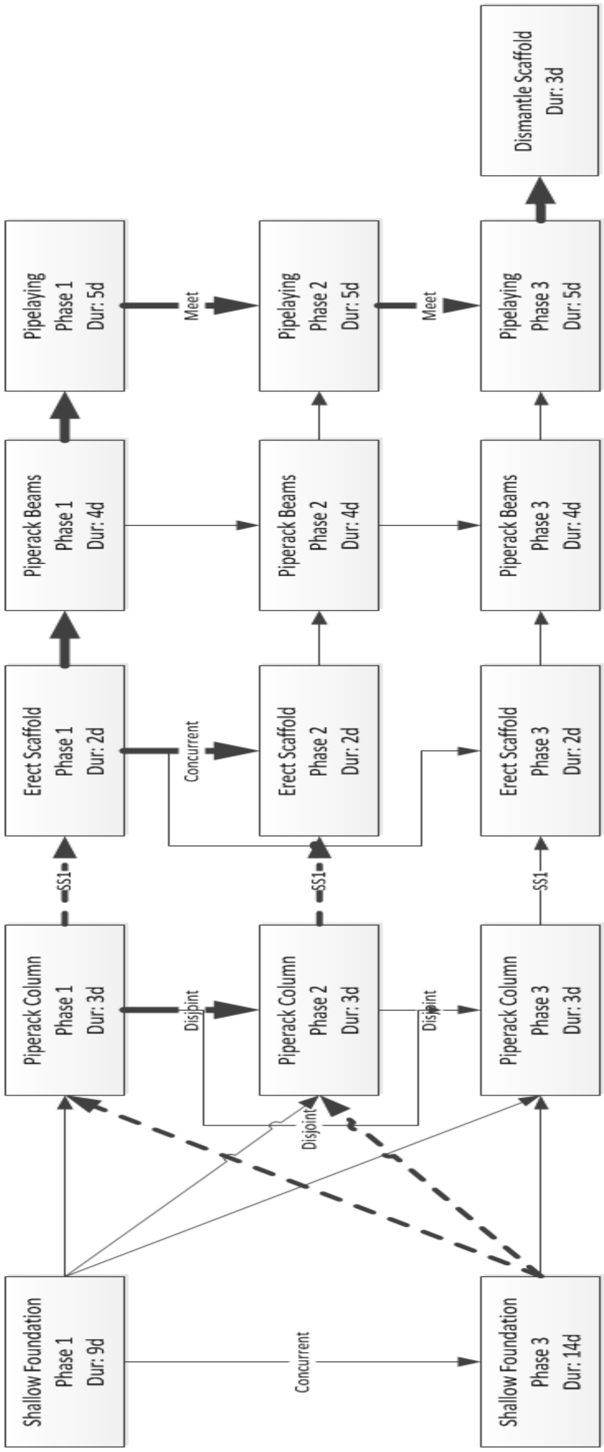


Figure 5.10. Case Study Activity Network

The results of solving the network are shown in Table 5.4. Two schedules are generated which eventually give the same project makespan of 42 days.

Table 5.4. Starting times of Activities

Activities	Schedule 1 Start Dates	Schedule 2 Start Dates
Shallow Foundation Phase 1	[0 .. 5]	[0 .. 5]
Shallow Foundation Phase 3	0	0
Piperack Column Phase 1	14	17
Piperack Column Phase 2	17	14
Piperack Column Phase 3	[20 .. 27]	[20 .. 27]
Erect Scaffold Phase 1	18	18
Erect Scaffold Phase 2	18	18
Erect Scaffold Phase 3	[21 .. 28]	[21 .. 28]
Piperack Beams Phase 1	20	20
Piperack Beams Phase 2	[24 .. 25]	[24 .. 25]
Piperack Beams Phase 3	[28 .. 30]	[28 .. 30]
Pipelaying Phase 1	24	24
Pipelaying Phase 2	29	29
Pipelaying Phase 3	34	34
Dismantle Scaffold	39	39

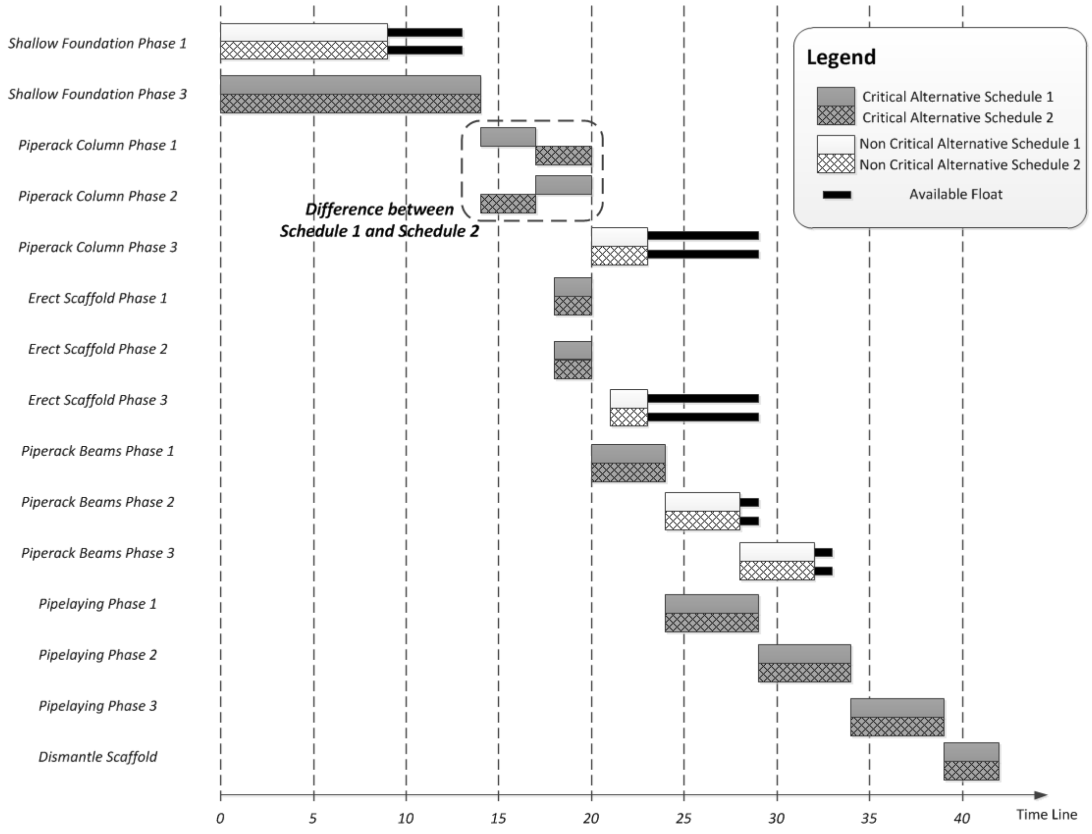


Figure 5.11. Gantt Chart of Schedule 1 and Schedule 2

From the constraints analysis, we may draw some interesting conclusions. Firstly from Figure 5.11, both "Piperack Column Phase 1" and "Piperack Column Phase 2" are super-critical as they lie on the critical path in both alternatives of Schedule 1 and Schedule 2, implying that management of the subcontractor for the piperack column installation is vital especially for the first two phases. Similarly, the work continuity requirement of the pipelaying activities also dictates the project makespan.

Secondly, another interesting observation can be made about the "super-critical" constraint set. Despite the super-criticality of some activities, some plan flexibility is still allowed to the Planner. For example, a Planner may choose to proceed with Schedule 1. In this case example, the "Piperack Column Phase 1" is critical under the consideration of both alternative schedules. However, if the activity "Piperack Column Phase 1" is delayed, then the alternative Schedule 2 may be chosen, with "Piperack Column Phase 2" commencing first.

Lastly, the concurrency of having to erect scaffolds for Phase 1 and 2 constrains the project makespan by imposing additional restrictions to the work sequence. By enforcing the concurrency constraint (Requirement 2), the second phase of the scaffold erection ("Erect Scaffold Phase 2") is unable to proceed earlier in either alternative Schedule 1 or 2, consequently making this activity supercritical. By relaxing this concurrency requirement, schedules with better makespan may be obtained. In Figure 5.12, the shortened Schedule 1 is shown with the original Schedule 1 for comparison. In the figure, it can be seen that relaxing the requirement allows the critical activities following it to be started earlier, resulting in the shorter makespan.

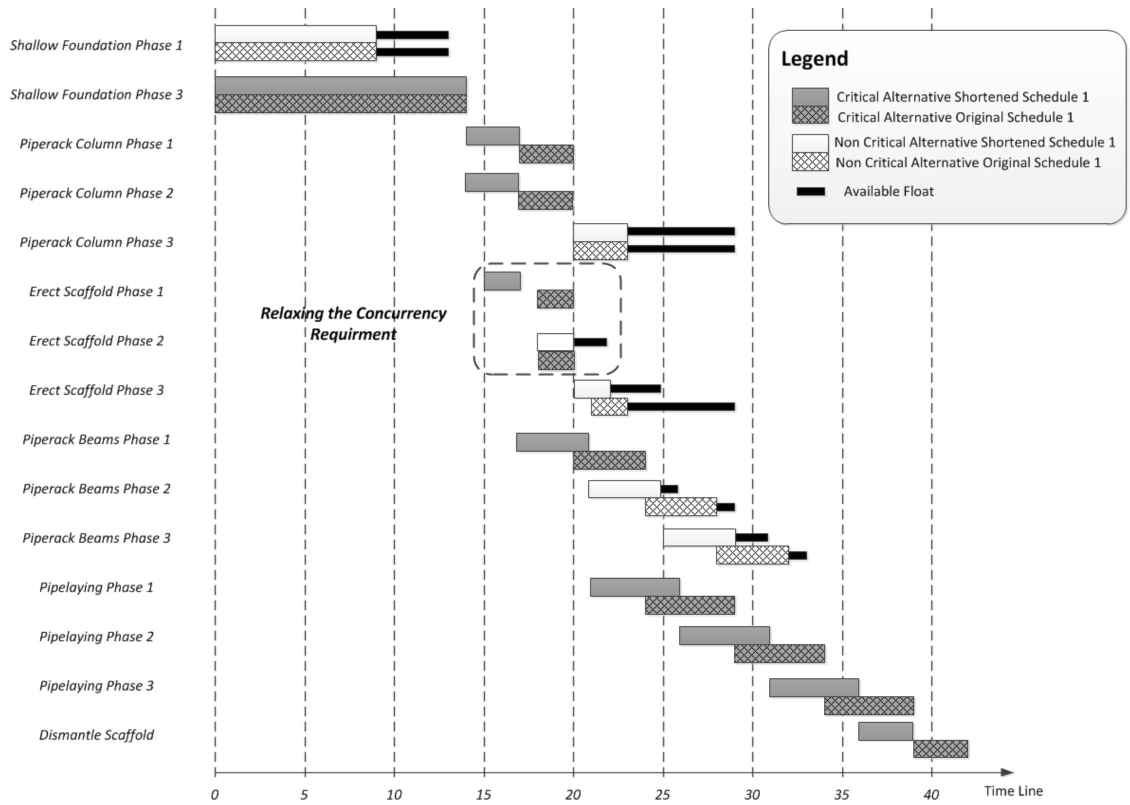


Figure 5.12. Gantt Chart showing Effect of Relaxing Concurrency Constraint

The above analysis allows the project manager to identify and analyse the critical constraints, as well as to identify the underlying construction requirement which leads to the critical constraint. The alternative schedules identified through the PDM++ model allows project managers to identify contingencies early to deal with uncertainties in the project schedule.

5.7. Concluding Remarks`

This chapter presents a modelling framework based on the temporal attributes of the construction requirements called PDM++. This framework extends upon the existing approaches in various ways:

- Allows complex representations like work continuity, overlapping activities and non-concurrency of activities to be accurately modelled within the modelling framework.
- Facilitate the modelling of interdependencies between the constraints of the activities, allowing preconditions of requirements to be captured.
- Allow hierarchical modelling or group assignments of activities.

This chapter then presents the use of logical operations in the modelling framework to address the above extensions. Further, it is shown in this chapter that the extended vocabulary can be made to mimic the Allen's temporal representation, which is a complete description of the temporal constraints between any two intervals. Finally, the chapter redefines constraint criticality in the presence of alternative scheduling.

The contribution of this Chapter is the use of the proposed modelling framework to enhance constructability through plan feasibility. This is done by first collating the temporal attributes arising from construction requirements, and then solving the model to obtain the feasible start times of each activity. This underscores the importance of construction requirements as formalising the planning considerations and assumptions for construction scheduling via a formalised knowledge representation of the construction requirement. This then forms the basis for the hypothesis of this dissertation.

Chapter 6. PDM++: Evaluation Algorithm and System Architecture

6.1. Introduction

This chapter presents the theoretical underpinning of the evaluation framework for PDM++, which will allow the temporal attributes of the construction requirements to be evaluated. The previous chapter has established the modelling framework for temporal construction requirements, and described the PDM++ semantics with enhanced expressiveness to more adequately describe the complexities of temporal construction requirements.

An evaluation approach based on the logical foundations discussed in the previous chapter is proposed, and the proof of concept of the correctness of the underlying BCSolver algorithm will be proven. The foundational logical theorems also provide the theoretical underpinnings for the system. Some of these logical theorems will be expressed mathematically using the ECL^iPS^e language. This expression then enables the implementation of the prototype system where ECL^iPS^e is used as a middleware with the proposed BCSolver, hence allowing for rapid prototyping and reuse. This system was extended to include the meta-interval representation for groups of activities.

6.2. Review of Frameworks for Conditional Constraints, Alternative Scheduling and Optional Activities

Currently, many traditional methods present only one sequence of work (or planning logic) when several feasible sequences may exist. This restricts the usefulness of the current planning models in allowing project planners to flexibly choose between alternative plans. For example, several different work methods may be available for executing a sequence of operations, while still achieving the same outcome. Other times, an activity may have several available modes of operation, usually with each mode having a different duration. This section presents a review of proposed systems which are able to distinguish the alternatives and generate feasible plans based on the available alternatives.

Additionally, optional activities, relationships and dynamic construction requirements (Chua and Yeoh, 2011) which occur due to the presence (or absence) of some stated precondition(s) are usually not adequately captured in traditional planning tools. The system should reason about the existence of such optional activities and conditional relationships, and include these in the feasible alternatives of the plan, if applicable.

Several frameworks exist which handle alternative process plans. Beck and Fox (2000) proposed a general model for describing optional activities by annotating each activity with a Boolean validity variable. In this framework, logical constraints between the variables took the form of *XorNodes*, *AndNodes* and *ActivityNodes* to model the possibility of choice among the process plans. In a similar approach, Barták and Cepek (2007) describe process alternatives using a modified directed acyclic graph model called *Parallel/Alternative Graph* where several alternative subgraphs are mapped onto one another, and specific branching nodes are specified to demarcate parallel processes from alternative processes. They later proposed tractable sub-classes of their approach with real applications, and established heuristic and edge-finding algorithms (Barták and Čepek, 2008). These two approaches explore the inclusion of optional activities in the alternatives, but assume the precedence constraints affecting the set of optional activities remains constant.

Kuster, *et al.* (2007) take a different approach from the aforementioned models by explicitly incorporating the concept of alternative activities directly into the framework of the Resource Constrained Project Scheduling Problem (RCPSP). The concept of the approach is to augment the problem with a superset of all possible elements (activities and constraints) of the problem. This concept is also adopted by the PDM++ framework; PDM++ also augments the traditional planning model with the superset of all possible elements, and is further able to represent conditional constraints via the proposed logical operators.

Another framework is the logical constraint-based scheduler found in CP Optimizer (Laborie and Rogerie, 2008). Their contribution was the incorporation of the notion of time-interval variables to intrinsically capture the concept of optionality, which bears some similarity to the IC data structure used in $ECL^{i}PS^{e}$, as well as defining logical constraints which act on the execution statuses of these variables (Laborie, *et al.*, 2009). Their logical approach bears some similarity to the one proposed in this dissertation, except that the logical constraints proposed here are generalised to infer the validity of constraints as well.

6.3. Overview of System Architectural Framework for Implementing PDM++

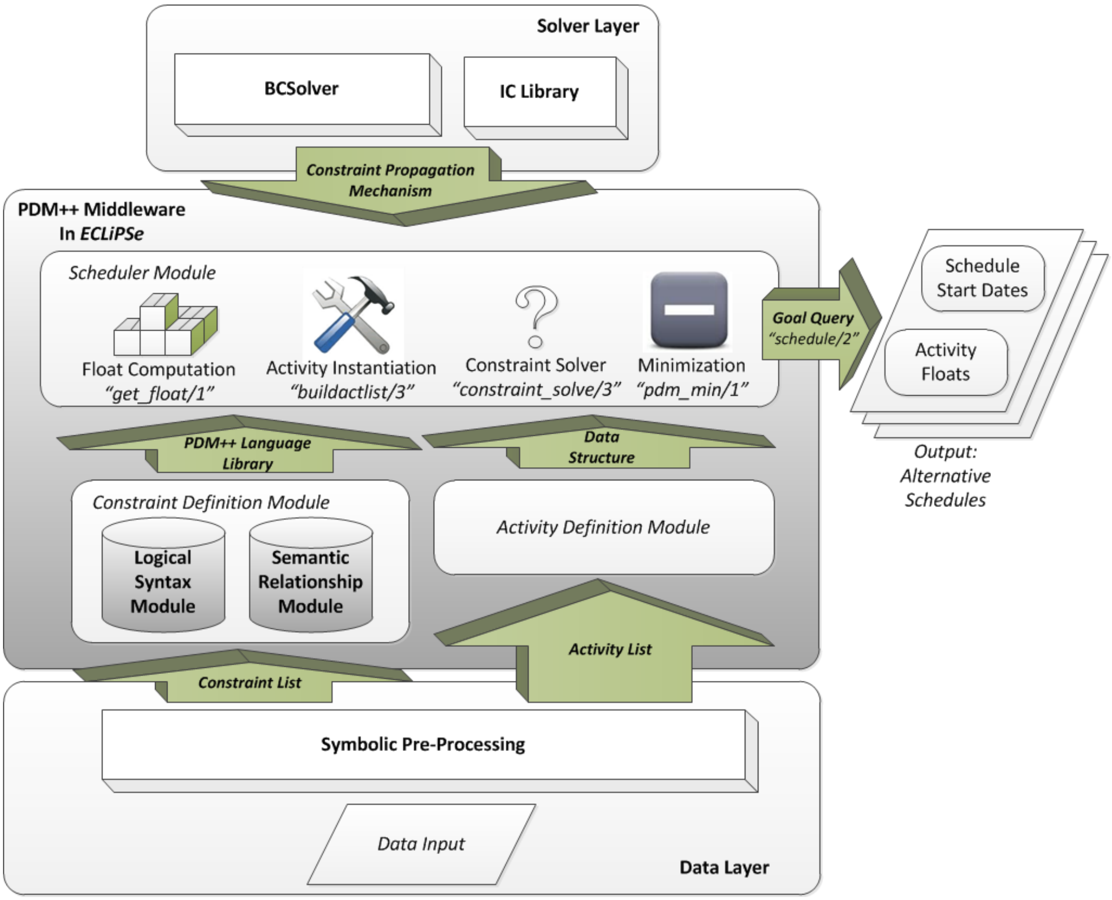


Figure 6.1. PDM++ System Architecture Framework

The implementation framework in Figure 6.1 describes a general fail-first evaluation strategy for solving PDM++ problems. The framework is designed to combine the strengths offered by a symbolic data pre-processing and numeric checking of the constraints. The reason for employing such a strategy is that the symbolic checking mechanism reasons about the structure of the problem. Additionally, the symbolic check has a lower computation complexity, but is not computationally complete. This means that symbolic checking alone is not always able to determine the feasibility of the problem. Hence, the symbolic check is employed first to determine infeasibilities early. On the other hand, the numeric solver is computationally complete, but suffers from higher complexity because it computes blindly and does not rely on inferences from the structure of the problem.

To this end, the framework proposed comprises several elements: A symbolic pre-processing algorithm within the Data Layer which will try to detect symbolic inconsistencies early in the framework, and the numeric solvers in the Solver Layer (Bounds Consistency Solver and Interval Constraint Library) which evaluates the problem. These will be covered in the following sections, along with the basic mathematical ideas behind the implementation.

A middleware implementation using ECL^iPS^e is also designed to provide an interface between the solver, data, and user. ECL^iPS^e was chosen as the system architecture platform for implementing the PDM++ modelling framework as it has native support for logical variables which is an important data structure for representing the activities in PDM++. Furthermore, its high-level language provides support for object-oriented modelling which allows for rapid software prototyping.

In general, PDM++ can be modelled as a constraint satisfaction problem {*V*, *C*} where *V* is the set of variables, with each variable having a domain to represent the starting times of the activities, and *{C}* is the set of constraints (or constraint store) including the Makespan constraint and the union of *{CS}* and *{DS}* where *{CS}* is the set of conjunctive constraints and *{DS}* is the set of disjunctive constraints. The data of the problem can be found in the data layer respectively. It can thus be represented as the following optimization problem with variables as the starting times of the activities:

Minimize $\sup\{Start_i + Dur_i\}$ where **Error! Bookmark not defined.**

$i \in \{\text{Activities}\}$ **(6.1)**

Subject to $\{C\}$ where $\{C\} := \{CS\} \cup \{DS\}$ **(6.2)**

$$\forall Start_i \in Z^+$$

The temporal relationships are represented as the constraints formulated between these variables and represented within the constraint store, *{C}*. PDM++ seeks to find all possible makespans that simultaneously satisfy the constraints; the makespan is defined as the supremum of the finish times of the set of activities. The constraints are then transferred to the solver layer for evaluation. During this process, the decision variables (start time of each activity) and their durations (known for all instances except for meta-intervals) are further assumed in the model to be positive and integer. The ECL^iPS^e middleware then retrieves the solution and presents the output.

6.4. ECL^iPS^e Middleware Layer

This section describes the main components of the ECL^iPS^e middleware layer, with the focus on the data interface through the activity and constraint lists, the scheduler module, and the generation of the data output. The middleware architecture of the PDM++ framework consists of three main modules: Activity Definition Module, Constraint Definition Module, and Scheduler Module. The Activity Definition Module defines the activity as a data structure, and this data structure is adopted by the Constraint Definition Module. The Constraint Definition Module defines the language of PDM++, which is required by the Scheduler Module to generate the results of the system query.

6.4.1 Activity and Constraint Lists

The data interface of the model comprises of a list of activity descriptions (tasks) given by a *tasks/1* clause and a list of constraints given by a *constore/1* clause:

```
% Input data: tasks/1 and constore/1
tasks([act_1/dur_1, … act_x/dur_x,…]).
constore([fs(act_1, act_2, 0), ff(act_2, act_3, 2), … , true ]).
```

where *act_x* refers to the name of Activity *X* and *dur_x* refers to the known duration of Activity *X* respectively. *constore/1* contains the new PDM++ relationships which is defined later in the Constraint Definition Module, and terminates with "true" for the entry of the last constraint. In the framework, the required meta-intervals are also defined within the model as a special type of activity.

6.4.2 Activity Definition Module

The Activity Definition Module defines the attributes of the activity as a data structure. In general, an activity (typically) represents a time interval, which is defined using the following structured data type:

```
:- local struct(act(name, type, start, duration, float, exist)).
```

In the above implementation, the structured data type representing the activity has the following six attributes:

1. *Name.* The name of the activity, which is used as a reference handler.

2. *Type.* The type of activity. An activity may be defined as a normal activity or a meta-interval. Meta-intervals are defined as a collection of activities with variable duration, which is initially initialized to the range *[0, Makespan]*.

3. *Start.* An interval representing the early and late starts of the activity.

4. *Duration.* The duration of the activity. This is assumed to be deterministic and known *a priori* for normal activities. For meta-intervals, the duration is determined after planning, and may take a possible set of values.

5. *Float.* The Total Float of the activity.

6. *Exist.* This charts the existence of an activity within the solution. This attribute may take on one of two values: Yes, or No. During initialization, this attribute is flagged as “No”, but when a constraint involving the activity is detected, the system flags this activity attribute as “Yes”.

Again, a key modelling assumption in PDM++ is that activities are assumed to be continuous (non splittable). This assumption allows the activities to be modelled solely using their start points through the linear function of Equation 5.2 (refer to Section 5.4).

6.4.3 Interval Constraint Library

In the context of this research, the constraint definition of PDM++ was implemented using the Interval Constraint library (IC Library). This IC Library is a built-in library within ECL^iPS^e, which sets the programming environment to implement finite domains of integers. Also, this is used to support the use of integer intervals as a data structure for computation (Apt, 2003).

6.4.4 Constraint Definition Module

The Constraint Definition Module defines the scope of the PDM++ Language Library through the specification of syntax and semantics. Syntax is the grammar or the mechanism of formal manipulation in a language, while semantics is the interpretation of the language. Within the Constraint Definition Module, this syntax is defined in the Logical Syntax Module and the semantics in the Semantic Relationship Module, which will be described in the next section.

The Constraint Definition Module is defined within the *ECL^{i}PS^e* middleware as *ECL^{i}PS^e* allows new constraints to be written directly without affecting the data, or the solver algorithms. This modularity allows greater flexibility for defining new constraints in PDM++ such as the meta-interval construct.

6.4.5 Scheduler Module

The Scheduler Module contains the main procedures used to evaluate a PDM++ model. The main procedures involved are *"buildactlist/3"* and *"constraint_solve/3"* procedures which are embedded within the Scheduler Module. Also included are the *"get_float/1"* procedure which evaluates the activity float based on the difference between its early and late starts, as well as the *"pdm_min/1"* which is adapted from the *CLP(R,Q)* Library and returns the minimum value of a range.

The *buildactlist/3* predicate has this structure:

```
% Predicate: buildactlist/3
buildactlist([],[],_).
buildactlist([Act_name/Dur|Rest], [Activity|Actlist], Upperbound):-
        Start#::[0..Upperbound],
        Activity = act{name:Act_name, type: Type, dur:Dur, start:Start, exist:[yes, no]},
        buildactlist(Rest, Actlist, Upperbound).
```

The purpose of the predicate is to create a list of activity definitions, and to assign the attributes of the activity definitions from the activity descriptions given in *tasks/1*.

The start variable of any one activity is an integer interval storing the early and late starts (ES and LS respectively). Upon initialization, the integer interval of the start variable is defined with ES = 0 and LS = *Upperbound*, where *Upperbound* is the sum of all activity durations and absolute values of lags in the model. The assignment of the start intervals to the individual activities is achieved through the *buildactlist/3* predicate. These integer intervals are then filtered / reduced via the constraint propagation mechanisms within the system.

The *constraint_solve/3* predicate is called to link the list of activities to the constraints, and defines (as well as solves for) the project end date (*Makespan* variable). This is achieved by defining the *Makespan* to be greater than or equal to the finish of each activity (equivalently defined to be the sum of the start and duration of the activity). The evaluation of the project end date is conditional on the existence of the activity, i.e. only activities which exist affect *Makespan*. The *Constraint* term in the *constraint_solve/3* predicate is unified with the appropriate PDM++ semantic and syntax which is unified with the definitions found in the PDM++ Language Library. The *constraint_solve/3* predicate has a structure as follows:

```
% Predicate: constraint_solve/3
constraint_solve([],_,_).
constraint_solve([Constraint|Rest_of_Constraint_list], Makespan, Activity_list):-
  (foreach(Activity, Activity_list),
  param(Makespan)
  do
          arg(dur of act, Activity, Dur),
          arg(start of act, Activity, Start),
          arg(exist of act, Activity, Exist),
          (Exist == yes ->
```

```
                Start + Dur #=< Makespan;
                true)
        ),
        Constraint,
        constraint_solve(Rest_of_Constraint_list, Makespan, Activity_list).
```

6.4.6 Generating the Output

Solving the plan requires entering the query *schedule(Activity_list, Makespan)* in the ECL^iPS^e shell environment (shown in Figure 6.2), where *Activity_list* and *Makespan* are entered as variables, generating either one of two different outcomes: First, the constraints cannot be fulfilled, and the query returns *No*, indicating no solution found. Second, the constraints are fulfilled, and the query returns *Yes*, with the evaluated project makespan and the corresponding activities with start times. Alternative feasible schedules may be generated as solutions to the problem, and these are captured with the proposed system architecture. The legend in the figure shows the format of the activity output, giving its name, type, start, duration and float respectively. This format is adopted for later figures in this chapter. In some of the following figures, the existence attribute is also appended to the format.

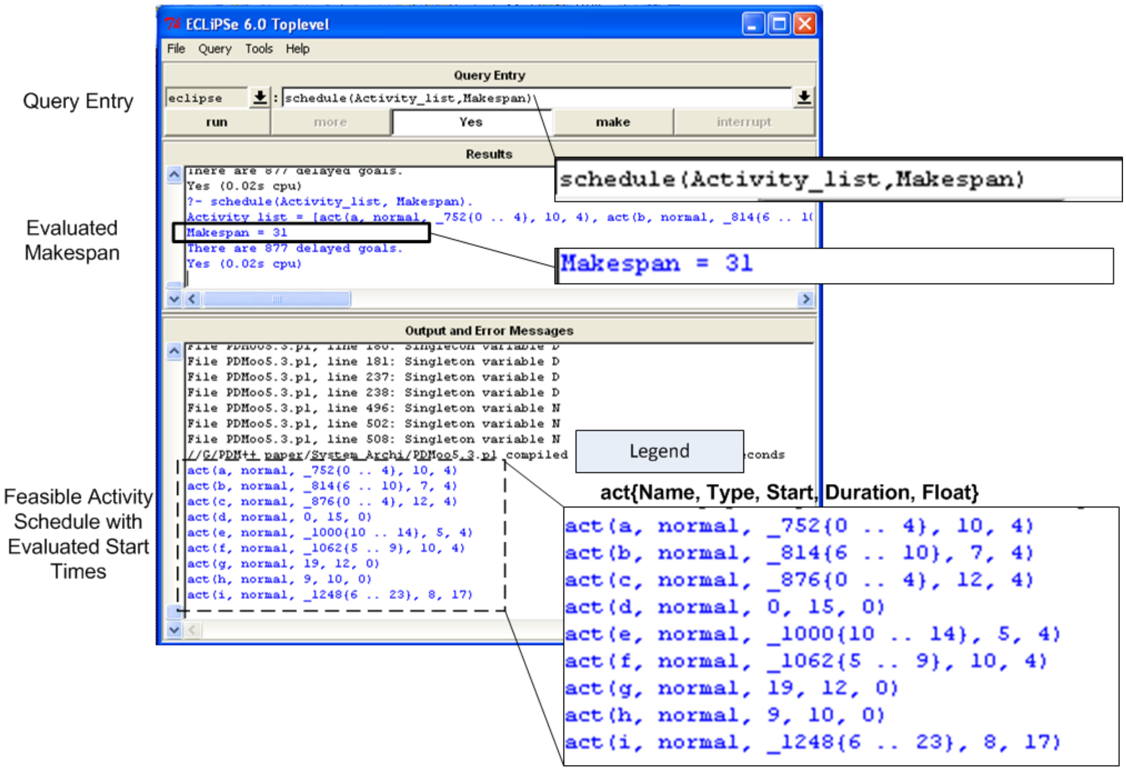


Figure 6.2. ***ECL^{i}PS^e*** **Shell Environment**

Figure 6.3 shows a traditional PDM model, and the corresponding output using the proposed PDM++ in *ECL^{i}PS^e* is shown in Figure 6.4.

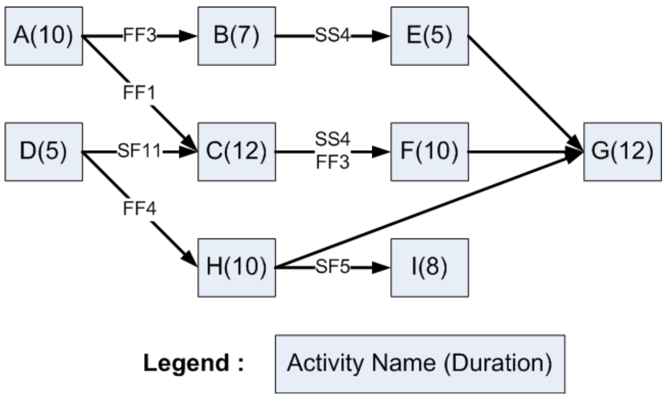


Figure 6.3. Example showing Usage of PDM++ on a PDM Network

Query:

Schedule(Activity_list, Makespan)

Output:

Makespan = 31

Activity_list =
act(a , normal , { 0 .. 4 } , 10 , 4 , yes),
act(b , normal , { 6 .. 10 } , 7 , 4 , yes),
act(c , normal , { 0 .. 4 } , 12 , 4 , yes),
act(d , normal , 0 , 15 , 0 , yes),
act(e , normal , { 10 .. 14 } , 5 , 4 , yes),
act(f , normal , { 5 .. 9 } , 10 , 4 , yes),
act(g , normal , 19 , 12 , 0 , yes),
act(h , normal , 9 , 10 , 0 , yes),
act(i , normal , { 6 .. 23 } , 8 , 17 , yes).

Legend: act(name, type, possible start time, duration, float, existence)

(a)

Query:

Schedule(Activity_list, 35)

Output:

Makespan = 35

Activity_list =
act(a , normal , { 0 .. 8 } , 10 , 8 , yes),
act(b , normal , { 6 .. 14 } , 7 , 8 , yes),
act(c , normal , { 0 .. 8 } , 12 , 8 , yes),
act(d , normal , { 0 .. 4 } , 15 , 4 , yes),
act(e , normal , { 10 .. 18 } , 5 , 8 , yes),
act(f , normal , { 5 ..13 } , 10 , 8 , yes),
act(g , normal , { 19 .. 23 } , 12 , 4 , yes),
act(h , normal , { 9 .. 13 } , 10 , 4 , yes),
act(i , normal , { 6 .. 27 } , 8 , 21 , yes).

Legend: act(name, type, possible start time, duration, float, existence)

(b)

Figure 6.4. Output of Example in ECLiPSe Shell

From Figure 6.4, the example in Figure 6.3 has an evaluated makespan of 31 days. The feasible activity schedule is also generated, where critical activities are activities with zero float (i.e. activities *d*, *g* and *h*). The non-critical activities have an interval of possible start times. For example, activity E has a start interval from 10 to 14, indicating an early start at 10 days and a late start at 14 days.

In addition to generating all feasible plans, the *schedule/2* query may also be used to check for all possible combinations of alternative schedules with makespans less than that defined in "*Makespan*". In the above example, a deadline for project completion is known to exist on Day 35. The query *schedule(Plan, 35)* is entered, and the output is obtained in Figure 6.5. Here, there are no critical activities[2], as each of the activities has an additional 4 days of float. This is because the Late Finish of the project has been delayed to Day 35. When alternative schedules are generated, alternatives with project end dates greater than the specified *Makespan* are disregarded.

[2] Criticality of activity is defined in this instance as activities with zero float. However, a more applicable definition of critical activities for such a case could be the activities that lie on the longest path.

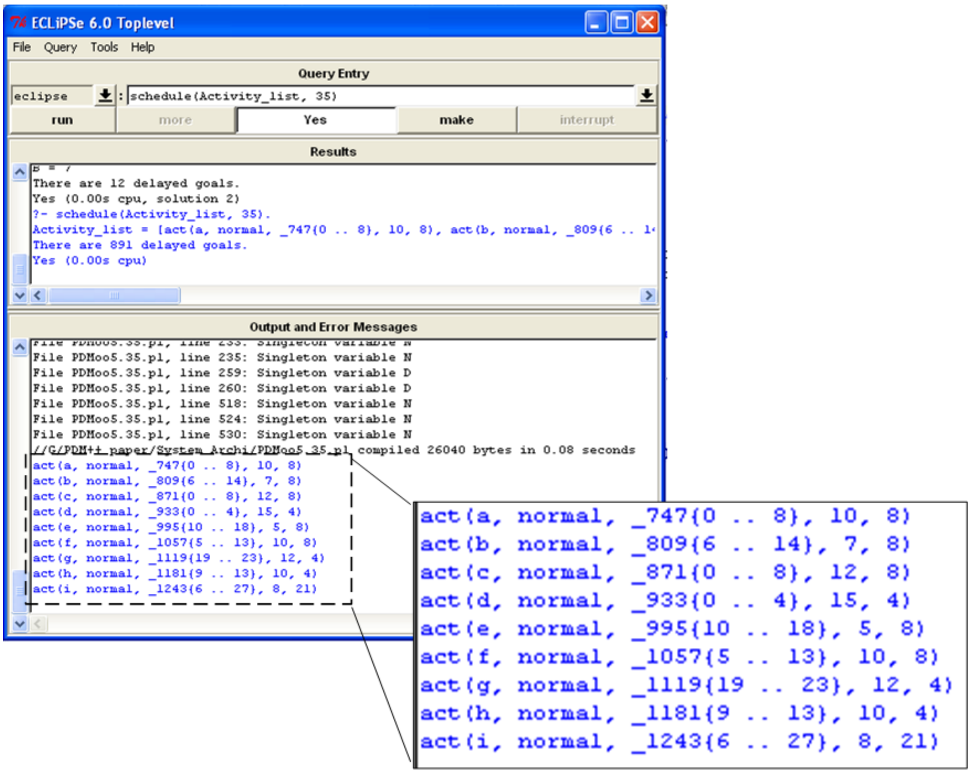


Figure 6.5. Output of Example in *ECL^{i}PS^e* for User Specified Makespan

This additional feature of the system for specifying the project makespan provides some modelling flexibility, as well as a tool for Planners to manage their projects by determining how many additional days of buffer is available to them. Consequently, this system is able to provide vital information which play a part in the decision making process during contingency planning.

6.5. PDM++ Language Library: Logical Foundations

The evaluation algorithm for PDM++ proposed in this section is based on a propositional logic foundation. This propositional logic foundation is represented by 8 basic binary relationships and 4 basic unary relationships as semantic literals and 3 levels of syntactic operators. In a logical context, syntax is the grammar or the mechanism of formal manipulation in a language, while semantics is the interpretation of the language. The structure of the language construct is shown in Figure 6.6, where each semantic literal represents a basic PDM++ relationship, and is acted on by the syntax operators denoted in the figure. For consistency of language with relevant literature, the terms “constraint” and “literals” may be used interchangeably to describe the same concept.

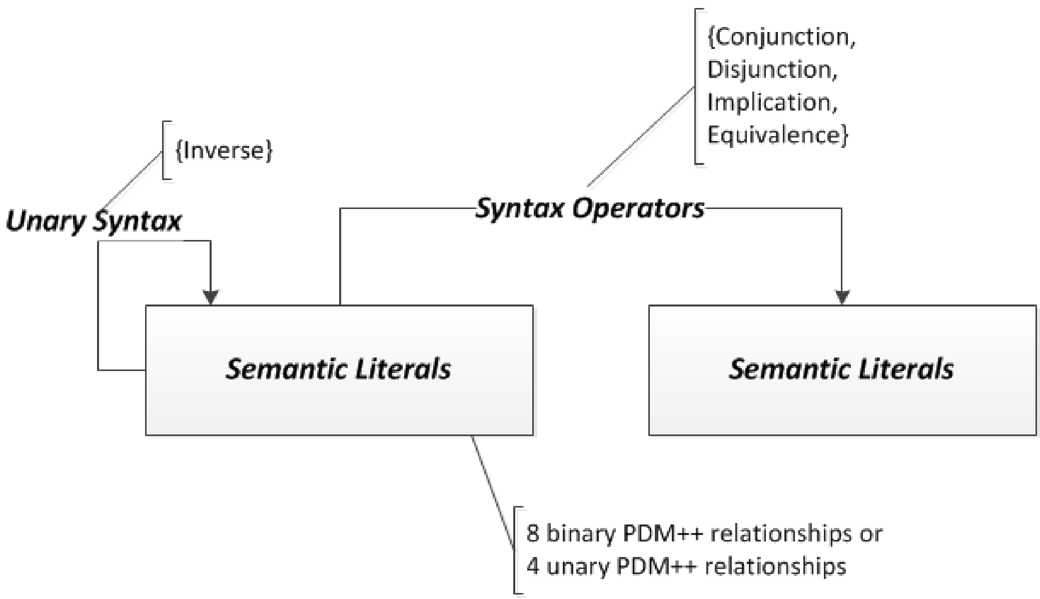


Figure 6.6. Semantics and Syntax Language Constructs of PDM++

6.5.1 Semantic Relationship Module

In PDM, the semantic relationships include the FS, FF, SF, SS relationships with minimal lag. PDM++ enriches the semantics of PDM by additionally defining a set of unary constraints, and a set of binary constraints with maximal lag, which follows the definition of Neumann and Schwindt (1997); this is the maximum amount of time that must occur between the start or end points of the activities.

Table 6.1 shows the basic binary PDM++ relationships with minimal and maximal lags. These relationships form the basic binary logical constructs which is necessary for the representation of temporal construction requirements. The maximal lags are differentiated graphically by a tilde symbol. Additionally, the same relationship is given with both its PDM++ nomenclature and its embellished PDM semantic. For example, the PDM++ relationship "X *Before(~m)* Y" is also referred to in its embellished PDM form of "X *FS(~m)* Y". In Table 6.1 and in all subsequent figures and tables within this chapter, the ECL^iPS^e representation is given with *Sx* denoting the Start of Activity *X*, *Dx* the Duration of *X*, *Sy* the Start of Activity *Y*, *Dy* the Duration of *Y* and *Lag* as the lag time.

Table 6.1. Basic Binary PDM++ Semantic

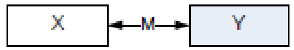
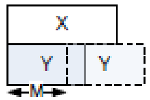
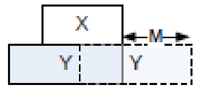
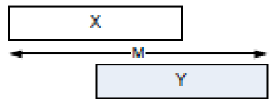
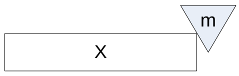
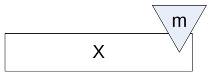
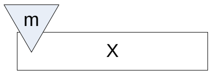
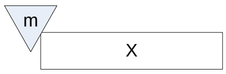
PDM++ Framework				
Minimal-lag	**Mathematical Definition with ECL^iPS^e**	**Maximal-lag**	**Mathematical Definition with ECL^iPS^e**	**Pictorial Representation**
X FS(m) Y *X Before(m) Y*	$X^- + d_x + m \le Y^-$ fs(Sx, Dx, Lag, Sy, Dy) :- Sx + Dx + Lag #=< Sy	X FS(~m) Y *X Before(~m) Y*	$X^- + d_x + m \ge Y^-$ fs_max(Sx, Dx, Lag, Sy, Dy) :- Sx + Dx + Lag #>= Sy	
X SS(m) Y *X Starts(m) Y*	$X^- + m \le Y^-$ ss(Sx, Dx, Lag, Sy, Dy) :- Sx + Lag #=< Sy	X SS(~m) Y *X Starts(~m) Y*	$X^- + m \ge Y^-$ ss_max(Sx, Dx, Lag, Sy, Dy) :- Sx + Lag #>= Sy	
X FF(m) Y *X Finishes(m) Y*	$X^- + d_x + m \le Y^- + d_y$ sf(Sx, Dx, Lag, Sy, Dy) :- Sx + Lag #=< Sy + Dy	X FF(~m) Y *X Finishes(~m) Y*	$X^- + d_x + m \ge Y^- + d_y$ sf_max(Sx, Dx, Lag, Sy, Dy) :- Sx + Lag #>= Sy + Dy	
X SF(m) Y *X Start-Finish(m) Y*	$X^- + m \le Y^- + d_y$ ff(Sx, Dx, Lag, Sy, Dy) :- Sx + Dx + Lag #=< Sy + Dy	X SF(~m) Y *X Start-Finish(~m) Y*	$X^- + m \ge Y^- + d_y$ ff_max(Sx, Dx, Lag, Sy, Dy) :- Sx + Dx + Lag #>= Sy + Dy	

Table 6.2 shows the unary constraints of PDM++, with its associated mathematical definitions and corresponding *ECL^iPS^e* representation (which will be elaborated in later sections). In this case, *m* represent due dates. For instance, X *Due-Before(m)* means that activity *X* must finish before Day *m*. Implicitly, *m* must be large enough that X^- does not violate the project start date (Day 0). These unary constraints form the basis for the complete representation of temporal constraints on a single activity.

Table 6.2. Basic Unary PDM++ Semantic

Semantic	Mathematical Definition with ECL^iPS^e	Pictorial Representation
X Due-Before (*m*)	$X^- + d_X \le m$ **due_before(Sx, Dx, M) :- Sx + Dx #=< M**	
X Due-After (*m*)	$X^- + d_X \ge m$ **due_after(Sx, Dx, M) :- Sx + Dx #>= M**	
X Start-Before (*m*)	$X^- \le m$ **start_before(Sx, Dx, M) :- Sx #=< M**	
X Start-After (*m*)	$X^- \ge m$ **start_after(Sx, Dx, M) :- Sx #>= M**	

The basic semantics for the relationships in PDM++ encompass both binary and unary constraints, and allow for greater expression of planning considerations, leading to an enhanced representation of construction requirements. From a propositional logical perspective, the above semantic relationships can be recognized as literals which cannot be further decomposed, e.g. "$X^- + d_X + m \le Y^-$". The term semantic literal is used in this dissertation to denote these atomic literals.

PDM++ may also be graphically represented as a constraint network $G = (V(D), E)$ where vertices V represent the construction activities each with an individual domain D being the activity start times, while edges E represents the temporal logic constraints/relationships defined between activities. Figure 6.7 shows the graphical representation of binary and unary relations in the PDM++ network. For example, the binary relationship in Figure 6.7 shows a Before(*0*) relationship (refer to Table 6.1) between activities B and A. Similarly, the unary relationship in Figure 6.7 shows a *Due-before(0)* constraint acting on activity A.

