The Pennsylvania State University

The Graduate School

College of Engineering

RESOURCE AND SPACE CONSTRAINED PROJECT SCHEDULING

A Dissertation in

Industrial Engineering

by

Daniel A. Finke

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

May 2010

The dissertation of Daniel A. Finke was reviewed and approved* by the following:

Deborah J. Medeiros
Associate Professor of Industrial Engineering
Dissertation Co-Advisor
Chair of Committee

Mark T. Traband
Dissertation Co-Advisor
Affiliate Faculty Member of Industrial Engineering

Catherine Harmonosky
Associate Professor of Industrial Engineering

John Messner
Associate Professor of Architectural Engineering

Michael Yukish
Research Associate
Special Member

Vaughn Whisker
Research Associate
Special Member

Paul Griffin
Professor of Industrial Engineering
Head of the Department of Industrial and Manufacturing Engineering

*Signatures are on file in the Graduate School

ABSTRACT

Production planning in large-scale industries such as construction and shipbuilding is a key component in the manufacturing process and has a significant effect on the overall cost and schedule of the final product. This is due to the fact that the successful fabrication of these products requires a significant amount of material and resources, all of which must be coordinated and appropriately scheduled. Project scheduling methods have been used to model and schedule activities in these industries with great success. Until recently, spatial resources, such as floor space and interior space, have not been taken into consideration in the planning phase of a project. Traditionally, spatial conflicts have been resolved locally as they occur, often resulting in costly schedule delays.

This thesis develops a framework to model space in a project scheduling problem comprised of installation activities within the interior of a large-scale product. The objective is to find a schedule with near-minimum makespan. The installation project activities are constrained by precedence relationships and limited resource availabilities. Space is modeled as a special resource. The space modeling approach approximates the space required by the activities and uses those space requirements to calculate spatial conflicts. Space conflicts are incorporated via a congestion function, which increases the duration of the installation activities with overlapping spatial requirements, reflecting productivity losses due to interference. The congestion function is based on the conflict volume and the length of time the conflict occurs.

A prototype software application is developed from the framework to demonstrate the feasibility of the method. A hypothetical problem and a case study problem are solved using the application. The results show that considering spatial constraints increased the project makespan from 7% to 16% for the hypothetical problems and 35% for the case study problem.

The developed framework establishes an integrated methodology to model the effects of worker and activity space congestion during schedule generation. In previous work, space conflicts were identified after the schedule was completed and resolved manually. The congestion function developed in this thesis explicitly considers types of space and the volume and time duration of conflicts among types. Previous work has either modeled multiple space types without estimating the effect on activity duration, or created a single function independent of space type.

By incorporating realistic space requirements and considering the effect of spatial conflicts during schedule generation, a more realistic project schedule can be created that limits costly schedule delays attributed to less detailed planning practices. Congestion is inherently present in installation projects and has been managed by front line supervisors with little help from formal scheduling tools. The framework developed in this thesis enables front line supervisors to account for and control congestion to minimize schedule delays and project costs.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to thank Dr. Medeiros for her guidance and tireless work guiding me through this process. Technical discussions with her and Mark Traband helped to focus the work and guide me through the process.

I would also like to thank my committee for taking the time to read the proposal and the final dissertation. The comments and feedback from the comprehensive exam helped ground the solution methodology while contributing to the quality of the final product.

Thank you to my wife, Erinn, for listening to my garage filling story several times and offering support when progress stalled.

I would also like to thank Chris Ligetti and Dave Hadka for their support in implementing the methodology into a working software application.

In addition, I would like to thank Joe Hadfield for providing the motivating example used to develop the methodology. His initial concept of the issue and future application formed the basis from which this work is derived.

Chapter 1

Introduction

Planning and scheduling work activities in production systems is a critical step in the manufacturing process. This is evident in the fact that a good schedule can reduce cost, improve quality and ensure on-time delivery of products (Heesom and Mahdjoubi 2004). The planning phase is often quite difficult and time consuming and in many organizations much of the detailed planning is left for real-time shop floor control and shopfloor supervisor intervention. In large-scale product industries such as building construction, shipbuilding or aerospace, vast amounts of material and resources are employed to build the final product. Coordination and timing of the work activities in these industries has become more critical as cost and schedule budgets for these products have decreased.

The large-scale nature of the products and required resources in these industries compounds the difficulty of planning, and it makes the impact of planning that much more significant. Traditionally, planning in these industries has been performed using project scheduling tools. Much research has been done on the general planning and scheduling of large-scale products and can be found in the project scheduling literature. Although there is a wealth of research in this area, there are still areas of opportunity for research and development. One of these open areas of research is the modeling and analysis of space in the planning and scheduling process.

Because of their physical size, large-scale product manufacturing facilities are inherently space constrained. For example, in the construction of a large building, the build site is usually not allowed to spread indefinitely. In most cases, prime space is located near the building for ease of lifting and handling of material to the workplace. Space within the building is also a constraint and changes as the construction process progresses (Winch and North 2006). As components (walls, fixtures, electrical systems, etc.) are added to the internal structure of the building, free working space is diminished, thus making spatial scheduling more critical.

In this work, the manufacturing of large products in the shipbuilding industry will be used to demonstrate a resource-constrained project scheduling-based methodology that seeks to minimize project makespan while accommodating space constraints. Drawing from current literature, it will be shown that space is a critical resource that must be taken into consideration in the planning and scheduling phase of construction. The inclusion of spatial resources in the project scheduling framework will be demonstrated and a methodology will be developed to solve the resulting resource-constrained project scheduling problem.

1.1 Background

The process of building a ship is complex and requires a significant amount of material and resources. The difficulty of managing the material and resources is amplified by the fact that project durations are long (~2-3 months to ~2-3 years). During these long durations, several issues arise that cause the process not to go as planned.

Planning and scheduling of the material and resources has a dramatic effect on the final cost of the product (Chevallier and Russell 1998). Therefore, it is essential that an effective plan be developed to ensure that the products are completed on time and within budget.

The traditional approach used to develop plans for products in these industries is to model the production of the product as a resource-constrained project scheduling problem (RCPSP). This approach uses traditional project scheduling methods to schedule the manufacturing activities. The RCPSP has been thoroughly researched and has spawned numerous solution procedures and problem extensions. Extensions to the RCPSP have been the result of researchers' attempts to accurately modeling the real-world problems such as integrating stochastic elements into the model or planning for multiple projects (products) using the same resources.

In the shipbuilding industry, planning and scheduling is a hierarchical process. First, a high-level plan is developed that defines broad activities and the predecessor and successor relationships between those activities. From the plan, a more detailed schedule characterized by start date, end date, duration, and assigned resources is developed.

To develop the list of activities, the ship is dissected into smaller construction units. Figure 1-1 shows a typical dissection of a ship down to the hull block level.

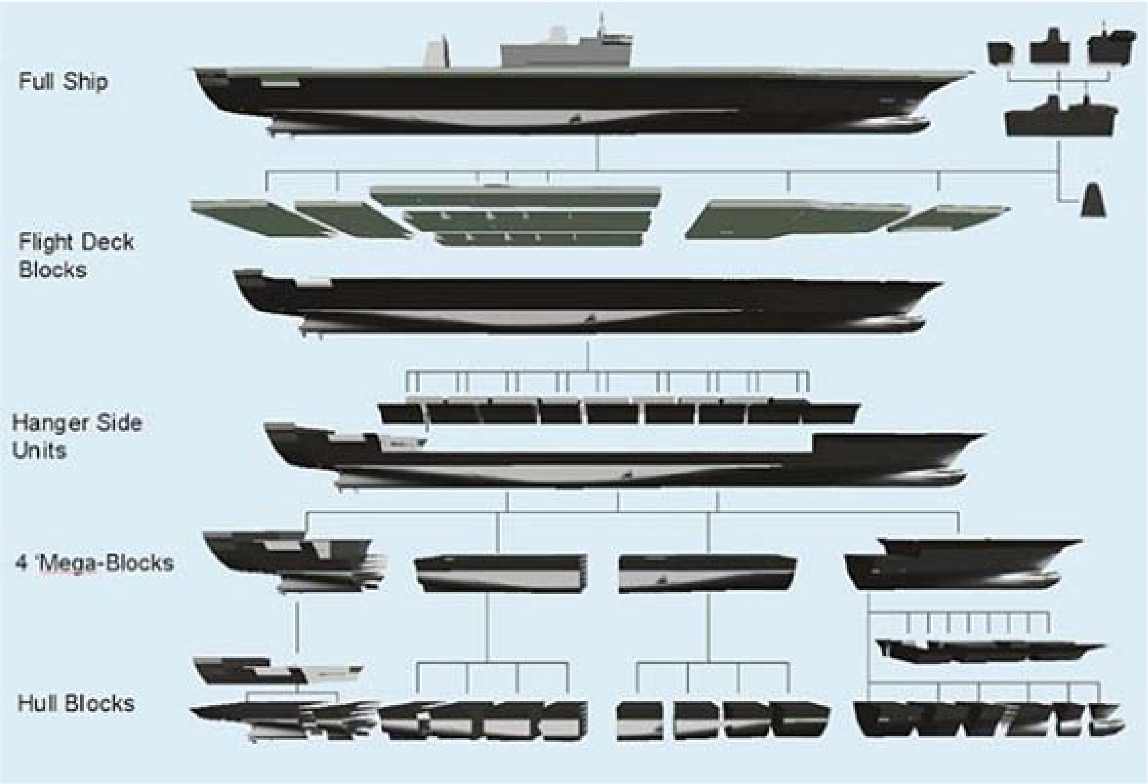


Figure 1-1: General Shipbuilding Product Breakdown Diagram (Beedall 2008).

The hull blocks shown in Figure 1-2 are further dissected into manageable work units, e.g. hull blocks to units, units to assemblies, and assemblies to piece parts. A duration is estimated for the activity of constructing each assembly and the required resources are identified. Once the activities are defined for the assemblies, the activities are defined for the units in a similar fashion. The planning phase usually stops at the unit level and then the scheduling phase begins. The scheduling phase is concerned with a much more detailed list of activities on all levels of the product structure. The specificity of the lowest level of detail for the activity definitions varies across general large-scale product industries and even among the different manufacturers within the same industry.

Because of the large quantity of detailed activities, scheduling systems often cannot or do not capture all of the resource requirements or perform resource capacity constraining functions. For this reason, the responsibility to allocate specific resources to activities is left to construction supervisors. Furthermore, space has little to no representation in most scheduling systems and is almost always left for the supervisors to allocate and modify.

This is a very simple description of the planning and scheduling phase of a project, but nonetheless shows some specific areas where there can be improvement in the process. The hierarchical nature of current planning procedures implicitly constricts manufacturing options at the lower levels of the hierarchical tree. Resource allocation at a detailed level is lacking. Space is not considered in most modern planning systems for the general large-scale product industries, but should be incorporated into the planning and scheduling functions just like any other resource.

Consideration of space is critical in the shipbuilding industry, because space is a constrained resource. There are two facets in space utilization that should be considered in the planning process; 1) build location space and 2) interior work space. Units and assemblies in shipbuilding are very large and require a great deal of floor space while they are being constructed. In addition to floor space, they also need to be located such that they can be accessed by resources such as cranes and lifts. A significant amount of work (thousands of labor hours) is performed inside these large construction units. Although the units and assemblies themselves are quite large, the interior space is quite limited. The final phase of construction often requires several groups of trade personnel (welders, electricians, ship fitters, etc.) to work within the same limited space during the same time frame. This results in worker congestion and frequently leads to schedule delays that affect the future activities and the overall project duration.

Floor space allocation (footprinting, work piece space allocation, or site planning) is the problem of allocating floor space within a defined boundary to large units of work for a period of time. For example, in the shipbuilding industry, large units are produced in buildings that have specialized resources (cranes, foundation support, fixtures, etc.) and a defined perimeter (building walls). The allocation of floor space to the units considers the specialized resources along with schedule and unit size to develop a floor space plan.

Spatial scheduling also includes allocating space within the shipbuilding units to perform final work functions (installation of piping, electrical systems, ventilation, etc.). As the internal space within the units becomes more confined, with the addition of work in place, and the amount of detail or finishing work increases (tasks that involve the installation of smaller pieces, but where there are generally more pieces to install), the coordination of work, resources, and space becomes increasingly difficult. In addition, the number of functions that must be accounted for increases the complexity even further, in that additional trade groups (plumbing, electrical, etc.) are required to perform these detail or finishing work activities. Additional trade groups introduce additional work space congestion. The interior work space allocation problem requires the definition of additional space types. In the floor space scheduling problem, an approach where a two-dimensional representation is replicated across time intervals can be used. However this approach is not an adequate model for the interior space problem. Instead, a three-dimensional representation of space requirements combined with the schedule (time) will need to be used to model the interior space planning problem.

Space allocation and scheduling has a major impact on the overall production schedule and until recently, it has been left to the manufacturing supervisors to effectively react to these effects. Research in the building construction industry has recently begun to integrate space planning into the project planning phase of building construction. There is still a need for advanced tools and methods for practitioners to fully incorporate space planning into practice. Therefore, the focus of this research is on the interior space planning problem.

1.2 Problem Description

This project focuses on the integration of space resources into the traditional planning and scheduling phases of large unit construction in general large-scale product industries. Although space planning is an issue in most of these industries, the shipbuilding industry has been chosen as the application environment for this work. This industry was chosen as the prototype industry because space is a constraint at almost every step in the construction process in this industry. In addition, the detail or finish work that is performed in ship construction is extremely susceptible to heavy space congestion and worker interferences.

The example system used in this project is the general construction of a ship at a typical U.S. shipyard. The ship is comprised of several units, house sized structures, which are comprised of about 30 major assemblies. The assemblies are constructed from subassemblies and piece parts. The general work and material flow is given in Figure 1-2.

Figure 1-2: General Product Work and Material Flow.

Resource allocation in shipbuilding project schedules is a critical component of the planning process. There are three types of resources: renewable resources, nonrenewable resources, and doubly constrained resources (Weglarz 1998; Brucker, Drexl et al. 1999).

> “Renewable resources (e.g. manpower, machines, tools, equipment, space, …) are available on a period-by-period basis, that is, the available amount is renewed from period to period. Only the total resource usage at every time instant is constrained. Nonrenewable resources (e.g. money, raw materials, energy, …), on the contrary, are available on a total project basis, with a limited consumption availability for the entire project. Doubly-constrained resources are constrained per period (e.g. per period cash flow) as well as for the overall project (e.g. total expenditures, overall pollution limits, …) (Weglarz 1998).”

From the preceding definition, both renewable and nonrenewable resources will be considered here. Resources such as labor trade groups and cranes are considered renewable. Space will be considered as both renewable and nonrenewable. Space that is required around a unit for setup, but is available for consumption once that activity is complete is renewable, however space that is consumed by work in place cannot be released and is considered nonrenewable.

To narrow the focus of the problem, this project will primarily be concerned with the outfitting phase of construction. This phase is comprised of the following activities: assembly installation, plumbing, electrical, ventilation, and other detail or finish work. Each of the outfitting activities is performed by a separate and distinct group of personnel, i.e., trade group. Each of the trade group types is comprised of a pool of individuals that are specifically trained in that trade.

In project scheduling research, several objectives have been studied. The most common is minimizing the makespan (C_{max}). Others, in multi-project environments, have developed algorithms to minimize the project tardiness. Some have taken a multi-criterion approach in which the multiple objectives are combined into a weighted objective function (Dawood and Sriprasert 2006). Still others have used overall project cost as the objective function. Cost is a very attractive performance measure in that most other performance measures/objective functions relate to cost, and in practice minimizing cost is the ultimate goal (Dain, Etherington et al. 2005; Dodin and Elimam 2008). It can be very difficult to develop the cost function for problems of this type, however. To simplify the problem, makespan will be used as the performance measure.

1.2.1 Problem Statement

The problem under consideration is the planning and scheduling of work activities and the associated resources so as to minimize the makespan of the outfitting phase of a shipbuilding unit. The problem will be constrained by trade group personnel and spatial resources, in addition to the precedence constraints of the installation activities. This problem is defined as the resource-constrained project scheduling problem with spatial resources (RCPSP-S).

Consider the outfitting problem discussed in the previous section where there is a requirement to outfit a shipbuilding unit, and that unit has a set of known activities, known resource requirements, deterministic and known activity durations, and a known activity precedence network. Interior space requirements are unknown at the outset and must be derived prior to being incorporated into the project scheduling problem. Further, the final due date of the unit is given and is assumed to have come from the high-level planning phase. In this system there are also a limited number of renewable resources. Finite resource pools for each trade group type define the maximum number of resources that can be allocated at a given time in the schedule. Intra-unit space requirements include installation travel paths for resources and materials and the space required to install a component.

The objective of this work is to develop a methodology to model and solve the resource-constrained project scheduling problem with spatial resources to minimize the overall project makespan.

The central thesis is that space must be taken into consideration during the scheduling phase of construction. Spatial conflicts that occur during construction can be used to model congestion and congestion can be used to accurately model the increased time span of activities. The increased activity durations in turn can cause an increase for the entire project duration which models the actual construction process more accurately than traditional RCPSP methods.

1.2.2 Solution Methods Discussion

A review of the literature in this field showed that very little research has been performed for the RCPSP-S. The RCPSP-S is fundamentally a resource-constrained project scheduling problem that is extended to include spatial resources. The RCPSP is known to be NP-hard which indicates that the RCPSP-S is also. Because of this classification, a heuristic procedure will be required to solve the problem. Many types of heuristic or search procedures exist. One such method, the genetic algorithm, has been shown in literature to work well for the RCPSP and is applied to the problem under consideration.

1.3 Dissertation Overview

Chapter 2 presents a review of relevant literature to describe the current state of research in this research area. The methodology will be developed and discussed in Chapter 3. Chapter 4 describes the experimentation section, where an example problem and a case study problem were solved using the developed methodology to characterize the effect of space on the RCPSP. In addition conclusions are drawn from the experimentation and presented. Chapter 5 offers recommendations for future work and describes the general applicability of the methodology.

Chapter 2

Literature Review

The foundation for the problem under consideration, resource-constrained project scheduling problem with spatial resources (RCPSP-S), is grounded in three areas of research: 1) project scheduling problems, 2) space modeling and analysis, and 3) building construction planning and scheduling. The problem described here is an extension of the general resource-constrained project scheduling problem. Current and foundational literature in project scheduling is reviewed and significant relevant results will be described. Space planning and scheduling is a relatively new area of research. There is, however, a body of literature to draw from that mostly comes from the planning and scheduling literature in the building construction industry. Current literature in these fields will be reviewed to establish the current state-of-the-art as it pertains to planning and scheduling in general large-scale product industries (building construction, shipbuilding, oil platform, etc.) that explicitly model and plan for space.

It should be noted that the bin packing problem is another theoretical problem that could be extended to model the underlying problem described here. Its applicability is most promising in the site layout problem, but would require a significant amount of theoretical extension to allow for modeling and solving the internal space allocation problem. This will be discussed further in Section 2.4.

2.1 Project Scheduling Literature

Project scheduling is the process of “ … allocating scarce resources over time to perform a given set of activities” (Weglarz 1998). Project scheduling is a very mature field of research with many models and solution procedures that consider most aspects of the underlying scheduling problem. The general project scheduling problem (PSP) assigns activity start and complete dates so as to minimize project completion time. Adding resource constraints to the PSP is the most common extension to the problem and is known as the resource-constrained project scheduling problem (RCPSP). Numerous other extensions and modifications to the general problem have been proposed and methodologies have been developed to solve them. In order to quickly differentiate and compare problem definitions, a classification scheme was developed and modeled after the $\alpha|\beta|\gamma$-scheme notation used in machine scheduling problems (Brucker, Drexl et al. 1999). In this scheme, α denotes the type of problem, for example, project scheduling (Metaxiotis, Askounis et al. 2002), multi-project (MPS), etc. The β gives the constraint types such as temporal (Temp), precedence (Prec), and the γ denotes the performance measure, for example, cost or makespan. The relevant and applicable extension categories for this thesis include the RCPSP, multiple mode RCPSP, planning and scheduling integration, and to some extent multiple project RCPSP. For a comprehensive review of literature in project scheduling and the numerous extensions refer to (Weglarz 1998; Brucker, Drexl et al. 1999).

2.1.1 Applicable Problems and Current Literature

The resource constrained project scheduling problem is the basis for most work in the project scheduling field. This is due to the fact that there is usually a limit to the number of resources in most industrial project scheduling problems. The general RCPSP denoted (PS/temp/C_{max}) is classified as NP-hard (Słowiński and Węglarz 1989; Dorndorf 2002) and analyzes problems with a single project. The resource-constrained project scheduling problem with spatial resources described here is related to two additional extensions in the project scheduling field: multi-mode problems, and integrated planning and scheduling problems.

Multi-mode problems include various processing options for activities in the project schedule (Dorndorf 2002). This model is used for problems that have alternative processing modes for each activity, which would modify the activity durations based on changing the number of resources or alternative resource group. In many large product manufacturing settings, activity durations are managed by adding or removing resources to stay on schedule. There are obviously limits (min and max) to the number of resources applied to an activity to modify the final duration. And in many cases, addition or subtraction of resources does not represent a linear decrease or increase in activity duration, respectively. Work in this area involves developing the modification function and determining its impact on the final project schedule.

The planning function is an iterative process that can be divided into three phases: strategic, tactical, and operative control (Wei, Nienhuis et al. 2009). The strategic phase occurs early in the planning process where the high-level activities are defined based on the early product design. In the tactical planning phase, the activity networks are verified, resources and detailed durations are assigned to the activities, and capacity planning is performed. The operative control phase is concerned with executing the plan and mitigating any disturbances to that plan (Wei, Nienhuis et al. 2009). Figure 2-1 shows the planning continuum in the shipbuilding industry (Wei, Nienhuis et al. 2009).

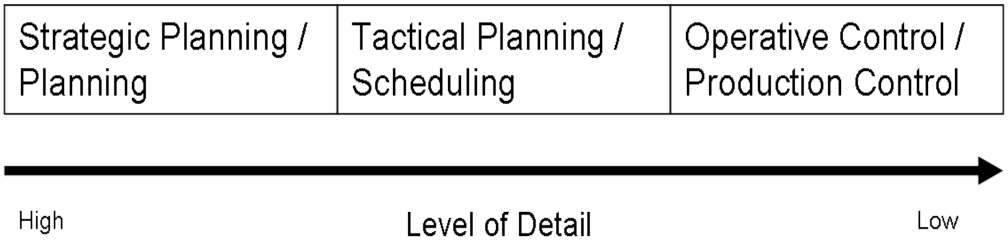


Figure 2-1: Planning and Scheduling Continuum (Wei, Nienhuis et al. 2009).

This general outline of the planning function varies across industries and within specific companies within an industry. In the United States shipbuilding industry and in the remainder of this thesis, the strategic phase is known as planning, the tactical phase is referred to as scheduling, and the operative control phase is known as production control. The focus of the RCPSP-S is in the Operative Control / Production Control area of the continuum where the activity network is defined (predecessor/successor relationships) and resource requirements and durations are assigned to the activities.

The project scheduling problem accepts activities and detailed estimated durations as an input with the objective to allocate resources to those activities in an effort to optimize an objective function. Generation of the activities and durations are performed in the planning phase (Yang, Zhao et al. 2005). Traditionally the planning problem is solved, and then the production scheduling problem is solved (Zhang and Zhang 1999; Yang, Zhao et al. 2005). This process is done iteratively to determine the best set of activities and the best schedule to meet the objective. Integration of the two problems results in a difficult problem that removes the iterative nature of the current practice and significantly increases the complexity of the problem. Several authors have reported research on the integrated problem (Zhang and Zhang 1999; Kolisch 2000; Kolisch and Hess 2000; Zhang, Saravanan et al. 2003; Yang, Zhao et al. 2005; Jain, Jain et al. 2006). Yang, Zhao et al. (2005) extended the basic problem by selecting the resources based on reliability using a fuzzy inference system and solved the problem using particle swarm optimization. Simulation has also been used to solve the integrated problem with reported success (Zhang and Zhang 1999). Kolisch (2000) described an approach to the manufacturing of general large-scale products that modeled the temporal and technological aspects of RCPSP, but also included space and material availability. In his model, spatial constraints were similar to traditional resources, in that there was a pool of work areas which were allocated to the activities similar to other resources.

2.1.2 The Resource-Constrained Project Scheduling Problem

The resource-constrained project scheduling problem is defined as follows. Consider a project that has N activities labeled j=1…N, where activities 1 and N represent dummy activities that mark the beginning and end of the project, respectively, and have zero duration and no resource requirements. The durations of the activities are integer valued, known a priori and denoted by d_i, where $(1 \le i \le N)$. The start and complete times are also assumed to be integer valued and are denoted by s_i, where $(1 \le i \le N)$ and C_i, where $(1 \le i \le N)$, respectively. There are K renewable resource types with r_{ik}, where $(1 \le i \le N, 1 \le k \le K)$ and k is the resource requirement for activity i and

a_k is the availability of resource type k. No preemption of activities is allowed. The mathematical formulation of the RCPSP is as follows (Wuliang and Yonghe 2008):

$$\begin{array}{lll} Min & C_n & \qquad () \\ s.t. & C_1 = 0 & \qquad () \\ & C_j - t_j \geq C_i \quad \forall (i, j) \in E & \qquad () \\ & \sum_{j \in A(t)} r_{jk} \leq a_k \qquad t = 1,2,3 \ldots C_n ;\, k = 1,2,3 \ldots . K & \qquad () \end{array}$$

Where E denotes the set of precedence constrained pairs of activities and $A(t)$ denotes the set of activities in progress in time interval $[t-1, t]$: $A(t) = \{ i \mid C_i - d_i < t \leq C_i \}$. The objective function of minimizing the completion time of the last activity is given in (1). Constraint (2) sets the start time of the first (dummy) activity to zero. Constraint set (3) ensures that the precedence constraints are met. The resource constraints are given in set (4).

2.1.3 Solution Procedures

Numerous methods have been developed for both the unconstrained and resource constrained problem including critical path method (CPM) (Kelley 1963), the Last Planner technique (Choo, Tommelein et al. 1999), mathematical programming, as well as heuristic procedures like Tabu Search, Particle Swarm Optimization and list scheduling (Brucker, Drexl et al. 1999). The remainder of this section will discuss the foundational solution procedures to the PSP and its extension to RCPSP.

The basic PSP is a deterministic model that is not resource constrained, but has precedence and technological constraints. Several methods have been demonstrated for these types of problems: the 1) Critical Path Method (CPM; (Kelley 1963), 2) the Metra-potential Method (MPM; (Hurink and Keuchel 2001), 3) the Critical Chain Method (Goldratt 1997), and 4) the Last Planner Technique (Choo, Tommelein et al. 1999). The CPM, is a common solution procedure in which the activities are scheduled in a window bounded by the earliest start and latest finish (Kelley 1963). The MPM, as described in (Hurink and Keuchel 2001), is a method based on minimum and maximum time lags and has been shown to be NP-hard. The CPM, on the other hand, is a polynomial time algorithm that has been used extensively in the field for a number of years.

The critical path method is a two step method that determines the window for the start time of an activity by finding the earliest start and latest finish times of the activity (Kelley 1963). The entire duration of the activity must fall within that window in order to not impact the other activities in the network. The earliest start of a given activity is determined by constructing a partial schedule with all ready activities and rolling that forward (forward pass) while minding the precedence constraints. The latest finish is determined from the latest possible project completion time and then, similar to the forward pass, the activities are rolled backward (backward pass) while again minding the precedence constraints. From this, the activities that form the longest path from start to finish represent the critical path.

The critical chain methodology, developed in (Goldratt 1997), is essentially the management of buffer time. In large projects, the estimation of activity durations is never an easy task, and most estimation techniques build in buffer time to safeguard against delinquencies. This buffer time is added to the activity duration for each of the elements in the activity. The final activity duration has a significant amount of buffer time and thus has an inflated duration which is inserted into the project network. In the critical chain methodology, the true activity durations are sought and a single time buffer is added to the end of the entire project which is significantly smaller than the cumulative buffers for the individual activities (Goldratt 1997). Operations managers can then budget time based on the total time buffer rather than the individual time estimates of the activities. This method enables planners and managers to manage the buffer time to minimize delinquent work.

The last planner technique is a method of planning that does not allow for the start of work on an activity until all of the requirements for that activity have been met (Choo, Tommelein et al. 1999). This means that an activity will not begin prior to the time when all of the required resources and materials are available for use. This technique generates a short term, usually one week horizon, plan of activities that the planner and/or manager can reasonably expect to finish given the resource and material availabilities (Choo, Tommelein et al. 1999). Within the technique, the plan is made at the beginning of the time horizon and its status is updated at the end to determine performance and correct any assumptions that were incorrect. In addition, this reflective analysis sets the starting point for the next time horizon's plan.

The primary extension to the PSP is the addition of resource constraints to the problem(Brucker, Drexl et al. 1999), which forms the resource-constrained project scheduling problem (RCPSP - $PS_m \mid prec \mid C_{\max}$). This problem is known to be much more difficult to solve and has been shown to be in the NP-hard class of problems (Brucker, Drexl et al. 1999; Wuliang and Yonghe 2008). The NP-hard classification indicates that for realistic problems there is no exact algorithm or solution procedure that can solve it to optimality in a reasonable amount of time (Baker 2001). Therefore, heuristic and other search type procedures have been developed to solve these problems. These methods do not guarantee optimality, but do show significant improvement in solution quality (close to optimal or a theoretical lower bound) and time.

Most problems in this class are formulated as mathematical programs, more specifically mixed integer linear programs (Brucker, Drexl et al. 1999). In their review article, (Brucker, Drexl et al. 1999) found that branch-and-bound techniques have been used to solve small problems and several techniques have been used to increase solution speed and thus increase the solvable problem size. These enhancements are mostly advanced pruning and bounding methods that reduce the number of branches that must be searched to find the optimal solution.

Brucker, Drexl et al. (1999) also discuss the current state-of-the-art in heuristic procedures for RCPSP problems. These procedures include priority rules, truncated branch-and-bound, linear programming based, simulated annealing, tabu search, and genetic algorithm among others. Their analysis shows that these heuristics work well for problems in this class and cite that genetic algorithm and simulated annealing approaches show the best results in terms of solution time and nearness to optimal.

It has been shown that genetic algorithms (GA) produce favorable results in problems with an underlying sequencing structure (Hartmann 1998; Hartmann and Kolisch 2000). Genetic algorithm is a meta-heuristic technique that generates permutation sequences and tests them against a performance measure, makespan in this case. It uses a genetic model to make modifications to sequences to move the search procedure around the decision space and avoid local optima. A GA is used as the underlying search procedure in the solution methodology of the problem described in this thesis because of the positive results found in literature (Hartmann 1998; Hartmann and Kolisch 2000).

Genetic algorithms, like all meta-heuristic techniques, have various parameters that govern the search. The various studies that have used GA have made modifications to these parameters to improve the performance in terms of run time and closeness to optimal. The parameters of a GA can be divided into two categories: structural and operational. Structural parameters are the foundational elements of the search which relate the problem variables to the search variables. The structural parameters are the encoding and decoding schemes. Operational parameters influence how the algorithm performs its search, independent of the relationship to the problem variables. Population size, crossover and selection operators, crossover and mutation rates, and overall number of generations are operational parameters.

There are a number of encoding and decoding schemes that can be used in a GA. The encoding component translates the list of activities into a sequence of numbers that is modified to generate different schedules (Hartmann 1998). Permutation-based schemes permute the list of activities and schedule them according to their position in the sequence. This type of encoding scheme has been shown to be the best for scheduling problems (Hartmann 1998). Decoding schemes are methods to schedule the activities by allocating resources and setting start and end times. The two most prominent schemes are the serial and parallel schedule generation schemes (SGS; Hartmann 1998). The serial SGS selects activities from the precedence feasible list and schedules them at the earliest possible time. This results in the creation of active schedules (schedules where no activity can be left shifted without changing the start times of other activities; Sprecher, Kolisch et al. 1995). The optimal solution to an RCPSP will always be contained within the list of active schedules (Hartmann and Kolisch 2000). Conversely, the parallel SGS method iterates through time periods rather than activity lists. This method creates non-delay schedules (schedules that have no inserted idle time on a resource) as opposed to active schedules (Sprecher, Kolisch et al. 1995). Parallel SGS does not necessarily contain an optimal schedule within its non-delay schedules but contains many fewer alternatives to examine than the serial SGS, which decreases the computational complexity (Hartmann and Kolisch 2000).

