Integrated Process Planning and Scheduling for a Complex Job Shop Using a Proxy Based Local Search

Andrew J. Henry

Dissertation submitted to the faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

In

Industrial and Systems Engineering

Kimberly P. Ellis, Chair

C. Patrick Koelling

Joel A. Nachlas

G. Don Taylor

October 1, 2015

Blacksburg, Virginia

Keywords: scheduling, process planning, local search, job shop

Integrated Process Planning and Scheduling for a Complex Job Shop Using a Proxy Based Local Search

Andrew J. Henry

ABSTRACT

Within manufacturing systems, process planning and scheduling are two interrelated problems that are often treated independently. Process planning involves deciding which operations are required to produce a finished product and which resources will perform each operation. Scheduling involves deciding the sequence that operations should be processed by each resource, where process planning decisions are known *a priori*. Integrating process planning and scheduling offers significant opportunities to reduce bottlenecks and improve plant performance, particularly for complex job shops.

This research is motivated by the coating and laminating (C&L) system of a film manufacturing facility, where more than 1,000 product types are regularly produced monthly. The C&L system can be described as a complex job shop with sequence dependent setups, operation re-entry, minimum and maximum wait time constraints, and a due date performance measure. In addition to the complex scheduling environment, products produced in the C&L system have multiple feasible process plans. The C&L system experiences significant issues with schedule generation and due date performance. Thus, an integrated process planning and scheduling approach is needed to address large scale industry problems.

In this research, a novel proxy measure based local search (PBLS) approach is proposed to address the integrated process planning and scheduling for a complex job shop. PBLS uses a proxy measure in conjunction with local search procedures to adjust process planning decisions with the goal of reducing total tardiness. A new dispatching heuristic, OU-MW, is developed to generate feasible schedules for complex job shop scheduling problems with maximum wait time constraints. A regression based proxy approach, PBLS-R, and a neural network based proxy approach, PBLS-NN, are investigated. In each case, descriptive statistics about the active process plan set are used as independent variables in the model. The resulting proxy measure is used to evaluate the effect of process planning local search moves on the objective function sum of total tardiness. Using the proxy measure to guide a local search reduces the number of times a detailed schedule is generated reducing overall runtime.

In summary, the proxy measure based local search approach involves the following stages:

- Generate a set of feasible schedules for a set of jobs in a complex job shop.
- Evaluate the parameters and results of the schedules to establish a proxy measure that will estimate the effect of process planning decisions on objective function performance.
- Apply local search methods to improve upon feasible schedules.

Both PBLS-R and PBLS-NN are integrated process planning and scheduling heuristics capable of addressing the challenges of the C&L problem. Both approaches show significant improvement in objective function performance when compared to local search guided by random walk. Finally, an optimal solution approach is applied to small data sets and the results are compared to those of PBLS-R and PBLS-NN. Although the proxy based local search approaches investigated do not guarantee optimality, they provide a significant improvement in computational time when compared to an optimal solution approach. The results suggest proxy based local search is an appealing approach for integrated process planning and scheduling in complex job shop environment where optimal solution approaches are not viable due to processing time.

Acknowledgements

First, I would like to thank my advisor, Kimberly Ellis, for her guidance, support, and patience. Throughout my educational journey she challenged me to grow and develop my thinking, while encouraging me to embrace a style that was uniquely mine. I would also like to thank each member of my committee. Patrick Koelling, Joel Nachlas, and G. Don Taylor. Their efforts helped improve the quality of this work. Additionally, each of them played a unique and important part in my professional development for which I am incredibly thankful.

This document represents the conclusion of a ten year educational tenure at Virginia Tech during this time I developed as a student, researcher, and a person. There are a plethora of educators, friends and family who I believe helped shape my thinking an ultimately this dissertation. To each of them, I want to say thank you. I owe a special thanks to Christian Wernz for his support and guidance early in my graduate studies, and Hannah Parks for all the help she provided over the years.

I want to thank the members of the Dover Lab for their camaraderie over the years, be it brainstorming, providing advice, or just talking. Specifically I would like to thank Steven Roesch, Charlie Crawford, and Bill Lucy.

The idea for this dissertation came from a summer spent at a Solutia manufacturing plant. This work would not have been possible without the support that I received from everyone at Solutia. Roger Bloeman, Frank Sharp, Kyle Wolfe, and others gave their time and ideas in support of this work. I am also grateful for the support received from the National Science Foundation (under grant IIP-0801919).

This journey may not have been possible, surely it wouldn’t have been as much fun, if it weren’t for my amazing group of friends. This document represents countless hours in Durham Hall but also numerous ISE happy hours, Hokie football games, and intermural sporting events. I owe a lot to the great group of people that were there with me over the years. Specifically, Lance Saunders, Cory Brozina, and Sudipto Aich were always there with advice when I needed it and a beer when I didn’t.

To the Sue and Al Volkmar: Thank you for everything. What started as conversations over coffee in the morning grew into an amazing friendship. You are really like a second family to me. I look forward to visiting in the future without the pressure to work.

To my family: I couldn’t ask for a more loving and supportive group. At every stage of my life you have been there for me. You have always believed in me and encouraged me. I wouldn’t be where I am today if it weren’t for you. You are a true source of inspiration. Thank you.

To my best friend, Julee Harlow: I wouldn’t have been able to complete this without your love and support. You believed in me and have been so supportive. You talked me though moments of panic and tolerated a lot of weekends spent working. I can’t wait for the next phase of our life together, one that doesn’t involve me spending most of my weekends working.

Table of Contents

List of Figures

List of Tables

1. Introduction

Improving manufacturing productivity through scheduling dates back the use of simple charting techniques for tracking progress and visualizing production plans (Binsse, 1887, Gantt, 1913) and has evolved to the complex metaheuristic approaches currently of interest (Pinedo, 2012). A number of researchers have successfully addressed scheduling problems in complex manufacturing environments (Candido *et al.*, 1998, Mason *et al.*, 2002, Ellis *et al.*, 2004).

A variety of flow-based classifications have been developed to aid in the discussion of production scheduling problems. Among these, a job shop scheduling problem is one in which n jobs must visit a subset of m machines, and the order in which a given job i visits the machines is job dependent. The problem is to generate a sequence of operations to be processed on each machine, taking into account job precedence constraints. The job shop scheduling problem is known to be one of the most difficult combinatorial optimization problems belonging to the class NP-complete (Lenstra and Rinnooy Kan, 1979).

While scheduling research focuses on sequencing operations on a set of machines, process planning involves evaluating the operations required to complete a job and selecting the machines on which a given operation is to be processed. Jobs may have multiple process plans, and the selection of a process plan from among a number of alternatives is often done with the objective of minimizing cost or time to manufacture a single job. Process plans generally serve as static input, in the form of a specified route for each job, to the scheduling problem. Process planning decisions that are made independent of shop floor or detailed scheduling information can lead to an imbalance in machine usage, unnecessary bottlenecks, and ultimately suboptimal plant performance. Likewise, scheduling without exploring process flexibility imposes unnecessary limitations on the scheduling system. Despite the drawbacks of disjoint decision making, many companies perform process planning and scheduling in separate departments. This disjoint treatment of process planning and scheduling also exists in academic work, as Uzsoy *et al.* (1994) notes in their review of production planning and scheduling models for semi-conductor manufacturing: "A salient point emerging from this review has been that problem areas have been compartmentalized, resulting in interrelated problems being considered in isolation. The relationship between production planning and shop-floor control has been almost completely ignored." The integration process planning and scheduling decisions offers the opportunity to leverage process flexibility and efficient scheduling practices to improve overall plant performance.

Although there is a plethora of research on scheduling, many questions in the field remain open, particularly when considering industry representative environments with a variety of complicating constraints and a large number of tasks to be scheduled. As a motivating example, the manufacturing of precision coating and window films is considered. This manufacturing system, in which quality is measured by the sum of total tardiness, includes a number of complicating scheduling constraints within a job shop environment, such as sequence dependent setups, minimum and maximum wait times, operation re-entry, as well as multiple process plans are considered. No research was found that investigates these characteristics in combination. In this research, a heuristic method for process planning and scheduling for a complex job shop environment is developed to contribute to the field of production scheduling in job shop environments.

The remaining sections of this dissertation are organized as follows. Section 2 presents a detailed description of the precision coating and window films manufacturing problem that motivates this research. Section 3 reviews the relevant literature for job shop scheduling and integrated process planning and scheduling. A mathematical model for both a single route scheduling problem and the integrated process planning and scheduling problem are provided in Section 4. The proposed proxy based local search (PBLS) approach is described in Section 5. Case study results are presented in Section 6, and conclusions and areas for future research are provided in Section 7.

2. Problem Description

The manufacturing of precision coating and window films is a complex, multi-level process. A film production facility produces finished product films by processing bulk film rolls through a variety of different machines. The coating and laminating (C&L) process is an integral part of a precision coating and window film manufacturing facility. The facility that motivated this research produces approximately 2,000 different film products across 9 machines. The C&L process generally incorporates of two main classes of machines: coaters and laminators. Laminating machines join two bulk film rolls to produce a single bulk laminated film roll. Coating machines take a bulk film roll and apply a chemical coating through either a bath or a yolk process. Coating machines may apply anti-static, scratch resistant, ultraviolet reflective, or adhesive coatings. Finished film products may visit multiple coating or laminating machines during the C&L process. This dissertation focuses on process planning and production scheduling for the C&L process, which has the characteristics of a job shop environment.

2.1 Job Shop Environment

A job shop scheduling problem is one in which n jobs must visit a subset of m machines. The order in which a given job i visits the machines is job dependent. Given demand for a set of jobs, the problem is to sequence the operations of these jobs on machines, taking into account operation precedence constraints. The objective of the classic job shop scheduling problem is to minimize makespan. The job shop scheduling problem is known to be one of the most difficult combinatorial optimization problems belonging to the class NP-complete (Lenstra and Rinnooy Kan, 1979).

A finished film product may require multiple laminating or coating steps. The sequence of the laminating and coating steps is based on product and manufacturing specifications that are considered fixed. Demand is known *a priori* along with the processing steps required to produce a given product. The routing of each finished product type differs based on the specified process plan. Given this, film production can be viewed as a job shop manufacturing environment because the steps required and order in which they are performed differ depending on final product specifications.

2.2 Re-entry

A finished product film may receive multiple coatings or multiple lamination layers and may visit the same machine multiple times. Re-entry, which occurs when a job visits the same workstation more than once during processing, can create scheduling complications. Thus, re-entry is important to consider when developing a machine schedule.

2.3 Due Date Performance

Much of the production in C&L is generated to fulfill specific customer orders. In a make-to-order environment such as this, customer satisfaction is directly influenced by the timeliness of an order. While many scheduling approaches seek to minimize overall makespan, the quality of a C&L production schedule is evaluated by due date performance, such as total tardiness. In this context, tardiness is the difference between the date a customer is promised an order and the date in which the order is actually made available. Scheduling even a single machine with the goal of minimizing weighted total tardiness has been shown to be NP-Hard (Lenstra *et al.*, 1977, Du and Leung, 1990).

2.4 Sequence Dependent Setups

A machine setup or changeover where some offline operation must be performed between successive operations, results in downtime. With sequence dependent setups, the changeover to be performed and the duration of the changeover depends not only on the next operation to be run on the machine but also the operation that was most recently processed on the machine. Sequence dependent setups are important to consider when generating quality schedules. Sequence dependent setups are prevalent in real world applications and, Panwalkar *et al.* (1973) reports that 70% of industrial schedulers face sequence dependent setups.

A finished product film can be described by the component films, the chemical coatings applied, and the film width. Laminating and coating machines process film of varying width and apply a variety of chemical coatings. A single laminating machine may process film of different widths, however, the machine must be setup for each particular film width. Consequently, a time-consuming changeover is required when changing the width of film that is processed on a laminating machine. Width changeovers on laminating machines are often on the order of two and a half hours. Similarly, coating machines apply different types coatings but an extensive setup maybe required when changing from one chemical coating to another. Chemical changeovers on the coating machines involve extensive cleaning, requiring as many as eight hours. Two finished film products may have the same width but different chemical coatings. Two jobs may be assigned to the same setup group on a laminator (minimum to no setup required). If these two jobs were performed in succession on a coating machine, then a long and costly setup would be required. Sequence dependent setups complicate the NP-Hard job shop scheduling problem. Efficient scheduling of C&L accounts for the sequence in which operations run on machines. Originally, simple dispatching rules were used to generate solutions for job shop scheduling problems with sequence dependent setups (Wilbrecht and Prescott, 1969). Recently, more specialized approaches for scheduling with sequence dependent setups have been considered (Allahverdi *et al.*, 2008), but sequence dependent setups in a job shop environment continue to present challenges.

2.5 Minimum and Maximum Wait Times

Several steps in the film manufacturing process require curing time after completion, where a film roll must wait a minimum amount of time before it can be processed on its next machine. This *minimum wait time* is specified as part of the process plan for a job and must be included in a feasible schedule. Similarly, there are several steps in the production process in which the successive operations must be performed within a set time window of completing the previous operation. These *maximum wait times* are required to ensure the quality of the finished film product. For example, some chemical coatings begin to deteriorate after application. A second sealing coat must be applied to stop deterioration. If the sealing coat is not applied within a specified amount of time, the film must be scrapped for quality reasons. A feasible production schedule must incorporate these minimum and maximum wait time constraints. Minimum wait time can be viewed as additional processing time required on a dummy machine (with infinite capacity) and therefore does not create a significant modeling challenge. However, maximum wait time, an important scheduling constraint in the C&L scheduling problem, is not widely discussed in job shop scheduling literature. Oddi *et al.* (2011) provide the only treatment of the subject found in the literature. They present a constraint programing based approach for scheduling with minimum and maximum wait times for a job shop with sequence dependent setups.