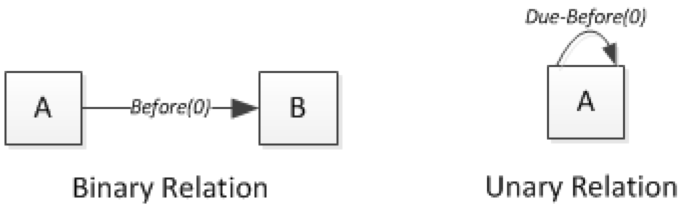


Figure 6.7. Graphical Representations of PDM++

6.5.2 Logical Syntax Module of PDM++

The syntax represents a set of higher-level operations that act on the semantics. Through the combination of semantic and syntax, Planners will be able to represent the complex temporal relationships given in the previous chapter, which commonly arise in the description of the temporal attributes of construction requirements. In the PDM++ modelling framework, the basic syntax has been defined as operations closely following the logical Boolean operations. The syntax has been differentiated into three levels of operations (Table 6.3): The basic level is comprised of the operators Conjunction ($\wedge$) and Disjunction ($\vee$), the intermediate level includes the Inverse ($\neg$) operator and the last level involves the conditional operators Implication ($\rightarrow$) and Equivalence ($\Leftrightarrow$) which are derived from the first two levels.

For the following, it suffices for the reader to understand that each of these operators is analogous to its corresponding syntax operator in propositional logic. For example, the inverse operator is similar in principle and operation to the negation operator in propositional logic, while the implication and equivalence operators are similar to the logical implication and logical equivalence respectively.

Table 6.3. 3 levels of Syntax Operations in PDM++

Level	Operators	*ECLⁱPS^e* Implementation
Basic	Conjunction ($\wedge$)	**conj(A, B) :- A, B.**
	Disjunction ($\vee$)	**disj(A, B) :- A; B.**
Intermediate	Inverse ($\neg$) Conjunction (De Morgan's Law)	**inv(conj(A, B)) :- disj(inv(A), inv(B)).**
	Inverse ($\neg$) Disjunction (De Morgan's Law)	**inv(disj(A, B)) :- conj(inv(A), inv(B)).**
Derived	Implication ($\rightarrow$)	**imply(A, B) :- inv(A), !; B.**
	Equivalence ($\Leftrightarrow$)	**equiv(A,B) :- (conj(A, B), ! ; conj(inv(A), inv(B))).**

6.5.2.1 Basic Syntax Operators: Conjunction and Disjunction

In Table 6.3, three levels of operators are presented. The derived operators are formulated from the combinations of the basic and intermediate operators. Here, A and B refer to variables, and may take the form of any of the unary or binary constraints of the Semantic Relationship Module.

The conjunction and disjunction operators form part of the Logical Syntax Module found in the PDM++ framework. The conjunction operator ($\wedge$) is used to define the scenario where two constraints are to be satisfied (true) simultaneously. The conjunction operator is the default (and implied) syntax in the traditional PDM modelling framework; all the constraints specified in a PDM model must be simultaneously satisfied. The conjunction operator is implemented with the *conj/2* clause in Table 6.3.

The disjunction operator ($\vee$) is used to define the situation when either one of the constraints specified is true. The introduction of the disjunction operator allows for the availability of alternative sequences of plans. A choice point is created when the disjunction operator is encountered, and a backtracking mechanism is initiated along the two paths created by the choice point. The implementation of *disj/2* is shown in Table 6.3.

The disjunction operator may be used to model two alternate sequences of work. For example, a pipe is to be installed in three segments, S1 to S2 to S3. It is however, possible to do the same installation in the opposite sequence of work (i.e. from S3 to S2 to S1). Such a relationship can be represented with the following equation.

$$((S1\ FS(0)\ S2) \wedge (S2\ FS(0)\ S3)) \vee ((S3\ FS(0)\ S2) \wedge (S2\ FS(0)\ S1)) \quad \textbf{(6.3)}$$

The corresponding *ECL^{i}PS^e* implementation is also presented as follows to show the transpositions of mathematical formulae to implemented code, where the segment activity is represented by *sX*, and the activity duration as *sX_dur* for the segment, X:

```
disj(
        conj( fs( s1, s1_dur, 0, s2, s2_dur ), fs( s2, s2_dur, 0, s3, s3_dur ) ),
        conj( fs( s3, s3_dur, 0, s2, s2_dur ), fs( s2, s2_dur, 0, s1, s1_dur ) )
)
```

6.5.2.2 Intermediate Syntax Operators: Inverse Operators

The inverse operator ($\neg$) is a key innovation of the system, and is used to depict an equivalent constraint under which the original semantic or meaning of the original constraint is not true. It lays the foundation for defining the conditional operators of implication and equivalence, by clearly defining constraint invalidity. Consequently, it enhances the system's capabilities over traditional AI Planners by enabling the system to reason about constraints.

The binary PDM++ constraints with maximal lags and the constraints with minimal lags are semantic inverses of each other. For example, the following equation shows the inverse operation on *X FS(m) Y*, resulting in *X FS(~(m-1)) Y*. The lag in the maximal relationship is decreased by 1 in Equation 6.4 due to the finite integer domain properties on which the system is built on. The inverse operations on the other semantic literals may be similarly derived.

$$\begin{aligned} \neg\ (X\ FS(m)\ Y) &= \neg\ (Sx + Dx + m \leq Sy) \\ &= Sx + Dx + m > Sy \\ &= Sx + Dx + (m-1) \geq Sy \\ &= X\ FS(\sim(m-1))\ Y \end{aligned} \qquad (6.4)$$

Equation 6.4 may be expressed in *ECL^{i}PS^e* using the following *inv/1* representation (the representation of other semantic literals is also given for completeness):

```
inv( fs( Sx, Dx, Lag, Sy, Dy ) ) :- Newlag is Lag -1, fs_max( Sx, Dx, Newlag, Sy, Dy ) ).
inv(fs_max(Sx, Dx, Lag, Sy, Dy)) :- fs(Sx, Dx, Newlag, Sy, Dy), Newlag is Lag + 1.
inv(ss(Sx, Dx, Lag, Sy, Dy)) :- ss_max(Start Sx, Dx, Newlag, Sy, Dy), Newlag is Lag - 1.
inv(ss_max(Sx, Dx, Lag, Sy, Dy)) :- ss(Start Sx, Dx, Newlag, Sy, Dy), Newlag is Lag + 1.
inv(ff(Sx, Dx, Lag, Sy, Dy)) :- ff_max(Sx, Dx, Newlag, Sy, Dy), Newlag is Lag - 1.
inv(ff_max(Sx, Dx, Lag, Sy, Dy)) :- ff(Sx, Dx, Newlag, Sy, Dy), Newlag is Lag + 1.
inv(sf(Sx, Dx, Lag, Sy, Dy)) :- sf_max(Sx, Dx, Newlag, Sy, Dy), Newlag is Lag - 1.
inv(sf_max(Sx, Dx, Lag, Sy, Dy)) :- sf(Sx, Dx, Newlag, Sy, Dy), Newlag is Lag + 1.
```

Similarly, the unary relationships of *Due-Before(m)* and *Due-After(m)*, and of *Start-Before(m)* and *Start-After(m)* are inverse relationships of each other, and may also be derived in the same manner as the binary relationships.

```
inv(due_before(Sx, Dx, Lag)) :- due_after(Sx, Dx, Newlag), Newlag is Lag + 1.
inv(due_after(Sx, Dx, Lag)) :- due_before(Sx, Dx, Newlag), Newlag is Lag - 1.
inv(start_before(Sx, Dx, Lag)) :- start_after(Sx, Dx, Newlag), Newlag is Lag + 1.
inv(start_after(Sx, Dx, Lag)) :- start_before(Sx, Dx, Newlag), Newlag is Lag - 1.
```

The intermediate inverse operator may also act on the basic operators of conjunction and disjunction. This may be necessary when the inverse operation is employed on more complex PDM++ constraints. Under such circumstances, the system applies De Morgan's Laws to the complex constraint "eagerly" (this means that the system automatically does so when the occasion arises), decomposing the original constraint into a series of disjunctions or conjunctions of constraints. The implementation of De Morgan's Laws in *ECL^i^PS^e^* is also depicted in Table 6.3.

For example, the inverse of the following constraint between Activities A and B can be evaluated using De Morgan's Laws: (*A* Concurrent *B).* This constraint represents a concurrency between A and B where A is contained within B, with the start of A less than the start of B, and the finish of A greater than the finish of B. A pictorial depiction of this can be found in Table 5.1. The application of the inverse states the conditions under which the concurrency is no longer valid: in this example, this being one of two conditions shown by the disjunction: B starts earlier than A, or it finishes later. The following shows the transformation using De Morgan's Laws:

$$\neg\ (\text{A}\ \textit{Concurrent}\ \text{B}) = \neg\ ((\text{A}\ \textit{SS(\sim 0)}\ \text{B}) \wedge (\text{A}\ \textit{FF(0)}\ \text{B}))$$

$$= \neg(\text{A}\ \textit{SS(\sim 0)}\ \text{B}) \vee \neg(\text{A}\ \textit{FF(0)}\ \text{B})$$

$$= (\text{A}\ \textit{SS(1)}\ \text{B}) \vee (\text{A}\ \textit{FF(\sim -1)}\ \text{B}) \qquad \textbf{(6.5)}$$

The equivalent implementation is ECL^iPS^e is then shown as follows:

inv(conj(ss_max(A, B, 0), ff(A, B, 0)))
= disj(inv(ss_max(A, B, 0)), inv(ff(A, B, 0)))
= disj(ss(B, A, 1), ff_max(A, B, -1))

6.5.2.3 Derived Syntax Operators: Implication and Equivalence Operators

The derived syntax operations of implication and equivalence are built on the basic and intermediate syntax operations. In addition to the basic and intermediate operators, the cut facility (!) is also employed to control the search mechanism within ECL^iPS^e. The cut facility is used to prune the search tree, as well as implement the meaning of the implication operator and equivalence operator where the false precondition leads to an empty implication or equivalence.

For example, the cut facility used in the following circumstance: **A, !; B**, would exhibit the behaviour that when the variable A is true, the system ignores the rest of the expression after the cut. When the inverse of clause A is true, the cut operator is reached, it commits the system to *inv*(A) without evaluating B. However, when the inverse of A fails, the system backtracks, and tries clause B. The two derived syntax operators of Implication and Equivalence are shown in Table 6.3.

The Implication operator ($\rightarrow$) is used to model conditional dependencies between constraints. "I $\rightarrow$ J" means that if I is true, then J is implied to be true. However, if I is false, then J may be either true or false. The "Implication" operator can be used for activities which become constrained only when certain pre-conditions occur, e.g. , "If Activity X finishes before Y, then Z can start after X", expressed as:

$$(X\ \textit{Before}\ Y) \rightarrow (Z\ \textit{After}\ X) \qquad \textbf{(6.6)}$$

In the above example, Z may or may not start after X if X does not finish before Y. Such conditional interdependency can be used for activities which become constrained only when certain pre-conditions occur. The Implication operator is implemented by using the *imply/2* clause in Table 6.3 where *imply/2* is defined as the disjunction of B, and the inverse of A: $\neg A \vee B$. In *ECL^i^PS^e^*, this is given as:

imply(A, B) :- inv(A), !; B.

The Equivalence Operator (↔) is used to model a form of causality which semantically corresponds to "If and only If" statements. Such "Equivalence" operators can be used to show situations where if the pre-condition is true, then the post-condition must also be true and vice versa. For example, "Activity X starts before Y if and only if Z finishes before W", expressed as

$$X\ FS(0)\ Y \leftrightarrow Z\ FS(0)\ W \quad \textbf{(6.7)}$$

The Equivalence operator is implemented in the system using the *equiv/2* clause (see Table 6.3), which is the disjunction of two cases: one case being that A and B are true, and the other being that the inverses of A and of B are true. The equivalence operator corresponds to the "XNOR" Boolean operator, and may be represented mathematically as $(A \wedge B) \vee (\neg A \wedge \neg B)$. The cut operator in *equiv/2* biases the system to investigate the case where A and B are both true before the latter.

equiv(A,B) :- conj(A, B), !; conj(inv(A), inv(B)).

6.6. Symbolic Pre-processing and BCSolver Algorithms

The final two remaining key components of the system architecture of Figure 6.1 are the symbolic pre-processing and BCSolver Algorithm. This section will discuss these two elements.

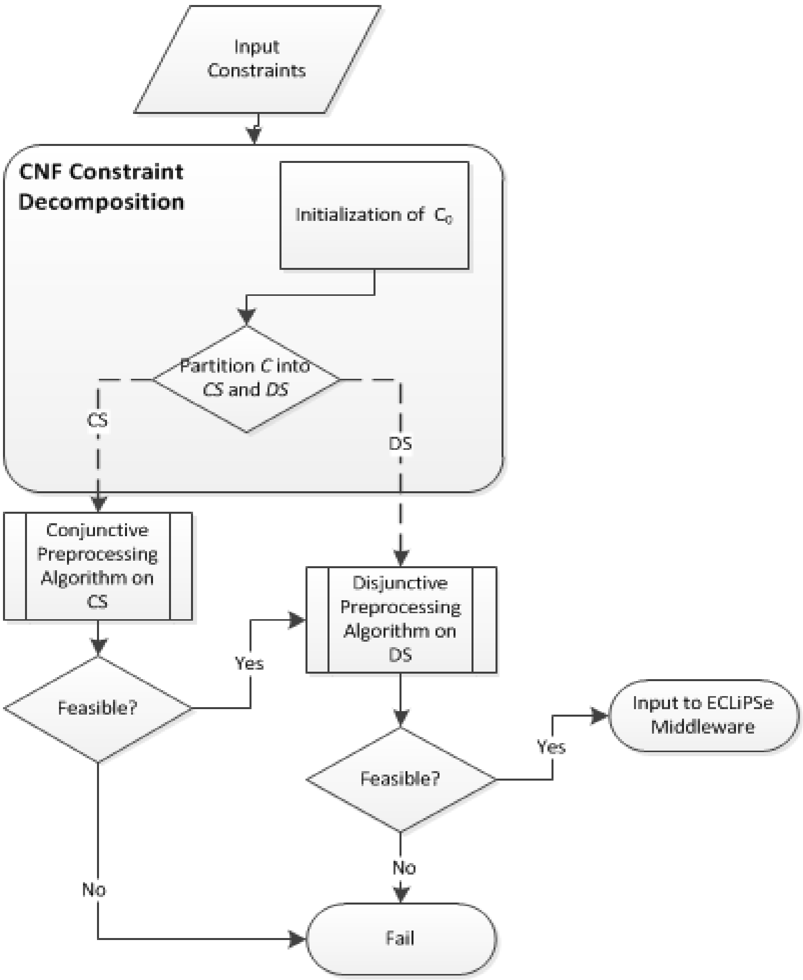


Figure 6.8 Symbolic Pre-Processing Flowchart

Figure 6.8 shows the symbolic pre-processing flowchart which uses the structure of the constraints to ensure constraint consistency by inferring infeasible combinations of the constraints early. This pre-processing flowchart is decomposed into the CNF Constraint Decomposition subroutine, and the conjunctive/disjunctive pre-processing algorithm.

6.6.1 Generating *CNF* Constraint Set and Initialization

The framework starts with the generation of the conjunctive constraint set *{CS}* and disjunctive constraint set *{DS}* from an initial list of constraints C_0. C_0 is initialized to decompose the PDM++ temporal relationships from the construction requirements into a combination of the eight basic binary constraints and four unary constraints making up the logical semantic (from Table 6.1 and Table 6.2 respectively), with the logical syntax. The decomposition will change the original temporal relationships into combinations of the basic PDM++ semantic literals.

```
Input: Initial List of Temporal Relationships C0
       Conjunctive Constraint Set CS
       Disjunctive Constraint Set, DS
Output: Transform the initial list of temporal relationships into its equivalent PDM++
literals and partition the literals into CS and DS

For each Temporal Relationships C(i) ∈ C0
       Decompose Relationship into equivalent PDM++ semantic literal

       If C(i) is atomic
              CS ← C(i) ∪ CS
       ElseIf C(i) is negative atomic
              Inverse C(i)
              CS ← C(i) ∪ CS
       ElseIf C(i) is not atomic
              V ← CNF(C(i))                    //Convert C(i) to Conjunctive
                                               Normal Form
              Decompose V ={V(1), V(2)...}     //where V is (V(1)∧V(2)∧...)

              For all V(i) in V,
              If V(i) is atomic
                     CS ← V(i) ∪ CS
              Else
                     DS ← V(i) ∪ DS
              EndIf
              Next
       EndIf
Next

End
```

Figure 6.9. PseudoCode for Initialization

If the constraint is a semantic literal, it may not be further decomposed, and this is added to the constraint set *{CS}*. For example, the PDM++ relationship (A *Contains* B) can be broken down into the following two relationships: (A *Starts* B), and (A

Finishes(~0) B). Both these two relationships are then added to *{C}*. Disjunctive clauses are retained, but transformed into the PDM++ semantic and syntax as follows in the given example: (A *Cannot-Occur* [L,U]) becomes a disjunction of (A *Due-before* L) or (A *Start-after* U). These are added to *{DS}*.

During the decomposition process, one aim of the framework is to achieve a Conjunctive Normal Form (CNF) for all complex (non-atomic) logical representations. In the PDM++ model, CNF is defined as either single PDM++ semantic, or a conjunction of disjunctions of PDM++ semantic literals. Every propositional formula can be converted into CNF by performing the following sequence of steps to preserve logical equivalence (adapted from (Ben-Ari, 1993)):

1. Push all negations inwards by using De Morgan's Theorem.

2. Eliminate double negations (inverse relationships) until only disjunctions and conjunctive literals remain.

3. Use the distributive laws in Equation 6.8 and Equation 6.9 to eliminate conjunctions within disjunctions.

$$\boldsymbol{A} \vee (\boldsymbol{B} \wedge \boldsymbol{C}) \leftrightarrow (\boldsymbol{A} \vee \boldsymbol{B}) \wedge (\boldsymbol{A} \vee \boldsymbol{C}) \quad (6.8)$$

$$(\boldsymbol{A} \wedge \boldsymbol{B}) \vee \boldsymbol{C} \leftrightarrow (\boldsymbol{A} \vee \boldsymbol{C}) \wedge (\boldsymbol{B} \vee \boldsymbol{C}) \quad (6.9)$$

4. Substitute negations with the equivalent inverse PDM++ relationships.

The Initialization Algorithm implemented and shown above has a run-time complexity of $O(r2^p)$ where r is the number of temporal relationships of the problem and p is the number of semantic literals in a relationship. In practice, the value of p is unlikely to be large, with r being in the order of hundreds for large projects, and p not being practically larger than twenty. This value is deduced from practice where the application of a key resource to major activities is unlikely to exceed this number during detailed construction planning. Furthermore, the Planner is usually able to adopt a top-down approach in developing a plan, and hence is able to determine the granularity and level of specifications and requirements for consideration within the model. Additionally, the hierarchical breakdown of plans alleviates the complexity of the problem by dealing with the plan piecemeal. Hence, despite the exponential blow-up due to the CNF conversion, it is unlikely that the number of literals will be prohibitive.

6.6.2 Symbolic Pre-Processing Algorithms

6.6.2.1 Conjunctive Pre-Processing Algorithm

The Conjunctive Pre-Processing Algorithm proposed in this section is a symbolic checking mechanism of the PDM++ semantic literals. Through the pre-processing process prior to solving, it potentially makes it easier for the later algorithms to solve. The following pre-processing algorithm achieves this through a set of rules which are proposed in this research that either reduce the number of constraints in the conjunctive constraint set, or detect inconsistencies early.

For binary literals involving activities X and Y, it is possible to re-express the literals in the following form of Equation 6.10. Let *t* be the sum of constants, including the durations of X and Y, as well as any lag values.

$$\boldsymbol{Y^{-} - X^{-} \blacksquare\, t \quad where\ \blacksquare \in \{\leq, \geq\}} \qquad \textbf{(6.10)}$$

We can now present the following rules when comparing two literals sharing the same activities:

Rule C1. If $\{\blacksquare_1, \blacksquare_2\} := \geq$ ***, retain the literal with the larger value of t.***

Given the following two constraints involving S and T: $T^- - S^- \geq 2$ and $T^- - S^- \geq 3$, satisfying $T^- - S^- \geq 3$ with any valid value of S and T automatically allows $T^- - S^- \geq 2$ to be simultaneously satisfied. $T^- - S^- \geq 2$ may be removed without any consequence on the system.

Rule C2. If $\{\blacksquare_1, \blacksquare_2\} := \leq$ ***, retain the literal with the smaller value of t.***

Similarly, consider the following two constraints involving S and T: $T^- - S^- \leq 3$ and $T^- - S^- \leq 2$. Then choosing valid values of S and T to satisfy $T^- - S^- \leq 2$ will also simultaneously satisfy $T^- - S^- \leq 3$. It is possible to remove $T^- - S^- \leq 3$ from the system.

Rule C3. If $\blacksquare_1 := \leq$; $\blacksquare_2 := \geq$; $t_1 \leq t_2$ ***or*** $\blacksquare_1 := \geq$; $\blacksquare_2 := \leq$; $t_1 \geq t_2$ ***, then an inconsistency is detected.***

An example of an inconsistency arises in the following example involving S and T: $T^- - S^- \leq 3$ and $T^- - S^- \geq 5$. If the valid values of S and T satisfy one constraint, it is not able to simultaneously satisfy the other.

The Conjunctive Pre-Processing Algorithm runs in $O(n^2)$ polynomial time, but the pruning power of the algorithm is weak, meaning that the algorithm is not able to infer a greater number of infeasible solutions as compared to other algorithms; however, it achieves a faster algorithmic processing time.

6.6.2.2 Disjunctive Pre-Processing Algorithm

The aim of the Disjunctive Pre-Processing Algorithm is to reduce the number of constraints in the constraint set symbolically, achieving a reduced Disjunctive Set *{DS}* which is equivalent to a smaller subset of disjuncts. This is achieved by the following rules, which are adapted from Stergiou and Koubarakis (1998) and Stergiou and Koubarakis (2000). In these rules, we will assume the following naming conventions: Let $DS(i) \in \{DS\}$ and further, let *DS(i,j)* be the *jth* Disjunct of *DS(i).*

Subsumption Rule 1. If $\{\blacksquare_1, \blacksquare_2\} := \geq$ ***, retain literal with smaller value of t.***

Subsumption Rule 2. If $\{\blacksquare_1, \blacksquare_2\} := \leq$ ***, retain literal with larger value of t.***

The above subsumption rules are introduced to determine the conditions when a one of two constituent disjunctive literals can be ignored from the system under the disjunctive operator. Both rules are "inverses" of Rule C1 and Rule C2.

Rule D1. If DS(i) contains a disjunct DS(i,j) that is subsumed by another disjunct DS(i,k) which is also in DS(i), then remove DS(i,j).

As an example, consider the following disjunctive constraint, *DS(1)*: $(T^- - S^- \geq 2) \vee (T^- - S^- \geq 3)$. Let *DS(1,1)* be the 1st disjunct of *DS(1)* and *DS(1,2)* be the 2nd disjunct of *DS(1)*. Then $DS(1,1) = (T^- - S^- \geq 2)$ and $DS(1,2) = (T^- - S^- \geq 3)$. Applying Subsumption Rule 1, *DS(1,1)* can be said to subsume *DS(1,2)*.

Rule D2. If DS(i) contains a disjunct DS(i,j) that is subsumed by a constraint $\boldsymbol{CS(k)} \in \{\boldsymbol{CS}\}$***, then remove DS(i).***

When a disjunct *DS(i,j)* is subsumed by a constraint *CS(k)*, then *DS(i)* can always be instantiated by selecting *DS(i,j)*. Hence, *DS(i)* will be vacuously true under all circumstances.

Rule D3. If any DS(i,j) is inconsistent relative to {CS} using Rule C3, then remove DS(i,j) from DS(i).

As an example, consider the following constraints involving R, S and T: $CS(1) = (T^- - S^- \geq 2)$, $DS(2,1) = (T^- - S^- \leq 1)$ and $DS(2,2) = (T^- - R^- \geq 3)$. *CS(1)* and *DS(2,1)* are inconsistent with each other, and hence *DS(2,1)* can be discarded, retaining only *DS(2,2)*.

The run time complexity of the Disjunctive Pre-Processing Algorithm is $O(2|\{CS)||\{DS\}|^2|\{j\}|^2)$, where $|\{CS\}|$ is the number of constraints in the Conjunctive Set, $|\{DS\}|$ is the number of constraints in the Disjunctive Set, and $\{j\}$ is the maximum number of disjuncts.

6.6.3 BCSolver Algorithm

The BCSolver Algorithm is adapted from the Bounds Consistency Algorithm (Jaffar, *et al.*, 1994) for the PDM++ framework. In the literature, there exist several versions of Bounds Consistency, based on differing views of the definition of being Bounds Consistent. Despite the differences, the basis remains the same, which is to relax the consistency requirement so that only the lower and upper bounds of the domain of each variable is narrowed. For the purposes of this research, we adopt the following definition for Bounds Consistency (Choi, *et al.*, 2006):

"A constraint is Bounds Consistent if for each bound of the domain of a variable there is an integer support for the values of the domain of the other variables occurring in the same constraint."

The BCSolver algorithm can be broken down into three parts: Activity initialization, Iteration and Revise. The Activity initialization is the assignment of the domain to the start times of the activities. Upon initialization of the domain, an interval *[lb,ub]* is assigned to the start times. *lb* represents the lower bound, and is the earliest feasible start time; *ub* represents the upper bound, and is the latest feasible start time.

```
Input: List of PDM++ Semantic Literals (Constraints), L
       Set of Activities X with respective lags m, duration d_X and domain of
       [lb, ub] indicating feasible start times
Output: List of Activities with reduced domains

For each l(j) ∈ L
       Q ← Q ∪ l(j)                          //initialize Queue Q
Next

While Q ≠ ∅ do
       Select l(j) ∈ Q
       If REVISE(activities(l(j))) Then      //where activities(l(j)) refers to the
                                             set of activities in l(j)

              Q ← { l(k) ∈ L | activities(l(j)) ∩ activities(l(k)) ≠ ∅ }

                                             //assign back to Q all other
                                             literals k containing the activities
                                             contained in literal j
       Else
              Q ← Q \ l(j)
       Endif
End Loop
```

Figure 6.10 Pseudocode for BCSolver Iterations

Figure 6.10 shows the implementation of a fixed-point iteration framework for BCSolver. To summarize, the PDM++ literals are added to a queue structure. Each literal is then updated using the REVISE function, such that if there are any changes to the domains[3] of the activities, a search of all other literals which contain the affected activities is carried out, and added to the queue for re-evaluation. If there are no changes, the literal is removed from the queue. If the domain of any activity becomes empty, the algorithm terminates prematurely and indicates that the problem is not feasible, and cannot be solved. From the idempotent[4] properties of the problem, a fixed-point is guaranteed where the algorithm will terminate. This termination

[3] The domains of the activities is the range of starting times bounded by the Early Start (Lower Bound) and the Late Start (Upper Bound).

[4] Idempotence is defined as the property such that *f(f(x) = f(x)* after multiple calls of the REVISE subroutine.

condition is when there are no further changes to the domains of the activities, and as no further literals are added to the queue, the queue empties.

```
Function REVISE(activities(l_(j)))

        If |(activities(l_(j)))| = 1 Then                           //single activity X
                If type(l(j)) := Due_before Then
                        X.ub ← min(X.ub, m - d_X)
                ElseIf type(l(j)) := Due_after Then
                        X.lb ← max(X.lb, m - d_X)
                ElseIf type(l(j)) :=  Start_before Then
                        X.ub ← min(X.ub, m)
                ElseIf type(l(j)) := Start_after Then
                        X.lb ← max(X.lb, m)
                EndIf

        ElseIf |(activities(l_(j)))| = 2 Then                       //activities X & Y
                If type(l(j)) := FS(m) Then
                        X.ub ← min(X.ub, Y.ub - d_X - m)
                        Y.lb ← max(X.lb + d_X + m, Y.lb)
                ElseIf type(l(j)) := FS(~m) Then
                        X.lb ← max(X.lb, Y.lb - d_X - m)
                        Y.ub ← min(X.ub + d_X + m, Y.ub)
                ElseIf type(l(j)) := SS(m) Then
                        X.ub ← min(X.ub, Y.ub - m)
                        Y.lb ← max(X.lb + m, Y.lb)
                ElseIf type(l(j)) := SS(~m) Then
                        X.lb ← max(X.lb, Y.lb - m)
                        Y.ub ← min(X.ub + m, Y.ub)
                ElseIf type(l(j)) := FF(m) Then
                        X.ub ← min(X.ub, Y.ub + d_Y - d_X - m)
                        Y.lb ← max(X.lb - d_Y + d_X + m, Y.lb)
                ElseIf type(l(j)) := FF(~m) Then
                        X.lb ← max(X.lb, Y.lb + d_Y - d_X - m)
                        Y.ub ← min(X.ub - d_Y + d_X + m, Y.ub)
                ElseIf type(l(j)) := SF(m) Then
                        X.ub ← min(X.ub, Y.ub + d_Y - m)
                        Y.lb ← max(X.lb - d_Y + m, Y.lb)
                ElseIf type(l(j)) := SF(~m) Then
                        X.lb ← max(X.lb, Y.lb + d_Y - m)
                        Y.ub ← min(X.ub - d_Y + m, Y.ub)
                EndIf
        EndIf

If domain_change(activities(l_(j))) Then
        Return true                             //changes in domain detected
Else
        Return false                            //no changes in domain
EndIf


End Function
```

Figure 6.11 Pseudocode for BCSolver REVISE function

Figure 6.11 shows the pseudo code for implementing the REVISE function. The REVISE function distinguishes both unary and binary semantic literals, and further provides filtering rules for domains of the activity/activities in each of the literals. These rules are derived from the mathematical expression given in Table 6.1 and Table 6.2.

As mentioned previously, the domain of the start times of activity X can be modelled with the interval [*X.lb, X.ub*]. For illustration purposes, [X] is introduced as a shortened form of this notation. For unary literals, the example "X *Due-Before(m)*" will be used to show the derivation of the rules.

$$[X] + d_X \leq m \qquad (6.11)$$

$$[X] \leq m - d_X \qquad (6.12)$$

$$\begin{pmatrix} X.ub \leq m - d_X \\ X.lb \leq m - d_X \end{pmatrix} \qquad (6.13)$$

From Equation 6.11, we get the form of Equation 6.12 where X is made the subject of the equation. This form of the equation allows us to infer the upper bound and lower bound of X as shown in Equation 6.13. Here, the upper bound may not be constrained, whereupon it retains its original value. If the upper bound of X becomes constrained, the lower bound is vacuously constrained as well. Other unary semantic can be reasoned in the same manner.

"X *Before(m)* Y" can be transformed into the following two forms as show:

$$[X] \leq [Y] - d_X - m \qquad (6.14)$$

$$[\boldsymbol{Y}] \geq [\boldsymbol{X}] + \boldsymbol{d_X} + \boldsymbol{m} \qquad (6.15)$$

From Equation 6.14, it is now possible to infer that the most effective filtering for the upper bound of X is achieved through Equation 6.16. Similarly, Equation 6.17 can also be inferred about the upper bound value of Y.

$$\boldsymbol{X.ub} = \mathbf{min}(\boldsymbol{X.ub}\,, \boldsymbol{Y.ub} - \boldsymbol{d_X} - \boldsymbol{m}) \qquad (6.16)$$

$$\boldsymbol{Y.lb} = \mathbf{max}(\boldsymbol{Y.lb}\,, \boldsymbol{X.ub} + \boldsymbol{d_X} + \boldsymbol{m}) \qquad (6.17)$$

6.6.3.1 Using BCSolver as an Independent Schedule Solver

BCSolver may be used independently to evaluate a schedule on a set of constraints which are restricted to the conjunctive cases by running the algorithm twice. The first run obtains the lower-bound values of the domain. This is similar to the forward pass of the CPM algorithm. After the first run, an additional constraint to define the makespan is entered as follows in Equation 6.18.

$$Makespan = \sup\{Start_i + Dur_i\}, \quad \text{where } i \in \{Activities\} \qquad (6.18)$$

A second run of BCSolver which is analogous to the backward pass of the CPM algorithm is done to determine the latest start times. The algorithm processes each constraint at most 2d times, where d is the size of the domain. In the worst case scenario, the constraint is reduced by a single value, and there are at most 2d such values in the worst case of the binary semantic. Assuming *e* constraints, and since processing each constraint only involves the minimum and maximum value of the domain which is 4 comparisons for the binary semantic, we can compute that the BCSolver algorithm has a worst case runtime complexity of *O(8ed)*.

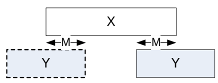
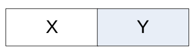
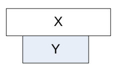
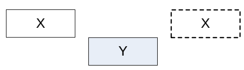
6.7. Implementing the Advanced Features of PDM++

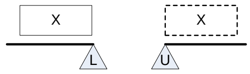
The following section gives an account of how the system is able to model the system requirements (SR1 to SR5) which were elaborated previously in Chapter 5.

6.7.1 Modelling Complex Temporal Relationships using Basic Syntax Operators

Using the syntactic operators of Disjunction and Conjunction introduced earlier, it is possible to create more complex semantics from the basic semantic literals of Table 6.1 and Table 6.2 to describe the events of concurrency and overlap of activities, work/resource continuity and disjointed activities, which commonly arise from construction requirements. These complex relationships were described previously in Chapter 5. Table 6.4 shows the mathematical description as well as the pictorial depictions of some of these complex constraints. The ECL^iPS^e implementations of the mathematical definitions are highlighted in bold.

Table 6.4. Complex PDM++ Relationships with ECLiPSe Representation

Semantic	Mathematical Definition with ECL^iPS^e	Pictorial Representation
X Overlaps(*m*) *Y*	$(X^- + d_X \geq Y^- + m) \wedge (Y^- + d_Y \geq X^- + m)$ **overlaps(Sx, Dx, Lag, Sy, Dy) :-** **conj(sf(Sx, Dx, Lag, Sy, Dy), sf(Sy, Dy, Lag, Sx, Dx))**	
X Overlaps(~*m*) *Y*	$(X^- + d_X \leq Y^- + m) \vee (Y^- + d_Y \leq X^- + m)$ **overlaps_max(Sx, Dx, Lag, Sy, Dy) :-** **disj(sf_max(Sx, Dx, Lag, Sy, Dy), sf_max(Sy, Dy, Lag, Sx, Dx))**	
X Meets *Y*	$(X^- + d_x \leq Y^-) \wedge (X^- + d_x \geq Y^-)$ **meets(Sx, Dx, Sy, Dy) :-** **conj(fs(Sx, Dx, 0, Sy, Dy), fs_max(Sx, Dx, 0, Sy, Dy))**	
X Contains *Y*	$(X^- \leq Y^-) \wedge (X^- + d_X \geq Y^- + d_Y)$ **contains(Sx, Dx, Sy, Dy) :-** **conj(ss(Sx, Dx, 0, Sy, Dy), ff_max(Sx, Dx, 0, Sy, Dy))**	
X Disjoint *Y*	$(X^- + d_X \leq Y^-) \vee (Y^- + d_Y \leq X^-)$ **disjoint(Sx, Dx, Sy, Dy) :-**	

	disj(fs(Sx, Dx, 0, Sy, Dy), fs(Sy, Dy, 0, Sx, Dx))	
X Cannot-Occur [*L*,*U*]	$\left(X^{-}+d_{X}\leq L\right)\vee\left(X^{-}\geq U\right)$ **cannot_occur(Sx, Dx, L, U) :- disj(due_before(Sx, Dx, L), start_after(Sx, Dx, U))**	

The constraints in Table 6.4 are implemented through the combination of the fundamental semantics in the Semantic Relationship Module with the syntax operators of the Logical Syntax Module. For example, the *overlaps/5* clause depicts the relationship between two activities such that there is at least an overlap of m days. Here, *X Overlaps(m) Y* is modelled as a conjunction of two basic PDM++ constraints: (*X FS(m) Y)* and *(Y FS(m) X)*. In similar fashion, other relationships may be obtained from the table.

Semantically, *X Meets Y* can be used to depict work or continuity constraints between two activities *X* and *Y*, where one activity must follow immediately after the other. *X* Contains *Y* is used to describe situations where there is a complete concurrency between *X* and *Y*. *X Disjoint Y* describes two activities that cannot be concurrent; one activity must finish before the other or vice versa. This may be used to depict several circumstances such as the modelling of key resources where two activities cannot occur simultaneously. For example, some safety issues where hotwork activities and activities involving the use of combustible adhesives may be modelled as disjointed activities within the PDM++ system. *X Cannot-Occur Y* is used to describe a time window during which an activity cannot occur; it must either finish before the time window, or else start after it. This may be used to model time windows of non-work for activities (also known as calendar constraints from the perspective of key resources).

6.7.2 Modelling Dynamic Construction Requirements using Intermediate and Derived Syntax Operators

From Chapter 5, Dynamic Construction Requirements were introduced as complex requirements that are conditional upon the fulfilment of other requirements. These dynamic requirements were previously shown to require the use of derived operators to exhibit the perdurant temporal behaviour of the requirement.

Using the derived syntax operators, it is possible to imply the existence of some relationships when specific pre-conditions are met, fulfilling the specification of the dynamic construction requirement. An example cited by (Fan and Tserng, 2006) but which was not addressed by them, describes the relationship between the *"Wall Painting" (WP)* and *"Floor Carpeting" (FC)* activities. While it is possible to schedule either activity first, FC (3 day duration) would require an additional activity *"Carpet Protection" (CP)* if WP (4 day duration) commenced after. This means that if FC commences first, then CP (also having 3 days duration) would not be necessary during FC. The above interdependency may be captured using the implication operator as follows in Equation 6.19. The ECL^iPS^e implementation is highlighted in bold in the following.

$$\text{WP disjoint FC} \wedge (\text{FC before WP} \rightarrow \text{CP contains FC}) \qquad \textbf{(6.19)}$$

```
conj(
  disjoint( WP, Dur_WP, FC, Dur_FC, 0 ),
    imply(
      before( WP, Dur_WP, FC, Dur_FC, 0 ),
      contains( CP, Dur_CP, FC, Dur_FC,0)
    )
 ).
```

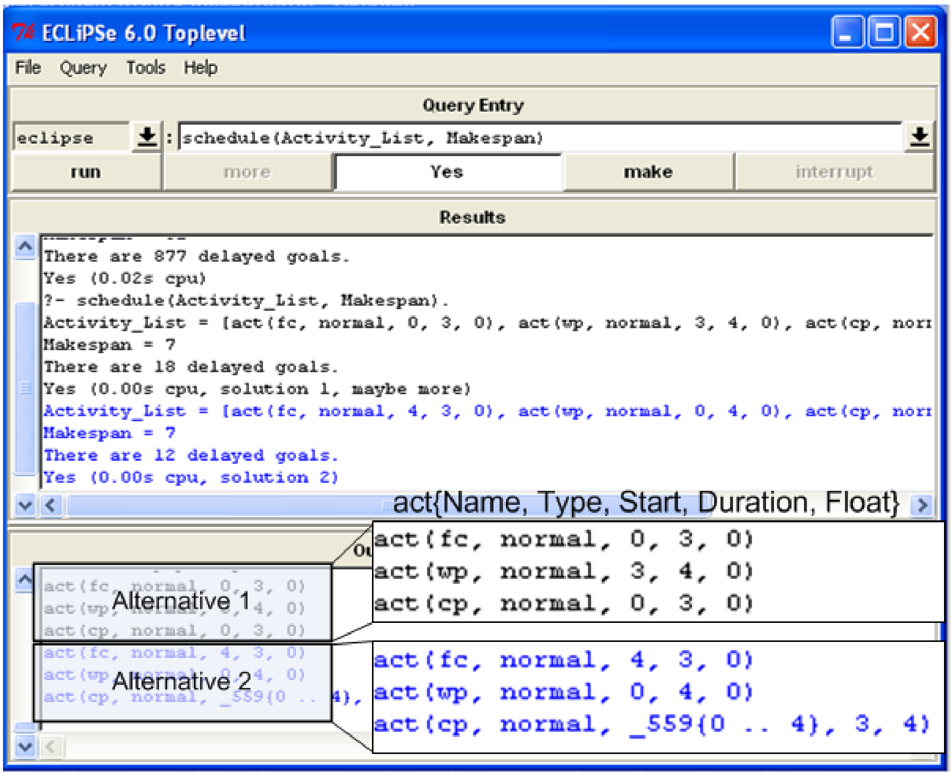


Figure 6.12. ECLiPSe Output for Implication Example

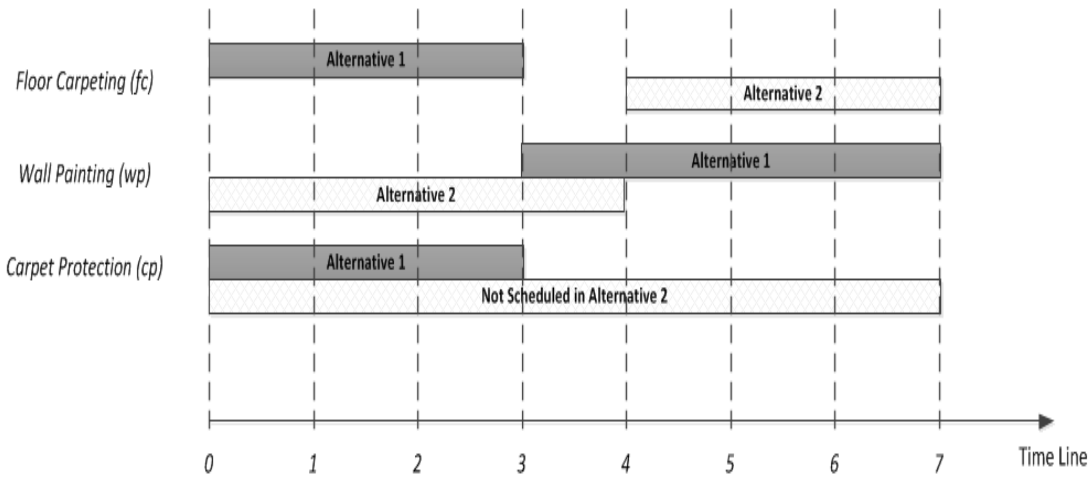


Figure 6.13. Gantt Chart showing Activities in Alternatives for Implication Example

Considering all these 3 activities, the system generates two alternative plans, both with an equivalent makespan of 7 days as shown in Figure 6.12. For the first alternative, FC (finishes by Day 3) occurs before WP (starts on Day 3), and CP is forced to be concurrent with FC (both FC and CP start on Day 0). For the second alternative, FC (starts after Day 4) starts after WP (starts on Day 4), and CP may be inferred to be not constrained. This means that CP is not needed, or not "active" in the schedule.

Using the example earlier, Equation 6.19 is changed such that the implication operator is replaced with an equivalence operator as shown in Equation 6.20, and the equivalent *ECL^iPS^e* implementation highlighted in bold.

$$\text{WP disjoint FC} \wedge (\text{FC before WP} \leftrightarrow \text{CP contains FC}) \quad (6.20)$$

```
conj(
  disjoint( WP, Dur_WP, FC, Dur_FC, 0 ),
    equiv(
      before( WP, Dur_WP, FC, Dur_FC, 0 ),
      contains( CP, Dur_CP, FC, Dur_FC,0)
    )
).
```

Here, the system returns three alternatives instead of two. In the first alternative, the sequence of activities and the makespan are the same as the previous instance of the same example when the implication operator was used instead. In the second alternative and third alternatives, WP finishes before FC. Unlike the single alternative sequence generated under the implication operation, the equivalence operation constrains CP to be "Not Contained" within FC. The meaning of "Not Contained" is such that CP must start before FC or it must finish after FC, resulting in the second and third alternative respectively. From the second alternative, the result is that CP is constrained to start from Day 0 to 3, resulting in a makespan of 7 days. From the third alternative however, the result of the system is that CP is constrained to finish after FC, and this results in CP starting on Day 5, as well as a longer makespan of 8 days.