Operational parameters are some of the most commonly studied elements in project scheduling literature (Hartmann 1998; Tamaki, Nishino et al. 1999; Li, Man et al. 2000; Zhao and Wu 2000; Hartmann 2002; Kumanan, Jegan Jose et al. 2006; Kim and Ellis 2008). In most cases, a relatively standard setting for these elements suffices for solving problems, however solutions can be found faster and converge more quickly if the operational parameters are tuned. Selection operators, the method by which individuals are selected to move to the next generation, have also shown to be important to the overall GA process (Hartmann 1998; Hartmann and Kolisch 2000). Again, there is a variety of options, but the ranking method has been shown to produce the best results (Hartmann 1998). It has been shown that extending the ranking method with an elitist strategy, where the best performers in a generation are selected to move to the next population, outperforms implementations where it is not used (Kim and Ellis 2008). The method by which children are generated (crossover method) is also another critical parameter in the GA framework (Hartmann 1998; Hartmann and Kolisch 2000). Two-point crossover has been shown to outperform the other available options on general RCPSP problems (Hartmann 1998). The rate parameters, crossover and mutation rates, population size and number of generations are parameters that can be modified to improve the convergence rate, but are not critical to the underlying ability of the algorithm to solve the problem (Hartmann 1998). These parameters can be set using published values and tuned to improve the performance of the GA in the experimentation phase.

The basic structure of a GA takes the following form:

1. Generate a set of precedence feasible sequences equal to the population size.
2. Evaluate each sequence.
3. Perform the crossover operation at the crossover rate.
4. Perform the mutation operation at the mutation rate.
5. Perform the selection operation to generate a new generation.
6. Go to step 2 until the number of generations has been reached.

In this section the solution procedures and techniques for the various general and applicable project scheduling problems were reviewed. Optimal procedures have been developed for most problems in the project scheduling environment, however their applicability is limited by the complexity of the problem. Because of this, many heuristics have been developed that produce near optimal, and often optimal, solutions to these difficult problems in a reasonable amount of computer processing time.

2.2 Review of Planning and Scheduling in the Construction Industry

Large building projects in the construction industry utilize project planning methodologies to help ensure project success through coordination of activities, resources and materials. Large products in the shipbuilding industry are no different, and current planning technologies in both industries are essentially identical with slight differences in the manufacturing processes. This difference is evident in the fact that in the construction industry, the resources and materials are brought to a single fixed work piece location, whereas in the shipbuilding industry large work pieces often move from location to location through the manufacturing process. In general this difference has little impact on the solutions given by the planning and scheduling systems. However, with increased movement, there is a greater need to include space planning into the project scheduling activities to ensure that space is available in the appropriate locations to maximize resource capabilities and minimize delay time. Planning and scheduling in the construction industry uses many of the tools and methodologies described earlier to determine a schedule of activities. In addition, the construction industry has introduced the use of computer aided design (CAD) models into the planning process.

Planning and scheduling in the construction industry can be a very complex and time consuming task because of the thousands of activities, materials, and resources in a typical large building construction project. Similar to solving the basic RCPSP, the usual practice is to divide the construction process into a small set of manageable and attainable high-level activities or milestones. The milestones are then subdivided further into more detailed activities. This is known as the work breakdown structure (Chirillo 1984). In construction type industries, even a high level (major activity level) may contain a significant number of activities that must be scheduled. By increasing the number of activities further, the difficulty of the problem is only amplified. Because of this, solution methods must be able to work on large scale systems in relatively short processing time. Several methods have been developed that are able to manage the large number of activities and resources that are part of a construction project.

Traditional methods are employed to plan and schedule work activities in the construction industry. The critical path method (Kelley 1963), PERT charting (probabilistic models), and more recently the critical chain method (Goldratt 1997) and the last planner technique (Choo, Tommelein et al. 1999) are examples of the traditional methods for planning and scheduling in the construction industry. Details of these methods have been presented in Section 2.1.3 Solution Procedures.

The traditional methods have had great success in the general planning and scheduling in the construction industry (East and Liu 2006). However, like any other difficult and complex problem, research has been done to extend the current methods as well as create new methods to improve the planning process. One extension is the inclusion of visualization in the construction planning process (Heesom and Mahdjoubi 2004). Visualization is the formation of a visible model of a conceptual idea or design. Modern multi-story buildings are being designed using CAD packages. Dimensionally accurate 3D models are generated for the building as a whole and most other design features of the building including rooms, structural elements, mechanical, electrical, plumbing, HVAC, etc., and sometimes the exterior sheathing of the building and intermediate steps in the construction process. These models are used in a variety of functions from generating detailed drawings for construction, to visual walk through simulations of both the end product and work in progress. These visualizations are powerful tools used to communicate design intent to the building developer, owner, and the construction managers alike.

2.3 Review of Space Planning and Scheduling Literature

A thorough examination of the current literature in spatial planning has revealed that the construction industry is the pioneer in local space modeling and analysis (Akinci, Fischer et al. 2002c). Most other general large-scale product industries account for space in their production activities in one way or another, but usually just reduce space to an integer resource and plan it as they would any other resource. Research in the construction industry has formalized spatial definitions and requirements and has proceeded to integrate these models into the planning process. Automatic generation of spatial requirements and explicit planning of those requirements are two of the main research focus areas in this field (Akinci, Fischer et al. 2002c).

Space is a resource. It is different from other resources in that there is not a finite pool from which an amount can be drawn for use in construction. Many researchers have transformed the available space into renewable resource, however this is not a true representation of the spatial requirements or allocation (De Frene, Schatteman et al. 2007). Transforming spatial resources into renewable resources makes it difficult to determine the appropriate grid size for allocation of different sized products. For instance, if the available space is broken into 10’ x 10’ work areas and there is a stack of construction blocks 10’ x 15’ that needs to be stored for some period of time, then it would require 2 - 10’ x 10’ units of space. This allocation would waste 50 square feet (5’ x 10’) that could be used for other purposes. In addition, the difficulty is compounded for installation type activities where the work in place continuously occupies the space from the beginning of the activity until the end of the project, never releasing it for future use (making space a non-renewable resource). Because of this, the shape of the object that requires space must be considered and contiguous space for the duration of its activity must be found and allocated. Space needs to be accounted for as both a renewable and non-renewable resource. Further, spatial requirements for activities are not deterministic and vary based on construction methods (Akinci, Fischer et al. 2002a), and current spatial conditions. Suppose the base footprint of a stack of construction blocks requires 100 square feet and therefore must be located on a work site that is 100 x 100 feet. There is some ambiguity in the actual footprint of the stack of blocks, i.e. the footprint could be 10’ x 10’ or 100’ x 1’. Although the latter is unlikely, it raises the question as to how to deduct the required footprint from the given area. In addition, the original estimate of 100 square feet is not given in any design or production handbook. The space requirement and consequent allocation of the blocks potentially affects everything later in the schedule. Traditionally, these detailed space allocation and planning issues have been performed and reconciled by production supervisors based on experience, without formal tools (Heesom and Mahdjoubi 2004).

In Chapter 1, the notion of general space types was discussed and two main types of space on a construction site were identified. They are: 1) site layout space and 2) local workspace area (Akinci, Fischer et al. 2002c). Spatial planning and scheduling in the construction industry has been examined from both these perspectives (Akinci, Fischer et al. 2002a). The site layout problem is the overall layout of the site in terms of construction area, material storage area, and ingress and egress of material and labor resources. The site layout problem has been defined as the allocation of space in and around the work site which minimizes the spatial interference between activities (Zouein and Tommelein 1999). Local workspace is the space that is needed to complete an activity which would include any required resource space and material space. The local workspace allocation problem is defined as the dynamic space allocation problem (McKendall and Jaramillo 2006).

Site layout planning and scheduling methodologies have been developed and studied extensively in general (Tommelein and Zouein 1993; Zouein and Tommelein 1999; Chau, Anson et al. 2004; Chau, Anson et al. 2005) in the construction industry, (Lee, Lee et al. 1995; Lee and Lee 1996; Lee, Lee et al. 1996; Park, Lee et al. 1996; Varghese and Duck Young 2005) in the shipbuilding industry, and in the general large-scale product industries (Kolisch 2000; Kolisch and Hess 2000; McKendall, Noble et al. 2005; McKendall and Jaramillo 2006). In the construction industry, these plans are relatively static, that is once the overall layout of the site has been established, it rarely changes over the course of the project. In other industries, shipbuilding and aerospace for example, the site layout can be much more dynamic due to the fact that the work pieces move from their build locations upon work completion and new products move into the open space. The most common method to plan and schedule floor space is to use 2-dimensional representations of objects and incorporate time through the integration of a production schedule (Lee and Lee 1996; Finke, Ligetti et al. 2006).

The local space allocation problem arises in general large-product industries in confined spaces such as rooms or within buildings. Examination of this problem has primarily been focused on building construction projects that have multiple floors with similar work content on each floor (Riley 1994; Riley and Tommelein 1996; Riley and Sanvido 1997; Thabet and Beliveau, 1997). The problem has been approached from a less repetitive systemic view which concentrates on developing spatial requirements within that space (Akinci, Fischer et al. 2002a; Akinci, Fischer et al. 2002b; Akinci, Fischer et al. 2002c). The approach in the less systemic view is to identify the spatial interferences and congestion as they occur in the project schedule. Once the spatio-temporal conflicts have been identified, the user manually resolves them by changing either the spatial requirement or the schedule. A deficiency in this line of research is the manual reconciliation, which could be rectified through the use of an algorithm. Other approaches to the problem include a tabu search approach (McKendall and Jaramillo 2006), and simulated annealing (McKendall, Noble et al. 2005), both of which show potential. Both approaches focus on minimizing the travel distance of resources to the work areas by assigning resources to work areas when actively performing an activity and to storage areas when idle (McKendall and Jaramillo 2006).

Two models have been developed for space planning: 1) the construction space model and 2) the planning process model (Riley 1994; Akinci, Fischer et al. 2002a; Akinci, Fischer et al. 2002b; Akinci, Fischer et al. 2002c). The space model describes the various types of space needed for the production of the building and is dependent upon the results of the process model. The process that is used to construct the product dictates the spatial requirements. These models are used to determine the work sequence and material delivery sequences in an effort to minimize the work interferences and overall congestion during the construction project. These two models encompass the entire spatial definitions (both site and local space) required to model and schedule space within the project scheduling framework.

2.3.1 Construction Space

In his doctoral dissertation Riley (1994), discussed the general space types and defined the various space types in a detailed taxonomy. The taxonomy provides a comprehensive definition and thorough description of space as it relates to construction. This section draws significantly from that work to provide the foundation of space in the general large-product industries. Space is divided into areas and paths (Riley 1994). Areas are spaces occupied by activities for a period of time; they include:

- Layout Area
- Unloading Area
- Storage area
- Staging area
- Prefabrication area
- Work area
- Tool and equipment area
- Hazard area
- Protected area

Paths are spaces required for movement of materials, people and other resources, and include three main paths: the material path, the personnel path and the debris path (Riley 1994).

Construction space is categorized into three categories; available space, construction process space, and work in place space. Available space is the space available within the confines of the building or ship unit that changes over time as components are installed and become work in place space. The construction process space is the space required by each activity for the worker(s), machines, and travel to the installation location. As the activity is further decomposed to work elements, this space is also decomposed into spaces that represent much more detail. The work in place space is the space that is occupied by the work pieces and installed items. These spaces and their relationship to one another are detailed in Figure 2-2.

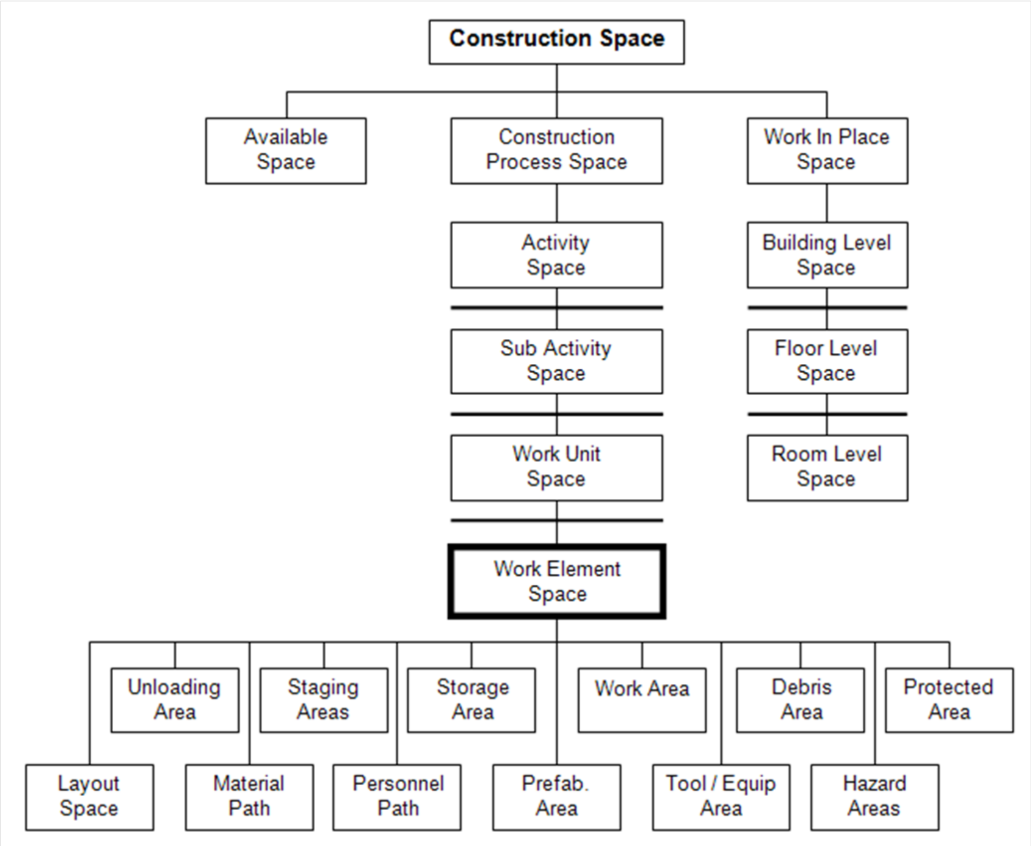


Figure 2-2: Types of Space in the Construction Space Model (Riley 1994).

The space types in the three categories are dependent upon each other in that the total volume of space is comprised of the sum of available space, work in place space and construction process space.

Akinci, Fischer et al. (2002b) have condensed the spaces into three categories: macro-level, micro-level, and paths. They define macro-level spaces as site level spaces that are used for storage, layout, staging, etc. Micro-level spaces are defined as the local spaces required to install the components and the components themselves. Paths are described as the spaces required for travel of labor, materials and debris to and from the worksite.

In addition to the definitions and descriptions of space types, there is the notion of space behaviors in the construction process planning model (Riley 1994). Space behaviors define how activities, craftsmen, materials, work elements and even other space entities interact with the various types of space in the construction worksite. Space behaviors form the foundation for determining the space requirements for a given activity. Defining the space requirements and automatically generating the spatial objects for those requirements are significant components of the problem being discussed in this thesis. Akinci, Fischer et al. (2002a) and Akinci, Fischer et al. (2002b) develop the basic foundation for defining the space requirements for activities and allow for various construction processes. The chosen construction process directly affects the spatial needs of the activity by setting the required resources, resources' work space, work access space, and also impacts hazard space and material and resource moving path spaces. The resources used have different spatial requirements based on physical size of the resource and on the execution space needed to access the work piece.

2.3.2 Planning and Scheduling Space

A thorough foundation for the definition of planning and scheduling space in the construction industry is developed in a construction space planning model (Riley 1994). The construction space planning model is comprised of the construction space model and the space planning process model. The construction space model provides a definition for the various types of space needed and used on a construction site as well as the patterns in which it is required and used. The space planning process model incorporates the construction space model along with the overall construction schedule, material information, design information and project constraints into a formal model of the planning process. The space planning model is then used to define the work sequence, material delivery schedule, and the site layout over time. In addition to this approach, a priority list scheme (De Frene, Schatteman et al. 2007) , a GA procedure (Dawood and Sriprasert 2006), and a knowledge-based expert system have been developed. Commercial project scheduling software packages also contain resource-constrained scheduling methods, however they are usually proprietary and do not have the ability to schedule non-renewable resources.

Performing work activities requires space to store raw materials, operational space for tradesman and equipment, and the unit workface. Several trade groups are required in the construction of large building projects. Poor planning and work management can lead to work congestion. Thomas, Riley et al. (2006) defined work congestion, identified the causes, and presented a case study to show the impact of work congestion on productivity. In addition, the authors discussed techniques to avoid work congestion on both macro- and micro-levels. They cited industry standards for nominal required space and provided empirical evidence that these standards may not be relevant for some types of activities.

A priority list heuristic method has been developed to solve spatial resource-constrained project scheduling problems (De Frene, Schatteman et al. 2007). In this model, spatial resources are resources that are required by a set of activities for an extended period of time and cannot be used by other activities during that time which allows them to describe space as a nonrenewable resource. De Frene, Schatteman et al.

(2007) grouped activities and reserved spatial resources for a period of time. Using priority lists, the procedure then constructed a feasible schedule by adding activities to the current schedule.

A genetic algorithm-based solution procedure, defined as a multi-constraint approach that considers physical, contractual, resource and information constraints in a multi-objective format has also been developed (Dawood and Sriprasert 2006). Using a linear weighted objective function, users have the ability to find the Pareto frontier of possible project plans as they give higher weights to the four constraint types in the objective function. In this approach, resources (including space) are incorporated and assigned during schedule construction. This is interesting in that most, if not all, of the other work being done in this area generates a spatial resource unconstrained project schedule and then uses some form of collision detection to identify spatial conflicts. The schedules are then altered based on the identified conflicts, in a trial and error manner. Although space is considered during the scheduling phase, space has been transformed into a renewable resource by defining the space availability and requirements in such a way that they draw from a pool of capacity. This dramatically reduces the complexity of the problem and facilitates the insertion of space allocation in the scheduling phase.

Thabet and Beliveau (1997) developed a knowledge-based expert system to solve the space-constrained and resource-constrained scheduling problem in multistory building projects. Their approach used a parallel schedule generation scheme (SGS) that created a resource and space feasible schedule. The parallel SGS ranked the activities based on simple dispatching rules and scheduled the activities according to the ranking sequence. Spatial representation of the activities was considered when allocating space within manually generated user pre-defined space blocks that represented the workspace. The congestion within the space blocks was modeled using a space criticality factor (SCF) that measured the congestion within the work space block. The SCF was used to determine performance decreases for the work being performed within the workspace and the durations of the activities were modified according to a pre-defined SCF curve. The level of scheduling detail used in the work of Thabet and Beliveau (1997) was on a much larger scale than the method developed in this thesis, for example, scheduling activities for the entire building versus scheduling activities for a room. In addition, only one activity related space type was considered which does not account for the several types of required activity spaces. Further, the SCF curve proposed by the authors was an estimate of the productivity decrease based on only a single activity type. This differs from the work presented in this thesis in that multiple activity space types are used to model the congestion within a workspace that is defined by the boundaries of the workspace, not a predefined block of space. Furthermore, the congestion function developed in this thesis is a composite function that models the effect of spatial conflicts between each of the multiple activity space types.

Resource allocation is a critical component in construction planning and execution. Traditional scheduling software packages (MS Project, Primavera, Artemis, etc.) are well suited to allocate and manage traditional resources and even space when it is transformed into a renewable resource, but are not equipped to handle all spatial resources. These commercial planning packages are in use at most organizations that have a need to plan large projects. The algorithms used by these packages are either proprietary or are similar if not the same as many of the solution procedures discussed in this chapter. Because these packages were designed and built for traditional project planning functions, many practitioners are capitalizing on their strengths and combining them with the increased visualization in the industry to incorporate space into the planning process. The combination of the two concepts into a single planning tool is known as 4-dimensional (4D) modeling.

2.3.3 4D Modeling

Researchers in the building construction industry have begun to develop a solution to the complex task of integrating spatial constraints into the planning phase (Adjei-Kumi and Retik 1997; Webb, Smallwood et al. 2004). This work couples computer aided design (CAD) information with schedule information to form a 4D (3D CAD + time) model (Adjei-Kumi and Retik 1997; Webb, Smallwood et al. 2004). These models enable users to identify the construction interferences and generate feasible work sequences. One drawback to this approach is that in simple 4D modeling techniques, only interferences related to final geometry are considered, not interferences related to temporary space needs such as worker spaces or support bracing. However, in large construction projects, the work sequence is linked both to the design (McKinney and Fischer 1998; Webb, Smallwood et al. 2004) and the space requirements needed for construction. Because of this, there is a need for a tool/methodology that includes all space requirements into the original scheduling task and not as a supplement to that process.

The 4D models are used by construction planners and managers to convey construction progress (Adjei-Kumi and Retik 1997; Dawood and Mallasi 2006; Horman, Orosz et al. 2006) and improve construction plans in terms of cost, schedule and quality (Koo and Fischer 2000; Jaafari, Manivong et al. 2001; Chau, Anson et al. 2003; Chau, Anson et al. 2004; Wang, Zhang et al. 2004; Tan, Messner et al. 2005). This is achieved by allowing various trade groups from superintendents to design engineers to view the schedule over time and identify space-time conflicts, critical sequence issues, and areas for schedule and design improvements. As this area of research has matured, the number of applications of the technology has also increased. The development of the 4D model from the construction schedule and geometry is discussed in (Hastings, Kibiloski et al. 2003). In addition, web based applications are being used to view construction schedules in four dimensions (Kang, Anderson et al. 2007), and have empirically shown improvement in detection and resolution of problems.

Construction planning using 4D modeling began by visualizing the construction progress through the use of the design CAD models (Adjei-Kumi and Retik 1997). This approach has traditionally been used to show building owners and developers the progress of the project at various points in the construction process (Chau, Anson et al. 2004). In this capacity, the model is considered a simple visualization and very little if any analytical value was added to the construction plan. These models are created during the construction planning phase of a project and are usually not updated once work has begun (Chau, Anson et al. 2004).

Although the visualization of the construction progress is powerful, the integration of the construction schedule with advanced project planning software significantly increased the value of the 4D models. Advanced project planning systems are very good at planning and allocating regular resources and by integrating them with 3D product models, space can be considered more easily in the construction schedule. This enables planners to visually identify spatial conflicts and modify the schedule to resolve those conflicts. Often the sequence of macro-level construction activities (structural, electrical, ventilation, etc.) can be altered in the schedule to remedy spatial conflicts (Horman, Orosz et al. 2006). Alternatively, a four dimensional technique has been developed that capitalizes on the visual aspects of CAD models and the time domain of Gantt charts to help construct better schedules (Dawood and Mallasi 2006).

Macro-level construction sequencing is the first step toward construction schedule optimization. 4D modeling provides a medium for planners and construction supervisors to visually and analytically evaluate construction schedules. The current state of the art in 4D modeling allows manual improvement through trial and error of various construction scenarios and sequences (Heesom and Mahdjoubi 2004). This extends macro-level sequencing by adding additional levels of detail into the model that represent micro-level (local work area planning, work area congestion analysis, etc.) planning. It is obvious that by adding additional levels of detail (macro to micro detail) additional information is needed. This information includes geometric representations of workspace, detailed 3D models of work activities and components, and the link between activity spaces and component geometry. Generation of the additional data in most cases prohibits planners from developing plans to this level.

One of the drawbacks of 4D construction modeling is the amount of work that is required to generate geometry for work space requirements (Akinci, Fischer et al. 2002a).

Adding to the difficulty is that alternative construction processes or sequences may have different space needs (Akinci, Fischer et al. 2002b). It has been noted that users need to be able to generate these models quickly and at various levels of detail so that they can evaluate several alternatives (Koo and Fischer 2000). Once the geometry has been created, the planner associates that space to a construction object for representation in the model. An automated method to generate micro-level work space geometries and the subsequent association to work unit objects has been developed (Akinci, Fischer et al. 2002a). The authors described a geometry generation tool based on a 3D bounding box approach and a knowledge base of construction practices to link the work space with the construction unit space. This link was established both geometrically and as part of the schedule. Based on this work, alternative build processes can be modeled so that the planner can evaluate each process visually and in Gantt chart form to choose the best approach in terms of project makespan. Likewise, an alternative model has been developed that uses a knowledge-based approach to link CAD geometry with activities (Adjei-Kumi and Retik 1997) to ease the burden for experienced planners and construction managers in the planning phase.

2.3.4 Congestion Modeling

Congestion is the measure of how much available work space is consumed by the spatial requirements of the activities being performed in the workspace at a given time (Mallasi 2006). Congestion modeling has taken several forms in previous research and several approaches have been developed to model congestion in an effort to reduce the productivity losses attributed to highly congested work areas.

One approach is a space-time conflict analysis that identifies conflicts between activity spaces and relates them to productivity losses for a given schedule in a 4D environment (Akinci and Fischer 1998). The space-time conflict or congestion is a function of the types of space involved in the conflict. In their approach, Akinci and Fischer (1998) developed a taxonomy of conflict types that were used in the analysis. The space-time conflict analysis was performed on a 4D model and the results of the analysis were presented to experienced planners and construction managers to either resolve the spatial conflicts or to modify the schedule to reduce the impact of the spatial conflicts.

Another approach, the critical space-time analysis method, has been developed that extends the space-time conflict analysis by developing a ranking of the conflict types in order of impact to the schedule (Mallasi 2006). Using this approach, congestion is modeled as a linear weighted multi-criteria function that includes: the ratio of conflicting space volumes, the total number of clashes, the total number of conflicting activities, the total number of conflicting space types, and a criterion function for the conflicting critical activities. This development provides more information to the planner or construction supervisor when making changes to the schedule or space allocation strategy.

An alternative critical space analysis approach is given by Winch and North (2006). In this work, congestion was modeled as the ratio of required space to available space. As the ratio approached unity, the congestion within the work area became critical. Spatial clashes were resolved manually by planners and construction managers.

In yet another approach to modeling congestion, a space capacity factor combined with user defined productivity loss curves is developed as part of a spatial scheduling procedure developed for repetitive work on multistory building construction (Thabet and Beliveau 1997). In this approach, work areas were modeled as blocks and activities are assigned to the blocks. Activity spaces were defined a priori and did not change over the project duration. The space capacity factor measured the congestion within the work space blocks and was used to make scheduling decisions in the knowledge-based system scheduling approach.

Guo (2002) presented a two-component method to model congestion. One component models the interference space percentage and the other considered the time that the spaces were in conflict. Using these two measures of congestion, additional information is available for the conflict resolution system to mitigate the effect of the spatial conflicts.

The congestion modeling approaches in literature all try to capture the spatial loading of a work space. Some of the approaches have enhanced the basic definition by adding the time domain and others have characterized the space types and conflict types in an analysis approach. The goal of each of the approaches is to provide a planner or scheduling procedure with information about the spatial loading of the work space so that the conflicts can be reduced or eliminated.

2.3.5 Review of Current 4D Planning and Scheduling Tools

The need for tools and methods in the area of 4D planning and scheduling is well established (Adjei-Kumi and Retik 1997; Akinci and Fischer 2000a; Akinci, Fischer et al. 2002b; Akinci, Fischer et al. 2002c). In response to this need, several commercially available tools have been developed from the reviewed research. Heesom and Mahdjoubi (2004) presented, discussed, and compared six tools that represented the state of the art in construction industry 4D modeling at the time. The following provides a brief table that illustrates the highlights of these six tools plus three more that have been published since the discussion published by Heesom and Mahdjoubi. Supplementing this review is Issa, Flood et al. (2003) whose work adds to the discussion for the current tools and systems that are commercially available to construction and large-scale product manufacturers. This brief table does not include the entire population of 4D planning tools. It is included to highlight the types of tools and the range of functionality that is currently available.

The six tools/applications described in (Heesom and Mahdjoubi 2004) include: *Schedule Simulator* (Bentley Systems), *Smart Plant Review* (Intergraph Inc.), *Project Navigator 2000* (VirtualSTEP), *FourDviz* (BALFOUR Technologies LLC/Infinity Technologies), *Common Point 4D* (Centre for Integrated Faciltiy Engineering, Stanford), and *Visual Project Scheduler*. In addition the authors also mention several research projects and prototype systems in development at various research institutions and centers. The relevant tools and projects are summarized in Table 2-1.

Table 2-1: Commercial Software Comparison

Tool	Schedule Information	Product Model	Linking Product to Process	Collision Detection	Collision Resolution
Bentley Navigator	MS Project, Primavera	Microstation	Manual	Auto	Manual
Smart Plant Review	Primavera	VRML	Manual	Auto	Manual
Project Navigator	MS Project, Primavera	VRML	Manual	Auto	Manual
FourDviz	Manual	DXF	Manual	None	Manual
Common Point 4D	MS Project, Primavera	AutoCAD, VRML	Manual	None	Manual
Visual Project Scheduler	MS Project, Primavera, SureTank	DXF	Manual	None	Manual
MovePlan / Move Schedule	MS Project, Primavera	Manual	Manual	Auto	Manual
4D Work Planner	4D Model	4D Model	Auto	Auto	Automated (not fully described)
VIRCON	MS Project	DXF, VRML	Auto	Auto	Semi-automated
AutoDesk Navisworks	MS Project, Primavera	Navisworks, AutoCAD, MicroStation, DWF, IGES, STEP, STL, VRML	Manual, Semi-automated	Auto	Manual
VICO Constructor	MS Project, Primavera	Revit, Tekla, ArchiCAD, CAD-Duct	Auto	Auto	Manual

There are three functions of 4D CAD tools: 1) schedule visualization, 2) schedule generation, and 3) schedule analysis and modification (Heesom and Mahdjoubi 2004). All of the 4D tools visualize the schedule, but for some of them, this is their only function. Schedule generation tools do not begin with a schedule from a planning system, but may accept activities with the idea that the tool enables the user to create a space feasible schedule. Tools in the schedule analysis and modification category provide significant analysis tools to the user for collision detection, easy modification of the project schedule, and support for collision resolution, all of which are additional to the animation of the initial project schedule.

2.4 The Bin Packing Problem

The bin packing problem can be described as filling a bin with cubes in such a way that it minimizes some performance measure, usually height of blocks in the bin (Dyckhoff 1990; Bischoff, Janetz et al. 1995; Lim, Rodrigues et al. 2005; Miyazawa and Wakabayashi 2007; Raj and Srivastava 2007; Westerlund, Papageorgiou et al. 2007). The general spatial scheduling problem can be modeled as a bin packing problem with the activities being the blocks and the bin the physical bounds of the spatial area . The length and width of the block would be represented by the installation piece length and width. The height of the block would be represented by the duration of the activity. This results in a 3-dimensional bin packing problem. An additional dimension could be added for the fourth dimension, that is, 3D installation piece + time.

The structure of the bin packing problem lends itself reasonably well to the site layout problem where installation pieces have limited precedence relationships and the objective is to determine where to allocate the space. In the micro-level space planning problem, the final location of the installed pieces is set and the objective is to determine when to perform the activity. In bin packing terms, the former problem decision space is in the x-y plane of the bin while the bulk of the latter is in the x-z plane, assuming a 3D implementation. The remainder of the micro level planning space is undetermined a priori and is generated as a result of the installation order.

Because the focus of this thesis is the micro-level space planning problem and the extensions to the bin packing problem that would be required (4-dimensional, precedence constraints, varying block sizes, and sequence dependent block sizes), the bin packing problem will not be used to model the underlying research problem.

2.5 Chapter Summary

This chapter summarized the current research in three areas: 1) project scheduling problems and solution methods, 2) space modeling and analysis, and 3) building construction planning and scheduling. These three areas form the theoretical foundation for the problem that is under consideration here in terms of definition, modeling approaches and solution procedures. The following chapter will draw upon this foundation to form a model and subsequent solution procedure for the resource constrained project scheduling problem with spatial constraints.

Chapter 3

Methodology

This chapter defines a solution methodology framework for the resource-constrained project scheduling problem with spatial resources (RCPSP-S). Specific components of the framework are defined and described in detail. The RCPSP-S problem arises in general large-scale product industries where space is a critical resource that should be taken into consideration when scheduling assembly and component installation activities. The framework describes a general method to obtain a minimum makespan schedule for a large-scale installation project subject to resource and spatial constraints. The method uses estimates of activity duration, space requirement estimates, and geometric representations of the work space as inputs. The primary research contribution is the definition of the RCPSP-S solution framework that integrates several algorithms and methods into a single comprehensive solution methodology. Specific algorithms for each of the components in the framework will be identified and the resulting methodology will be demonstrated by application to an installation project in the shipbuilding industry.

3.1 Introduction

Project planning in general large-scale product industries is comprised of several levels of detail all of which interact to form the final production schedule. High-level strategic plans are developed and used to generate lower-level capacity plans. Capacity plans develop a rough cut resource allocation and work location. The capacity plans are used to create detailed production schedules that are executed by the shop floor and monitored and updated by the production control function. Spatial concerns are rarely included in any level of the planning / scheduling continuum. The space planning problem can be divided into two levels: a macro-level and a micro-level.

The macro-level is concerned with site layout and material movements. The purpose of the macro-level is to establish the spatial environment by gathering the high-level schedule (unit start and due dates) for all of the units (large assemblies) within the planning horizon. Site planning should be performed in the capacity planning function. It should be noted that site layout planning is not a trivial problem and is the focus of much research, but is out of the scope of the problem under consideration.