2.6 Multiple Processing Plans

A processing plan specifies the manufacturing steps required to produce a finished product to meet required customer requested specifications. Machines may have multiple functions, and facilities often have multiple machines of the same type. As a result, there may be multiple potential processing plans for the same finished product.

For example, the motivating C&L department consists of eight machines including two laminators and six coating/finishing machines. In addition, the capabilities of these machines differ slightly. A production version specifies the operations that need to be performed to complete production of a given film. Given that newer machines may have additional functionality, the number of operations in a production version may differ for the same finished product. Thus, finished product may have multiple production versions. Considering multiple machines in C&L have the same production capabilities, each operation in a production version may have multiple machines on which it can be processed.

Figure 2.1 illustrates the relationship between production versions and multiple machines for product WF-1. As shown, there are two production versions for product WF-A.

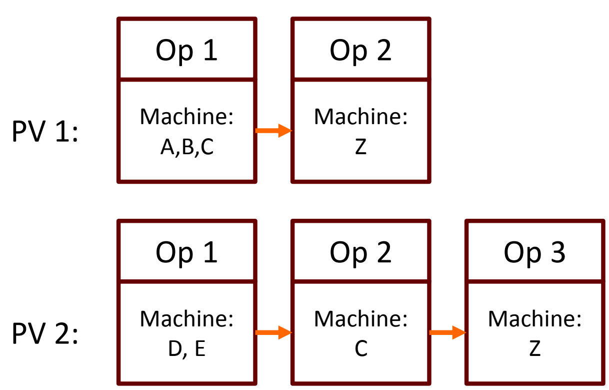


Figure 2.1 Process Plan Flexibility for WF-A

Production version 1 requires 2 operations, where operation 1 can be processed on either machine A, B or C, and operation 2 is processed on machine Z. Thus, production version 1 results in three possible process plans. Production version 2 requires three operations to complete production. Operation 1 has two possible machines on which it can be performed. One of the five possible process plans for product WF-A must be selected for the product WF-A to be scheduled. The selection of a process plan for a given product can have significant influence on the quality of the schedule generated. Appropriate attention must be paid to process plan selection as well as detailed production scheduling. The problem of selecting a production plan as well as generating a detailed schedule for production is known as integrated production planning and scheduling (Li *et al.*, 2010) and is the focus of this research.

3. Literature Survey

In this literature survey, the topics of process planning, sequencing and scheduling in a job shop environment are covered along with relevant work in integrated process planning and scheduling. The notation of Graham *et al.* (1979) is used when discussing different classes of scheduling problems. For example, $Jm||C_{max}$ describes the classic job shop scheduling problem with m machines with the objective of minimizing maximum completion time or makespan.

The classical job shop problem involves the sequencing of $|N|$ jobs to be processed on a subset of $|M|$ machines. Each job, i, consists of set of operations $\{O_i\}$ which must be completed in succession (precedence constrained). In the classic job shop problem, each operation p of job i is processed on a specific machine m for a predetermined processing time $\rho_{i,p}$ which cannot be interrupted (non-preemption). Each machine m can process at most one operation at a given time (capacity constraint). Each job has its own set of operations and flow through the machines. The objective of the classic job shop problem is to minimize makespan or the maximum completion time $Jm||C_{max}$. Several review papers provide an overview and background of the classic job shop scheduling problem. (Vaessens *et al.*, 1996, Jain and Meeran, 1999, Blazewicz *et al.*, 2003).

When considering the special case where each job i must visit each machine m exactly one time there is an upper bound of $(|N|!)^{|M|}$ possible solutions where $|N|!$ represents the maximum number of ways $|N|$ operations can be sequenced on a given machine. The job shop scheduling problem is of type NP-Complete. As a result, approaches to solve the job shop scheduling problem to optimality are limited to relatively small problems such as 20 jobs across 10 machines, or in the case of sequence dependent setups 20 jobs of 5 unique types across 10 machines (Artigues *et al.*, 2004).

The literature review is structured based on characteristics of the performance films manufacturing problem as follows:

- Job shop scheduling problem with tardiness
- Job shop scheduling problem with sequence dependent setups, and

- Integrated production planning and scheduling (IPPS) for job shop environments.

3.1 Job Shop Scheduling with Tardiness

With a rise in globalization and the potential for customers to source a product from not only the least expensive but most reliable and timely supplier, companies are increasingly concerned about their level of customer service, reducing lead times, and ensuring on-time delivery (Ovacik and Uzsoy, 1996). A natural extension of the classical job shop problem is to consider an objective function based on job due-date or customer requested date. In such problems, the classic C_{max} objective function is replaced with an objective function dependent not only on job completion time C_i but also job due date d_i. A common due-date based objective function is the sum of weighted tardiness $\sum w_i t_i$, where tardiness (t_i) is defined as the positive difference between job completion time and due date, such that $t_i = \max\{C_i - d_i, 0\}$ and w_i is a job dependent weighting factor. The problem of sequencing jobs on a single workstation with the objective of minimizing the sum of weighted tardiness $(1||\sum w_i t_i)$ is known to be NP-Hard (Lenstra *et al.*, 1977). The job shop scheduling problem $Jm||\sum w_i t_i$ is strongly NP-Hard as it has the single machine problem as a special case, where $m = 1$.

3.1.1 Exact Solution Approaches with Tardiness

Although the classic job shop problem is known to be NP-Complete, exact solution approaches have been developed that guarantee optimal solutions for the problem. Jain and Meeran (1999) provide a review of the classic job shop problem including work done on exact solution approaches. The review by Blazewicz *et al.* (2003) also summarizes the classic job shop problem and exact solution approaches. Exact solution approaches for job shop scheduling often utilize branch and bounding techniques, however, the approaches are generally not for large-scale industry problems.

When minimizing sum of weighted tardiness, several branch and bound techniques have been applied to the classic job shop scheduling problem. Balas (1969) develops a branch and bound approach to the $Jm||\sum w_i t_i$ problem. In his approach each node represents a completed schedule and branches are created by swapping adjacent operations on the critical path. This swapping concept is extended by Brucker *et al.* (1994) where sets of operations sequentially processed on a given machine are defined. The head and tail operations are swapped across sets. Other branching techniques including arch insertion, operation insertion, and active schedule generation have been explored. Singer and Pinedo (1998) provide a comparison of 8 different branch and bound techniques. They find an arc insertion approach, in which the order in which machines are explored is guided by a shifting bottleneck heuristic for total weighted tardiness (Pinedo and Singer, 1999) out performs all other approaches, for larger problems (10 jobs, 10 machines) in terms of computational time required when considering total weighted tardiness.

3.1.2 Shifting Bottleneck with Tardiness

The sifting bottleneck (SB) heuristic first introduced by Adams *et al.* (1988) focuses on generating a good solution for the classic job shop scheduling problem $Jm||C_{max}$. This heuristic decomposes the overall scheduling problem into multiple single machine scheduling problems $(1|r_i, d_i|L_{max})$. Precedence constraints from the original problem are handled by operation release times (r_i) in the sub-problems and an operation is assigned a due-date $\left(d_{i,p}\right)$ in the sub-problem based on the longest path between operation p of job i and the sentinel dummy operation. The SB heuristic optimizes the single-machine scheduling sub-problem on all un-scheduled machines while fixing the schedule for the machine identified as most critical or as the bottleneck. The heuristic repeats until all machine schedules have been fixed. An optional re-optimization step can be applied at the end of each iteration as proposed by Ovacik and Uzsoy (1992) to improve objective function performance at the expense of extra processing time. The SB heuristic includes the following six steps:

- Step 1: Let M be the set of all machines. Let M_o represent the set of machines that has been scheduled. Initially, $M_o = \emptyset$.
- Step 2: Solve a single machine scheduling problem for each machine $m \in M \backslash M_o$. This step is referred to as the Sub-problem Solution Procedure (SSP).
- Step 3: Identify the bottleneck or most critical machine $m^* \in M \backslash M_o$.
- Step 4: Sequence machine m^* based on the results from the SSP in Step 2. Update set $M_o = M_o \cup m^*$
- Step 5: Optional re-optimization of the schedule for each machine $m \in M_o$.
- Step 6: If $|M_o| < |M|$ go to Step 2, else stop.

The classical job shop scheduling problem focuses on minimizing makespan or the overall time to complete production on of all jobs. Minimizing maximum lateness in the single machine sub-problems (Step 2) works well to minimize makespan in the overall problem. Therefore, the original SB heuristic applies the algorithm of Carlier (1982) to solve the NP-Hard $1|r_i, d_r|L_{max}$ problem to optimality (Lenstra *et al.*, 1977). The relationship between the L_{max} objective function in the sub-problem and the C_{max} objective of the overall problem makes the objective function value of each sub-problem an ideal candidate for determining the critical or bottleneck machine (Step 3).

When the objective function of the job shop scheduling problem is based on due-date auch as, $Jm||\sum w_i T_i$, an alternative sub-problem procedures and machine criticality measures may outperform those used to solve the $Jm||C_{max}$ problem. Holtsclaw and Uzsoy (1996) perform a review of machine criticality measures (MCM, Step 3) and sub-problem solution procedures (SSP, Step 2) for solving job shop scheduling problems with due-date related objectives. For many L_{max} problems, they report that using an earliest due-date dispatching method for solving the single machine sub-problems (Step 2) out performed more sophisticated methods. However, when the objective function of the overall scheduling problem is to minimize the sum of weighted tardiness $Jm||\sum w_i t_i$ the selection of a SSP (Step 2) and the identification of the most critical or bottleneck machine (Step 3) become less straight forward. In such cases, it is unclear when sequencing a single machine how a single sequencing decision will influence the tardiness of the overall job.

Vepsalainen and Morton (1987) investigate the application of various dispatching rules to the $Jm|r_i|\sum w_i t_i$ scheduling problem and found that the Apparent Tardiness Cost (ATC) dispatching rule had the best performance. The ATC method calculates an importance factor $I_{i,p,m}(t)$ for each operation p of job i that is processed on machine m and dispatches jobs based on their importance factor, where t is the current machine time. $I_{i,p,m}(t)$ is calculated as

$$I_{i,p,m}(t) = \frac{w_i}{\rho_{i,p}} \exp\left(-\frac{\left(d_{i,p} - \rho_{i,p} - t\right)^+}{k\bar{\rho}}\right). \tag{1}$$

In Equation (1), w_i is a weighting factor for each job i, k is a look ahead parameter, $\bar{p}$ is the average processing time of operations that have yet to be sequenced on machine m, and $d_{i,p}$ is a local due date for operation p of job i. The ATC heuristic involves the following steps:

Step 1: Let Θ_m be the set of all operations to be processed on machine m. Let Θ_m^o be the set of operations already sequenced on machine m. Let t be the current ready time of machine m. Initially, $t = 0$ and $\Theta_m^o = \emptyset$.

Step 2: Calculate $I_{i,p,m}(t)$ for each operation $(i,p) \in \Theta_m \backslash \Theta_m^o$. Let (i^*, p^*) be the operation with the highest importance I.

Step 3: Sequence operation p^* of job i^* next on machine m. With start time equal to $\min\left(r_{i^*,p^*}, t\right)$.

Step 4: Update $t = \min\left(r_{i^*,p^*}, t\right) + p_{i^*,p^*}$, and $O_m^o = O_m^o \cup (i^*, p^*)$

Step 5: If $|\Theta_m^o| = |\Theta_m|$ terminate. Else, return to step 2.

Note that the value of parameter k balances the ATC between a Weighted Shortest Processing Time (WSPT) rule and a Least Work Remaining (LWR) rule. Pinedo and Singer (1999) apply the SB heuristic approach to solve a $Jm|r_i|\sum w_i t_i$ problem where a modified ATC dispatching rule is used as the sub-problem solution procedure (SSP). The authors select the sub-problem with the worst objective function as the most critical machine (Step 3).

Pinedo and Chao (1999) propose an alternate method for selecting the bottleneck machine when minimizing weighted tardiness is the objective. Their method is based on a machine criticality measure Z_m where,

$$Z_m = \sum_{i=1}^{|N|} w_i (C_i'' - C_i') \exp\left(-\frac{(d_i - C_i'')^+}{K}\right). \tag{2}$$

In Equation (2), C_i'' represents the completion time of job i from the current iteration, C_i' represents the completion time from the previous iteration, and K serves as a scaling parameter. When applied in Step 3 of the SB heuristic, the machine with the largest Z_m score is considered the most critical.

3.1.3 Neighborhood Search with Tardiness

Neighborhood search (NS) techniques have been successfully applied to the classic job shop scheduling problem. NS methods evaluate solutions in a neighborhood $N(x)$ around the current solution x. Where, x represents a complete and feasible schedule to the overall problem. Defining a neighborhood from which feasible solutions can be generated quickly is key to the efficient application of a NS technique. For job shop scheduling problems, the most common neighborhood definition is referred to as N_1 (Van Laarhoven *et al.*, 1992). The N_1 neighborhood around x consists of all solutions that can be generated by swapping the position of two adjacent jobs on a given machine in the original permutation x. In the absence of re-entrant flows, where a job visits any given machine only one, the N_1 neighborhood consists of feasible permutations. Additionally, solutions in the N_1 neighborhood are easy to generate.