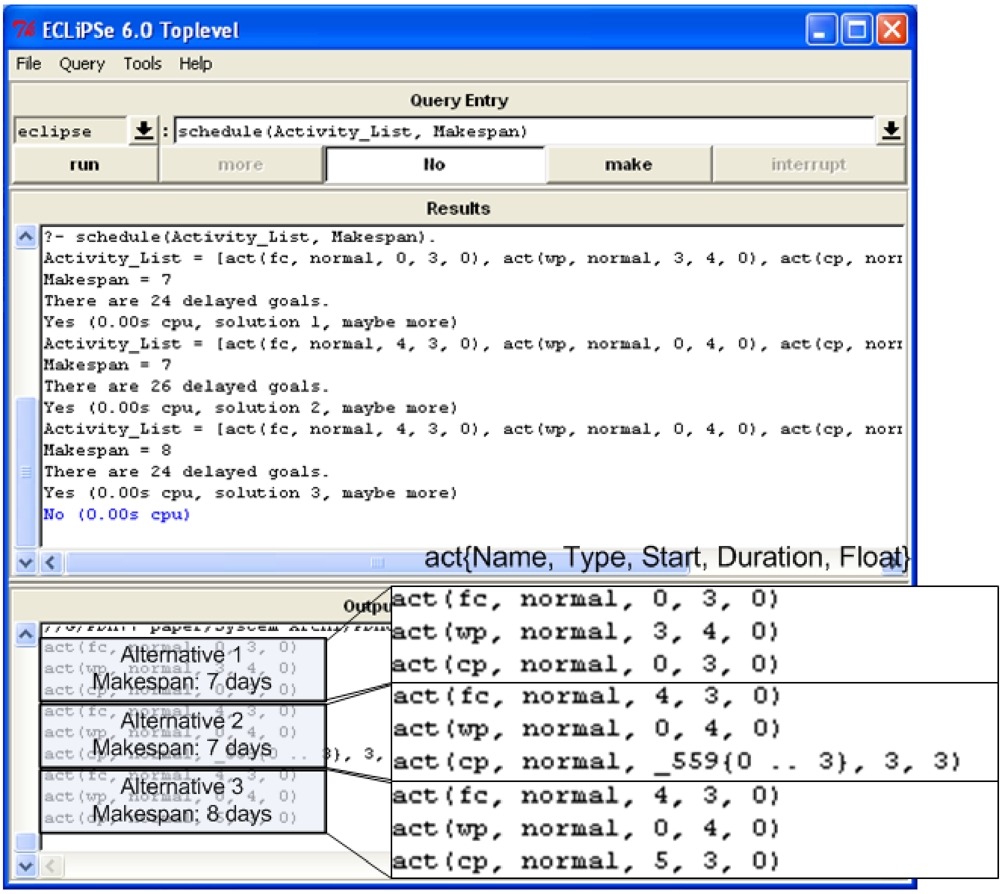


Figure 6.14. ECLiPSe Output for Equivalence Example

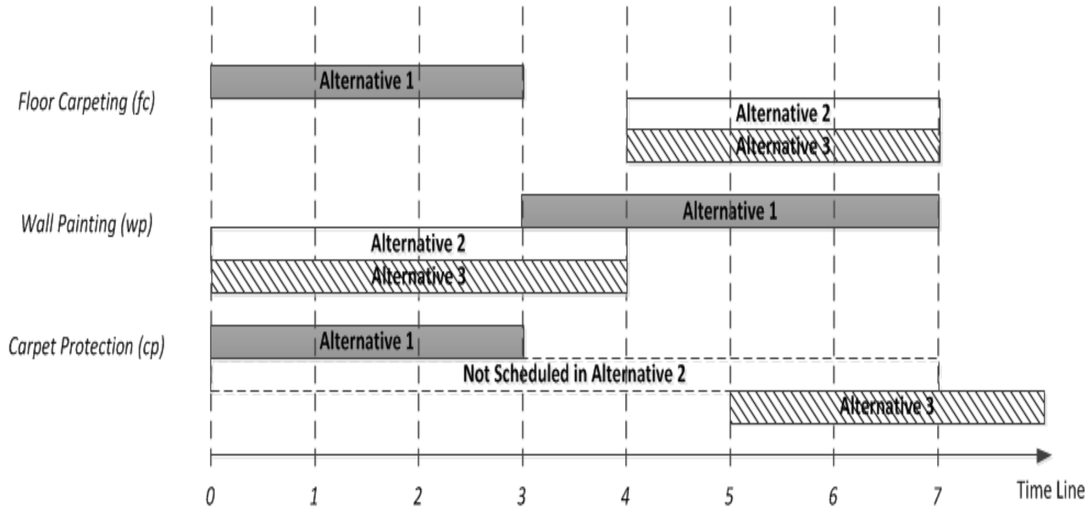


Figure 6.15. Gantt Chart showing activities in alternatives for Equivalence Example

These proposed logical syntax operators defined within the Logical Syntax Module allow for more complex semantic descriptions to be devised, describing more realistic constraints arising from planning considerations and construction requirements. Other operators may be implemented for greater variety of semantic descriptions, but will not be covered within the scope of this dissertation.

6.7.3 Meta-Interval Implementation

The Meta-interval construct for representing groups of activities, as well as usable periods between activities was introduced by Chua and Yeoh (2011). This representation enables hierarchical planning through higher level abstractions of a group of activities, as well as a way to represent the state of a construction product or resource which affect the execution of an activity. The construct is dependent upon its constituent activities, and has the characteristic of having a variable duration which is not known *a priori*. Four types of meta-intervals were introduced (Start-Finish(SF), Finish-Finish(FF), Start-Start(SS) and Finish-Start(FS)) as shown in Figure 5.4 where the meta-interval is defined by Start Activity (*Sa*) and End Activity *(Ea*). The implementation clauses in ECL^iPS^e are given on the right of the figure.

As an example, the implementation of the Start-Finish meta-interval as a constraint within PDM++ is achieved using the *meta_def_sf/6* clause shown in Figure 6.16. Here, it can be observed that the implementation is achievable by a conjunction of several clauses, which ties the start of the meta-interval *M* to the start of *Sa* (i.e. to start on the same day), and the finish of the meta-interval to the finish of *Ea* (i.e. to finish on the same day):

$$((M\ SS(0)\ SA) \wedge (M\ SS(\sim 0)\ SA)) \wedge ((M\ FF(0)\ EA) \wedge (M\ FF(\sim 0)\ EA)) \qquad (6.21)$$

```
meta_def_sf( Start_M, Dur_M, Start_Sa, Dur_Sa, Start_Ea, Dur_Ea) :-
        conj(
          conj(
            ss( Start_M, Dur_M, Start_Sa, Dur_Sa, 0 ),
            ss_max(Start_M, Dur_M, Start_Sa, Dur_Sa, 0 )),
          conj(
            ff( Start_M, Dur_M, Start_Ea, Dur_Ea, 0 ),
            ff_max(Start_M, Dur_M, Start_Ea, Dur_Ea, 0 )
          )
        ).
```

where *Start_M*, *Start_Sa*, and *Start_Ea* and *Dur_M*, *Dur_Sa*, and *Dur_Ea* represent the starts and durations of activities *M*, *Sa* and *Ea* respectively. The other types of meta-intervals may be similarly defined.

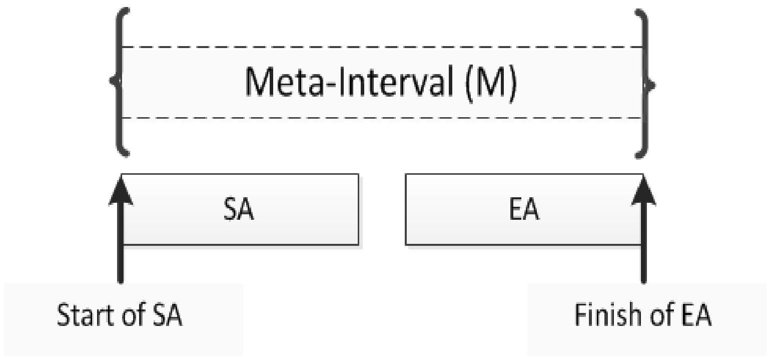


Figure 6.16. Start-Finish Meta-Interval Definition

Sometimes *Sa* and *Ea* of the meta-interval is unknown, but may be replaced with dummy activities of zero duration. This enables the meta-interval to provide a handler which allows the system to implement constraints on groups of activities, hence simplifying model specification and increasing readability of the model. For example, consider the activities *X*, *Y*, *Z*, and *W* in Figure 6.17a with the precedence shown. Knowing that X and Y are activities in subsystem M1, and Z and W are activities in subsystem M2, the precedence relations may be simplified into that shown in Figure 6.17b.

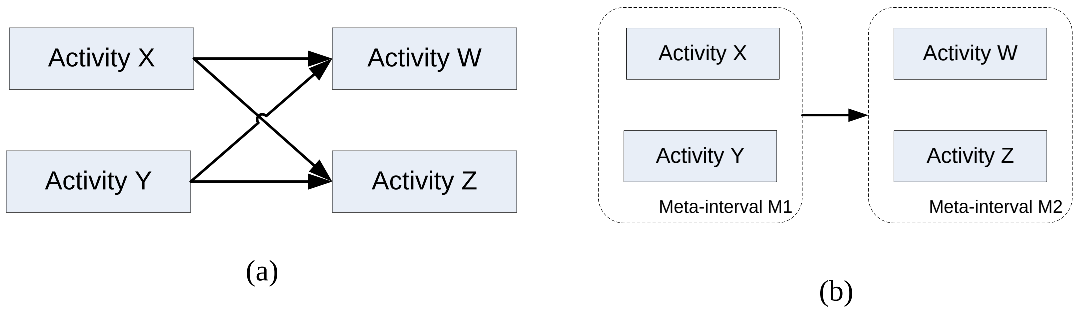


Figure 6.17. Meta-Intervals for Simplifying Repetitive Relationships between Groups of Activities

The above discussion features the implementation of the meta-interval within the system. From a user's perspective, the foregoing discussion regarding the implementation may be ignored. He/she only needs to input the definition of the meta-interval using the *meta_def* clauses in Figure 5.4, before treating the meta-interval as any other activity and specifying the temporal constraints relating to it.

In summary, meta-intervals increase the expressiveness of the system by allowing reasoning over time intervals not usually defined by activities but which may be useful for representing construction requirements. One such time interval is the usable period of a scaffold system, as will be seen later in the case study in Chapter 8.

6.8. Concluding Remarks

This chapter introduced the foundational knowledge necessary for implementing the evaluation framework. The system architecture for implementing the evaluation framework based on the $ECL^{i}PS^{e}$ constraint logic programming system is also elaborated, with suitable extensions to fully represent the PDM++ model, thereby increasing the expressiveness of the framework. This allows the features of the system to support the temporal reasoning needed for construction requirements.

The PDM++ problem is NP-hard. The "hardness" of the problem comes from two areas: disjunctive constraints and converting the propositional logic form of the problem to a suitable CNF. Despite this, the algorithms presented in this chapter have the potential to be further optimized. Zhang and Wu (1998) have proposed speed ups to the Bounds Consistency algorithm which rely on the structural support of the problem. However, with reference to the PDM++ problem, the speedups involved are linear. Further research has been carried out in the field of disjunctive constraints, where significant progress has been made to deal with the disjunctive constraints. Tsamardinos and Pollack (2003) have implemented a hybrid conflict-directed backjumping, semanic branching and no-good based reasoning to achieve significant speedups for similar disjunctive temporal problems. Also, other weaker forms of consistencies may be used to achieve the speedup (Schwalb and Dechter, 1997).

Other practical aspects of the problem which require further research include the incorporation of activity splitting into the system. Also it was previously noted that the conjunctive pre-processing algorithm has weak pruning capabilities. This may be enhanced using more expensive algorithms like path consistency to detect infeasibilities earlier.

The key contribution of the concepts in this chapter is the observation that the complex constraints introduced in Chapter 5, which were based on Allen's relations can be decomposed into the 8 binary and 4 unary relationships as the basic semantic of the language, with the logical operators as its syntax. The inverse operator introduced in this chapter is a key mechanism for deriving the implication and equivalence operators, by enabling the system to define the conditions when a constraint is false. These contributions enable the system architecture to handle the complex temporal constraints arising from construction requirements.

Chapter 7. Multi Objective Genetic Algorithm for Resolving Dynamic Construction Requirements under Spatial-Temporal Considerations

7.1. Introduction

The earlier contributions of this dissertation focussed on the spatial and temporal perspectives of the construction requirement separately. The spatial model in Chapter 4 was used to demonstrate the quantification of workspace congestion and conflict, while the temporal model in Chapter 5 dealt with the complexities of construction requirements, particularly from the perspective of dynamic construction requirements where alternative process capabilities are present.

A Multi-Objective Genetic Algorithm (MOGA) is proposed in this chapter to solve the workspace congestion problem with alternative process capabilities. This combined problem is termed the Dynamic Construction Requirements Problem with Spatial-Temporal Considerations (DCR-ST). A review of existing techniques is presented, and the problem formulation for the dynamic requirements problem with spatial temporal conflicts is presented. The structure of a modified NSGA-II to solve the DCR-ST problem is established as the main objective of this chapter. A performance analysis of the algorithm is shown to establish the effectiveness of the proposed algorithm to resolve the problem; an improved schedule found by the algorithm is compared to an initial schedule randomly generated, and the reasons for the improvements discussed.

7.2. Review of Relevant Literature

7.2.1 Overview of the mmRCPSP/max Problem

The Multi-Mode Resource Constrained Project Scheduling Problem with Maximal Lags (mmRCPSP/max) is chosen for review as it comes closest to representing the dynamic requirements problem under spatial temporal consideration. Specific deviations between the two problems will be discussed in this section. The mmRCPSP/max is defined as the resource constrained scheduling problem with multiple activity modes and subject to generalised precedence or maximal lag temporal constraints, and resource constraints. This problem is strongly NP-hard (Hartmann and Briskorn, 2010), and the resource constraints refer to discrete renewable resource (i.e. the total available number of resources at any given instant of the project is constant). The multiple modes of the problem refer to the various activity modes under which the activity may assume to achieve the objective.

Several deviations of the proposed DCR-ST are observed with the mmRCPSP/max problem. Firstly, DCR-ST generalises the modes to include not only activity modes, but constraint modes as well. In the first instance, the precedence constraints involving the activities are fixed, while in the second instance, the temporal constraints may be seen as having its own particular mode of operation. This implies that the activity list may not be the same for the same problem under different modes.

Secondly, the spatial resource defined by the proposed metrics CPI and DSI are not discrete, but continuous in nature, and does not follow the standard resource behaviour in current Operation Research literature (Węglarz, *et al.*, 2011). Furthermore, it cannot be fully defined as a renewable resource as the DSI is activity specific, and the capacity of the space resource is dependent not only upon the (spatial) interaction with other activities in the same time period, but also on its own utilization of the resource. Despite the notable differences, the general solution methods for solving the mmRCPSP/max can be adapted to solve the DCR-ST problem.

7.2.2 Solving the mmRCPSP/max using Exact and Meta-heuristic Methods

Various sources (De Reyck and Herroelen, 1999, Ballestín, *et al.*, 2011) acknowledge that the current literature on mmRCPSP/max problems is lacking. However, several exact and heuristic methods are still available. De Reyck and Herroelen (1999) identified the problem and presented a tabu search procedure for solving the problem. Brucker and Knust (2001) similarly recognised that the problem could be decomposed into two separate problems: Mode assignment and RCPSP with fixed mode assignments. Each sub-problem was recognized to be NP-hard. This approach was adopted by this work.

Exact methods to solve the problem have met with limited success (Heilmann, 2003), although several new developments involving integration of neighbourhood search have highlighted several leads to finding complete solutions of the problem (Zhu, *et al.*, 2006). However, the lack of more recent results has guided this research to explore meta-heuristic approaches to solve the problem.

Various heuristic methods in the current literature have been studied with promising results. One heuristic method relied on using different activity priorities and mode priorities to determine the best objective function value (Heilmann, 2001). A multi-pass approach was used. Similarly, M. Calhoun, *et al.* (2002) demonstrated the applicability of a similar heuristic using tabu search. More recently, Ballestín, *et al.* (2011) demonstrated how evolutionary algorithms could be employed by encoding the modes within the chromosome structure. This approach was subsequently adapted for use in this research.

7.3. Mathematical Formulation of the Problem

The DCR-ST problem is defined as follows: *i* activities are to be scheduled according to its construction requirements. Each construction requirement may be represented as a set of PDM++ temporal relations. Each activity is defined by a set of entities with spatial and temporal characteristics. The spatial characteristics place a demand on the workspace availability, and should not exceed the Planner's critical value, *C*. This critical value defines the extent to which a Planner will allow the workspace availability to support the spatial demand of the entities. In order to accommodate the demand, the problem allows the schedule to be extended to ensure that the workspace availability is able to accommodate the spatial demand. This extension is termed the planning horizon, and may be defined by the Planner.

The DCR-ST has been expressed generally in previous chapters and from the literature review conducted above is recognized as a variant of the mmRCPSP/max problem. In more detail, the problem is re-represented here mathematically as the following multi-objective optimization problem:

Minimize $$\boldsymbol{f} = \left\{ \sup_{\boldsymbol{i}}\{\boldsymbol{Start_i} + \boldsymbol{Dur_i}\}, \boldsymbol{CPI_{Avg}} \right\} \qquad \boldsymbol{i} := \{\boldsymbol{Activities}\} \qquad (7.1)$$

Subject to $$x_{l,\text{mod}\,e}\left(Start_j + p_{\min}\right) \le Start_k \qquad \forall j,k \in i \qquad (7.2)$$

$$Start_j + p_{\max} \ge x_{l,\text{mod}\,e}\left(Start_k\right) \qquad \forall j,k \in i \qquad (7.3)$$

$$x_{l,\text{mod}\,e}\left(Start_j\right) \le q_{\min} \qquad \forall j \in i \qquad (7.4)$$

$$Start_j \ge q_{\max}\, x_{l,\text{mod}\,e} \qquad \forall j \in i \qquad (7.5)$$

$$DSI_{ent} < C \qquad \forall ent \in N \qquad (7.6)$$

$$\forall Start_i \in Z^+ \qquad (7.7)$$

$$x_{l,\text{mod}e} = \begin{cases} 1 \\ 0 \end{cases} \qquad (7.8)$$

The solution space for each decision variable (activity start) $Start_j$ is assumed to be discrete and positive (Equation 7.7), with Dur_j the given duration of each activity j. p_{max}, p_{min}, q_{max} and q_{min} are time constants related to the PDM++ constraint type, and will be discussed in the following sections. Space time entities *ent* are elements of the activity *j*, of which there are a total number of *N* such entities. Another decision variable $x_{l,mode}$ is used to indicate if the constraint *l* is active when associated with a particular *mode* to form the constraint set. If the constraint is active, then $x_{l,mode}$ is 1, and 0 otherwise, as shown in Equation 7.8. In this research, the mode is defined as a feasible alternative sequence of work, and is a generalisation of the mmRCPSP definition of mode, which focused primarily on activity execution modes.

7.3.1 Multiple Objective Functions

DCR-ST has two conflicting objectives as shown in Equation 7.1, reflecting the trade-off between the total completion of the plan (Makespan), and the utilization of the workspace:

1. **Makespan** is to be minimized to constrain the total completion time of the activities. The assumption is maintained for the given problem: Activities are non-preemptive, and cannot be interrupted. As in Chapter 5, the makespan is defined as the maximum value of the set of finish times of each activity as follows:

$$f_1 = \sup_i \{Start_i + Dur_i\} \qquad \boldsymbol{i} := \{\boldsymbol{Activities}\} \qquad (7.9)$$

2. **Congestion Penalty Indicator** is to be minimized to constrain the congestion values between the workspace entities arising from the activity. Tighter space constraints increase the congestion indicator, indicating potential workflow problems arising due to the increased demands for the limited available space. To handle multiple modes with different numbesr of activities and space entities, the average value CPI_{Avg} is adopted as a fitness measure. CPI_{Avg} is evaluated across the total space entities, *ent* in the set of *N*. This is defined by the following Equation 7.10, which is based on Equation 4.9 and Equation 4.10:

$$f_2 = CPI_{Avg} = \frac{1}{N}\sum_{ent \in N}^{ent} CPI_{ent} = \frac{1}{N}\sum_{ent \in N}^{ent} \frac{1-\exp^{\frac{\mathrm{DSI}_{ent}\cdot\alpha}{C}}}{1-\exp^{\alpha}} \qquad \text{if } \mathrm{DSI}_{ent} < C \tag{7.10}$$

In general, the shorter the Makespan of the project, the higher is the utilization of the workspace resource; and vice versa. From the formulation of the workspace utilization in Equation 4.10, shorter makespans could cause a higher degree of temporal overlap within the schedule, causing the *CPI* for the schedule to increase. This justifies MOGA as a solution technique.

7.3.2 DCR-ST Constraints

7.3.2.1 Generalisations of the basic PDM++ semantic literals

The model constraints are generalisations of the 8 binary semantic literals (refer to Equation 7.2 and Equation 7.3) and the 4 unary semantic literals (Equation 7.4 and Equation 7.5) of PDM++. The variables p_{min}, p_{max}, q_{min} and q_{max} are used to denote time constant values which are dependent on the PDM++ constraint type (type of semantic literal) used. p_{min} and q_{min} are used to refer to the binary and unary minimal type literals respectively, while p_{max}, and q_{max} are used to refer to the binary and unary maximal type. For the semantic literals involving Activities *i* and *j*, dur_i and dur_j refer to the durations of *i* and *j* respectively, while *m* refers to the original lag values of the constraints.

Table 7.1. Parameter Values of p_{min}, p_{max}, q_{min} and q_{max}

Time Constant Values	Constraint Type	Equation of Time Constants *p* and *q*
p_{min}	Before(m)/After(m)	$p := dur_i + m$
p_{max}	Before(~m)/After(~m)	
p_{min}	Starts(m)/Started-by(m)	$p := m$
p_{max}	Starts(~m)/Started-by(~m)	
p_{min}	Finishes(m)/Finished-by(m)	$p := dur_i + m$
p_{max}	Finishes(~m)/Finished-by(~m)	
p_{min}	Start-Finish(m)	$p := dur_i - dur_j + m$
p_{max}	Start-Finish(~m)	
q_{min}	Due-Before(m)	$q := m - dur_i$
q_{max}	Due-After(m)	
q_{min}	Start-Before(m)	$q := m$
q_{max}	Start-After(m)	

7.3.2.2 Active/Inactive Constraint Set and Activities

Part of the solution framework lies in distinguishing active and inactive constraint sets, and the activities which make up these sets. Only the active constraint sets are considered when defining a particular mode of the problem. The activation of a constraint adheres to the following rules:

1. If two PDM++ literals are connected by a conjunctive operator, then both literals must be active.

	$X^- \geq Y^-$	$\wedge$	$X^- + d_X \leq Y^- + d_Y$
Mode 1:	Literal is Active		Literal is Active

Figure 7.1. Active Literals under Conjunctive Operators

2. If two PDM++ literals are connected by a disjunctive operator, there exist distinct modes where each literal is active (assuming that the operators do not lead to infeasible solutions).

	$X^- + d_X \leq Y^- + m$	$\vee$	$Y^- + d_Y \leq X^- + m$
Mode 1:	Literal is Active		Literal is Inactive
Mode 2:	Literal is Inactive		Literal is Active

Figure 7.2. Active Literals under Disjunctive Operators

3. If two PDM++ literals are connected by an implication operator, then if the literal representing the precondition is true (and consequently, active), then the following literal must also be active. If the literal representing the precondition is false, then the following literal is not active.

	$X^- + d_X \leq Y^- + m$	$\rightarrow$	$Y^- + d_Y \leq X^- + m$
Mode 1:	Literal is True		Literal is Active
Mode 2:	Literal is False		Literal is Inactive

Figure 7.3. Active Literals under Implication Operators

Only activities which are part of the active constraint set are active for a particular mode. Active constraints involving other logical operators in PDM++ may be constructed from a combination of the above rules.

7.4. Implementation of a Genetic Algorithm for the DCR-ST Problem

In summary, the DCR-ST problem can be decomposed into two distinct sub-problems: A Mode Assignment Problem (MAP) and the RCPSP for a particular mode assignment. In general, the MAP of this problem is known to be NP-hard with worst case run-time complexity of exponential order $O(2^D)$ where D is the number of disjunctive literals. However, in practice, the Planner will often identify the relevant feasible construction strategies which correspond to the modes of the problem. This limits the complexity of the MAP problem. In this implementation, the MAP problem is trivialized through the prior identification of specific construction strategies by the Planner. Instead, the modes are incorporated into the general RCPSP as part of the chromosome design, as will be discussed later.

Additionally, the Planner identifies a relevant planning horizon which limits the search of the genetic algorithm. This planning horizon indicates the extent to which the schedule can be lengthened to accommodate the worksite congestion by controlling the amount of temporal overlaps between the activities. A long planning horizon is often not desirable in practice, as it could amount to unnecessary delays leading to liquidated damages. Also, it may not be entirely realistic as the earliest possible completion of a schedule is often the top priority of Planners.

A parameter-less multi objective genetic algorithm (MOGA) based on the popular NSGA-II (Deb, *et al.*, 2000) is proposed to solve the resulting optimization problem which describes DCR-ST. This parameter-less characteristic is achieved through self-adaptation of the crossover and mutation probabilities.

7.4.1 Model Overview

The NSGA-II is an elitist Pareto based approach which is computationally fast. Among its advantages is the use of a parameter-less approach with a simple and efficient constraint handling mechanism for finding solutions on the Pareto front while preserving the diversity of its solutions. Hence, NSGA-II is chosen as the underlying search mechanism to resolve the multi-objective problem of this research. Figure 7.4 shows the main mechanisms of the NSGA-II Algorithm, with the various adaptations adopted for this problem.

An initial pre-processing of the data is used to determine if the problem is initially infeasible. A check of the combined utilization factor ρ of all the entities under the different modes is carried out against the Planner's Critical value *C*. Any infeasible mode is eliminated. If all modes are eliminated, the algorithm terminates prematurely with infeasibility.

An initial population, P_0 with a predetermined population size N_{pop} is randomly generated at the start for the first generation. Each solution in P_0 is evaluated, with P_0 being renamed as P_{parent}. Selection, Crossover and Mutation of the solutions in P_{parent} is carried out to form a new child population P_{child}. Both P_{parent} and P_{child} are combined into a new pool and sorted according to dominance. An elitist selection of the best candidates in the new pool is retained, with the inferior candidates removed. The elitist selection of solutions becomes candidate solutions in the new P_{parent}. This is repeated until the termination condition of the number of generations is met.

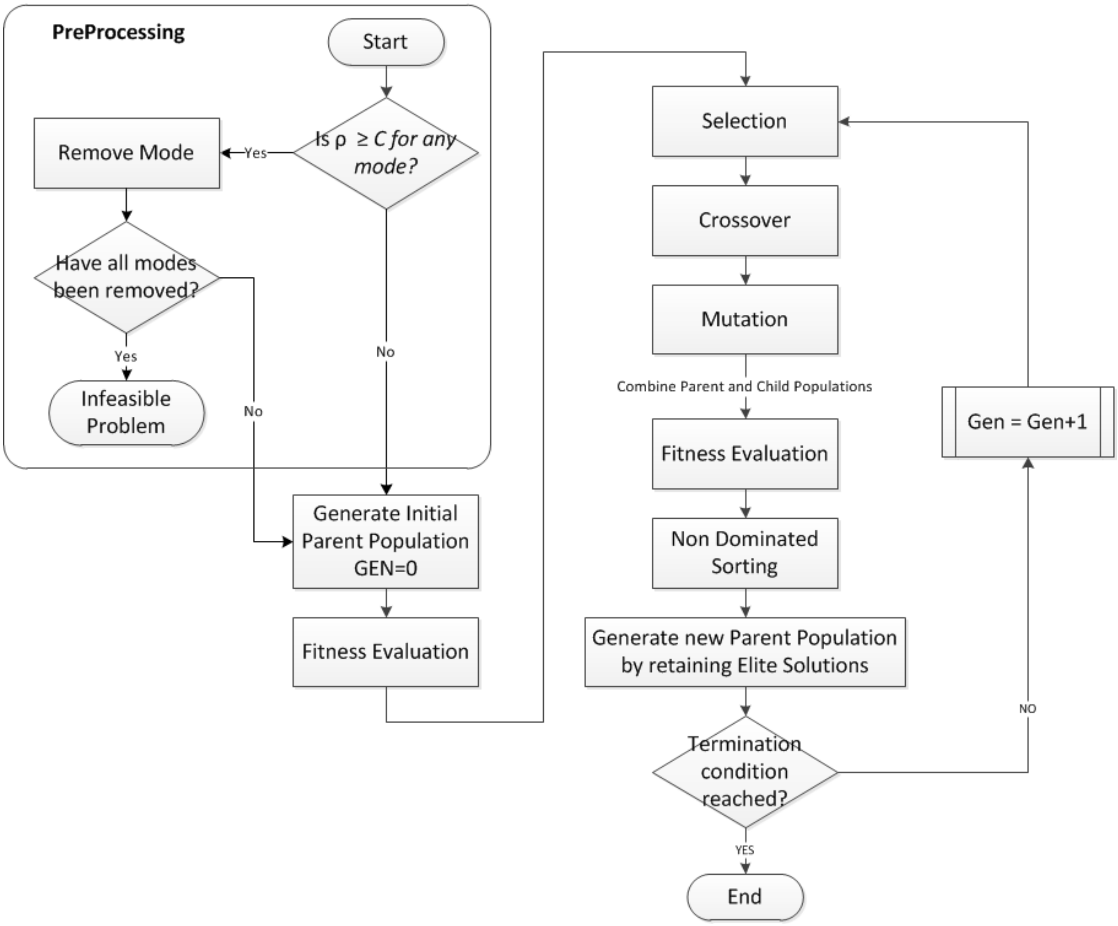


Figure 7.4. Overview of Main GA Algorithm

7.4.2 Chromosome Design and Representation of Solutions

The chromosome design is an important part of the problem representation; it represents the mapping of the decision variable in the solution space to the gene (Chan, *et al.*, 1996). An improper choice of the chromosome is likely to lead to weak performance of the algorithm. Using a variation of the random keys approach (Bean, 1994), the chromosome structure design in Figure 7.5 is based upon the order in which the activity is chosen for scheduling. The key advantage of this design is that infeasibilities arising from the PDM++ temporal constraints are avoided completely. Also, the real-valued encoding for all segments allows for standard genetic algorithm techniques to be applied without much modification. Each encoding takes a real value between 0 and 1.

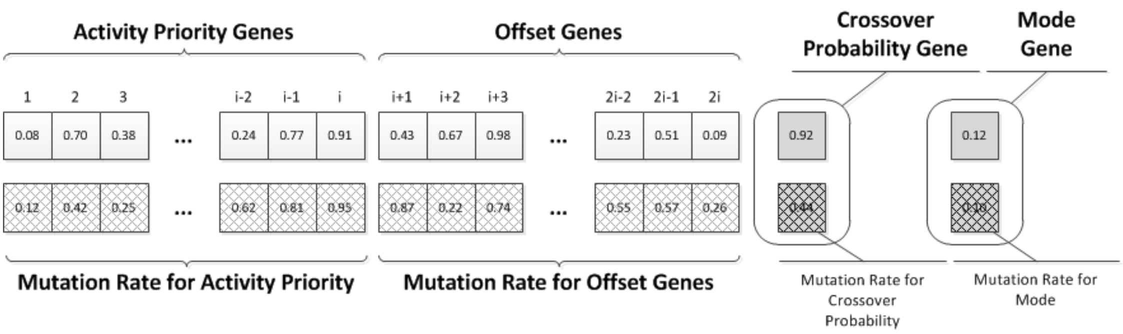


Figure 7.5. Chromosome Structure

The chromosome design is segmented into 4 different levels:

1. Activity Priority Genes are randomly generated real-valued keys which encode the priority of the activity based on its topological ordering. Higher priority activities are chosen for scheduling first. Each activity is distinguished by its position in the chromosome, of which there are i activities.
2. Offset Genes are also randomly generated real-valued genes which encode the offset from the earliest possible start within the feasible time window available for each activity. For each Schedule Gene and Offset Gene, there is a self-adaptive mutation rate $p_{m,i}$ assigned to it.

3. A self-adaptive crossover probability gene is also encoded as a real-value. A mutation rate is also available for this gene.
4. The Mode Gene is also a randomly generated real-value gene which encodes the alternative sequence and the active constraint set applicable. It also has a self-adaptive mutation rate $p_{m,mode}$ assigned.

The novelty of the approach used in this algorithm is this encoding structure which explicitly separates the workspace resource constraints from the temporal constraints. The PDM++ temporal constraints (Equation 7.2 to Equation 7.5) are dealt with by maintaining temporal feasibility through the priority value encoding of the chromosome structure, while the workspace resource constraint (Equation 7.6) is handled by the elitism-based constraint handling scheme within the proposed genetic algorithm (refer to Section 7.4.7.2). The key observation giving rise to this scheme is that a solution must display temporal feasibility as a necessary condition before considering the effects of the workspace resource constraint. Hence, the chromosome structure reflects this observation; the activity priority gene plays a primary role in maintaining the temporal feasibility, while the offset gene is important in reducing the constraint violations arising from the workspace resource unavailability/infeasibility.

7.4.3 Chromosome Encoding/Decoding using BCSolver

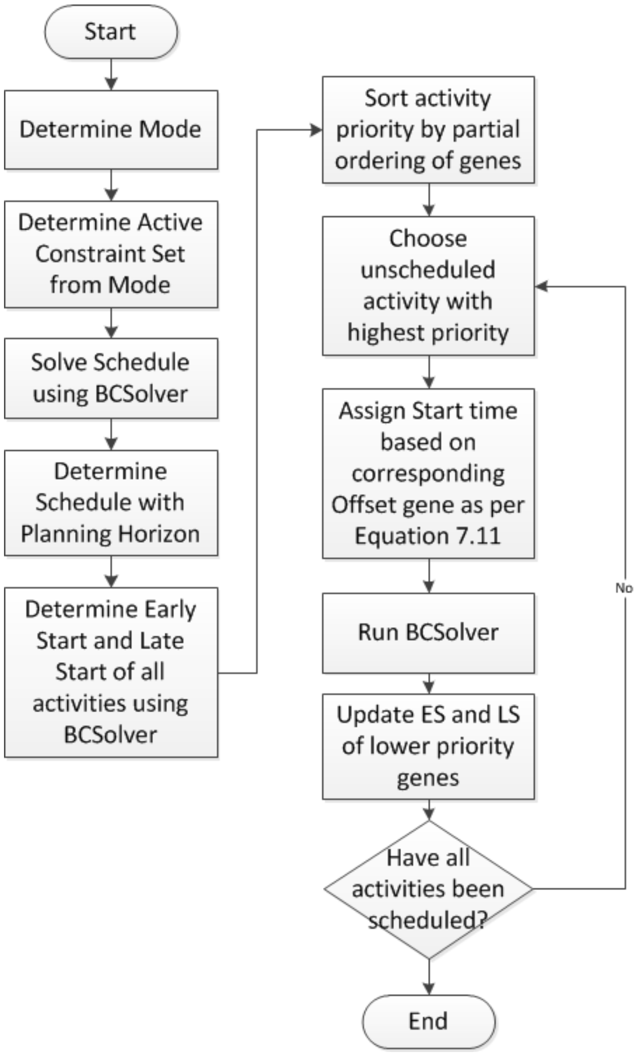


Figure 7.6. Decoding Algorithm

In the implementation, the decoding of the chromosome is dependent upon the Mode Gene. Based on the mode encoded within the Mode Gene, an active constraint set is generated. The planning horizon is also predetermined by the Planner as a percentage of the early start schedule based on this active constraint set.

Using the BCSolver algorithm from Chapter 6, the initial lower and upper bounds of the starting times of each activity is generated from the active constraint set and the planning horizon. These lower and upper bounds correspond to the early start and late starts of the activity. BCSolver is used due to its low computational overhead and its direct applicability to resolving PDM++ temporal constraints with minimal and maximal lags.

From the activity priority genes, a partial ordering among the activities can be generated. The higher priority activities are allowed to choose their start times from the available feasible time window, which is the difference between the lower and upper bounds of the start times. The choice of the start time is determined by the offset gene for the corresponding activity as shown in Equation 7.11:

$$Start_p = gene_{i+p} \times (LS_p - ES_p) + ES_p \qquad \textbf{(7.11)}$$

where $Start_p$ refers to the start time of activity p out of i activities, $gene_{i+p}$ refers to the value of the gene at position ($i+p$), LS_p and ES_p refer to the late start (upper bound) and early start (lower bound) of p respectively.

This chosen start time is assigned to the activity, and BCSolver is called to update the new lower and upper bounds of the lower priority activities based on the active constraint set indicated by the mode. This ensures that the PDM++ temporal constraint feasibility is preserved. The procedure is continued with the activity having the next highest priority, with new early starts and late starts updated incrementally in BCSolver. This incremental updating is achieved by allowing BCSolver to choose the affected constraints, and revising only those affected constraints for updating, instead of running all the constraints again.

The decoding procedure discussed above is an extension to the current float method applied by Chua, *et al.* (1997) where the activity priority is also allowed to evolve as part of the algorithm. The algorithm is repeated until all activities have been assigned a feasible start (scheduled).

7.4.4 Binary Tournament Selection

A binary tournament selection is used to randomly choose two individuals from the population to compete against each other. The better of the two solutions will subsequently appear in the subsequent population. This is repeated until the new population size is filled. This selection method was chosen to favour stronger candidate solutions for populating the next generation, generating a "survival pressure" which is an important characteristic for determining the rate of convergence of the genetic algorithm (Back, 1994). In the literature, binary tournament selection compares favourably to other selection mechanisms (Back, 1994).

7.4.5 Crossover Operator

A two point crossover operator is used with a self-adaptive crossover probability. Crossover operations randomly exchange genetic material with the possibility that "good" solutions may lead to "better" ones. Crossover takes building blocks from two individuals (dubbed "Father" and "Mother") and combines them into two new candidate solutions (also called "Offspring"). A random number (*Rnd*) is selected, and compared against the encoded crossover probability of both the father and mother (denoted as $P_{c,Father}$ and $P_{c,Mother}$ respectively). If the crossover probability of the father candidate solution governs, then Offspring 1 will be generated respectively. A similar rule applies for the mother candidate solution and Offspring 2. The crossover of the genetic material is done by choosing two points at random along the length of the chromosome string, and then interchanging the blocks as illustrated in the following Figure 7.7.

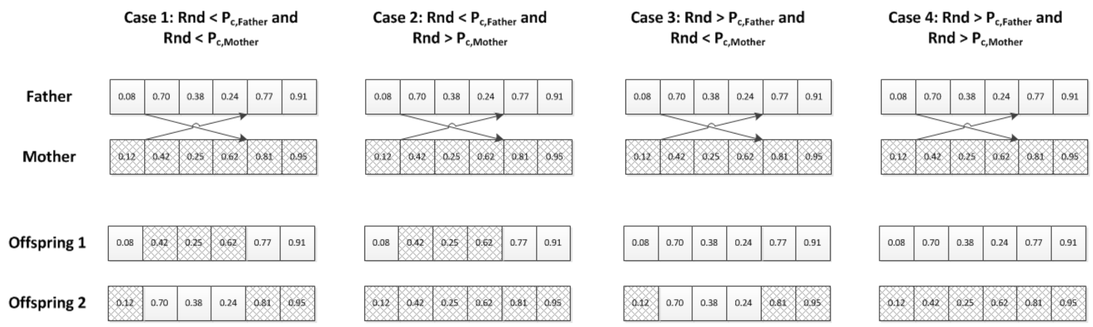


Figure 7.7 Illustration of Two Point Crossover Operator

Choosing the proper crossover probability is known to be problem specific, and is a difficult task in general (Smętek and Trawiński, 2011). Hence, the self-adaptive method for choosing crossover probabilities was selected. This self-adaptive method is effectively an "evolution" of the "evolution" parameters of the problem and has met with some success (Maruo, *et al.*, 2005). This means that the best value of the crossover probability is left to the MOGA to ascertain.

7.4.6 Mutation Operator

A self-adaptive mutation operator was chosen for the same reason as the self-adaptive crossover: It is not easy to set the probability of mutation as a control parameter for the GA, and its choice is a vital factor in the success of the genetic algorithm (Serpell and Smith, 2010). It has been found that self-adaptation exhibits good results for a dynamic combination of adaptation for both crossover and mutation rates under the effects of elitism (Bäck, *et al.*, 2000).

The mutation operator works by randomly changing the genetic material in the gene as illustrated in Figure 7.8. In this implementation, a gene based mutation probability is selected. This means that each gene is subject to mutation individually. Here, the decision to mutate a particular gene depends on the corresponding self-encoded mutation rate. A random number is chosen, and if it falls below the mutation encoding, the gene value is mutated. Elitism is then subsequently used to influence the selection of the mutation rate (Smith and Fogarty, 1996).

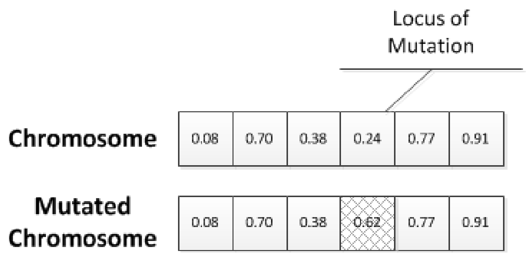


Figure 7.8. Illustration of Mutation Operator

In general, mutation is a way of preventing premature convergence towards local optima, and helps to direct the search towards different search spaces, creating new possibilities that may not be present in the initial pool of solutions (Coello Coello, 2001).

7.4.7 Evaluation of the Objectives of the DCR-ST

7.4.7.1 Fitness Evaluation

Evaluating the fitness requires several inputs from the 3D CAD model and its associated schedule. The information needed is shown in the following diagram (Figure 7.9). Here, the 3D model allows the spatial information to be obtained. This information refers to the list of active space entities, spatial utilization and space entity interaction matrix. The list of active space entities are the workspace and pathspace entities involved in the active schedule, which is determined by the mode. The spatial utilization is determined by the Planner and may be appended as attributes to the 3D

Model. A script was written to elicit the overlapping volumes between the various active space entities within the 3D model, and the results recorded in a Space Entity Interaction Matrix. The Space Entity Interaction Matrix contains a pair-wise interaction matrix between any two workspace/pathspace entities within the 3D model. The values of this upper-triangular matrix indicate the overlapping volume between the space entities, also known as the spatial overlaps.

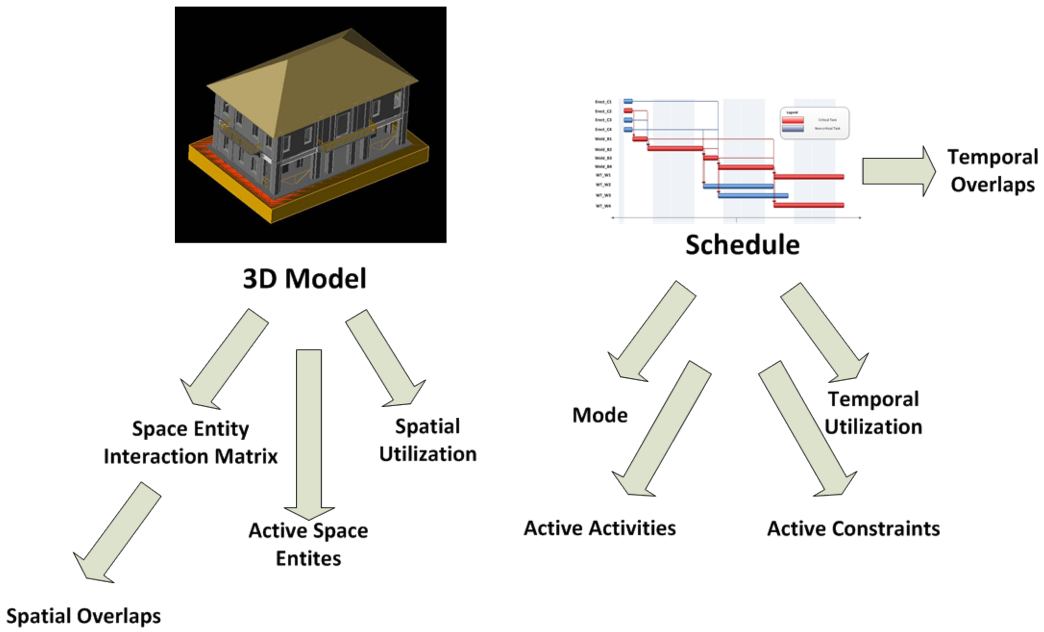


Figure 7.9. External Inputs Required for Evaluating the Model

The schedule allows the amount of temporal overlaps, temporal utilization, active activities and active constraints pertaining to a particular mode to be obtained. Again, the active activities and active constraints are dependent upon the active mode of the available alternative schedules. For each of these activities, the planner inputs the temporal utilization as an additional attribute into each activity. From the active activities, the temporal overlaps between activities are determined by the program dynamically at run-time as needed. These temporal information are necessary for the evaluation of the objective functions of the model.

In Figure 7.10, the fitness evaluation procedure is shown with the corresponding information dependencies needed at each step of the algorithm on the right. The inputs needed for each step is provided in the dependency diagram with algorithm generated inputs denoted with a "+" sign, and aforementioned external inputs denoted with a "++" sign. Intermediate outputs of the steps in the algorithm are denoted with a "-" sign. Each input and output is intentionally shaded to reflect a corresponding relationship with each step of the algorithm in which it is involved in.

The fitness evaluation algorithm starts with the decoding of the chromosome to determine the mode, the activities, with the start times and durations of the associated entities. The mode determines which activities are active, as well as which constraints are to be included. The utilization factor ρ is calculated from the spatial and temporal utilization provided. The values of the spatial and temporal utilization are dependent upon the mode of the problem. Schedule information is also integrated into the system in the form of activity lists; from this list of active activities, the existence of the entities can also be inferred. The DSI values of the entities are evaluated based on the existing entities within the mode. The CPI_{Avg} is evaluated as shown in Equation 7.10, and included in to the fitness vector.