The micro-level is associated with detailed scheduling of the assembly and installation activities within the units. The focus of the work in this thesis is intra-unit project scheduling of space and resources at the micro-level. Scheduling the micro-level activities with their spatial requirements can only be done in the production schedule generation level. At this level, detailed resource allocations are performed and start and end dates are assigned to the activities. This detailed information is needed to generate the spatial requirements for the activities.

The remaining sections of this chapter discuss the approach used to model the RCPSP-S. Section 3.2 presents an example that motivated the research and Section 3.3 gives a problem statement, states the objectives and documents the assumptions. Section 3.4 describes the integrated solution methodology framework and specific supporting algorithms used to solve the RCPSP-S. The spatial requirements and behaviors are presented in Section 3.5 and Section 3.6 describes the heuristic search algorithm used to generate sequences for the schedule generation algorithm, which is presented in Section 3.7. Section 3.8 is a summary of the chapter.

3.2 Motivating example

Planning and scheduling in the shipbuilding industry is difficult. The products are often complex and are comprised of many thousands of piece parts. As those piece parts are assembled, the work in process can become quite large. In the shipbuilding industry, the piece parts are joined to form assemblies, and the assemblies are joined to form modules or shipbuilding units. These units are moved from place to place in the manufacturing facility to maximize the use of resources located in certain areas. The units have a substantial footprint or space requirement that must be accounted for in areas such as a building that has a limited amount of space available for placement of the units. Planning the space for the units within a building is an example of the macro-level planning.

Micro-level scheduling is concerned with the scheduling of activities that occur within the units. Unit assemblies are initially empty ship spaces, roughly the size of a room in a house, that are defined by walls or bulkheads. Workers install many components within the spaces in the outfitting phase of construction. Pipes, lights, shelves, hangers, and valves are all examples of components that are installed in the outfitting phase. In some cases, there are precedence constraints for the installation of the components. For example, a pipe hanger is installed prior to the pipe that it supports. In most cases, the plans that come from the capacity planning function do not give strict start and end dates for the activities. The activities are given a range of time in which to be performed. It is left for the supervisor to determine the schedule and allocate resources (primarily labor) to the activities. Each activity requires at least one resource for execution and the ship space can become quite congested when all of the workers and material are located within the confined space. The congestion results in unplanned delays which may extend the completion time of the unit or require overtime to correct.

3.3 Problem Statement, Objectives and Assumptions

The RCPSP-S extends the classical RCPSP model to include spatial requirements. As previously mentioned, each activity will require additional space for laydown, travel and execution. The modified problem takes the same form as in the classical RCPSP, however as space is added, the durations of the activities are allowed to increase. Specifically, the RCPSP-S is a project scheduling problem where components are installed into a shipbuilding unit. Activities represent installation tasks; each activity is associated with a CAD solid model that describes the geometry of the component to be installed and specifies the final location and orientation. Activities have known durations, precedence relationships, and resource requirements. In addition, they have work space requirements that depend on the component geometry. Activities compete for limited space within the unit, and space conflicts result in increased activity duration.

The objectives of this work are: 1) to develop an approach for representing project delays caused by space conflicts and congestion, and 2) to develop a methodology which produces a project schedule with near-minimal makespan.

Several assumptions have been made to define the bounds of the problem under consideration. The following list gives the assumptions:

1. The availability of the renewable resources remains constant throughout the project duration.

   There are four resource types included in this approach. Each type has a fixed number of members and is held constant throughout the duration of the project. These resources are considered renewable resources.

2. Activity space is a doubly-constrained resource.

   Space is both a renewable and non-renewable resource. Spaces such as laydown, travel path and a portion of the execution space are used for the duration of the activity and then released at the completion of the task. Work in place space and the portion of the execution space that contains the work in place is consumed by the activity and is not released for future use.

3. Workspace requirements for an activity can be derived from the component geometry and the number of resources required for the task.

   Activities require space to perform the work in the task. The spatial requirements are generated as functions of the component geometry as described in the space definitions in Section 3.5. Additional space requirements are derived from the number of resources required to perform the activity and result in a space envelope for a person(s) to perform the work.

4. A space conflict between two activities results in a delay for the second activity to be scheduled.

   The scheduling approach uses a serial schedule generation scheme, which schedules a single (current) activity at a time according to a sequence generated by the genetic algorithm. It is assumed that if the space of the current activity conflicts with space from a previously scheduled activity, the current activity duration will be increased proportional to the conflict volume, while the other previously scheduled activity durations remain unchanged.

5. The activity network does not allow for work in place to absolutely block out installation of future components.

   It is possible for the installation of a component to restrict the installation of another component. The assumption is that the activity network does not allow blocking of the component installation for every possible sequence of activities, i.e. at least one sequence will be free from blockage. Some sequences evaluated may encounter a blockage and are heavily penalized to reduce the occurrence of the sequence or portion of the sequence in future generations of the search.

3.4 Overview of Methodology

The developed methodology is an integrated collection of algorithms and methods that have been linked together to form a solution methodology framework for the RCPSP-S. Some of the algorithms have been adapted from published research and others have been developed specifically for this thesis. The following section describes the developed framework at a high level.

3.4.1 RCPSP-S Algorithm Framework

The methodology developed in this thesis is defined by a framework of interconnected algorithms and methods that accept a network of activities and generates a near minimum makespan, resource-constrained schedule that considers the spatial requirements of the activities. The framework draws from several areas of research: resource-constrained project scheduling, heuristic search procedures, activity space generation, and congestion modeling. Figure 3-1 shows the flowchart of the main algorithm components that form the framework.

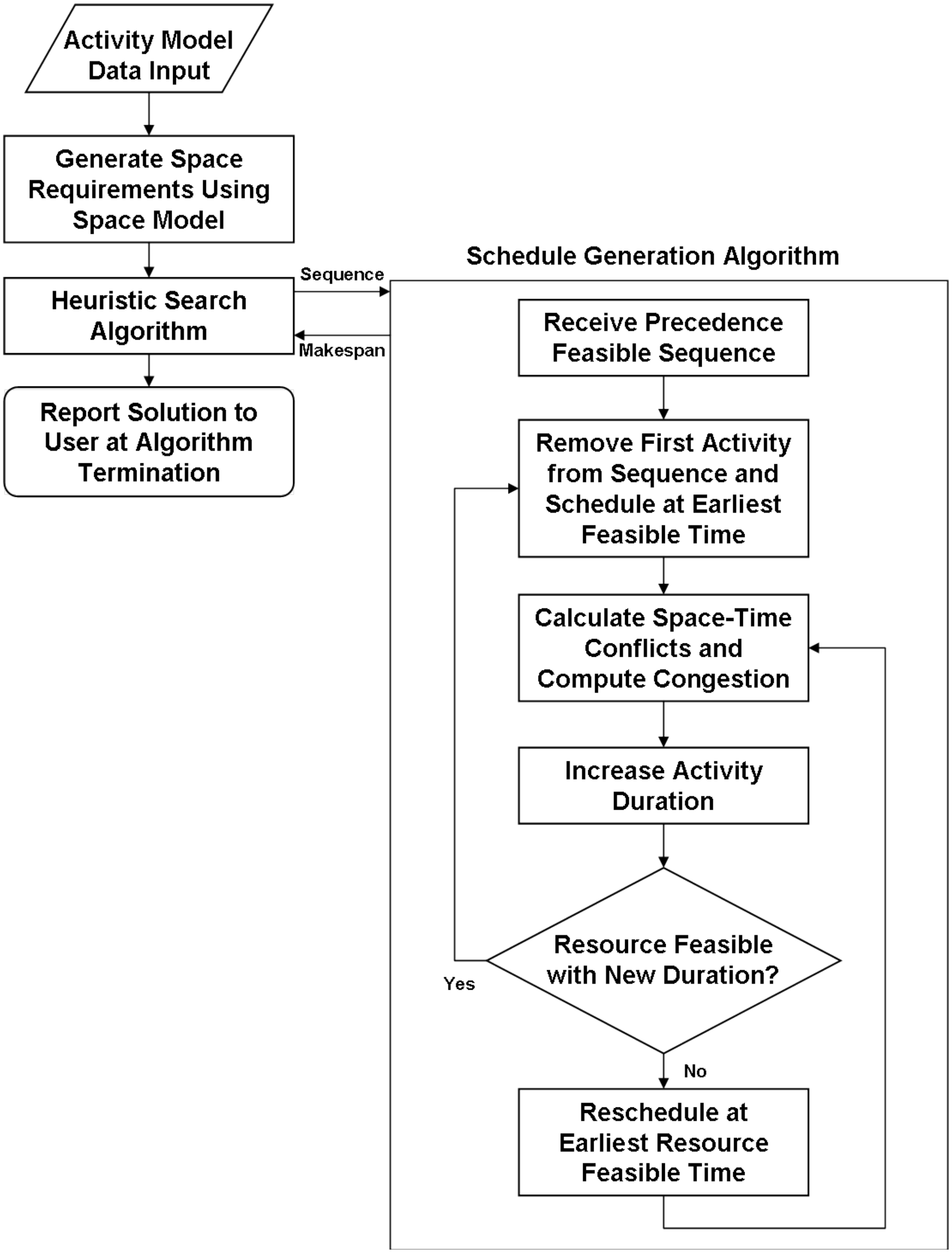


Figure 3-1: Framework Flowchart.

The framework describes the relationships between each of the algorithms used to solve the resource-constrained project scheduling problem with spatial constraints. The activities in the precedence network follow an activity model and are used to generate the spatial requirements defined in a space model. A heuristic search algorithm combined with a schedule generation algorithm iteratively searches through several possible schedules in pursuit of the minimum makespan schedule. The schedule generation algorithm interfaces with a congestion function to determine the impact of spatial conflicts on the duration of the activities. The heuristic search algorithm searches until the stopping criteria for the algorithm have been met and the schedule with the minimum makespan found is reported to the user as the solution.

3.4.1.1 Activity Model

Project scheduling is the term used to define the problem of assigning start and stop dates and resources to activities so as to optimize a performance measure such as makespan. The project scheduling problem begins with a network of activities that is defined by predecessor relationships. Each activity is also defined by a set of parameters that can be referred to as the *Activity Model*. The *Activity Model* is as follows:

- ID
- Type
- Required Resources
- Predecessors and Successors Lists
- Duration
- Assigned Resources
- Start Date
- End Date

The ID is the activity identifier that is used to differentiate it from other activities in the network. The activity Type describes the type of work the activity entails. Each activity also requires a number of resources. The Required Resources attribute defines both the number and type of resources that are required to complete the work. The activity network defines the predecessor relationships and this information is passed to the activity in the Predecessor and Successor Lists. The Duration attribute defines the nominal duration of the activity without any spatial interferences or resource availability constraints. The Assigned Resources attribute is designated as a result of performing the scheduling algorithm. This attribute contains the resource(s) that has been selected by the scheduling algorithm to fulfill the requirement. The Start and End Date attributes are also assigned by the scheduling algorithm and represent the exact start and stop dates of the activity.

3.4.1.2 Space Model

In addition to the information included in the *Activity Model*, each activity also requires space. The space requirements for the activities are generated using CAD data for each of the components. The CAD data is a detailed three-dimensional (3D) product model that includes detailed geometric information about where to place the installed components.

The initial step in accounting for space is to link the geometry to the activities. Linking the geometry to the activities is usually a manual process (Akinci and Fischer 2000; Akinci, Fischer, et al. 2002a; Akinci, Fischer, et al. 2002b), however in some industries designers are working with planners to attribute the design elements with process information (Cahill 2006). When models are attributed with process information, the process of linking the geometry to an activity is simplified. The simplest activity attribution method is to identify the component name in the activity name.

In addition to the geometry of the components, each activity has support space requirements. The generation of support spaces and workspace geometry is the process of enhancing or adding to the space that the activity geometry represents. This is accomplished by adding extra space around the work pieces for execution space and additional space for support functions. Support spaces are areas within the unit that are used for travel of labor and materials, temporary storage areas for material, hazard areas, etc. A portion of the *Types of Space in the Construction Space Model* (Riley 1994) figure from Chapter 2 is given in Figure 3-2 and shows the support spaces. All of the boxes under the Work Element Space box are considered support spaces, with the exception of the Work Area. Work Area space is considered execution space.

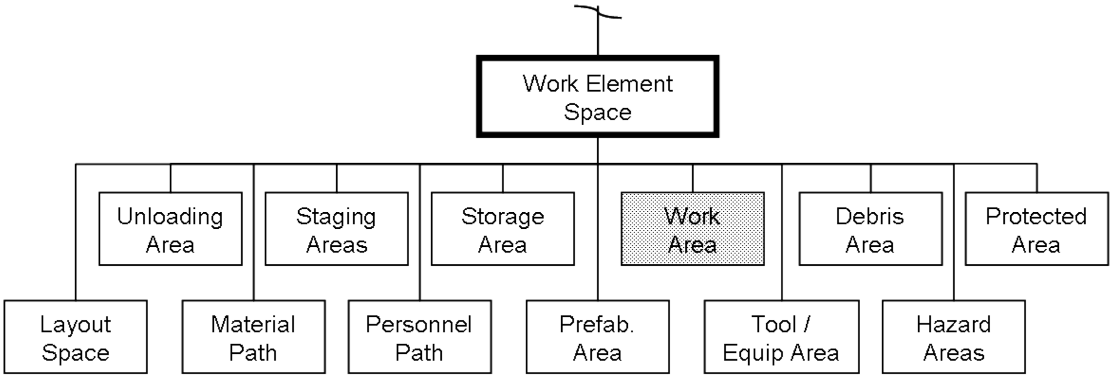


Figure 3-2: Support Spaces and Workspace Geometry (Riley 1994).

The spaces identified in Figure 3-2 represent all of the possible spaces for any general large-product industry. Specific scheduling problem instances in any industry may only include a subset of these activities to fully define the spatial requirements. In any specific implementation of spatial modeling, each space type should be evaluated to determine if it applies to the problem. If a particular space is identified as being applicable to the problem under consideration, then a method to generate that space for each activity or group of activities should be implemented. Space generation algorithms are the focus of current research in the building construction industry (Akinci and Fischer 2000; Akinci, Fischer, et al. 2002a; Akinci, Fischer, et al. 2002b).

The inputs to the additional space generation components are a list of activities that have the geometry of the installation component linked to them. The space generation algorithm uses these inputs to form a detailed space requirement definition, which is termed the *Space Model*.

Space Model

- Geometry of Available Space Within Unit
- Execution Space (Work Piece and Working Envelope)
- Support Space Definitions and Requirements

3.4.1.3 Heuristic Search Algorithm

A heuristic search algorithm is used to generate feasible activity sequences to be evaluated by the schedule generation algorithm. Heuristic search algorithms such as simulated annealing, tabu search, and genetic algorithm follow an iterative ‘generate and test’ process. In the context of scheduling, the algorithm iteratively generates a set of precedence feasible sequences to be tested or evaluated. The search is guided to an objective by using information from one iteration to modify the decision variables in the next iteration. In project scheduling problems with makespan as the performance metric, the objective is to find the schedule with the minimum makespan.

3.4.1.4 Schedule Generation Algorithm

Evaluation of a sequence entails generating a resource-constrained schedule that accounts for space through the use of a congestion function. The classical RCPSP without considering space is a very mature area of research. The two predominate schedule generation schemes are serial and parallel methods (Hartmann and Kolisch 2000). These concepts have been discussed previously in Chapter 2.

Although the classical RCPSP is a mature area of research, the inclusion of spatial resources in the methods has not been explored in great detail in published literature. One approach to including space in a schedule generation scheme is to employ a congestion function to model the effect of space requirement conflicts on the duration of the activities. Congestion modeling is also a current area of research (Akinci and Fischer 2000; Akinci, Fischer, et al. 2002c; Mallasi 2006; Tabet and Belieau 1997) that has primarily focused on correcting predefined schedules using the congestion function to identify heavily congested areas/times. In the framework described here, congestion modeling approaches are integrated with project scheduling methods to quantify the impact of spatial conflicts on the durations of the activities.

3.4.1.5 Discussion and Research Contribution

For each algorithmic component in the framework (Space Generation, Heuristic Search Algorithm, and Schedule Generation Algorithm), there may be several algorithms available for implementation. For example, the Search Algorithm component could be represented by tabu search, simulated annealing or genetic algorithm. The framework was specifically designed in this manner to enable future research to evaluate alternative algorithms or enhance individual components separately from the overall solution methodology.

The framework developed in this thesis represents a contribution to the current body of research in the space scheduling field. Each of the algorithmic components in the framework have been studied in isolation or in pairs, however integration of the various algorithmic components into a comprehensive algorithm framework has not been reported in current literature.

3.4.2 Framework Component Algorithms

The developed methodology uses the RCPSP-S algorithm framework and defines specific algorithms for each component.

3.4.2.1 Project Scheduling: Activity Definitions and Relationships

Activities are defined by a higher level planning system by defining the specific attribute values of the *Activity Model*. The attributes that are determined by the RCPSP-S algorithm are left blank at the outset and filled during execution of the algorithm.

3.4.2.2 Space: Types and Modeling Approach

Expert knowledge of the product and process should be used to determine the additional space requirements for support spaces and execution spaces. Often, this knowledge is gained from the supervisor, who must identify, define, and allocate support spaces for the period of scheduling performance (Akinci and Fischer 2000; Akinci, Fischer et al. 2002a). The space generation mechanism used here, loosely follows the automated procedure found in literature (Akinci, Fischer et al. 2002a), in which the authors discuss a procedure to develop the additional spaces based on a generic work space ontology and a project specific 4D production model.

Applicable support space types for a given problem are dependent upon the characteristics of the industry, activities and other problem specific details. In the motivating example two support space types appear to be the most applicable for the problem under consideration. These support spaces are: laydown and travel path spaces. Other support spaces could be considered applicable to this problem, however, these two spaces were chosen as a representative sample to demonstrate the RCPSP-S algorithm. These two support spaces represent the smallest subset of spaces that enable the component to enter the unit and move to its final installation location.

The laydown space is the space required within the unit, just inside the unit entryway, used to transition from one material handling device to another. For example, a crane may be used to move the component to the entryway and a hand truck is used to move the component from the entry point to the final installation location.

The travel path space is the space required by the component to move from the entry point to the final installation location. In the developed methodology, a nested approach is used to define the travel path. This approach approximates the travel path space and ensures that a feasible travel path exists for the component.

Execution space is present in all local spatial scheduling problems. The generation of the execution space loosely follows that found in Akinci, Fischer et al. (2002a). Execution space is defined as the space required by the activity to install the component once it is placed at its installation location. This includes the space required by the worker(s) and a nominal working envelope around the component for reach and access.

3.4.2.3 Heuristic Search Algorithm

A genetic algorithm (GA) is used for the Heuristic Search Algorithm component of the framework. The GA method is derived from the permutation-based genetic algorithm method presented in (Hartmann 1998; Hartmann and Kolisch 2000), where the authors show that a priority-based encoding scheme and serial schedule generation scheme outperform the other encoding and decoding schemes.

The GA is primarily used to generate sequences to be evaluated by the schedule generation algorithm. Genetic algorithm is a heuristic search procedure that is modeled after the evolutionary process. The algorithm creates sets of sequences to be evaluated by the schedule generation algorithm. The sets of sequences are termed generations. Each generation is derived from information about schedule makespan from the previous generation. Heuristic procedures are defined as algorithms that use information from one generation to develop the next as the search moves toward an objective.

A GA uses a set of predefined parameters to govern the search. These parameters dictate how the initial generation of sequences is developed, as well as each subsequent generation. In addition, the parameters establish the criteria for the algorithm to terminate and present the solution to the user.

3.4.2.4 Schedule Generation Algorithm Definition

The schedule generation scheme used in this work is a serial schedule generation scheme (SGS) adapted from Hartmann and Kolisch (1998). The serial SGS accepts a sequence from the GA and schedules the activity at the earliest resource feasible time. Once the activity is scheduled, the impact of space on the activity duration is estimated through the use of a congestion function. After all of the activities in the sequence are scheduled, the makespan of the schedule is reported back to the GA.

The congestion function used in this thesis was developed for the problem described in Section 3.3. The developed congestion function draws from relevant literature (Akinci and Fischer 2000; Akinci, Fischer, et al. 2002c; Mallasi 2006; Tabet and Belieau 1997) to form a composite function based on conflict volumes of space types.

3.5 Calculation of Space Requirements

The execution, laydown, and travel path spaces of each installation component are calculated at initialization of the algorithm. From the *Space Model* each activity has a component associated with it and each component has a solid geometry representation. The three spaces are generated as functions of the solid geometry and the resource requirements for each activity.

3.5.1 Execution Space

Execution space is the space needed around the component for reach and access and the worker(s) to perform the work. The reach and access space of a component is modeled as a solid object created by generating the bounding box of the unit and increasing its size by 0.25 feet on all sides with a net increase of 0.5 feet in each dimension. The worker space is modeled as a rectangular prism that is 3’x3’x6’ if the installation location requires the worker to be standing and 3’x3’x3’ if kneeling. Figure 3-3 shows an example of the execution space for a component and the worker assigned to perform the task. The worker spaces were represented as rectangular prisms as a simple method for approximating the space requirements. Cylindrical or more complex shapes could be used as an alternative to more closely model the true space, however for simplicity, rectangular shapes were chosen to demonstrate the developed methodology.

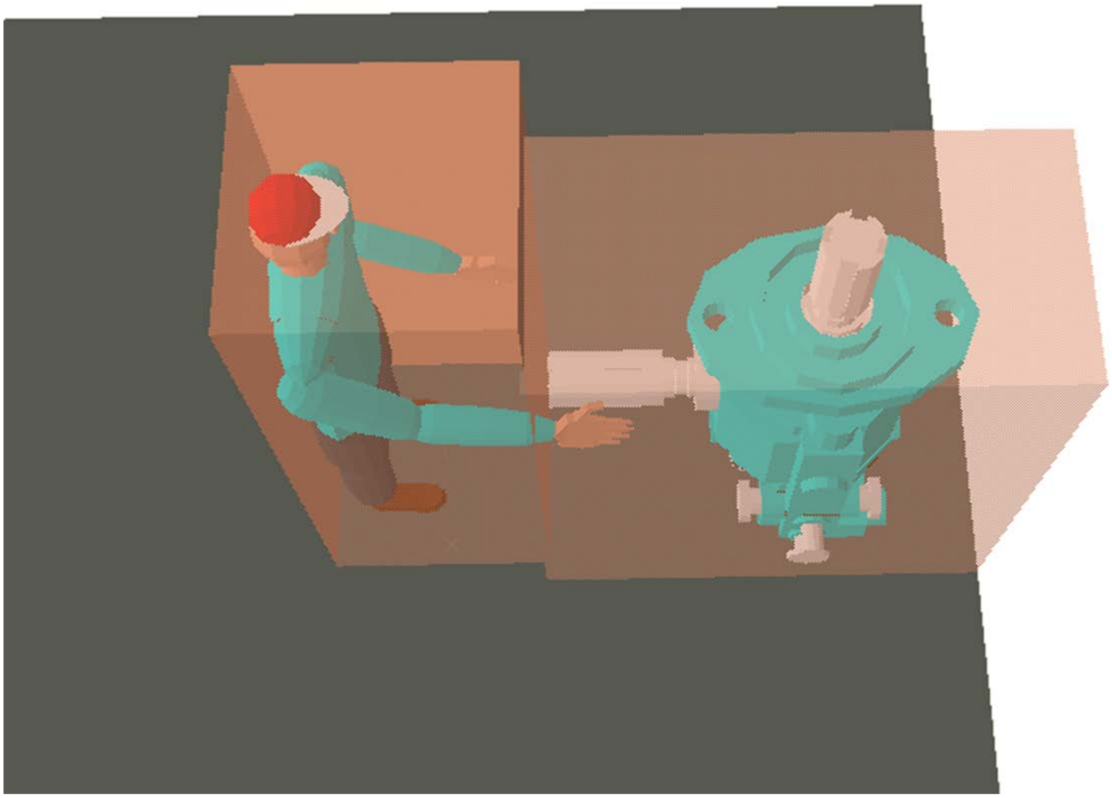


Figure 3-3: Execution Space Example.

An activity may require more than one worker to install the component. In this case, two worker spaces would be modeled and included in the execution space. It is assumed that at most two workers will be used to perform an installation. The location of the worker spaces follows the assumption that they will be "in front of" the component for installation and the reach and access space is for getting to the sides and back of the component if necessary. The front of the component is the face of the bounding box of the component that is closest to the center of the unit. If a second worker is needed for installation, the space will be located to one side or the other of the component, chosen at random. This method loosely follows that found in Akinci, Fischer et al. (2002a). More complex models of execution space could be developed; such models are outside the focus of this thesis.

3.5.2 Laydown Space

The laydown space is generated in one of two ways depending on the complexity of the geometry of the installation component. In most cases, the laydown space is simply a copy of the geometry placed at the predefined laydown point, at the entry point of the unit. When the geometry of the component is complex, i.e., a cabinet with several shelves, the bounding box of the component is used. The bounding box representation reduces the computation time of the space generation with little effect on the performance of the algorithm. The component is oriented at the laydown point in the same orientation that it is in for installation.

3.5.3 Travel Path Space

Travel path space is generated by two different methods: 1) straight line swept volume and 2) grid-based search procedure. The swept volume method is a simple representation of the travel path and is used to initially define the travel path space. If the swept volume travel path space conflicts with another activity's space during evaluation of the sequence a grid-based search algorithm is used.

3.5.3.1 Straight Line Swept Volume

The travel path space is first modeled as a swept volume of the component from the laydown point to the installation location. This travel path is generated during initialization. The swept volume follows a path along the floor until it is below the installation location, then travels up to the correct elevation within the unit. This approach follows the assumption that the component would be moved to the installation location with a hand truck (along the floor), then lifted into place.

3.5.3.2 Grid-Based Search Algorithm

The A* search algorithm is a best-first search algorithm that is used to find the shortest path from an initial state to a goal state (Hart, Nilsson et al. 1968). In the A* search algorithm used here, the unit is divided into a grid and the component is moved from grid point to grid point in the direction of the installation location. At each grid point, a collision detection check is performed between the component and all other previously installed components (work in place) to ensure that the grid point is feasible. In addition, a line is drawn from the previous grid point to the current grid point and checked to see if it passes through any work in place. The line check ensures that the component can feasibly move from the previous point to the current point without colliding with work in place.

The grid spacing is a key parameter in the computation time of the search. Large grid spacing results in fast searches because there are fewer points to check. However, a component may not be able to move from one point to the next in problem instances where the components are installed close together, i.e., less than the grid space. Conversely, small grid spacing increases the number of points to check which increases computation time, but increases the resolution. Knowing this relationship, a nested approach is used to try and balance the number of points visited with the granularity needed for the search to find a path. The initial grid spacing used is a function of the size of the installation component and a constant and takes the form in Equation 3.1.

Let compXdim = the size of component in the X dimension,
compYdim = the size of the component in the Y dimension,
compZdim = the size of the component in the Z dimension,
gridMin = the nominal minimum grid size, and
gridMax = the nominal maximum grid size.

$$\text{Grid Space} = \text{MIN}(\text{MAX}(\text{compXdim}/2, \text{compYdim}/2, \text{compZdim}/2, \text{gridMin}), \text{gridMax}) \tag{3.1}$$

The maximum function in Equation 3. 1. finds the largest dimension of the component or a spacing of gridMin length units. The half-width of the component dimensions are used to ensure that the component geometry will overlap from one grid point to the next forming a continuous path. Figure 3-4 shows an example of the grid spacing using the half-width dimensions of the components. Case A illustrates movement along one axis and case B shows movement along two axes. In both cases the geometry of the two components overlaps.

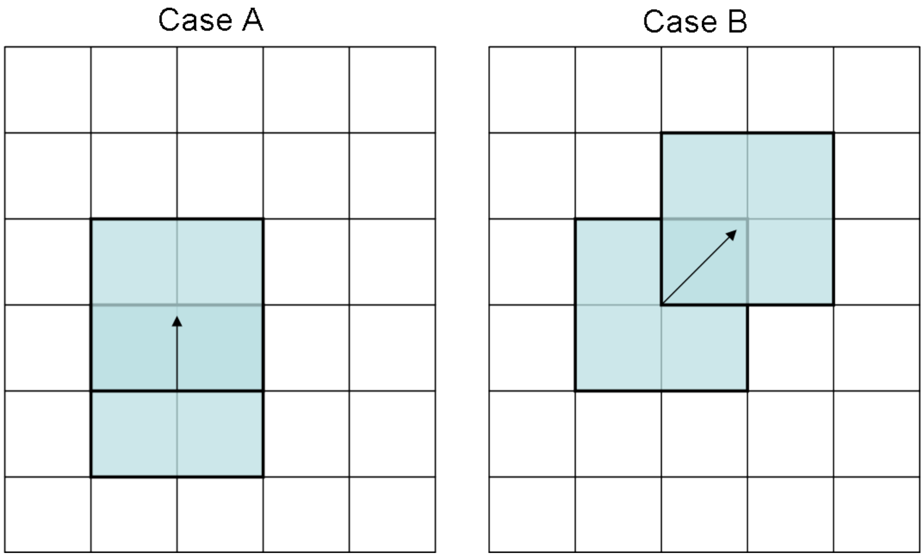


Figure 3-4: Grid Spacing Using Half Dimensions.

The minimum function in Equation 3. 1 limits the grid spacing to gridMax units. The gridMin and gridMax parameters are estimates of the minimum and maximum grid spacing, respectively and provided by the user.

If the search does not find a feasible travel path, the grid space is divided in half and the search restarts. If no feasible travel path is found using this smaller grid size, the grid size is again divided in half. By starting large and working to smaller grid sizes, the relationship between grid spacing and computation time can be exploited, having the majority of the searches find feasible paths with the large grid sizes, but not exclude feasible installation paths simply because the grid spacing resolution was not small enough.

The orientation of the components in the initial searches was held constant as the installation orientation. If no feasible travel path is found, reorientation of the component may be necessary. The component is moved along a grid spacing set at a constant minGrid units and allowed to rotate 45 and 90 degrees along the z axis when it encounters a collision with work in place. If there is no collision with work in place at a grid point when rotated, the search backs to the previous point and checks to see if there is a collision at this new orientation. The back step is performed to ensure that the new orientation can be achieved in the space between the two points without colliding with work in place.

There are two noticeable issues with the travel path method: grid spacing and rotation limitations. The proper grid spacing is a function of the component being installed, the unit where the installation is being performed, and the packing of the components within the unit space. The grid spacing values presented have been selected based on the motivating example and should be re-evaluated as the methodology is applied to other problem instances. Similarly, rotation about the z axis was chosen as most appropriate for the motivating example. In general, the components should be able to rotate in all three dimensions at any point along the grid. The computational complexity of a search that allowed three-dimensional rotation would be quite large and is out of the scope of this thesis.

3.5.4 Work in Place Space

Work in place space is the space occupied by the components that have been installed or are in the process of being installed. The unit is initially empty and as activities are scheduled the interior space becomes occupied by the geometry of components. The support spaces for the installed components are removed from spatial calculations as the activities are completed.

The actual component geometry is used to represent the work in place space, however an interesting point arises when the component is a structure that encloses a void, i.e., a sheet metal cabinet. Consider 2 cases: 1) an empty sheet metal cabinet with the door pre-installed (a closed object with a void) and 2) the same cabinet without the door pre-installed (an open object). In the case where the door is pre-installed, the enclosed void is treated as work in place space. In the case where the door is not pre-installed, the void space is treated as available space and is not used in the congestion function calculations.

3.6 Heuristic Search Algorithm – Genetic Algorithm

The implementation of the genetic algorithm follows the basic structure given in Chapter 2 and a flowchart of the methodology is given in Figure 3-5.

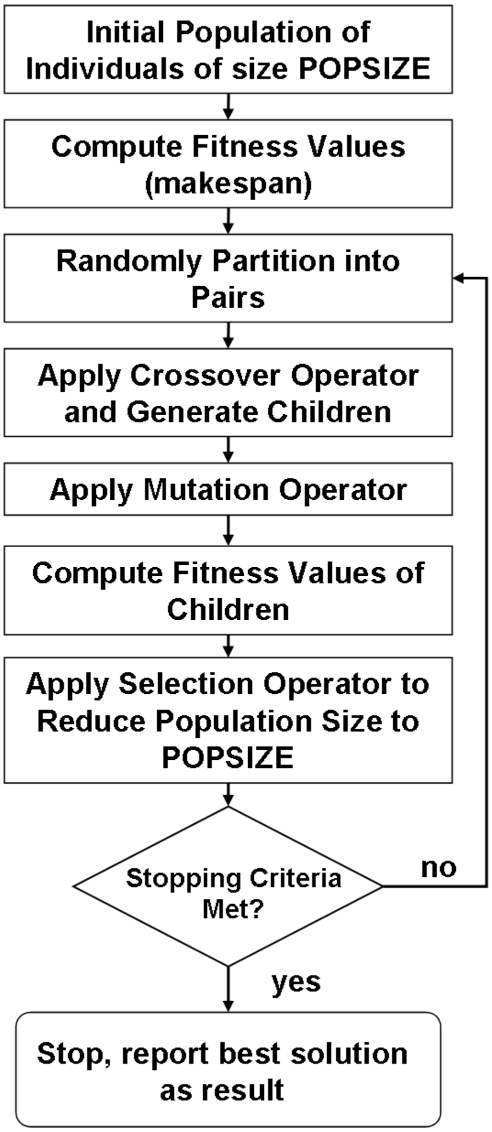


Figure 3-5: Genetic Algorithm Flowchart.