Several neighborhood search approaches tailored for due date based objective fuctions have been developed. He *et al.* (1996) propose a simulated annealing based approach for minimizing total tardiness in a job shop. In contrast to the typical pairwise swap N_1 neighborhood scheme, He *et al.* (1996) employ a more complicated operation shifting method. The neighborhood structure used leads to a computationally expensive but effective application. Valls *et al.* (1998) develop a tabu search based heuristic for scheduling in an actual job shop environment. The problem that they consider involves release times, due dates, as well as single and parallel machine stations. They treat due dates and machine capacity as a non-binding constraints. The search picks the next solution probabilistically based on the degree in which the solution violates the due date and capacity constraints. González *et al.* (2010) provides a tabu based approach for job shops with setup times that demonstrates good performance on the popular datasets of Ovacik and Uzsoy (1994).

3.1.3.1 Tabu Search

Tabu Search (TS) algorithm is a neighborhood search algorithm that stores a short-term memory or tabu list T of solutions recently visited. The solution x' with the best objective function value in the neighborhood $N(x)$ around the current solution x is selected such that x' is not on the tabu list T. The tabu list helps avoid the search algorithm from cycling. A more complex version of a TS can involve a long term memory list to further aid in diversifying the search path. Tabu search has been applied to the classic job shop scheduling problem (Dell'Amico and Trubian, 1993, Barnes and Chambers, 1995). TS is considered one of the best methods for generating good solutions to the classic job shop problem (Watson *et al.*, 2003).

3.1.3.2 Simulated Annealing

Simulated annealing (SA) is a stochastic neighborhood search heuristic based on the physical property of annealing. With SA, a feasible solution x' is identified in the neighborhood $N(x)$

around the current solution x. The feasible solution x' is accepted as the new current solution with a probability of 1 if the objective function $f(x')$ improves over objective function $f(x)$ (such that $f(x') < f(x)$ for a minimization problem). If $f(x') > f(x)$ then x' is accepted as the current solution with the probability $e^{-\Delta/t}$ where Δ is the difference between the objective function value of x' and x and t is a temperature parameter that influences the probability of accepting non-improving solutions. The non-zero probability of accepting a non-improving solution helps the SA algorithm from getting stuck in local optimum. Kolonko (1999) provides an SA approach for the classic job shop scheduling problem and shows that an SA approach does not necessarily converge when applied to job shop scheduling problems. To overcome this issue the author applies a multiple start approach in which multiple SA heuristics are used in a genetic algorithm framework. The results reported are competitive with other methods for classic job shop scheduling heuristics and could be improved though more advanced methods for modifying the key parameter t.

3.1.4 Genetic Algorithms with Tardiness

Genetic algorithms (GAs) have been used to find solutions to a variety of complex optimization problems. The principals of GAs presented by Holland (1975) mimic natural evolutionary behavior. The basic approach of a GA is first to create an initial population or set of initial solutions. Each initial solution receives a fitness score based on the desired objective function. The next generation of solutions is populated through three main reproduction methods. First, the best solution from a current population is copied over to the next generation to insure the best solution is not lost. The second reproduction method is asexual and involves mutating a current solution to develop a new feasible solution. The third method for populating the next generation of solutions in a GA is through crossover in which two “parent” solutions are mixed to create a feasible “offspring” solution. This reproduction process is repeated until some termination criteria is reached. Encoding schemes, crossover, and mutation operators should be defined for the specific GA application. When the specific problem structure is considered for the crossover and mutation schemes, more efficient GA evolution is possible. Vázquez and Whitley (2000) present a review of research applying genetic algorithms to job shop scheduling. Additionally, Hart *et al.* (2005) provide readers with a review of using GAs for solving various classes of scheduling problems.

3.1.4.1 *Encoding Schemes for Job Shop Scheduling*

A key element of applying a genetic algorithm to solve any problem is encoding, which supports the development of new feasible generations. The original approach presented by Holland (1975) utilizes a binary string method to encode solutions into chromosomes. When developing an encoding and decoding scheme, the feasibility, legality, and mapping characteristics are considered. In a feasible coding scheme, chromosomes can be decoded into feasible solutions for the problem. Legality of a coding scheme refers to the potential of decoding a chromosome to represent a solution to the overall problem. Meanwhile, mapping refers to the uniqueness of a chromosome as it is mapped to a solution to the scheduling problem (Cheng *et al.*, 1996). A number of encoding schemes exist for the job shop scheduling problem (Cheng *et al.*, 1996).

3.1.4.2 *Job Shop Scheduling with GAs*

Genetic algorithms are applied to traditional job shop scheduling problems with various objective functions, for example $Jm||\frac{\sum C_i}{n} + C_{max}$ and $Jm||\sum e_i + t_i^2$ (Hax and Meal, 1975, Falkenauer and Bouffouix, 1991, Candido *et al.*, 1998, Manikas and Chang, 2009). GAs have been used to address several complicating constraints present in the C&L scheduling problem. For example, Wang and Wang (1997) provide a single machine hybrid scheduling method for problems with sequence dependent setups with the objective of minimizing earliness and tardiness with a penalty for setups. New methods for encoding, mutation, and crossover continue to emerge. The goal is to arrive at a better solution faster. Along those lines, Cai *et al.* (2000) developed encoding schemes, crossover, and mutation operations specifically for job shop problems and showed improved performance when combining the results with a local search. To further improve the efficiency of GA based heuristic approaches, researchers have combined them with other heuristic approaches. For example, Cheung and Zhou (2001) present a hybrid approach to scheduling $Jm|s_{i,j}|C_{max}$ where the first job of each machine is assigned using a GA while the remaining jobs are scheduled using a heuristic. Having the first job selected for each machine reduces the work performed by the heuristic and showed improved objective function performance. Dorndorf and Pesch (1995) proposed a combination GA and SBH approach for job shop scheduling. The GA determines the sequence of machines to be scheduled in the SBH. The sequence generated by the GA replaces the machine criticality decision in the SBH. Candido *et al.* (1998) developed a multi-step scheduling method that relied on two GAs to schedule a complex job shop. Their approach handles multiple routings, sequence dependent setups and batching, where multiple jobs can be processed together. Similar to other approaches they rely on a modified scheduling heuristic based on the work of Giffler and Thompson (1960). This approach is not equipped to handle maximum wait time constraints.

A key component to the performance of a GA is the initial solution. Developing an initial population using a heuristic based approach may improve the GAs performance (Jones *et al.*, 1999). For the classic job shop scheduling problem, GA approaches are often found to be inferior to other heuristic approaches, such as simulated annealing or tabu search (Anderson *et al.*, 1997).

Mattfeld and Bierwirth (2004) present a GA formulation specifically for job shop problems with tardiness objectives. They use a tunable scheduling method that allows for the generation of non-delay, hybrid, and active schedules. In part this tunable scheduling approach modifies the size of the search space as non-delay schedules are a proper subset of active schedules. To address the computational demands of the GA a multistage decomposition is used in which jobs are grouped based on their release times and multiple smaller scheduling problems are considered. They found moderate levels of decomposition (< 6 subsets) greatly decreased computational time with no statistically significant degradation of algorithms performance with respect to sum of average job tardiness, maximum tardiness, and number of tardy jobs. Zhou *et al.* (2009) present a hybrid genetic algorithm specifically designed to minimize weighted tardiness in a job shop environment. In their approach, the first operation of each machine sequence is selected using a GA and the sequencing of all other operations is handled using a variety of dispatching heuristic approaches including EDD (earliest due date), SPT (shortest processing time), the R&M heuristic of Rachamadugu and Morton (1982) and COVERT (cost OVER time). Zhou *et al.* (2009) show that their hybrid approach out performs that GA of Mattfeld and Bierwirth (2004) in most cases. Following the selection of the first operation for each machine using a GA, the R&M heuristic for dispatching all subsequent operations is reported to have the best performance.

3.2 *Job Shop Scheduling with Setups*

3.2.1 Exact Solution Approaches with Setups

The job shop scheduling problem with sequence dependent setups is a challenging NP-Hard problem. Artigues *et al.* (2004) provide an exact solution approach based on branch and bound techniques for solving problems with 10 machines, 15 jobs, and 5 unique setup categories. The algorithm performs well when compared to other similar methods (Brucker and Thiele, 1996).

3.2.2 Shifting Bottleneck with Setups

The shifting bottleneck (SB) heuristic developed by Adams *et al.* (1988) solves the individual machine scheduling sub-problems $1|r_i|C_{max}$. With the inclusion of sequence dependent setups in the main scheduling problem $Jm|s_{i,j}|C_{max}$, the single machine scheduling problem $1|r_i, s_{i,j}|C_{max}$ is NP-Hard (Lenstra *et al.*, 1977) and the algorithm presented by Carlier (1982) is not useful for generating optimal solutions. Given the complexity associated with solving the sub-problem, a heuristic method is used to increase computational efficiency. For the job shop problem with sequence dependent setups and weighted tardiness $Jm|s_{i,j}|\sum w_i t_i$, Lee and Pinedo (1997) present an Apparent Tardiness Cost with Setups (ATCS) dispatching heuristic to handle the SSP portion of the SB heuristic (Step 2). The ATCS dispatching heuristic is similar to the ATC with an additional term to account for the impact of sequence dependent setups. Specifically, the priority of operation q of job j on machine m where operation p of job i was most recently processed is

$$I_{j,q,m}(t,i,p) = \frac{w_j}{\rho_{j,q}} \exp\left(-\max\frac{(d_{j,q} - \rho_{j,q} - t, 0)}{k_1\bar{\rho}}\right) \exp\left(-\frac{s_{i,p,j,q}}{k_2\bar{s}}\right). \tag{3}$$

In Equation (3), k_2 is a scaling parameter and $\bar{s}$ is the average setup time between the most recent operation p of job i and all unscheduled operations that are processed on machine m. Lee *et al.* (1997) and Lee and Pinedo (1997) provide methods for calculating k_1 and k_2.

Job shop scheduling problems with sequence dependent setups often arise in semiconductor manufacturing and testing problems (Gupta and Sivakumar, 2006). Ovacik and Uzsoy (1992) apply a SB based heuristic to semiconductor test scheduling. The authors later improve upon their method by exploiting special structures of the problem and employing a rolling horizon method in the subproblem solution procedure (Ovacik and Uzsoy, 1994, Ovacik and Uzsoy, 1995, Ovacik and Uzsoy, 1996). Mason *et al.* (2002) present a modified shifting bottleneck approach to a semiconductor manufacturing environment which they classify as a complex job shop with batching, re-entrant work flow, job release times, and sequence dependent setups, $\left(Jm\middle|r_i, s_{i,j}, B, recrc\middle|\sum w_i T_i\right)$. Their approach applies a modified ATCS dispatching heuristic that incorporates tool groups capable of batching multiple jobs together. When selecting the most critical machine the authors employ the MCM of Pinedo and Chao (1999) discussed above. Pfund *et al.* (2008) extend the modified shifting bottleneck approach of Mason *et al.* (2002) to consider a multi-criteria objective.

Sun and Noble (1999) utilize the SB framework where a Lagrangian relaxation approach is used to solve the SSP (Step 2) for a problem with sequence dependent setups, release times and a tardiness based objective function, $Jm|s_{i,j}, r_i|\sum w_i t_i$. Pfund *et al.* (2008) utilize a shifting bottleneck procedure for problems with setups, due dates and precedence constraints $Jm|s_{i,j}, d_i|C_{max}$. Balas *et al.* (2008) apply a shifting bottleneck procedure for problems with setups, due dates and precedence constraints $Jm|s_{i,j}, d_i|C_{max}$. Their method considers the single machine scheduling problem in Step 2 as traveling salesman problem with time windows (TSTW) and solve each sub-problem using a dynamic programing approach (Balas and Simonetti, 2001) which has good results in terms of computational complexity.

3.2.3 Neighborhood Search with Setups

While effective for a classic job shop problem, neighborhood search methods are rarely applied to job shop scheduling problems with sequence dependent setups. Neighborhood search methods that apply a classic N_1 search neighborhood to problems with sequence dependent setups, struggle to escape local minima. Additionally, the development of neighborhood definitions that lead to quick generation of feasible neighboring solutions is challenging. For these reasons, the application of TS or SA to job shop problems with sequence dependent setups is rare. In one of the few examples that surfaced, Schmidt (2001) attempts to apply TS to a job shop problem with sequence dependent setups with various neighborhood methods and confirms its application is not efficient.

3.2.4 Hybrid Approaches

Hybrid meta heuristic methods that cast a wider net or use multiple starting locations have been shown to be useful (Kolonko, 1999). Manikas and Chang (2009) present an alternative scatter search that compared favorably to other methods including tabu search, simulated annealing, and genetic algorithms. The neighborhood scheme and termination criteria however, are unclear. Tabu search has also been applied as a part of a larger scheduling method (Pezzella and Merelli, 2000). More recently, Naderi *et al.* (2010) apply a meta heuristic based on simulated annealing to the SDJS problem. The algorithms efficiency is largely influence by the neighborhood scheme and the fine-tuning of parameters such as temperature. Given the difficulty of defining an efficient neighborhood scheme, genetic algorithm based methods tend to be favored for solving complex job shop problems.

3.3 Integrated Process Planning and Scheduling

Integrated process planning and scheduling seeks to address both the process planning tasks and the scheduling tasks fundamental to manufacturing. Of the unique aspects of the C&L job shop environment, due-date performance, sequence dependent setups, and minimum and maximum wait times can all be considered scheduling constraints. The existence of multiple routings, however, falls into the area of process planning. Process planning involves determining detailed instructions (i.e. machine routing, specific resource settings) to produce a product to a set of technical specifications (Tan and Khoshnevis, 2000). Scheduling involves developing a sequence in which jobs will be processed on machines along with their specific start times. Process planning and scheduling are often handled as two disjoint problems, where the outputs of the process planning problem (i.e. machine routings, and processing times) serve as inputs to the scheduling problem. Integrated Process Planning and Scheduling (IPPS) seeks to integrate the decision making in both the process planning and scheduling decisions to improve plant productivity (Li *et al.*, 2010).