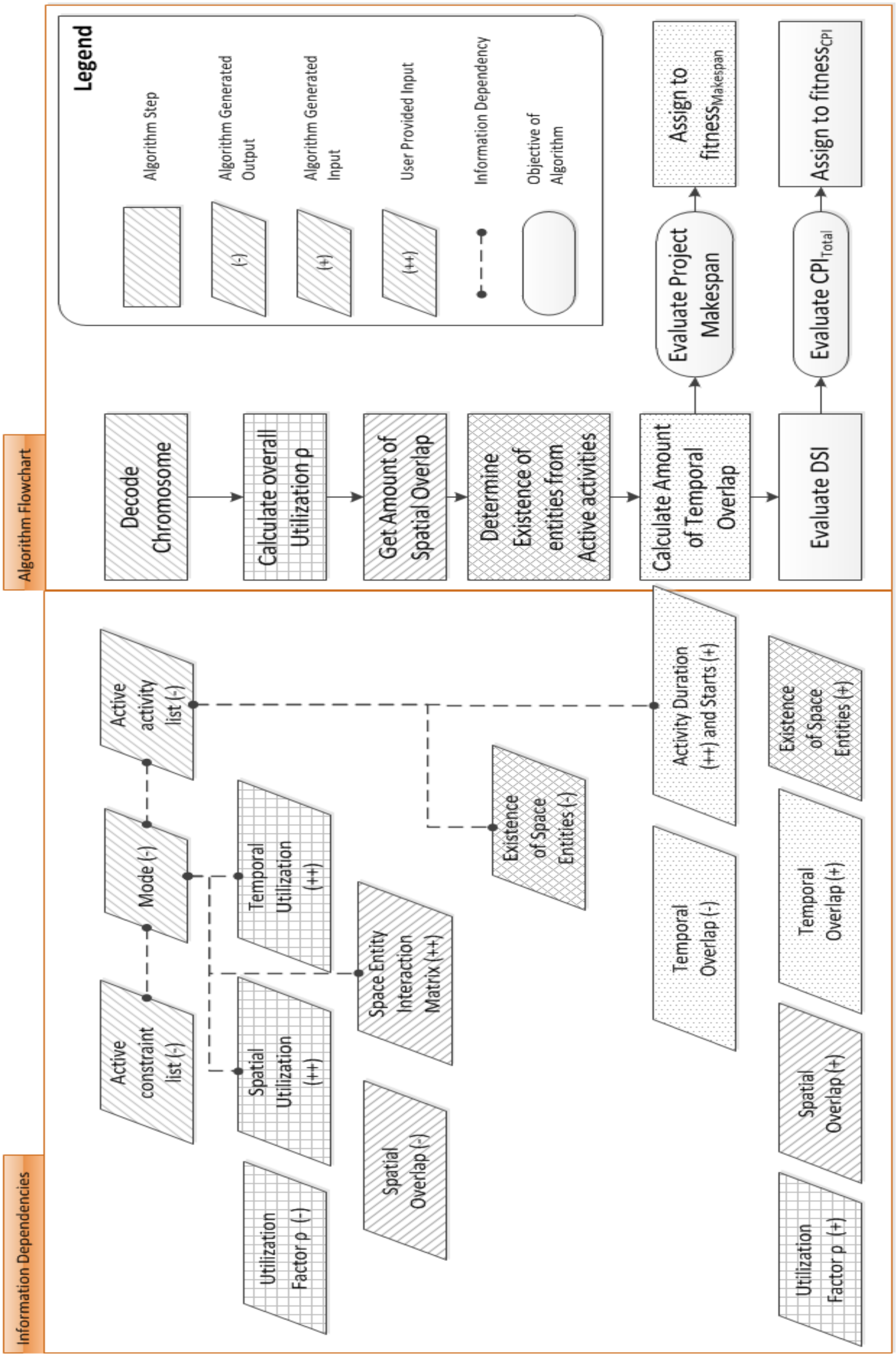


Figure 7.10. Fitness Evaluation Flowchart with Information Dependencies

From the start times and the durations of the activities, the project makespan can also be evaluated (as per Equation 7.9), and included within the fitness vector of the chromosome.

7.4.7.2 Non-Dominated Sorting with Constraint Handling

The non-dominated ranking of the fitness of the solution is conducted by evaluating one solution against all the other solutions in the population to determine if it is dominated. A dominated solution is defined as one which displays an inferior fitness value for all criteria as compared to another solution. Mathematically, domination can be expressed as the following Equation 7.12:

$$x \succ y \iff f_i(x) \leq f_i(y) \; and \; \exists j: f_j(x) < f_j(y) \qquad \textbf{(7.12)}$$

Since the temporal feasibility arising from the PDM++ constraints has been handled by the encoding scheme, the genetic algorithm provides a constraint handling mechanism for handling the violation of entities having DSI value above the Planner's critical value (refer to Equation 7.6). Each entity which violates this constraint is termed a constraint violation. The overall constraint violation for each chromosome is then evaluated. Therefore, this is the total number of constraints which have been violated in a solution (as shown in Equation 7.13).

$$Constraint\ Violation = \sum |\{Violated\ Constraints\}| \qquad \textbf{(7.13)}$$

The concept of constraint domination ((Deb, *et al.*, 2000) is introduced where a solution *i* is said to constraint-dominate another solution *j* if any of the following is true:

1. Solution *i* has a smaller constraint violation than solution *j*.
2. If both solutions *i* and *j* have the same value of constraint violations, and solution *i* dominates solution *j* as in Equation 7.12.

If a solution is found which is not constraint-dominated by the other solutions, it is added to a subpopulation which is maintained as the non-dominated front. These non-dominated solutions are removed, and the remaining population is checked again to find new non-dominated solutions. These new solutions are placed in a subsequent front, and the process repeated until all solutions have been placed in a front.

The key advantage of this elitist based ranking of constraint violations is that it does away with penalty functions and penalty values which are often problem specific. This method is also algorithmically inexpensive, being comparable to the elitist dominance ranking mechanism without constraint handling (Deb, *et al.*, 2000).

7.4.7.3 Diversity Preservation

Within the solutions making up the Pareto front, a crowding distance operator is implemented as part of the NSGA-II algorithm to maintain diversity. This measure provides a density estimate of the solutions surrounding a particular solution in the population. More specifically, the crowding distance of a point is the estimate of the size of the largest cuboid enclosing itself without including any other point in the same non-dominated front as shown in Figure 7.10 with the points on the Pareto front demarcated in black.

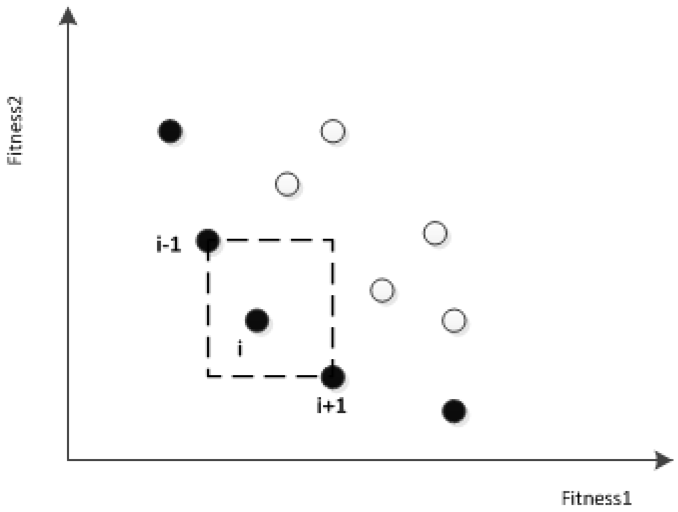


Figure 7.11. Crowding Distance

The crowding distance operator is evaluated by sorting the solutions according to its objective value for all objectives. An arbitrarily large number is given to the first and last solutions. For all other solutions between the first and last, the following normalized crowding distance evaluation is carried out:

$$crowding_i = \sum_k^m \frac{(f_k(i+1) - f_k(i-1))}{(f_{k,max} - f_{k,min})} \quad (7.14)$$

where i is the point in the population, k is the specific objective out of a total of m objectives. $f_{k,min}$ and $f_{k,max}$ refer to the minimum and maximum fitness values of the objective k.

7.5. Performance of Proposed Algorithm via an Illustrative Case Study

The performance of the proposed genetic algorithm is discussed in this section using a real life case study as a test example. The focus of this section will be to show the applicability of the proposed method for resolving worksite congestion issues, rather than on the computational performance of the algorithm. This is due in part to the uniqueness of the DCR-ST problem, where there are no comparative benchmark problems on which to test the performance of the proposed genetic algorithm, nor are there any known competing algorithms (to the author's best knowledge) which are fully capable of modelling the problem.

Moreover, from a computational perspective, the main control parameters of the genetic algorithm (crossover probability and mutation probability) are encoded genetically using self-adaptive parameters as chromosomes within the algorithm. This is because the performance of such parameters may be problem specific.

The test example involving Mechanical and Electrical (M&E) services installation in a section of an underground subway station is used as an illustrative case involving 46 activities and 85 space entities with four identified modes (feasible alternative work sequences). The modes contained between 60 to 66 PDM++ literals. A planning horizon may be represented mathematically as:

$$\boldsymbol{Planning\ Horizon} = \boldsymbol{sup}_{\boldsymbol{mode}}\{\boldsymbol{Makespan}\} + \boldsymbol{Float} \tag{7.15}$$

A planning horizon of 10% of the longest project makespan was used in this example. This worked out to an allowable addition of 7 days float to the original 74 day makespan. The algorithm was allowed to run for 100 generations with a population size of 50 and the convergence of the algorithm can be seen from Figure 7.12. In this figure, all solutions from the five generations were plotted for comparison.

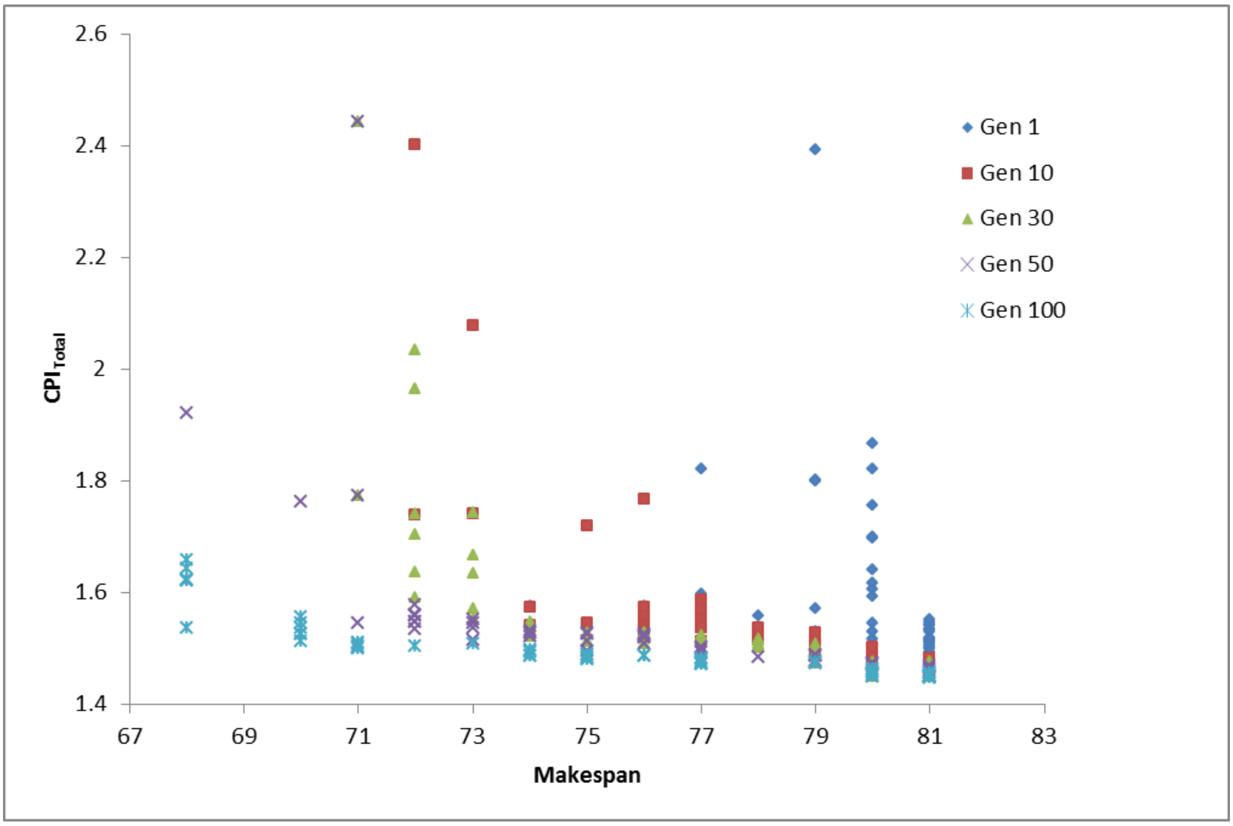


Figure 7.12. Convergence of Algorithm to Pareto Front after 100 Generations

The results of the convergence show the difference between the initially generated population and the subsequent generations. By the 10th generation, the algorithm has "learnt" sufficiently to improve the Pareto front. In the initial generation, the solutions tend to have modal makepsans of 79 and 80 days. However, the improvements made by the algorithm cause the solutions to spread out across the Pareto front, increasing the number of solutions at each of the earlier makespans. By the 30th generation, the Pareto front is further improved, and there is but a slight difference between the results of the 50th and 100th generation.

In terms of the computational time of the algorithm, it is observed that the increase in computational time is almost linear with respect to the number of generations run in the algorithm. The result of the computational time against the number of generations is shown in Figure 7.13.

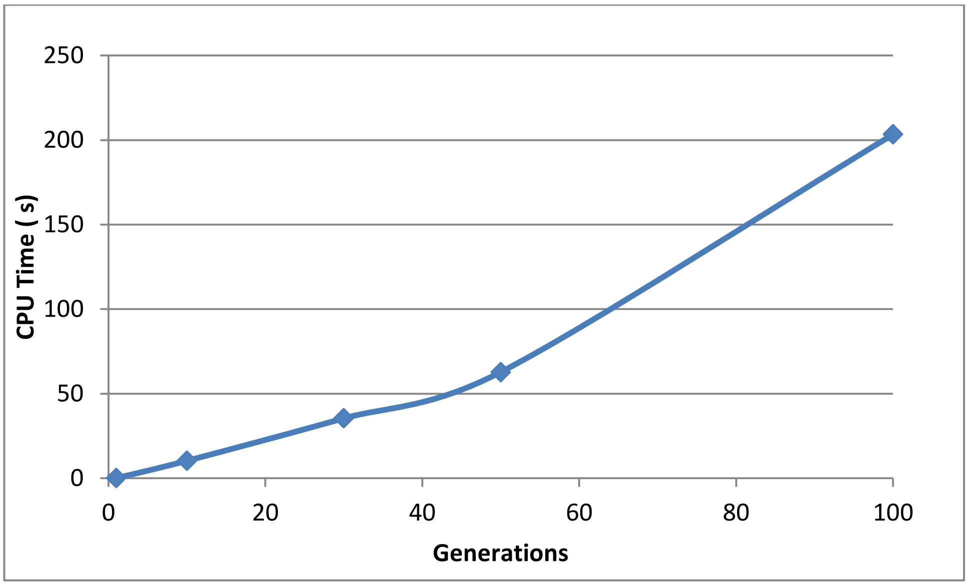
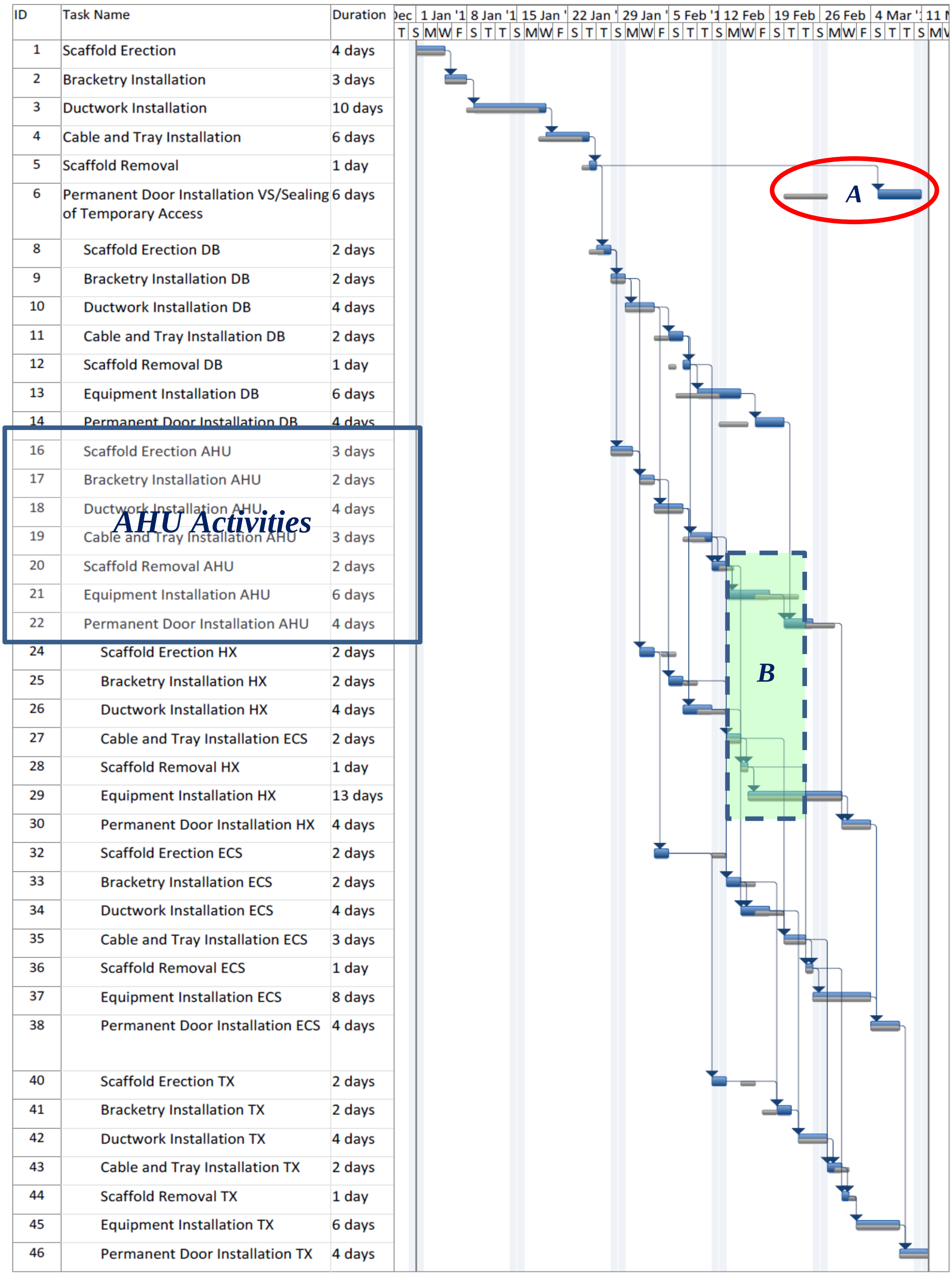


Figure 7.13. CPU Computational Time

For analysis, two solutions will be chosen for comparison, with the activities from each solution superimposed one over the other. The resultant Gantt Chart is shown in Figure 7.14. The first candidate solution is taken from the 30th generation (schedule with lighter shade task bars at the bottom), while the second candidate solution is from the 100th generation (schedule with darker shade task bars at the top). The candidate solution of the 100th generation displays the better solution (CPI_{Avg} value of 0.0182) as compared to the solution in the 30th generation (CPI_{Avg} value of 0.0176).

The difference in the values can be deduced from the case data. The Air Handling Unit (AHU) activities are typically more workspace intensive as the equipment and labour requirements are much higher. In particular, the equipment installation activities of the AHU and Heat Exchanger (HX) require major lifting operations. One reason for the improved schedule in the 100^{th} generation candidate solution is that the two major activities are staggered (as shown in the shaded box B in Figure 7.14). Another reason for the improved schedule is that the algorithm suggests a construction strategy where the Permanent Door Installation of the Ventilation Shaft (VS) is delayed as much as possible (this is denoted by the oval A). This delay will also result in a staggering of this activity away from the other major activities taking place simultaneously.



ID	Task Name	Duration
1	Scaffold Erection	4 days
2	Bracketry Installation	3 days
3	Ductwork Installation	10 days
4	Cable and Tray Installation	6 days
5	Scaffold Removal	1 day
6	Permanent Door Installation VS/Sealing of Temporary Access	6 days
8	Scaffold Erection DB	2 days
9	Bracketry Installation DB	2 days
10	Ductwork Installation DB	4 days
11	Cable and Tray Installation DB	2 days
12	Scaffold Removal DB	1 day
13	Equipment Installation DB	6 days
14	Permanent Door Installation DB	4 days
16	Scaffold Erection AHU	3 days
17	Bracketry Installation AHU	2 days
18	Ductwork Installation AHU	4 days
19	Cable and Tray Installation AHU	3 days
20	Scaffold Removal AHU	2 days
21	Equipment Installation AHU	6 days
22	Permanent Door Installation AHU	4 days
24	Scaffold Erection HX	2 days
25	Bracketry Installation HX	2 days
26	Ductwork Installation HX	4 days
27	Cable and Tray Installation ECS	2 days
28	Scaffold Removal HX	1 day
29	Equipment Installation HX	13 days
30	Permanent Door Installation HX	4 days
32	Scaffold Erection ECS	2 days
33	Bracketry Installation ECS	2 days
34	Ductwork Installation ECS	4 days
35	Cable and Tray Installation ECS	3 days
36	Scaffold Removal ECS	1 day
37	Equipment Installation ECS	8 days
38	Permanent Door Installation ECS	4 days
40	Scaffold Erection TX	2 days
41	Bracketry Installation TX	2 days
42	Ductwork Installation TX	4 days
43	Cable and Tray Installation TX	2 days
44	Scaffold Removal TX	1 day
45	Equipment Installation TX	6 days
46	Permanent Door Installation TX	4 days

Figure 7.14. Gantt Chart Comparing Candidate Solutions of 30th (lighter shade/bottom) and 100th Generation (darker shade/top)

7.6. Concluding Remarks

This chapter presents a genetic algorithm framework for solving the Dynamic Construction Requirements problem under spatial temporal constraints. The purpose of this chapter is to provide a method of resolving work site congestion issues arising from the tightly packed schedule through a meta-heuristic means. This is achieved by allowing activities to move around within permissible feasible bounds to improve the objective value represented by the congestion penalty indicator, CPI_{Avg}. The applicability of the method is shown through a performance analysis.

One novelty of the method lies in the design of the chromosome which makes use of the Bounds Consistency algorithm in BCSolver as a decoding mechanism to ensure temporal feasibility, while allowing the constraint handling mechanism within the NSGA-II algorithm to handle the workspace constraint violations. This design facilitates learning within the GA, and may be a promising avenue for further research.

Several extensions to the above can be made. Firstly, the problem only deals with work space resources, while the PDM++ deals with the key resource considerations. Multiple pooled resources may be incorporated into the multiple resources by employing the Serial Schedule Generation Scheme with Unscheduling (S-SGSU) devised by Ballestín, *et al.* (2011).

Further analysis of the algorithm is also possible by comparing it with an exact method to better determine the performance of the algorithm. However, as the purpose of this research is to propose a method to solve the DCR-ST problem to near-optimality rather than to propose a method to outdo existing ones, the comparison between exact and other heuristic methods will be left to future work.

Chapter 8. Case Study and Analysis

8.1. Introduction

This chapter presents detailed results for three case studies to demonstrate the key concepts from earlier chapters. The first case study demonstrates the applicability of the quantification methodology for minimising workspace conflict and analysis on a schedule repair problem.

The second case study describes the temporal modelling of construction requirements on a pipe rack installation project. The modelling framework presented in Chapter 5 is used to show how the temporal aspects of construction requirements can be explicitly embedded in a construction plan, resulting in the modelling of alternative schedules along with the representation of other complex constraints. The case study is evaluated using the prototype discussed in Chapter 6, and a discussion of how the PDM++ model differs from the traditional PDM is also carried out.

The final case study describes the optimization of a schedule subject to both the spatial and temporal aspects of construction requirements using the modified multi-objective genetic algorithm developed earlier in Chapter 7. The model allows the Planner to draw conclusions about the interaction of space and time from alternative modes of construction work sequences. General strategies for minimizing congestion are then discussed in relation to the problem.

8.2. Case Study 1: Minimising Congestion during Schedule Repair for Internal Refurbishment of Oil Refinery Reactor Column

The oil refinery refurbishment example in Chapter 4 is now solved to optimality. The multi-objective genetic algorithm proposed in Chapter 7 is used to solve this problem by restricting the float in the planning horizon to zero (refer to Equation 7.15). This means that the activities on the critical path maintain zero float, and are not allowed to extend beyond the early start project makespan; only non-critical activities may be rearranged within the bounds of their available float with the objective of minimizing the worksite congestion. This transforms the problem from a multi-objective problem into a single objective problem of minimizing the congestion penalty index. Such a problem may be termed as a schedule repair problem.

In this case study, the objective function is modified as shown:

$$fitness = CPI_{Total} + \left(\text{Penalty} \times \sum_i \{ Start_i + Dur_i \} \right) \quad \textbf{(8.1)}$$

A penalty function is added to the objective comprising the CPI_{Total}. This penalty function is a product comprising the sum of the start of the activities and an arbitrarily small penalty value to create schedule pressure to the early start. This schedule pressure means that the algorithm ranks the solutions with earlier activity start times higher. This ensures a one-to-one mapping of the chromosome space to the solution space, so that two solutions exhibiting the same CPI_{Total} value (fitness value) can be differentiated, with the solution having earlier start times preferred.

An arbitrarily small penalty value of 0.0001 was chosen in this case study. When choosing such a penalty value, care should be taken such that the magnitude of the sum of the start times does not overly influence the CPI_{Total} value. It is expected that the

CPI_{Total} value in this problem has a "worst-case" magnitude of order 1 (i.e. up to 10). Similarly, the sum of the start times has a "worst-case" magnitude of order 3 (i.e. up to 1000). Hence, the penalty value should have a magnitude of order -3 or smaller. For conservativeness, a penalty value of order -4 was chosen. In practice, using the idea of a penalty value is reasonable as it is reflective of the Planner's preference.

The original problem presented in Chapter 4 is straightforward with only 2 non-critical activities. To make this case study more interesting, the following changes were made: To expedite the work, the Planner has suggested a new construction method to allow for concurrent work to be carried out. However, additional preventive/safety measures have been put in place to ensure that the concurrent work can be carried out safely. These preventive measures have caused one of the activities (*Trim_Baffle*) to be longer than in the original plan presented in Chapter 4.

Figure 8.1 shows the Planner's proposed schedule. A window of interest is identified as shown in the figure, where the activities in consideration are not critical, and are thus available for temporal rescheduling. This window consists of 7 activities with variable start times. Only one of the activities is critical (Trim existing baffle plates) while the others have available float. Between these activities, there are 18 workspaces and pathspaces with the properties shown in Table 8.1. For schedule repair, the activities are to respect the precedence constraints between them, but cannot extend beyond the "Trim existing baffle plates" activity, as this would cause them to be on the critical path and unnecessarily delay the overall project schedule.

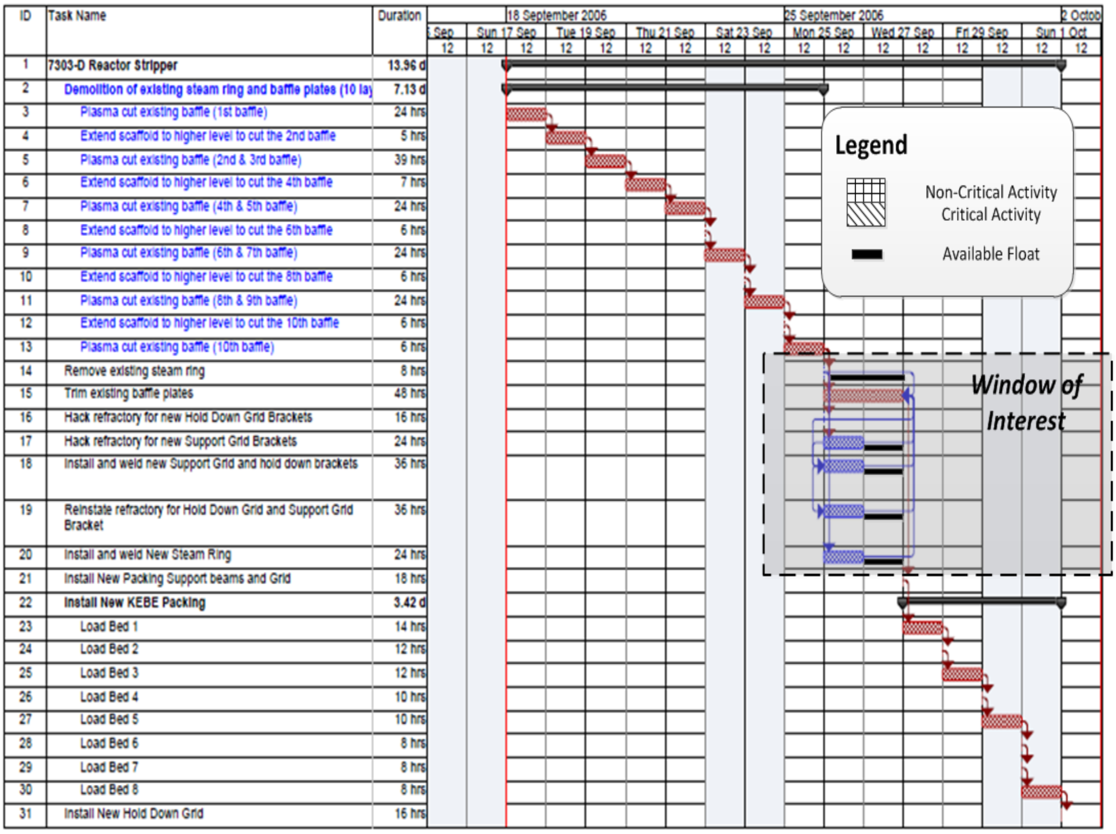


Figure 8.1. Gantt Chart of Proposed Alternative with Time Window of Interest

In general, the sequence of work involved segregating the workspace into two, an upper workspace containing the Hold Down Grid and its supporting brackets, and a lower workspace containing the Support Grid with its brackets, via a protective system put in place during the "Trim existing Baffle Plates" activity. An opening through the protective system allowed the workers to access the upper workspace as shown in the schematic (Figure 8.2). The protective system was later removed during the installation activity of the support grid.

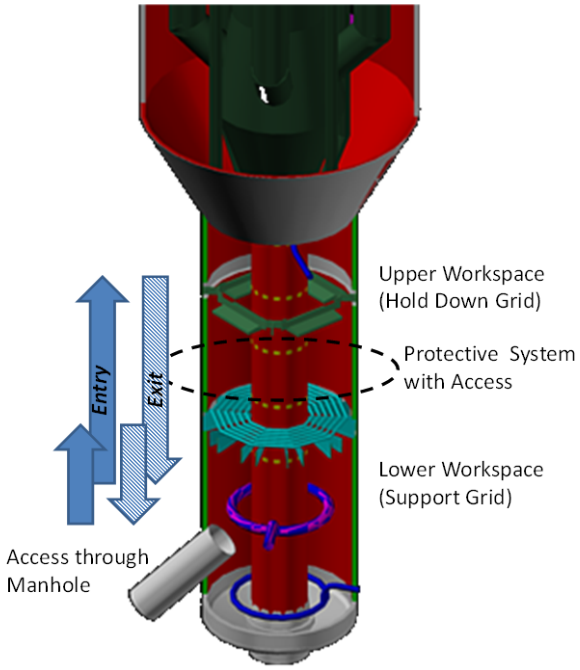


Figure 8.2. Workspace Access Schematic

Table 8.1. Properties of Space Entities

Work Package Entities	Volume(m^3)	U_t	U_s	ρ
Trim_Baffle_WS	59.32	1.000	0.032	0.180
Refactory_Remove_WS_Upper Workspace	9.72	1.000	0.198	0.445
Refactory_Remove_WS_Lower Workspace	9.72	1.000	0.198	0.445
SupportBracket_WS	10.22	1.000	0.188	0.433
SteamRing_Removal_WS	7.37	1.000	0.260	0.510
HoldDown Bracket_WS	10.22	1.000	0.188	0.433
Trim Baffle_PS	14.63	0.300	0.131	0.199
SupportBracket_PS	18.79	0.300	0.102	0.175
Steamring_Removal_PS	4.86	0.300	0.395	0.344
Refactory_Removal_PS_ Upper Workspace	11.00	0.300	0.174	0.229
Refactory_Removal_PS_ Lower Workspace	18.79	0.300	0.102	0.175
HoldDown Bracket_PS	11.28	0.300	0.170	0.226
Refactory_Install_PS_ Upper Workspace	11.00	0.300	0.175	0.229
Refactory_Install_PS_ Lower Workspace	18.79	0.300	0.102	0.175
Refactory_Install_WS_ Upper Workspace	9.72	1.000	0.201	0.448
Refactory_Install_WS_ Lower Workspace	9.72	1.000	0.198	0.445
Steamring_Install_WS	7.37	1.000	0.163	0.404
Steamring_Install_PS	4.86	0.300	0.247	0.272

The genetic algorithm ran with population size of 500 over 200 generations with the results generated as per Figure 8.3. The GA was able to improve the solutions found, finally arriving at a schedule with a congestion penalty index of value 2.623093. The resultant schedule for the activities in the window of interest is illustrated in the Gantt Chart of Figure 8.4. In Figure 8.4, it can be seen that the activities are staggered to reduce the overlapping of the interfering space entities.

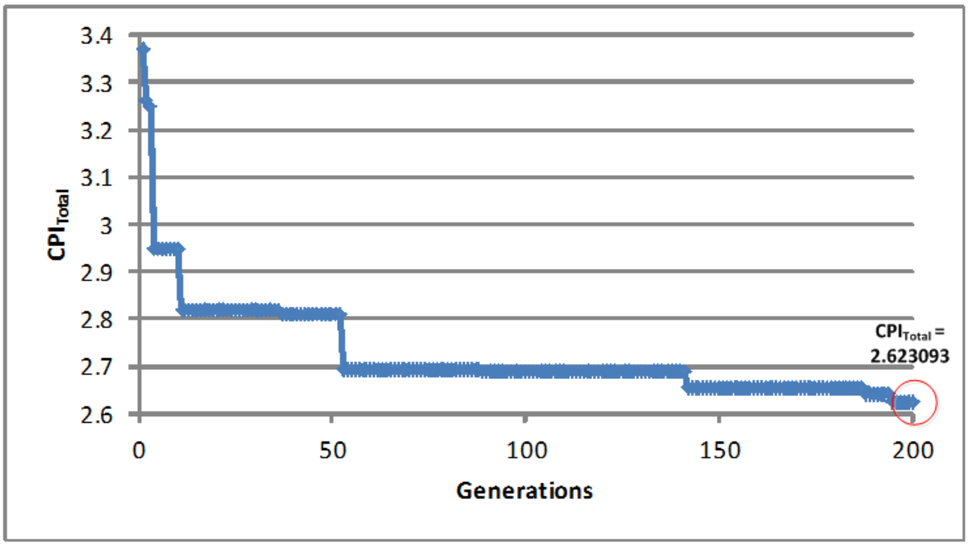


Figure 8.3. Convergence of CPI_{Total} over 200 Generations in Schedule Repair Case

Remove existing Steam Ring	8hrs
Trim existing Baffle Plates	72hrs
Hack Refactory for new Hold Down Grid Brackets	16hrs
Hack Refactory for new Support Grid Brackets	24hrs
Install and Weld new Support Grid and Hold Down Grid Brackets	36hrs
Reinstate Refactory for Support Grid and Hold Down Grid Brackets	36hrs
Install and Weld New Steam Ring	24hrs

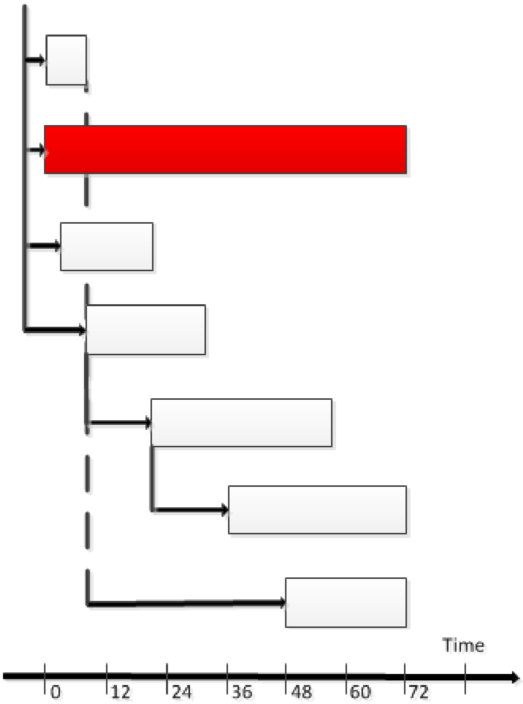


Figure 8.4. Gantt Chart showing Improved Schedule after 200 Generations

8.2.1 Effect of Consuming Float on Congestion

The case study in the preceding section shows the applicability of the genetic algorithm resolution methodology for schedule repair to minimize the conflict arising from workspace congestion. In this section, a comparison is made between the early start schedule and the best sequence found by the genetic algorithm. The effect of temporally delaying activities in the construction schedule on lowering the amount of congestion onsite will be analysed. Delaying the activities consumes the float times available, but is able to reduce the amount of temporal overlap between the activities, resulting in lower CPI_{Total} computation.

For comparison, the initial early start schedule and the improved schedule are compared to demonstrate how the improvement between the schedules was achieved. From Chapter 4, the CPI_{Total} computation is dependent upon the individual Dynamic Space Interference (DSI) indicators (Equation 4.9), and these are used to analyse the effects of temporally delaying the activities. The results are shown in Table 8.2.

Table 8.2. Comparison of DSI for Early Start Schedule and Improved Schedule

Work Package Entities	DSI (Early Start Schedule)	DSI (Improved Schedule)	Difference in DSI
Trim_Baffle_WS	0.4523	0.3599	0.0923
Refactory_Remove_WS_Upper Workspace	1.1799	0.7142	0.4656
Refactory_Remove_WS_Lower Workspace	1.2282	0.8909	0.3373
SupportBracket_WS	1.1564	0.9430	0.2134
SteamRing_Removal_WS	1.3238	0.9292	0.3945
HoldDown Bracket_WS	1.0939	0.8288	0.2651
Trim Baffle_PS	0.7507	0.5732	0.1776
SupportBracket_PS	0.6271	0.4710	0.1562
Steamring_Removal_PS	1.2859	0.8966	0.3893
Refactory_Removal_PS_ Upper Workspace	0.7646	0.4617	0.3029

Refactory_Removal_PS_ Lower Workspace	0.8891	0.4614	0.4277
HoldDown Bracket_PS	0.7203	0.5455	0.1748
Refactory_Install_PS_ Upper Workspace	0.7298	0.5256	0.2042
Refactory_Install_PS_ Lower Workspace	0.7145	0.4528	0.2617
Refactory_Install_WS_ Upper Workspace	1.1299	0.8163	0.3136
Refactory_Install_WS_ Lower Workspace	1.1915	0.808	0.3834
Steamring_Install_WS	1.0880	0.6401	0.4480
Steamring_Install_PS	1.0661	0.6189	0.4472

Intuitively, DSI can be thought of as an abstraction of the ratio of space demand to availability placed on a space-time-volume. Recall that the space-time-volume can be thought of as a multi-dimensional volume containing the product of the spatial and temporal dimensions. This means that DSI values exceeding 1 have a greater demand than the availability of the space-time-volume.

From the results of Table 8.2, the early start schedule is infeasible, and subject to high amounts of congestion. 10 of the 18 work package entities exceed 1. Through shifting the activity start times within their available float, the improved schedule is able to reduce the DSI values. Now, none of the space entities are infeasible with respect to congestion, and reductions of up to 46% are achieved in terms of DSI values. However, some of the work package entities are still indicative of potentially high values of congestion (DSI values more than 0.85), and these require greater attention from the Planner.

The work package entities with potentially high values of congestion are identified as: *Refactory_Remove_WS_Lower_Workspace*, *SupportBracket_WS*, and *SteamRing_Removal_WS*. For the *SteamRing_Removal_WS*, the high DSI value is due to its inherently high utilization (From Table 8.1, 0.5104). However, for *Refactory_Remove_WS_Lower_Workspace*, and *SupportBracket_WS*, an additional consideration is the amount of spatial-temporal overlap with other interfering entities.

As a counter-example, *Refactory_Install_WS_Upper_Workspace* has a higher utilization, but due to its lower amount of interference, has a lower DSI value compared to *Refactory_Remove_WS_Lower_Workspace*, and *SupportBracket_WS*.

In summary, the proposed congestion penalty indicator CPI allows a quantitative measure of worksite congestion to be used within an optimization problem. The resolution of the optimization problem via a genetic algorithm search provides a reasonable schedule which is able to avail a mitigated strategy through the temporal arrangement of the activities to reduce the congestion problem.

8.3. Case Study 2: Piperack Installation

Another illustrative case study is presented to demonstrate the applicability, expressiveness and power of PDM++ for capturing schedule constraints arising from construction requirements. This case study is based on a gas pipeline installation in a refinery, stretching over 300m. The contractors are to extend the existing piping to allow for new pipes to be placed. The activities are divided into 5 sections as shown in Figure 8.5. Sections 1 and 5 refer to the construction of two concrete pipe bridges over existing water channels (shaded in Figure 8.5) with its associated foundation and subsequent pipeline installation phases. The main pipeline consists of a steel pipe rack with shallow foundations, and is divided into three sections (Sections 2 to 4). Scaffolds are to be erected to enable welding work.

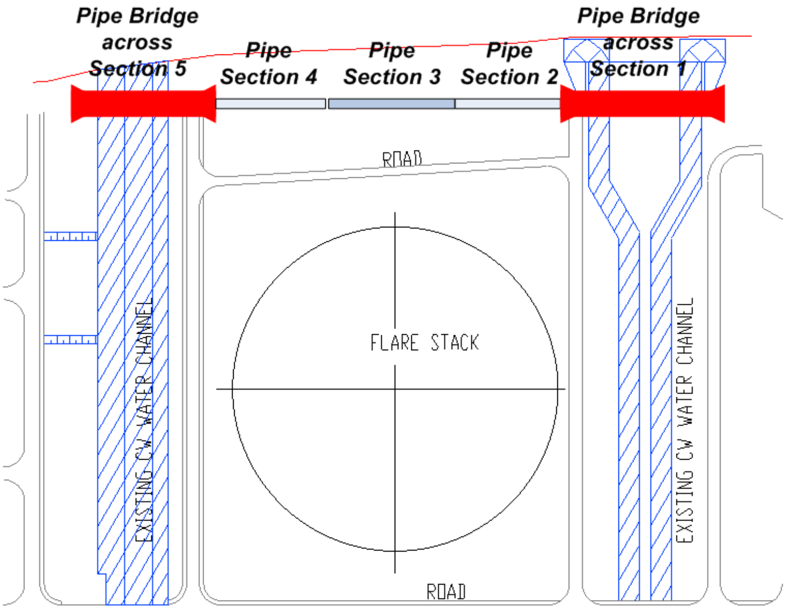


Figure 8.5. Elevation View of Pipeline Installation Layout

There are several main activities in the construction process. The abbreviation of each activity is provided in brackets with a suffix denoting the section in which the activity is carried out; the number in square brackets denotes the activity duration in days.

1. The construction of the bridge foundation (BF1[10], BF5[10])

2. The construction of the Pipe Bridges (PB1[7], PB5[7])

3. The construction of pipe support and foundation system (PF2[9], PF3[9], PF4[9])

4. The construction of the steel pipe rack structure (PSS2[5], PSS3[5], PSS4[5])

5. The erection and dismantling of the scaffold in each section for supporting the construction of the pipe rack (ES2[3], ES3[3], ES4[3]

and DS2[1], DS3[1], DS4[1] respectively) which is further modelled by 3 meta-intervals (S2, S3, S4) depicting the usable periods of the scaffold (the interval between the erection of the scaffold and the commencement of its dismantling)

6. The pipe installation process (PI1[3], PI2[3], PI3[3], PI4[3], PI5[3]), with a meta-interval representing the complete installation sequence ({Alt}). {Alt} is used to model the alternate sequence available for the pipe installation process.

7. An optional Protective Staging activity (PS[3]) if PI1 is installed after Day 60 (refer to R4 below).

There are several major requirements foreseen by the Engineer:

R1. Only one micropiler, a key resource, is available for the bridge foundation work, although work can begin with Section 1 or 5.

R2. The shallow foundation phases for the piperack are to be done sequentially from Sections 2 to 4 due to routing issues and enforcing work continuity of the crew.

R3. The pipeline installation may start on either side of existing infrastructure, i.e. from Sections 1 to 5 or from Sections 5 to 1.

R4. Additional protective measures are required if pipeline in Section 1 is installed after Day 60 due to tight spatial constraints with an adjacent project.

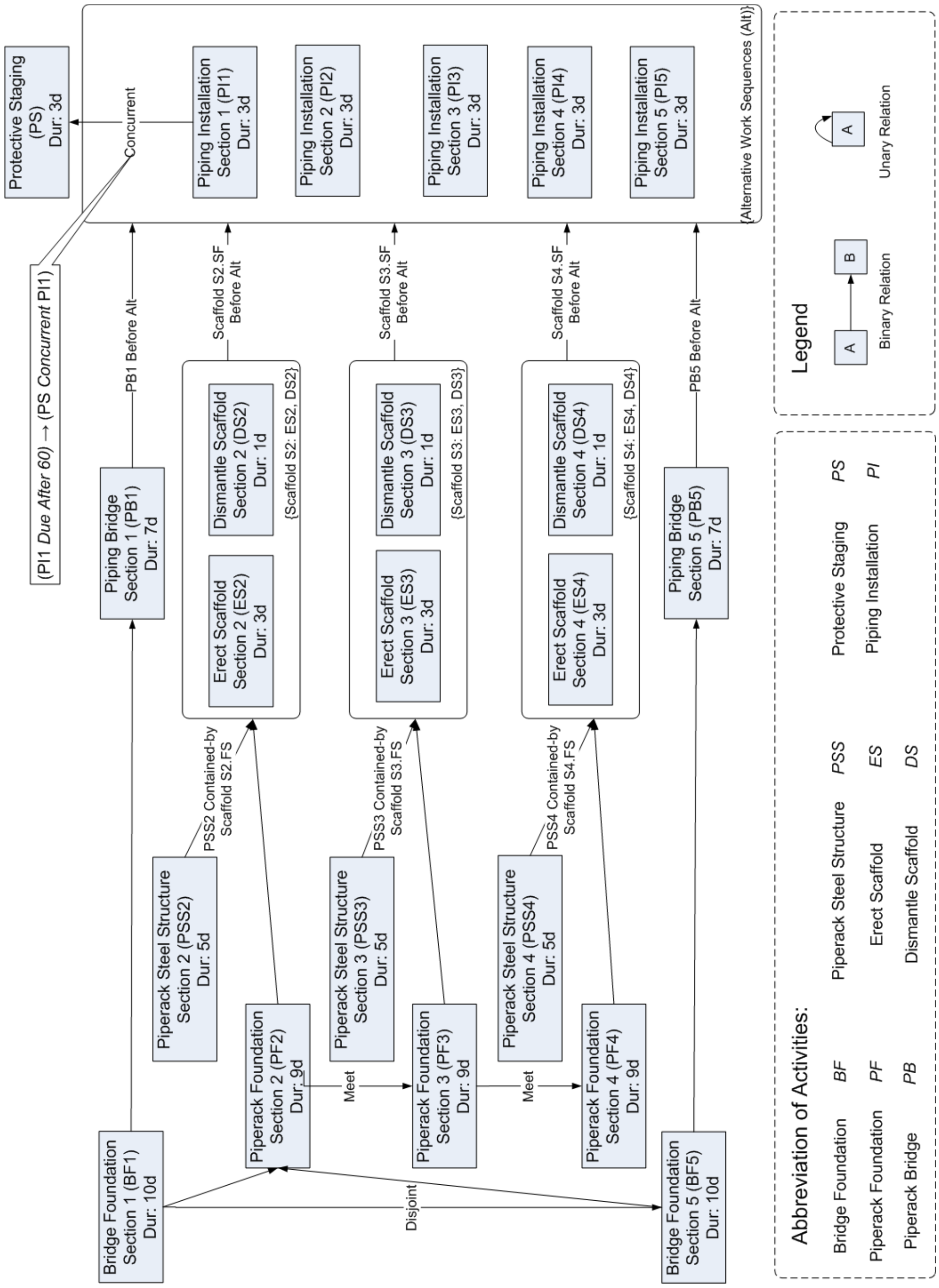


Figure 8.6. PDM++ Constraint Network

Figure 8.6 above depicts the resulting constraint network based on the schedule implications of the above construction requirements. The activities are abbreviated as shown in the Figure. Annotated arcs depict the new PDM++ relationship types, while the non-annotated arcs represent simple precedence or *"Before(0)"* relationships.