The main function of the genetic algorithm is to generate sequences for evaluation. There are two phases of sequence generation; initial feasible sequence population and all subsequent sequences. At initialization, a population of feasible sequences is generated. Each sequence is evaluated by scheduling the activities in the order specified, considering both resource availability and space availability, and calculating the makespan of the schedule. At the end of processing each generation of sequences, the GA creates a new generation. The process repeats until a termination condition is reached.

3.6.1 Initial Feasible Sequences

The GA begins by generating a set of precedence feasible sequences. This set represents the initial population. Each sequence in the initial population is determined by generating a random sequence that is reorganized to be precedence feasible. The reorganization process is defined in the following steps.

1. Generate a random sequence.
2. Take the first activity in the sequence and add it to the precedence feasible list.
3. Take the next activity from the sequence and add it to the precedence feasible list according to the following:
    a. If no activity on the precedence feasible list contains this activity in its precedence tree, i.e. it is a predecessor to it, or any of its predecessors add it to the end of the precedence feasible list.
    b. Otherwise, add the activity to the precedence feasible list in the position just before the activity that contains it.
4. Repeat step 3 until all activities in the initial random sequence have been evaluated.

Consider the following supporting example.

Let N = the number of activities,

$PTree_i$ = the precedence tree of activity i.

S=a random sequence of the activities, $\{a_1, \ldots a_N\}$, and

PFList = the precedence feasible list.

The first position of PFList is initially occupied by a_1. If a_2 is in $PTree_1$, then PFList = $\{a_2,a_1\}$, else it is $\{a_1,a_2\}$. The subsequent activities in S are checked against each element of PFList and added to the list according to this method. For example, the network diagram in Figure 3-6 shows a simple network of five activities.

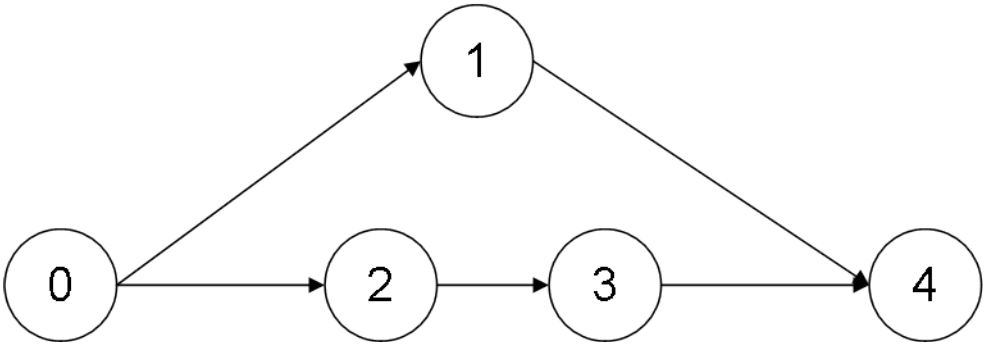


Figure 3-6: Example Network.

Let S={3,1,2,0,4}. Following the reorganization method,

1. PFList ={3}
2. PFList={3,1}, since activity 1 is not in the precedence tree of activity 3.
3. PFlist={2,3,1}, activity 2 precedes activity three.
4. PFlist={0,2,3,1}, similarly, activity 0 precedes activity 2.
5. PFlist={0,2,3,1,4}, activity 4 is preceded by all other activities, and is thus put at the end of the list. The precedence feasible list then becomes one member in the initial population and this process is continued until the set is complete.

3.6.2 Subsequent Sequences

Once all of the sequences in a generation have been evaluated, a new generation is formed by performing a crossover operation, followed by a mutation operation, and a selection operation. The crossover and mutation operations create diversity in the next generation by modifying the sequences. The result of these two operations is a set of sequences that is twice the size of the population. The selection operation selects a population sized set of sequences to move to the next generation.

The crossover operation used here is a two point crossover, where two “parent” sequences mother (M) and father (F) are selected at random from the population and used to create two “child” sequences daughter (D) and son (S). Two integers are randomly generated between 1 and the number of activities, N. Let p_1 be the smaller integer and p_2 be the larger. The first p_1 activities from M are used to create the first p_1 activities of D, that is,

$a_i^D = a_i^M$, where $i = 1 \dots p_1$.

The positions $i = p_1 + 1 \dots p_2$ of D are filled from F such that,

$a_i^D = a_j^F$, for $i = p_1 + 1 \dots p_2$, where j is the lowest index such that $a_k^F \notin \{a_1^D, \dots, a_{i-1}^D\}$.

The last $N - p_2$ activities of M are used to complete the last activities of D such that,

$a_i^D = a_j^M$, for $i = p_2 + 1 \dots N$, where j is the lowest index such that $a_k^M \notin \{a_1^D, \dots, a_{i-1}^D\}$.

The S child is created similarly using the first p_1 activities of F, the next p_2-p_1 activities from M and the last N-p_2 activities from F (Hartmann 1998). Figure 3-7 graphically illustrates the crossover operation.

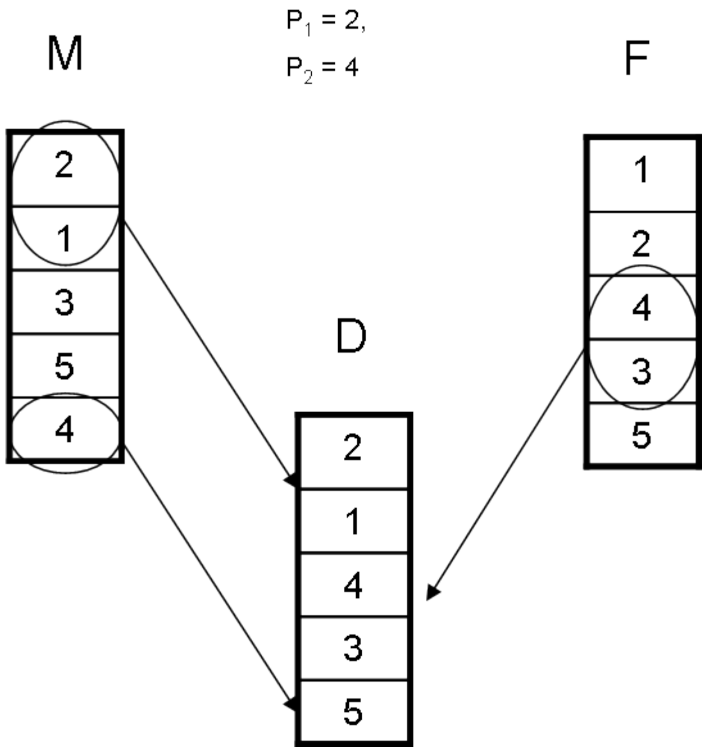


Figure 3-7: Graphical Representation of the Crossover Operation.

Notice that the last activity in D is actually taken from F instead of M because each activity can only be in the child sequence once. The new sequences developed from the crossover operation are evaluated as they are generated to obtain a fitness function value.

The crossover operation is only carried out a certain percentage of the time. The crossover rate dictates this percentage and is set at 95% as that has been empirically shown to be a good setting for this parameter (Hartmann 1998).

Each sequence must remain in a precedence feasible state. The initial population is a set of precedence feasible sequences and it has been proven that precedence feasibility of the children sequences is maintained through the crossover operation (Hartmann 1998).

The mutation operation modifies a sequence to encourage diversification that could not occur through the crossover operation. This operation simply iterates through the sequences in the population and performs a swap operation. This occurs at a rate set at 20% (Hartmann 1998). If precedence feasibility is not maintained with the swap, the operation is not performed and the original (child) sequence is left unchanged.

The crossover operation creates a set of sequences twice the size of the population size parameter. The selection operation down selects the sequences to conform to the parameter setting. In this implementation, the selection process is an elitist binary tournament with replacement approach. The elitist component selects the two sequences from the population with the best makespans, breaking ties arbitrarily, and automatically selects them for the next generation. The remaining sequences are filled using a binary tournament selection, where two sequences are selected at random from the current population and the sequence with the smallest makespan is selected for the next generation. The replacement component indicates that when a sequence is selected it is copied to the next generation and remains in the pool for future selection. The binary tournament with replacement has been shown to work well for the RCPSP (Mitchell 1996).

3.6.3 Stopping Rule

The genetic algorithm continuously generates a population of sequences and evaluates them according to a scheduling procedure. This process continues until the number of generations has been exhausted and each sequence within the population has been evaluated. As the population size and number of generations increase, the probability of finding the optimal solution increases, however so does the solution time. The relationship between good solutions and solution time is one of diminishing returns, i.e. solution quality improves dramatically in the initial stages of the algorithm, and very slight improvements are made in the latter stages. The standard in the current literature is to set the population size to 20 and the number of generations to 50, which allows the algorithm to evaluate at most 1000 unique sequences (Brucker and Kraemer 1996; Hartmann 1998).

3.7 Schedule Generation Algorithm - Sequence Evaluation

The evaluation of each sequence creates a schedule from the sequence. The makespan of the resulting schedule is the fitness of the sequence. The scheduling method is based on the serial scheduling generation scheme where the activities are scheduled at the earliest resource feasible time. Scheduling an activity at the earliest resource feasible time entails allocating resources, determining the spatial conflicts, calculating a congestion factor, and adjusting the duration of the activities.

3.7.1 Schedule Activity at Earliest Feasible Time

The allocation of resources simply finds the earliest possible time that all of the required resources of each type are available and reserves them. In this implementation, each resource is modeled individually rather than as a pool of generic resources that are drawn upon to fulfill the requirements. In this manner, a specific resource unit assigned to an activity will remain assigned for the entire activity duration.

Because each resource is modeled individually, a concept known as available window is used. The resource available window is a set of time slots for which the resource unit is idle. At initialization, the resource available window set has one member that is bounded by the project scheduled start and some nominal time after the project due date, say one month. Placing the end of the time window past the project due date ensures that the resource will be available if the project should take longer than expected due to duration increases caused by resource competition or space congestion. When a resource is selected for use by an activity, the available window is split into two time slots using the activity start and end times. Figure 3-8 shows this concept graphically. The scheduling algorithm searches through each resource's available windows and finds the earliest time at which a requirement can be met.

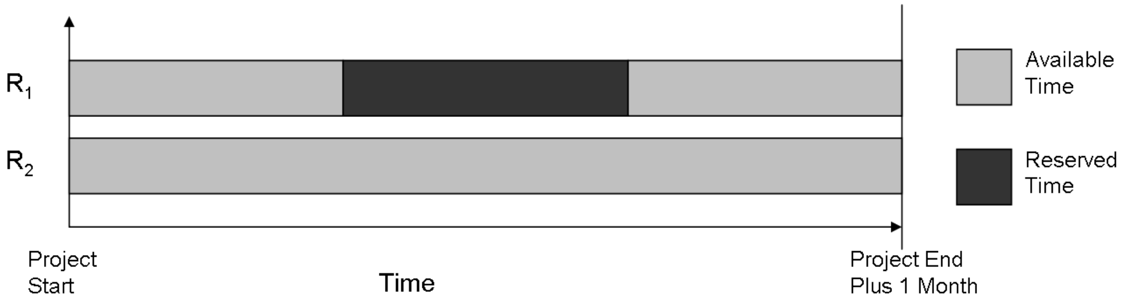


Figure 3-8: Time Window Concept.

Once the start time of the activity is determined the available windows for each resource are updated and the method begins checking spatial conflicts. The spatial conflicts are quantified and passed to the congestion function which recalculates the duration of the activity.

3.7.2 Calculate Space-Time Conflicts and Congestion Computation

Activities require space for installation. When several activities are being performed concurrently, the spatial requirements may conflict with one another. This spatial conflict is defined as congestion. As congestion increases, the time that it takes to perform an activity also increases. The approach developed here attempts to accurately model the space requirements of each activity, which will provide an opportunity to quantify the amount of spatial conflict. The amount of conflict, measured as a function of the conflict volume, can then be used to estimate the congestion within the work space. This congestion function is used to model the effect of spatial conflicts on the duration of the activities.

Each of the space types in the model can potentially conflict with one another. These spatial conflicts are used to calculate the congestion. However, the effect on congestion is a function of the types of spaces that are in conflict.

3.7.2.1 Types of Congestion

Spatial conflicts occur between work in place space and the three types of space described earlier (laydown, travel path, and execution space). The general relationship between each type of conflict is given in Table 3-1. Each box in the table shows the installation scenarios for two components, A and B. Consider the case where the installation activity of component A has been scheduled and is in progress during the installation activity for B which is currently being scheduled. In Table 3-1, the cell pairs under each space type indicate if the spaces are allowed to conflict and the effect of their conflict, if they do. Labels for the effects (1-3) are given below the Allowed/Not Allowed box. Further details are provided below the table for each of the labels.

Table 3-1: Spatial Conflicts Between Space Types.

		Component A			
		Component Space	Execution Space	Travel Path Space	Laydown Space
Component B	Component Space	Not Allowed			
		1			
	Execution Space	Allowed	Allowed		
		2	2		
	Travel Path Space	Allowed	Allowed	Allowed	
		2	2	3	
	Laydown Space	Allowed	Allowed	Allowed	Allowed
		2	3	3	3

1. Two component spaces overlapping would be a design flaw. This situation essentially results in two components occupying the same space. This type of space overlap is not allowed.
2. Spatial conflicts in this category will result in a duration increase for the installation of the activity currently being scheduled. The amount of increase will be determined through the congestion function. See Congestion Function section below.
3. The overlapping of these space types will have no effect on the activities. These overlaps involve transient space types that occur at the beginning of the activity. The overlaps that occur in this category will be on such a small time scale (minutes or hours) compared to the activity duration (days/weeks) that they can be accommodated without affecting the duration of the installation activity.

3.7.2.2 Effect of Congestion on Activity Duration

It has been shown that workspace congestion decreases the productivity of construction activities (Mallasi 2006). This productivity loss is reflected in increased span times for the tasks in the schedule (Akinci and Fischer 1998). Each of the activities has a nominal duration which represents the duration of the activity should it be performed in an un-congested and unconstrained work area. As congestion is increased, activity duration also increases. It is assumed that all previously scheduled activities take priority over the current activity being scheduled and that only duration of the activity currently being scheduled is increased due to congestion. It is necessary to assume this so as to not cause an endless loop of duration increases due to spatial conflicts of concurrent activities. For example, if the durations of all concurrent activities were increased, then when a new activity was scheduled, they would need to be updated again if they were still in process. As the activity durations are updated, the resource allocations would also need to be revisited and would cause additional disturbances to the schedule that could cause additional duration increases to occur. This iterative process would continue and could result in a schedule that greatly exaggerates the effect of congestion on the duration of the activities.

A congestion function is used to model the duration increase of each activity. The various spatial conflicts that occur when an activity is scheduled have different effects on the duration increase depending on the type of spaces that are conflicting. The congestion function developed in this thesis is a composite function with a multiplicative component and an additive component applied to the activity duration. The multiplicative component represents conflicts that result in duration increases that are a function of the duration of the activity. The additive component of the congestion function quantifies spatial conflicts that add time to the activity, but are independent of the activity duration.

The multiplicative component of the congestion function is comprised of two spatial conflict elements: 1) conflicts involving execution space and work in place and 2) a factor that models the general congestion within the unit. There are four conflict types included in the additive component: 1) execution space conflicts with other activity execution spaces, 2) laydown space conflicts with work in place, 3) travel path space conflicts with execution space, and 4) travel path space conflicts with work in place.

3.7.2.2.1 Execution Space and Work in Place Conflicts

A conflict between execution space and work in place is modeled as the intersection volume of the two solids representing the spaces. A flowchart describing the calculation of the percent 4D conflict volume is given in Figure 3-9.

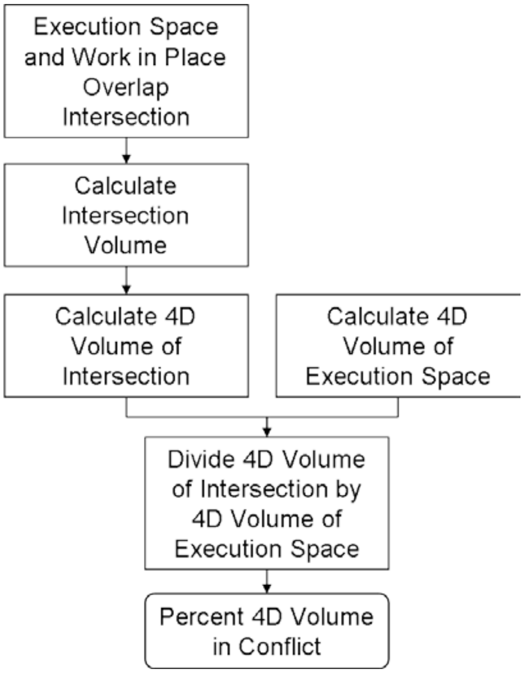


Figure 3-9: Calculation of Percent 4D Conflict.

The intersection volume of the two spaces is multiplied by the duration of the conflict to form a four-dimensional (4D) conflict volume. The 4D conflict volume is then divided by the 4D volume of the execution space to obtain the percentage of 4D space in conflict (Percent 4D Conflict). The 4D execution space is calculated by multiplying the execution space volume of the current activity by its duration. The effect of the conflict on the duration of the activity is a function of the Percent 4D Conflict. The duration increases as the Percent 4D Conflict increases, but not at a constant rate. Relatively small conflicts will not create duration increases, however at certain points, the duration increases at much higher rates because worker access is severely decreased.

The duration increase function should be derived from expert knowledge and historical performance. However, in this methodology, the author’s estimate of the duration increase percentages at given levels of percent 4D conflict are used. Table 3-2 shows an example table used to determine the duration increase at nominal values of 4D conflict. Then Nominal Percent 4D Conflict column represents nominal values of 4D conflict. The author estimated duration increase percentage values are given in the Estimate Percent Duration Increase column. Future research for this problem would develop a more comprehensive approach to developing the duration increases at various levels of percent 4D conflict.

Table 3-2: Estimated Values of Percent Duration Increase at Nominal Values of Percent 4D Conflict.

Nominal Percent 4D Conflict Levels (%)	Estimated Percent Duration Increase (%)
0	0
25	2.5
50	62.5
75	110
100	160

In this example, a third order regression model is fit to the tabulated values. For the example in Table 3-2, the result is Equation 3.2.

Let PDI = the percent duration increase and

P4DC = the Percent 4D Conflict.

$$PDI = 2.21(P4DC)^3 - 3.12(P4DC)^2 + 2.32(P4DC) \qquad 3.2$$

The regression analysis and ANOVA table used to create this relationship are given in Appendix A.

The estimated values of Percent Duration Increase at the nominal Percent 4D Conflict Volumes are plotted in Figure 3-10. The corresponding regression function values at the nominal points are also plotted.

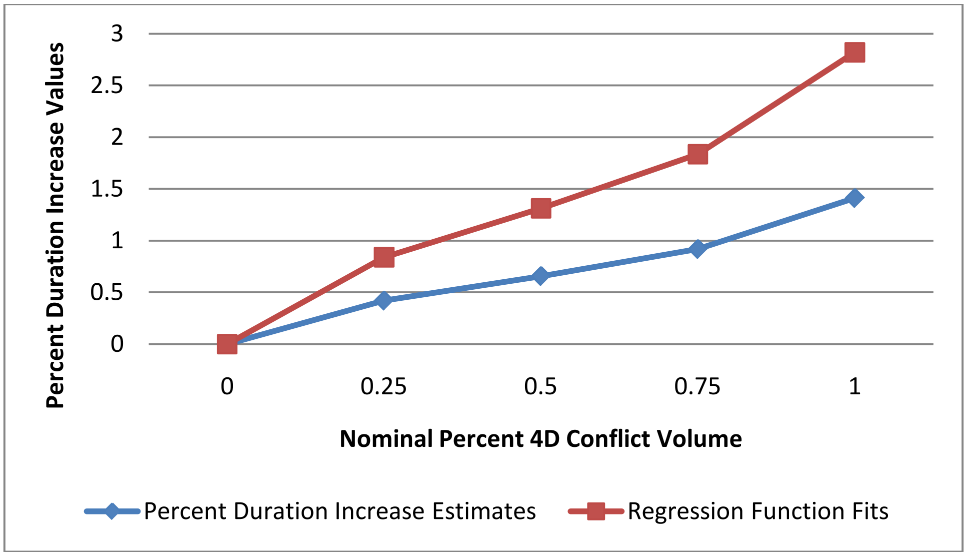


Figure 3-10: Congestion Function Component Values Plot.

The values show very little increase in the activity duration at small levels of conflict and much larger duration increases at higher levels of percent conflict.

3.7.2.2.2 General Congestion

Congestion within the unit can be affected by support services and by the space taken up by previously installed components (Guo 2002; Winch and North 2006). In the shipbuilding industry, many support services are required during installation including ventilation, air hoses, gas hoses, temporary lighting and fixtures. All of these items add to the congestion within the unit, but cannot be explicitly modeled because of the number of additional elements and lack of geometrical representation of those elements.

This thesis focuses on congestion due to previously installed components. To estimate this congestion, a working envelope is created around the work location for the activity being scheduled. The size of the working envelope depends on the specific problem instance; for the motivating example the envelope is 4.5’x4.5’x9’. It is centered around the working location, and shifted if necessary to remain within the bounds of the unit. The percentage of the envelope volume that is occupied by work in place is calculated. As before, effect of the percentage of used space on the duration is a function derived by a table of values similar to the execution space/work in place spatial conflicts defined in Section 3.7.2.2.1. A regression model is fit to the data. The regression analysis and ANOVA table for the motivating example are given in Appendix A.

3.7.2.2.3 Execution Space – Execution Space Conflicts

Execution space conflicts occur when multiple activities are in process simultaneously, and result in a duration increase for the activity currently being scheduled. The increase is a function of the conflict volume and the duration of the conflict. Calculation of the duration increase follows the method of Section 3.7.2.2.1.

For this conflict type, the maximum duration increase is limited to the duration of the conflict. Suppose that activity B is being scheduled and its execution space conflicts 100% with the (ongoing) activity A. The conflict would be eliminated by waiting to start activity B until A is completed. Figure 3-11 graphically illustrates this example.

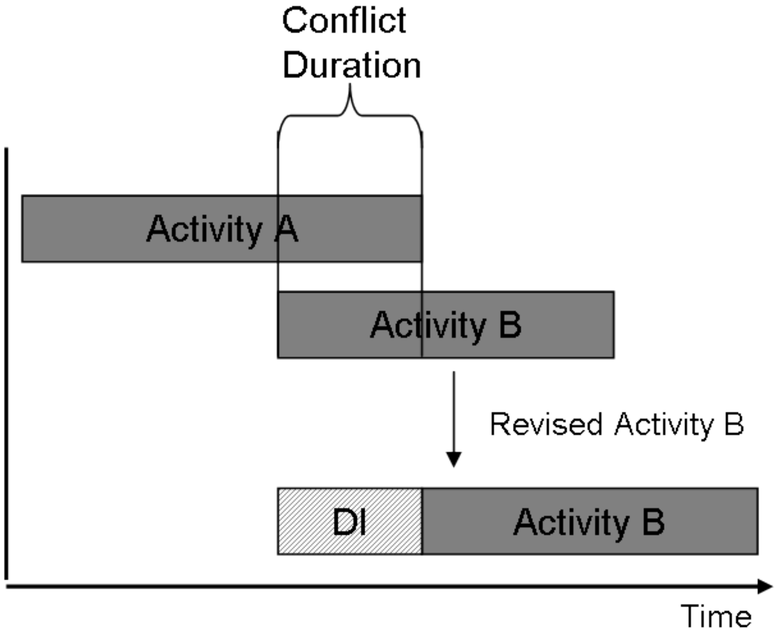


Figure 3-11: Execution Space – Execution Space Duration Increase Example.

The DI area on the revised Activity B indicates the maximum duration increase caused by this type of spatial conflict.

3.7.2.2.4 Laydown – Work in Place Conflicts

The laydown space for a component is used to transfer from one material handling device to another; this space is located near the entry point of the unit. In practice, there are several transfer methods available with varying degrees of difficulty. As the difficulty of the transfer process increases, so too does the time it takes to make the transfer. The nominal transfer process duration is included in the nominal duration for the activity. However, if the laydown space conflicts with some work in place, the method will change and result in an increase in the activity duration. The transfer process is assumed to be relatively short compared to the duration of the activity, which indicates that the duration increase caused by this conflict type will be small. A duration increase function similar to the previous conflict types is used for this conflict type. However, the conflict is calculated as a 3D volume rather than a 4D volume. There is no time component due to the fact that no time is explicitly allocated to the laydown operation, and as a result of the assumption that the transfer time will be small relative to the activity duration.

3.7.2.2.5 Travel Path Conflicts

The travel path is initially modeled as a straight line swept volume that simplistically estimates the travel path from the laydown point to the installation location. If the straight line swept volume travel path conflicts with any other space for a given activity, then a more complex grid-based search is used to compute a more complex travel path.

For this conflict type the percent of 3D space in the collision is used to estimate the duration increase, because travel time from the laydown point to the install location is assumed to be minimal, similar to laydown space conflicts. The travel path can conflict with execution space and work in place space and may increase the duration of the activity.

The travel path space is initially modeled as the swept volume of the component moving to its install location, first in the x-y plane then in the z. If there is any conflict between that travel space and any execution space or work in place, then a search procedure is used to generate a more detailed travel path. The search procedure is based on the A* search algorithm discussed in Section 3.5.3.2. The work space is divided into a grid. The search method moves a geometric representation of the installation component from grid point to adjacent grid point in the direction of the install location. The search method finds the shortest path between the laydown point and the install location. A new swept volume path is generated along the path found from the search algorithm. Spatial conflicts between the search method generated swept volume travel path and the execution spaces of the in process activities and the percent of the volume in collision are calculated. Figure 3-12 illustrates the travel path conflict calculation method.

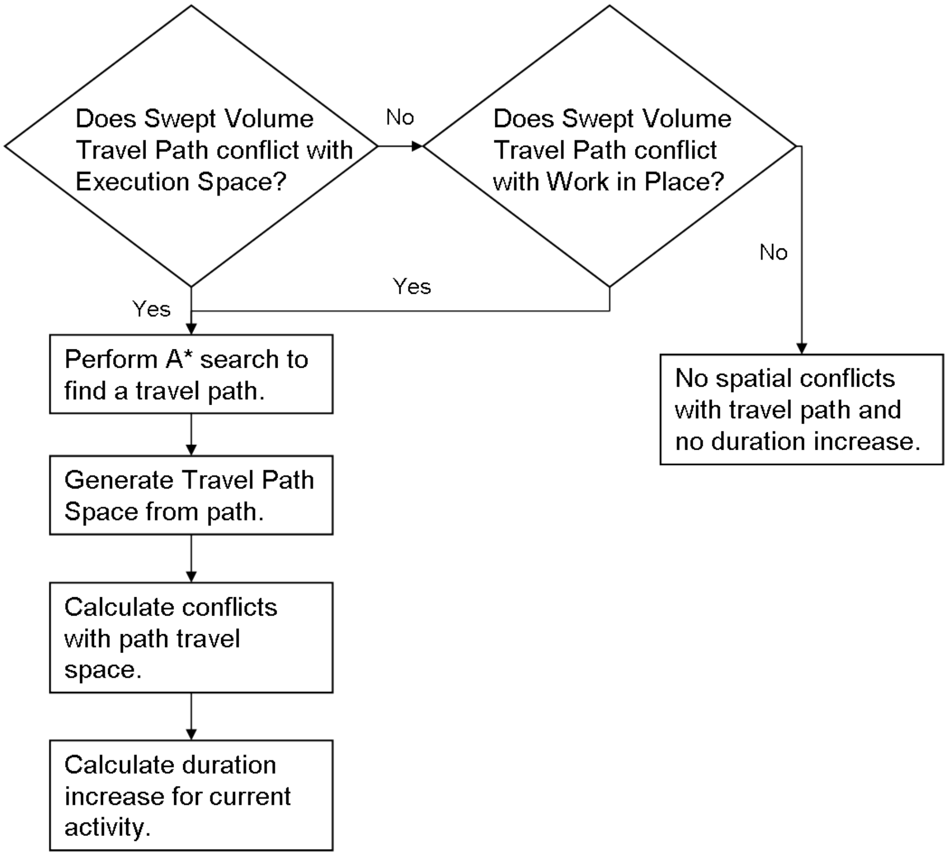


Figure 3-12: Travel Path Conflict Calculation Flowchart.

Calculation of the duration increase for the current activity is similar to that of the previous conflict type found in Section 3.7.2.2.4. Conflicts between travel path space and execution space are used in the calculation. Conflicts between travel path space and work in place do not factor into the calculation because, the search procedure finds a path that is free of conflicts with work in place. The search method explicitly checks each grid point in the path for these types of conflicts and eliminates them as feasible points.

If the A* search method is used to determine a travel path and no spatial conflicts occur, a duration increase is still added to the current activity. This duration increase is used to approximate the additional time the component will spend in travel due to the search path as opposed to the simple swept volume. If a travel path is found in the search method without needing rotation, a nominal time unit penalty (noRotateTime) is added to the duration to approximate the additional time required to move the component into place. A larger time unit increase (rotateTime) is added if rotation was needed. These two parameters are defined by the user and are estimations based on expert knowledge and historical data. The estimates chosen for the motivating example are the best guess estimates of the author. A method of developing the estimates should be addressed in future work in coordination with duration increase functions.

3.7.2.2.6 Congestion Function

Each of the congestion elements is combined into a single congestion function with two components: the multiplicative and additive. The final function takes the form in Equation 3.3.

Congestion Function Multiplicative = Execution/WIP Congestion + Used
Space Factor

Congestion Function Additive = Execution/Execution Congestion + Laydown/Work in Place Congestion + Travel Congestion Factor 3.3

3.7.3 Increase Activity Duration

The congestion function is applied to the nominal duration of the current activity during the scheduling phase of the algorithm. If the duration of the activity increases, the algorithm checks the resource requirements to ensure that they are still feasible. If the resource requirements cannot be met with the new duration, the activity is rescheduled at the earliest resource feasible time.

3.7.4 Rescheduling Activities

Rescheduling the activity is performed by searching through the time windows of the resources to find the number of resources that are required by the activity. The activity is scheduled in the earliest resource feasible time. The space-time conflicts are calculated and the congestion is also calculated for the activity in the new position. The duration is increased according to the congestion function and the resource feasibility is checked. This cycle continues until the activity can be scheduled feasibly. Once the activity is scheduled the algorithm proceeds to the next activity in the sequence for scheduling.

3.8 Chapter Summary

This chapter presented a solution methodology framework for the resource-constrained project scheduling problem with spatial resources (RCPSP-S). The objective of this framework is to determine a near-minimal makespan schedule subject to resource constraints and spatial requirements. A method based on the genetic algorithm heuristic search procedure was developed and described. The method accepts an *Activity Model* and a *Space Model* as the inputs and schedules activities at the earliest resource feasible start time. The method considers the spatial requirements of the activities and provides an approach to model the spatial conflicts using a congestion function. The congestion function is used to increase the duration of the activities to model the effect of congestion on the makespan of the project schedule. The congestion function is comprised of five elements related to spatial conflicts that occur within the installation phase of a general large-product industry. Each element is a function that models the effect of congestion on activity duration.

The main research contribution of this work is the comprehensive solution methodology framework that integrates several algorithms that have either been adapted from relevant literature or developed for a specific component in the framework. Ancillary contributions to the current body of research include the explicit inclusion of space and space types into a schedule generation algorithm and the congestion function that accounts for various space types in a composite nature that directly affects the duration of the activity that is currently being scheduled.

Chapter 4

Experimentation, Results and Conclusions

This chapter presents a description and review of the testing of the algorithm. In general, the purpose of testing the algorithm is to see how well it performs compared to existing methods that solve the same or similar problems. In this case, there is no direct method or problem set for comparison in the current literature. Therefore the testing performed in this thesis is to prove that the algorithm can solve the RCPSP-S and to characterize the effect of space on the underlying RCPSP.

To characterize the effect of space, the algorithm was tested against a well known problem set for the RCPSP, PSLib (Kolisch and Sprecher 1997), with and without spatial resources. The purpose of testing the algorithm on the problem sets without taking space into consideration was to establish a baseline performance of the underlying project scheduling algorithm. Spatial resources were added to a problem instance at three levels to determine the effect of space. In addition, the algorithm was tested on a real world model generated from design data of a Torpedo Weapons Retriever vehicle.