Process planning seeks to reduce production costs associated with a given job. Treated independently of the scheduling effort, this single criteria focus can lead to unbalanced machine allocations and infeasible scheduling conditions as well as schedules with unnecessary bottlenecks and setups. Scheduling often seeks to minimize production makespan or job tardiness. Scheduling with fixed process plans can lead to sub-optimal makespans or due-date performance (Tan and Khoshnevis, 2000). Thus solving each of these problems independently may result in limitations. Additionally, their individual objective functions may also be in conflict (Chryssolouris *et al.*, 1985). Minimizing production cost at the process planning level may lead to poor machine balancing and long makespans or poor due-date performance. IPPS leverages process planning decisions along with detailed scheduling decisions and plant floor status to improve manufacturing operations. Using this information IPPS is able to generate realistic processing plans and schedules that are able to take into account multiple production alternatives. Iwata *et al.* (1978) present perhaps the first discussion of IPPS, and develop a branch and bound approach for solving the problem in a job shop environment. Though published 7 years later, Chryssolouris *et al.* (1984) and Chryssolouris *et al.* (1985) are often credited for contributing the first work to the field. Their two papers provide a decision making process that alters process plans when necessary. The review papers of Tan and Khoshnevis (2000) and Li *et al.* (2010) provide background information and current research initiatives for IPPS.

3.3.1 Classes of IPPS Problems

In their review paper Li *et al.* (2010) identify three classes of IPPS research: non-linear process planning (NLPP), closed-loop process planning (CLPP), and distributed process Planning (DPP). NLPP is also known as alternative process planning approach as all alternative processing plans are determined and then plan selection and scheduling decisions are made. CLPP is an online IPPS method that generates processing plans based on current shop floor feedback. DPP simultaneously develops new processing plans and scheduling plans. Given manufacturing constraints of the C&L problem all possible processing plans are known *a priori*. Thus, the C&L problem can be considered an IPPS problem with a non-linear process planning structure (NLPP).

3.3.2 Types of Flexibility

The three types of flexibility in process planning include: operation, sequence, and process flexibility (Benjaafar and Ramakrishnan, 1996). *Operation flexibility* refers to the situation where alternative machines can process an operation. With operation flexibility, the cost and processing time of performing an operation on different machines may vary. *Sequence flexibility* refers to the ability to change the order in which operations of a particular job are processed (Lin and Solberg, 1991). *Processing flexibility* exists when a part feature can be obtained through an alternative operation or set of operations. The C&L problem contains both operation and process flexibility.

3.3.3 Problem Representations

A mathematical representation provides a structure for communicating the complexities and constraints of a complex problem. A graphical depiction of a complex problem serves as a useful tool that can aid in discussion. A good graphical representation can intuitively communicate problem specific information that may otherwise be muddled in pages of data and written details.

Kim and Egbelu (1999) provide a mathematical model for integrated production planning and scheduling. Similar to the C&L problem, their mixed integer linear programming model considers $|N|$ jobs with a predefined number of processing plans in a job shop environment. Their model, however, assumes setups are negligible and seeks to minimize the schedule makespan. Given the computational complexity of the problem, a pre-processing method is suggested which greatly reduces the computational time. For the C&L problem, a mixed integer linear program (MILP) is presented in Section 4.2.

A structured method for displaying processing plans is important to facilitate communication. Much of the work in IPPS relies on a network representation for potential processing plans (Ho and Moodie, 1996). Dummy start and ending nodes are used along with intermediate process nodes for each job. The network representation uses OR nodes to represent alternative routings or processing flexibility. A directed path from the start node to the end node represents a complete processing plan. Figure 3.1 represents a processing plan network for a product WF-A. The vector X can be used to represent a given processing plan where $x_1 = 0$ represents taking the left branch at the OR-1 node.

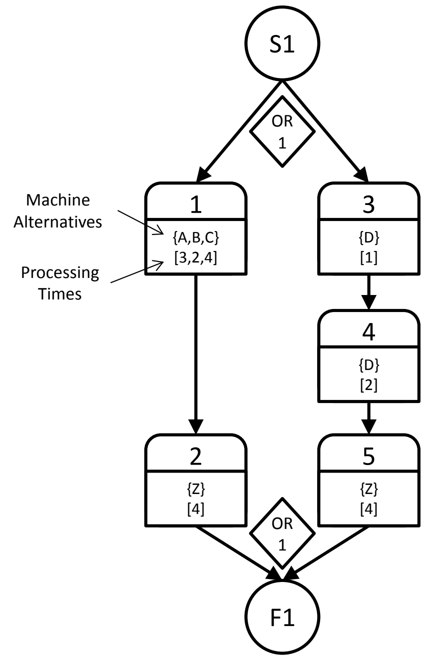


Figure 3.1 Process Plan Flexibility Network

3.3.4 Solution Approaches

The job scheduling problem can be viewed as special simplified case of the IPPS problem where each job has only one processing plan. IPPS for a job shop with sequence dependent setups is an NP-Hard problem as it has the single process plan job shop scheduling problem as a subset. Thus, most approaches for addressing the IPPS rely on heuristic based approaches for developing good and feasible solutions quickly. Researchers have applied a variety of methods heuristic methods to IPPS problems including multi-agent systems, neighborhood search techniques, and genetic algorithms.

3.3.4.1 Multi-Agent Systems

Multi-agent systems (MAS) are often used to solve hierarchical problems of competing objectives. For this reason, researchers have applied agent-based approaches to the IPPS problem. Agents are components or actors which act in such a manner as to accomplish a task on its own behalf (Nwana and Ndumu, 1997). Several reviews have been conducted on the use of agent-based approaches for production planning as well as IPPS (Weiming *et al.*, 2006, Zhang and Xie, 2007). Gu *et al.* (1997) may be the first work that uses a multi-agent based approach for process planning where machine scheduling was also considered for a job. In their model a job-agent and machine-agents bid and negotiate to generate a process plan and schedule for part production. Extending this work to IPPS with multiple parts, Lim and Zhang (2004) apply the MAS approach to support process planning and scheduling in a dynamic environment through the use of contracts and bidding. In their work, a scheduling agent considers current machine loading, upcoming schedules, and job due-dates. Methods for IPPS using MAS have been applied to collaborative manufacturing environments, dynamic production planning problems (DPP), and online IPPS systems (Wang and Shen, 2003, Shukla *et al.*, 2008, Li *et al.*, 2010). Fujii *et al.* (2008) apply an evolutionary artificial neural network based MAS approach to simultaneously solve production planning and scheduling problem. As with most MAS approaches, their focus was on adaptive online process planning and dynamic scheduling decisions and are not currently as well suited for more advanced scheduling problems such as the C&L problem presented here (Weiming *et al.*, 2006).

3.3.4.2 Neighborhood Search

As with the traditional scheduling problem researchers have looked at neighborhood search approaches to solve IPPS problems. Palmer (1996) applies a simulated annealing approach to the IPPS problem with process flexibility and considers various cost functions including tardiness. The neighborhood method used does not guarantee a neighboring solution is feasible. Thus infeasible neighbors must be discarded which increases computational time. More recently, simulated annealing has been applied to IPPS problems with processing, operation sequence, and scheduling flexibility (Hax and Meal, 1975, Li and McMahon, 2007). Weintraub *et al.* (1999) extend the simulation based-approach of Hodgson *et al.* (1998) for job shop scheduling with due-dates and incorporate alternative process plans through a tabu-search method. The simulation based scheduling approach relies on a lower bound estimation technique and dispatching heuristic that is not well suited for environments with sequence dependent setups. Kis (2003) develop a heuristic approach based on a tabu search to solve a job shop scheduling problem with processing alternatives where the neighborhood concept is based on the critical path.

3.3.4.3 Genetic Algorithms

Genetic algorithms have been shown to be successful at solving complex problems. A number of researchers have examined GA based approaches for the IPPS problem (Li *et al.*, 2010). Considering the production planning decision and the scheduling decision as symbiotic partners, Kim *et al.* (2003) develop a genetic algorithm based approach relying on symbiotic evolutionary for IPPS in a job shop setting with operation flexibility (OF), sequencing flexibility (SF) and processing flexibility (PF). The co-evolutionary algorithm they propose handles the scheduling and process planning portions of the problem at the same time. A simple encoding scheme is used and schedule feasibility is ensured through the dispatching heuristic of Giffler and Thompson (1960). Like most other IPPS methods, sequence dependent setups, along with minimum and maximum wait times are not considered.

3.3.4.4 Other Search Techniques

Recently, several techniques designed to search large solution spaces have been applied the IPPS problem. Particle swarm optimization seeks to mimic the movement of a flock of birds or school of fish to explore the solution space (Kennedy, 2010). Guo *et al.* (2009) develop a particle swarm optimization method that is able to solve even complex problems in less time than a GA or SA based approach. Rossi and Dini (2007) utilize a disjunctive graph representation of the IPPS problem with setups and seek to solve the problem using an ant colony based optimization method. Their ant colony approach explores the disjunctive graph in a similar manner to ants exploring a picnic site in search of the best path to food. Kim and Egbelu (1999) develop two heuristic approaches for an NLPP, IPPS problem, where process plans are known *a priori*. Their heuristics relies on a bounding module that is not easily extended to cases of sequence dependent setups.

3.4 Conclusions from Literature

This literature review has highlighted relevant work in the fields of complex job shop scheduling and integrated process planning and scheduling. Only one paper found makes an attempt at job shop scheduling with maximum wait type constraints (Oddi *et al.*, 2011). Their approach however does not include the process planning aspects present in the C&L problem. While research in the field is extensive, the review of literature reveals several gaps for addressing IPPS problems with characteristics of the C&L problem. Table 3.1 lists the characteristics of the C&L problem, along with several popular solution approaches to job shop scheduling problems. The check marks indicate that a given solution approach has been used to address a problem with a specific characteristic.

Table 3.1 Problem Features and Solution Approaches

	SBH	SBH / ATC	Modified SBH	Constraint Programming	IPPS	GA
Job Shop	✓	✓	✓	✓	✓	✓
Due Date		✓	✓	✓	✓	✓
Sequence Dependent Setups			✓	✓		✓
Re-Entry			✓			✓
Multiple Routings					✓	✓
Minimum Wait Time	~	~	~	✓	~	~
Maximum Wait Time				✓		
Primary Source	Adams *et al.* (1988)	Pinedo and Singer (1999)	Mason *et al.* (2002)	Oddi *et al.* (2011)	Li *et al.* (2010)	Candido *et al.* (1998)

✓ Directly Treated

~ Trivial Extension

As seen in Table 3.1, a gap in the literature exists for addressing the complex environment described by the C&L problem. Specifically, no literature was found for addressing the maximum wait time constraint along with the other characteristics.

Additionally, most of the metaheuristics reviewed rely on a dispatching heuristic based on the work of Carlier (1982) for single machine scheduling or Giffler and Thompson (1960) for multiple machine problems. A dispatching heuristic was not found during the literature search to quickly generate feasible solutions to the C&L problem with maximum wait time constraints.

4. Problem Formulation

To address the C&L scheduling problem, a mathematical model for scheduling $|N|$ physical jobs on $|M|$ physical machines is developed. The mathematical model allows for an examination of the structure and computational complexity of the problem. First, a formulation is presented in which jobs are assumed to have just one production version and the machine on which an operation is to be performed is pre-specified. This formulation is equivalent to the scheduling problem that assumes a processing plan decision has been made for all jobs. This initial formulation is then expanded to include the selection of a processing plan as a portion of the model.

4.1 Formulation for Single Process Plan (Route) Scheduling Problem

In this model, jobs must be processed on a given subset of machines. Thus, the machine on which an operation is run and the processing time are inputs to the model.

Indices and Sets

N	Set of physical jobs	
M	Set of machines	
O_i	Set of operations for job i	
$i\ ,j$	Indices for jobs	$i,j = 0,\dots,|N|+1$
$p\ ,q$	Indices for operations	$p\ ,q\ = 1,\dots,|O_i|$

Decision Variables

$X_{i,p,j,q}$	Binary variable that equals 1 if operation p of job i precedes operation q of job j, and 0 otherwise.
$CT_{i,p}$	Completion time of operation p of job i.
t_i	Tardy time of job i.

Parameters

$m_{i,p}$	Routing parameter indicating the machine on which operation p of job i is processed.
$\rho_{i,p}$	Processing time required to perform operation p of job i.

$s_{i,p,j,q}$	Setup time required between operation p of job i and operation q of job j.
$h_{i,p}$	Holding time required after operation p of job i.
$\omega_{i,p}$	Maximum wait (idle) time allowed after operation p of job i, where $\omega_{i,p}$ equals some large number if no maximum wait time is specified.
d_i	Due-date or desired completion time for job i.
$\mathcal{M}$	Large positive number.

The following assumptions are included in the single production plan formulation.

1. The order in which a job visits machines and the processing time on each machine is fixed *a priori*.
2. No preemption is allowed, once an operation starts on a machine it must be completed.
3. A machine, m, can process at most one operation at a given time.
4. Jobs 0 and $|N| + 1$ are dummy jobs that are the first and last jobs to run on each machine respectively. Dummy jobs have zero processing time.
5. Sequence dependent setup time, $s_{i,p,j,q}$, depends on both operation p of job i which most recently was processed on machine m and operation q that will be processed next on machine m.
6. Setups are separable, such that the setup for operation q of job j on machine m, following operation p of job i, $s_{i,p,j,q}$, can be performed even if job j has not completed its previous operation, as long as operation p is completed and machine m is idle.
7. The objective function is based on total tardiness, the positive difference between the completion time, $CT_{i,|O_i|}$, and due date, d_i, for each job.