Requirement R1 is depicted by a "*Disjoint*" relationship between BF1 and BF5:

R1: **(BF1 *Disjoint* BF5)** **(8.2)**

which is used to model the effect of having only one available micropiler to support the construction of BF1 and BF5. From the schedule perspective, either activity may be chosen to begin first, but both cannot overlap with each other.

Requirement R2 is depicted by "*Meets*" relationships between PF2, PF3 and PF4:

R2: **(PF2 *Meets* PF3) ∧ (PF3 *Meets* PF4)** **(8.3)**

This constrains the activities to be continuous.

The Alternative Work Sequences (arising from Requirement R3) for the pipeline installation from Sections 1 to 5 is defined and depicted as a meta-interval {Alt: (PI1, PI2, PI3, PI4, PI5)} in Figure 8.6. The association between the activities from Sections 1 to 5, and from Sections 5 to 1 within the meta-interval is represented as:

R3: **((PI1 *Before* PI2) ∧ (PI2 *Before* PI3) ∧ (PI3 *Before* PI4) ∧ (PI4 *Before* PI5)) ∨ ((PI1 *After* PI2) ∧ (PI2 *After* PI3) ∧ (PI3 *After* PI4) ∧ (PI4 *After* PI5))**

(8.4)

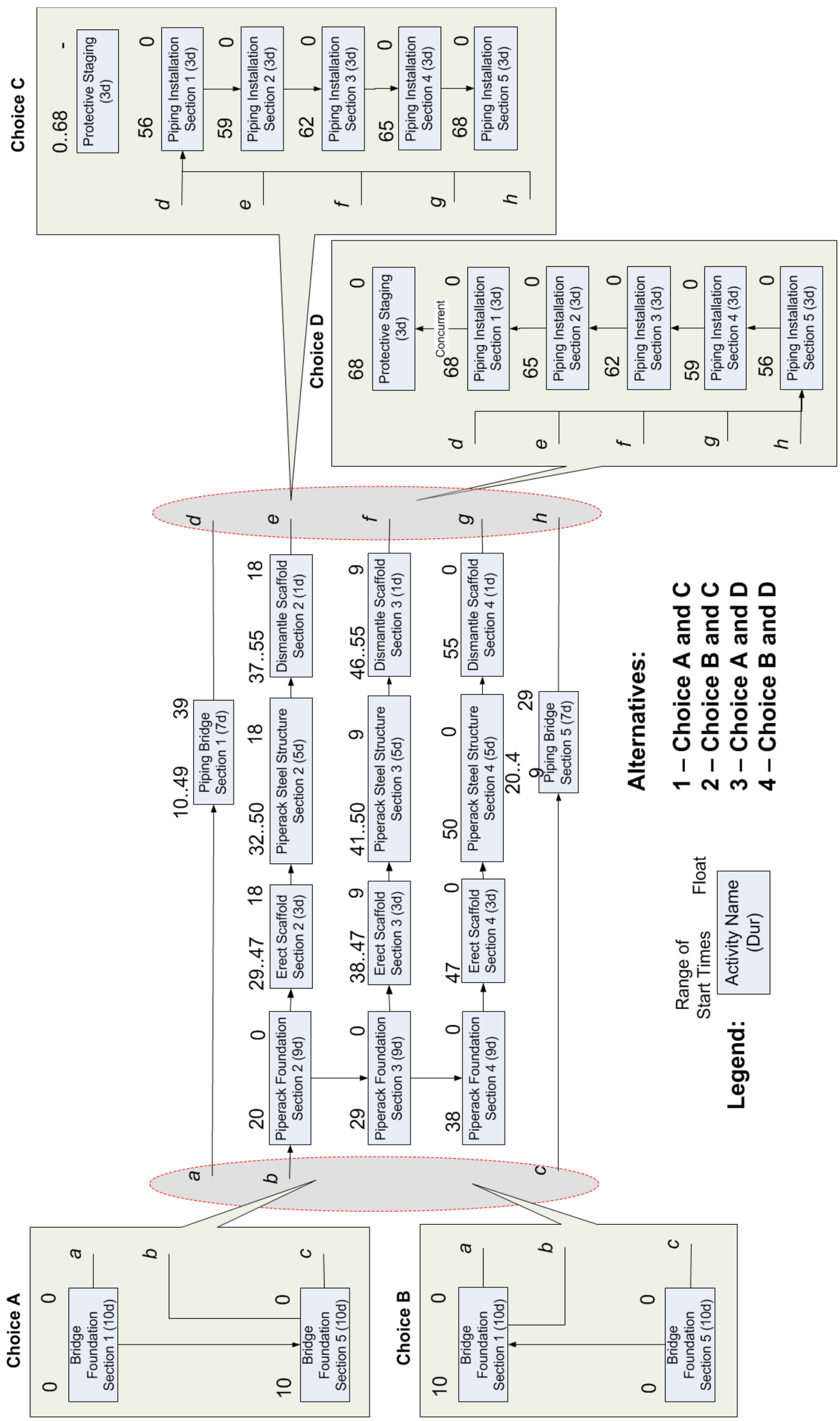


Figure 8.7. Activity Solutions indicating Start Intervals and Floats

In the same way, the protective staging activity (of Requirement R4) becomes necessary if the pipeline installation in Section 1 finishes after Day 60. In other words, the protective staging (PS) must happen concurrently with the pipeline installation of Section 1 (PI1) if and only if the pipeline installation ends after Day 60. In some instances when the pipeline is completed before the given due date, PS will not be necessary. This planning logic may be represented as:

R4: **(PI1 *Due-After 60)* → (PS *Concurrent* PI1)** **(8.5)**

Furthermore, the scaffolding requirements for the installation of the piperack steel structure from Sections 2 to 4 (PSS2, PSS3, PSS4) may be represented by three meta-intervals specified using the representation schema introduced in Chapter 5:

{Name.Interval: Comprised Activities, Start Activity, End Activity} **(8.6)**

Generally these are:{Scaffold S2: (ES2, DS2), ES2, DS2}, {Scaffold S3: (ES3, DS3), ES3, DS3}, {Scaffold S4: (ES4, DS4), ES4, DS4}. Since the installation has to be carried out within the usable duration of the scaffolds (defined by the FS interval), this requirement for Section 2 can be expressed as:

(PSS2 *Contained-by* Scaffold S2.FS) **(8.7)**

Moreover, the pipe installation (PI1 to PI5) requires that the structural works are completed and the scaffolds dismantled. Since the pipe installation activities are comprised within the meta-interval, Alt, the requirement for the removal of the scaffold at Section 2 can be simplified as:

(Scaffold S2.SF *Before* Alt) **(8.8)**

where a single constraint is used to draw the relationship between the two groups of activities. Effectively, Equation 8.8 means that the scaffold activities must be completed before the start of any of the pipe installation activities.

The temporal meaning of the above Equations 8.7 and 8.8 are also illustrated in the following Figure 8.8. Consequently, the scaffolding requirements for the other sections may be similarly expressed. The meta-interval representation enables succinct model representation through a simple and clear depiction of the relationships between the scaffold resources and other activities.

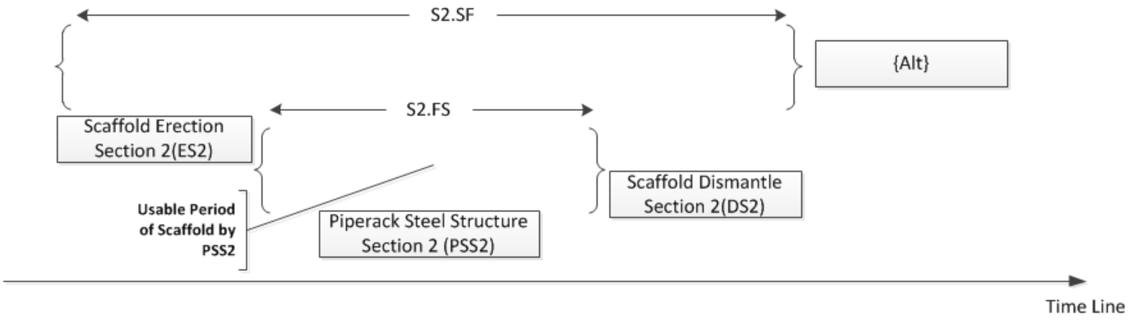


Figure 8.8. Illustration of Scaffold Requirements using Meta-Interval

To illustrate the comprehensiveness of defining the data inputs into the system, the following figure (Figure 8.9) shows the breakdown of requirements as constraints, and the definition of the meta-intervals with the activities for input into ECL^iPS^e.

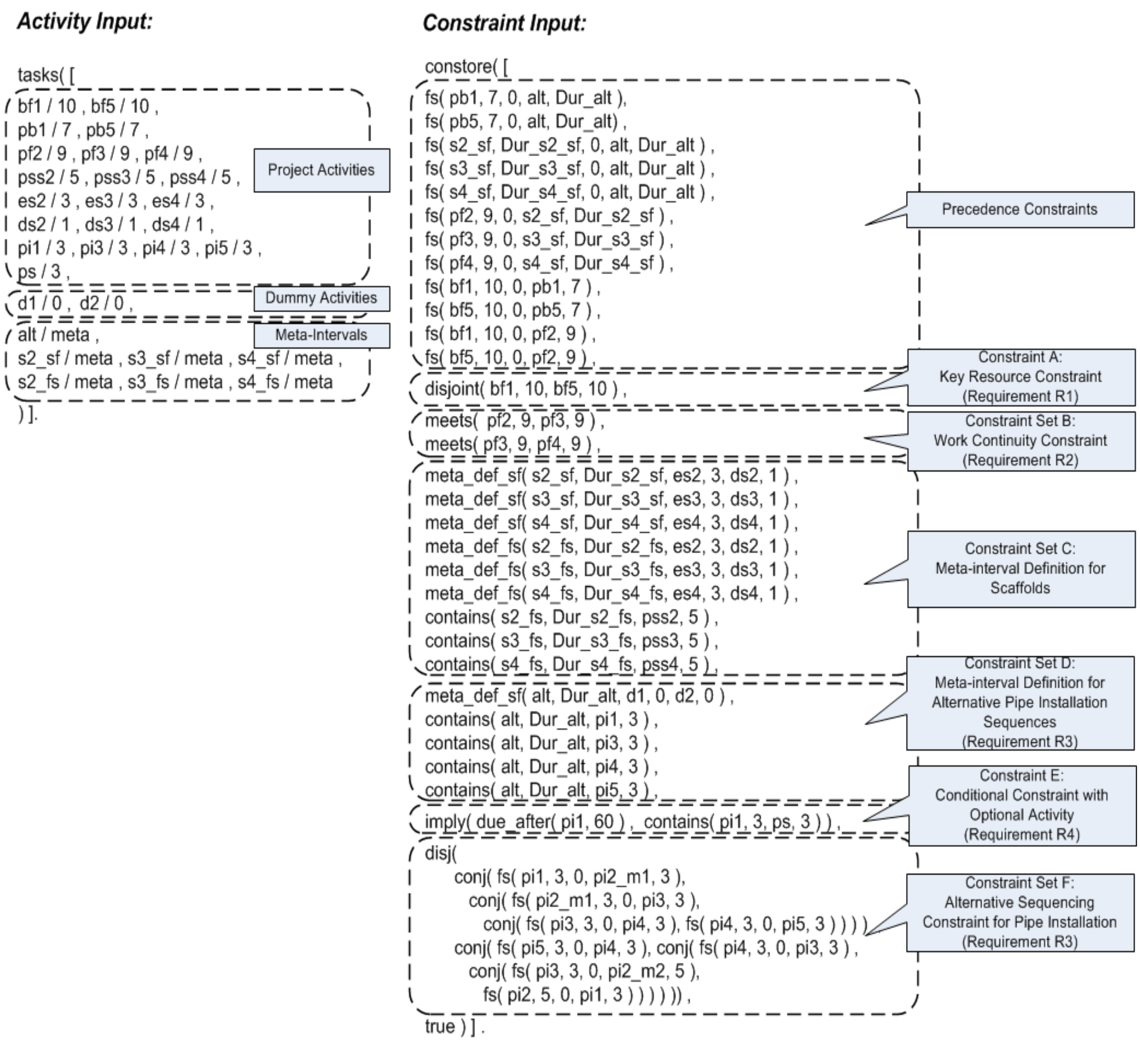


Figure 8.9 Activity and Constraints input in *ECL^{i}PS^e* for Case Study

8.3.1 Discussion and Analysis of Case Study 2

Minimizing the project makespan, the above network offers 4 different alternative plans, each with a makespan of 71 days. The result of the analysis is presented in Figure 8.7. The domain values in "Range of Start Time" refer to the feasible starting times for each activity. For example, PSS2 has a start interval from {32 .. 50}, indicating an early start at 32 days and a late start at 50 days. The float for each activity is also shown in Figure 8.7. and activities with 0 float designated as critical. The 4 alternatives are composed from the 2 alternative choices arising from

Requirement R1 (Choices A and B) and 2 alternatives from R3 (Choices C and D). Figure 8.7 is not intended to be a CPM diagram, but rather a depiction of the alternative sequence of activities.

Alternatives 1 and 3 begin with activity BF1 while Alternatives 2 and 4 begin with the alternate activity BF5 (due to R1). Alternatives 1 and 2, and Alternatives 3 and 4 differ in the sequence for the pipeline installation with the former pair following the sequence Sections 1 to 5 while the latter pair follow the alternate sequence from Sections 5 to 1 (due to R3). Since PI1 occurs after Day 60 in Alternatives 3 and 4, the PS becomes necessary and runs concurrently with PI1 starting on Day 68. On the other hand, the PS has a domain value of {0 .. 68} in Alternatives 1 and 2. This activity is either independent of all other activities, or is unnecessary. From R4 the latter can be concluded.

8.3.2 Model Comparison with traditional PDM

This section provides a short discussion on the differences between the model developed by PDM++ and traditional PDM, using the case study as a point of comparison. The following Figure 8.8 shows the original PDM developed for the above case study, with the critical path highlighted by bold arrows. The original PDM was planned with the requirements R1 to R4 in mind, and the resulting sequence of work is similar to that of Alternative 3. In both cases, PDM and PDM++ result in the same makespan of 71 days. While Figure 8.6 (PDM++) and Figure 8.10 (PDM) look similar, some differences are noted.

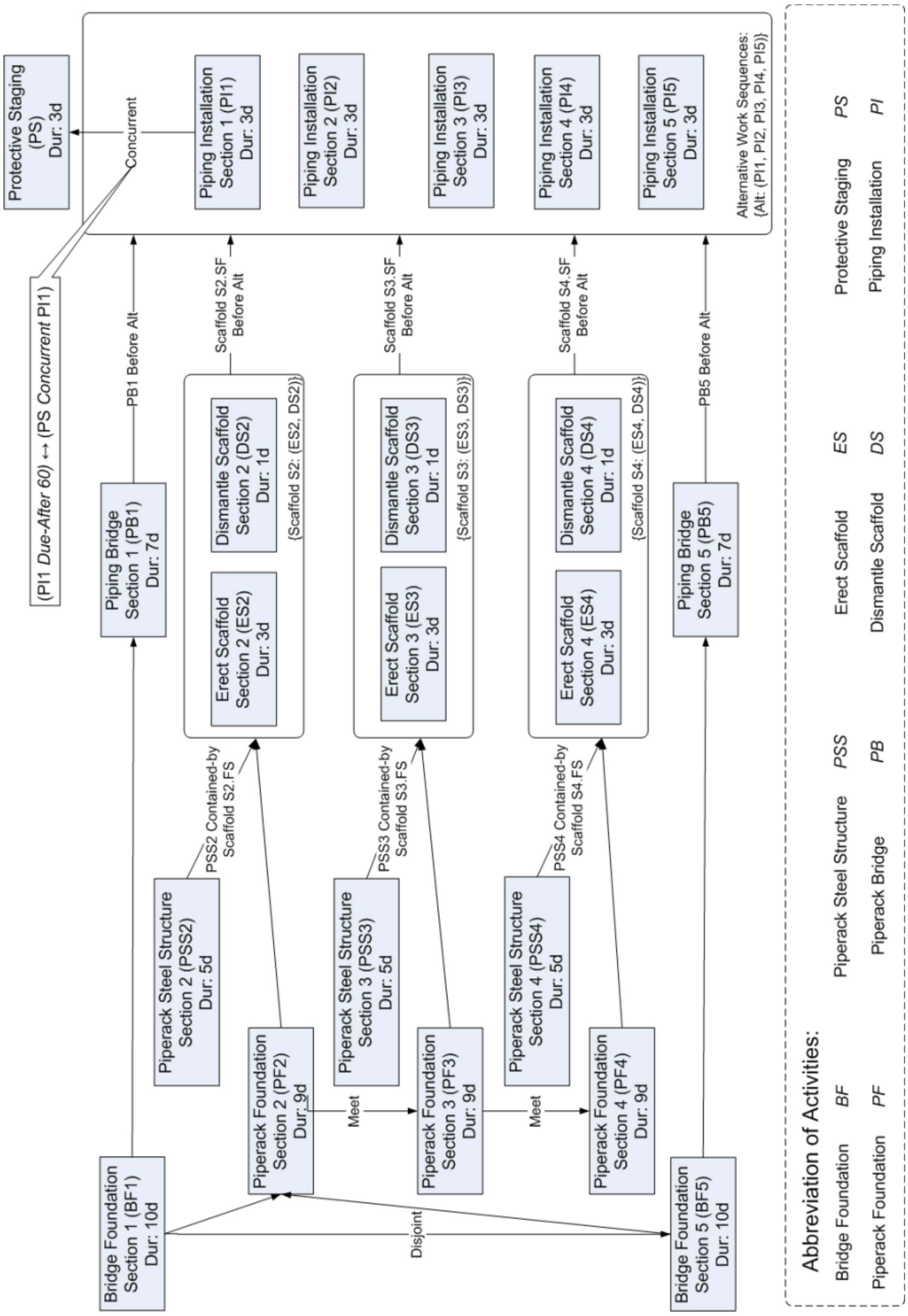


Figure 8.10. Original PDM Network Model

The traditional PDM plan in Figure 8.10 only displays a single planning logic even when multiple logics exist (4 alternatives). From the construction requirements, the PDM++ model (Figure 8.6) is able to provide greater semantic description which allows for the generalisation of multiple planning logics.

In R1, the "*Disjoint*" relationship generalises the existence of two different planning logics, where one logical path starts with BF1 and the other with BF5. In R3, it is possible to start pipe installation from Sections 1 to 5 or from Sections 5 to 1. This relationship characterizes the disjunctive nature of construction requirements, which cannot be adequately represented using PDM.

Dynamic requirement R4 cannot be adequately captured in Figure 8.10. Firstly, traditional PDM does not allow for conditions to be built into the logic. Secondly, the SS0 and FF0 logics in Figure 8.10 which is used to represent concurrency, forms a loop, which would give rise to errors in PDM. Hence, PDM++ is demonstrated to be able to capture interdependent requirements in the form of conditional logic between temporal constraints. If proper consideration of the construction requirements was given as with PDM++ the planners could have done without Protective Staging.

In Figure 8.6, the "*Meet*" relationship is used to enforce work continuity between the activities which cannot be implemented in PDM. Due to the activities being on the critical path, this construction requirement is satisfied superficially in PDM (shown in Figure 8.10). PDM++, however, will be able to enforce the "*Meet*" relationship for work/resource continuity even if the activity is not critical.

8.4. Case Study 3: Congested MEP Installation in Underground MRT Station

This case study involves a Mechanical, Electrical and Piping (MEP) construction involving the Power Supply, Transmission Services and Heating, Ventilation and Air Conditioning (HVAC) Installation processes in a section of a Mass Rapid Transit (MRT) subway station 30m below ground level. MEP installation processes typically commence after the major building works have been completed, and involves erecting of scaffolding within the work site, followed by the ceiling bracketry installation, ducting works, cable tray with cabling works, dismantling of scaffold, HVAC equipment installation, and the occasional permanent door installation in that order. Each task is handled by a specialist crew under the supervision of a coordination manager (Planner). The following Figure 8.11 shows the layout of the case study area.

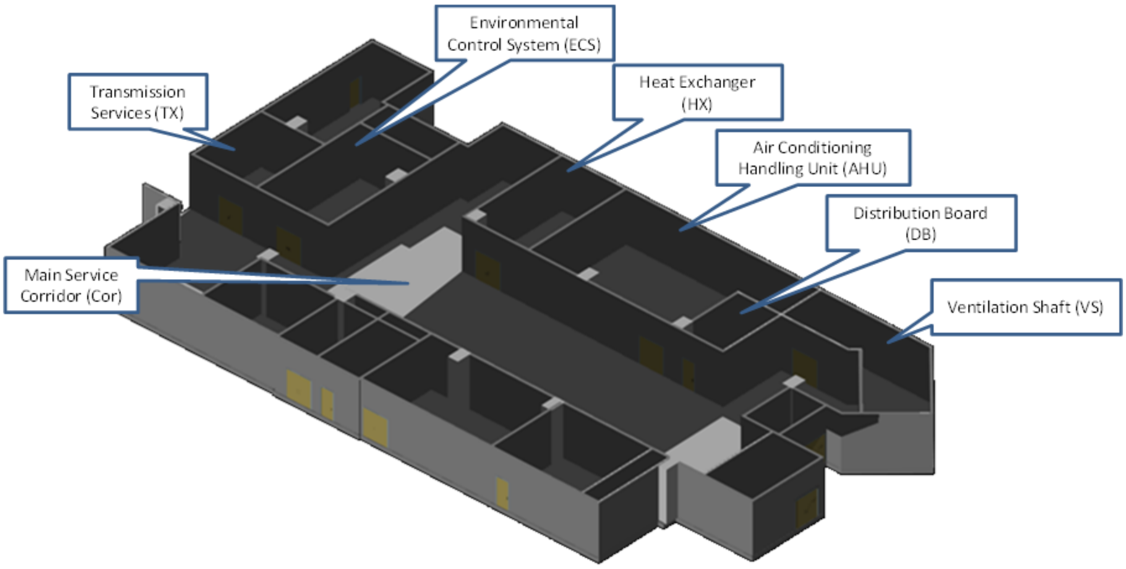


Figure 8.11. Site Layout

In general, the Main Service Corridor (Cor) contains the main ventilation ducts, water pipes, cable trays and cables for power and other services. Its location serves as the main conduit for connecting all the services leading into and out of the other rooms adjacent to the corridor. The Transmission Services (TX) room contains the data transmission cables, co-axials etc needed for the proper running of the systems. There are three environmental management systems for managing indoor air quality involved in this project. These systems are connected via fresh air ducts and exhaust air ducts to the Ventilation Shaft (VS): The Environmental Control System (ECS), Air Handling Unit (AHU) and Heat Exchanger (HX). ECS contains the terminals for monitoring and controlling the other environmental systems; the AHU contains the air handling unit and the Heat Exchanger for providing air conditioning throughout the station. A Distribution Board (DB) room houses the electrical power systems for this section of the station.

Several major requirements for the above works are modelled as follows:

R1. The scaffolds are erected first to support the ceiling works installation involved in the subsequent activities. These activities involve the bracketry, ducting, cabling activities in that order. The equipment installation and permanent door installation activities are to be done sequentially after the scaffold have been dismantled. The above information is summarised in Figure 8.12. In Figure 8.12, the numerals below the activity names indicate the activity suffix designated to that particular type of activity, which is appended to the location of the activity. Hence, a scaffold erection activity in the heat exchanger room would have the following activity code: HX2. The corresponding PDM++ constraint for R1 is shown in Equation 8.9. Each activity is handled individually by a subcontractor and his specialist crew.

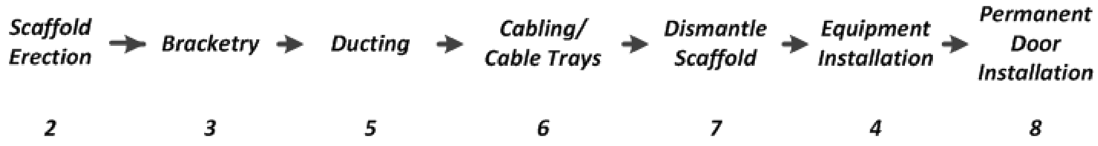


Figure 8.12. Sequence of Work

R1: DB2 Before DB3 ∧ DB3 Before DB5 ∧ DB5 Before DB6 ∧ DB6 Before DB7 ∧ DB7 Before DB4 ∧ DB4 Before DB8 **(8.9)**

R2. The sequence of work in the Main Service Corridor follows that in Figure 8.12. Additionally, work on the adjoining rooms may only start after the scaffold has been removed from the Main Service Corridor. This is to facilitate logistic movement of equipment and labour. Moreover, the scaffold is to be reused for the later scaffold activities in the adjoining rooms to the corridor. The scaffold activities in the adjoining rooms are represented using the following meta-interval {Scaffold: (HX2, AHU2, ECS2)}. The order of the rooms in this meta-interval is immaterial, and the sequence of work may be represented using Equation 8.10 as shown:

R2: Cor2 Before Cor7 ∧ Cor7 Before DB2 ∧ DB2 Before Scaffold.SF ∧ Scaffold.SF Before ECS2 **(8.10)**

R3. The crews are limited in number, and may occupy only one room at a time. The Distribution Board (DB) Room is to be done first to provide the power cables needed for the rest of the rooms. The ventilation rooms consisting of the Air Handling Unit (AHU), Heat Exchanger (HX) Room and the Environmental Control Station (ECS) may be done in any order. However, the AHU has a contractual handing-over milestone on Day 65. The Transmission Services (TX) Room has the latest handing-over date to the client. One example of the room sequence for Permanent Door Installation is shown below in Equation 8.11. Room sequences for other activities may be similarly inferred.

R3: (DB8 Before Door.SF ∧ Door.SF Before ECS8) ∧ (HX8 Disjoint AHU8 ∧ AHU8 Disjoint ECS8 ∧ ECS8 Disjoint HX8) **(8.11)**

The AHU handing-over milestone may be represented using the following:

R3a: (AHU8 Due-Before 65) **(8.12)**

R4. The ventilation shaft (VS) provides the main access into the underground facility. A crane is sited at the mouth of the shaft to lower the heavy equipment into the facility. However, this is to be partially closed off after Day 45 due to interfering works at the surface of the station, affecting the equipment installation activities in the respective rooms. Under such circumstances, the equipment is to be dismantled and lowered piece by piece into the shaft, before assembling on-site. The impact of this is that the equipment installation activities are lengthened, with higher incidences of temporal and spatial utilization. For logistic purposes, the equipment is to commence transportation into the worksite one day through the ventilation shaft before the actual installation specified in the construction plan. This requirement R4 may be expressed using the following PDM++ relationship:

R4: [(X4 Due-Before 45) → (X4.m1 after Scaffold_Remove.SF) ∧ (X4.m1 before Door.SF)] ∨ [¬ (X4 Due-Before 45) → (X4.m2 after Scaffold_Remove.SF) ∧ (X4.m2 before Door.SF)] **(8.13)**

where $X \in \{ECS, AHU, HX\}$, and *m1*, *m2* refer to the execution modes of the equipment installation activities, and will be elaborated in detail in the next section.

8.4.1 Temporal Sequencing Strategies for Mitigating Congestion

8.4.1.1 Description of Modes of Construction Sequences

Based on the above requirements, a PDM++ model is constructed to show the various requirements in Figure 8.13. The model comprises of 44 activities with the equipment installation activities having different modes. In total, there are 88 different space entities (one path space and one activity workspace per activity). Additional information involving the properties of the space entities (duration, utilization factors and existence) may be found in Appendix A.3.

From the initial analysis, four feasible alternative sequences of work are found which fulfil the requirements stated above. Each of these feasible alternative sequences is defined as a mode, and depicts a temporal strategy in which the proposed genetic algorithm schedules the activities to minimize clashing via the temporal overlapping of activities.

A float value equivalent to about 10% of the longest early start makespan of all the modes was added to the early start makespan to obtain the planning horizon. For comparison purposes, the longest planning horizon was used for all the modes. The modes with the longest early start schedule were evaluated as that of Mode 1 and Mode 2 as seen in the following Table 8.3. Mode 5 will not be discussed until the next section, but is included here for completeness.

Table 8.3. Temporal Sequencing Mode Properties

	Mode 1	*Mode 2*	*Mode 3*	*Mode 4*	*Mode 5*
Early Start Makespan	72	72	68	68	74
Float (10% of Makespan)	7	7	7	7	7
Planning Horizon (Makespan + Float)	79	79	75	75	81

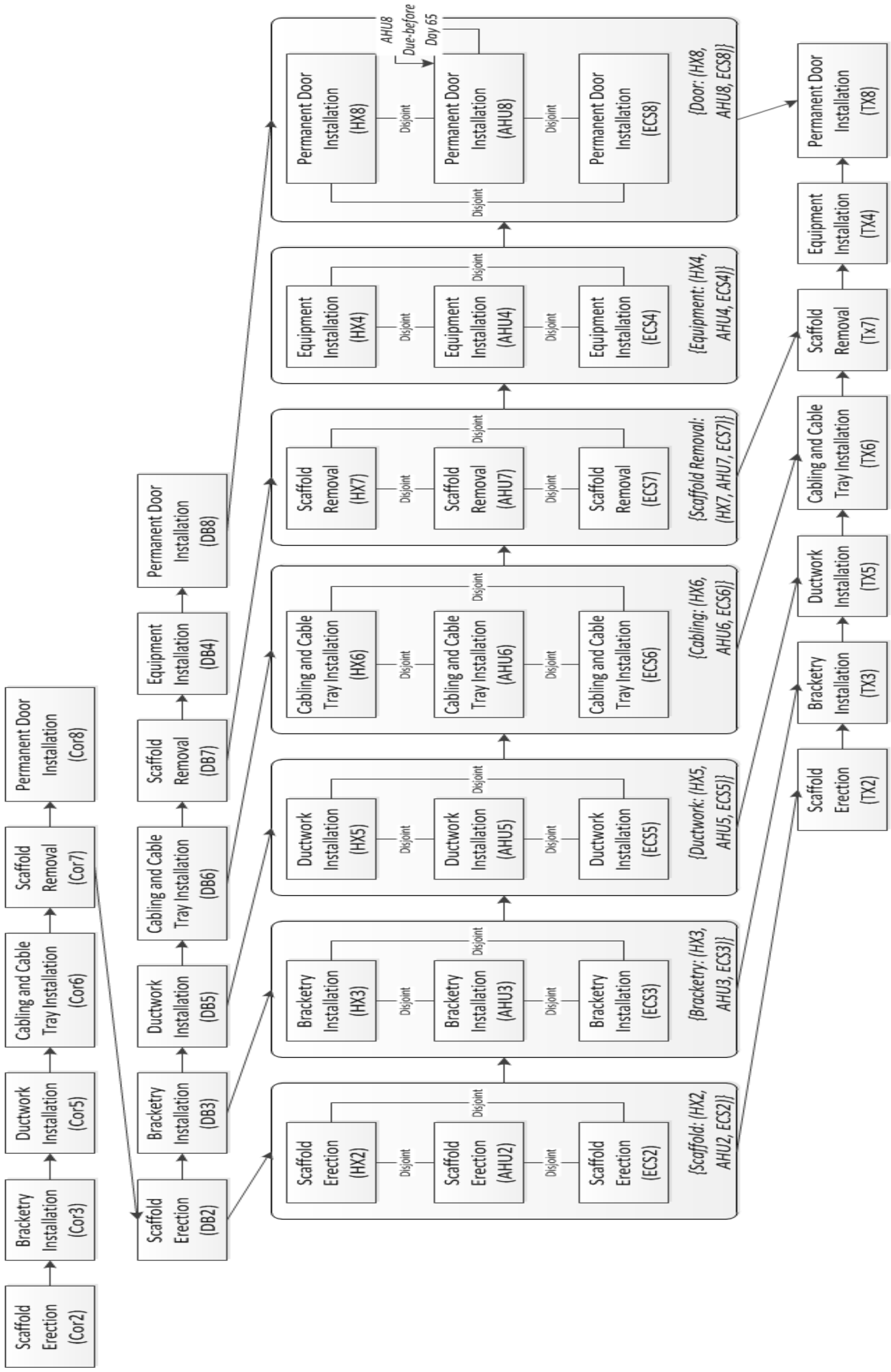


Figure 8.13. PDM++ Network of HVAC Installation Case Study

The four temporal strategies are illustrated as follows (from Figure 8.14 to Figure 8.17), with the sequence of the rooms following the order detailed in each mode. These strategies arise from the following construction sequences:

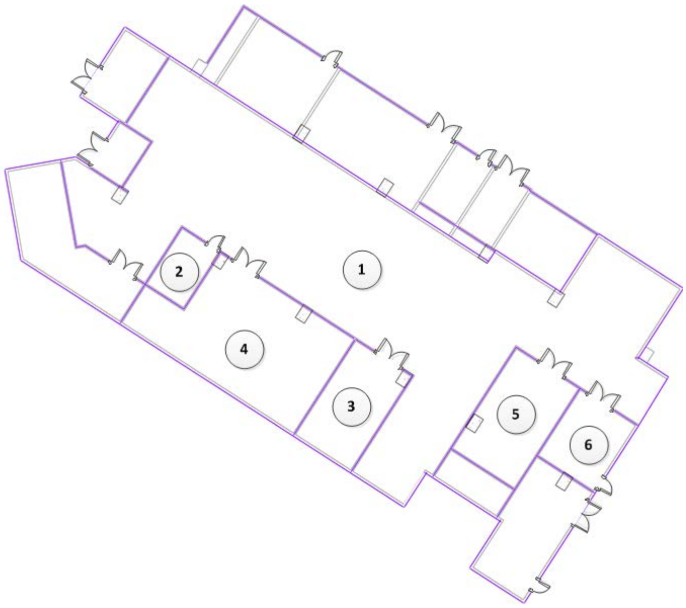


Figure 8.14. Mode 1: (1)Cor → (2)DB → (3)HX → (4)AHU → (5)ECS → (6)TX

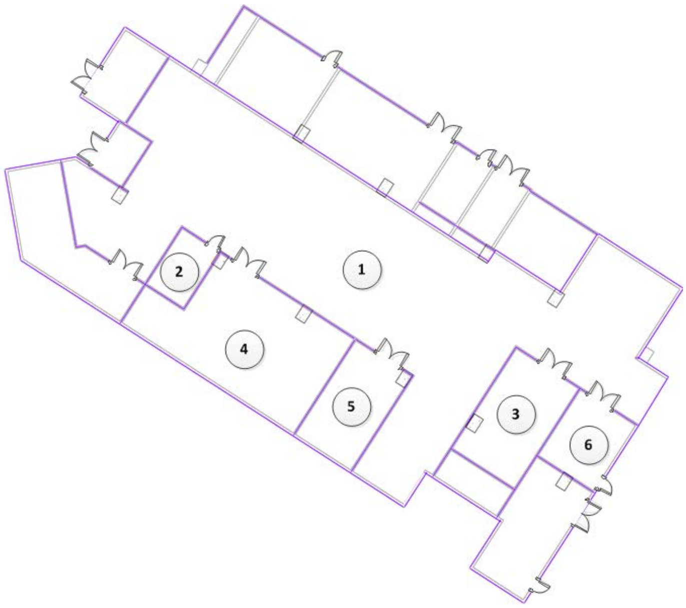


Figure 8.15. Mode 2: (1)Cor → (2)DB → (3)ECS → (4)AHU → (5)HX → (6)TX

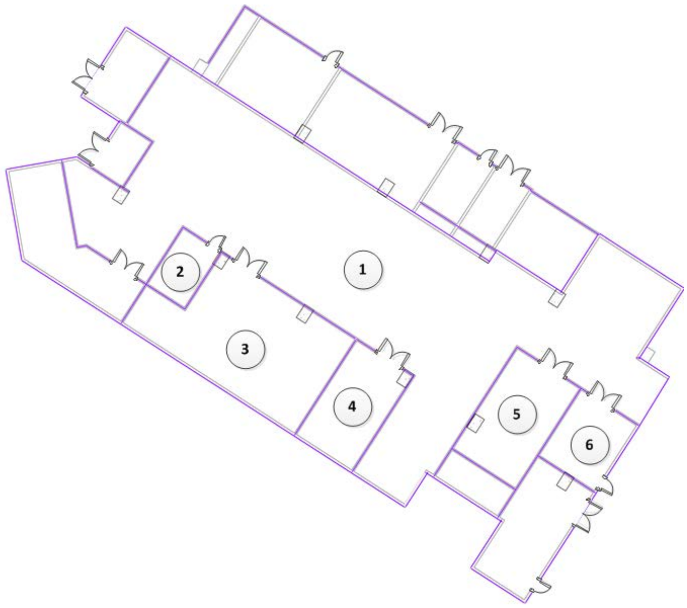


Figure 8.16. Mode 3: (1)Cor → (2)DB → (3)AHU → (4)HX → (5)ECS → (6)TX

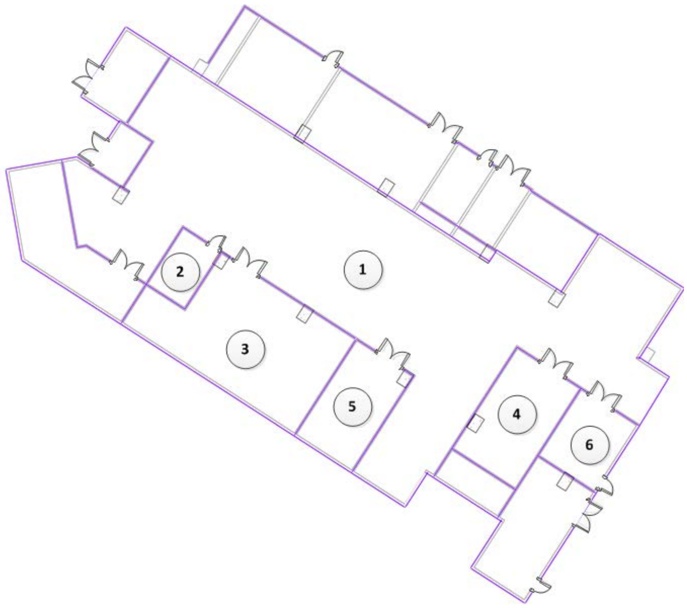


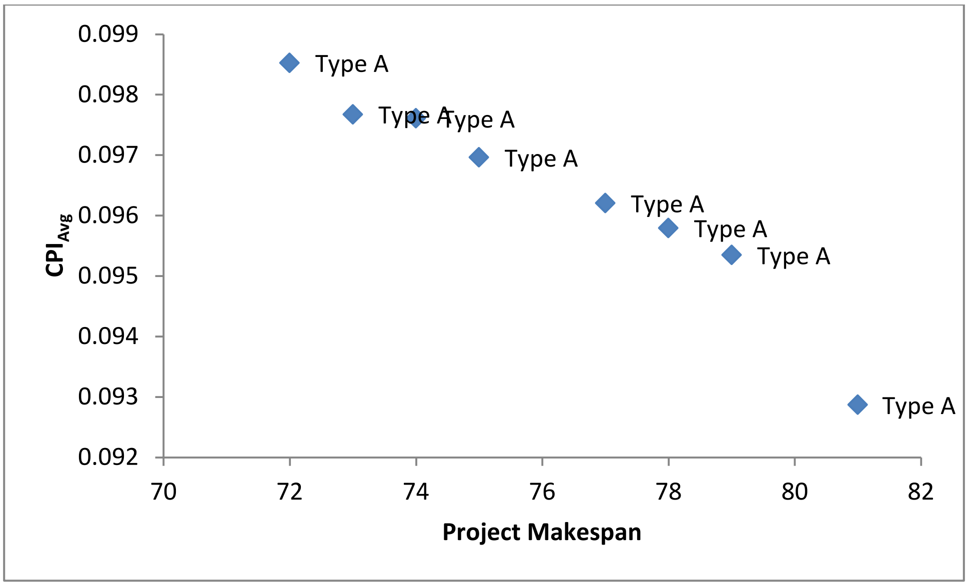
Figure 8.17. Mode 4: (1)Cor → (2)DB → (3)AHU → (4)ECS → (5)HX → (6)TX

8.4.1.2 Results of GA

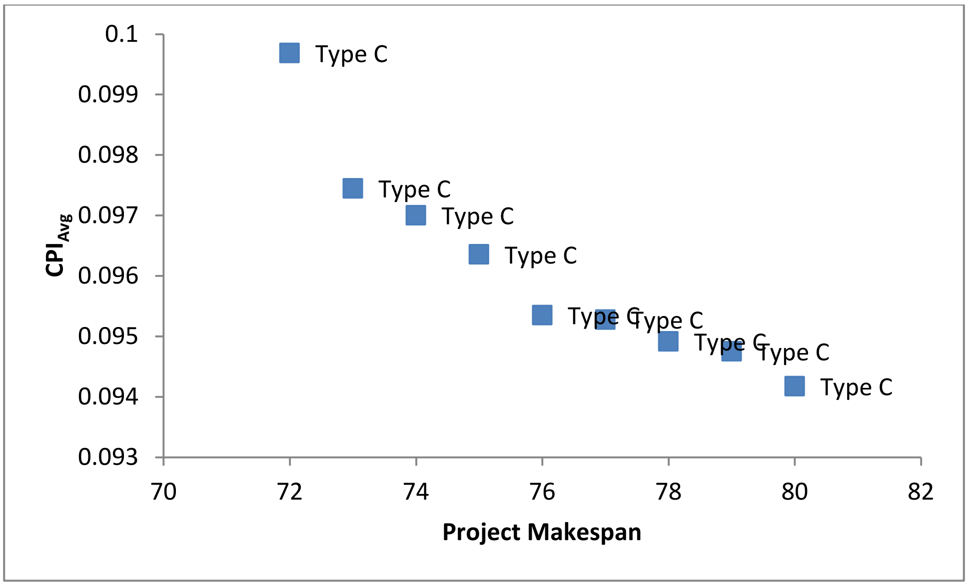
The results of the genetic algorithm on the four modes are presented in the following figure (Figure 8.18), which shows the non-dominated solutions comprising the approximated Pareto front, within the fitness landscape of the problem after 200 generations using a population size of 100 for each mode.

Each non-dominated solution in the figure has a label denoting its configuration type. This configuration type can be referenced from the table at the bottom of the figure, and is denoted as Type A, Type B etc. Each type is a classification of the list of equipment installation activity execution modes[5], where *m1* denotes the original shorter activity execution mode, while *m2* denotes the longer activity execution mode resulting from the closure of the ventilation shaft stated in Requirement R4. For example, a Type C configuration means that HX4 follows the longer activity execution mode *m2*, while AHU4 and ECS4 follow the shorter execution mode *m1*.

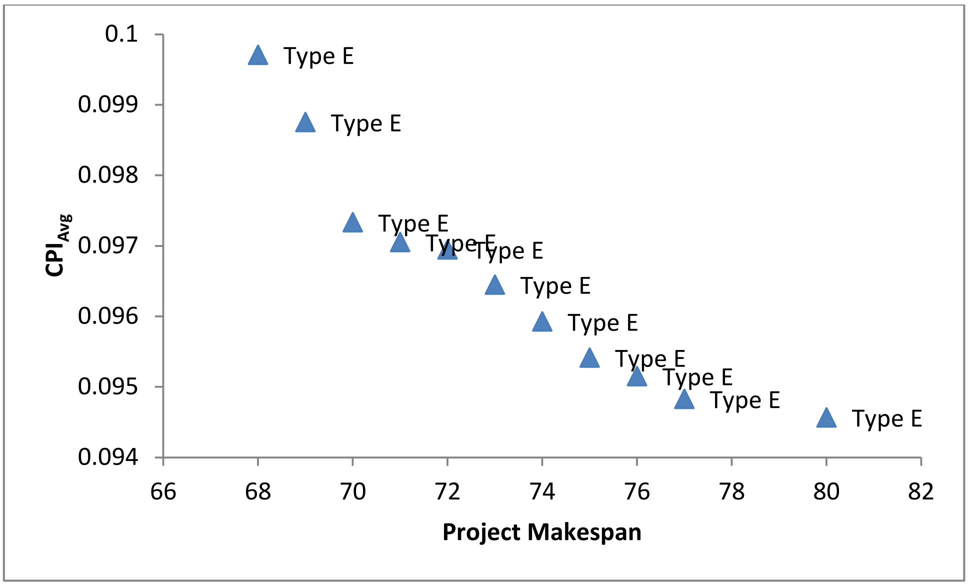
[5] The term "Execution Mode" will be used specifically to denote activity modes; this is to maintain consistency with the terminology used in related literature. The "Mode" in this chapter refers to the generalised case denoting a feasible sequence of work.