4.1 Implementation Discussion

The RCPSP-S methodology was implemented in Java© programming language. The software incorporated several open source libraries with custom code to obtain the desired functionality. A graphical user interface was developed to enable interaction with the user. Within the interface, the user can load a problem, view the network, view the geometry, run the algorithm, and view the output from the algorithm in a Gantt chart. Views of the software implementation and a brief explanation of each function are given in Appendix B. The software was developed and all tests and experiments were performed on a Dell Precision M6300 using a 2.4GHz Intel® Core™2 Duo CPU T7700.

4.1.1 Space Generation and Implementation

The laydown, travel path, and execution spaces of each installation component are calculated at initialization of the algorithm. Each activity has a component associated with it and each component has a solid geometry representation. The three spaces were generated as described in the previous chapter and are functions of the solid geometry and the resource requirements for each activity.

The solid geometry representation of the components and their respective support spaces were modeled as Java3D© objects using the open source Java3D library UnBBoolean (Balby 2009). This library includes functions to perform CAD operations such as union, intersection, and difference of the solid objects. The spatial conflicts were modeled as intersections of two solid objects. The intersection method iteratively calculates the intersection volume of the current activity's spaces, the spaces of in-process activities, and the work in place space. The volume of the intersection is estimated using a Monte Carlo-based method that counts the percentage of random points contained within the resulting intersection volume. The number of points used is determined by finding the volume of the bounding box, rounding it to the nearest integer.

The points are randomly generated within the bounding box and identified as inside or outside the intersection of the spaces. The percentage of points contained within the intersection is multiplied by the volume of the bounding box, resulting in an approximation of the volume of the intersection space. For this application, an exact volume calculation is not needed because the congestion function is not sensitive to relatively small changes in the volume value. A small example is given in Appendix A to demonstrate this assertion.

4.1.2 Congestion Function Implementation

The methodology for developing the congestion function components given in Chapter 3 was used to develop specific equations for the components. The equations for each of the components were generated by fitting a regression model to a set of values estimated by the author.

Table 4-1 shows the values used to generate the duration increase function for the Execution Space – Work in Place conflict type. The values in Nominal Percent 4D Conflict Level column represent nominal values that were used to estimate the duration increase effect. The Estimated Percent Duration Increase column values represent the estimated duration increase.

Table 4-1: Execution Space and Work in Place Conflict Effect Values.

Nominal Percent 4D Conflict Levels (%)	Estimated Percent Duration Increase (%)
0	0
25	2.5
50	62.5
75	110
100	160

A regression model was fit to the tabulated values. Several models of various orders were fit to the data. A third order model was the highest order model where all predictors were significant, i.e. p-value less than 0.05. The third order model was chosen as the function for this type of conflict and is given in Equation 4.1.

Let PDI = the percent duration increase and

P4DC = the Percent 4D Conflict.

$$PDI = 2.21(P4DC)^3 - 3.12(P4DC)^2 + 2.32(P4DC) \qquad 4.1$$

A regression analysis for each of the conflict types was performed in a similar fashion. The regression model that resulted in the best fit for the data was used as the congestion function component. The best fit equations for each conflict type are given in Table 4-2.

Let PDI = the percent duration increase,

DI = the duration increase,

P4DC = the Percent 4D Conflict, and

PC = the percentage of space volume in conflict.

Table 4-2: Congestion Function Component Equations.

Conflict Type	Regression Equation	
Multiplicative Components		
Execution – Work in Place	$PDI = 2.21(P4DC)^3 - 3.12(P4DC)^2 + 2.32(P4DC)$	4.2
General Congestion	$PDI = 5.46(P4DC)^3 - 9.14(P4DC)^2 + 4.24(P4DC)$	4.3
Additive Components		
Execution Space – Execution Space	$DI = 1.80(P4DC)^3 - 2.54(P4DC)^2 + 1.84(P4DC)$	4.4
Laydown – Work in Place	$DI = 0.1 * PC$	4.5
Travel Path – Execution Space	$DI = -1.03(P4)^2 + 2.03(PC)$	4.6

The data tables, regression analyses and supporting ANOVA tables used to create these relationships are given in Appendix A.

The estimated duration increases are domain specific and can be obtained by consulting with experts for a particular problem set. A new set of regression equations would then be computed.

4.1.3 Genetic Algorithm Implementation and Parameter Settings

An open source software library, jMetal (Metaheuristic Algorithms in Java) was used as the framework for the genetic algorithm in this implementation (Durillo, Nebro et al. 2006). The library was extended and customized to solve the project scheduling problem. The encoding and decoding schemes were developed around the framework and are described in Chapter 3. In addition, the spatial representations of the activities and their support spaces were integrated into the framework.

The genetic algorithm parameters were set using values and methods found in current literature as described in Chapter 3. Table 4-3 shows the settings used in this implementation for testing the methodology. The settings used in this implementation follow standard settings found in current literature (Brucker and Kraemer 1996; Hartmann 1998) for RCPSP solution procedures.

Table 4-3: Genetic Algorithm Parameters

Parameter	Setting
Population Size	20
Number of Generations	50
Crossover Operator	Two-Point Crossover
Crossover Rate	0.95
Mutation Operator	Random Pair wise Swap
Mutation Rate	0.20
Selection Operator	Binary Tournament with Elitism and Replacement

In the software implementation, a user is presented with the initial settings for the GA parameters and allowed to modify them prior to the beginning of the search.

4.2 Baseline Performance

The implemented methodology was used to solve several resource-constrained project scheduling problems (RCPSP) without space to establish a baseline performance of the underlying genetic algorithm. The baseline performance testing was performed to ensure that the genetic algorithm was able to satisfactorily solve the RCPSP. In addition, the baseline performance was used to characterize the effect of space on the project scheduling problem.

The project scheduling library (PSLib) is a library of benchmark project scheduling problems generated for researchers to test new algorithms in the resource constrained project scheduling area (Kolisch and Sprecher 1997). The library consists of problem sets with 30, 60, 90, and 120 activity networks that have either a known optimal solution or an established lower bound. Within each set, 48 different problem networks are defined and 10 instances of each problem are given for consideration. The RCPSP-S algorithm was tested on the 30 and 60 activity problem instances, for a total of 960 problems. Table 4-4 shows the results of testing against the problem instances without taking space into consideration.

Table 4-4: RCPSP-S Algorithm Performance without Space.

Deviation from Optimal	30 Activity Set	60 Activity Set
Min	0.0%	0.0%
Max	10.6%	18.8%
Average	1.29%	3.24%

The developed method solved the 30 activity set with an average deviation from optimal of 1.29% and the 60 activity set with an average deviation of 3.24%. In addition, the algorithm solved to optimality in 69.6% of the 30 activity instances and 62.5% of the 60 activity instances. Appendix C presents further details of these results.

4.3 Scenario Experimentation

One problem from the 30 activity PSLib data set was chosen to perform testing with spatial resources. The problem consisted of 30 non-dummy activities and four resource types with resource pool sizes of: R1(24), R2(23), R3(25), and R4(33). The resource requirements for each activity ranged from 0 to 10 for each resource type. The activity network is shown in Figure 4-1. The problem instance is given in Appendix D.

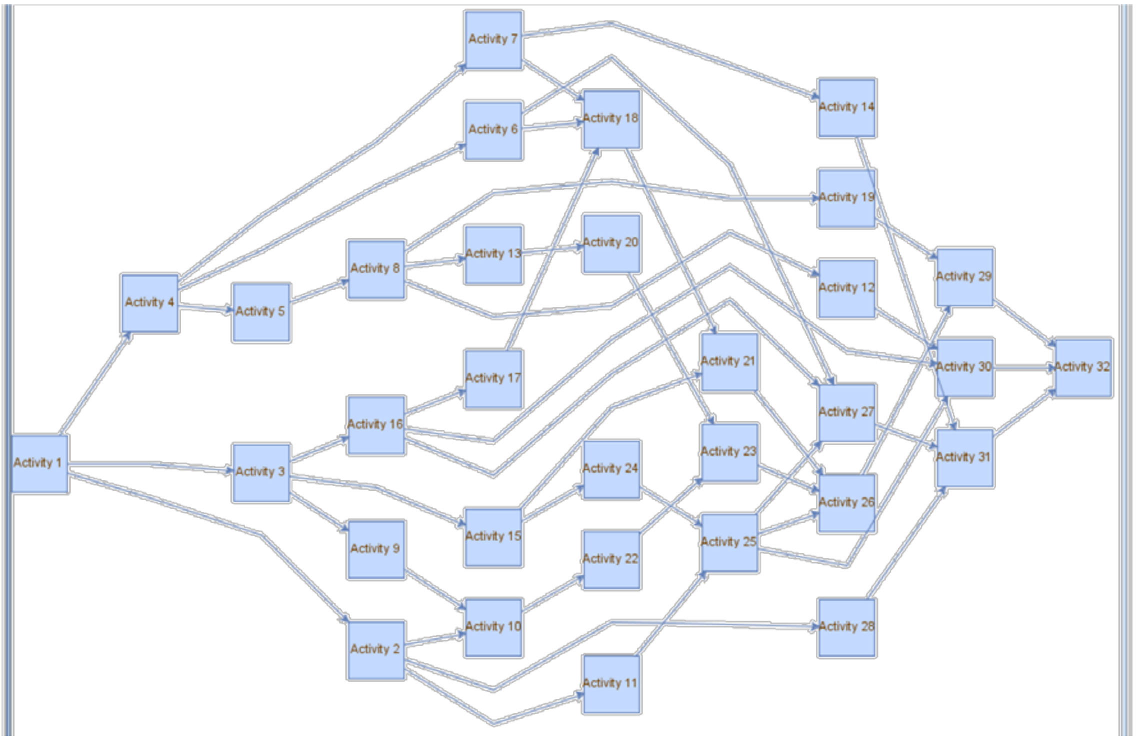


Figure 4-1: Activity Network.

A generic unit space of 9’x9’x9’ was used to represent the interior space within which the components would be installed. Generic rectangular prisms were generated at random to represent each of the installation components. The sizes of the components were randomly generated to fill out the unit space at three levels: 1)10% occupied

(Scenario 1), 2) 25% occupied (Scenario 2), and 3) 40% occupied (Scenario 3), i.e., the installed components within the space occupy 10%, 25%, and 40% of the 9’x9’x9’ space. Figure 4-2 shows a model of the work space and the components for Scenario 1.

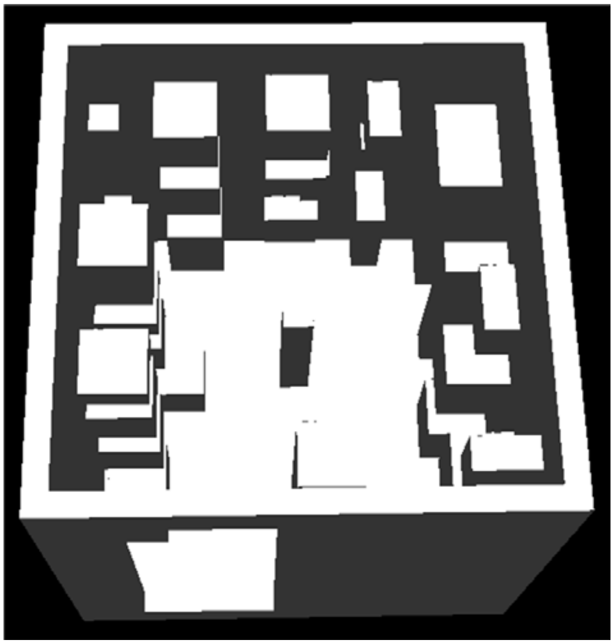


Figure 4-2: Scenario 1 Workspace and Components.

In comparison, Figure 4-3 shows the model for scenario 3 in which 40% of the space within the unit is consumed by the installed components.

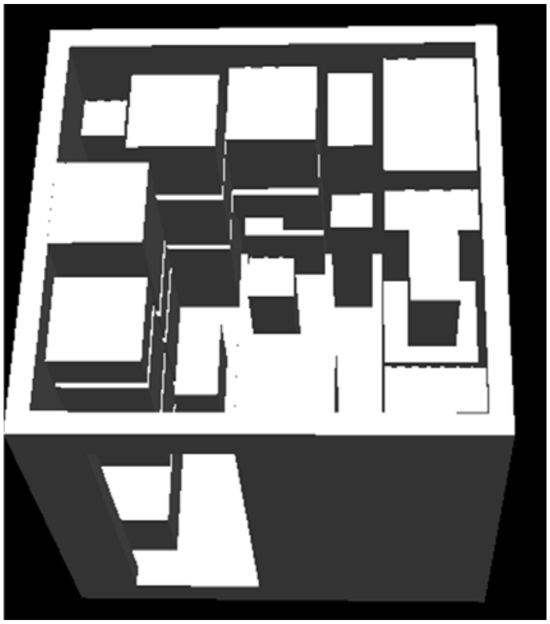


Figure 4-3: Scenario 3 Workspace and Components.

The laydown and travel path spaces were generated using the approach described in Chapter 3. The execution space was generated by growing each of the components by 0.5 feet in each dimension and adding the worker space.

In the test problem, the resource requirements for an activity ranged from 0 to 10 of each of the four resource types (welders, pipe welders, electricians, and riggers), with an average of 17 resources per activity. This quantity of worker type resources is not representative of the installation activities considered within the scope of this work, where one or two worker resource units are typical. Therefore, the average activity resource requirements were used to calculate the number of worker spaces required for the activity. If an activity required less than 17 total resources, then only one worker was needed. If the activity required 17 or more resources then two worker spaces were used. The original problem resource requirements and availabilities remained the same as given in the PSLib data set. For example, consider an activity that had the following resource requirements: R1(3), R2(0), R3(10), and R4(5). Each of the resource requirements would be fulfilled from the resource pools given in the problem, however since the total number of required resources (18) was greater than 17, two of the resources would represent worker type resources with corresponding spatial representations. The location of the worker space(s) was set at the interior face for the first worker and the side for the second.

Each of the three space scenarios was solved using the RCPSP-S algorithm and the results are given in Table 4-5.

Table 4-5: Test Problem Results.

	Optimal without Space	RCPSP-S without Space	Scenario 1 (10%)	Scenario 2 (25%)	Scenario 3 (40%)
Makespan	42	43	46	48	50
CPU Time (2.4GHz)	N/A	25 sec.	124 hrs., 35 min.	250 hrs., 53 min.	248 hrs., 26 min.

The results show that by adding spatial resources into the problem, the makespan of the project increases from 43 time units to 46, 48, and 50 with 10%, 25%, and 40% occupied space, respectively. This represents an increase in makespan of 7% for Scenario 1, 12% for the Scenario 2 and 16% for Scenario 3.

In Scenario 1, the duration was increased in six activities (20%) with an average time increase for these activities of 1 time unit. The average duration increase over all activities is 0.20 time units.

For Scenario 2, a total of 12 activities (40%) experienced a duration increase. The maximum time increase was 5 time units and occurred relatively early in the activity network with activity number 2. The average duration increase for the 12 activities was 1.83 time units, and the average increase for all activities was 0.73 time units.

In Scenario 3, a total of 16 activities (53%) had increased durations, with activity number 2 also recording the largest duration increase of 5 time units. The average duration increase of the activities that had duration increases was 2.06 time units. The average overall duration increase of the activities was 1.1 time units.

4.4 Computation Time Discussion

The computer processing time (CPU) for the case where space is not considered is approximately 30 seconds. Whereas, the cases that include space required between 124 and 251 hours of CPU processing time. This significant difference in processing times could prohibit the use of the methodology in practice. Some of the CPU time can be attributed to inefficient coding practices, which indicates that the code should be re-examined and optimized to improve performance. The bulk of the CPU time is spent in the A* algorithm determining a travel path. In highly congested scenarios, similar to Scenarios 2 and 3, the A* algorithm was invoked for roughly half of the activities in each sequence. The time to perform the search ranged from 1 to 30 seconds. Although these values are relatively small, they represent one search for a single activity. These times are amplified by the iterative nature of the GA procedure. For example, over the course of an entire GA solution run of 1000 schedules, assuming the A* algorithm was invoked for half the activities (15) and each invocation required an average of 15 seconds of CPU time, travel path generation alone takes 62.5 hours. The nested grid size approach was developed in an effort to reduce the computation time for the A* search path generation, however additional research is required to improve performance.

4.5 Case Study-Torpedo Weapons Retriever

The final test in the experimentation phase was to model and solve a real-world example from the shipbuilding industry. For this experiment, model geometry was provided and a subject matter expert was enlisted to help generate the activities and activity network. The Torpedo Weapons Retriever (TWR) is a tugboat sized vessel, approximately 75 feet long. It is used in shallow water for retrieval and towing operations. When built, this vessel was not designed using a 3D product modeling approach. However, modern ship design is primarily performed in CAD systems and 3D product models are being developed as a result. Most models of current vessels are not publicly releasable because they are either proprietary or are subject to security controls. Because the models are not publically releasable, it is difficult to conduct research that requires a model of a ship. To alleviate the difficulty, the shipbuilding industry, through the National Shipbuilding Research Program (NSRP), has modeled the TWR using CAD software and made the models available to the public for research (Benthall, Briggs et al. 2004; Gischner, Lazo et al. 2006).

The pilot house of the TWR was chosen as the space for testing. There are 40 components installed in the pilot house. The pilot house is 19 feet long, 19 feet wide, and 8 feet high. Figure 4-4 shows the model of the TWR pilot house.

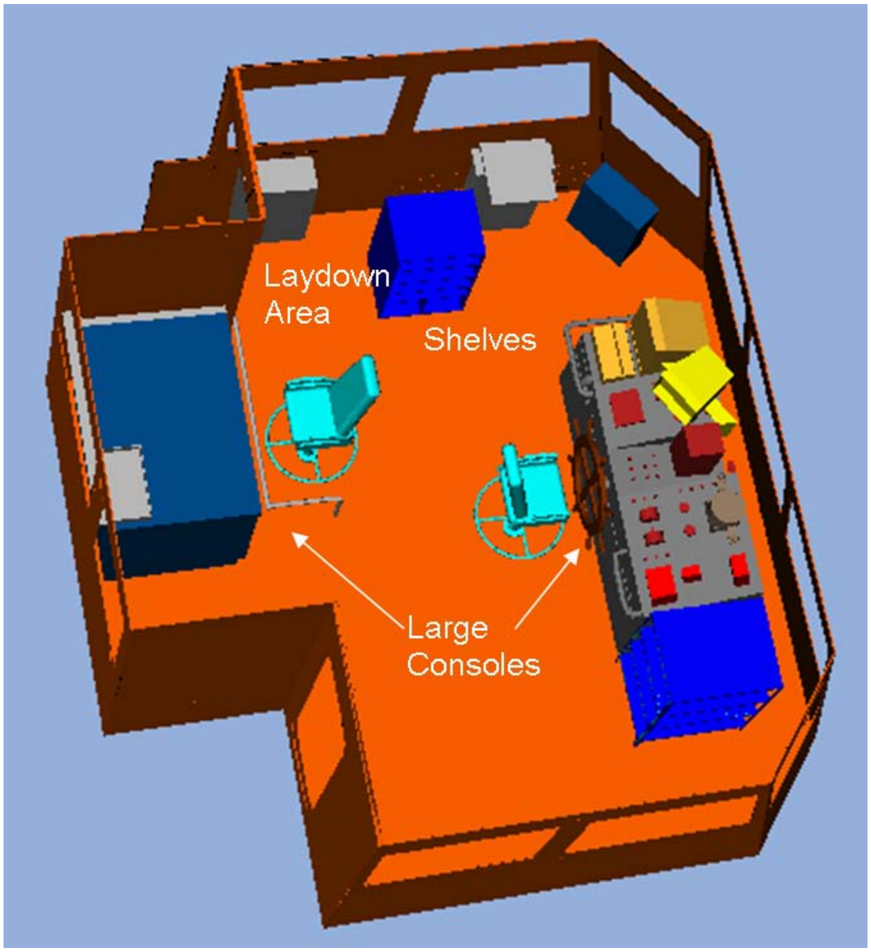


Figure 4-4: TWR Pilot House Model.

The components in the pilot house include several gauges and controls located on a large console. Several shelves are also installed along the top side of Figure 4-4. Another large console table with a railing and chair are installed on the left side of the unit. The laydown point for the components is assumed to be just inside the door in the upper left corner of the unit as indicated in Figure 4-4.

The TWR data set only included the geometry for the pilot house. Additional information was added to the model so that it could be used as a test case. An installation activity was created for each of the 40 components and the geometry was linked to the activity. The activities were arranged into a network of predecessor relationships and durations. Resource requirements of the activities and resource pool sizes were estimated by a domain expert.

The network precedence structure was constructed using the logical layout of the components. For example, the large console must be installed prior to installing the gauges and controls that reside on it. The network diagram is given in Figure 4-5**.** Although the text of the diagram is illegible, the general precedence structure is apparent. Two main activities, indicated in the diagram (1 and 2), precede several subsequent activities that have no other predecessors. This type of network indicates that several activities can be performed in parallel.

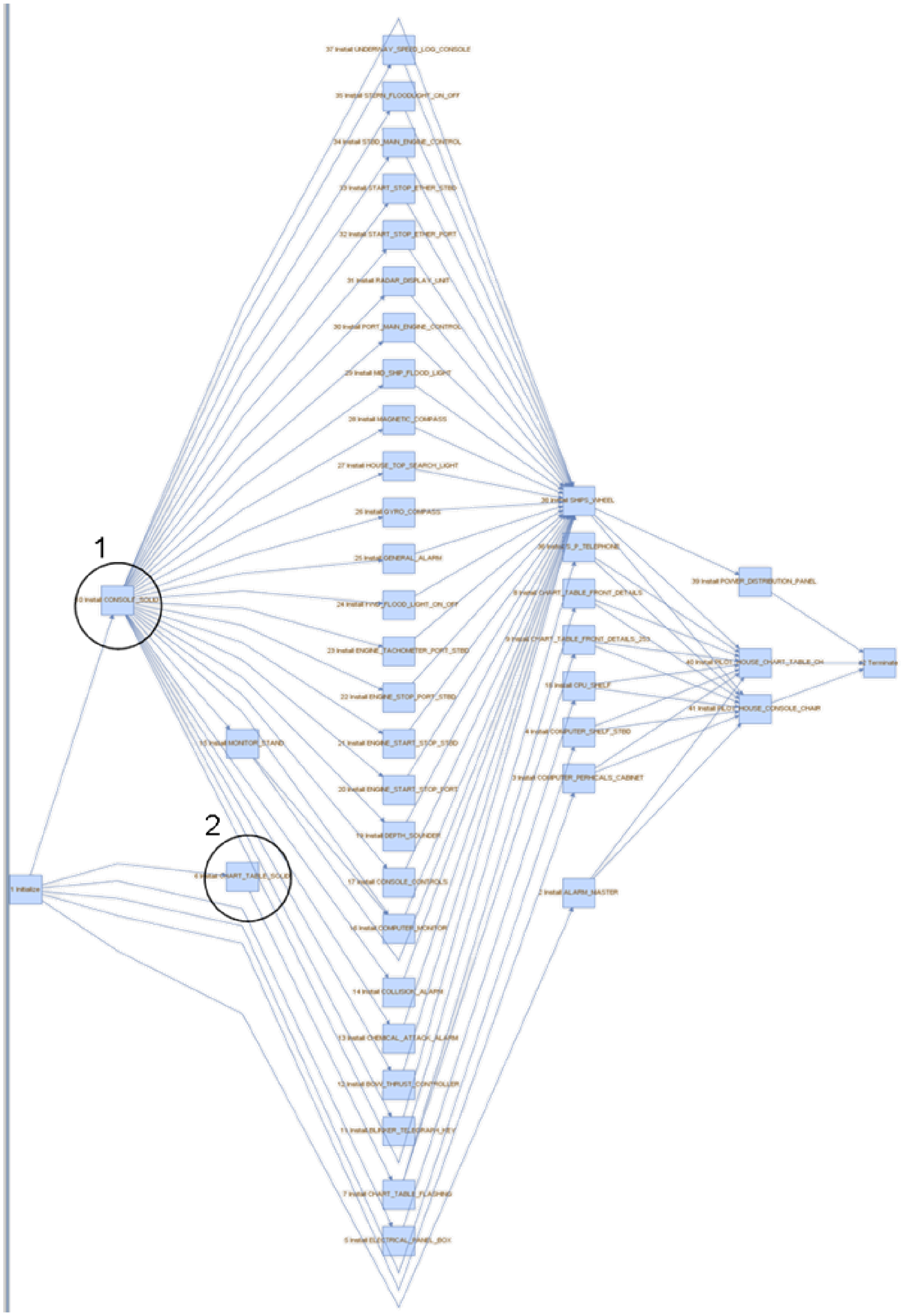


Figure 4-5: TWR Network Diagram.

Four resource types were considered for the TWR problem. Five resources of each type were available for allocation to an activity. The detailed information for this problem is given in Appendix E.

4.5.1 TWR Problem Baseline Solution

A baseline for comparison was established for the problem by solving the TWR problem using the developed methodology without considering the spatial requirements of the activities. The population size was set to 10 and the number of generations was set to 10. A two point crossover operator was used and the crossover rate was set to 0.95. Swap mutation at a rate of 0.2 was used for the mutation operator. The binary tournament with elitism and replacement was used as the selection method. The resulting near-minimum makespan was found to be 78 time units. Figure 4-6 shows the resulting Gantt chart of the schedule. The size of text in the figure is illegible due to the size of the project network. Therefore three areas of detail are presented in figures following Figure 4-6. Although the text is illegible, the structure of the schedule can be discerned. Two main activities are started at the beginning of the project, detail area 1, followed by several successor activities, detail area 3. The successor activities are only preceded by the two main activities and are only constrained by the availability of resources.

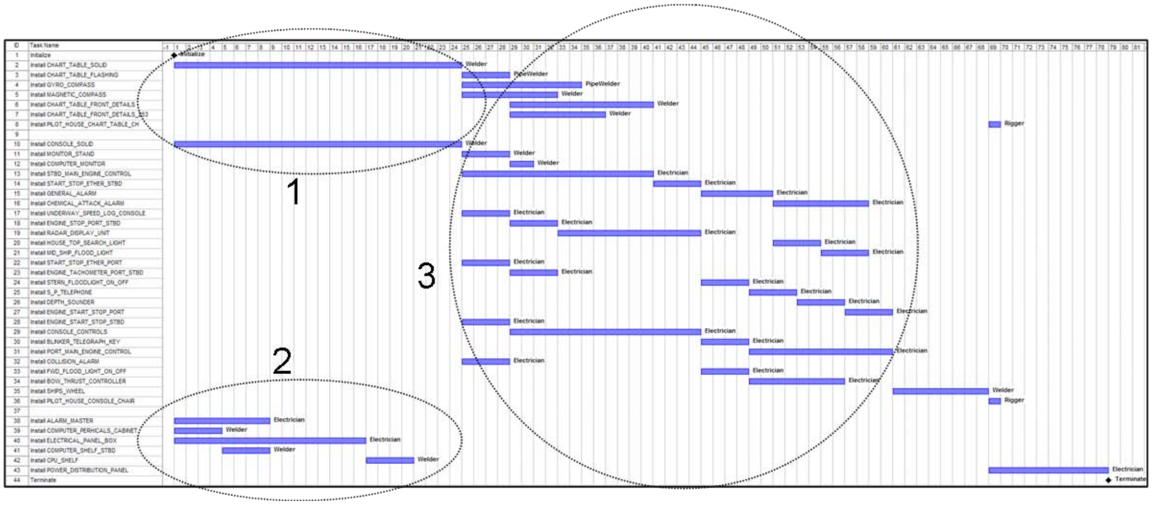


Figure 4-6: TWR Space-Unconstrained Baseline Solution Gantt Chart.

Detail area 1 from Figure 4-6 shows the two main initial activities. Figure 4-7 shows the Gantt chart section for detail area 1.

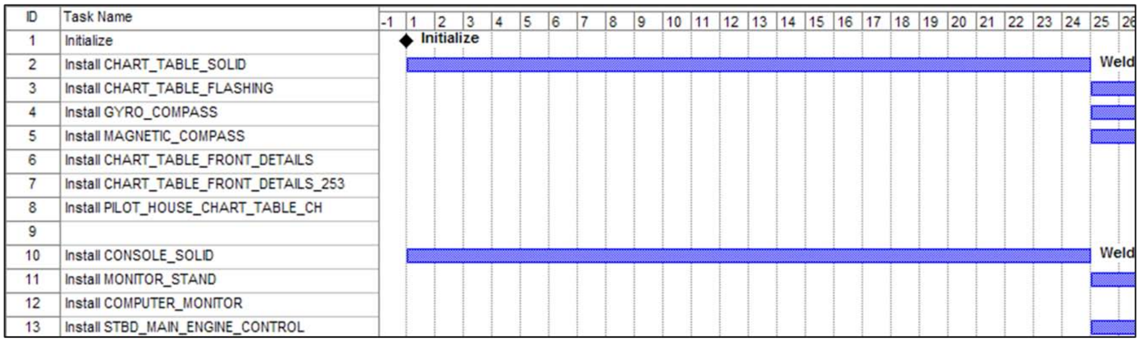


Figure 4-7: Baseline Solution Gantt Chart Detail Area 1.

The Install CHART_TABLE_SOLID and Install CONSOLE_SOLID activities are the two main precedence constraint activities.

Detail area 2 from Figure 4-6 shows activities that are not preceded by the two main activities and can thus start at the beginning of the project and are not on the critical path. The portion of the overall Gantt chart in detail area 2 is shown in Figure 4-8. The type of resource used in the activity is given at the end of activity bar in the Gantt chart.

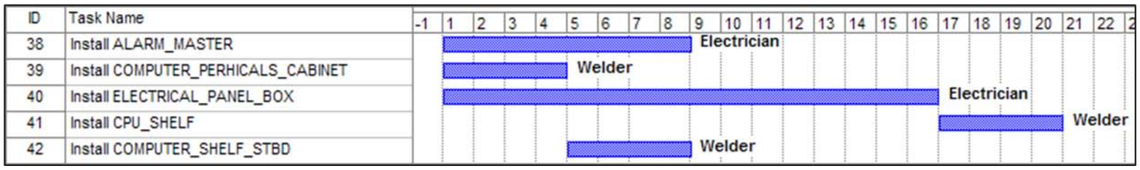


Figure 4-8: Baseline Solution Gantt Chart Detail Area 2.

The activities preceded by the two main activities in the network are shown in detail area 3 from Figure 4-6. Detail area 3 is shown in Figure 4-9.

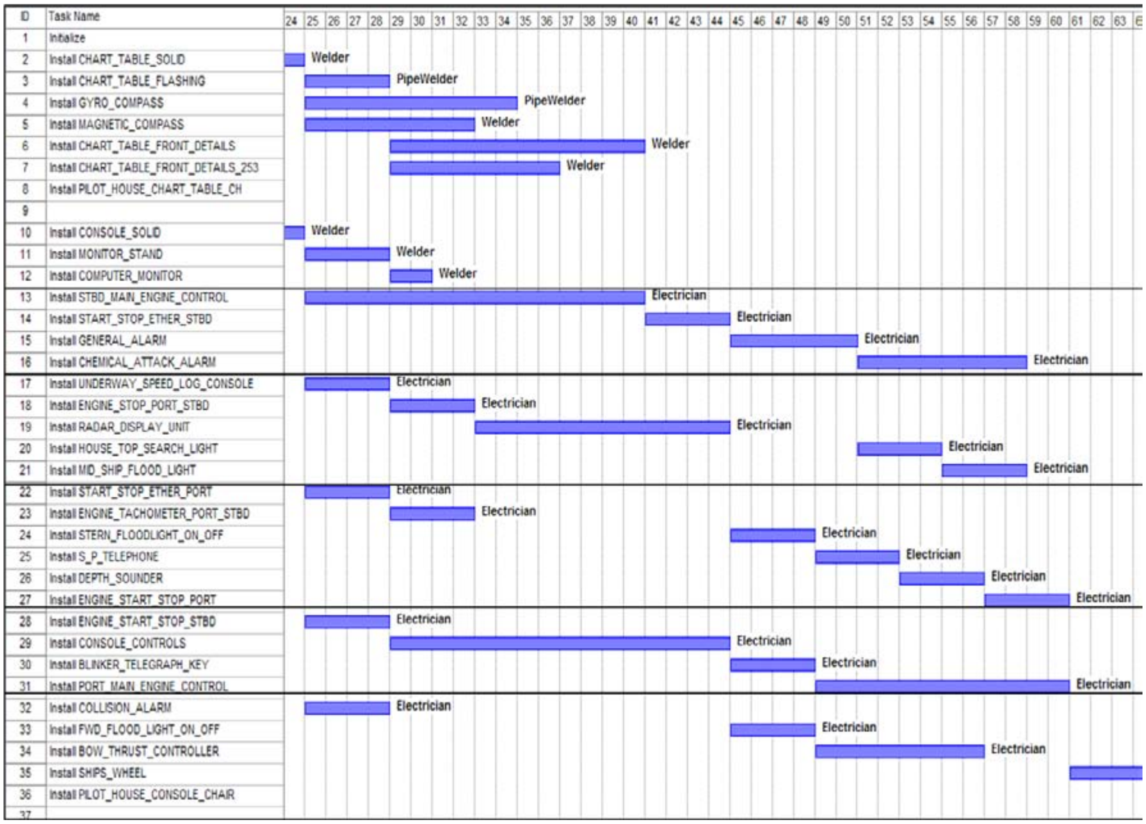


Figure 4-9: Baseline Solution Gantt Chart Detail Area 3.