4.1.1 Graphical Representation

The sequence dependent job shop scheduling problem can be modeled as a disjunctive graph $G = (N, Z, X)$, where each operation is represented on the graph as a node in N. Each node is referenced using the (job, operation number, machine) triplet (i, p, m). Operation precedence constraints are represented using directed arcs in Z connecting node (i, p, m) to node $(i, p + 1, m')$. Operations to be executed on the same machine are initially linked through an undirected arc set X. This undirected arc set represents all potential sequences on a given machine. The sequencing of jobs on a single machine involves finding a directed chain that connects each

(i, p, m) node that is processed on machine m. Figure 4.1 illustrates a disjunctive graph for a problem 3 jobs, 5 processes, and 5 machines. For simplicity we do not include the dummy job 0 or $N + 1$ in this figure. Figure 4.2 depicts just the nodes and disjunctive arcs associated with machine 3. The costs on arcs in Figure 4.2 represent the processing time for operation p of job i (tail node) plus the sequence dependent setup time required to begin processing the operation associated with the head node. This graphical insight inspires the formulation of the mathematical model presented in Section 4.1.2.

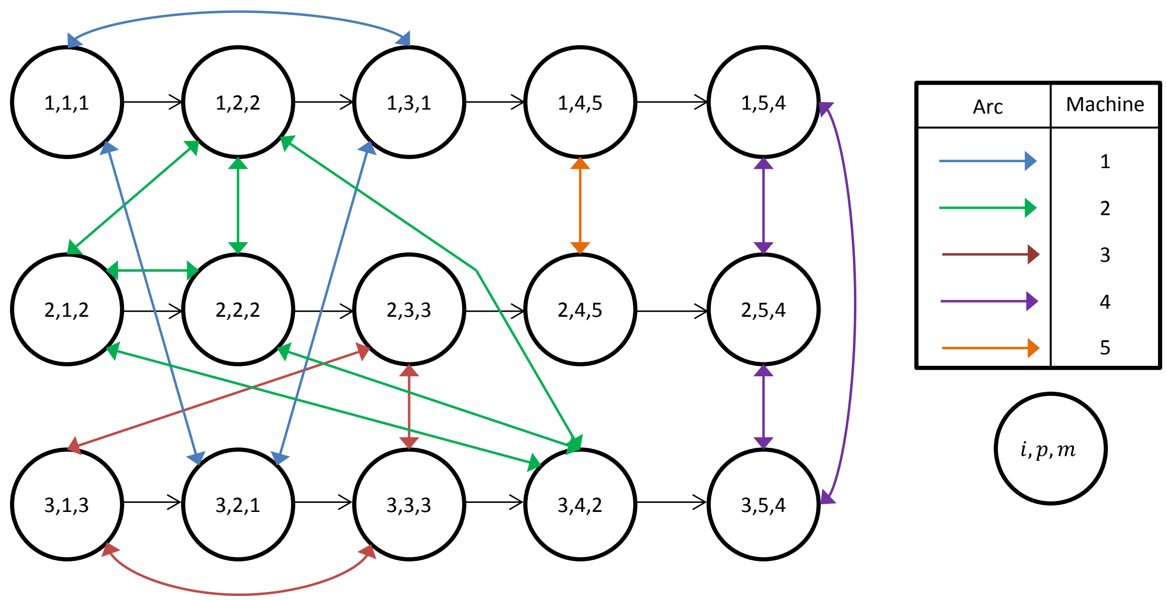


Figure 4.1 Disjunctive Graph Representation

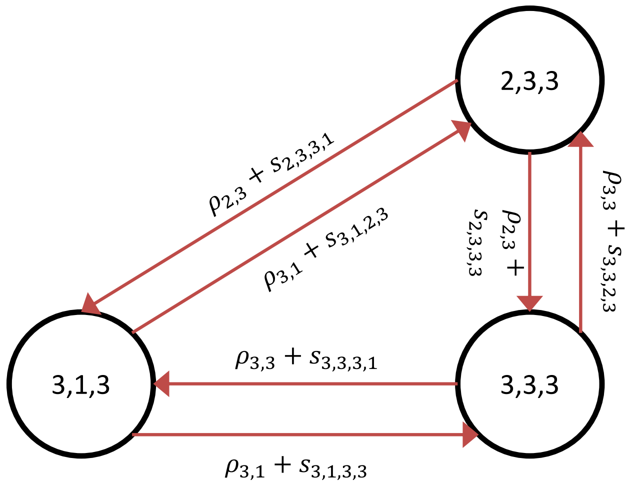


Figure 4.2 Machine Three Sequence Graph

4.1.2 Model Formulation

Minimize $\sum_{i=1}^{|N|} t_i$

Subject to:

$$\sum_{j=1}^{|N|+1} \sum_{q=1}^{|O_j|} X_{i,p,j,q} = 1 \qquad \forall i = 0, \dots, |N|, \quad \forall p = 1, \dots, |O_i| \tag{4}$$

$$\sum_{i=0}^{|N|} \sum_{p=1}^{|O_i|} X_{i,p,j,q} = 1 \qquad \forall j = 1, \dots, |N| + 1, \quad \forall\, q = 1, \dots, |O_j| \tag{5}$$

$$\sum_{j=1}^{|N|+1} \sum_{q=1}^{|O_j|} X_{0,p,j,q} = 1 \qquad \forall p = 1, \dots, |O_0| \tag{6}$$

$$\sum_{i=0}^{|N|} \sum_{p=1}^{|O_i|} X_{i,p,|N|+1,q} = 1 \qquad \forall q = 1, \dots, |O_{|N|+1}| \tag{7}$$

$$X_{i,p,i,p} = 0 \qquad \forall i, \quad \forall p = 1, \dots, |O_i| \tag{8}$$

$$CT_{i,p} + \left(s_{i,p,j,q} + \rho_{j,q}\right)X_{i,p,j,q} \leq CT_{j,q} + \left(1 - X_{i,p,j,q}\right)\mathcal{M} \qquad \forall i = 0, \dots, |N|\,, \; \forall j = 0, \dots, |N| + 1, \; \forall p = 1, \dots, |O_i|, \; \forall q = 1, \dots, |O_j| \tag{9}$$

$$CT_{i,p+1} \geq CT_{i,p} + h_{i,p} + \rho_{i,p+1} \qquad \forall i = 1, \dots, |N|, \; \forall p = 1, .., |O_i| - 1 \tag{10}$$

$$CT_{i,p+1} \leq CT_{i,p} + \omega_{i,p} + \rho_{i,p+1} \qquad \forall i = 1, \dots, |N|, \; \forall p = 1, .., |O_i| - 1 \tag{11}$$

$$CT_{i,p} - d_i \leq t_i \qquad \forall i = 1, \dots, |N|, \; \forall p = 1, \dots, |O_i| \tag{12}$$

$$CT_{i,p} \geq 0, t_i \geq 0, \qquad X_{i,p,j,q} \in 0,1 \qquad \forall i, j, p, q \tag{13}$$

$$X_{i,p,j,q} = 0 \qquad \forall i, p, j, q \mid m_{i,p} = m_{j,q} \tag{14}$$

The objective function minimizes the sum of job tardiness, t_i. Constraint sets (4) and (5) ensure that exactly one operation precedes every non-dummy operation and likewise only one operation succeeds each non-dummy operation. Constraints (6) and (7) establish that exactly one operation succeeds each operation of dummy job 0 and one operation precedes each operation of dummy job $|N| + 1$. Constraint set (8) ensures that an operation does not follow itself. Constraint set (9) calculates the completion time for each operation and ensures that only one operation is processed on a machine at a time. Operation precedence constraints including minimum and maximum wait time constraints are articulated in constraint sets (10) and (11). Constraint set (12) captures job tardiness. Finally, constraint set (13) includes basic non-negativity constraints and constraint set (14) ensures that operation q of job j can only follow operation p of job i if they are share the same machine, as indicated by $m_{i,p} = m_{j,q}$.

4.2 Formulation for Integrated Process Planning and Scheduling

While the single production plan formulation provides insight on structure and complexity, it does not fully capture the complexity of the C&L production planning and scheduling problem. Thus an integrated production planning and scheduling (IPPS) model is presented.

Indices and Sets

N	Set of physical jobs	
M	Set of machines	
V_i	Set of production plans for job i	
$O_{i,v}$	Set of operations in production plan v of job i	
$i\ ,j$	Indices for jobs	$i,j = 0, \dots, \lvert N \rvert + 1$
v, h	Indices for production plans	$v, h = 1, \dots, \lvert V_i \rvert$
p, q	Indices for operations	$p\ ,q\ = 1, \dots, \lvert O_{i,v} \rvert$

Decision Variables

$Y_{i,v}$	Binary variable that equals 1 if job i uses production plan v, and 0 otherwise.
$X_{i,v,p,j,h,q}$	Binary variable that equals 1 if operation p from production plan v of job i precedes operation q of production plan h of job j, and 0 otherwise.
$CT_{i,v,p}$	Completion time of operation p of production plan v for job i.
t_i	Tardy time of job i.

Parameters / Coefficients

$m_{i,v,p}$	Routing parameter indicating the machine on which operation p in production plan v of job i is processed.
$\rho_{i,v,p}$	Processing time required to perform operation p from production plan v for job i.
$s_{i,v,p,j,h,q}$	Setup time required between operation p of production plan v for job i and operation j of production plan h for job j.
$h_{i,v,p}$	Holding time required after operation p in production plan v of job i.
$\omega_{i,v,p}$	Maximum wait (idle) time allowed after operation p in production plan v of job i. $\omega_{i,v,p}$ equals some large number if the operation does not have a maximum wait time specified.
d_i	The desired completion time for job i.
$\mathcal{M}$	Large positive number.

4.2.1 Model Formulation

The following assumptions are included in the IPPS formulation.

1. Exactly one processing plan, v, is selected for each job, i.
2. The order in which a job visits machines and the time processing time on each machine is based on the processing plan, v, selected.
3. No preemption is allowed; such that once an operation starts on a machine is must be completed.
4. A machine m can process at most one operation at a given time.
5. Jobs 0 and $|N| + 1$ are dummy jobs that are the first and last jobs to run on each machine respectively. Dummy jobs have zero processing time.
6. Sequence dependent setup time, $s_{i,v,p,j,h,q}$, depends on both operation p of production plan v for job i which most recently was processed on machine m and operation q of production plan h for job j, that will be processed next on machine m.
7. Setups are separable, such that the setup for operation q of production plan v for job j on machine m, following operation p, $s_{i,v,p,j,h,q}$, can be performed even if job j has not completed its previous operation, as long as operation p is completed and machine m is idle.
8. The objective function is based on total tardiness, the positive difference between each job's completion time and its due date d_i.

Minimize $\sum_{i=1}^{|N|} t_i$

Subject to:

$$\sum_{v=1}^{|V_i|} Y_{i,v} = 1 \qquad \forall i = 0, \dots, |N| + 1 \qquad (15)$$

$$\sum_{j=1}^{|N|+1} \sum_{h=1}^{|V_j|} \sum_{q=1}^{|O_{j,h}|} X_{i,v,p,j,h,q} = Y_{i,v} \qquad \forall i = 0, \dots, |N|, \ \forall v = 1, \dots, |V_i|, \ \forall p = 1, \dots, |O_{i,v}| \qquad (16)$$

$$\sum_{i=0}^{|N|} \sum_{v=1}^{|V_i|} \sum_{p=1}^{|O_{i,v}|} X_{i,v,p,j,h,q} = Y_{j,h} \qquad \forall j = 1, \dots, |N| + 1, \ \forall h = 1, \dots, |V_j|, \ \forall q = 1, \dots, |O_{j,h}| \qquad (17)$$

$$\sum_{j=1}^{|N|+1} \sum_{h=1}^{|V_j|} \sum_{q=1}^{|O_{j,v}|} X_{0,v_o,p,j,h,q} = 1 \qquad \forall p = 1, \dots, |O_{o,v_o}| \tag{18}$$

$$\sum_{i=0}^{|N|} \sum_{v=1}^{|V_i|} \sum_{p=1}^{|O_{i,v}|} X_{i,v,p,|N|+1,h_o,q} = 1 \qquad \forall q = 1, \dots, |O_{|N+1|,h_o}| \tag{19}$$

$$X_{i,v,p,i,v,p} = 0 \qquad \forall i, \; \forall v = 1, \dots, |V_i|, \; \forall p = 1, \dots, |O_{i,v}| \tag{20}$$

$$\begin{aligned} CT_{i,v,p} + s_{i,v,p,j,h,q} + \rho_{j,h,q} &\le CT_{j,h,q} \\ &+ (1 - X_{i,v,p,j,h,q})\mathcal{M} \\ &+ (1 - Y_{i,v})\mathcal{M} + (1 - Y_{j,h})\mathcal{M} \end{aligned} \qquad \begin{aligned} &\forall i = 0, \dots, |N|, \\ &\forall j = 1, \dots, |N| + 1, \\ &\forall v = 1, \dots, |V_i|, \\ &\forall h = 1, \dots, |V_j|, \\ &\forall p = 1, \dots, |O_{i,v}|, \\ &\forall q = 1, \dots, |O_{j,h}| \end{aligned} \tag{21}$$

$$\begin{aligned} CT_{i,v,p+1} \ge CT_{i,v,p} + h_{i,v,p} + \rho_{i,v,p+1} \\ - \mathcal{M}(1 - Y_{i,v}) \end{aligned} \qquad \begin{aligned} &\forall i = 1, \dots, |N|, \\ &\forall v = 1, \dots, |V_i|, \\ &\forall p = 1, \dots, |O_{i,v}| - 1 \end{aligned} \tag{22}$$

$$CT_{i,v,p+1} \le CT_{i,v,p} + w_{i,vp} + \rho_{i,v,p+1} \qquad \begin{aligned} &\forall i = 1, \dots, |N|, \\ &\forall v = 1, \dots, |V_i|, \\ &\forall p = 1, \dots, |O_{i,v}| - 1 \end{aligned} \tag{23}$$

$$CT_{i,v,|O_{i,v}|} - d_i - \mathcal{M}(1 - Y_{i,v}) \le t_i \qquad \begin{aligned} &\forall i = 1, \dots, |N|, \\ &\forall v = 1, \dots, |V_i| \end{aligned} \tag{24}$$

$$CT_{i,v,p} \ge 0, t_i \ge 0, \qquad X_{i,v,p,j,h,q} \in 0,1 \qquad \forall i, j, v, h, p, q \tag{25}$$

$$X_{i,v,p,j,h,q} = 0 \qquad \forall i, j, v, h, p, q \; | m_{i,v,p} = m_{j,h,q} \tag{26}$$