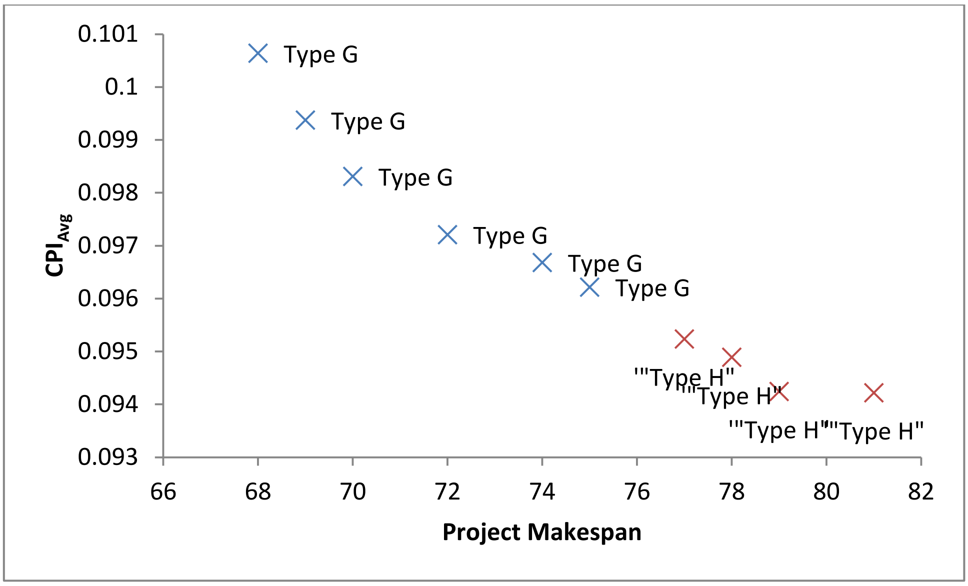
(a) Mode 1 Results



(b) Mode 2 Results



(c) Mode 3 Results



(d) Mode 4 Results

	Type A	Type B	Type C	Type D	Type E	Type F	Type G	Type H
HX4	*m1*	*m1*	*m2*	*m2*	*m1*	*m2*	*m2*	*m2*
AHU4	*m1*	*m2*	*m1*	*m2*	*m1*	*m1*	*m1*	*m1*
ECS4	*m2*	*m2*	*m1*	*m1*	*m2*	*m2*	*m1*	*m2*

Table of Active Equipment Activities

Figure 8.18. Results of GA for different Modes

Combining the results by superimposing the solutions within the different modes onto one another, gives the following resultant figure (Figure 8.19).

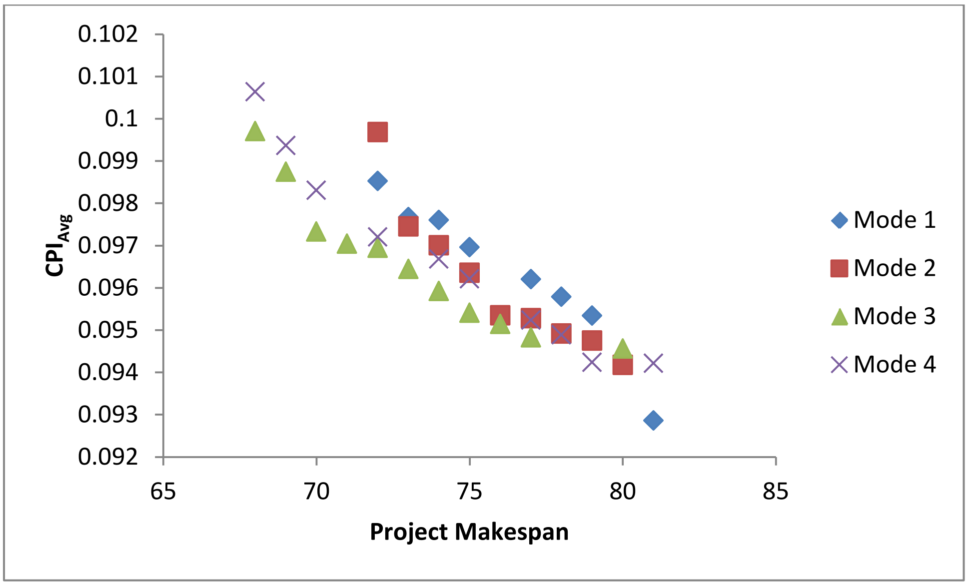


Figure 8.19. Superimposed Results for GA (Modes 1 to 4)

As a confirmation of the applicability of the proposed algorithm, a combined single run of the algorithm across all four modes with a population size of 200 over 400 generations was conducted (Figure 8.20). The Pareto front of the combined single run agrees closely with the superimposed results, with the modes corresponding to the superimposed solutions italicized while the modes corresponding to the combined single run solutions are denoted in bold as shown in the figure.

The difference in the CPI_{Avg} values of the non-dominated front in the superimposed and the combined single-run case differ by a maximum of 1.11%, and the Pareto front of the solutions obtained from the combined single-run outperforms that obtained from the superimposed results. Also, in both the combined single run and the superimposed results, Mode 3 is the dominating sequence of work for the earlier project makespans from 68 to 77. However, the combined single run and the superimposed results differ on the dominating mode after Day 77.

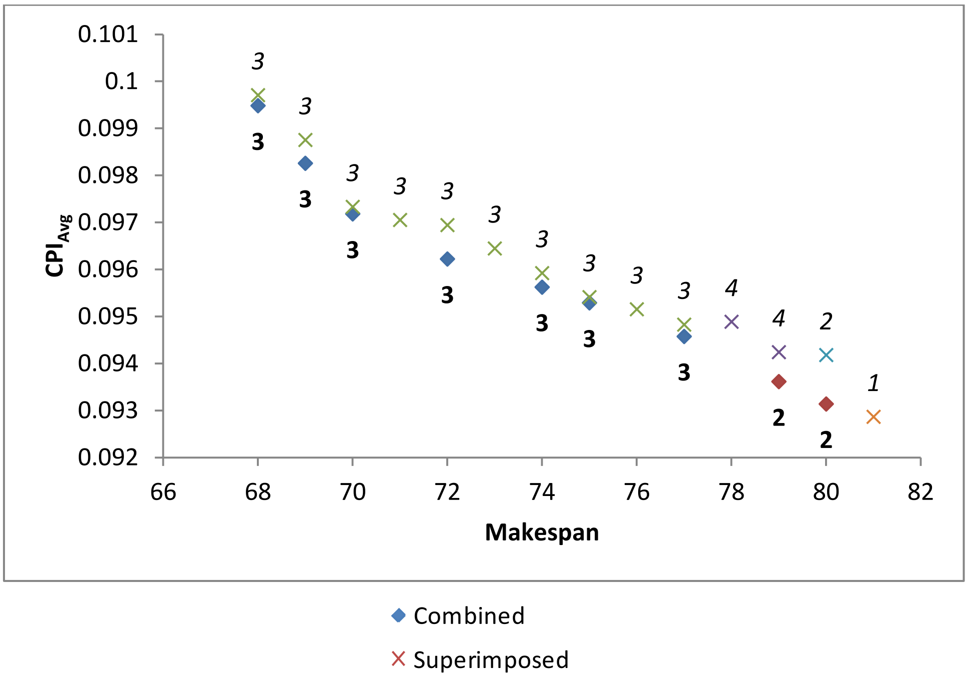


Figure 8.20. Comparison of Combined Single-Run versus Superimposed Results

8.4.1.3 Understanding the Congestion Values

CPI_{Avg} gives a single fitness value for managing worksite congestion; however it does not allow the Planner to have a sense of the entities which are subject to congested conditions. For a better analysis of the factors leading to congestion, the DSI of the workspace and pathspace entities from the Pareto front are presented in the table as follows:

Table 8.4. Comparison of DSI values for Non-Dominated Schedules on Day 68 and Day 80 for Modes 1 to 4

Space Entity	Makespan	Day 68	Day 80	Difference in DSI
	CPI Value	0.099176	0.093139855	
Scaffold Erection		0.301954418	0.301954418	0
Bracketry Installation		0.301954418	0.301954418	0
Ductwork Installation		0.325395687	0.325395687	0
Cable and Tray Installation		0.325395687	0.325395687	0
Scaffold Removal		0.301954418	0.301954418	0
Permanent Door Installation VS		0.495694166	0.419523539	0.076171
Scaffold Erection DB		0.352766841	0.352766841	0
PS Scaffold Erection DB		0.100176882	0.100176882	0
Bracketry Installation DB		0.352766841	0.352766841	0
PS Bracketry Installation DB		0.308589248	0.180769077	***0.12782***
Ductwork Installation DB		0.352766841	0.352766841	0
PS Ductwork Installation DB		0.319674543	0.318302423	0.001372
Cable and Tray Installation DB		0.352766841	0.352766841	0
PS Cable and Tray Installation DB		0.444574414	0.30606308	***0.138511***
Scaffold Removal DB		0.352766841	0.352766841	0
PS Scaffold Removal DB		0.404830369	0.308589248	0.096241
Equipment Installation DB		0.410960934	0.410960934	0
PS Equipment Installation DB		0.424667864	0.229734977	***0.194933***
Permanent Door Installation DB		0.473286383	0.473286383	0
PS Permanent Door Installation DB		0.331210655	0.299248784	0.031962
Scaffold Erection HX		0.373001923	0.373001923	0
PS Scaffold Erection HX		0.232010757	0.264362678	-0.03235
Bracketry Installation HX		0.373001923	0.373001923	0
PS Bracketry Installation HX		0.332083738	0.207650877	***0.124433***
Ductwork Installation HX		0.417028828	0.417028828	0
PS Ductwork Installation HX		0.277385522	0.279762906	-0.00238
Cable and Tray Installation HX		0.373001923	0.373001923	0
PS Cable and Tray Installation HX		0.238585882	0.199104587	0.039481
Scaffold Removal HX		0.373001923	0.373001923	0
PS Scaffold Removal HX		0.377059515	0.457352698	-0.08029
Equipment Installation HX		0.475486021	0.475486021	0
PS Equipment Installation HX		0.494336009	0.276931716	***0.217404***
Permanent Door Installation HX		0.419523539	0.419523539	0
PS Permanent Door Installation HX		0.187430235	0.128601688	0.058829
Scaffold Erection AHU		0.346410162	0.346410162	0
PS Scaffold Erection AHU		0.305071089	0.334231739	-0.02916
Bracketry Installation AHU		0.346410162	0.346410162	0
PS Bracketry Installation AHU		0.394898989	0.374376303	0.020523
Ductwork Installation AHU		0.391578004	0.391578004	0
PS Ductwork Installation AHU		0.401482323	0.352536915	0.048945
Cable and Tray Installation AHU		0.346410162	0.346410162	0
PS Cable and Tray Installation AHU		0.358398246	0.353207732	0.005191
Scaffold Removal AHU		0.346410162	0.346410162	0
PS Scaffold Removal AHU		0.371542327	0.333535078	0.038007
Equipment Installation AHU		0.43204938	0.43204938	0
PS Equipment Installation AHU		0.495029362	0.326322429	***0.168707***
Permanent Door Installation AHU		0.419523539	0.419523539	0
PS Permanent Door Installation AHU		0.303866687	0.276450684	0.027416
Scaffold Erection ECS		0.347497794	0.347497794	0
PS Scaffold Erection ECS		0.250576731	0.14357242	0.107004
Bracketry Installation ECS		0.347497794	0.347497794	0
PS Bracketry Installation ECS		0.28896754	0.212274741	0.076693
Ductwork Installation ECS		0.388514345	0.388514345	0
PS Ductwork Installation ECS		0.32362132	0.212272431	***0.111349***
Cable and Tray Installation ECS		0.388514345	0.388514345	0
PS Cable and Tray Installation ECS		0.26236378	0.233399528	0.028964
Scaffold Removal ECS		0.347497794	0.347497794	0
PS Scaffold Removal ECS		0.405505729	0.254358468	***0.151147***
Equipment Installation ECS		0.442974508	0.442974508	0
PS Equipment Installation ECS		0.281032302	0.284181356	-0.00315
Permanent Door Installation ECS		0.419523539	0.419523539	0
PS Permanent Door Installation ECS		0.14296903	0.237981004	-0.09501
Scaffold Erection TX		0.387298335	0.387298335	0
PS Scaffold Erection TX		0.23400012	0.142585495	0.091415
Bracketry Installation TX		0.447213595	0.447213595	0

Space Entity	Makespan	Day 68	Day 80	Difference in DSI
	CPI Value	0.099176	0.093139855	
PS Bracketry Installation TX		0.336758018	0.31446533	0.022293
Ductwork Installation TX		0.447213595	0.447213595	0
PS Ductwork Installation TX		0.27657991	0.201984435	0.074595
Cable and Tray Installation TX		0.447213595	0.447213595	0
PS Cable and Tray Installation TX		0.272431468	0.22315058	0.049281
Scaffold Removal TX		0.387298335	0.387298335	0
PS Scaffold Removal TX		0.218238436	0.284281357	-0.06604
Equipment Installation TX		0.447213595	0.447213595	0
PS Equipment Installation TX		0.209891827	0.169678421	0.040213
Permanent Door Installation TX		0.419523539	0.419523539	0
PS Permanent Door Installation TX		0.094129028	0.180609218	-0.08648

From Table 8.4, 40 entities had no change to their DSI value despite the increment of the project makespan. However, 27 entities registered improvements in DSI (reduced congestion) while 8 entities had their DSI values increased indicating increased congestion. The significant improvements (above 10%) are italicized. In general, a maximum improvement of 0.217 on one of the path space entities was observed, while a 0.09 increase in congestion was found on another entity. From the perspective of space-time-volume, this maximum improvement corresponds to a 21.7% increase in spatial-temporal flexibility.

8.4.1.4 Discussion of Temporal Re-sequencing Strategies

From the results, Mode 3 dominates the solutions from the other Modes. Mode 3 represents the traditional method of working the rooms in tandem starting from the DB room nearest the Vent Shaft and along the corridor to the other rooms (Refer to Figure 8.16 for illustration of the sequence). The other modes represent construction sequences where the order of the rooms is out of sync, but with the AHU constructed before Day 60.

The dominance of Mode 3 over the other modes in the earlier makespans of the model agrees with workspace strategies suggested by Thomas, *et al.* (2006) and

Thomas and Horman (2006), where the availability of float to move noncritical activities away from critical activities is an effective way of reducing site congestion.

To better understand the results suggested by the algorithm, the *DSI* equation is recapped as shown below to aid in the following discussion:

$$DSI_A = \rho_A + \sum_i \left(\rho_i \cdot \frac{S_{iA}}{S_A} \cdot \frac{t_{iA}}{t_A} \right) \qquad \forall i \in \text{Interfering Space Entities} \tag{4.4}$$

In general, the *DSI* equation is influenced by three separate values: The amount of spatial overlap $S_{i,A}/S_A$, the amount of temporal overlap $t_{i,A}/t_A$ and its own utilization factor ρ_A.

In this case study, the range of makespans is divided into three windows: The first window contains the range of makespans from Day 68 to Day 72. The second window contains the range of makespans from Day 72 to Day 77, and the final window from Day 77 to Day 81.

In the first window, there are only 2 modes which exist during this period: Mode 3 and Mode 4. This is because this window exists before the earliest possible schedules of Modes 1 and 2. Here, Mode 3 is seen to dominate Mode 4. From Figure 8.16 and Figure 8.17, Mode 3 differs from Mode 4 by the sequence of the HX and the ECS room. In Mode 3, HX starts earlier and ECS later, while in Mode 4, this order is reversed. More importantly, for each sequence the room which starts later is impacted by the closure of the ventilation shaft, which causes the equipment installation activity associated with the room to have a longer duration. This longer duration means that the potential for temporal overlaps with other activities is greater.

From the data given in Appendix A.3, the equipment installation activity at the heat exchanger, HX4 has a higher utilization factor (0.4755) as compared to its counterpart in ECS4 (0.4430), due to the heat exchange equipment in HX4 being larger and hence exhibiting a higher spatial-temporal demand than the equipment in ECS4. From the results, it is surmised that this difference in the utilization factor (coupled with the potential for greater temporal overlap due to the increment in duration) is the main contributing factor to distinguishing Mode 3 and Mode 4. From another perspective, this means that it is better to prevent the higher utilization activities from delaying, as it may potentially overlap temporally with other activities. In practice, this implies that it is better to transport the larger equipment first rather than have the equipment parts assembled onsite at the risk of causing congestion with other activities.

In the second window, all four modes are now available. Mode 2 dominates Mode 1 for the same reasons surmised above. However, both modes are dominated by Mode 3. The reason for this domination is that Mode 3 has more days of float to rearrange the activities in its schedule. This rearrangement of activities will allow the temporal overlaps to reduce, subsequently reducing the *DSI* computation. This in turn reduces the CPI_{Avg} value. The implication of this is that if a sufficiently long planning horizon were available, the schedule pressure on the activities decreases, allowing the activities to distribute themselves such that the temporal overlaps are minimized, leading to reduced congestion.

In the final window, Mode 3 is dominated by other modes at longer makespans. This may imply that adopting an “out-of-sequence” approach may be the better solution strategy to minimizing on-site congestion in this case study if a sufficiently long planning horizon were permissible, and the utilization factors of the activities are not marginally different. It can be observed that Mode 3 represents the traditional “in tandem” sequence, where the rooms are completed in topological order. An “out-of-sequence” strategy means that the rooms are not constructed in a topological sequence; rather, the construction should occur at sites as far away from each other as is technologically feasible. It should be noted the rooms are technologically independent, i.e. precedent constraints or other technological constraints due to construction requirements are not present. This also means that work in one room does not affect the work in another. The impact of this strategy lies in reasoning about the logistical path spaces due to the temporal sequencing of the activities in the rooms. This idea is illustrated in the following Figure 8.21.

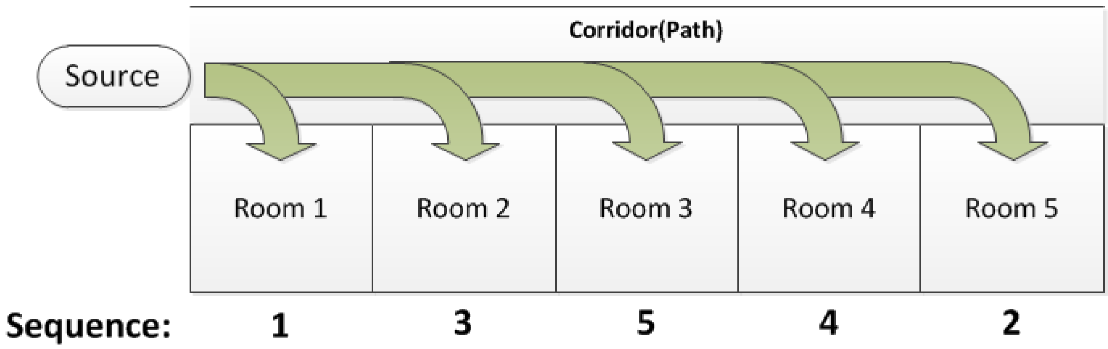


Figure 8.21. Construction Sequence as a Temporal Strategy

A possible reason for this is that the pathspaces modelled for further activities are larger. The proposed indicators are dependent on the amount of space interference between the entities, and this interference is in turn, a function of the modelled pathspace. Such a strategy may be possible from a space economics perspective; it means that the space demand is spread out, with lower potential interference faced by operators on the pathspace. This works out to greater flexibility by the operator to 'work around' any potential congestion faced by other interfering pathspaces and activities. Consequently, congestion on the work site may be minimized

The "out-of-sequence" strategy assumes several things: Other factors like cost are not included in the consideration of the model. Working out-of-sequence may bring additional transportation cost to the construction works. The additional inconvenience may also result in lower productivity and consequently longer activity durations. Also, the strategy is possible only if the rooms are technologically independent, which means that work continuity between adjacent rooms is not necessary. Work continuity implies that a certain directional flow of work is needed for construction. Some configurations of ducting work may require such continuity, which can be handled by specifying the sequence as a requirement within the proposed PDM++ modelling framework.

Another observation of the results is that the better solutions favour the activity execution modes which have lesser impact on the congestion of other activities (for this case study, the shorter duration execution modes are seen to have the lesser impact due to the lower potential for temporal overlapping between the activities). In this problem, the equipment installation activities had the additional requirement of finishing before the sealing of the ventilation shaft; otherwise, the equipment would have to be dismantled and assembled on-site at the expense of the activity bearing a higher duration. It was found that in this particular case study, the genetic algorithm generally tend to favour solutions with early equipment delivery dates, allowing the equipment to be delivered pre-assembled, and hence reducing the duration. This consequently allowed the temporal overlapping between the activities to be minimized, reducing the on-site congestion.

8.4.2 Spatial Re-sequencing Strategies for Mitigating Congestion

A spatial re-sequencing strategy was also explored by the Planner. One proposal was to provide an additional access route by leaving an opening between the ventilation shaft and the AHU room. Another opening was left between the AHU and HX rooms as shown in the following Figure 8.22. This allowed two different access routes: The new access route served the AHU and HX rooms, while the original access route along the main service corridor continued to serve the DB, ECS and TX rooms. The feasible construction sequence identified closely resembles the sequence in Mode 1, and is shown in Figure 8.23.

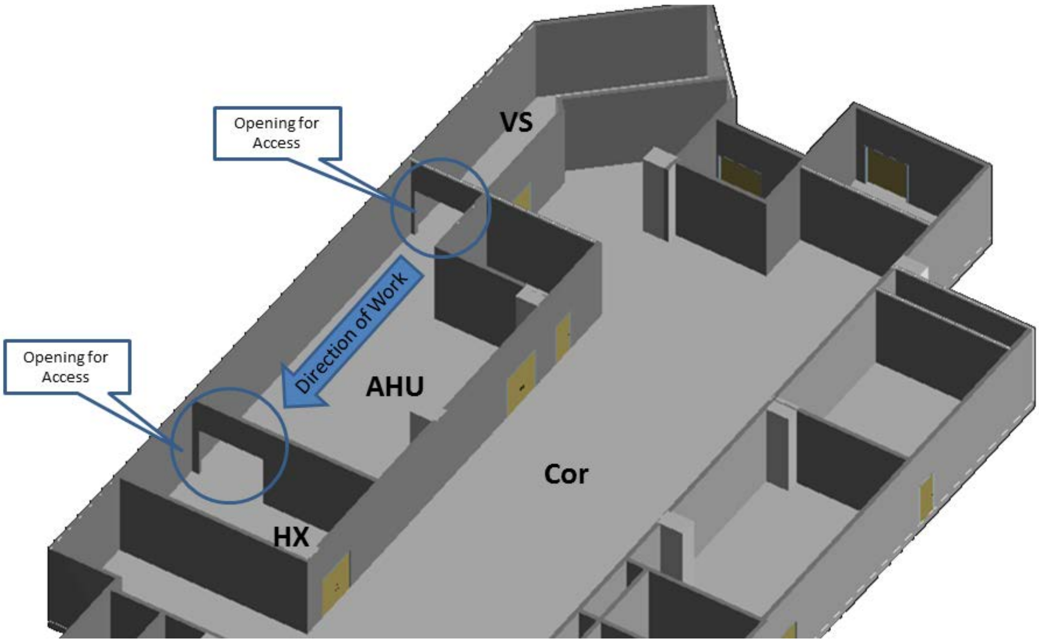


Figure 8.22. Provision of Additional Access to VS

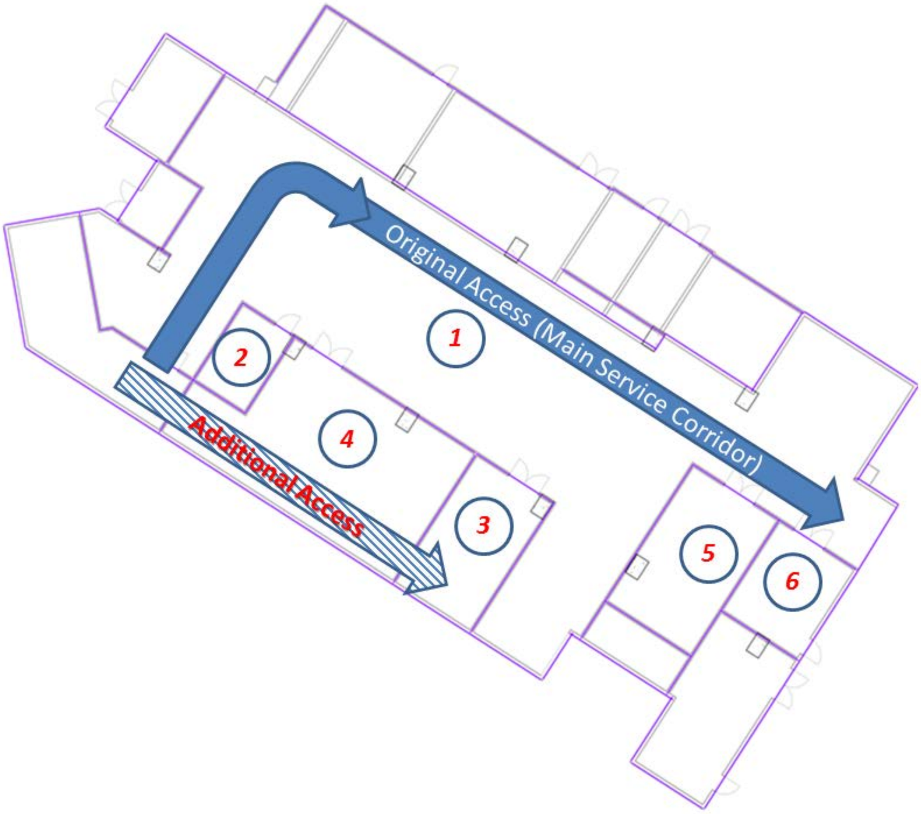


Figure 8.23. Mode 5 depicting Additional Access Routes

In addition to the original activities in Figure 8.13, two optional activities ("Sealing of Access for HX", "Sealing of Access for AHU") become mandatory in this new mode of construction sequence. Details of the two additional activities may be found in Appendix A3 under the column Mode 5. The amendments to the PDM++ network are also shown in the following Figure 8.24, where the additional optional activities are shaded. The consequence of the addition of the two activities is an increased early start makespan of 74 days as seen in Table 8.3.

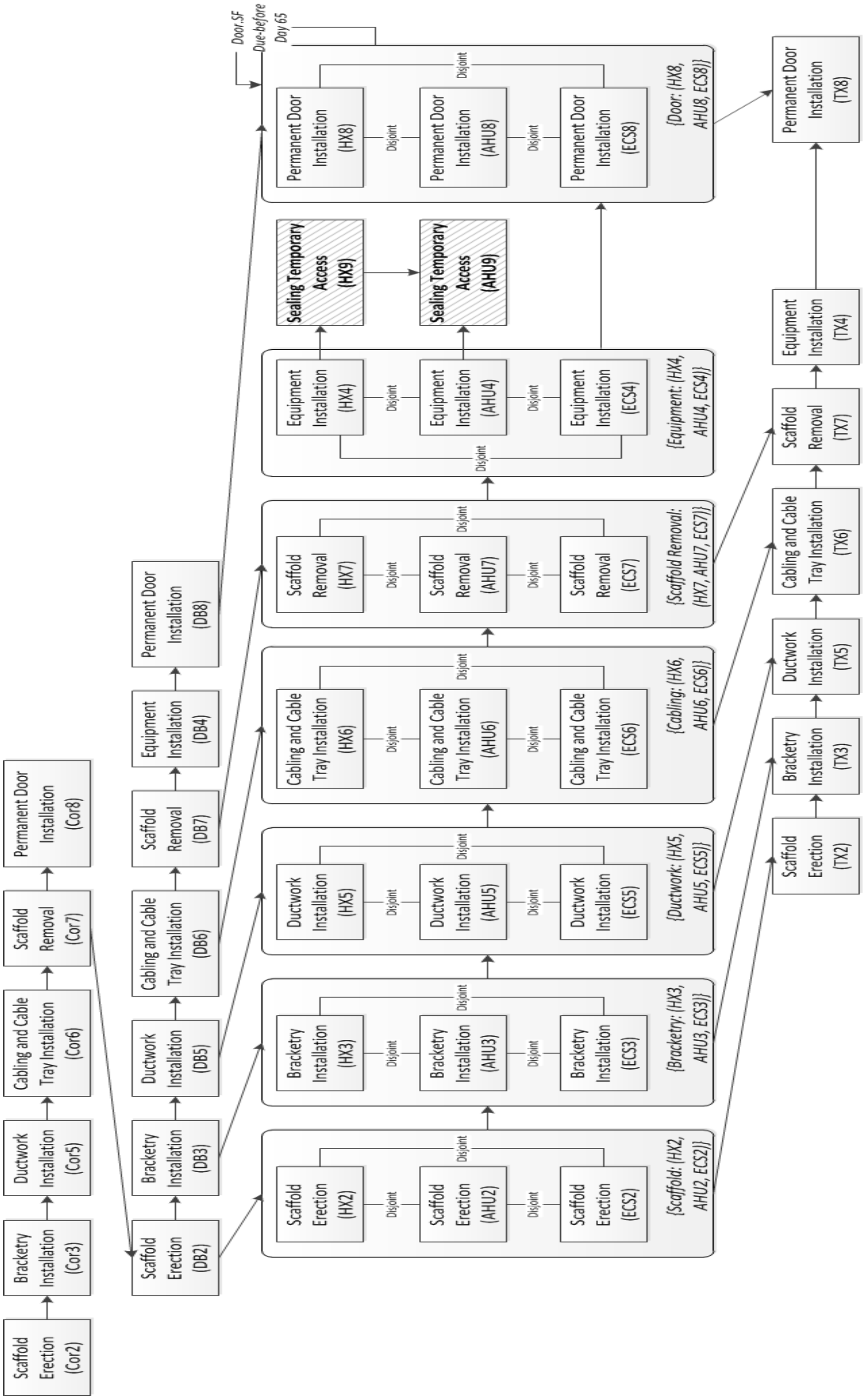


Figure 8.24. Amended PDM++ Network with Optional Mode 5 Activities

8.4.2.1 Results of GA

The result of the genetic algorithm for solving the mode corresponding to the space scheduling strategy is shown in the following Figure 8.25. Mode 5 follows a similar project sequence to Mode 1, and the Type A configuration of equipment installation activity modes governs the schedule, which means that the shorter AHU4 and HX4 equipment installation activities are chosen. Hence both activities are to be completed before the closure of the Vent Shaft.

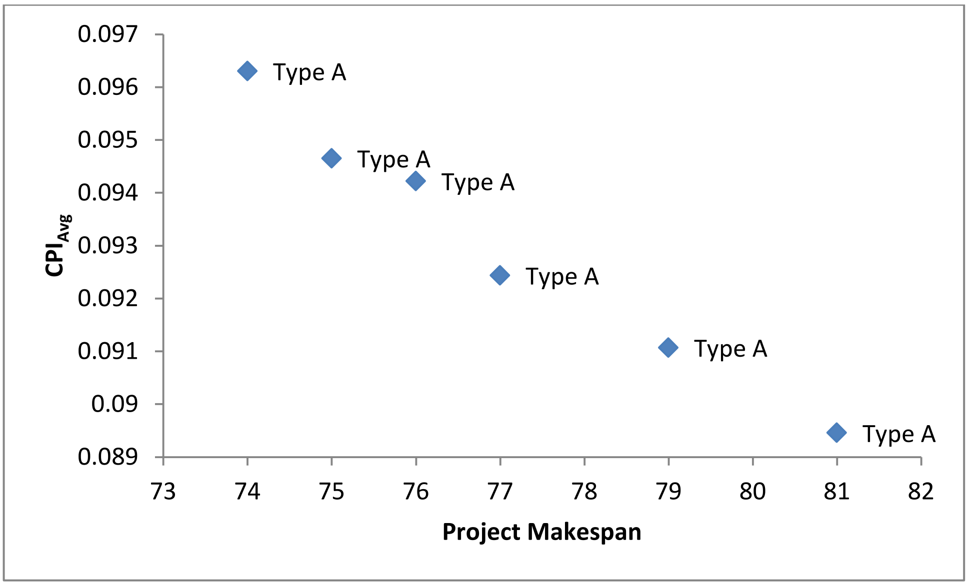


Figure 8.25. Results of GA for Mode 5

The result of the space scheduling strategy is compared against the temporal strategies discussed in the preceding section by superimposing both results to obtain the new Pareto front from the combined solutions (Figure 8.26).

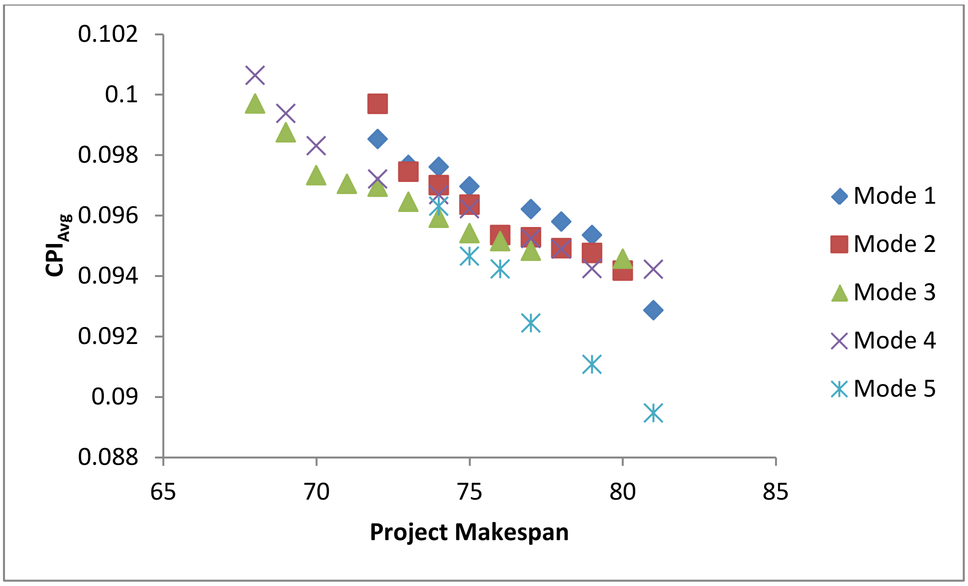


Figure 8.26. Superimposed Results for GA (including Mode 5)

A confirmatory single-run of the algorithm based on a population size of 200 over 200 generations is carried out to validate the results obtained in Figure 8.26. This is presented in the following Figure 8.27.

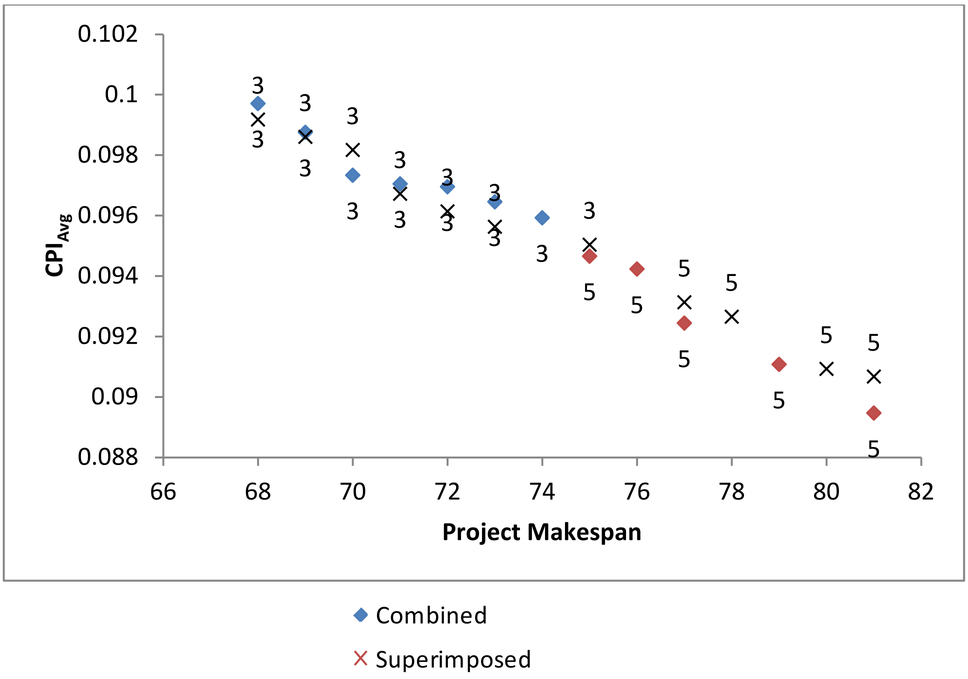


Figure 8.27. Comparison of Combined Single-Run versus Superimposed Results including Consideration for Mode 5

8.4.2.2 Understanding the Congestion Values

The following Table 8.5 shows the comparison between the DSI values of the non-dominated schedules on Day 80 for Mode 5 and Mode 2. Mode 2 was chosen as the basis for comparison as it is the "best" value found by the genetic algorithm during the temporal sequencing analysis. The entities displaying significant differences (more than 10%) in DSI values are italicized in the table.

Table 8.5. Comparison of DSI values for Non-Dominated Schedules on Day 80 for Mode 5 and Mode 2

Space Entity	**Mode**	**Mode 5**	**Mode 2**	**Difference in DSI**
	CPI Value	0.09092	0.093139855	
Scaffold Erection		0.301954	0.301954	0
Brackertry Installation		0.301954	0.301954	0
Ductwork Installation		0.325396	0.325396	0
Cable and Tray Installation		0.325396	0.325396	0
Scaffold Removal		0.301954	0.301954	0
Permanent Door Installation VS		0.419524	0.419524	0

Space Entity	Mode	Mode 5	Mode 2	Difference in DSI
	CPI Value	0.09092	0.093139855	
Scaffold Erection DB		0.352767	0.352767	0
PS Scaffold Erection DB		0.100177	0.100177	0
Bracketry Installation DB		0.352767	0.352767	0
PS Bracketry Installation DB		0.100177	0.180769	-0.08059
Ductwork Installation DB		0.352767	0.352767	0
PS Ductwork Installation DB		0.100177	0.318302	-0.21813
Cable and Tray Installation DB		0.352767	0.352767	0
PS Cable and Tray Installation DB		0.140473	0.306063	-0.16559
Scaffold Removal DB		0.352767	0.352767	0
PS Scaffold Removal DB		0.100177	0.308589	-0.20841
Equipment Installation DB		0.410961	0.410961	0
PS Equipment Installation DB		0.211464	0.229735	-0.01827
Permanent Door Installation DB		0.473286	0.473286	0
PS Permanent Door Installation DB		0.218796	0.299249	-0.08045
Scaffold Erection HX		0.373002	0.373002	0
PS Scaffold Erection HX		0.196382	0.264363	-0.06798
Bracketry Installation HX		0.373002	0.373002	0
PS Bracketry Installation HX		0.61815	0.207651	0.410499
Ductwork Installation HX		0.417029	0.417029	0
PS Ductwork Installation HX		0.390417	0.279763	0.110654
Cable and Tray Installation HX		0.373002	0.373002	0
PS Cable and Tray Installation HX		0.467529	0.199105	0.268424
Scaffold Removal HX		0.373002	0.373002	0
PS Scaffold Removal HX		0.27174	0.457353	-0.18561
Equipment Installation HX		0.475486	0.475486	0
PS Equipment Installation HX		0.346402	0.276932	0.06947
Permanent Door Installation HX		0.419524	0.419524	0
PS Permanent Door Installation HX		0.31593	0.128602	0.187329
Sealing of Temporary Access HX (optional activity)		0.245607	0	0.245607
PS Sealing of Temporary Access HX (optional activity)		0.238077	0	0.238077
Scaffold Erection AHU		0.412373	0.34641	0.065963
PS Scaffold Erection AHU		0	0.334232	-0.33423
Bracketry Installation AHU		0.401722	0.34641	0.055312
PS Bracketry Installation AHU		0	0.374376	-0.37438
Ductwork Installation AHU		0.416314	0.391578	0.024736
PS Ductwork Installation AHU		0	0.352537	-0.35254
Cable and Tray Installation AHU		0.34641	0.34641	0
PS Cable and Tray Installation AHU		0	0.353208	-0.35321
Scaffold Removal AHU		0.34641	0.34641	0
PS Scaffold Removal AHU		0	0.333535	-0.33354
Equipment Installation AHU		0.432049	0.432049	0
PS Equipment Installation AHU		0	0.326322	-0.32632
Permanent Door Installation AHU		0.419524	0.419524	0
PS Permanent Door Installation AHU		0	0.276451	-0.27645
Sealing of temporary access AHU (optional activity)		0.29282	0	0.29282
PS Sealing of temporary access AHU (optional activity)		0	0	0
Scaffold Erection ECS		0.347498	0.347498	0
PS Scaffold Erection ECS		0.1278	0.143572	-0.01577
Bracketry Installation ECS		0.347498	0.347498	0
PS Bracketry Installation ECS		0.221827	0.212275	0.009552
Ductwork Installation ECS		0.388514	0.388514	0
PS Ductwork Installation ECS		0.205596	0.212272	-0.00668
Cable and Tray Installation ECS		0.388514	0.388514	0
PS Cable and Tray Installation ECS		0.241334	0.2334	0.007935
Scaffold Removal ECS		0.347498	0.347498	0
PS Scaffold Removal ECS		0.249141	0.254358	-0.00522
Equipment Installation		0.442975	0.442975	0
PS Equipment Installation		0.195593	0.284181	-0.08859
Permanent Door Installation ECS		0.419524	0.419524	0
PS Permanent Door Installation ECS		0.079216	0.237981	-0.15877
Scaffold Erection TX		0.387298	0.387298	0
PS Scaffold Erection TX		0.207222	0.142585	0.064636
Bracketry Installation TX		0.447214	0.447214	0
PS Bracketry Installation TX		0.249423	0.314465	-0.06504
Ductwork Installation TX		0.447214	0.447214	0
PS Ductwork Installation TX		0.202991	0.201984	0.001007

Space Entity	Mode	Mode 5	Mode 2	Difference in DSI
	CPI Value	0.09092	0.093139855	
Cable and Tray Installation TX		0.447214	0.447214	0
PS Cable and Tray Installation TX		0.18332	0.223151	-0.03983
Scaffold Removal TX		0.387298	0.387298	0
PS Scaffold Removal TX		0.091425	0.284281	-0.19286
Equipment Installation TX		0.447214	0.447214	0
PS Equipment Installation TX		0.190111	0.169678	0.020432
Permanent Door Installation TX		0.419524	0.419524	0
PS Permanent Door Installation TX		0.074648	0.180609	-0.10596

In general, 25 space entities were found to have improved congestion (reduced DSI values) while 15 were found to have increased DSI values. The optional activities are ignored in this analysis for the time being for a fair comparison of the change in the DSI values between entities in the two modes. The greatest increase in congestion is observed to be 0.41, but this is mitigated by the significant overall reduction of DSI values in other entities. The results give an indication to the distribution of the space demand and supply using the spatial re-sequencing strategy.

8.4.2.3 Discussion of Spatial Re-sequencing Strategies

A comparison of solutions in Figure 8.27 is again indicative of the validity of the proposed genetic algorithm. The solutions from both the combined single run and the superimposed results indicate that Mode 5 begins to dominate the solutions when the project makespan exceeds 74 days. This means that when the Mode 5 schedule is chosen, it is immediately a better solution than the temporal sequencing strategies introduced earlier (Modes 1 to 4) from a congestion perspective.

Comparing the results of the single-run and superimposed data, the single-run genetic algorithm compares competitively with a maximum difference of 1.35% in the evaluated CPI_{Avg} values.

The results of the GA suggest that spatial re-sequencing is an effective strategy for minimizing congestion on site. The spatial re-sequencing can be seen to dominate the other solutions upon its introduction as an additional activity mode. In this case study, the spatial re-sequencing strategy focused on changing the route through the main service corridor. By diverting the access routes for some activities, it is possible to minimize congestion by reducing the spatial overlapping between the activities and their corresponding logistic path spaces. Again, from the DSI formulation in Equation 4.4, spatial overlapping of entities is one of the main components for quantifying the spatial temporal demand.

8.5. Concluding Remarks

In this chapter, three case studies are presented to demonstrate how the traditional planning and scheduling framework can be augmented through the consideration of construction requirements. The first case study is used to illustrate the application of the proposed genetic algorithm for minimizing workspace congestion, by demonstrating its use on the schedule repair of a congested oil refinery tower. This case study serves as a validation that the proposed indicator for measuring and quantifying workspace utilization from Chapter 4 is usable as an objective within an optimization framework. Additionally, the case study discusses why the indicator is valid, by comparing the solution found with the initial schedule proposed by the Planner.

In the second case study, the generalised modelling of construction requirements is illustrated. The case study presents interesting characteristics of the construction requirements: Availability of alternatives arising from conditional and optional activities like the conditional provision of preventive measures; complex temporal constraints like work continuity; and useful intervals not commonly represented by activities like meta-intervals to represent useful intervals for scaffold requirements.

The PDM++ modelling framework is applied to the case study to show how complex planning considerations can be expressed through the incorporation of construction requirements into the construction plan. A comparison with the traditional PDM is carried out to show the advantages which may be achieved using the new PDM++ modelling framework proposed.

In the final case study, the proposed genetic algorithm is extended to solve a problem incorporating the aforementioned characteristics of the construction requirements, with the additional consideration of worksite congestion. An evaluation of the solutions provided by the genetic algorithm leads to several interesting conclusions:

1. Re-ordering the temporal sequence is a viable option to minimize congestion. Particularly, provisioning for a longer planning horizon increases the availability of float for rearranging activities to lower congestion. Under special cases where technical dependencies do not affect the sequence, it was found that an “out-of-sequence” strategy for planning may also be adopted to minimize congestion.
2. From the results of the case study, spatially re-locating workspaces and pathspaces in the model can significantly improve congestion during construction.