The activities that are preceded by the Install CHART_TABLE_SOLID activity are shown at the top of the figure. With the resource requirements and availabilities given in the problem, they do not represent critical path activities.

The activities that succeed the Install CONSOLE_SOLID are shown on lines 11-34. Organization of these activities determines the critical path and thus the makespan of the project. The activities contained within the horizontal bars were performed by the same resource. Gaps in the work profile of a resource indicate that two resources were used to perform the activity. The resource profiles for the four types of resources are given in Figure 4-10**.**

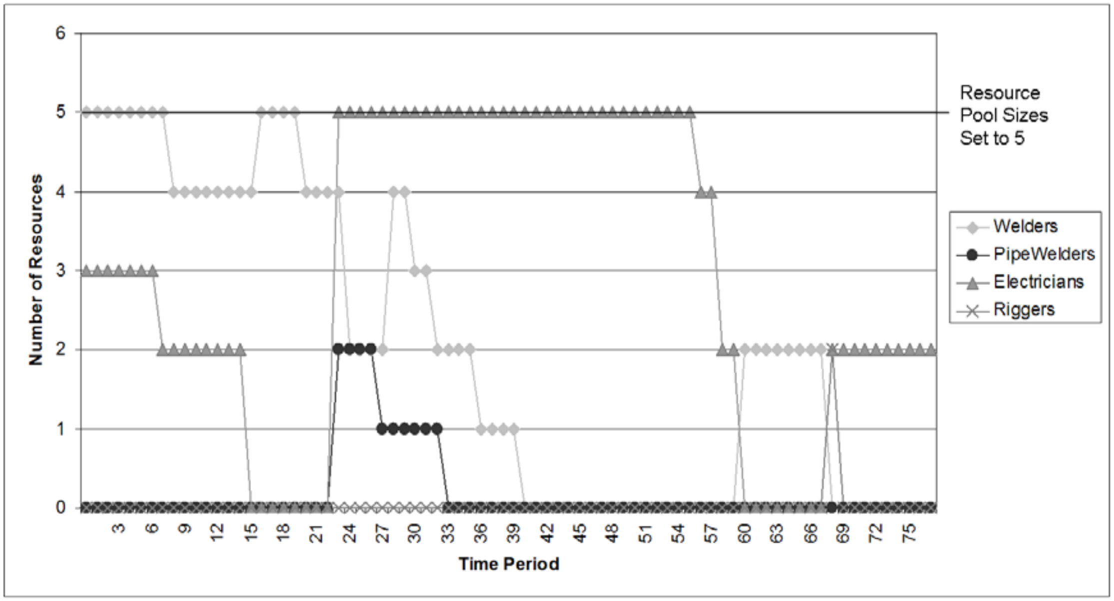


Figure 4-10: Resource Profiles for Baseline Solution.

The resource profiles show that welder resources are the constraining resource at the beginning of the schedule and that electrician resources constrain the middle of the schedule. Pipe welder and rigger resources are not constraints to the problem because the requirements for these resources do not exceed the pool limit.

4.5.2 TWR Problem Considering Space

The parameters for the problem considering space were set identically to the baseline TWR problem and the spatial requirements for the activities were added to the problem. The population size and number of generations were reduced in both TWR tests because the complexity of the component geometry causes significant increases in processing time. The component space occupies approximately 4% of the internal space of the unit. Activity space was generated for each of the activities according to the methods discussed in Chapter 3.

The problem was solved using the developed methodology and the resulting makespan was 105 time units, or 35% greater than the baseline. The duration was increased for 34 of the activities and the average increase was 3.59 time units.

The solution time was nearly 54 hours of CPU time for the 100 (population size of 10 and 10 generations) schedules evaluated. Similar to the example problems, travel path generation using the A* algorithm was responsible for the majority of the CPU time. A total of 35 activities required the A* algorithm to find a travel path.

The Gantt chart for the schedule found by the algorithm is given in Figure 4-11. The structure is similar to that presented in the baseline case. However the makespan of the project has increased. Detail areas are shown and portions of the Gantt chart have been enlarged to view the specific charts for those areas.

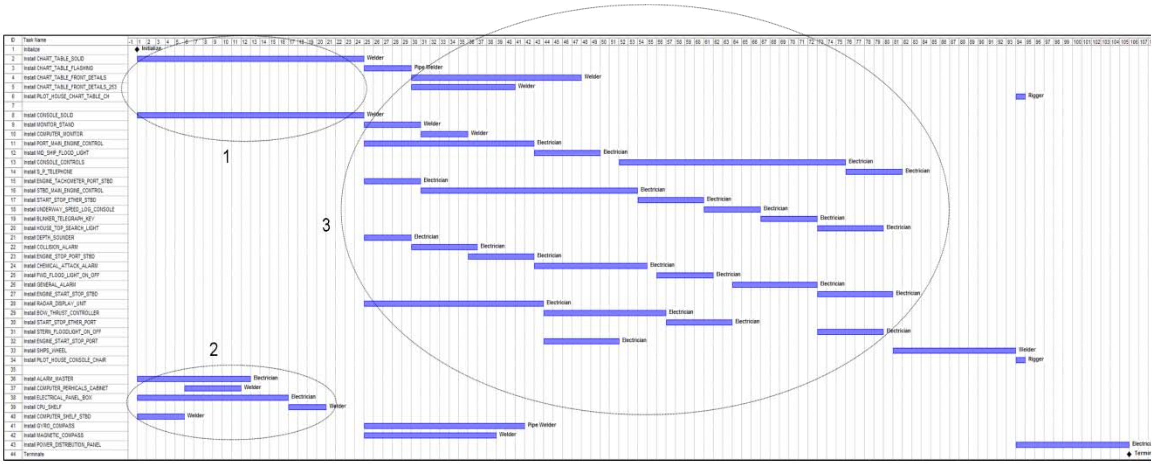
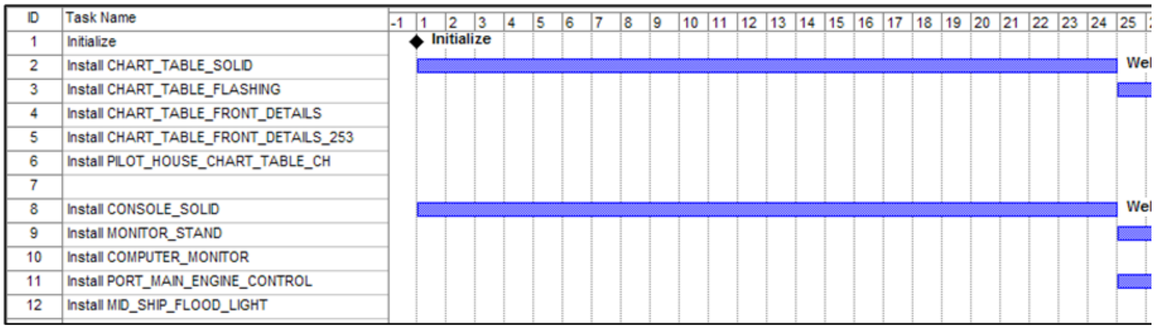


Figure 4-11: TWR Space-Constrained Baseline Solution Gantt Chart.

Detail area 1 in Figure 4-12 shows the two main activities, Install CHART_TABLE_SOLID and Install CONSOLE_SOLID. The durations for these two activities did not increase from the case where space was not considered. This was an expected result. The precedence network structure limits the number of activities that can be in process concurrently with the two main consoles. Further, the Install CONSOLE_SOLID activity is on the critical path which indicates that sequences that placed this activity prior to other allowable activities would have shorter makespans because the potential duration increase from spatial conflicts is reduced.



ID	Task Name
1	Initialize
2	Install CHART_TABLE_SOLID
3	Install CHART_TABLE_FLASHING
4	Install CHART_TABLE_FRONT_DETAILS
5	Install CHART_TABLE_FRONT_DETAILS_253
6	Install PILOT_HOUSE_CHART_TABLE_CH
7	
8	Install CONSOLE_SOLID
9	Install MONITOR_STAND
10	Install COMPUTER_MONITOR
11	Install PORT_MAIN_ENGINE_CONTROL
12	Install MID_SHIP_FLOOD_LIGHT

Figure 4-12: TWR Solution Gantt Chart Detail Area 1.

Detail area 2, shown in Figure 4-13 shows the activities that are performed concurrently to the two main activities.

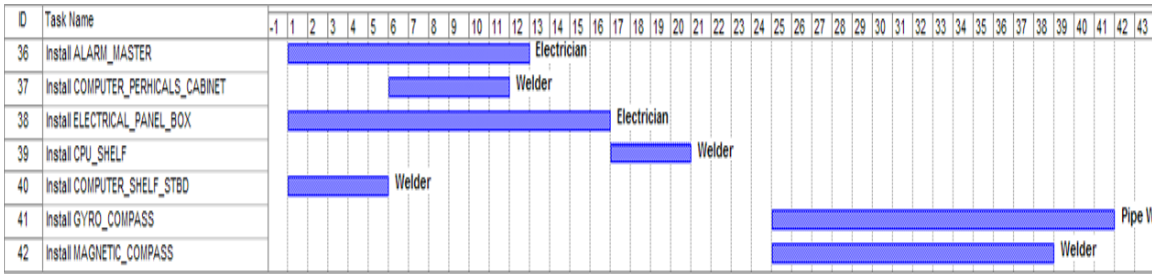


Figure 4-13: TWR Solution Gantt Chart Detail Area 2.

Of the five activities in Gantt chart that would be in process during the two main activities, lines 36-40 in Figure 4-13, three experienced duration increases due to congestion. The makespan would have been increased, if the genetic algorithm had generated a sequence in which these activities preceded the Install CONSOLE_SOLID activity.

Detail area 3 of the Gantt chart is shown in Figure 4-14. The structure of this section is similar to that found for the case where space was not considered, however the durations have been increased for almost all of the activities. These activities all occur within close proximity and it was expected that the congestion that results from this would increase the durations of the activities.

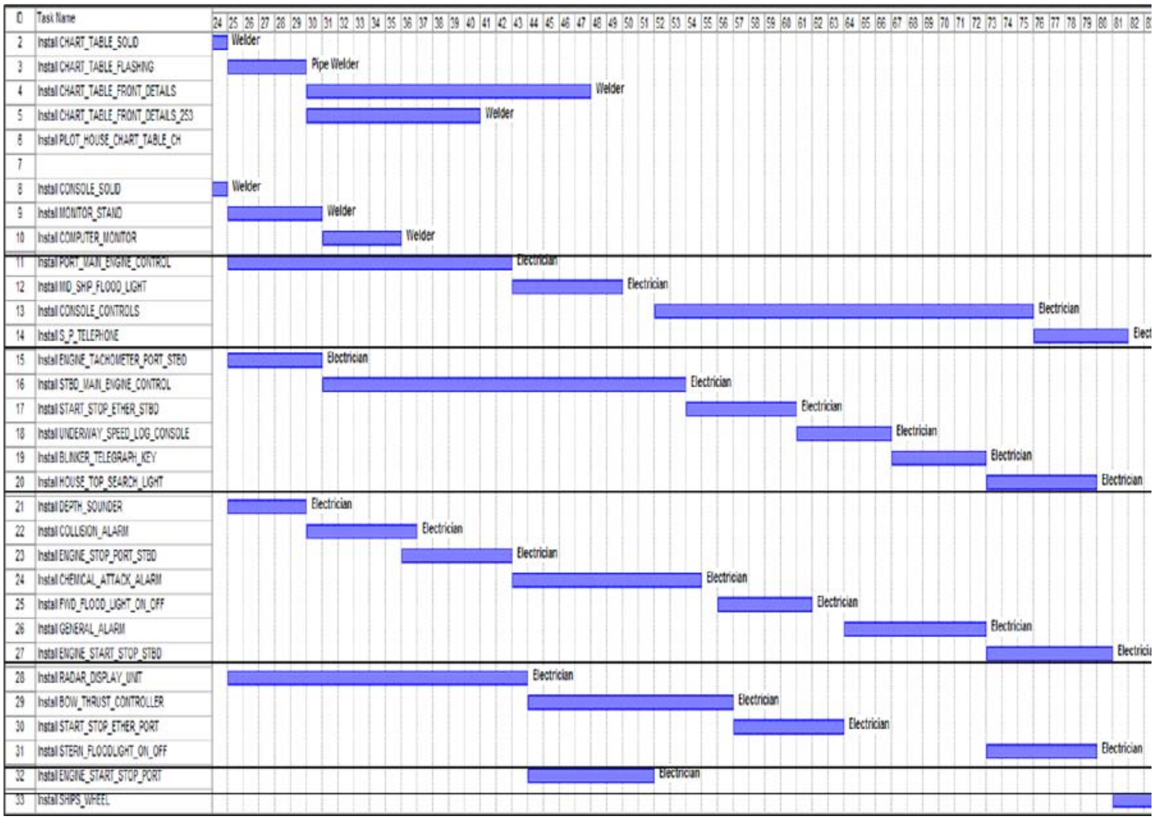


Figure 4-14: TWR Solution Gantt Chart Detail Area 3.

The duration for every activity that installed a component on the main console was increased due to congestion. The results of this test showed that the used space factor component of the congestion function was responsible for the majority of the duration increases. For these activities, the working envelope of the used space factor was occupied at almost 50% for each of these activities. The main console accounted for the majority of the occupied space. The congestion caused by execution space and work in place space also caused the durations of these activities to increase. The worker space conflicted with the main console for the majority of the activities because of the location of the worker space in relation to the installation components. Results showed that laydown and travel path space conflicts had little effect on the activity duration increases.

The resource profile for the best schedule found by the algorithm is given in Figure 4-15.

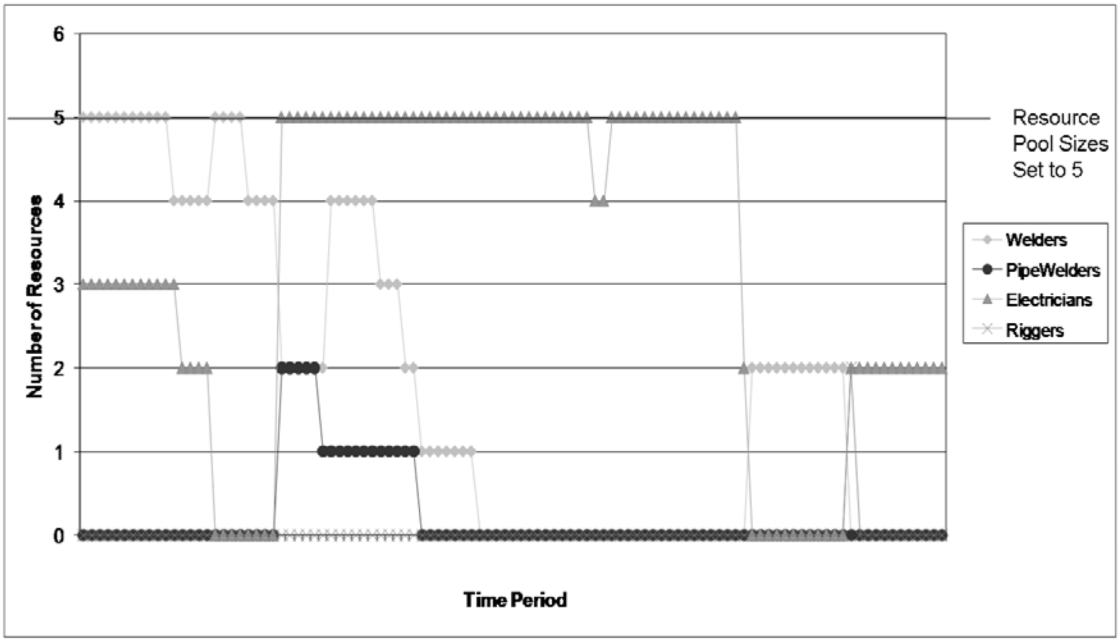


Figure 4-15: Resource Profiles for Space-Constrained Solution.

Again, similar to the profiles found for the case without space, the welder resource type is the constraining resource at the beginning of the schedule and the electrician resource type constrains the middle of the schedule. An interesting extension of this line of testing would be to vary the pool sizes of the two constraining resource types to further examine the behavior of the system. Increasing the pool sizes might have little effect on the project makespan due to the added congestion from the additional workers in the space. Likewise, decreasing the pool sizes could, in theory, shorten the makespan by removing congestion. The decrease in pool sizes of the constraining resources would reduce the number of activities that could potentially be in process simultaneously. Reducing the number of concurrently processing activities could potentially decrease congestion if those activities are being performed in close proximity. The project makespan would be shortened if decrease in productivity due to congestion was greater than the increase in productivity of having additional resources to perform the activities.

4.6 Discussion

The software implementation of the developed methodology was used to solve an example problem and a case study problem from the shipbuilding industry. The goal of solving the problems was to demonstrate the approach and to characterize the effect of space on the underlying RCPSP.

The methodology was first tested on a set of problem instances from PSLib project scheduling problem library to establish a baseline performance without considering space. This step was performed to ensure that the underlying project scheduling methodology was able to solve the RCPSP problem. In addition, the results of this experiment were used as a baseline for comparison against tests performed on the same problems when space was included. The results showed that the underlying RCPSP scheduling algorithm satisfactorily solved the problems and that there would be no issues with this component of the methodology.

The example problem was chosen randomly from the PSLib data set and modified to include spatial representations of the components that were installed. The activity network, resource requirements, and resource pool availabilities were taken directly from the problem instance. The methodology was tested at three levels of occupied space.

Results showed that, as the occupied space increases the makespan also increases. This is a result of spatial conflicts between the work in place and the spatial requirements of the activity being installed and the overall congestion function parameter, the used space factor. It can be concluded that as the amount of space consumed by installation components increases, the project makespan also increases.

When comparing the makespan of the problem without space and the problem with space, results showed increases between 7% and 16% for the example scenarios and 35% for the TWR case. Previous research shows a 30%-65% efficiency loss due to congestion (Akinci and Fischer 1998; Dawood and Mallasi 2006; Mallasi 2006). The results obtained for the example scenarios indicate a smaller increase in makespan than the literature would suggest, however the analyses that were performed in the reviewed literature did not use project makespan as the performance measure. The performance efficiency was calculated by comparing actual labor hours to standard labor hours for performing a unit of work. Using the efficiency metric found in literature the results show an efficiency loss of 4% for Scenario 1, 13% for Scenario 2, 20% for Scenario 3, and 42% for the TWR example. The results found in this study show smaller efficiency losses due to congestion for the example scenarios than the literature would suggest. The efficiency losses experienced in the TWR example are within the range of published findings. However, the discrepancy in the first test set indicates a need to perform additional research to develop information based duration increase functions.

The design used to test the methodology in the TWR case study example was similar to that of the PSLib test problems: establish a baseline without space, include space and compare the results. Results showed a 35% increase in makespan.

The network for the TWR problem has very few precedence constraints. This type of network allows several activities to be in process concurrently. In addition, many of the components were installed in close proximity to each other. The combined effect of both of these characteristics results in an increased demand for space within the unit. If the network contained more precedence relationships, the activities would be performed in a serial fashion, reducing the number of concurrent activities and the demand for space. In effect, the methodology creates a schedule that serializes the activities in cases like the TWR. The activities that occur concurrently and in close proximity to one another are delayed such that they are performed in a serialized fashion. This result is assumed to happen in practice as well. Using the developed methodology will enable a construction supervisor to understand the implications of scheduling several concurrent activities in close proximity. In addition, the methodology will find a schedule that minimizes the effect of the spatial congestion and has the shortest makespan.

4.7 Chapter Summary

This chapter discussed the software implementation of the developed methodology, described the testing procedure and provided a discussion of the results. The tests were performed to show that it can solve the RCPSP-S. Further, it was tested against a known library of problems to establish a baseline performance of the underlying RCPSP algorithm. Three tests were performed on an example problem from the PSLib, where spatial requirements for the activities were generated such that 10%, 25%, and

40% of the internal space of the unit was occupied by installation components. These tests were performed in an effort to characterize the effect of space. Results show that makespan of the project increases when space is considered. In addition, the RCPSP-S algorithm was used to solve a real world example problem of the TWR pilot house.

Chapter 5

Summary and Future Work

This chapter draws conclusions from the results of the tests performed in the previous chapter and gives some direction for future work in this area of research. Several conclusions can be made from the tests and include a discussion on the ability of the methodology to model and solve problems in this area, the performance of the methodology on a problem where the percentage of space within the unit occupied by the components varied, and the overall effect of congestion on the project makespan. There are a number of areas for future work on this problem and the developed methodology. These areas are explored to provide a line of future research on this problem.

5.1 Discussion

Project scheduling in general large-product industries is a complex task that has primarily been performed by construction supervisors. They use their knowledge and experience to make decisions often with little or no understanding of the impact of those decisions on future activities.

A methodology framework was developed to solve the project scheduling problem with spatial requirements. The framework was used as the basis for the development of a methodology that implemented specific algorithms for the various algorithmic components of the framework. The implemented methodology uses a genetic algorithm to generate sequences of activities and guide the search. A serial schedule generation scheme that considers space is used to schedule the activities. The spatial requirements of the activities are used to estimate the congestion by calculating spatial conflict volumes and translating them into a duration increase through a congestion function. The congestion function is a parameterized set of equations that can be modified according to the characteristics of the problem domain. The thesis of this work is that the true makespan of a project schedule can be calculated by modeling the space required by the activities and using a congestion function that increases the nominal activity durations based on the spatial conflicts of the activities. The main contribution to the body of research in this field is the integrated solution framework that includes activity spatial requirements into the resource-constrained project scheduling methodology.

An example problem and a real world case study were solved using the methodology to demonstrate the approach. The results showed that considering spatial constraints increased the makespan from 7% to 16% for the example problems and 35% for the TWR case study problem.

5.2 Future Work

The objective of this work was to develop a methodology to solve the RCPSP-S problem. In doing so, several assumptions and simplifications were required to bound the problem. Future work in this area of research would be to relax the assumptions by developing more robust methods of space generation, path planning, and duration increase functions. An additional future work topic would be to identify a case study example from a real world project to compare the developed methodology to the approach used in practice. To enhance the methodology developed in this thesis, six specific areas need to be addressed: linking geometry to activities, generation of execution space, defining behavior for components that enclose voids, robust path planning, duration increase functions, and comparison with a real world case study.

5.2.1 Linking Geometry to Activities

In many general large-scale product industries the design and planning functions occur separately due to the nature of the products that are being constructed and the vast number of pieces that need to be assembled to form the final product. In addition, the complexity of the product restricts how many activities there are because coordination of too many activities becomes an overwhelming task. Because of this it is often difficult to map or link the geometry to an activity. This thesis presented a simplistic method for linking geometry to activities knowing that in most cases the activity name calls out the piece to be installed. However, in practice, this relationship is not always trivial.

A method is needed to algorithmically map the component geometry to the installation activity. Initially, this could be an enhanced graphical user interface that enables the user to ‘click and drag’ geometry to an activity for association. This is still a manual process however.

During the design phase, designers create a product model, geometry and many attributes that relate the geometry to other components within the design. Recent design efforts have centered around adding planning and scheduling attributes to the product models. With the addition of these attributes, a methodology could be developed to use them to automatically generate the information required to link that geometry to the activity responsible for installing that component. Research would need to be performed to find the current state of product model attribution, identify the additional, if any, attributes that would be needed and include them in the design process. Additional research would be needed to interpret those attributes and create a methodology to link the geometry to the activity.

5.2.2 Execution Space Generation

Generation of activity spaces is a key component to this research. In this thesis, a simple method was used to generate execution space. It was assumed that a nominal amount of space would be required around the component for installation and that the worker space would be located 'in front' of the component and 'to the side' of the component if two workers were needed. This execution space generation approach is applied to all components regardless of activity type, location, or duration. In addition, the generated space is held constant for the duration of the activity.

Further research in execution space generation should provide an approach that models the true execution space required by an activity. Closely observing the installation practices in the field should provide further definition of the problem and provide a set of production rules that are generally followed in practice. A rules-based system could be developed that would generate the required space for a given activity based on the activity type, size of the component, number of required workers, location in the unit, location of other activities, and duration.

Execution space for an activity in practice is a dynamic space that changes over the duration of the activity. An approach that models the dynamic nature of the execution space would complement the methodology developed here. The rules-based system developed for the general execution space could be extended to include a probability distribution of space occupation over time. A method based on the probability that the space is being occupied by the worker could be integrated into the current work by extending the spatial conflict models and updating the congestion function equations.

5.2.3 Components Enclosing Voids

Components such as lockers, cabinets and shelves have open spaces within the boundaries of their geometry that, in theory, could be used by a worker. Using the example discussed in Section 3.5.4, if a locker component is currently in place and the current activity is installing a component adjacent to that locker, the execution space of the current activity may overlap with the interior of the locker. The current approach considers only the solid volume of the sheet metal comprising the locker, so the volume of the intersection would be very small. In practice, this assumption may or may not hold. If the void is completely enclosed within the geometry, then the overlap should be considered as a work in place conflict with execution space. However, if the void is open on one side, then it is possible that there will be less conflict.

Further research should focus on defining and characterizing void spaces and developing rules to control their effect on congestion. The integration of an approach for void spaces into the current methodology would involve additional development in two areas: 1) identification of void areas during initialization and 2) modification of the congestion function to estimate the impact of void space overlaps on the duration of the activities.

5.2.4 Travel Path Planning

The current travel path space generation approach begins with a simple swept volume approach and moves to an A* search procedure if there are spatial conflicts with the swept volume and work in place or execution space. The travel path planning approach used in this thesis suffers from three key limiting issues: static component orientation, nominal rotation analysis during travel path generation, and it is computationally intractable for even small test problems.

The orientation of the components is assumed to be the same as the installation orientation in the swept volume approach and is allowed to rotate only by 45 and 90 degrees around the vertical axis during the search procedure. The orientation assumption holds when the dimensions of the components are relatively similar, i.e. cubes or short pipe sections. However, long pipes and other components that have one dimension that is significantly greater than the others cause the simple swept volume method to be significantly inaccurate or even invalid.

When the A* search procedure is used, the current procedure allows very limited options for reorienting the component. Additional research is needed to identify an initial orientation of the component that minimizes the need to use the search procedure to find a travel path. Path planning is an open area of research and has a significant body of knowledge from which to draw upon. Complex search procedures exist for three-dimensional rotation of components and should be investigated and integrated into this methodology.

In addition, the material handling mechanism used to move the component from the laydown point to the final installation location should be considered during travel path generation. The material handling mechanism will affect the range of possible orientations, as well as the space required to move the component into place.

The computation time of the A* search algorithm was discussed in Section 4.4 where an informal accounting of the overall solution time revealed that the majority of the solution time was spent generating the travel path using the A* algorithm. Although many techniques (hierarchical grid sizing, bounding the grid spacing, and nominal rotation during the search) were employed, the current generation method is not computationally tractable and further research is needed to reduce the computation time of travel path planning.

To fully utilize the methodology developed in this thesis a fast travel path generation algorithm that considers three-dimensional rotation, alternative component orientations and material handling devices should be developed and implemented.

5.2.5 Duration Increase Functions

The duration increase functions relate spatial conflicts to a duration increase. The method used to create the functions in this work presented a table with known values of spatial conflict and the author estimated the relative duration increase for each value. This approach places a significant amount of cognitive burden on the user by asking them to relate two things that are not normally related. Even if a shipbuilding expert, for example a construction supervisor, were presented with the table, it would be quite difficult to estimate the effects of two volumes of space on the duration of the activity.

A method to further define the duration increase functions should be the focus of future research. This method would initially request information from construction supervisors similar to the approach used in this work. The gathered information would provide for further definition of the relationships between conflict volumes and duration increases. In addition, historical data and information should be collected and used define the functions. The revised functions should then be validated on a current installation project. The functions developed in this proposed method would be industry specific, but the development approach could be applied in any general large-product industry.

5.2.6 Real-World Case Study

There are no published methods that can be used to directly compare with the performance of the methodology developed in this thesis. For this reason, the TWR case study presented in Chapter 4 was used to demonstrate the methodology, but lacked a true comparison between the developed method and current practice. To fully demonstrate the capability of the developed methodology a real world case study problem should be identified and used for comparison. An example of a case study problem for comparison would be a recently completed shipbuilding unit. The historical records of the project would need to be analyzed and informational interviews of the managers and decision makers on the project would need to be performed in order to understand the current practice. In addition, the geometry, activity and resource information would need to be collected and used as inputs to the developed methodology.

An obvious comparison point would simply be the makespan of the completed project and the makespan found by the solution methodology. However, the actual durations of the activities and sequencing decisions made by the shop floor management could also be compared.

Although the purpose of the case study problem would be for comparing the two methods, additional enhancements to the developed methodology could be identified and included as future work.

5.3 Summary

Although the motivating example for this research was the shipbuilding industry, the methodology could be applied in other general large-product industries, such as the building construction or refinery construction. This work would be especially applicable to confined spaces where the activity network allows several activities to be performed concurrently and detailed installation schedules are needed to minimize the congestion.

The opportunity for improving installation planning is just beginning in the large product industry, but has been the topic of active research in the building construction industry for some time (Akinci and Fischer 1998). The shipbuilding industry will be able to extend the current research through the use of detailed 3D product models developed in ship design, which are often lacking in the construction industry.

Riley and Sanvido (1997) state that, "Support tools to assist with space planning must be developed to minimize the planning effort and potential downstream costs of detail planning." The developed methodology is a support tool to help planners and construction managers identify congestion within the workspace and develop schedules that minimize the congestion and approach minimal makespan.

Akinci, Fischer et al. (2002c) cite the need for a method that has an enhanced representation of complex shapes. In most cases, the geometry from the CAD system is used in this thesis. Simplification of the geometry by bounding box representation was performed to control the computation time of the algorithm.

The main research contribution of this work is the integrated schedule generation framework that considers the actual geometry of the installation components and close approximations of the activity space requirements. The congestion function approach developed in this thesis, although similar to that found in literature (Thabet and Beliveau 1997), extends that work by integrating critical space research (Mallasi 2006). This integration and extension resulted in a congestion function that is a composite function comprised of congestion components from the various space type conflicts.

Bibliography

Adjei-Kumi, T. and A. Retik (1997). Library-based 4D visualization of construction processes, London, Engl, IEEE.

Akinci, B. and M. Fischer (1998). Time-space conflict analysis based on 4D production models, Boston, MA, USA, ASCE, Reston, VA, USA.

Akinci, B. and M. Fischer (2000). 4D workplanner - A prototype system for automated generation of construction spaces and analysis of time-space conflicts, Stanford, CA, United States, American Society of Civil Engineers.

Akinci, B., M. Fischer, et al. (2002a). "Automated generation of work spaces required by construction activities." Journal of Construction Engineering and Management **128**(4): 306.

Akinci, B., M. Fischer, et al. (2002b). "Representing work spaces generically in construction method models." Journal of Construction Engineering and Management **128**(4): 296.

Akinci, B., M. Fischer, et al. (2002c). "Formalization and automation of time-space conflict analysis." Journal of Computing in Civil Engineering **16**(2): 124.

Baker, K.R. (2001). Elements of sequencing and scheduling. Hanover: Tuck School of Business, Dartmouth College.

Balby, D. (2009). UnBBoolean, http://sourceforge.net/projects/unbboolean/**:** Open source library for CAD operations using Java3D.

Beedall, R. "Future aircraft carrier (CVF) Queen Elizabeth Class: Part 19." Navy Matters. N.p. 16 Jan. 2009. 15 March 2010 < http://navy-matters.beedall.com/cvf1-19.htm>.

Benthall, L., T. Briggs, et al. (2004). "Enabling interoperability between US shipyards." Journal of Ship Production **20**(2): 90.

Bischoff, E. E., F. Janetz, et al. (1995). "Loading pallets with non-identical items." European Journal of Operational Research **84**: 681-692.

Blazewicz, J., J. K. Lenstra, et al. (1983). "Scheduling subject to resource constraints: classification and complexity." Discrete Applied Mathematics **5**(1): 11.

Brucker, P., A. Drexl, et al. (1999). "Resource-constrained project scheduling: Notation, classification, models, and methods." European Journal of Operational Research **112**(1): 3.

Brucker, P. and A. Kraemer (1996). "Polynomial algorithms for resource-constrained and multiprocessor task scheduling problems." European Journal of Operational Research **90**(2): 214.