The objective function minimizes sum of job tardiness, t_i, across all jobs. Exactly one process plan must be selected for each job, as required by constraint set (15). Constraint sets (16) and (17) ensure that exactly one operation precedes every non-dummy operation and likewise only one operation succeeds, each non-dummy operation, and that operations are only sequenced if the process plan to which they belong is selected. Constraints (18) and (19) establish that exactly one operation succeeds each operation of dummy job 0 and one operation precedes each operation of dummy job $|N| + 1$. Constraint set (20) ensures that an operation does not follow itself. Constraint set (21) calculates the completion time for each operation and ensures that only one operation is processed on a machine at a time. Operation precedence constraints including minimum and maximum wait time constraints are articulated in constraint sets (22) and (23). Constraint set (24) captures job tardiness. Finally, constraint set (25) includes basic non-negativity constraints and constraint set (26) ensures that operation q of production plan h from job j can only follow operation p of production plan v from job i if they share the same machine, as indicated by $m_{i,v,p} = m_{j,h,q}$.

5. Proxy Based Local Search for IPPS

Process planning decisions influence the performance of the schedules generated. For example, for a parallel machine problem, processing all jobs on one machine, while the other machine remain idle, will likely result in poor performance with respect to makespan, flowtime and other objectives. For a parallel machine problem, distributing the jobs among multiple machines tends to improve scheduling objective function performance with respect to makespan, flowtime, and other objectives. As the scheduling environment increases in complexity, however, it is more difficult to develop these intuitive insights regarding the effects of process planning decisions on objective function performance.

In this research, a novel proxy based local search approach is developed to address both process planning and scheduling decisions. In this approach, descriptive statistics are used to develop insights on the overall process planning decisions for a given demand set. For example, total processing time on each machine, ρ_m^{total}, total number of operations on each machine, o_m^{total}, machine load balance which can be measured by the standard deviation of machine processing times, σ_ρ, and the standard deviation of machine operation count, σ_o, are used to determine preferred process planning decisions to improve the scheduling performance measure. Relating objective function performance to these statistics may provide insights into process planning decisions.

The proxy based local search (PBLS) approach developed in this dissertation involves the following three main stages.

- Generate a set of feasible schedules for set of jobs in a complex job shop.
- Evaluate the parameters and results of the schedules to establish a proxy measure that will estimate the effect of process planning decisions on objective function performance.
- Apply local search methods to improve upon feasible schedules.

In the following sections each stage is discussed in more detail and the overarching concept is summarized in Section 5.4.

5.1 Generate Feasible Schedules using a Scheduling Heuristic

In scheduling, dispatching heuristics are often used to quickly generate feasible job schedules. These heuristics iteratively assign jobs to machines based on some ranking logic. Simple dispatching heuristics, such as shortest processing time (SPT), earliest due date (EDD), and least work remaining (LWR), have been used in the literature (Hopp and Spearman, 2008). While a variety of dispatching heuristics exist in the literature, none were found that simultaneously address sequence dependent setups, wait time windows, and due date performance for job shop scheduling problems. In this section the unique attributes of the C&L scheduling problem are discussed and a dispatching heuristic to generate feasible schedules to the problem is described. For simplicity, the production plan subscript is dropped as production plans are assumed fixed when applying the scheduling heuristic.

5.1.1 Minimum Wait Constraint

A minimum wait or holding time is required between some operations in the manufacturing process to allow for chemical curing to take place. These wait times extend the traditional precedence constraint to include $h_{i,p}$ which is the minimum amount of time required between the completion of operation p of job i and the downstream operation $p+1$. For operations that do not require additional wait time Equation (27) is an operation precedence constraint with $h_{i,p} = 0$.

$$ST_{i,p+1} \geq CT_{i,p} + h_{i,p}, \tag{27}$$

5.1.2 Maximum Wait Constraint

To ensure production quality, some steps in the manufacturing process must be completed within a set time window of one another. In addition to the traditional operation precedence constraint, (27), jobs with maximum wait time constraints are considered in the C&L scheduling problem. In the case of a maximum wait time constraint the downstream operation must be started within some set time window of the completion of the upstream operation such that:

$$ST_{i,p+1} \leq CT_{i,p} + \omega_{i,p} \tag{28}$$

where $\omega_{i,p}$ is the maximum amount of time allowed to elapse between the completion of operation p and the start of operation $p+1$ for job i. When generating a feasible schedule, this maximum wait time constraint must be considered.

In the absence of maximum wait time constraints, an operation's ready time, $r_{i,p}$, is the completion time of the upstream operation plus any holding or transportation time as shown in Equation (29).

$$r_{i,p} = CT_{i,p-1} + h_{i,p-1} \tag{29}$$

If an operation is constrained by a downstream maximum wait constraint, the ready time depends not only on the previous operation but also on the availability of the downstream machine. In cases where multiple operations of a given job have maximum wait style constraints, the ready time of an upstream operation depends on the availability of all downstream operations.

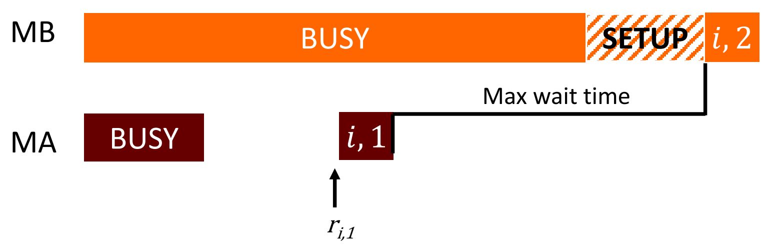


Figure 5.1 Maximum Wait Constrained Ready Time

As illustrated in Figure 5.1, the ready time for operation 1 for job i is constrained by the earliest time that operation 2 can be processed on machine A. If operation 1 is executed anytime prior to $r_{i,1}$ the maximum wait time between operations 1 and 2 is violated. Therefore, in the presence of a maximum wait constraint, the ready time of an operation p for job i is calculated as the maximum of two terms: the completion of the upstream operation, $CT_{i,p-1}$ plus the minimum wait time between operation $p-1$ and operation p, and the ready time of the downstream machine minus the current operations processing time and maximum wait.

$$r_{i,p} = \max\{CT_{i,p-1} + h_{i,p-1}, m_{i,p+1}(t) + s_{x,x,i,p+1} - \rho_{i,p} - \omega_{i,p}\} \tag{30}$$

In this expression, $m_{i,p+1}(t)$ is the ready time of the machine on which operation $p+1$ of job i is processed. Setup time required to begin processing operation $p+1$ is denoted by $s_{x,x,i,p+1}$, $\rho_{i,p}$ is the processing time of operation p of job i, and $\omega_{i,p}$ is the maximum wait time between operation p and operation $p+1$. The first portion of the ready time calculation is the basic operation precedence constraint with a minimum wait time constraint, and the second term accounts for the maximum wait time constraint. Calculating ready times for operations based on Equation (30) is critical for generating feasible schedules to the C&L problem.

5.1.3 Ovacik and Uzsoy Dispatching Heuristic

A variety of ranking methods have been used in conjunction with dispatching heuristics in the literature. Rules such as Earliest Due Date (EDD), Modified Due Date (MDD), COVERT, and S/ONP have been used in cases where the emphasis due date based performance. While these rules have shown promise in a variety of applications, they do not directly consider sequence dependent setups, a key aspect of the C&L problem.

For scheduling in a complex job shop environment, Ovacik and Uzsoy (1994) develop a dispatching type methodology that considers the jobs waiting in queue on a given machine along with additional shop floor information when selecting, referred to here forth as the OU heuristic. Their approach uses a deterministic simulation to expand the candidate set of jobs that could be scheduled on a given machine. They select the operation to schedule based on the results of this look-a-head (LA) simulation. The LA simulation is a sub-problem that sequences a small number, β, jobs and determines the optimal sequence for minimizing makespan of the sub-problem. This approach provides adequate results to complex job shop problems without a significant sacrifice to computational speed. The original heuristic is presented below. Section 5.1.4 presents amendments required to address the maximum wait constraints present in the C&L problem and a modified OU heuristic, OU-MW, is presented in Section 5.1.5.

Step 1: Let A be all operations whose precedence constraints have been met. In the case of initialization $A = \{o_{i,1} | i \in N\}$.

Step 2: Select machine m^* which satisfies $min\{m(t)\}$. In the case of a tie then select the operation o_{i^+,p^+} from set A that has the earliest due date. Let $m^* = m_{i^+,p^+}$

Step 3: Use earliest due date to simulate the sequencing of all operations in queue on machines $m \neq m^*$. Once operation $o_{j,q}$ has been "scheduled" the simulated ready time of operation $o_{j,q+1}$ is recorded as $\hat{r}_{j,q+1} = \widehat{CT}_{j,q} + h_{j,q}$

Step 4: Let A^{m^*} be all operations whose precedence constraints have been met and thus are in queue at machine m^*. Thus $A^{m^*} = \{o_{i,p} \in A | m_{i,p} = m^*\}$.

Step 5: Identify the operation $o_{i',p'}$ from A^{m^*} with the earliest due date. Let, $\mathbf{T} = m^*(t) + \rho_{i',p'} + s_{x,x,i',p'}$ where $s_{x,x,i'p'}$ is the setup time required to begin processing operation $o_{i',p'}$ on machine m^*.

Step 6: Let set $J = \{o_{i,p} \in A^{m^*} | r_{i,p} \leq \mathbf{T}\}$.

Step 7: Let B^{m^*} be all operations not currently in A^{m^*} but requires processing on machine m^* and whose precedence constraint is one operation from being met.

Step 8: Include in J any operation from B^{m^*} whose simulated ready time on machine m^*, $\hat{r}_{i,p+1}$, is less than $\mathbf{T}$. The updated set can be expressed as $J = J \cup \{o_{i,p} | \hat{r}_{i,p} \leq \mathbf{T}, m_{i,p} = m^*\}$

Step 9: Select operation o_{i^*,p^*} from J using some priority method and sequence on machine m^*.

- a: LAJ: Select operation o_{i^*,p^*} from set J with earliest job due date
- b: LAO: Select operation o_{i^*,p^*} from set J whose operation on machine m^* has the earliest local due date. Where, local due date, $d_{i,p} = d_i - 1.25 \sum_{q=p+1}^{|O_i|} \rho_{i,q} + h_{i,q}$.
- c: $LAJ(\beta)$: Select the β operations from set J with the earliest due dates. These operations are then sequenced optimally with respect to minimizing make span. The first operation in the optimal sequence, o_{i^*,p^*} is selected to be scheduled.
- d: $LAO(\beta)$: Select the β operations from set J with the earliest local due dates. These operations are then sequenced optimally with respect to minimizing make span. The first operation in the optimal sequence, o_{i^*,p^*} is selected to be scheduled.

Step 10: Sequence operation o_{i^*,p^*} on machine m^* and remove it from set A.

- a: If operation o_{i^*,p^*} is not currently in queue for machine m^*, operation o_{i^*,p^*-1} must first be sequenced on its respective machine.
- b: Sequence o_{i^*,p^*-1} on machine m_{i^*,p^*-1} by fixing the sequence generated to support Step 3.

Step 11: If o_{i^*,p^*+1} exists, add it to set A.

Step 12: If $A = \emptyset$ terminate, otherwise update the earliest operation start time r, and machine ready times, $m(t)$, for all operations in set A and go to Step 2.

The OU heuristic is able to generate better results than a myopic dispatching heuristic because operations that are not yet waiting in the queue are considered (Step 7). Additionally, the $LAJ(\beta)$ and $LAO(\beta)$ approaches in Step 9 uses a makespan optimization, which lends itself well to improving the solution quality of job shop problems with sequence dependent setups.

5.1.4 Amendments to OU Heuristic for Maximum Wait

In this section five amendments to the original heuristic are presented which are required to guarantee feasibility in the C&L problem.

Amendment 1: After sequencing an upstream maximum wait operation, m^* should be selected as the machine on which the downstream maximum wait operation is to be processed.

Proof 1: Assume such a consideration is not imposed.

Consider the case where $o_{i,p}$ has just been sequenced and $o_{i,p+1}$ be an downstream maximum wait operation, thus

$$CT_{i,p+1} - \rho_{i,p+1} \leq CT_{i,p} + \omega_{i,p}. \tag{31}$$

Assume, operation $o_{i,p+1}$ requires processing on machine m'. Without Amendment 1 machine m^* is selected in Step 2 such that $m^* \neq m'$.