Chapter 9. Conclusion and Future Recommendations

9.1. Overview of Construction Requirements Driven Planning and Scheduling

This dissertation proposes an overarching framework to incorporate spatial and temporal attributes of Construction Requirements in construction workflow planning and scheduling. The term "Construction Requirements Driven Planning and Scheduling" is coined to emphasize the importance of early construction input in planning the construction sequence via Construction Requirements which represent the key preconditions for construction This then forms the basic hypothesis of this dissertation: Construction Planning and Scheduling should be driven by construction knowledge in the form of construction requirements. This knowledge represented in the construction requirement captures workspace interactions and sequencing rationale for constructability analysis in the form of determining constructible schedules.

The outline of the proposed framework follows the structure of this dissertation. The initial stage of the framework deals with representing construction requirements. To achieve this, this research identifies the core characteristics of a construction requirement entity: spatial, temporal and abstract perspectives, which are necessary to describe construction requirements. These entities and the interrelationships between them then enable the construction requirements taxonomy to be developed, which is intended to be domain independent and easily extendable.

The proposed overarching framework incorporates the spatial aspect of construction requirements through the proposed indicators of *DSI* and *CPI* in Chapter 4, which allow workspace congestion to be quantified. This quantification then allows workspace requirements to be modelled as non-functional requirements, where the quantified indicators can be specified as a resource for construction scheduling.

Similarly, the temporal aspect of construction requirements is modelled using a new framework called PDM++. PDM++ serves as a taxonomy for the temporal attribute interrelationships. As a modelling framework, it enhances the traditional planning techniques in CPM to allow complex and conditional constraints to be added to the problem.

Finally, the overarching framework uses a multi-objective genetic algorithm to incorporate the temporal and spatial attributes to generate a constructible schedule from the perspective of workspaces and conditional temporal constraints.

9.2. Conclusions and Research Contribution

In summary, the key contributions of this work include the formalisation of the construction requirement taxonomy based on identified core characteristics, which embody the temporal, spatial and abstract attributes, and the interactions between them. A spatial modelling and analysis framework incorporating four-dimensional computer-aided design for detecting conflict and congestion in construction workspaces is proposed. This framework is based on spatial demand and supply, and captures the utilization of space from this perspective. The temporal aspects of the construction requirements are represented using a proposed modelling framework, PDM++. This allows the capture of static and dynamic requirements, complex temporal constraints and hierarchical modelling of plans. Finally, a multi-objective genetic algorithm is used to resolve both spatial and temporal aspects of the construction requirement, with case studies demonstrating the validity of the approach.

9.2.1 Ontological Framework for Describing Construction Requirements

The ontology model for construction requirements proposed in this dissertation seeks to achieve a formal description of construction requirements. The approach used to determine this formal description starts with describing the immutable core characteristics of construction requirement entities: abstract, spatial and temporal attributes. These entities are then composed to form construction requirements, which are demonstrated to be able to model functional, non-functional and work space requirements.

The key advantage of this approach is that the construction requirement taxonomy is flexible and extendable, able to span across various domains of knowledge, thus answering the challenges stated in Section 1.3.1. While this thesis did not raise any examples to validate the ontological framework, the applicability of the ontological framework to capture spatial, temporal and measurable abstract qualities needed to define construction requirements is the key focus of this research. As an illustration on the flexibility and extendibility using these captured qualities, future domain-specific taxonomies may be built using the knowledge constructs proposed in this research. For example, specific construction methods may be recognised as being a compilation of several different types of construction requirements.

9.2.2 Quantification Method for Analysing Spatial Temporal Conflict and Congestion

An ontological description of space utilization is introduced, from which a workspace conflict taxonomy was developed which specifically distinguishes various classes of conflict, of which congestion may be the most difficult class to analyse.

Simply considering spatial overlaps of workspaces is inadequate for classifying congestion as a form of conflict.

Arising from the challenges raised in Section 1.3.2, a few key ideas are proposed. First, the concept of utilization describes the spatiotemporal supply and demand, and may be abstracted by a space-time-volume. In this way, the usual space economics has been extended to comprise a fourth dimension of time explicitly in the analysis of congestion conflicts. This extension allows the interaction of space and time to be captured.

Second, the operative-level utilization perspective is important to distinguish spatiotemporal congestion of overlapped workspaces. The operation space determines the minimum space necessary for an activity to be executed. This operative-level utilization perspective is an abstraction of the workflow in the activity, and provides the basis for the characterisation of both spatial and temporal utilizations. The above ideas lead to the development of two indicators in the proposed approach: *DSI* and *CPI*. The *DSI* indicator may be extended to allow the effect of several interacting entities to be incorporated as well.

While *DSI* measures the local workaround, *CPI* allows the schedule in a critical time window to be evaluated, analysed and compared. It is derived analogous to the Utility Theory. Similarly, it incorporates a preference trade-off to elicit a tolerance value, which determines the `disutility' of congestion to the Planner. The *CPI* indicator is then the sum of the 'disutility' values of each space entity in the critical time window. The illustrative case study in Chapter 4 demonstrates the application of the above indicators, and illustrates the impact of utilization in identifying congestion as a form of conflict.

The proposed indicators can be used to complement current constructability analysis involving 4D CAD. 4D CAD offers an effective medium to visually conceptualize construction plans, as well as visualizing changes to the construction sequence. The indicators then allow the effects of congestion to be captured in current 4D CAD models as part of the constructability analysis as well.

9.2.3 Modelling and Evaluation of Temporal Attributes of Construction Requirements

This dissertation illustrates the necessity for planning from construction requirements and provides a modelling framework for achieving it. Several challenges were identified earlier in Section 1.3.3. These challenges have been addressed through the introduction of the proposed temporal modelling framework PDM++. PDM++ overcomes the challenges through the extension of the present PDM model via the use of logical operatives and covers the complete Allen's interval relationship representation. This not only allows for greater semantic expression from the model to capture various relationships unambiguously, but also allows the logical operatives to facilitate the representation of complex conditional relationships. This further enables the framework to generalise the different alternatives in a single model through this representation.

The proposed system architecture to resolve the temporal attributes of the construction requirements is proposed as part of this dissertation. The system architecture identifies the 8 binary semantic literals and the 4 unary semantic literals as the basic building blocks for PDM++. Incorporating basic semantic literals are the key extensions applied to traditional CPM methods, in particular PDM, where the current semantic of FF, FS, SS, and SF minimal lag relationships are expanded to include corresponding relationships with maximal lags.

Further, three levels of logical operators are also introduced to enhance the description of activity relationships. Of these, the inverse operator is introduced to facilitate reasoning on constraints. The Implication and Equivalence operators are used to model the conditional constraints which arise from dynamic construction requirements. These extensions enable the proposed system to fulfil the identified system requirements in Chapter 5. The proposed system, PDM++ subsumes the traditional PDM.

The meta-interval construct is another key research contribution, and provides the mechanism to represent requirements that are contingent on specific time intervals which may span several activities. The meta-interval may be used to model requirements that depend on the states of construction products or resources, such as the scaffold in the case example. The use of meta-intervals allows for relationships to be expressed at higher levels of plan abstraction, which may sometimes be necessary for construction requirements.

9.2.4 Multi-Objective Genetic Algorithm

A multi-objective genetic algorithm based on the NSGA-II algorithm is used to determine a constructible schedule subject to various construction requirements including conditional dynamic requirements and workspace requirements.

The main novelty of the approach lies in the segregation of the temporal requirements from the workspace resource requirements. The chromosome encoding scheme reflects this segregation by defining two sets of chromosomes. The first set determines the priority of the activity, and uses an iterative bounds consistency approach (BCSolver algorithm) to maintain the temporal feasibility in the solution space. This approach is an extension of current float techniques in other GAs.

The second chromosome set then determines the start time of the activity subject to the influence of the workspace resource, and an additional chromosome is used to encode the mode of the problem, which represents one of the available and feasible sequences of work. The mutation rates and the crossover probability are also encoded to create a self-adaptive genetic algorithm, thus eliminating the need to tune the parameters of the model to suit specific problems.

The results of the genetic algorithm seem promising, with the illustrative case study validating the combined modelling framework of PDM++ under the influence of spatial congestion.

9.3. Limitations and Recommendations for Future Work

This section will address several limitations in the current work, and recommend some ideas on how to address these shortcomings in future work.

9.3.1 Limitations and Future Work for Construction Requirements

Several issues for modelling construction requirements still remain to be addressed. Firstly, the requirements gathering process in the present framework may be tedious for large projects. Plans to automatically generate construction requirements from 4D CAD models is being planned as part of the future work to automatically generate the construction requirements. Such a process may be achieved by implementing a template based approach to automatically identify and generate the construction requirement. Based on the core attributes of the entities, it is subsequently possible for the system to automatically validate the model, i.e. check the function is fulfilled by the behaviour, or if the goal is satisfied by the performance of the system. Planners then need only verify the requirements gathered.

Secondly, a hierarchy of requirements may be possible, and with the definition of this hierarchy of requirements, new inter-requirement relationships may need to be explored. With this hierarchy of requirements and its interactions, a requirements network can be abstracted. The purpose of the requirements network is to enhance the traceability of construction requirements. This is achieved through capturing early requirements as higher level requirements. The traceability of a construction requirement is the ability to describe the lifecycle of the construction requirement. Requirements traceability through identifying the interrelationships between various requirements is a useful tool for determining the impact of change when a construction requirement, being derived requirements from Client Requirements is subject to changes.

The ontological approach proposed here is presently only applicable to measurable quantities. However, some quantities are either immeasurable, or extremely ambiguous. Future approaches could explore fuzzy techniques to capture and validate the construction requirement. However, such approaches are inherently difficult due to the need to identify the appropriate contextual setting before assigning a suitable fuzzy value.

One potential research challenge for hierarchical construction requirements is in defining a suitable temporal logic which is consistent across various hierarchical levels. An example of varying temporal logic across several levels of granularity was highlighted by Hegazy and Menesi (2010), and the following example (Figure 9.1) excerpted from their work. Consider the following time based relationship between two activities A and B. At a lower hierarchical level, FS relationships may be defined between the tasks of A and B, but aggregated at a higher level, the relationship changes to an SS2 relationship between A and B.

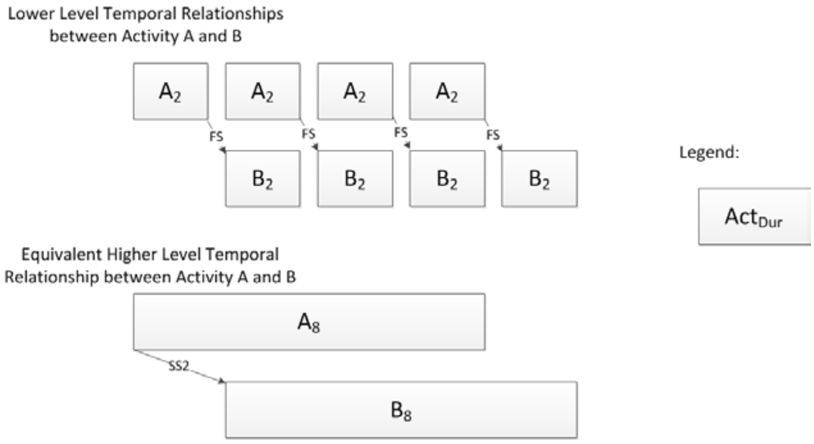


Figure 9.1. Equivalent Temporal Relationships at Different Levels of Granularity

9.3.2 Using Construction Requirements for Change Management

Construction requirements are subject to unforeseen changes, which could arise from clients through the form of variation orders or via changes in regulations. A key feature of the requirements network is that it is an information-rich representation which is readily captured, and provides a visual method to determine who is responsible and potentially impacted by changes in the requirements network. A change in a single requirement could propagate to other requirements through the requirements network.

The output of this study will provide a handle for Planners to understand how a change in the requirements has a cascading impact on other requirements. This becomes especially useful when Planners are able to make an informed decision on which requirement may have to be mitigated to address the change. Such a change could take the form of resource requirements or construction methodologies.

Other avenues of exploring how to mitigate change in the requirements network may be explored in future work. Such work could involve consideration of function redundancy, and building additional construction requirement options into the system to mitigate and manage risk.

9.3.3 Improving Spatial Models: Stochastic Representations of Space and Incorporating Productivity into Models

One basic assumption of the *DSI* and *CPI* indicators is that the operators are able to accommodate one another, achieving a "compromise" which allows them to achieve "local" work around. This "compromise" is reflected in the U_s and U_t values which "averages" the utilization in the space entity. The proposed method captures the utilization of activities without having to deal with the details of the location of the operators, while enabling the abstraction of information at the higher activity level. This implied assumption may be challenged and addressing this forms the basis for future work which could involve stochastically analysing the location of the operator and determining its effects on the degree of interference between activities.

4D CAD is a useful tool for analysing the effects of congestion on productivity (Chau, *et al.*, 2005). Despite this, the analysis methodology for relating productivity to 4D CAD is hitherto non-existent. A few difficulties are faced when attempting to relate productivity to spatial constraints. Firstly, congestion occurs due to overmanning and stacking of trades. The effects of both phenomena on productivity are not well documented. Secondly, it is often difficult to isolate productivity figures due to overmanning or stacking of trades.

Despite the difficulties involved, it could still be possible to relate spatial requirements from the 4D CAD model with productivity. The current DSI equation (given by Equation 3.5) assumes a completely linear relation with equal weightages allocated to overmanning and stacking. However, this may not be true. Assuming that the interactions of overmanning and stacking of trades is minimal and thus independent, this would give rise to a new equation of the form shown:

$$DSI_A = f(\rho_A) + \sum_i \left[f_i \left(\rho_i \cdot \frac{S_i}{S_A} \cdot \frac{t_i}{t_A} \right) \right] \quad \textbf{(9.1)}$$

From analysing the functional form of the above equation, it can be seen that $f(\rho_A)$ corresponds to the effects of overmanning on the activity. The second part of the function $f_i\left(\rho_i \cdot \frac{s_i}{s_A} \cdot \frac{t_i}{t_A}\right)$ corresponds to the effects of "Stacking of Trades", as it shows the spatial and temporal overlaps between the different trades. By assuming the independence of workspaces mentioned above, the above form removes the effect of multiple interactions of overlapping workspaces. By providing such a relationship, it would be possible to correlate data of observed productivity to spatial requirements. This would allow 4D CAD to evolve from merely a visualisation tool to an analytical tool, with practical usage by project managers.

9.3.4 Improving Temporal Models: Handling Activity Splitting, Resource Levelling and Requirement Preferences

The adoption of Construction Requirements Driven Planning could present some initial learning difficulties despite its congruency to natural language. Also, the present framework is currently incapable of handling activity splitting, and resource levelling, but this will be addressed in future work.

Additional work to transform the model from deterministic to stochastic is also being carried out. The deterministic model presented in this dissertation forms the basis for the stochastic one, which could include uncertainty in activity durations, as well as imprecise constraints. User preference for alternative construction requirements is another area of investigation. In its present incarnation, PDM++ does not explicitly evaluate preferential treatment of different constraints in construction requirements, but this could be extended in future work. Such user preference would allow Planners to specify which construction method is more attractive, and also analyse possible alternative combinations based on choice.

Increasing the expressiveness of PDM for planning has its limitation: the tradeoff between expressiveness and computational complexity. Increasing expressiveness invariably increases the computational complexity. The underlying algebraic structure of the constraints can be identified as belonging to the class of Disjunctive Temporal Problem (DTP) which is known to be NP-complete (Stergiou and Koubarakis, 1998). However, efficient solution techniques are known to exist (Tsamardinos and Pollack, 2003), and further investigation of the application of these techniques to resolving more complicated instances of PDM++ will be carried out.

9.3.5 Improving Requirements Analysis: Identifying Redundant Requirements, Constraints and Quantifying Requirements Flexibility

From a practical perspective, increasing the expressiveness of PDM does not remove the onus of ensuring and checking that construction plans are error free from the Planner's scope of work. On the contrary, PDM++ requires that the Planner is familiar with logical concepts, so that the construction knowledge and requirements within the plan are well represented within the PDM++ context. However the rewards of articulating the plan early in the construction process will greatly outweigh the effort and increased learning curve on the part of the Planner.

Also, the application of the PDM++ framework for project control, delay analysis and risk management is the subject of future work, where new techniques and perspectives will be introduced. These techniques will allow for tracking the fulfilment of construction requirements. Methods for handling the generation of too many feasible alternative schedules, which ironically makes planners "spoilt for choice".

One idea of enhancing the analysis of requirements is to redefine the criticality of the requirement through the quantification of flexibility of the requirement. The concept of Requirement Flexibility is introduced as a measure of the possible combinations of start dates which satisfy the requirement. This measure is analogous to the concept of total float of activities in critical activities. The distinction of the requirement flexibility is that where the concept of float applies to activities, requirement flexibility applies to constraints and requirements. The flexibility of constraints can be described as a measure of the allowable permutations of the constraint based on the corresponding domain values, i.e. how adaptive the constraint is to perturbations in the network. For unary constraints, the constraint flexibility, S_C is equivalent to the activity's total float, and can be calculated from the following equation

$$S_C = \sum_{i \in D_X} \left(C_{x_i} \right) \qquad \textbf{(9.2)}$$

where D_x is the domain of the activity x, and C_x refers to the instantiation of the constraint when activity x assumes the value of i. However, the flexibility for more complex constraints and requirements is not so easily addressed.

The concept of the temporal requirement network is a proposed technique to show an interesting duality between activities and their requirements, which allows a generalisation of the impact of alternatives across the activities and constraints to be captured. Based on this generalisation, a better flexibility measure of the constraint may be proposed as a form of duality to the criticality of the activity.

Despite this difficulty, the advantages of flexibility on the schedule are obvious. Firstly, evaluating flexibility allows the Planner to decide if sufficient temporal flexibility is available to reduce the impact of uncertainty on the schedule. Additionally, the comparison of alternative sequences implied by the construction requirement can be meaningfully evaluated. Finally, the impact of relaxing a requirement may also be quantified.

9.3.6 Investigating effect of α on DCR-ST

The congestion tolerance, α requires more study to understand its impact on the DCR-ST problem. Through the computational experiments carried out, it was found that CPI values obtained may not be sensitive enough to small improvements to DSI. This is because the CPI values scale the lower valued DSI values much lesser than the higher DSI values. While this is reasonable, it implies that α may have some impact on the fitness landscape of the problem. More investigation is required to determine reasonable value for α.

9.3.7 Using DPLL to enhance DCR-ST

Presently, the modes (feasible sequences of work) of the DCR-ST problem are being generated naively. Extensions to this include the adaptation of the DPLL(T) algorithmic framework, which is largely based on the Davis-Putnam-Logemann-Loveland (DPLL) Procedure (Davis, *et al.*, 1962). Where the DPLL algorithm was originally formulated for Boolean propositional satisfiability problems (SAT), the DPLL(T) algorithm extends DPLL with domain independent back end solvers (Nieuwenhuis, *et al.*, 2006), allowing it to move beyond SAT. The validity of such an approach has been justified by Coelho and Vanhoucke (2011).

As stated previously, the DCR-ST problem may be decomposed into two sub-problems: Mode assignment problem and Project scheduling problem. The mode assignment problem is especially NP-hard. This framework would provide a more efficient way to generate the available modes for a DCR-ST problem.

Appendix

A.1. Discussion on selection of weights *a* and *b* for *ρ*

Let *a* and *b* be normalised weights in Equation 4.3 as shown:

$$\rho = \sqrt[a+b]{U_s^a \times U_t^b} \tag{4.3}$$

$$\rho = U_S^{a/a+b} \times U_t^{b/a+b} \tag{A.1}$$

Let $v = a/a+b$ and $1 - v = b/a+b$, Equation A.1 becomes

$$\rho = U_S^v \times U_t^{1-v} \tag{A.2}$$

Taking natural logarithm on both sides and differentiating implicitly,

$$\ln\rho = \ln(U_S^v \times U_t^{1-v})$$

$$\ln\rho = v\ln U_s + (1-v)\ln U_t$$

$$\ln\rho = v\ln U_s + (1-v)\ln U_t$$

$$\frac{\partial}{\partial v}\ln\rho = \frac{\partial}{\partial v}v\ln U_s + \frac{\partial}{\partial v}(1-v)\ln U_t$$

$$\frac{\rho'}{\rho} = \ln U_s - \ln U_t$$

$$\rho' = U_s^v \times U_t^{1-v} \times (\ln U_s - \ln U_t) \tag{A.3}$$

Equation A.3 gives the rate of change of ρ with respect to the normalised weight *v*, and this allows us to evaluate the amount of change expected on the utilization factor when a small change occurs in the weight *v*. A numerical study of the equation is conducted (Table A.1), with the spatial utilization $U_s \in (0, 1]$, temporal utilization $U_t \in (0, 1]$, and normalised weight $v \in (0, 1]$ varied within its allowable bounds, and the results presented below. From an engineering perspective, small values of U_s, U_t and v are restricted to a value of 0.01 as such small ratios are unlikely to occur, and their occurrence may signal to the Planner that the model is too coarse, and may need to be refined for better usability.

Table A.1. Sensitivity Study of Utilization Factor

$v = 0.01$

0.01	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00792	-0.0225	-0.05815	-0.09862	-0.14221	-0.1881	-0.23581	-0.28502	-0.33553	-0.38716	-0.43979
0.05	0.001636	0	-0.00688	-0.02734	-0.0528	-0.08147	-0.11251	-0.14544	-0.17992	-0.21574	-0.25272	-0.29073
0.1	0.002356	0.00349	0	-0.01377	-0.0326	-0.05469	-0.07919	-0.1056	-0.13359	-0.16293	-0.19345	-0.22502
0.2	0.003087	0.007028	0.00698	0	-0.01211	-0.02753	-0.0454	-0.0652	-0.0866	-0.10938	-0.13335	-0.15837
0.3	0.003519	0.009121	0.011107	0.008142	0	-0.01147	-0.02541	-0.0413	-0.05881	-0.0777	-0.09779	-0.11896
0.4	0.003827	0.010616	0.014056	0.013959	0.008655	0	-0.01113	-0.02423	-0.03895	-0.05507	-0.07239	-0.09079
0.5	0.004068	0.011781	0.016356	0.018495	0.015403	0.008946	0	-0.01092	-0.02347	-0.03742	-0.05259	-0.06884
0.6	0.004265	0.012737	0.018242	0.022215	0.020939	0.016284	0.009133	0	-0.01077	-0.02295	-0.03634	-0.05082
0.7	0.004433	0.013548	0.019841	0.025371	0.025635	0.02251	0.01688	0.009263	0	-0.01067	-0.02256	-0.03554
0.8	0.004578	0.014253	0.021231	0.028113	0.029715	0.027919	0.023611	0.017311	0.00936	0	-0.01059	-0.02226
0.9	0.004707	0.014876	0.02246	0.030537	0.033322	0.032701	0.029563	0.024427	0.017636	0.009434	0	-0.01052
1	0.004822	0.015434	0.023562	0.032711	0.036557	0.036989	0.034898	0.030807	0.025056	0.017891	0.009492	0

$v = 0.05$

0.05	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00742	-0.02052	-0.05158	-0.08608	-0.1227	-0.16085	-0.20018	-0.24048	-0.28159	-0.32339	-0.3658
0.05	0.001744	0	-0.0067	-0.02587	-0.04915	-0.07496	-0.10261	-0.13167	-0.1619	-0.19309	-0.22513	-0.2579
0.1	0.002584	0.003588	0	-0.01339	-0.0312	-0.05174	-0.07425	-0.09829	-0.12359	-0.14993	-0.17718	-0.20522
0.2	0.00348	0.007429	0.007176	0	-0.01192	-0.02678	-0.04376	-0.06239	-0.08237	-0.10348	-0.12556	-0.1485
0.3	0.004032	0.009798	0.011606	0.008275	0	-0.01134	-0.0249	-0.04017	-0.05685	-0.07471	-0.09359	-0.11336
0.4	0.004436	0.011536	0.014858	0.014352	0.008756	0	-0.01103	-0.02384	-0.03809	-0.05356	-0.07008	-0.08753
0.5	0.004757	0.012918	0.017443	0.019185	0.015721	0.009026	0	-0.01084	-0.02316	-0.03673	-0.05137	-0.06695
0.6	0.005024	0.014068	0.019597	0.023213	0.021528	0.016551	0.0092	0	-0.01071	-0.02269	-0.03576	-0.04979
0.7	0.005254	0.015057	0.021448	0.026675	0.026519	0.02302	0.017109	0.009321	0	-0.01061	-0.02234	-0.03504
0.8	0.005455	0.015924	0.023073	0.029716	0.030904	0.028704	0.024059	0.017511	0.00941	0	-0.01054	-0.02207
0.9	0.005635	0.016699	0.024524	0.032431	0.034819	0.033779	0.030266	0.024826	0.017814	0.009478	0	-0.01048
1	0.005798	0.017399	0.025835	0.034886	0.03836	0.03837	0.03588	0.031442	0.025416	0.018052	0.009533	0

$v = 0.1$

0.1	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00685	-0.01829	-0.0444	-0.07262	-0.10203	-0.13227	-0.16312	-0.19446	-0.22618	-0.25823	-0.29057
0.05	0.00189	0	-0.00647	-0.02414	-0.04494	-0.06756	-0.09145	-0.11629	-0.14188	-0.1681	-0.19484	-0.22202
0.1	0.002899	0.003714	0	-0.01293	-0.02953	-0.04827	-0.06851	-0.08987	-0.11213	-0.13512	-0.15874	-0.1829
0.2	0.004042	0.007962	0.007429	0	-0.01168	-0.02587	-0.0418	-0.05906	-0.07737	-0.09655	-0.11646	-0.13702
0.3	0.004779	0.010717	0.012262	0.008445	0	-0.01118	-0.02427	-0.0388	-0.05449	-0.07114	-0.08859	-0.10674
0.4	0.005335	0.0128	0.015924	0.014858	0.008882	0	-0.01091	-0.02336	-0.03704	-0.05174	-0.0673	-0.08361
0.5	0.005785	0.014494	0.018905	0.020084	0.016128	0.009127	0	-0.01074	-0.02277	-0.03587	-0.04988	-0.06467
0.6	0.006166	0.015929	0.021434	0.024524	0.022287	0.01689	0.009284	0	-0.01063	-0.02236	-0.03504	-0.04854
0.7	0.006497	0.01718	0.023639	0.028399	0.027667	0.023673	0.017399	0.009393	0	-0.01054	-0.02206	-0.03442
0.8	0.006792	0.018292	0.025601	0.031849	0.032457	0.029716	0.024631	0.017765	0.009473	0	-0.01048	-0.02182
0.9	0.007057	0.019295	0.027372	0.034964	0.036786	0.035177	0.031169	0.025335	0.01804	0.009534	0	-0.01043
1	0.007299	0.02021	0.028988	0.03781	0.04074	0.040169	0.037145	0.032256	0.025874	0.018254	0.009583	0

$v = 0.2$

0.2	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00583	-0.01453	-0.03291	-0.05168	-0.07056	-0.08945	-0.10832	-0.12715	-0.14593	-0.16466	-0.18334
0.05	0.002221	0	-0.00603	-0.02101	-0.03756	-0.05488	-0.07264	-0.0907	-0.10897	-0.12739	-0.14593	-0.16455
0.1	0.003649	0.003981	0	-0.01207	-0.02646	-0.04202	-0.05832	-0.07513	-0.0923	-0.10975	-0.12743	-0.14528
0.2	0.005454	0.009146	0.007962	0	-0.01122	-0.02414	-0.03814	-0.05291	-0.06826	-0.08405	-0.1002	-0.11665
0.3	0.006715	0.01282	0.013686	0.008794	0	-0.01086	-0.02306	-0.03621	-0.05007	-0.06449	-0.07937	-0.09463
0.4	0.007714	0.015759	0.018292	0.015924	0.009142	0	-0.01067	-0.02243	-0.03503	-0.04827	-0.06206	-0.07629
0.5	0.008555	0.018247	0.022206	0.022012	0.016973	0.009333	0	-0.01055	-0.02202	-0.03423	-0.04703	-0.06034
0.6	0.009286	0.020423	0.02564	0.027372	0.023887	0.017589	0.009455	0	-0.01046	-0.02173	-0.03365	-0.04612
0.7	0.009937	0.022369	0.028717	0.032189	0.030113	0.025036	0.017995	0.009539	0	-0.0104	-0.02151	-0.03321
0.8	0.010527	0.024137	0.031518	0.036585	0.035802	0.031849	0.025816	0.018283	0.0096	0	-0.01035	-0.02134
0.9	0.011067	0.025762	0.034098	0.040639	0.041057	0.038149	0.033056	0.026383	0.018499	0.009647	0	-0.01032
1	0.011568	0.02727	0.036494	0.044412	0.045953	0.044023	0.039811	0.033946	0.026813	0.018666	0.009684	0

$v = 0.3$

0.3	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00497	-0.01154	-0.02439	-0.03678	-0.04879	-0.06049	-0.07193	-0.08314	-0.09415	-0.10499	-0.11568
0.05	0.002608	0	-0.00563	-0.01829	-0.0314	-0.04457	-0.0577	-0.07075	-0.0837	-0.09655	-0.1093	-0.12195
0.1	0.004594	0.004267	0	-0.01126	-0.0237	-0.03658	-0.04965	-0.0628	-0.07598	-0.08915	-0.10229	-0.1154
0.2	0.007359	0.010506	0.008534	0	-0.01077	-0.02252	-0.0348	-0.04741	-0.06022	-0.07317	-0.08621	-0.09931
0.3	0.009436	0.015335	0.015275	0.009158	0	-0.01056	-0.02191	-0.03378	-0.046	-0.05846	-0.07111	-0.0839
0.4	0.011156	0.019402	0.021012	0.017067	0.009408	0	-0.01043	-0.02154	-0.03312	-0.04504	-0.05722	-0.06961
0.5	0.01265	0.022971	0.026083	0.024124	0.017863	0.009544	0	-0.01036	-0.02129	-0.03266	-0.04435	-0.0563
0.6	0.013984	0.026184	0.030671	0.03055	0.025601	0.018316	0.009629	0	-0.0103	-0.02111	-0.03231	-0.04382
0.7	0.015197	0.029124	0.034886	0.036485	0.032776	0.026477	0.01861	0.009687	0	-0.01026	-0.02098	-0.03205
0.8	0.016316	0.031849	0.038804	0.042025	0.039492	0.034135	0.027059	0.018817	0.009729	0	-0.01023	-0.02087
0.9	0.017357	0.034396	0.042476	0.047235	0.045825	0.041371	0.035057	0.027475	0.01897	0.009762	0	-0.01021
1	0.018334	0.036794	0.045943	0.052167	0.051832	0.048248	0.042668	0.035726	0.027787	0.019087	0.009787	0

$v = 0.4$

0.4	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00423	-0.00917	-0.01808	-0.02618	-0.03374	-0.04091	-0.04776	-0.05436	-0.06075	-0.06695	-0.07299
0.05	0.003064	0	-0.00525	-0.01592	-0.02625	-0.03621	-0.04583	-0.05518	-0.06428	-0.07317	-0.08186	-0.09038
0.1	0.005784	0.004573	0	-0.01051	-0.02124	-0.03185	-0.04227	-0.0525	-0.06254	-0.07241	-0.08211	-0.09167
0.2	0.009929	0.012068	0.009146	0	-0.01034	-0.02101	-0.03176	-0.04248	-0.05313	-0.0637	-0.07417	-0.08454
0.3	0.013258	0.018345	0.017049	0.009537	0	-0.01026	-0.02082	-0.03152	-0.04226	-0.053	-0.06371	-0.07438
0.4	0.016133	0.023887	0.024137	0.018292	0.009683	0	-0.0102	-0.02069	-0.03132	-0.04202	-0.05277	-0.06351
0.5	0.018706	0.028919	0.030638	0.026439	0.018799	0.009759	0	-0.01017	-0.02059	-0.03116	-0.04182	-0.05253
0.6	0.021059	0.03357	0.036689	0.034098	0.027438	0.019074	0.009806	0	-0.01015	-0.02051	-0.03103	-0.04164
0.7	0.023242	0.03792	0.04238	0.041355	0.035674	0.028	0.019247	0.009837	0	-0.01013	-0.02046	-0.03093
0.8	0.025288	0.042025	0.047773	0.048274	0.043561	0.036585	0.028361	0.019366	0.00986	0	-0.01011	-0.02041
0.9	0.02722	0.045923	0.052914	0.054902	0.051146	0.044866	0.037179	0.028612	0.019452	0.009877	0	-0.0101
1	0.029057	0.049646	0.057838	0.061276	0.058464	0.052877	0.045731	0.037598	0.028796	0.019518	0.009891	0

$v = 0.5$

0.5	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.0036	-0.00728	-0.0134	-0.01863	-0.02333	-0.02766	-0.03171	-0.03555	-0.03919	-0.04269	-0.04605
0.05	0.003599	0	-0.0049	-0.01386	-0.02194	-0.02941	-0.03641	-0.04304	-0.04937	-0.05545	-0.06131	-0.06699
0.1	0.007281	0.004901	0	-0.0098	-0.01903	-0.02773	-0.03599	-0.04389	-0.05148	-0.05882	-0.06592	-0.07281
0.2	0.013397	0.013863	0.009803	0	-0.00993	-0.01961	-0.02898	-0.03806	-0.04687	-0.05545	-0.06381	-0.07198
0.3	0.018629	0.021944	0.019029	0.009932	0	-0.00997	-0.01978	-0.02941	-0.03883	-0.04805	-0.05709	-0.06594
0.4	0.023331	0.029408	0.027726	0.019605	0.009966	0	-0.00998	-0.01986	-0.02961	-0.03921	-0.04866	-0.05795
0.5	0.027662	0.036407	0.035988	0.028976	0.019784	0.009979	0	-0.00999	-0.01991	-0.02973	-0.03943	-0.04901
0.6	0.031715	0.04304	0.043889	0.038057	0.029408	0.019864	0.009986	0	-0.00999	-0.01993	-0.0298	-0.03957
0.7	0.035545	0.049372	0.051484	0.046874	0.038828	0.029612	0.019906	0.00999	0	-0.00999	-0.01995	-0.02984
0.8	0.039194	0.055452	0.058815	0.055452	0.048051	0.03921	0.029726	0.019931	0.009993	0	-0.00999	-0.01996
0.9	0.042689	0.061314	0.065917	0.063813	0.057086	0.048656	0.03943	0.029795	0.019947	0.009994	0	-0.01
1	0.046052	0.066987	0.072814	0.071976	0.065944	0.057951	0.049013	0.039568	0.029842	0.019959	0.009995	0

$v = 0.6$

0.6	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00306	-0.00578	-0.00993	-0.01326	-0.01613	-0.01871	-0.02106	-0.02324	-0.02529	-0.02722	-0.02906
0.05	0.004227	0	-0.00457	-0.01207	-0.01834	-0.02389	-0.02892	-0.03357	-0.03792	-0.04202	-0.04592	-0.04965
0.1	0.009167	0.005253	0	-0.00915	-0.01705	-0.02414	-0.03064	-0.03669	-0.04238	-0.04777	-0.05291	-0.05784
0.2	0.018077	0.015924	0.010506	0	-0.00954	-0.01829	-0.02644	-0.0341	-0.04135	-0.04827	-0.0549	-0.06128
0.3	0.026176	0.026251	0.021238	0.010343	0	-0.00968	-0.0188	-0.02744	-0.03567	-0.04356	-0.05115	-0.05846
0.4	0.033739	0.036205	0.031849	0.021012	0.010256	0	-0.00976	-0.01907	-0.028	-0.03658	-0.04487	-0.05288
0.5	0.040906	0.045834	0.042272	0.031756	0.020821	0.010204	0	-0.00981	-0.01925	-0.02836	-0.03718	-0.04573
0.6	0.047761	0.055181	0.052501	0.042476	0.031518	0.020686	0.01017	0	-0.00984	-0.01937	-0.02861	-0.0376
0.7	0.054362	0.064283	0.062543	0.05313	0.042261	0.031316	0.020587	0.010145	0	-0.00986	-0.01945	-0.0288
0.8	0.060748	0.073169	0.07241	0.063697	0.053002	0.042025	0.031156	0.020513	0.010127	0	-0.00988	-0.01952
0.9	0.066948	0.081863	0.082115	0.07417	0.063715	0.052766	0.041817	0.031028	0.020455	0.010113	0	-0.00989
1	0.072987	0.090384	0.091668	0.084545	0.074382	0.063512	0.052531	0.041642	0.030925	0.020409	0.010101	0

$v = 0.7$

0.7	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00261	-0.00459	-0.00736	-0.00944	-0.01116	-0.01265	-0.01398	-0.0152	-0.01632	-0.01736	-0.01833
0.05	0.004965	0	-0.00427	-0.01051	-0.01534	-0.0194	-0.02297	-0.02618	-0.02912	-0.03185	-0.0344	-0.03679
0.1	0.01154	0.00563	0	-0.00853	-0.01527	-0.02101	-0.02608	-0.03067	-0.03489	-0.0388	-0.04248	-0.04594
0.2	0.024391	0.018292	0.01126	0	-0.00916	-0.01707	-0.02412	-0.03055	-0.03649	-0.04202	-0.04724	-0.05217
0.3	0.03678	0.031402	0.023704	0.010771	0	-0.00941	-0.01786	-0.0256	-0.03278	-0.03949	-0.04582	-0.05183
0.4	0.048791	0.044574	0.036585	0.02252	0.010556	0	-0.00954	-0.01832	-0.02648	-0.03413	-0.04137	-0.04825
0.5	0.06049	0.057701	0.049654	0.034803	0.021912	0.010435	0	-0.00963	-0.01861	-0.02706	-0.03506	-0.04267
0.6	0.071927	0.070747	0.062804	0.047409	0.033781	0.021542	0.010357	0	-0.00969	-0.01882	-0.02747	-0.03573
0.7	0.083139	0.083697	0.075979	0.060221	0.045998	0.033119	0.021292	0.010303	0	-0.00973	-0.01897	-0.02779
0.8	0.094154	0.096547	0.089148	0.073169	0.058465	0.045041	0.032655	0.021112	0.010263	0	-0.00976	-0.01909
0.9	0.104994	0.109299	0.102293	0.086208	0.071113	0.057223	0.044349	0.032312	0.020976	0.010232	0	-0.00979
1	0.115677	0.121953	0.115403	0.099308	0.083898	0.069607	0.056301	0.043825	0.032048	0.020869	0.010208	0

$v = 0.8$

0.8	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00222	-0.00365	-0.00545	-0.00672	-0.00771	-0.00855	-0.00929	-0.00994	-0.01053	-0.01107	-0.01157
0.05	0.005832	0	-0.00398	-0.00915	-0.01282	-0.01576	-0.01825	-0.02042	-0.02237	-0.02414	-0.02576	-0.02727
0.1	0.014528	0.006034	0	-0.00796	-0.01369	-0.01829	-0.02221	-0.02564	-0.02872	-0.03152	-0.0341	-0.03649
0.2	0.03291	0.021012	0.012068	0	-0.00879	-0.01592	-0.02201	-0.02737	-0.03219	-0.03658	-0.04064	-0.04441
0.3	0.051681	0.037564	0.026457	0.011216	0	-0.00914	-0.01697	-0.02389	-0.03011	-0.0358	-0.04106	-0.04595
0.4	0.070557	0.054877	0.042025	0.024137	0.010864	0	-0.00933	-0.01759	-0.02504	-0.03185	-0.03815	-0.04402
0.5	0.089449	0.072642	0.058324	0.038143	0.023061	0.01067	0	-0.00945	-0.01799	-0.02582	-0.03306	-0.03981
0.6	0.108319	0.090704	0.075128	0.052914	0.036205	0.022433	0.010548	0	-0.00954	-0.01828	-0.02638	-0.03395
0.7	0.127149	0.108974	0.0923	0.068258	0.050066	0.035025	0.02202	0.010463	0	-0.0096	-0.0185	-0.02681
0.8	0.145931	0.127395	0.109754	0.084049	0.06449	0.048274	0.034227	0.021728	0.010401	0	-0.00965	-0.01867
0.9	0.16466	0.145929	0.127429	0.100201	0.079371	0.062057	0.047034	0.033649	0.02151	0.010354	0	-0.00968
1	0.183335	0.16455	0.145283	0.116649	0.094633	0.076286	0.060342	0.046121	0.033212	0.02134	0.010316	0

$v = 0.9$

0.9	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00189	-0.0029	-0.00404	-0.00478	-0.00533	-0.00578	-0.00617	-0.0065	-0.00679	-0.00706	-0.0073
0.05	0.006851	0	-0.00371	-0.00796	-0.01072	-0.0128	-0.01449	-0.01593	-0.01718	-0.01829	-0.0193	-0.02021
0.1	0.01829	0.006467	0	-0.00743	-0.01226	-0.01592	-0.0189	-0.02143	-0.02364	-0.0256	-0.02737	-0.02899
0.2	0.044405	0.024137	0.012935	0	-0.00844	-0.01486	-0.02008	-0.02452	-0.0284	-0.03185	-0.03496	-0.03781
0.3	0.072617	0.044935	0.029529	0.011681	0	-0.00888	-0.01613	-0.02229	-0.02767	-0.03246	-0.03679	-0.04074
0.4	0.102035	0.067561	0.048274	0.025869	0.011181	0	-0.00913	-0.01689	-0.02367	-0.02972	-0.03518	-0.04017
0.5	0.132274	0.09145	0.068509	0.041803	0.024269	0.010911	0	-0.00928	-0.0174	-0.02463	-0.03117	-0.03714
0.6	0.163125	0.11629	0.08987	0.059059	0.038804	0.023361	0.010742	0	-0.00939	-0.01776	-0.02533	-0.03226
0.7	0.194457	0.141884	0.112127	0.077368	0.054492	0.037041	0.022774	0.010625	0	-0.00947	-0.01804	-0.02587
0.8	0.226181	0.168098	0.135123	0.096547	0.071136	0.051738	0.035874	0.022362	0.010541	0	-0.00953	-0.01825
0.9	0.258233	0.194836	0.158742	0.116464	0.088588	0.067299	0.049881	0.035042	0.022057	0.010476	0	-0.00958
1	0.290567	0.222024	0.182901	0.137018	0.10674	0.083606	0.064673	0.048539	0.034418	0.021822	0.010426	0

$v = 1.0$

1	0.01	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.01	0	-0.00161	-0.0023	-0.003	-0.0034	-0.00369	-0.00391	-0.00409	-0.00425	-0.00438	-0.0045	-0.00461
0.05	0.008047	0	-0.00347	-0.00693	-0.00896	-0.0104	-0.01151	-0.01242	-0.0132	-0.01386	-0.01445	-0.01498
0.1	0.023026	0.006931	0	-0.00693	-0.01099	-0.01386	-0.01609	-0.01792	-0.01946	-0.02079	-0.02197	-0.02303
0.2	0.059915	0.027726	0.013863	0	-0.00811	-0.01386	-0.01833	-0.02197	-0.02506	-0.02773	-0.03008	-0.03219
0.3	0.102036	0.053753	0.032958	0.012164	0	-0.00863	-0.01532	-0.02079	-0.02542	-0.02942	-0.03296	-0.03612
0.4	0.147555	0.083178	0.055452	0.027726	0.011507	0	-0.00893	-0.01622	-0.02238	-0.02773	-0.03244	-0.03665
0.5	0.195601	0.115129	0.080472	0.045815	0.025541	0.011157	0	-0.00912	-0.01682	-0.0235	-0.02939	-0.03466
0.6	0.245661	0.149094	0.107506	0.065917	0.041589	0.024328	0.010939	0	-0.00925	-0.01726	-0.02433	-0.03065
0.7	0.297395	0.184734	0.136214	0.087693	0.059311	0.039173	0.023553	0.010791	0	-0.00935	-0.01759	-0.02497
0.8	0.350562	0.221807	0.166355	0.110904	0.078466	0.055452	0.0376	0.023015	0.010683	0	-0.00942	-0.01785
0.9	0.404983	0.260133	0.19775	0.135367	0.098875	0.072984	0.052901	0.036492	0.022618	0.0106	0	-0.00948
1	0.460517	0.299573	0.230259	0.160944	0.120397	0.091629	0.069315	0.051083	0.035667	0.022314	0.010536	0

Legend

U_s | U_t | $\Delta\rho<10\%$ | $10\%<\Delta\rho<40\%$ | $\Delta\rho>40\%$

The objective of the results is to determine the impact of overall utilization ρ for a 10% change in weight *v*. From the results, it can be concluded that the model is fairly robust under most conditions of normalised weightages and utilization factors. However, the model is sensitive to the following two combinations of factors: A high normalised weightage with high spatial utilization and low temporal utilization; and a low normalised weightage with high temporal utilization and low spatial utilization. Under these two conditions, a change of about 40% to the overall utilization factor ρ may be expected.

A.2. Proof of Correctness of the BCSolver Algorithm

This appendix shows the mathematical properties of the system which allow for the conclusion that the Bounds Consistency algorithm is sound.

It is possible to abstract the integer values in the domain of an activity X as being ordered values on a timeline as shown in the following figures. An initial assumption is made that the value d_i on the timeline is not a valid instantiation of X. However, its neighbouring values d_{i-1} and d_{i+1} are both valid instantiations. It is further assumed that the activity belongs to a binary constraint, which involves another activity Y. After some mathematical manipulation of the temporal relationship, the constraint between X and Y may be expressed in the form shown in Equation A.4 where t is a positive integer. Without loss of generality, it is assumed that Y can be treated as a single time point which is valid under the constraint. The following two cases are now considered.