Cahill, P.D. (2006). "CAD-plan connector." Journal of Ship Production **22**(3): 113.

Chau, K. W., M. Anson, et al. (2005). "4D dynamic construction management and visualization software: 2. Site trial." Automation in Construction **14**(4): 525.

Chau, K. W., M. Anson, et al. (2003). "Implementation of visualization as planning and scheduling tool in construction." Building and Environment **38**(5): 713.

Chau, K. W., M. Anson, et al. (2004). "Four-dimensional visualization of construction scheduling and site utilization." Journal of Construction Engineering and Management **130**(4): 598.

Chevallier, N. and A. D. Russell (1998). "Automated schedule generation." Canadian Journal of Civil Engineering **25**(6): 1059.

Chirillo, L. D. (1984). Product Work Breakdown and Statistical Analysis: Prerequisites for Robotics in Shipbuilding. Technical Paper - Society of Manufacturing Engineers**.**

Choo, H. J., I. D. Tommelein, et al. (1999). "WorkPlan: Constraint-based database for work package scheduling." Journal of Construction Engineering and Management **125**(3): 151.

Dain, O. M., D. W. Etherington, et al. (2005). Automated Scheduling to Minimize Shipbuilding Cost. The Twelfth International Conference on Computer Applications in Shipbuilding, Busang, Korea.

Dawood, N. and Z. Mallasi (2006). "Construction workspace planning: Assignment and analysis utilizing 4D visualization technologies." Computer-Aided Civil and Infrastructure Engineering **21**(7): 498.

Dawood, N. and E. Sriprasert (2006). "Construction scheduling using multi-constraint and genetic algorithms approach." Construction Management and Economics **24**(1): 19.

De Frene, E., D. Schatteman, et al. (2007). A Heuristic Methodology for Solving Spatial Resource-constrained Project Scheduling Problems, Kotholieke Universiteit Leuven**:** 32.

Dodin, B. and A. A. Elimam (2008). "Integration of equipment planning and project scheduling." European Journal of Operational Research **184**(3): 962.

Dorndorf, U. (2002). Project Scheduling with Time Windows: From Theory to Applications. New York, NY, Physica-Verlag Heidelbberg New York.

Durillo, J. J., A. J. Nebro, et al. (2006). {jMetal}: A Java Framework for Developing Multi-Objective Optimization Metaheuristics, Departamento de Lenguajes y Ciencias de la Computacion, University of Malaga.

Dyckhoff, H. (1990). "A Typology of Cutting and Packing Problems." European Journal of Operational Research **44**: 145-159.

East, E. W. and L. Y. Liu (2006). "Multiproject planning and resource controls for facility management." Journal of Construction Engineering and Management **132**(12): 1294.

Finke, D. A., C. B. Ligetti, et al. (2006). Shipyard Space Allocation and Scheduling. 2006 SNAME Maritime Technology Conference & Expo and Ship Production Symposium, Fort Lauderdale, FL.

Gischner, B., P. Lazo, et al. (2006). "Enhancing interoperability throughout the design and manufacturing process." Journal of Ship Production **22**(3): 172.

Goldratt, E. M. (1997). Critical chain. Great Barrington, MA, North River Press.

Guo, S.-J. (2002). "Identification and resolution of work space conflicts in building construction." Journal of Construction Engineering and Management **128**(4): 287.

Hart, P. E., N. J. Nilsson, et al. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths." IEEE Transactions on Systems Science and Cybernetics **4**(2): 100-107.

Hartmann, S. (1998). "Competitive genetic algorithm for resource-constrained project scheduling." Naval Research Logistics **45**(7): 733.

Hartmann, S. (2002). "A self-adapting genetic algorithm for project scheduling under resource constraints." Naval Research Logistics **49**(5): 433.

Hartmann, S. and R. Kolisch (2000). "Experimental evaluation of state-of-the-art heuristics for the resource-constrained project scheduling problem." European Journal of Operational Research **127**(2): 394.

Hastings, J., J. Kibiloski, et al. (2003). "Four-dimensional modeling to support construction planning of the stata center project." Leadership and Management in Engineering **3**(2): 86.

Heesom, D. and L. Mahdjoubi (2004). "Trends of 4D CAD applications for construction planning." Construction Management and Economics **22**(2): 171.

Horman, M. J., M. P. Orosz, et al. (2006). "Sequence planning for electrical construction." Journal of Construction Engineering and Management **132**(4): 363.

Hurink, J. and J. Keuchel (2001). "Local search algorithms for a single-machine scheduling problem with positive and negative time-lags." Discrete Applied Mathematics **112**(1-3): 179.

Issa, R. R. A., I. Flood, et al., Eds. 4D CAD and Visualization in Construction: Developments and Applications. Lisse, A.A. Balkema, 2003.

Jaafari, A., K. K. Manivong, et al. (2001). "VIRCON: interactive system for teaching construction management." Journal of Construction Engineering and Management **127**(1): 66.

Jain, A., P. K. Jain, et al. (2006). "An integrated scheme for process planning and scheduling in FMS." International Journal of Advanced Manufacturing Technology **30**(11-12): 1111.

Kang, J. H., S. D. Anderson, et al. (2007). "Empirical study on the merit of web-based 4D visualization in collaborative construction planning and scheduling." Journal of Construction Engineering and Management **133**(6): 447.

Kelley, J. (1963). The critical-path method: Resources planning and scheduling. Englewood Cliffs, NJ, Prentice-Hall.

Kim, J.-L. and R. D. Ellis, Jr. (2008). "Permutation-Based Elitist Genetic Algorithm for Optimization od Large-Sized Resource-Constrained Project Scheduling." Journal of Construction Engineering and Management **134**(11): 904-913.

Kolisch, R. (2000). "Integrated scheduling, assembly area- and part-assignment for large-scale, make-to-order assemblies." International Journal of Production Economics **64**(1-3): 127.

Kolisch, R. and K. Hess (2000). "Efficient Methods for Scheduling Make-to-order Assemblies Under Resource, Assembly Area and Part Availablity Constraints." International Journal of Production Research **38**(1): 207-228.

Kolisch, R. and A. Sprecher (1997). "PSPLIB - a project scheduling problem library." European Journal of Operational Research **96**(1): 205.

Koo, B. and M. Fischer (2000). "Feasibility study of 4D CAD in commercial construction." Journal of Construction Engineering and Management **126**(4): 251.

Kumanan, S., G. Jegan Jose, et al. (2006). "Multi-project scheduling using an heuristic and a genetic algorithm." International Journal of Advanced Manufacturing Technology **31**(3-4): 360.

Lee, J. K., K. J. Lee, et al. (1995). "DAS intelligent scheduling systems for shipbuilding." AI Magazine **16**(4): 78.

Lee, K. J. and J. K. Lee (1996). "Spatial scheduling system and its application to shipbuilding: DAS- CURVE." Expert Systems with Applications **10**(3-4): 311.

Lee, K. J., J. K. Lee, et al. (1996). "A Spatial Scheduling System and its Application to Shipbuilding: DAS-CURVE." Expert Systems with Applications **10**(3/4): 311-324.

Li, Y., K. F. Man, et al. (2000). "Genetic Algorithm to Production Planning and Scheduling Problems for Manufacturing Systems." Production Planning & Control **11**(5): 443-458.

Lim, A., B. Rodrigues, et al. (2005). "3-D container packing heuristics." Applied Intelligence **22**(2): 125.

Mallasi, Z. (2006). "Dynamic quantification and analysis of the construction workspace congestion utilising 4D visualisation." Automation in Construction **15**(5): 640.

McKendall, A. R., J. S. Noble, et al. (2005). "Simulated annealing heuristics for managing resources during planned outages at electric power plants." Computers & Operations Research **32**(1): 107.

McKendall, J. A. R. and J. R. Jaramillo (2006). "A tabu search heuristic for the dynamic space allocation problem." Computers & Operations Research **33**(3): 768.

McKinney, K. and M. Fischer (1998). "Generating, evaluating and visualizing construction schedules with CAD tools." Automation in Construction **7**(6): 433.

Metaxiotis, K. S., D. Askounis, et al. (2002). "Expert systems in production planning and scheduling: A state-of-the-art survey." Journal of Intelligent Manufacturing **13**: 253-260.

Mitchell, M. An Introduction to Genetic Algorithms. Cambridge, Massachusetts, The MIT Press, Massachusetts Institute of Technology, 1996.

Miyazawa, F. K. and Y. Wakabayashi (2007). "Two- and three-dimensional parametric packing." Computers and Operations Research **34**(9): 2589.

Park, K., K. Lee, et al. (1996). "Modeling and solving the spatial block scheduling problem in a shipbuilding company." Computers & Industrial Engineering **30**(3): 357.

Raj, P. and R. K. Srivastava (2007). Analytical and heuristic approaches for solving the spatial scheduling problem, Singapore, Inst. of Elec. and Elec. Eng. Computer Society.

Riley, D. R. (1994). Modeling the Space Behavior of Construction Activities. Architectural Engineering, The Pennsylvania State University. **Ph.D.:** 295.

Riley, D. R. and V. E. Sanvido (1997). "Space Planning Method for Multistory Building Construction." Journal of Construction Engineering and Management **123**(2): 171-180.

Riley, D. R. and I. D. Tommelein (1996). "Space planning tools for multi-story construction." Congress on Computing in Civil Engineering, Proceedings: 718.

Słowiński, R. and J. Węglarz, Eds. Advances in Project Scheduling. Studies in Production and Engineering Economics. New York, Elsevier Science Publishers B.V., 1989.

Sprecher, A., R. Kolisch, et al. (1995). "Semi-active, active, and non-delay schedules for the resource-constrained project scheduling problem." European Journal of Operational Research **80**(1): 94.

Tamaki, H., E. Nishino, et al. (1999). "A Genetic Algorithm Approach to Multi-Objective Scheduling Problems with Earliness and Tardiness Penalties." IEEE **9**: 46-52.

Tan, B., J. I. Messner, et al. (2005). Planning construction trade flow: A visual process, Cancun, Mexico, American Society of Civil Engineers.

Thabet, W. Y. and Y. J. Beliveau (1997). "SCaRC: Space-Constrained Resource-Constrained Scheduling System." Journal of Computing in Civil Engineering **11**(1): 48.

Thomas, H. R., D. R. Riley, et al. (2006). "Fundamental principles for avoiding congested work areas-A case study." Practice Periodical on Structural Design and Construction **11**(4): 197.

Tommelein, I. D. and P. P. Zouein (1993). "Interactive dynamic layout planning." Journal of Construction Engineering and Management **119**(2): 266.

Varghese, R. and Y. Duck Young (2005). Dynamic spatial block arrangement scheduling in shipbuilding industry using genetic algorithm, Piscataway, NJ, USA, IEEE.

Wang, H. J., J. P. Zhang, et al. (2004). "4D dynamic management for construction planning and resource utilization." Automation in Construction **13**(5 SPEC ISS): 575.

Webb, R. M., J. Smallwood, et al. (2004). "The potential of 4D CAD as a tool for construction management." Journal of Construction Research **5**(1): 43.

Weglarz, J., Ed. (1998). Project Scheduling: Recent Models, Algorithms and Applications. International Series in Operations Research & Management Science. Norwell, MA, Kluwer Academic Publishers.

Wei, Y., U. Nienhuis, et al. (2009). Two Approaches to Scheduling Outfitting Processes in Shipbuilding. SNAME Annual Meeting & Expo 2009. Providence, RI.

Westerlund, J., L. G. Papageorgiou, et al. (2007). "A MILP model for N-dimensional allocation." Computers and Chemical Engineering **31**(12): 1702.

Winch, G. M. and S. North (2006). "Critical space analysis." Journal of Construction Engineering and Management **132**(5): 473.

Wuliang, P. and W. Yonghe (2008). PSO for solving RCPSP, Piscataway, NJ, USA, IEEE.

Yang, Y., F. Zhao, et al. (2005). Integration of process planning and production scheduling with particle swarm optimization (PSO) algorithm and fuzzy inference systems, Chongging, China, SPIE.

Zhang, D. and H. C. Zhang (1999). "Simulation study of an object-oriented integration testbed for process planning and production scheduling." International Journal of Flexible Manufacturing Systems **11**(1): 19.

Zhang, Y. F., A. N. Saravanan, et al. (2003). "Integration of process planning and scheduling by exploring the flexibility of process planning." International Journal of Production Research **41**(3): 611.

Zhao, C. and Z. Wu (2000). "A Genetic Algorithm for Manufacturing Cell Formation with Multiple Routes and Multiple Objectives." International Journal of Production Research **38**(2): 385-395.

Zouein, P. P. and I. D. Tommelein (1999). "Dynamic layout planning using a hybrid incremental solution method." Journal of Construction Engineering and Management **125**(6): 400.

Appendix A

Congestion Function Component Generation

This appendix provides a detailed summary of the congestion function parameter equations. Each conflict type is presented with the table of values estimated by the author. A regression model for each component was fit and the supporting statistical information is provided.

A.1 Execution Space and Work in Place Spatial Conflict Congestion Function Component

Table A-1 shows the data used to generate the function used for the conflicts arising due to execution space and work in place. The Percent 4D Conflict column represents the percentage of volume in conflict between execution space and work in place over the time that the activity currently being scheduled is in process. The Percent Duration Increase column is the resulting congestion effect of the conflict volume amount.

Table A-1: Execution Space and Work in Place Conflict Effect Values.

Nominal Percent 4D Conflict Levels (%)	Estimated Percent Duration Increase (%)
0	0
25	2.5
50	62.5
75	110
100	160

The resulting regression analysis was performed using Minitab© 15 and is given as follows:

Regression Analysis: Percent Duration versus P4DC, P4DC Sqrd, P4DC cubed

```
The regression equation is
Percent Duration Increase = 2.32 P4DC - 3.12 P4DC Sqrd + 2.21 P4DC cubed


Predictor       Coef  SE Coef      T      P
Noconstant
P4DC          2.3246   0.6307   3.69  0.010
P4DC Sqrd    -3.1220   0.9311  -3.35  0.015
P4DC cubed    2.2147   0.3272   6.77  0.001


S = 0.220096


Analysis of Variance

Source          DF       SS      MS        F      P
Regression       3  162.756  54.252  1119.93  0.000
Residual Error   6    0.291   0.048
Total            9  163.046


Source      DF   Seq SS
P4DC         1  140.378
P4DC Sqrd    1   20.159
P4DC cubed   1    2.219
```

A.2 General Used Space

Table A-2 shows the data used to generate the function used for the used space component.

Table A-2: General Used Space Congestion Effect Values.

Nominal Percent 4D Conflict Levels (%)	Estimated Percent Duration Increase (%)
0	0
25	6.25
50	25
75	50
100	100

The resulting regression takes the following form:

Regression Analysis: Percent Duration versus PC, PC Sqrd, PC cubed

```
The regression equation is
Percent Duration Increase = 4.24 PC - 9.14 PC Sqrd + 5.46 PC cubed


Predictor      Coef  SE Coef      T      P
Noconstant
PC            4.239    1.354   3.13  0.020
PC Sqrd      -9.136    1.999  -4.57  0.004
PC cubed     5.4566   0.7024   7.77  0.000


S = 0.472405


Analysis of Variance

Source          DF      SS      MS       F      P
Regression       3  339.98  113.33  507.81  0.000
Residual Error   6    1.34    0.22
Total            9  341.32


Source      DF  Seq SS
PC           1  246.10
PC Sqrd      1   80.41
PC cubed     1   13.47
```

A.3 Execution Space and Execution Space Conflict Congestion Function Component

Table A-3 shows the data used to generate the function used for the conflicts arising due to execution space of the current activity and execution spaces of other concurrent activities.

Table A-3: Execution Space Conflicts with Execution Space Effect Values.

Nominal Percent Conflict Levels	Estimated Duration Increase (nominal time units)
0	0
25	0.02
50	0.5
75	0.85
100	1.25

The resulting regression takes the following form:

Regression Analysis: Percent Duration versus P4DC, P4DC Sqrd, P4DC cubed

```
The regression equation is
Percent Duration Increase = 1.84 P4DC - 2.54 P4DC Sqrd + 1.80 P4DC
cubed


Predictor      Coef  SE Coef      T      P
Noconstant
P4DC         1.8430   0.4770   3.86  0.008
P4DC Sqrd   -2.5363   0.7042  -3.60  0.011
P4DC cubed   1.7968   0.2475   7.26  0.000


S = 0.166457


Analysis of Variance

Source          DF       SS      MS        F      P
Regression       3  103.792  34.597  1248.64  0.000
Residual Error   6    0.166   0.028
Total            9  103.958


Source      DF  Seq SS
P4DC         1  89.100
```

```
P4DC Sqrd    1  13.231
P4DC cubed   1   1.460
```

A.4 Laydown Space and Work in Place Conflict Congestion Function Component

Table A-4 shows the data used to generate the function used for the conflicts arising due to the laydown space of the current activity and work in place.

Table A-4: Laydown Space and Work in Place Conflict Effect Values.

Nominal Percent Conflict Levels (%)	Estimated Duration Increase (nominal time units)
0	0
25	0.025
50	0.05
75	0.075
100	0.1

The resulting regression takes the following form:

Regression Analysis: Percent Duration Increase versus PC, PC Sqrd

```
The regression equation is
Percent Duration Increase = 0.100 PC + 0.000000 PC Sqrd


Predictor         Coef     SE Coef  T  P
Noconstant
PC            0.100000    0.000000  *  *
PC Sqrd     0.00000000  0.00000000  *  *


S = 0


Analysis of Variance

Source          DF         SS         MS  F  P
Regression       2  0.0187500  0.0093750  *  *
Residual Error   3  0.0000000  0.0000000
Total            5  0.0187500
```

```
Source    DF     Seq SS
PC         1  0.0187500
PC Sqrd    1  0.0000000
```

A.5 Travel Path Space and Execution Space Conflict Congestion Function Component

Table A- shows the data used to generate the function used for the conflicts arising due to the travel path space of the current activity and the execution spaces of the other concurrent activities.

Table A-5: Travel Path Space and Execution Space Conflict Effect Values.

Nominal Percent Conflict Levels (%)	Estimated Duration Increase (nominal time units)
0	0
25	0.45
50	0.75
75	0.95
100	1

The resulting regression takes the following form:

Regression Analysis: Percent Duration Increase versus PC, PC Sqrd

```
The regression equation is
Percent Duration Increase = 2.03 PC - 1.03 PC Sqrd


Predictor       Coef  SE Coef       T      P
Noconstant
PC           2.03355  0.02102   96.72  0.000
PC Sqrd     -1.03226  0.02448  -42.16  0.000


S = 0.00695608


Analysis of Variance

Source          DF      SS      MS         F      P
Regression       2  2.6674  1.3337  27562.67  0.000
```

```
Residual Error   3  0.0001  0.0000
Total            5  2.6675


Source   DF  Seq SS
PC        1  2.5813
PC Sqrd   1  0.0860
```

A.6 Volume Calculation Sensitivity Demonstration

The volume calculation is based on a Monte Carlo method that estimates the volume of an object. This method is an approximation method that uses random sampling to estimate the volume. Because it employs random sampling, there is some degree of error in the calculation of the volume of the object. The congestion function developed in this thesis uses the volume calculations to determine the increase in duration of an activity during the scheduling phase of the methodology. Since the volume calculation is an estimate of the true volume of the object and thus will have error associated with it, the congestion function should not be extremely sensitive to the error of the calculation, i.e. the final duration increase should not be affected by the variations in the volume calculations.

A hypothetical problem with known activity space and space conflict volumes can be used to demonstrate the sensitivity of the congestion function to the error associated with the volume calculation. The hypothetical problem values are used to calculate the congestion function values for various levels of volume error. Each congestion function component is tested to determine how it is affected by the volume calculation error. In addition, the overall congestion function is calculated and used to determine the duration increase for a hypothetical activity.

Consider a hypothetical problem where two activity spaces are in conflict. Let the volume of one of the activities be A, the volume of the other activity be B, and the conflict volume be C. Let the values in Table A-6 represent the actual volume values of A, B, and C.

Table A-6: Hypothetical Problem Actual Volume Values.

A	B	C
20	15	10

Assume that A and B represent Work in Place and Execution Space, respectively, and C represents the conflict volume of those two spaces. The congestion function component for conflicts of this type is represented by Equation A.1.

$$PDI = 2.21(P4DC)^3 - 3.12(P4DC)^2 + 2.32(P4DC) \tag{A.1}$$

The result of inserting the volume values with no error into Equation A.1 is:

$PDI = 0.8148.$

To analyze the effect of calculation error on the congestion function, the error associated with the volume calculations is applied and the resulting congestion component values are computed. Assume that the error on the volume calculation is ±10% of the actual values presented in Table A-6. The high (+10%) and low (-10%) values of error are applied to each of the volume actual values. The resulting volume values were used to calculate the congestion function component value using Equation A.1. The calculated values are shown in Table A-7.

Table A-7: Congestion Values for Execution Space – Work in Place Conflicts.

		B		
		Low (-10%)	Actual	High (+10%)
C	Low (-10%)	0.8148	0.7462	0.6958
	Actual	0.9048	0.8148	0.7520
	High (+10%)	1.0145	0.8950	0.8148

The results shown in Table A-7 indicate the congestion function component for Execution Space conflicts with Work in Place can vary between 15% less and 25% more than the values obtained using the actual volumes of the objects.

Similar calculations were performed for the remaining four components in the congestion function using the congestion function equations given in Section 4.1.2. The results are given in Table A-8.

Table A-8: Congestion Values with Error for Each Conflict Type.

Conflict Type	No Error	B(High) C(Low)	B(Low) C(High)	B(High) C(Act.)	B(Low) C(Act.)	B(Act.) C(High)	B(Act.) C(Low)
Execution Space - Work In Place	0.8148	0.6958	1.0145	0.7520	0.9048	0.8950	0.7462
General Congestion	0.3822	0.4795	0.3403	0.4279	0.3448	0.3473	0.4330
Execution Space - Execution Space	0.6311	0.5400	0.7866	0.5829	0.7009	0.6932	0.5784
Laydown -Work in Place	1.0000	0.9000	1.1000	1.0000	1.0000	1.1000	0.9000
Travel Path Space - Execution Space	0.8956	0.8008	0.9702	0.8520	0.9385	0.9348	0.8472

Table A- shows the minimum and maximum deviations from the values using the actual volume values.

Table A-9: Congestion Values for Execution Space – Work in Place Conflicts.

Congestion Function Component / Conflict Type	Minimum Deviation (%)	Maximum Deviation (%)
Execution Space - Work In Place	-15	25
General Congestion	-11	25
Execution Space - Execution Space	-14	25
Laydown -Work in Place	-10	10
Travel Path Space - Execution Space	-11	8

The results presented in Table A-9 show that the individual components are affected by the error in the volume calculation. The Execution Space – Work in Place component is the most affected. The results also indicate that the congestion components that involve conflicts between Laydown Space and Work in Place, and Travel Path Space and Execution space are less affected than the other components in the congestion function.

Although the percentage deviation for the individual components appears to be large, these deviations do not have a dramatic effect on the overall congestion function. The overall congestion function value was calculated for each of the error levels and applied to a hypothetical activity with a nominal duration of 10 time units. The resulting durations ranged from 24 to 27 time units. A duration of 25 time units was found using the actual volume values given in Table A-6. Figure A-1 shows the effect of the volume calculation error on the congestion function through the resulting duration values.

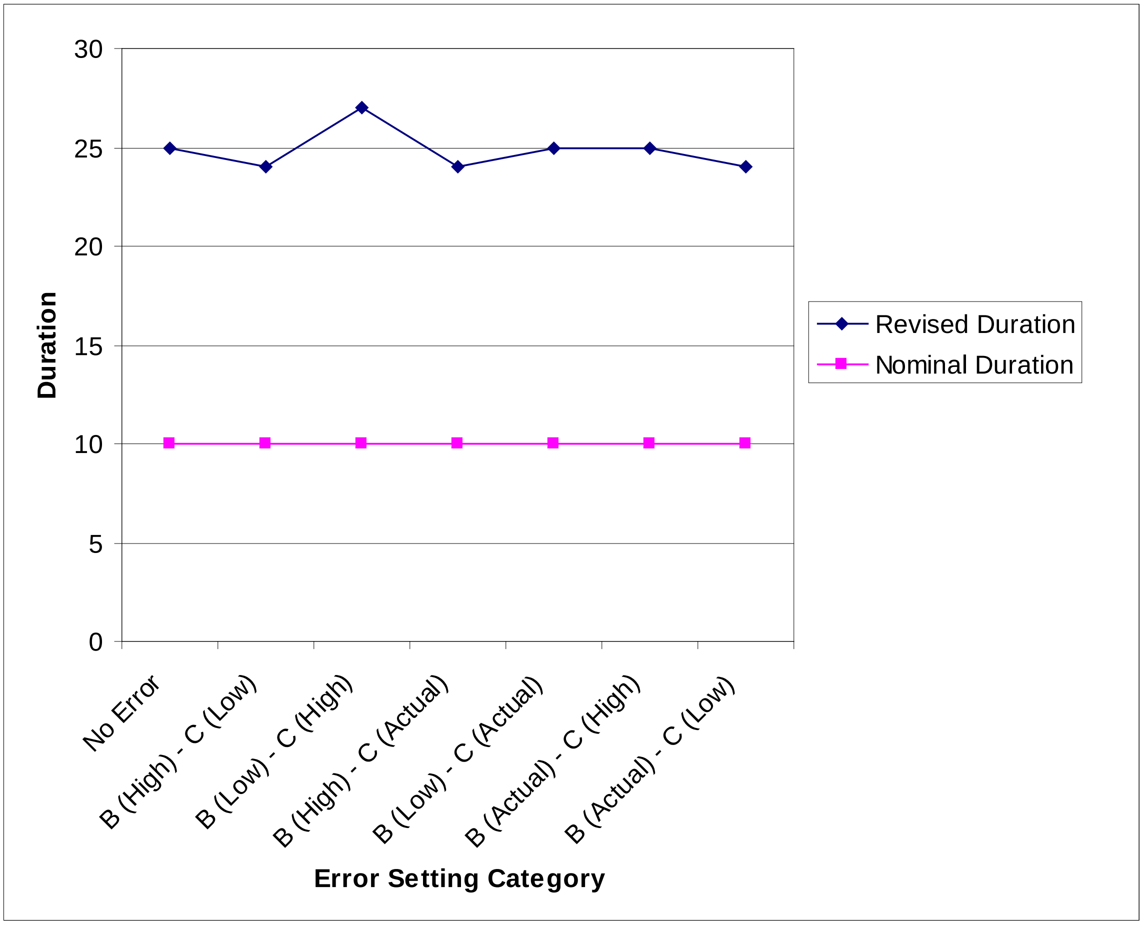


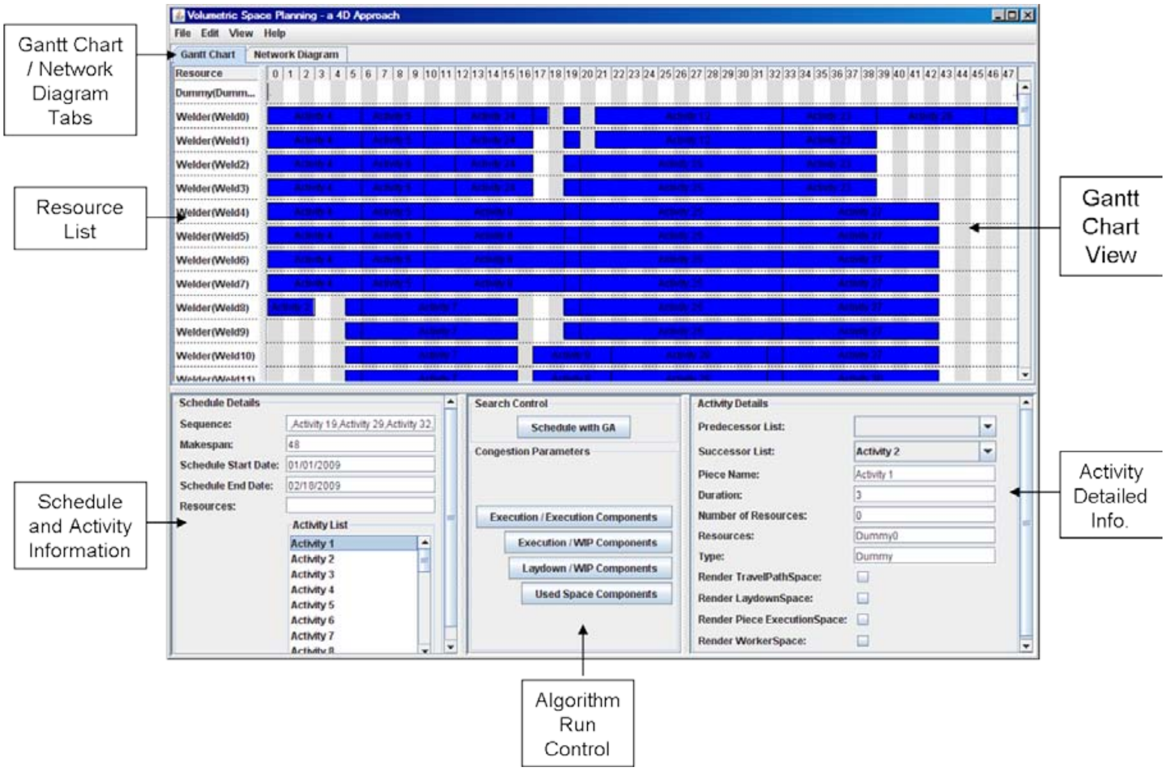
Figure A-1: Volume Calculation Error Effect on Activity Duration.

This demonstration shows that the error associated with the volume calculation does have an effect on the congestion function and the final duration increase amount. However, this demonstration shows the worst case result of a 10% error in volume calculation and it would be expected that this case would happen very rarely in practice.

Appendix B

Software Implementation

The software implementation of the developed methodology has two screens: the main interface window and the geometry window. The main window provides the user with the capability to load a problem, view the schedule and activity information, and run the algorithm. Figure B-1 shows an example of the main window.



Schedule Details

Sequence: Activity 19,Activity 29,Activity 32
Makespan: 48
Schedule Start Date: 01/01/2009
Schedule End Date: 02/18/2009
Resources:

Activity List

- Activity 1
- Activity 2
- Activity 3
- Activity 4
- Activity 5
- Activity 6
- Activity 7
- Activity 8

Search Control

Congestion Parameters

Activity Details

Predecessor List:
Successor List: Activity 2
Piece Name: Activity 1
Duration: 3
Number of Resources: 0
Resources: Dummy0
Type: Dummy
Render TravelPathSpace: ☐
Render LaydownSpace: ☐
Render Piece ExecutionSpace: ☐
Render WorkerSpace: ☐

Figure B-1: Software Main Window.

The Gantt chart view shows the specific resources and associated activities on a time line. The schedule and activity information are presented to the user on the bottom left pane of the main window. In this pane the user can see the list of activities that are under consideration and the results of applying the algorithm to a problem. If the user selects an activity from the list, the details of the activity are presented in the activity details pane on the right side of the window. The activity details pane gives detailed information about the activity including predecessors, successors, resource requirements and the type of activity. In addition, the user is also given the ability to view the spatial requirements of the activity by checking a check box.

The user can also view a diagram of the problem network by clicking on the network diagram tab at the top of the window. Figure B-2 shows a network diagram for an example problem.