Let operation $o_{j,q}$ be selected for scheduling on machine m^*, and assume $o_{j,q}$ is an upstream maximum wait operation and $o_{j,q+1}$ also requires processing on machine m'. Therefore, following scheduling $o_{j,q}$, both operations $o_{j,q+1}$and $o_{i,p+1}$ are in queue at machine m', $A^{m'} \supseteq \{o_{j,q+1}, o_{i,p+1}\}$.

For simplicity and without loss of generality, assume

$$m'(t) = 0, CT_{i,p} = CT_{j,q}, \omega_{i,p} = \omega_{j,q} \tag{32}$$

and all setups are negligible, $S_{x,x,y,y} = 0$.

From the machine precedence constraint (implied by constraint 6 in the MILP problem from Section 4.1.2) assuming operation $o_{j,q+1}$ is sequenced before operation $o_{i,p+1}$ leads to

$$CT_{j,q+1} + \rho_{i,p+1} \leq CT_{i,p+1}. \tag{33}$$

Operation precedence constraint requires

$$CT_{j,q+1} \geq CT_{j,q} + \rho_{j,q+1} \tag{34}$$

Where

$$\rho_{i,p+1} = \rho_{j,q+1} = \omega_{j,q} + \epsilon = \omega_{i,p} + \epsilon. \tag{35}$$

From (32) $CT_{j,q} = CT_{i,p}$ and from (35) $\rho_{j,q+1} = \omega_{i,p} + \epsilon$. Therefore, the appropriate substitutions in (34) lead to

$$CT_{j,q+1} \geq CT_{i,p} + \omega_{i,p} + \epsilon. \tag{36}$$

Substituting $CT_{j,q+1}$ from (36) into machine precedence constraint in (33) results in

$$CT_{i,p} + \omega_{i,p} + \epsilon \leq CT_{i,p+1} - \rho_{i,p+1} \tag{37}$$

which violates the maximum wait constraint in (31). Following the same steps it can be shown that the maximum wait constraint for operation $o_{j,q+1}$ is violated if $o_{i,p+1}$ is sequenced before $o_{j,q+1}$ on machine m'.

□

Amendment 2: Include any downstream maximum wait constrained operation in set J. Additionally, if $LAJ(\beta)$ or $LAO(\beta)$ are used in Step 10, any downstream maximum wait constrained operation must be included in the set of β operations.

Proof 2: Assume such a consideration is not imposed.

From Equation (28) and machine precedence constraint (implied by constraint 6 in the MILP formulation from Section 4.1.2),

$$m^*(t) + s_{x,x,i,p} \leq ST_{i,p} \leq CT_{i,p-1} + \omega_{i,p-1}. \tag{38}$$

For simplicity and without loss of generality, assume $m^*(t) = 0$, leading to

$$s_{x,x,i,p} \leq CT_{i,p-1} + \omega_{i,p-1}. \tag{39}$$

Additionally, assume $h_{i,p} = 0$ which implies from Equation (7) that $r_{i,p} = CT_{i,p-1}$. Let $o_{j,q} \in A^{m^*}$ have a due date such that $d_j < d_z \forall o_{z,x} \in A^{m^*}$. This implies

$$\mathbf{T} = \rho_{j,q} + s_{x,x,j,q}. \tag{40}$$

From Step 5 of the OU heuristic in Section 5.1.3,

$$o_{i,p} \notin J \text{ if } r_{i,p} > \mathbf{T}. \tag{41}$$

Assuming operation $o_{i,p}$ is not considered in J per (41) and $o_{j,q}$, is selected for scheduling in Step 9, then

$$m^*(t) \geq s_{x,x,j,q} + \rho_{j,q}. \tag{42}$$

This implies that

$$CT_{i,p} \geq s_{x,x,j,q} + \rho_{j,q} + s_{j,q,i,p} + \rho_{i,p} \tag{43}$$

The maximum wait constraint on wait time constraint on operation $o_{i,p}$ from Equation (28) is violated for any case where

$$\rho_{j,q} + s_{x,x,j,q} + s_{j,q,i,p} > \omega_{i,p-1}. \tag{44}$$

□

Amendment 3: The clear time, which refers to the time required for an operation to be completed on a machine and the machine to be prepared for another operation, must be considered when establishing set J in Step 7 and expanding set J in Step 9. This results in $J = \{o_{i,p} \in A^{m^*} | r_{i,p} \leq \mathbf{T}, r_{i,p} + s_{i,p,j,q} < CT_{j,q-1} + \omega_{j,q-1}\}$ (Step 7) , and
$J = J \cup \{o_{i,p} | \hat{r}_{i,p} \leq \mathbf{T}, m_{i,p} = m^*, r_{i,p} + s_{i,p,j,q} < CT_{j,q-1} + \omega_{j,q-1}\}$ (Step 9).

Proof 3: Assume such a condition is not imposed.
Consider a set J with two operations, $o_{i,p}$ and $o_{j,q}$, where $o_{j,q}$ is maximum wait time constrained.

When sequencing with LAO or LAJ, operation $o_{i,p}$ is selected due to more pressing due date. For all,

$$\rho_{i,p} + s_{i,p,j,q} > CT_{j,q-1} + \omega_{j,q-1}, \tag{45}$$

the maximum wait time constraint imposed on operation $o_{j,q}$ is violated.

□

Amendment 4: Sequencing using EDD in Step 3 should stop after an upstream maximum wait constrained operation is sequenced.

Proof 4: By example, consider the case where $o_{1,2}$ is in queue on machine m^* and due to a maximum wait time constraint the latest start time for operation $o_{1,2}$ is $ST_{1,2}^{max}$. For simplicity and without loss of generality assume $m^*(t) = 0$. Operations $o_{2,1}$ and $o_{3,1}$ are in queue on machine m', and $m'(t) = 0$. Both downstream operations $o_{2,2}$ and $o_{3,2}$ are maximum wait constrained and must be carried out on machine m^*.

Sequencing machine m' as per Step 3, assume operation $o_{2,1}$ takes precedence over operation $o_{3,1}$ implying $\hat{r}_{2,2} < \hat{r}_{3,2}$.

Applying Amendment 3 to the formation of set J, only the clear time of operation $o_{3,2}$ satisfies the latest start time constraint of operation $o_{1,2}$,

$$\max\{\rho_{2,1}, s_{x,x,2,2}\} + \rho_{2,2} + s_{2,2,1,2} > ST_{1,2}^{max} \tag{46}$$

$$\max\{\rho_{3,1}, s_{x,x,3,2}\} + \rho_{3,2} + s_{3,2,1,2} < ST_{1,2}^{max}. \tag{47}$$

Therefore, operation $o_{2,2}$ is not included in set J and the resulting set $J = \{o_{1,2}, o_{3,2}\}$. Now let operation $o_{3,2}$, be the operation selected in Step 9. By selecting operation $o_{3,2}$ the sequence on machine m' must be fixed. Taking the sequence from Step 3 both operations $o_{2,2}$ and $o_{3,2}$ arrive at machine m^*. It has been determined that operation $o_{3,2}$ will be sequenced next. After scheduling $o_{3,2}$ machine queue for machine m^*, $A^{m^*} = \{o_{1,2}, o_{2,2}\}$. Both operations in A^{m^*} are maximum wait constrained with latest start time $ST_{1,2}^{max}$, and $ST_{2,2}^{max}$.

From Amendment 1, machine m^* is considered for sequencing again as the queue A^{m^*} still contains a maximum wait time constrained downstream operation.

Consider the two possible sequences on machine m^* $\{o_{1,2}, o_{2,2}\}$ and $\{o_{2,2}, o_{1,2}\}$. Sequence $\{o_{2,2}, o_{1,2}\}$ is infeasible by (46) above. Alternatively considering first $\{o_{1,2}, o_{2,2}\}$. This sequence is infeasible for any case where $s_{3,2,1,2} + \rho_{1,2} + s_{1,2,2,2} > \omega_{1,1}$.

□

To provide a more concrete demonstration a numerical example is provided.

Table 5.1 Setup Matrix

	1,2	**2,2**	**3,2**
1,2	-	10	5
2,2	10	-	10
3,2	2.5	10	-
Empty	0	0	0

Table 5.2 Operation Process Time and Maximum Wait Details

Operation Id	**Process Time**	**Max Wait**
1,1	10	12
1,2	10	-
2,1	5	12
2,2	10	-
3,1	5	12
3,2	10	-

Table 5.3 Job Due Dates

Job Id	**Due Date**
1	100
2	50
3	75

Assume, $ST_{1,2}^{max} = 30$. Using Equation (30)

$$r_{2,1} = \max\{0, 0 - 5 - 12\} = 0,$$

$$r_{3,1} = \max\{0, 0 - 5 - 12\} = 0.$$

Calculating the simulated ready time to support Step 3:

$$\hat{r}_{2,2} = \rho_{2,1} = 5$$

$$\hat{r}_{3,2} = \rho_{2,1} + s_{2,1,3,1} + \rho_{3,1} = 10$$

$$\widehat{ST}_{2,2}^{max} = 17$$

$$\widehat{ST}_{3,2}^{max} = 22.$$

Calculating the clear times for operations $o_{2,2}$ and $o_{3,2}$,

$$o_{2,2}: \hat{r}_{2,2} + \rho_{2,2} + s_{2,2,1,2} = 5 + 10 + 10 = 30$$

$$o_{3,2}:\ \hat{r}_{3,2} + \rho_{3,2} + s_{3,2,1,2} = 10 + 10 + 2.5 = 22.5.$$

Set $J = \{o_{1,2}, o_{3,2}\}$, since $o_{2,2,}$ does not satisfy the clearing time condition introduced by Amendment 3. Assuming $LAJ(\beta)$ is used in Step 9, the optimal sequence of operations from set J on machine m^* with respect to makespan is identified:

Makespan sequence $\{o_{1,2}, o_{3,2}\}$: $\hat{r}_{1,2} + \rho_{1,2} +\ s_{1,2,3,2} + \rho_{3,2} = 43$

Makespan sequence $\{o_{3,2}, o_{1,2}\}$: $\hat{r}_{3,2} + \rho_{3,2} +\ s_{3,2,1,2} + \rho_{1,2} = 32.5$

Select sequence $\{o_{3,2}, o_{1,2}\}$ with the minimum makespan. Fixing sequence on machine m', generated in Step 3, implies scheduling $o_{2,1}$.

$$ST_{2,2}^{max} = 5 + 12 = 17$$

Because $m^*(t) = 32.5$ maximum wait time constraint between $o_{2,1}$ and $o_{2,2}$ is violated.

Applying the condition proposed in Amendment 4 would result in no ready time being generated for $o_{3,2}$ as sequencing on machine m' would stop after $o_{2,1}$ which is tied to $o_{2,2}$ by a maximum wait time constraint. In that case set J would consist only of operation $o_{1,2}$.

In the scenario where $o_{2,1}$ is not constrained by a maximum wait time constraint but operation $o_{3,1}$ is, both operations $o_{2,2}$ and $o_{3,2}$ would have estimated ready times generated. Due to clearing time condition, operation $o_{2,2}$ would not be considered in set J. However, not including $o_{2,2}$ in set J would not affect the solution feasibility as $ST_{2,2}^{max} = \infty$.

Amendment 5: Operation $o_{i,p}$ should be excluded from set J in Step 7 and Step 9 if $\omega_{i,p} < \infty$ and $A^{m_{i,p+1}}$ already contains a maximum wait constrained operation.

Proof 5a: Assume operation re-entry. Let $o_{i,p}$ and $o_{j,q}$ be in queue on machine m^*, and operation $o_{j,q+1}$ requires processing on machine m^* (operation re-entry). Additionally, assume $\omega_{i,p-1} < \infty$, and $\omega_{j,q} < \infty$. Thus, $o_{i,p}$ and $o_{j,q+1}$ are downstream maximum wait constrained with

$$ST_{i,p}^{max} = CT_{i,p-1} + \omega_{i,p-1} < \infty \tag{48}$$

and

$$ST_{j,q+1}^{max} = CT_{j,q} + \omega_{j,q} < \infty, \tag{49}$$

respectively.

From Amendment 3, operation $o_{j,q}$ is included in set J if the clear time constraint is satisfied,

$$CT_{j,q} + s_{j,q,i,p} \leq ST_{i,p}^{max}. \tag{50}$$

Equation (50) implies that there exists a non negative value ϵ^+, such that

$$CT_{j,q} + s_{j,q,i,p} + \epsilon^+ = ST_{i,p}^{max}. \tag{51}$$

Assuming such a value exists, $J = \{o_{i,p}, o_{j,q}\}$. Let operation $o_{j,q}$ be selected to schedule in Step 9 and $o_{j,q+1}$ is added A^{m^*}, in Step 11, with

$$ST_{j,q+1}^{max} = CT_{j,q} + \omega_{j,q}. \tag{52}$$

From Amendment 1, machine m^* is selected for scheduling. If

$$s_{j,q,j,q+1} + \rho_{j,q+1} + s_{j,q+1,i,p} \leq CT_{j,q} + s_{j,q,i,p} + \epsilon^+ \tag{53}$$

then operation $o_{j,q+1}$ can be sequenced before $o_{i,p}$ without violating the maximum wait time constraint of either operation. However, if the condition in (53) is not satisfied then operation $o_{i,p}$ must be sequenced before operation $o_{j,q+1}$ to avoid violating the maximum wait time constraint on $o_{i,p}$. However, sequencing $o_{i,p}$ before $o_{j,q+1}$, violates the maximum wait time constraint on $o_{j,q+1}$ for all cases where

$$s_{j,q,i,p} + \rho_{i,p} + s_{i,p,j,q+1} > \omega_{j,q}. \tag{54}$$

□

Proof 5b: Assume no operation re-entry**.** Let $o_{i,p}$ and $o_{j,q}$ be in queue on machine m^*. Additionally, assume $\omega_{i,p-1} < \infty$, and $\omega_{j,q} < \infty$. Thus, $o_{i,p}$ and $o_{j,q+1}$ are downstream maximum wait constrained with

$$ST_{i,p}^{max} = CT_{i,p-1} + \omega_{i,p-1} < \infty \tag{55}$$

and

$$ST_{j,q+1}^{max} = CT_{j,q} + \omega_{j,q} < \infty, \tag{56}$$

respectively. From Amendment 3, operation $o_{j,q}$ is included in set J if

$$CT_{j,q} + s_{j,q,i,p} \leq ST_{i,p}^{max}. \tag{57}$$

Equation (57) implies that there exists a non negative value ϵ^+, such that

$$CT_{j,q} + s_{j,q,i,p} + \epsilon^+ = ST_{i,p}^{max}. \tag{58}$$

Assuming such a value exists, $J = \{o_{i,p}, o_{j,q}\}$. Let operation $o_{j,q}$ be selected to schedule in Step 9 and $o_{j,q+1}$ is added $A^{m'}$, in Step 11, with

$$ST_{j,q+1}^{max} = CT_{j,q} + \omega_{j,q}. \tag{59}$$

From Amendment 1, machine m' is selected for scheduling. Let $o_{l,k}$ and $o_{j,q+1}$ be in queue on machine m'.