Case 1: $Y - X \leq t$

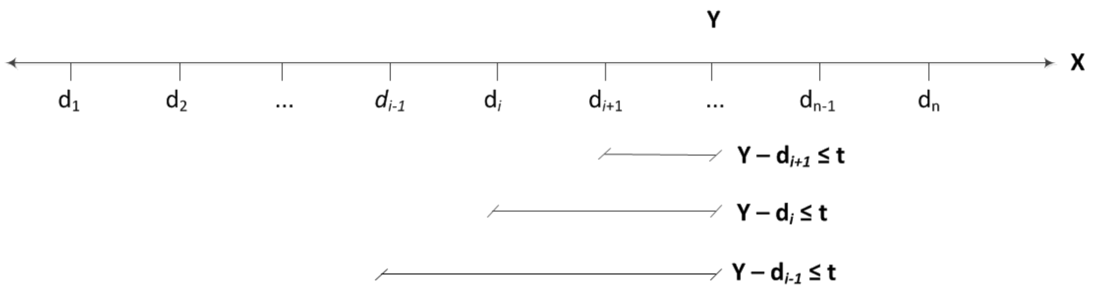


Figure A.1. Domain of X and Y for Case 1

Given that d_{i-1} is a valid instantiation of X, then it can be concluded that the following constraint is a valid:

$$\boldsymbol{Y - d_{i-1} \leq t} \quad \textbf{(A.4)}$$

Also as d_i is an invalid instantiation, it follows that the following constraint is invalid:

$$\boldsymbol{Y - d_i \leq t} \quad \textbf{(A.5)}$$

The distance between d_i and d_{i-1} can be denoted by a positive integer n, i.e. $d_i - d_{i-1} = n$. The interval $Y - d_{i-1} \leq t$ can now be expressed in terms of Y and d_i as:

$$\boldsymbol{Y - d_i \leq t - n} \quad \textbf{(A.6)}$$

Comparing Equation A.5 and A.6 using Rule C2, it can be seen that A.6 subsumes A.5, which means that if A.6 is valid then A.5 must also be valid. However, this contradicts the assumption of invalidity of A.5.

<u>Case 2: $Y - X \geq t$</u>

The same proof is constructed as in the previous case using d_i and d_{i+1}. From Rule C1, it can again be proven by contradiction that d_i must be valid.

For unary constraints, proof by inspection of the domain of X is sufficient to show that d_i must be valid.

The implication of the above proof is that there cannot be an invalid instantiation between the upper and lower bounds. Hence BCSolver is adequate for the problem. A similar proof has been provided by Dechter, *et al.* (1991) for Simple Temporal

Problems (STP), which is a tractable subset of TCSP (Temporal CSPs). In it, they propose a weighted distance graph to prove that a valid and consistent STP is also a minimal graph. This implies that the domains in the valid STP cannot be narrowed further.

A.3.MEP Installation Case Study Data

Table A.2 shows the space entities and their characteristic durations and utilization factors under the various modes. Space entities bearing the name of the activity indicate the workspace while those with a PS prefix denote the path space of the activity. The activities in italics denote optional activities which only occur in Mode 5, while the activities in bold denote activities with alternative modes. *Dur* refers to the duration of the entity, and *Ex* refers to the existence of the entity under that particular mode (1 means it is active, 0 means it does not exist).

Table A.2 Space Entity Case Data

Space Entity	Mode 1 to Mode 3			Mode 1	Mode 2	Mode 3	Mode 4	Mode 5			
	Dur	U_s	U_t	Ex	Ex	Ex	Ex	Dur	U_s	U_t	Ex
Scaffold Erection	4	0.113971	0.8	1	1	1	1	4	0.113971	0.8	1
Bracketry Installation	3	0.113971	0.8	1	1	1	1	3	0.113971	0.8	1
Ductwork Installation	10	0.132353	0.8	1	1	1	1	10	0.132353	0.8	1
Cable and Tray Installation	6	0.132353	0.8	1	1	1	1	6	0.132353	0.8	1
Scaffold Removal	1	0.113971	0.8	1	1	1	1	1	0.113971	0.8	1
Permanent Door Installation VS	6	0.22	0.8	1	1	1	1	6	0.22	0.8	1
Scaffold Erection DB	2	0.155556	0.8	1	1	1	1	2	0.155556	0.8	1
PS Scaffold Erection DB	2	0.033451	0.3	1	1	1	1	2	0.033451	0.3	1
Bracketry Installation DB	2	0.155556	0.8	1	1	1	1	2	0.155556	0.8	1
PS Bracketry Installation DB	2	0.033451	0.3	1	1	1	1	2	0.033451	0.3	1
Ductwork Installation DB	4	0.155556	0.8	1	1	1	1	4	0.155556	0.8	1
PS Ductwork Installation DB	4	0.033451	0.3	1	1	1	1	4	0.033451	0.3	1
Cable and Tray Installation DB	2	0.155556	0.8	1	1	1	1	2	0.155556	0.8	1
PS Cable and Tray Installation DB	2	0.033451	0.3	1	1	1	1	2	0.033451	0.3	1
Scaffold Removal DB	1	0.155556	0.8	1	1	1	1	1	0.155556	0.8	1
PS Scaffold Removal DB	1	0.033451	0.3	1	1	1	1	1	0.033451	0.3	1
Equipment Installation DB	6	0.211111	0.8	1	1	1	1	6	0.211111	0.8	1
PS Equipment Installation DB	6	0.045398	0.3	1	1	1	1	6	0.045398	0.3	1
Permanent Door Installation DB	4	0.28	0.8	1	1	1	1	4	0.28	0.8	1
PS Permanent Door Installation DB	4	0.033451	0.3	1	1	1	1	4	0.033451	0.3	1
Scaffold Erection HX	2	0.173913	0.8	1	1	1	1	2	0.173913	0.8	1
PS Scaffold Erection HX	2	0.058545	0.3	1	1	1	1	2	0.136194	0.3	1
Bracketry Installation HX	2	0.173913	0.8	1	1	1	1	2	0.173913	0.8	1
PS Bracketry Installation HX	2	0.058545	0.3	1	1	1	1	2	0.136194	0.3	1
Ductwork Installation HX	4	0.217391	0.8	1	1	1	1	4	0.217391	0.8	1
PS Ductwork Installation HX	4	0.073182	0.3	1	1	1	1	4	0.170242	0.3	1
Cable and Tray Installation HX	2	0.173913	0.8	1	1	1	1	2	0.173913	0.8	1
PS Cable and Tray Installation HX	2	0.058545	0.3	1	1	1	1	2	0.136194	0.3	1
Scaffold Removal HX	1	0.173913	0.8	1	1	1	1	1	0.173913	0.8	1
PS Scaffold Removal HX	1	0.058545	0.3	1	1	1	1	1	0.136194	0.3	1
Equipment Installation HX (mode 1)	6	0.282609	0.8	**1**	**0**	**0**	**0**	6	0.282609	0.8	**1**
PS Equipment Installation HX (mode 1)	6	0.095136	0.3	**1**	**0**	**0**	**0**	6	0.221315	0.3	**1**
Equipment Installation (mode 2)	13	0.282609	0.8	**0**	**1**	**1**	**1**	13	0.282609	0.8	**0**
PS Equipment Installation (mode 2)	13	0.095136	0.3	**0**	**1**	**1**	**1**	13	0.221315	0.3	**0**
Permanent Door Installation HX	4	0.22	0.8	1	1	1	1	4	0.22	0.8	1
PS Permanent Door Installation HX	4	0.029273	0.3	1	1	1	1	4	0.068097	0.3	1
Sealing of Temporary Access HX (optional activity)	6	0.25	0.8	0	0	0	0	6	0.25	0.8	***1***
PS Sealing of Temporary Access HX (optional activity)	6	0.036591	0.3	0	0	0	0	6	0.085121	0.3	***1***
Scaffold Erection AHU	3	0.15	0.8	1	1	1	1	3	0.15	0.8	1
PS Scaffold Erection AHU	3	0.177073	0.3	1	1	1	1	3	0.3	0.3	1
Bracketry Installation AHU	2	0.15	0.8	1	1	1	1	2	0.15	0.8	1
PS Bracketry Installation AHU	2	0.177073	0.3	1	1	1	1	2	0.3	0.3	1
Ductwork Installation AHU	4	0.191667	0.8	1	1	1	1	4	0.191667	0.8	1
PS Ductwork Installation AHU	4	0.226259	0.3	1	1	1	1	4	0.383333	0.3	1
Cable and Tray Installation AHU	3	0.15	0.8	1	1	1	1	3	0.15	0.8	1
PS Cable and Tray Installation AHU	3	0.177073	0.3	1	1	1	1	3	0.3	0.3	1
Scaffold Removal AHU	2	0.15	0.8	1	1	1	1	2	0.15	0.8	1
PS Scaffold Removal AHU	2	0.177073	0.3	1	1	1	1	2	0.3	0.3	1
Equipment Installation AHU (mode 1)	6	0.233333	0.8	**0**	**0**	**1**	**1**	6	0.233333	0.8	**0**
PS Equipment Installation AHU (mode 1)	6	0.275446	0.3	**0**	**0**	**1**	**1**	6	0.466667	0.3	**0**
Equipment Installation AHU (mode 2)	15	0.233333	0.8	**1**	**1**	**0**	**0**	15	0.233333	0.8	**1**
PS Equipment Installation AHU (mode 2)	15	0.275446	0.3	**1**	**1**	**0**	**0**	15	0.466667	0.3	**1**
Permanent Door Installation AHU	4	0.22	0.8	1	1	1	1	4	0.22	0.8	1
PS Permanent Door Installation AHU	4	0.039349	0.3	1	1	1	1	4	0.066667	0.3	1
Sealing of temporary access AHU (optional	6	0.2	0.8	***0***	***0***	***0***	***0***	6	0.2	0.8	***1***

Space Entity	Mode 1 to Mode 3			Mode 1	Mode 2	Mode 3	Mode 4	Mode 5			
	Dur	U_s	U_t	Ex	Ex	Ex	Ex	Dur	U_s	U_t	Ex
activity)											
PS Sealing of temporary access AHU (optional activity)	6	0.039349	0.3	***0***	***0***	***0***	***0***	6	0.066667	0.3	***1***
Scaffold Erection ECS	2	0.150943	0.8	1	1	1	1	2	0.150943	0.8	1
PS Scaffold Erection ECS	2	0.041834	0.3	1	1	1	1	2	0.041834	0.3	1
Bracketry Installation ECS	2	0.150943	0.8	1	1	1	1	2	0.150943	0.8	1
PS Bracketry Installation ECS	2	0.041834	0.3	1	1	1	1	2	0.041834	0.3	1
Ductwork Installation ECS	4	0.188679	0.8	1	1	1	1	4	0.188679	0.8	1
PS Ductwork Installation ECS	4	0.052293	0.3	1	1	1	1	4	0.052293	0.3	1
Cable and Tray Installation ECS	3	0.188679	0.8	1	1	1	1	3	0.188679	0.8	1
PS Cable and Tray Installation ECS	3	0.052293	0.3	1	1	1	1	3	0.052293	0.3	1
Scaffold Removal ECS	1	0.150943	0.8	1	1	1	1	1	0.150943	0.8	1
PS Scaffold Removal ECS	1	0.041834	0.3	1	1	1	1	1	0.041834	0.3	1
Equipment Installation ECS (mode 1)	4	0.245283	0.8	**0**	**1**	**0**	**0**	4	0.245283	0.8	**0**
PS Equipment Installation ECS (mode 1)	4	0.067981	0.3	**0**	**1**	**0**	**0**	4	0.067981	0.3	**0**
Equipment Installation (mode 2)	8	0.245283	0.8	**1**	**0**	**1**	**1**	8	0.245283	0.8	**1**
PS Equipment Installation (mode 2)	8	0.067981	0.3	**1**	**0**	**1**	**1**	8	0.067981	0.3	**1**
Permanent Door Installation ECS	4	0.22	0.8	1	1	1	1	4	0.22	0.8	1
PS Permanent Door Installation ECS	4	0.020917	0.3	1	1	1	1	4	0.020917	0.3	1
Scaffold Erection TX	2	0.1875	0.8	1	1	1	1	2	0.1875	0.8	1
PS Scaffold Erection TX	2	0.027862	0.3	1	1	1	1	2	0.027862	0.3	1
Bracketry Installation TX	2	0.25	0.8	1	1	1	1	2	0.25	0.8	1
PS Bracketry Installation TX	2	0.037149	0.3	1	1	1	1	2	0.037149	0.3	1
Ductwork Installation TX	4	0.25	0.8	1	1	1	1	4	0.25	0.8	1
PS Ductwork Installation TX	4	0.037149	0.3	1	1	1	1	4	0.037149	0.3	1
Cable and Tray Installation TX	2	0.25	0.8	1	1	1	1	2	0.25	0.8	1
PS Cable and Tray Installation TX	2	0.037149	0.3	1	1	1	1	2	0.037149	0.3	1
Scaffold Removal TX	1	0.1875	0.8	1	1	1	1	1	0.1875	0.8	1
PS Scaffold Removal TX	1	0.027862	0.3	1	1	1	1	1	0.027862	0.3	1
Equipment Installation TX	6	0.25	0.8	1	1	1	1	6	0.25	0.8	1
PS Equipment Installation TX	6	0.037149	0.3	1	1	1	1	6	0.037149	0.3	1
Permanent Door Installation TX	4	0.22	0.8	1	1	1	1	4	0.22	0.8	1
PS Permanent Door Installation TX	4	0.018574	0.3	1	1	1	1	4	0.018574	0.3	1

References

Akinci, B., Fischer, M., and Kunz, J. (2002a). "Automated Generation of Work Spaces Required by Construction Activities." *Journal of Construction Engineering and Management*, 128(4), 306-315.

Akinci, B., Fischer, M., Kunz, J., and Levitt, R. (2002b). "Representing Work Spaces Generically in Construction Method Models." *Journal of Construction Engineering and Management*, 128(4), 296-305.

Akinci, B., Fischer, M., Levitt, R., and Carlson, R. (2002c). "Formalization and Automation of Time-Space Conflict Analysis." *Journal of Computing in Civil Engineering*, 16(2), 124-134.

Allen, J. (1983). "Maintaining knowledge about temporal intervals." *Commun. ACM*, 26(11), 832-843.

Allen, J. F. (1984). "Towards a general theory of action and time." *Artificial Intelligence*, 23(2), 123-154.

Apt, K. (2003). *Principles of Constraint Programming*, Cambridge University Press.

Apt, K. R., and Wallace, M. (2007). *Constraint Logic Programming using ECL^{i}PS^{e}*, Cambridge University Press.

Back, T. "Selective pressure in evolutionary algorithms: a characterization of selection mechanisms." *Proc., Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, 57-62 vol.51.

Bäck, T., Eiben, A., and Van der Vaart, N. (2000). "An Emperical Study on GAs “Without Parameters”." *Parallel Problem Solving from Nature PPSN VI*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, eds., Springer Berlin / Heidelberg, 315-324.

Ballestín, F., Barrios, A., and Valls, V. (2011). "An evolutionary algorithm for the resource-constrained project scheduling problem with minimum and maximum time lags." *Journal of Scheduling*, 14(4), 391-406.

Bansal, V. K. (2011). "Use of GIS and Topology in the Identification and Resolution of Space Conflicts." *Journal of Computing in Civil Engineering*, 25(2), 159-171.

Baptiste, P., Pape, C. L., and Nuitjen, W. (2001). *Constraint-Based Scheduling: Applying Constraint Programming to Scheduling Problems*, Kluwer Academic Publishers.

Barták, R., and Cepek, O. (2007). "Temporal Networks with Alternatives: Complexity and Model." *Proceedings of the Twentieth International Florida Artificial Intellignce Research Society Conference (FLAIRS)*, AAAI Press, Florida, USA, 641-646.

Barták, R., and Čepek, O. (2008). "Nested Precedence Networks with Alternatives: Recognition, Tractability, and Models." *Artificial Intelligence: Methodology, Systems, and Applications*, D. Dochev, M. Pistore, and P. Traverso, eds., Springer Berlin / Heidelberg, 235-246.

Bean, J. C. (1994). "Genetic Algorithms and Random Keys for Sequencing and Optimization." *ORSA Journal on Computing*, 6(2), 154-160.

Beck, J. C., and Fox, M. S. (2000). "Constraint-directed techniques for scheduling alternative activities." *Artificial Intelligence*, 121(1-2), 211-250.

Ben-Ari, M. (1993). *Mathematical logic for computer science*, Prentice-Hall International.

Bo-Christer, B. (1992a). "A conceptual model of spaces, space boundaries and enclosing structures." *Automation in Construction*, 1(3), 193-214.

Bo-Christer, B. (1992b). "A unified approach for modelling construction information." *Building and Environment*, 27(2), 173-194.

Bouchlaghem, D., Kimmance, A. G., and Anumba, C. J. (2004). "Integrating product and process information in the construction sector." *Ind. Mgmt & Data Sys.*, 104(3), 218-233.

Bowers, J. A. (2000). "Multiple schedules and measures of resource constrained float." *J Oper Res Soc*, 51(7), 855-862.

Brucker, P., and Knust, S. (2001). "Resource-Constrained Project Scheduling and Timetabling." *Practice and Theory of Automated Timetabling III*, E. Burke, and W. Erben, eds., Springer Berlin / Heidelberg, 277-293.

Callahan, M. T., Quackenbush, D. G., and Rowings, J. E. (1992). *Construction Project Scheduling*, McGraw-Hill, New York, USA.

Caseau, Y., and Laburthe, F. (1994). "Improved CLP scheduling with task intervals." *Proceedings of the eleventh international conference on Logic programming*, MIT Press.

Chan, W.-T., Chua, D. K. H., and Kannan, G. (1996). "Construction Resource Scheduling with Genetic Algorithms." *Journal of Construction Engineering and Management*, 122(2), 125-132.

Chan, W. T., and Paulson, B. C. J. (1987). "Exploratory design using constraints." *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing*, 1, 59-71.

Chandrasekaran, B., and Josephson, J. R. (2000). "Function in Device Representation." *Engineering with Computers*, 16(3), 162-177.

Chau, K. W., Anson, M., and Zhang, J. P. (2004). "Four-Dimensional Visualization of Construction Scheduling and Site Utilization." *Journal of Construction Engineering and Management*, 130(4), 598-606.

Chau, K. W., Anson, M., and Zhang, J. P. (2005). "4D dynamic construction management and visualization software: 1. Development." *Automation in Construction*, 14(4), 512-524.

Choi, C., Harvey, W., Lee, J., and Stuckey, P. (2006). "Finite Domain Bounds Consistency Revisited." *LNCS AI 2006: Advances in Artificial Intelligence*, A. Sattar, and B.-h. Kang, eds., Springer Berlin / Heidelberg, 49-58.

Chua, D. K. H., Chan, W. T., and Govindan, K. (1997). "A TIME-COST TRADE-OFF MODEL WITH RESOURCE CONSIDERATION USING GENETIC ALGORITHM." *Civil Engineering Systems*, 14(4), 291-311.

Chua, D. K. H., and Song, Y. (2003). "Application of component state model for identifying constructability conflicts in a merged construction schedule." *Advances in Engineering Software*, 34(11-12), 671-681.

Chua, D. K. H., and Yeoh, K. W. (2011). "PDM++: Planning Framework from a Construction Requirements Perspective." *Journal of Construction Engineering and Management*, 137(4), 266-274.

CII (July 1986). *Constructability: A Primer*, Construction Industry Institute, Austin, Texas.

Coelho, J., and Vanhoucke, M. (2011). "Multi-mode resource-constrained project scheduling using RCPSP and SAT solvers." *European Journal of Operational Research*, 213(1), 73-82.

Coello Coello, C. A. (2001). "A Short Tutorial on Evolutionary Multiobjective Optimization." *Evolutionary Multi-Criterion Optimization*, E. Zitzler, L. Thiele, K. Deb, C. Coello Coello, and D. Corne, eds., Springer Berlin / Heidelberg, 21-40.

Cysneiros, L., Julio, and Jaime (2001). "A Framework for Integrating Non-Functional Requirements into Conceptual Models." *Requirements Engineering*, 6(2), 97-115.

Darwiche, A., Levitt, R. E., and Hayes-Roth, B. (1989). "OARPLAN: Generating Project Plans in a Blackboard System by Reasoning about Objects, Actions and Resources." *Technical Report No. 2*, Centre for Integrated Facility Engineering, Stanford University, United States.

Davis, M., Logemann, G., and Loveland, D. (1962). "A machine program for theorem-proving." *Commun. ACM*, 5(7), 394-397.

Dawood, N., and Mallasi, Z. (2006). "Construction Workspace Planning: Assignment and Analysis Utilizing 4D Visualization Technologies." *Computer-Aided Civil and Infrastructure Engineering*, 21(7), 498-513.

Dawood, N., and Sriprasert, E. (2006). "Construction scheduling using multi-constraint and genetic algorithms approach." *Construction Management and Economics*, 24(1), 19 - 30.

De Reyck, B., and Herroelen, W. (1999). "The multi-mode resource-constrained project scheduling problem with generalized precedence relations." *European Journal of Operational Research*, 119(2), 538-556.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II." *Parallel Problem Solving from Nature PPSN VI*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. Merelo, and H.-P. Schwefel, eds., Springer Berlin / Heidelberg, 849-858.

Dechter, R., Meiri, I., and Pearl, J. (1991). "Temporal constraint networks." *Artificial Intelligence*, 49(1-3), 61-95.

Deng, Y. M. (2002). "Function and behavior representation in conceptual mechanical design." *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 16(05), 343-362.

Douglas, I. I. I. E. E., Calvey, T. T., McDonald, J. D. F., and Winter, R. M. (2006). "The Great Negative Lag Debate." *AACE International Transactions*, 2.1-2.7.

Echeverry, D., Ibbs, C. W., and Kim, S. (1991a). "A knowledge-based approach to support the generation of construction schedules." *Comp. & Struc.*, 40(1), 59-66.

Echeverry, D., Ibbs, C. W., and Kim, S. (1991b). "Sequencing Knowledge for Construction Scheduling." *Journal of Construction Engineering and Management*, 117(1), 118-130.

Ekholm, A., and Fridqvist, S. (2000). "A concept of space for building classification, product modelling, and design." *Automation in Construction*, 9(3), 315-328.

El-Bibany, H. (1997). "Parametric Constraint Management in Planning and Scheduling: Computational Basis." *Journal of Construction Engineering and Management*, 123(3), 348-353.

El-Diraby, T. A., Lima, C., and Feis, B. (2005). "Domain Taxonomy for Construction Concepts: Toward a Formal Ontology for Construction Knowledge." *Journal of Computing in Civil Engineering*, 19(4), 394-406.

El-Diraby, T. E., and Kashif, K. F. (2005). "Distributed Ontology Architecture for Knowledge Management in Highway Construction." *Journal of Construction Engineering and Management*, 131(5), 591-603.

El-Gohary, N. M., and El-Diraby, T. E. (2010). "Domain Ontology for Processes in Infrastructure and Construction." *Journal of Construction Engineering and Management*, 136(7), 730-744.

El-Rayes, K., and Moselhi, O. (2001). "Optimizing Resource Utilization for Repetitive Construction Projects." *Journal of Construction Engineering and Management*, 127(1), 18-27.

Fan, S.-L., and Tserng, H. P. (2006). "Object-Oriented Scheduling for Repetitive Projects with Soft Logics." *Journal of Construction Engineering and Management*, 132(1), 35-48.

Fan, S.-L., Tserng, H. P., and Wang, M.-T. (2003). "Development of an object-oriented scheduling model for construction projects." *Automation in Construction*, 12(3), 283-302.

Fischer, M. (2006). "Formalizing Construction Knowledge for Concurrent Performance-Based Design." *Intelligent Computing in Engineering and Architecture*, I. Smith, ed., Springer Berlin / Heidelberg, 186-205.

Fischer, M. A. (1993). "Automating constructibility reasoning with a geometrical and topological project model." *Computing Systems in Engineering*, 4(2–3), 179-192.

Fischer, M. A., and Aalami, F. (1996). "Scheduling with Computer-Interpretable Construction Method Models." *Journal of Construction Engineering and Management*, 122(4), 337-347.

Fisher, D. J., Anderson, S. D., and Rahman, S. P. (2000). "Integrating Constructability Tools into Constructability Review Process." *Journal of Construction Engineering and Management*, 126(2), 89-96.

Gero, J. S., and Kannengiesser, U. (2004). "The situated function–behaviour–structure framework." *Design Studies*, 25(4), 373-391.

Gordon, C., Akinci, B., and James H. Garrett, J. (2007). "Formalism for Construction Inspection Planning: Requirements and Process Concept." *Journal of Computing in Civil Engineering*, 21(1), 29-38.

Griffith, A., and Sidwell, A. C. (1993). "Development of constructability concepts, principles and practices." *ECAM*, 4(4), 295-310.

Guo, S.-J. (2002). "Identification and Resolution of Work Space Conflicts in Building Construction." *Journal of Construction Engineering and Management*, 128(4), 287-295.

Hajdu, M. (1997). *Network Scheduling Techniques for Construction Project Management*, Springer.

Hanlon, E. J., and Sanvido, V. E. (1995). "Constructability Information Classification Scheme." *Journal of Construction Engineering and Management*, 121(4), 337-345.

Harhalakis, G. (1990). "Special features of precedence network charts." *European Journal of Operational Research*, 49(1), 50-59.

Harhalakis, G., J Davies, B., and Manzoor, T. (1987). "Generalized algorithm for the time analysis of summary activities." *International Journal of Project Management*, 5(1), 11-18.

Hartmann, S., and Briskorn, D. (2010). "A survey of variants and extensions of the resource-constrained project scheduling problem." *European Journal of Operational Research*, 207(1), 1-14.

Hartmann, T., and Fischer, M. (2007). "Supporting the constructability review with 3D/4D models." *Building Research & Information*, 35(1), 70-80.

Heesom, D., and Mahdjoubi, L. (2004). "Trends of 4D CAD applications for construction planning." *Construction Management and Economics*, 22(2), 171-182.

Hegazy, T., and Menesi, W. (2010). "Critical Path Segments Scheduling Technique." *Journal of Construction Engineering and Management*, 136(10), 1078-1085.

Heilmann, R. (2001). "Resource–constrained project scheduling: a heuristic for the multi–mode case." *OR Spectrum*, 23(3), 335-357.

Heilmann, R. (2003). "A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags." *European Journal of Operational Research*, 144(2), 348-365.

Hyari, K., and El-Rayes, K. (2006). "Optimal Planning and Scheduling for Repetitive Construction Projects." *Journal of Management in Engineering*, 22(1), 11-19.

Jaafari, A. (1984). "Criticism of CPM for Project Planning Analysis." *Journal of Construction Engineering and Management*, 110(2), 222-233.

Jaffar, J., and Maher, M. J. (1994). "Constraint logic programming: a survey." *The Journal of Logic Programming*, 19-20(Supplement 1), 503-581.

Jaffar, J., Maher, M. J., Stuckey, P. J., and Yap, R. H. C. (1994). "Beyond Finite Domains." *Proceedings of the Second International Workshop on Principles and Practice of Constraint Programming*, Springer-Verlag, 86-94.

Jakowski, P., and Sobotka, A. (2006). "Scheduling Construction Projects Using Evolutionary Algorithm." *Journal of Construction Engineering and Management*, 132(8), 861-870.

Jongeling, R., Kim, J., Fischer, M., Mourgues, C., and Olofsson, T. (2008). "Quantitative analysis of workflow, temporary structure usage, and productivity using 4D models." *Automation in Construction*, 17(6), 780-791.

Jongeling, R., and Olofsson, T. (2007). "A method for planning of work-flow by combined use of location-based scheduling and 4D CAD." *Automation in Construction*, 16(2), 189-198.

Jureta, I. J., Mylopoulos, J., and Faulkner, S. (2009). "A core ontology for requirements." *Applied Ontology*, 4(3), 169-244.

Kamara, J. M., Anumba, C. J., and Evbuomwan, N. F. O. (1999). "Client Requirements Processing in Construction: A New Approach Using QFD." *Journal of Architectural Engineering*, 5(1), 8-15.

Kamara, J. M., Anumba, C. J., and Evbuomwan, N. F. O. (2000). "Establishing and processing client requirements - a key aspect of concurrent engineering in construction." *Engineering Construction & Architectural Management*, 7(1), 15-28.

Kamara, J. M., Anumba, C. J., and Evbuomwan, N. F. O. (2002). *Capturing Client requirements in Construction Projects*, Thomas Telford Publishing, London.

Kamat, V. R., and Martinez, J. C. (2001). "Visualizing Simulated Construction Operations in 3D." *Journal of Computing in Civil Engineering*, 15(4), 329-337.

Kamat, V. R., and Martinez, J. C. (2005). "Dynamic 3D Visualization of Articulated Construction Equipment." *Journal of Computing in Civil Engineering*, 19(4), 356-368.

Kamat, V. R., Martinez, J. C., Fischer, M., Golparvar-Fard, M., Pena-Mora, F., and Savarese, S. (2011). "Research in Visualization Techniques for Field Construction." *Journal of Construction Engineering and Management*, 137(10), 853-862.

Kartam, N., and Flood, I. (1997). "Constructability Feedback Systems: Issues and Illustrative Prototype." *Journal of Performance of Constructed Facilities*, 11(4), 178-183.

Kartam, S., Ballard, G., and Ibbs, C. W. (1997). "Introducing a New Concept and Approach to Modeling Construction." *Journal of Construction Engineering and Management*, 123(1), 89-97.

Klein, E. E., and Herskovitz, P. J. (2005). "Philosophical foundations of computer simulation validation." *Simulation & Gaming*, 36(3), 303-329.

Koo, B., and Fischer, M. (2000). "Feasibility Study of 4D CAD in Commercial Construction." *Journal of Construction Engineering and Management*, 126(4), 251-260.

Koo, B., Fischer, M., and Kunz, J. (2007). "Formalization of Construction Sequencing Rationale and Classification Mechanism to Support Rapid Generation of Sequencing Alternatives." *Journal of Computing in Civil Engineering*, 21(6), 423-433.

Kuster, J., Jannach, D., and Friedrich, G. (2007). "Handling Alternative Activities in Resource-Constrained Project Scheduling Problems." *IJCAI*, M. M. Veloso, ed., 1960-1965.

Laborie, P., and Rogerie, J. (2008). "Reasoning with Conditional Time-intervals." *PROCEEDINGS OF THE TWENTY-FIRST INTERNATIONAL FLORIDA ARTIFICIAL INTELLIGENCE RESEARCH SOCIETY (FLAIRS) CONFERENCE*, D. C. Wilson, and H. C. Lane, eds., The AAAI Press, Menlo Park, California, Coconut Grove, Florida, 555-560.

Laborie, P., Rogerie, J., Shaw, P., and Vilim, P. (2009). "Reasoning with Conditional Time-Intervals. Part II: An Algebraical Model for Resources." *PROCEEDINGS OF THE TWENTY-SECOND INTERNATIONAL FLORIDA ARTIFICIAL INTELLIGENCE RESEARCH SOCIETY (FLAIRS) CONFERENCE*, H. C. Lane, and H. W. Guesgen, eds., The AAAI Press, Menlo Park, California, Sanibel Island, Florida, 201-206.

Lee, T.-J., and Soh, C.-K. (1993). "An integrated approach to generate construction schedules." *Computing Systems in Engineering*, 4(2–3), 307-315.

M. Calhoun, K., Deckro, R. F., Moore, J. T., Chrissis, J. W., and Van Hove, J. C. (2002). "Planning and re-planning in project and production scheduling." *Omega*, 30(3), 155-170.

Mahalingam, A., Kashyap, R., and Mahajan, C. (2010). "An evaluation of the applicability of 4D CAD on construction projects." *Automation in Construction*, 19(2), 148-159.

Maher, M. L., Simoff, S. J., and Mitchell, J. (1997). "Formalising building requirements using an Activity/Space Model." *Automation in Construction*, 6(2), 77-95.

Mahoney, J. J., and Tatum, C. B. (1994). "Construction Site Applications of CAD." *Journal of Construction Engineering and Management*, 120(3), 617-631.

Mallasi, Z. (2006). "Dynamic quantification and analysis of the construction workspace congestion utilising 4D visualisation." *Automation in Construction*, 15(5), 640-655.

Maruo, M., Lopes, H., and Delgado, M. (2005). "Self-Adapting Evolutionary Parameters: Encoding Aspects for Combinatorial Optimization Problems." *Evolutionary Computation in Combinatorial Optimization*, G. Raidl, and J. Gottlieb, eds., Springer Berlin / Heidelberg, 154-165.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., and Oltramari, A. (2003). "Ontology library(final)." *Technical Report, Wonder Web Deliverable D18*, Institute of Cognitive Science and Technology, Italian National Research Council.

McKinney, K., and Fischer, M. (1998). "Generating, evaluating and visualizing construction schedules with CAD tools." *Automation in Construction*, 7(6), 433-447.

Mitrovic, D., Male, S., Hunter, I., and Watson, A. (1999). "Large Scale Engineering project processand user requirements." *ECAM*, 6(1), 38-50.

Moder, J. J., Phillips, C. R., and Davis, E. W. (1983). *Project Management With Cpm, Pert and Precedence Diagramming*, Van Nostrand Reinhold Company Inc, New York, USA.

Morad, A. A., and Beliveau, Y. J. (1994). "Geometric-Based Reasoning System for Project Planning." *Journal of Computing in Civil Engineering*, 8(1), 52-71.

Mylopoulos, J., Chung, L., and Nixon, B. (1992). "Representing and using nonfunctional requirements: a process-oriented approach." *Software Engineering, IEEE Transactions on*, 18(6), 483-497.

Navon, R., Shapira, A., and Shechori, Y. (2000). "Automated Rebar Constructability Diagnosis." *Journal of Construction Engineering and Management*, 126(5), 389-397.

Neumann, K., and Schwindt, C. (1997). "Activity-on-node networks with minimal and maximal time lags and their application to make-to-order production." *OR Spectrum*, 19(3), 205-217.

Neumann, K., and Zhan, J. (1995). "Heuristics for the minimum project-duration problem with minimal and maximal time lags under fixed resource constraints." *Journal of Intelligent Manufacturing*, 6(2), 145-154.

Nguyen, T.-H., and Oloufa, A. A. (2001). "Computer-Generated Building Data: Topological Information." *Journal of Computing in Civil Engineering*, 15(4), 268-274.

Nieuwenhuis, R., Oliveras, A., and Tinelli, C. (2006). "Solving SAT and SAT Modulo Theories: From an abstract Davis--Putnam--Logemann--Loveland procedure to DPLL(*T*)." *Journal of ACM*, 53(6), 937-977.

O'Brien, J. J., and Plotnick, F. L. (2010). *CPM in Construction Management, Seventh Edition*, McGraw-Hill Companies Inc, New York, USA.

Park, J., Kim, B., Kim, C., and Kim, H. (2011). "3D/4D CAD Applicability for Life-Cycle Facility Management." *Journal of Computing in Civil Engineering*, 25(2), 129-138.

Plotnick, F. L. (2006). "RDM--Relationship Diagramming Method." *AACE International Transactions*, 8.1-8.10.

Project Management Institute, P. (2008). *A guide to the project management body of knowledge (PMBOK® guide)*, Project Management Institute.

Riley, D. R., and Sanvido, V. E. (1995). "Patterns of Construction-Space Use in Multistory Buildings." *Journal of Construction Engineering and Management*, 121(4), 464-473.

Riley, D. R., and Sanvido, V. E. (1997). "Space Planning Method for Multistory Building Construction." *Journal of Construction Engineering and Management*, 123(2), 171-180.

Schwalb, E., and Dechter, R. (1997). "Processing disjunctions in temporal constraint networks." *Artificial Intelligence*, 93(1–2), 29-61.

Serpell, M., and Smith, J. E. (2010). "Self-Adaptation of Mutation Operator and Probability for Permutation Representations in Genetic Algorithms." *Evolutionary Computation*, 18(3), 491-514.

Skibniewski, M., Arciszewski, T., and Lueprasert, K. (1997). "Constructability Analysis: Machine Learning Approach." *Journal of Computing in Civil Engineering*, 11(1), 8-16.

Smętek, M., and Trawiński, B. (2011). "Investigation of Genetic Algorithms with Self-Adaptive Crossover, Mutation, and Selection." *Hybrid Artificial Intelligent Systems*, E. Corchado, M. Kurzynski, and M. Wozniak, eds., Springer Berlin / Heidelberg, 116-123.

Smith, J., and Fogarty, T. C. "Self adaptation of mutation rates in a steady state genetic algorithm." *Proc., Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, 318-323.

Soibelman, L., Liu, L. Y., Kirby, J. G., East, E. W., Caldas, C. H., and Lin, K.-Y. (2003). "Design Review Checking System with Corporate Lessons Learned." *Journal of Construction Engineering and Management*, 129(5), 475-484.

Soibelman, L., and Pena-Mora, F. (2000). "Distributed Multi-Reasoning Mechanism to Support Conceptual Structural Design." *Journal of Structural Engineering*, 126(6), 733-742.

Song, Y. (2006). "Intermediate Function Requirements for Constructability Analysis." PhD, National University of Singapore, Singapore.

Song, Y., and Chua, D. K. H. (2005). "Detection of spatio-temporal conflicts on a temporal 3D space system." *Advances in Engineering Software*, 36(11-12), 814-826.

Song, Y., and Chua, D. K. H. (2006). "Modeling of Functional Construction Requirements for Constructability Analysis." *Journal of Construction Engineering and Management*, 132(12), 1314-1326.

Song, Y., and Chua, D. K. H. (2007). "Temporal Logic Representation Schema for Intermediate Function." *Journal of Construction Engineering and Management*, 133(4), 277-286.

Staub-French, S., Fischer, M., Kunz, J., Ishii, K., and Paulson, B. (2003a). "A feature ontology to support construction cost estimating." *AI EDAM*, 17(02), 133-154.

Staub-French, S., Fischer, M., Kunz, J., and Paulson, B. (2003b). "An Ontology for Relating Features with Activities to Calculate Costs." *Journal of Computing in Civil Engineering*, 17(4), 243-254.

Stergiou, K., and Koubarakis, M. (1998). "Backtracking algorithms for disjunctions of temporal constraints." *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence,* American Association for Artificial Intelligence, Madison, Wisconsin, United States, 248-253.

Stergiou, K., and Koubarakis, M. (2000). "Backtracking algorithms for disjunctions of temporal constraints." *Artif. Intell.*, 120(1), 81-117.

Stumpf, A. L., Ganeshan, R., Chin, S., and Liu, L. Y. (1996). "Object-Oriented Model for Integrating Construction Product and Process Information." *Journal of Computing in Civil Engineering*, 10(3), 204-212.

Tamimi, S., and Diekmann, J. (1988). "Soft Logic in Network Analysis." *Journal of Computing in Civil Engineering*, 2(3), 289-300.

Tantisevi, K., and Akinci, B. (2007). "Automated generation of workspace requirements of mobile crane operations to support conflict detection." *Automation in Construction*, 16(3), 262-276.

Tatum, C. B., Vanegas, J. A., and Williams, J. M. (1986). "Constructability Improvement During Conceptual Planning." Construction Industry Institute, The University of Texas at Austin.

Thabet, W. Y., and Beliveau, Y. J. (1994a). "HVLS: Horizontal and Vertical Logic Scheduling for Multistory Projects." *Journal of Construction Engineering and Management*, 120(4), 875-892.

Thabet, W. Y., and Beliveau, Y. J. (1994b). "Modeling Work Space to Schedule Repetitive Floors in Multistory Buildings." *Journal of Construction Engineering and Management*, 120(1), 96-116.

Thomas, H. R., and Horman, M. J. (2006). "Fundamental Principles of Workforce Management." *Journal of Construction Engineering and Management*, 132(1), 97-104.

Thomas, H. R., Riley, D. R., and Sinha, S. K. (2006). "Fundamental Principles for Avoiding Congested Work Areas---A Case Study." *Practice Periodical on Structural Design and Construction*, 11(4), 197-205.

Tsamardinos, I., and Pollack, M. E. (2003). "Efficient solution techniques for disjunctive temporal reasoning problems." *Artificial Intelligence*, 151(1-2), 43-89.

Ugwu, O. O., Anumba, C. J., and Thorpe, A. (2005). "Ontological foundations for agent support in constructability assessment of steel structures—a case study." *Automation in Construction*, 14(1), 99-114.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T. (1996). "Supporting conceptual design based on the function-behavior-state modeler." *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 10(04), 275-288.

Umeda, Y., Takeda, H., Tomiyama, T., and Yoshikawa, H. (1990). "Function, behaviour, and structure." *Applications of Artificial Intelligence in Engineering: Proceedings of the Fifth International Conference*, J. S. Gero, ed., Springer-Verlag, Berlin, Boston, USA, 177-194.

Vanhoucke, M. (2006). "Work Continuity Constraints in Project Scheduling." *Journal of Construction Engineering and Management*, 132(1), 14-25.

Vilain, M., Kautz, H., and Beek, P. v. (1990). "Constraint propagation algorithms for temporal reasoning: a revised report." *Readings in qualitative reasoning about physical systems*, Morgan Kaufmann Publishers Inc., 373-381.

Węglarz, J., Józefowska, J., Mika, M., and Waligóra, G. (2011). "Project scheduling with finite or infinite number of activity processing modes – A survey." *European Journal of Operational Research*, 208(3), 177-205.

Wiest, J. D. (1981). "Precedence diagramming method: Some unusual characteristics and their implications for project managers." *Journal of Operations Management*, 1(3), 121-130.

Winch, G. M., and North, S. (2006). "Critical Space Analysis." *Journal of Construction Engineering and Management*, 132(5), 473-481.

Winstanley, G., Chacon, M. A., and Levitt, R. E. (1993). "Model-Based Planning: Scaled-Up Construction Application." *Journal of Computing in Civil Engineering*, 7(2), 199-217.

Yamazaki, Y. (1995). "An integrated construction planning system using object-oriented product and process modelling." *Construction Management and Economics*, 13(5), 417-426.

Yurchyshyna, A., and Zarli, A. (2009). "An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction." *Automation in Construction*, 18(8), 1084-1098.

Zhang, Y., and Wu, H. (1998). "Bound Consistency on Linear Constraints in Finite Domain Constraint." *Proceedings of Thirteenth European Conference on Artificial Intelligence (ECAI-1998)*, H. Prade, ed., John Wiley and Sons, Brighton, UK, 265-266.

Zheng, D. X. M., Ng, S. T., and Kumaraswamy, M. M. (2004). "Applying a Genetic Algorithm-Based Multiobjective Approach for Time-Cost Optimization." *Journal of Construction Engineering and Management*, 130(2), 168-176.

Zhu, G., Bard, J. F., and Yu, G. (2006). "A Branch-and-Cut Procedure for the Multimode Resource-Constrained Project-Scheduling Problem." *INFORMS Journal on Computing*, 18(3), 377-390.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). "SPEA2: Improving the Strength Pareto Evolutionary Algorithm." *TIK-Report 103*, Swiss Federal Institute of Technology (ETH) Zurich.

Zouein, P. P., and Tommelein, I. D. (2001). "Improvement Algorithm for Limited Space Scheduling." *Journal of Construction Engineering and Management*, 127(2), 116-124.

Zozaya-Gorostiza, C., Hendrickson, C., and Rehak, D. R. (1990). "A knowledge-intensive planner for construction projects." *Building and Environment*, 25(3), 269-278.

List of Publications Related to This Research

International Conferences

D.K.H. Chua, K.W. Yeoh, Y. Song (2007). *Modelling the Spatial Temporal Utilization of Construction Space Requirements using 4D CAD*, in B.H.V. Topping, (Editor), Proceedings of the Eleventh International Conference on Civil, Structural and Environmental Engineering Computing, Civil-Comp Press, Stirlingshire, UK, Paper 85, 2007. doi:10.4203/ccp.86.85

Martinus van de Ruitenbeek, Yeoh Ker-Wei, Florin Leon (2008). *Invariant Problem Understanding for Structural Engineers*, Proceedings of the 11th International Conference on Computer Graphics and Artificial Intelligence, 3IA'2008, pp. 211-216

Florin Leon, Yeoh Ker-Wei, Martinus van de Ruitenbeek (2008). *A Neurological Agent Model for Requirements Specification in Structural Engineering Design*, Third International Conference on Design Computing and Cognition, DCC'08, Atlanta, Georgia, USA, June 2008

D.K.H. Chua, Yeoh Ker-Wei (2009). *A Framework for Construction Requirements Based Planning Utilizing Constraints Logic Programming*, 17th International Conference of the International Group for Lean Construction, IGLC17’2009, Taipei, Taiwan, July 2009

Nguyen Thi Qui, David K.H. Chua, and Ker-Wei, Yeoh (2010). *Functional Requirement Oriented Framework for Schedule Generation*, 6th International conference on Innovation in Architecture, Engineering and Construction (AEC), State College, Pennsylvania, USA, June 2010.

T.Q. Nguyen, David K.H. Chua, and K.W. Yeoh (2010). *Extended Functional Requirement Model for Construction Schedule Computation*, 23rd KKCNN Symposium on Civil Engineering, Taipei, Taiwan, November 2010

K.W. Yeoh, and David K.H. Chua (2012). *Mitigating Workspace Congestion: A Genetic algorithm Approach*, 3rd International Conference of Engineering, Production and Project Management, University of Brighton, UK, September 2012

International Journals (Submitted)

D.K.H. Chua, Yeoh Ker-Wei, Y. Song (2010). *Quantification of Spatial Temporal Congestion in 4D CAD*, Journal of Construction Engineering and Management, 136(6), 641-649

D.K.H. Chua, K.W. Yeoh (2011). *PDM++: A Planning Framework from Construction Requirements Perspective*, Journal of Construction Engineering and Management, 137(4), 266-274

Z. Liu, D.K.H. Chua, K.W. Yeoh, E.L.S Abbott (2011). *Aggregate Production Planning for Shipbuilding with Variation-Inventory Tradeoffs*, International Journal of Production Research, 49(20), 6249-6272