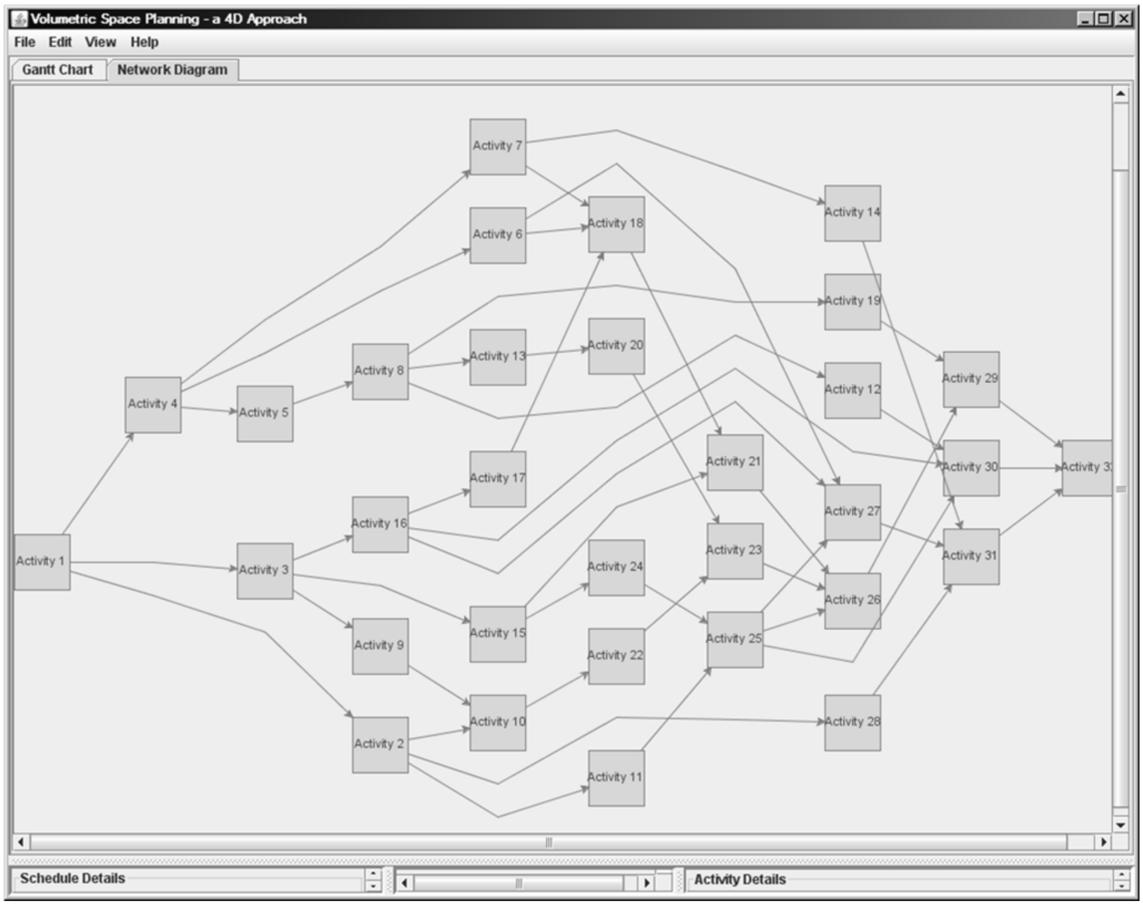
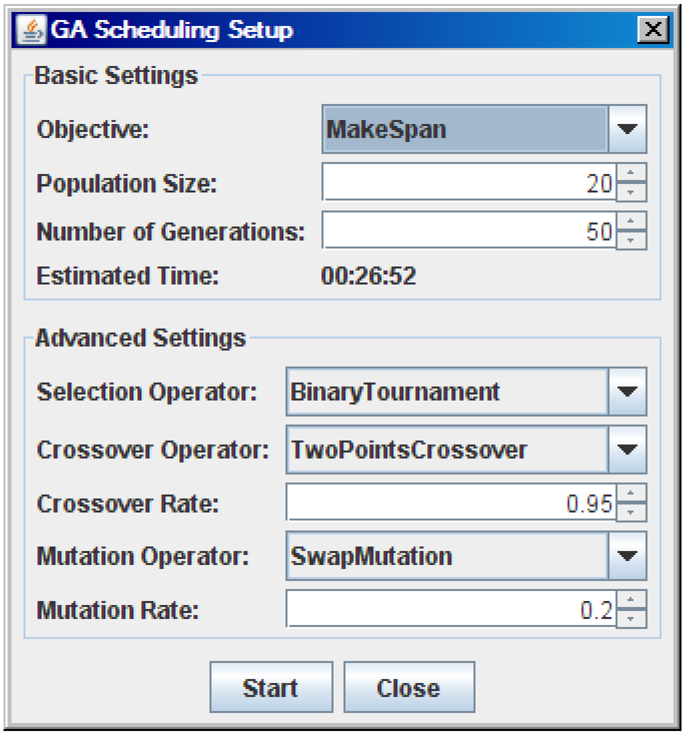


Figure B-2: Network Diagram.

Each box represents an activity and the arrows show the precedence relationships. The network diagram shows the precedence constraints of the activities and indicates the relative complexity of the problem under consideration.

The algorithm run control pane is used to initiate the scheduling algorithm. Figure B-3 shows the run control window that is presented to the user when the Schedule with GA button is pressed.



Objective: MakeSpan
Population Size: 20
Number of Generations: 50
Estimated Time: 00:26:52
Selection Operator: BinaryTournament
Crossover Operator: TwoPointsCrossover
Crossover Rate: 0.95
Mutation Operator: SwapMutation
Mutation Rate: 0.2

Figure B-3: Genetic Algorithm Run Control Dialog.

Each of the genetic algorithm parameters are presented to the user for modification.

The geometry window displays the geometrical information of the problem. Figure B-4 shows an example of the geometry window.

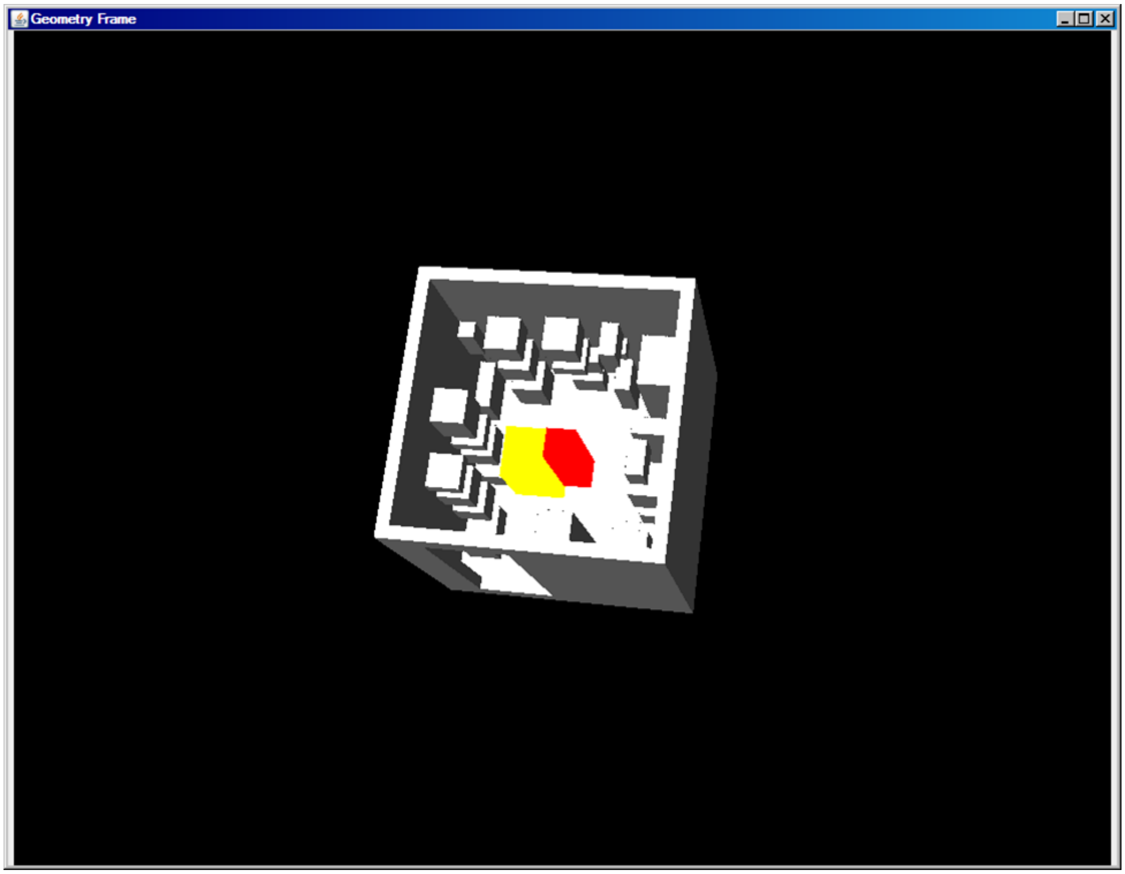


Figure B-4: Geometry Window.

The unit space and the geometric representations of the components are shown in the geometry window. In addition, if the user selects a space representation from the check boxes on the main page, those spaces will appear in the geometry window. The execution space for an activity is presented in Figure B-4 and represented as the colored boxes.

Appendix C

Baseline Results (PSLib)

The results of the baseline analysis for problems in the PSLib problem set are given in Table C-1. Time unit deviations are given in the first column followed by the number of occurrences of the deviation value for the 30 and 60 activity problem sets.

Table C-1: Deviations from Optimal without Space.

Deviation From Optimal	Number of Instances (30 Activity Set)	Number of Instances (60 Activity Set)
0	334	300
1	46	13
2	33	18
3	23	16
4	17	10
5	10	13
6	11	13
7	6	14
8	0	6
9	0	7
10	0	8
11	0	9
12	0	8
13	0	11
14	0	7
15	0	7
16	0	7
17	0	3
18	0	4
19	0	1
20	0	2
21	0	1
22	0	1
23	0	2

Appendix D

Sample PSLib Problem

D.1 Test RCPS Problem Definition

```
************************************************************************
file with basedata            : j30_26.bas
initial value random generator: 20374
************************************************************************
projects                      :  1
jobs (incl. supersource/sink ):  32
horizon                       :  164
RESOURCES
  - renewable                 :  4   R
  - nonrenewable              :  0   N
  - doubly constrained        :  0   D
************************************************************************
PROJECT INFORMATION:
pronr.  #jobs rel.date duedate tardcost  MPM-Time
    1     30      0       41       21       41
************************************************************************
PRECEDENCE RELATIONS:
jobnr.    #modes  #successors   successors
   1        1          3           2   3   4
   2        1          3          10  11  28
   3        1          3           9  15  16
   4        1          3           5   6   7
   5        1          1           8
   6        1          2          18  27
   7        1          2          14  18
   8        1          3          12  13  19
   9        1          1          10
  10        1          1          22
  11        1          1          25
  12        1          1          30
  13        1          1          20
  14        1          1          31
  15        1          2          21  24
  16        1          3          17  27  30
  17        1          1          18
  18        1          1          21
  19        1          1          29
```

```
  20         1      1      23
  21         1      1      26
  22         1      1      23
  23         1      1      26
  24         1      1      25
  25         1      3      26  27  30
  26         1      1      29
  27         1      1      31
  28         1      1      31
  29         1      1      32
  30         1      1      32
  31         1      1      32
  32         1      0
************************************************************************
REQUESTS/DURATIONS:
jobnr. mode duration  R 1  R 2  R 3  R 4
------------------------------------------------------------------------
  1      1      0      0    0    0    0
  2      1      2      1    2    4    0
  3      1      5      0    5    9   10
  4      1      6      8   10   10    0
  5      1      4      8    3    0    8
  6      1      2      3    1    8    5
  7      1      9      5    0    5   10
  8      1      9      4    0    0    3
  9      1      4      8    0    8   10
 10      1      8      0    0    0    5
 11      1      7      1    6    6    6
 12      1     10      2   10    3    8
 13      1      1     10    8    2    8
 14      1      1      1    3    6    1
 15      1      1      4    0    9    9
 16      1      5      0    9    6    9
 17      1      2      4    3    9   10
 18      1      5      0   10    9    0
 19      1      5      2    7    9    0
 20      1      9      0    5    4    1
 21      1     10      0    8    0    8
 22      1      1      9    0    0    6
 23      1      5      4    8    3    8
 24      1      4      5    7    3    5
 25      1     10      9    0    0    1
 26      1      5      1    8    0    7
 27      1      9      7    3    4    7
 28      1     10      7    0    0    3
```

```
 29      1         2         1    3    0   10
 30      1         10        4    0    3    7
 31      1         3         0    4    5    1
 32      1         0         0    0    0    0
********************************************************************************
RESOURCEAVAILABILITIES:
  R 1  R 2  R 3  R 4
   24   23   25   33
********************************************************************************
```

D.2 Test RCPS Problem Space Definitions

For the three test scenarios (10%, 25%, and 40% occupied), each of the components were defined by three parameters: Size x, Size y, and Size z, that represent the length of the component in the x, y, and z dimensions, respectively. Table D-1 shows the component dimensions for Scenario 1 (10% occupied space).

Table D-1: Test Scenario 1 Component Dimensions.

Index	Piece	Size x	Size y	Size z
1	Dummy			
2	Install Piece 1	1.26	1.89	5.04
3	Install Piece 2	0.63	0.63	0.63
4	Install Piece 3	0.63	1.26	2.52
5	Install Piece 4	0.63	1.26	1.26
6	Install Piece 5	1.26	0.63	1.26
7	Install Piece 6	0.63	0.63	1.26
8	Install Piece 7	1.26	1.26	0.63
9	Install Piece 8	1.26	1.26	1.26
10	Install Piece 9	1.26	0.63	5.04
11	Install Piece 10	1.26	1.26	1.89
12	Install Piece 11	1.26	1.26	1.26
13	Install Piece 12	1.89	1.26	1.26
14	Install Piece 13	0.63	1.89	1.89
15	Install Piece 14	1.26	1.26	1.26
16	Install Piece 15	0.63	0.63	1.89
17	Install Piece 16	1.26	1.26	1.26
18	Install Piece 17	1.26	1.26	0.63
19	Install Piece 18	1.26	1.26	1.89
20	Install Piece 19	1.26	1.26	1.89
21	Install Piece 20	1.26	1.26	1.89
22	Install Piece 21	0.63	1.26	1.26
23	Install Piece 22	1.26	0.63	1.89
24	Install Piece 23	1.26	1.26	1.89
25	Install Piece 24	1.26	0.63	1.26
26	Install Piece 25	1.26	1.89	1.26
27	Install Piece 26	1.26	1.26	1.26
28	Install Piece 27	1.26	1.26	1.26
29	Install Piece 28	1.26	1.26	5.04
30	Install Piece 29	0.63	0.63	3.78
31	Install Piece 30	1.26	1.26	0.63
32	Dummy			

The component dimensions for Scenario 2 (25% occupied space) are given in Table D-2.

Table D-2: Test Scenario 2 Component Dimensions.

Index	Piece	Size x	Size y	Size z
1	Dummy			
2	Install Piece 1	1.712	2.568	6.848
3	Install Piece 2	0.856	0.856	0.856
4	Install Piece 3	0.856	1.712	3.424
5	Install Piece 4	0.856	1.712	1.712
6	Install Piece 5	1.712	0.856	1.712
7	Install Piece 6	0.856	0.856	1.712
8	Install Piece 7	1.712	1.712	0.856
9	Install Piece 8	1.712	1.712	1.712
10	Install Piece 9	1.712	0.856	6.848
11	Install Piece 10	1.712	1.712	2.568
12	Install Piece 11	1.712	1.712	1.712
13	Install Piece 12	2.568	1.712	1.712
14	Install Piece 13	0.856	2.568	2.568
15	Install Piece 14	1.712	1.712	1.712
16	Install Piece 15	0.856	0.856	2.568
17	Install Piece 16	1.712	1.712	1.712
18	Install Piece 17	1.712	1.712	0.856
19	Install Piece 18	1.712	1.712	2.568
20	Install Piece 19	1.712	1.712	2.568
21	Install Piece 20	1.712	1.712	2.568
22	Install Piece 21	0.856	1.712	1.712
23	Install Piece 22	1.712	0.856	2.568
24	Install Piece 23	1.712	1.712	2.568
25	Install Piece 24	1.712	0.856	1.712
26	Install Piece 25	1.712	2.568	1.712
27	Install Piece 26	1.712	1.712	1.712
28	Install Piece 27	1.712	1.712	1.712
29	Install Piece 28	1.712	1.712	6.848
30	Install Piece 29	0.856	0.856	5.136
31	Install Piece 30	1.712	1.712	0.856
32	Dummy			

The component dimensions for Scenario 3 (40% occupied space) are given in Table D-3.

Table D-3: Test Scenario 3 Component Dimensions.

Index	Piece	Size x	Size y	Size z
1	Dummy			
2	Install Piece 1	2	3	8
3	Install Piece 2	1	1	1
4	Install Piece 3	1	2	4
5	Install Piece 4	1	2	2
6	Install Piece 5	2	1	2
7	Install Piece 6	1	1	2
8	Install Piece 7	2	2	1
9	Install Piece 8	2	2	2
10	Install Piece 9	2	1	8
11	Install Piece 10	2	2	3
12	Install Piece 11	2	2	2
13	Install Piece 12	3	2	2
14	Install Piece 13	1	3	3
15	Install Piece 14	2	2	2
16	Install Piece 15	1	1	3
17	Install Piece 16	2	2	2
18	Install Piece 17	2	2	1
19	Install Piece 18	2	2	3
20	Install Piece 19	2	2	3
21	Install Piece 20	2	2	3
22	Install Piece 21	1	2	2
23	Install Piece 22	2	1	3
24	Install Piece 23	2	2	3
25	Install Piece 24	2	1	2
26	Install Piece 25	2	3	2
27	Install Piece 26	2	2	2
28	Install Piece 27	2	2	2
29	Install Piece 28	2	2	8
30	Install Piece 29	1	1	6
31	Install Piece 30	2	2	1
32	Dummy			

D.3 Component Installation Locations

For each of the three scenarios, the installation locations were held constant and only the size of the components were modified to obtain the desired level of occupied space. Table D-4 provides the final installation locations for all of the components.

Table D-4: Component Installation Locations.

Index	Piece	Location X	Location Y	Location Z
1	Dummy			
2	Install Piece 1	3.5	3	-0.5
3	Install Piece 2	1.75	4	0.25
4	Install Piece 3	1.75	3.5	-2.5
5	Install Piece 4	1.75	3.5	2
6	Install Piece 5	0	4	-3.5
7	Install Piece 6	0.5	4	-1.25
8	Install Piece 7	0	3.5	0.5
9	Install Piece 8	0	3.5	2.25
10	Install Piece 9	3.5	0.5	-0.5
11	Install Piece 10	-2.25	3.5	-3
12	Install Piece 11	-2.25	3.5	-0.25
13	Install Piece 12	-3	0	-3.5
14	Install Piece 13	-4	3	-3
15	Install Piece 14	-2.25	3.5	2
16	Install Piece 15	-4	4	0.5
17	Install Piece 16	-3.5	0.25	-1.25
18	Install Piece 17	-3.5	0.25	0.5
19	Install Piece 18	-3.5	0.25	2.75
20	Install Piece 19	3.5	-1.25	-3
21	Install Piece 20	3.5	-1.25	0.25
22	Install Piece 21	3.5	-1.25	3.5
23	Install Piece 22	3.5	-4	-3
24	Install Piece 23	3.5	-3.5	0.25
25	Install Piece 24	3.5	-4	3
26	Install Piece 25	-3.5	-2.75	-3.5
27	Install Piece 26	-3.5	-2	-1.25
28	Install Piece 27	-3.5	-2	1
29	Install Piece 28	0.5	-3.5	-0.5
30	Install Piece 29	0	0	-1.5
31	Install Piece 30	-3.5	-2	2.75
32	Dummy			

The locations are relative to the test unit that is modeled as a 9’ x 9’ x 9’ room and is shown in Figure D-1 with the components in their final installation locations.

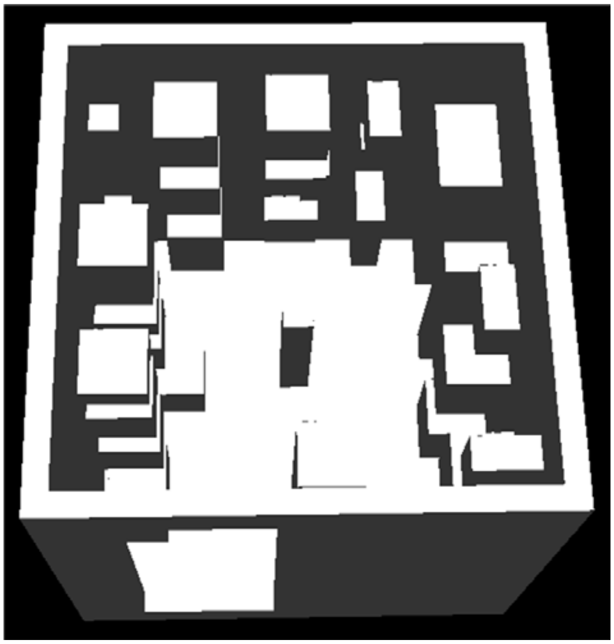


Figure D-1: Test Unit Space.

Appendix E

TWR Problem

Index	Activity Name	Piece Reference
1	Initialize	null
2	Install ALARM_MASTER	ALARM_MASTER.wrl
3	Install COMPUTER_PERHICALS_CABINET	COMPUTER_PERHICALS_CABINET.wrl
4	Install COMPUTER_SHELF_STBD	COMPUTER_SHELF_STBD.wrl
5	Install ELECTRICAL_PANEL_BOX	ELECTRICAL_PANEL_BOX.wrl
6	Install CHART_TABLE_SOLID	CHART_TABLE_SOLID.wrl
7	Install CHART_TABLE_FLASHING	CHART_TABLE_FLASHING.wrl
8	Install CHART_TABLE_FRONT_DETAILS	CHART_TABLE_FRONT_DETAILS.wrl
9	Install CHART_TABLE_FRONT_DETAILS_253	CHART_TABLE_FRONT_DETAILS_253.wrl
10	Install CONSOLE_SOLID	CONSOLE_SOLID.wrl
11	Install BLINKER_TELEGRAPH_KEY	BLINKER_TELEGRAPH_KEY.wrl
12	Install BOW_THRUST_CONTROLLER	BOW_THRUST_CONTROLLER.wrl
13	Install CHEMICAL_ATTACK_ALARM	CHEMICAL_ATTACK_ALARM.wrl
14	Install COLLISION_ALARM	COLLISION_ALARM.wrl
15	Install MONITOR_STAND	MONITOR_STAND.wrl
16	Install COMPUTER_MONITOR	COMPUTER_MONITOR.wrl
17	Install CONSOLE_CONTROLS	CONSOLE_CONTROLS.wrl
18	Install CPU_SHELF	CPU_SHELF.wrl
19	Install DEPTH_SOUNDER	DEPTH_SOUNDER.wrl
20	Install ENGINE_START_STOP_PORT	ENGINE_START_STOP_PORT.wrl
21	Install ENGINE_START_STOP_STBD	ENGINE_START_STOP_STBD.wrl
22	Install ENGINE_STOP_PORT_STBD	ENGINE_STOP_PORT_STBD.wrl
23	Install ENGINE_TACHMETER_PORT_STBD	ENGINE_TACHOMETER_PORT_STBD.wrl
24	Install FWD_FLOOD_LIGHT_ON_OFF	FWD_FLOOD_LIGHT_ON_OFF.wrl
25	Install GENERAL_ALARM	GENERAL_ALARM.wrl
26	Install GYRO_COMPASS	GYRO_COMPASS.wrl
27	Install HOUSE_TOP_SEARCH_LIGHT	HOUSE_TOP_SEARCH_LIGHT.wrl
28	Install MAGNETIC_COMPASS	MAGNETIC_COMPASS.wrl
29	Install MID_SHIP_FLOOD_LIGHT	MID_SHIP_FLOOD_LIGHT.wrl
30	InstallPORT_MAIN_ENGINE_CONTROL	PORT_MAIN_ENGINE_CONTROL.wrl

Index	Activity Name	Piece Reference
31	Install RADAR_DISPLAY_UNIT	RADAR_DISPLAY_UNIT.wrl
32	Install START_STOP_ETHER_PORT	START_STOP_ETHER_PORT.wrl
33	Install START_STOP_ETHER_STBD	START_STOP_ETHER_STBD.wrl
34	Install STBD_MAIN_ENGINE_CONTROL	STBD_MAIN_ENGINE_CONTROL.wrl
35	Install STERN_FLOODLIGHT_ON_OFF	STERN_FLOODLIGHT_ON_OFF.wrl
36	Install S_P_TELEPHONE	S_P_TELEPHONE.wrl
37	Install UNDERWAY_SPEED_LOG_CONSOLE	UNDERWAY_SPEED_LOG_CONSOLE.wrl
38	Install SHIPS_WHEEL	SHIPS_WHEEL.wrl
39	Install POWER_DISTRIBUTION_PANEL	POWER_DISTRIBUTION_PANEL.wrl
40	Install PILOT_HOUSE_CHART_TABLE_CH	PILOT_HOUSE_CHART_TABLE_CH.wrl
41	Install PILOT_HOUSE_CONSOLE_CHAIR	PILOT_HOUSE_CONSOLE_CHAIR.wrl
42	Terminate	null

Index	Nominal Duration	Type	Res 1 Req	Res 2 Req	Res 3 Req	Res 4 Req	Predecessors
1	0	Dummy	0	0	0	0	-1
2	8	ElectricalPanel	0	0	1	0	1
3	4	Foundation	1	0	0	0	1
4	4	Foundation	1	0	0	0	1
5	16	ElectricalPanel	0	0	2	0	1
6	24	Foundation	2	0	0	0	1
7	4	Pipe	0	1	0	0	6
8	12	Foundation	1	0	0	0	7
9	8	Foundation	1	0	0	0	7
10	24	Foundation	2	0	0	0	1
11	4	ElectricalPanel	0	0	1	0	10
12	8	ElectricalPanel	0	0	1	0	10
13	8	ElectricalPanel	0	0	1	0	10
14	4	ElectricalPanel	0	0	1	0	10
15	4	Foundation	1	0	0	0	10
16	2	Foundation	1	0	0	0	10|15
17	16	ElectricalPanel	0	0	2	0	10
18	4	Foundation	1	0	0	0	5
19	4	ElectricalPanel	0	0	1	0	10
20	4	ElectricalPanel	0	0	1	0	10
21	4	ElectricalPanel	0	0	1	0	10

Index	Nominal Duration	Type	Res 1 Req	Res 2 Req	Res 3 Req	Res 4 Req	Predecessors
22	4	ElectricalPanel	0	0	1	0	10
23	4	ElectricalPanel	0	0	1	0	10
24	4	ElectricalPanel	0	0	1	0	10
25	6	ElectricalPanel	0	0	2	0	10
26	10	Pipe	0	1	0	0	10
27	4	ElectricalPanel	0	0	1	0	10
28	8	Foundation	1	0	0	0	10
29	4	ElectricalPanel	0	0	1	0	10
30	12	ElectricalPanel	0	0	1	0	10
31	12	ElectricalPanel	0	0	2	0	10
32	4	ElectricalPanel	0	0	1	0	10
33	4	ElectricalPanel	0	0	1	0	10
34	16	ElectricalPanel	0	0	1	0	10
35	4	ElectricalPanel	0	0	1	0	10
36	4	ElectricalPanel	0	0	1	0	10
37	4	ElectricalPanel	0	0	1	0	10
38	8	Foundation	2	0	0	0	10|11|12|13|14|15|16|17|19|20|21|22|23|24|25|26|27|28|29|30|31|32|33|34|35|37
39	10	ElectricalPanel	0	0	2	0	38
40	1	Foundation	0	0	0	1	2|3|4|8|9|18|36|38
41	1	Foundation	0	0	0	1	2|3|4|8|9|18|36|38
42	0	Dummy	0	0	0	0	39|40|41

Index	Location x	Location y	Location z	Laydown x	Laydown y	Laydown z
1	0	0	0	0	0	0
2	-24.7702	24.3095	-6.6098	-33.0000	24.3095	-7.2500
3	-29.1875	23.2179	-7.5000	-32.7500	23.2179	-7.0000
4	-24.9095	23.5302	4.6539	-23.0000	14.3000	-8.7500
5	-27.0834	23.6000	-7.9375	-33.0000	23.6000	-6.7500
6	-34.7000	23.0955	-2.7500	-30.5000	23.0955	-6.5000

Index	Location x	Location y	Location z	Laydown x	Laydown y	Laydown z
7	-34.2500	24.0000	-1.0000	-31.0000	24.0000	-3.5000
8	-34.9500	23.6693	-2.7500	-30.2500	23.6693	-6.2500
9	-37.1980	25.1694	-0.8854	-33.5000	25.1694	-6.7500
10	-25.7000	22.6500	-0.9500	-33.5000	22.8500	-5.2500
11	-25.1100	25.2922	2.8021	-32.9846	22.0000	-7.1010
12	-25.6399	24.9176	-1.4688	-32.7778	21.4271	-6.6498
13	-23.8519	25.4484	-1.6993	-32.7655	21.4660	-6.6411
14	-24.5310	25.2468	-1.0326	-32.7367	21.5064	-6.6268
15	-24.0674	25.1618	-1.3851	-33.0000	22.8500	-7.0000
16	-24.3889	26.1864	-1.3208	-33.0000	22.0000	-7.2000
17	-24.6180	25.0461	0.2523	-32.1791	21.5383	-6.7120
18	-27.0834	25.1784	-7.5000	-32.7500	23.0000	-6.7500
19	-24.1189	25.7603	3.0000	-32.9366	21.7439	-6.7022
20	-25.4441	24.9539	0.7523	-32.8551	21.5680	-6.7222
21	-25.4441	24.9539	1.4583	-32.8551	21.5680	-6.7222
22	-25.9234	24.8117	1.4583	-32.6779	21.5872	-6.9506
23	-24.8706	25.1459	1.4550	-32.6714	21.4826	-6.8634
24	-23.9063	25.4106	0.7523	-33.0651	21.5864	-6.8290
25	-23.8519	25.4484	-1.0326	-32.8796	21.5504	-6.8660
26	-24.4038	25.5302	-0.0417	-32.9217	22.1263	-6.9004
27	-24.8650	25.1259	0.7523	-32.7240	21.5239	-6.9055
28	-24.0864	25.5041	1.5208	-32.4840	21.7933	-6.8967
29	-24.1460	25.3393	0.7523	-32.8930	21.5729	-6.8858
30	-25.6521	25.1017	1.2083	-32.6901	21.7257	-6.8354
31	-24.8622	25.5614	-2.8438	-32.8500	22.2500	-7.0000
32	-25.6838	24.9500	0.7523	-33.0496	21.6056	-6.8287
33	-25.6838	24.9500	2.1644	-32.8041	21.5519	-6.9093
34	-25.6521	25.1017	1.7083	-32.6367	21.7472	-6.9320
35	-24.3856	25.2683	0.7523	-32.6957	21.5295	-6.7666
36	-24.6214	25.2200	-1.7188	-32.7346	21.5451	-6.8933
37	-25.8128	25.1705	2.9479	-32.7288	21.8113	-6.8974
38	-26.5205	24.0718	0.0000	-33.2500	24.0718	-6.7500
39	-32.6875	23.9317	-7.8750	-33.0000	23.9317	-7.0000
40	-32.5000	23.7753	-2.5000	-32.5000	23.7753	-6.6500
41	-28.0000	23.9022	0.0000	-32.2500	23.9022	-7.0000
42	0	0	0	0	0	0

VITA

Daniel A. Finke

August 2004 to Present	The Pennsylvania State University, University Park, PA Doctor of Philosophy in Industrial Engineering
August 2000 to May 2002	The Pennsylvania State University, University Park, PA Masters of Science in Industrial Engineering and Operations Research
August 1995 to May 2000	New Mexico State University, Las Cruces, NM Bachelor of Science in Industrial Engineering

Publications

Finke, D.A., Ligetti, C.B., Traband, M.T., and A.G. Roy (2008). "*Activity-Based Spatial Scheduling*", Journal of Ship Production, **24**(1):12.

Burnett, G.A., Medeiros, D.J, Finke, D.A., and M.T. Traband 2008. Automating the development of shipyard manufacturing models. In *Proceedings of the 2008 Winter Simulation Conference*, ed. S.J. Mason, R.R. Hill, L. Monch, O. Rose, T. Jefferson, and J. W. Fowler, 1761-1767. Miami, FL: Institute of Electrical and Electronics Engineers Inc.

Finke, D.A., Medeiros, D.J., and Traband, M.T., "*Multiple Machine JIT Scheduling: A Tabu Search Approach*", International Journal of Production Research, **45**(21):4899, November 2007.

Finke, D.A., Ligetti, C.B., Traband, M.T., and A.G Roy (2007). "*Shipyard Space Allocation and Scheduling.*", Journal of Ship Production, **23**(4):197, November 2007.

Traband, M.T., Finke, D.A., Hadfield, J., and R. Santos (2004). "*Shipbuilding facility planning and design: A product-centric approach.*", Journal of Ship Production, **20**(4):240.

Finke, D.A, Medeiros, D.J., and M.T. Traband 2002. Shop scheduling using tabu search and simulation. In *Proceedings of the 2002 Winter Simulation Conference*, ed., E. Yucesan, C.-H. Chen, J.L. Snowdon, and J.M. Charnes, 1013-1017. San Diego, CA: Institute of Electrical and Electronics Engineers Inc.

Rosen, S.L., Geist, C.A, Finke, D.A., Nanda, J., and R.R. Barton 2001. Graphical methods for robust design of a semiconductor burn-in process. In *Proceedings of the 2001 Winter Simulation Conference*, ed., B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer, 1231-1237. Arlington, VA: Institute of Electrical and Electronics Engineers Inc.

Williams, D.L., Finke, D.A, Medeiros, D.J., and M.T. Traband 2001. Discrete simulation development for a proposed shipyard steel processing facility. In *Proceedings of the 2001 Winter Simulation Conference*, ed., B.A. Peters, J.S. Smith, D.J. Medeiros, and M.W. Rohrer, 882-887. Arlington, VA: Institute of Electrical and Electronics Engineers Inc.