Additionally, assume $\omega_{l,k} < \infty$. Thus, $o_{l,k+1}$ is downstream maximum wait constrained with

$$ST_{l,k+1}^{max} = CT_{l,k} + \omega_{l,k} < \infty. \tag{60}$$

It can be shown that a set of conditions exist where by $o_{l,k}$ is selected to be sequenced on resulting in $o_{l,k+1}$ being added to A^{m^*}, with a maximum wait time constrained described in (60). From Amendment 1, machine m^* is selected for sequencing. If

$$\max\{CT_{j,q} + s_{j,q,l,k+1}, CT_{l,k}\} + \rho_{l,k+1} + s_{l,k+1,i,p} \leq CT_{j,q} + s_{j,q,i,p} + \epsilon^+ \tag{61}$$

then operation $o_{l,k+1}$ can be sequenced before $o_{i,p}$ without violating the maximum wait time constraint of either operation. However, if the condition in (61) is not satisfied, then operation $o_{i,p}$ must be sequenced before operation $o_{l,k+1}$ to avoid violating the maximum wait time constraint on $o_{i,p}$. However, sequencing $o_{i,p}$ before $o_{l,k+1}$, violates the maximum wait time constraint on $o_{l,k+1}$ for all cases where

$$CT_{j,q} + s_{j,q,i,p} + \rho_{i,p} + s_{i,p,l,k+1} > CT_{l,k} + \omega_{l,k}. \tag{62}$$

□

Note that optimal sequences generated for $LAO(\beta)$ or $LAJ(\beta)$ are feasible with respect to maximum wait time constraints.

5.1.5 OM-MW Dispatching Heuristic for Maximum Wait in Complex Job Shop

Step 1: Let A be all operations whose precedence constraints have been met. In the case of initialization $A = \{o_{i,1} | i \in N\}$.

Step 2: Select machine m^* which satisfies $min\{m(t)\}$. In the case of a tie then select the operation o_{i^+,p^+} from set A that has the earliest due date and let $m^* = m_{i^+,p^+}$.

Step 3: Use earliest due date, to simulate the sequencing of operations in queue on machines $m \neq m^*$. Once operation $o_{j,q}$ has been "scheduled" the simulated ready time of operation $o_{j,q+1}$ is recorded as $\hat{r}_{j,q+1} = \widehat{CT}_{j,q} + h_{j,q}$. Sequencing is stopped once an upstream maximum wait constrained operation has been sequenced (Amendment 4).

Step 4: Let A^{m^*} be all operations whose precedence constraints have been met and thus are in queue at machine m^*. Thus $A^{m^*} = \{o_{i,p} \in A | m_{i,p} = m^*\}$.

Step 5: Identify the operation $o_{i',p'}$ from A^{m^*} with the earliest due date.

a: Identify $o_{i'',p''}$ from A^{m^*} with maximum wait constraint, such that, $ST_{i'',p''}^{max} < \infty$.

Step 6: Let, $\mathbf{T} = m^*(t) + \rho_{i',p'} + s_{x,x,i',p'}$ where $s_{x,x,i',p'}$ is the setup time required to begin processing operation $o_{i',p'}$ on machine m^*.

Step 7: Let

$$J = \left\{o_{i,p} \in A^{m^*} \middle| r_{i,p} \leq \mathbf{T}, \widehat{CT}_{i,p} + s_{i,p,i'',p''} \leq ST_{i'',p''}^{max}, \left|A_{i,p+1}^{m}\right|_{max} = 0\right\} \cup o_{i'',p''}$$

a: Where $\widehat{CT}_{i,p} = m^*(t) + s_{x,x,i,p} + \rho_{i,p}$,

b: $|A^{m_{i,p+1}}|_{max}$ is the number of down stream maximum wait constrained operations currently in queue on machine $A^{m_{i,p+1}}$ (Amendment 5), and

c: $o_{i'',p''}$ is an operation on m^* with maximum wait constraint (Amendment 2).

Step 8: Let B^{m^*} be all operations not currently in A^{m^*} but require processing on m^*, and whose upstream operation is currently in queue on its respective machine.

Step 9: Include those operations from B^{m^*} in set J, whose simulated ready time on machine m^*, generated using EDD dispatching (modified by Amendment 4), $\hat{r}_{j,q}$, is less than $\mathbf{T}$, whose scheduling does not affect the downstream max wait operation $o_{i'',p''}$'s latest start time requirement (Amendment 3). The updated set can be expressed as $J = J \cup \left\{o_{j,q} \middle| \hat{r}_{j,q} \leq \mathbf{T}, m_{j,q} = m^*, \widehat{CT}_{j,q} + s_{j,q,i'',p''} \leq ST_{i'',p''}^{max}\right\}$.

a: Where $\widehat{CT}_{j,q} = m^*(t) + s_{x,x,j,q} + \rho_{j,q}$, and

b: $ST_{i'',p''}^{max}$ is the latest start time of operation $o_{i'',p''}$, $ST_{i'',p''}^{max} = CT_{i'',p''-1} + \omega_{i'',p''-1}$.

Step 10: Select operation o_{i^*,p^*} from J using some priority method and sequence on machine m^*.

a: LAJ: Select operation o_{i^*,p^*} from set J with earliest job due date

b: LAO: Select operation o_{i^*,p^*} from set J whose operation on machine m^* has the earliest local due date. Where, local due date = $d_{i,p} = d_i - 1.25\sum_{q=p+1}^{|O_i|} \rho_{i,q} + h_{i,q}$.

c: $LAJ(\beta)$: the β operations from set J with the earliest due dates. These operations are then sequenced optimally with respect to minimizing make span. The first operation in the optimal sequence, o_{i^*,p^*} is selected to be scheduled.

d: $LAJ(\beta)$: the β operations from set J with the earliest local due dates. These operations are then sequenced optimally with respect to minimizing make span. The first operation in the optimal sequence, o_{i^*,p^*} is selected to be scheduled.

Step 11: Sequence operation o_{i^*,p^*} on machine m^* and remove it from set A.

a: If operation o_{i^*,p^*} is not currently in queue for machine m^*, operation o_{i^*,p^*-1} must first be sequenced on its respective machine.

b: Sequence o_{i^*,p^*-1} on machine m_{i^*,p^*-1} by fixing the sequence generated to support Step 3.

Step 12: If o_{i^*,p^*+1} exists, add it to set A.

Step 13: If $A = \emptyset$ terminate, otherwise update the earliest operation start time r, and machine ready times, T^m, for all operations in set A.

Step 14: If $\omega_{i^*,p^*} < \infty$ then set $m^* = m_{i^*,p^*+1}$ and go to Step 3, else go to Step 2. (Amendment 1)

5.1.6 Limitations to OU-MW

The OU heuristic presented in 5.1.3 is capable of generating feasible solutions to the job shop scheduling problems, with sequence dependent setups, re-entry flow, and minimum wait constraints. The OU-MW heuristic presented in Section 5.1.4 extends the applicability of the dispatching heuristic to include problems with multiple maximum wait constraints per job, when the maximum wait constraints are not on successive operations. Feasible solutions can be generated for any demand set where for all operations, $\omega_{i,p} < \infty$ implies that both $\omega_{i,p-1} = \infty$ and $\omega_{i,p+1} = \infty$. Problems with more than one successive maximum wait operation can result in gridlock. Amendment 5 prevents any operation from being scheduled and a feasible schedule is not possible without re-scheduling an already sequenced operation. For problems where the number of successive maximum wait constrained operations on a given job is limited to one, the dispatching heuristic is always able to find a feasible solution.

The scheduling heuristic is capable of quickly generating good feasible solutions to the job shop scheduling problem with sequence dependent setups, re-entry flow, minimum wait constraints, a limited number of maximum wait constraints, and a due date based objective function. The C&L problem that motivates this research contains at most one maximum wait constraint per job.

5.2 Generating a Proxy Measure

By relating objective function performance to a basic performance measures, the complexity associated with evaluating the effect process planning decisions have on objective function performance is reduced. Ideally, a proxy measure should be easily calculated without generating a detailed schedule. Different data sets may have different attributes that drive objective function performance. For example, for one data set, machine m may be a bottleneck machine and thus an effective proxy measure for objective function performance would consider the amount of processing time on machine m, ρ_m^{total}. Another demand set may have no jobs that require processing on machine m and thus ρ_m^{total} provides no insight into objective function performance. Because each data set is unique, a robust procedure for generating an effective proxy measure is key to the general implementation of a proxy measure based approach. This section describes two processes for generating a proxy measure, a modified stepwise regression and a back propagating neural network. The following notation will be used throughout this section.

Indices and Sets

$\boldsymbol{S}^T$	Set of schedules used for training	
$P_{k,i}$	Process planning decision for job i in schedule k	
$\boldsymbol{P}_k$	Set of process planning decisions for jobs in schedule k	
$\phi_{k,z}$	Descriptive statistic z for schedule k	
$\boldsymbol{\Phi}_\mathrm{k}$	Set of descriptive statistics for schedule k	
$f(\cdot)$	Objective function	
$\hat{f}(\boldsymbol{\Phi}_\mathrm{k})$	Proxy measure	
z, y	Descriptive statistics indices	$z, y = 1, \dots, \lvert\boldsymbol{\Phi}_\mathrm{k}\rvert$
k, g	Schedule indices	$k, g = 1, \dots, \lvert\boldsymbol{S}^T\rvert$

Descriptive process planning statistics are values that can be calculated without generating a detailed schedule. For schedule k, a set of descriptive statistics, $\phi_{k,z}$, are generated. The set of all descriptive statistics for a given schedule is denoted as $\boldsymbol{\Phi}_k$, where $\boldsymbol{\Phi}_k = \{\phi_{k,1}, \dots, \phi_{k,|\boldsymbol{\Phi}_k|}\}$. These descriptive statistics are used as the independent variable in the proxy measure model. The descriptive statistics used in this dissertation are, total processing time on machine m, ρ_m^{total}, total number of operations on machine m, o_m^{total}, as well as standard deviation of these terms, σ_ρ and σ_o. Any meaningful measure that can be calculated without generating a detailed schedule may be used as descriptive statistics in a proxy measure model.

While a deterministic relationship between process planning decisions and objective function values exists, the complexity of this interaction is intractable from a computational modeling standpoint. To generate a proxy measure, two approaches are explored: a modified step-wise regression based approach and a back-propagating neural network. The results of the two different methods for generating a proxy measure are compared in Section 6. Additionally, Section 6.2.5 provides a comparison of the proxy measures to a random walk for guiding process planning decisions.

5.2.1 Training Data for Proxy Measure

Training data consists of a set of observed independent and dependent variables that will be used as input to build the proxy model. For a given demand set, $|\boldsymbol{S}^T|$ detailed schedules are generated for training, each with a unique set of process planning decisions $\boldsymbol{P}_k$. This results in a matrix of descriptive process planning statistics, or independent variables $\underline{\boldsymbol{\Phi}}$, and a vector of observed objective function values $\underline{f}$.

5.2.1.1 Input Data Pre-Processing

Only independent variables that contain information are usfeful in training the proxy measure. For this reason, a data processing method is used to remove any redundant descriptive statistics that are the same for each observation. For example, if the number of operations on machine 1 is the same as the number of operations on machine 6 for all observations, then the number of operations on machine 6 is removed as a descriptive statistic (independent variable). If the descriptive statistic z for schedule k is equivalent to the descriptive statistic y for schedule k across all schedules, such that,

$$\phi_{k,z} = \phi_{k,y} \;\; \forall k \in \boldsymbol{S}^T, \tag{63}$$

then $\phi_{k,y}$ is removed for all k in $\boldsymbol{S}^T$. Additionally, if a descriptive statistic has the same value for each observation, then the descriptive statistics it is removed, as it provides no additional information regarding the effect process planning decisions have on the objective function. If the descriptive statistic z has the same value in schedule k as it does for all other schedules in $\boldsymbol{S}^T$,

$$\phi_{k,z} = \phi_{g,z} \;\; \forall g \in \boldsymbol{S}^T. \tag{64}$$

Then the statistic z is removed from all schedules $\boldsymbol{S}^T$. The scale of descriptive process planning statistics can range in value. Thus, each value is normalized to between 0 and 1 based on the minimum and maximum observed value from the training set. Equation (65) describes the scaling where $\phi_{g,z}$ is the maximum observed value for process planning statistic z in the training set and $\phi_{u,z}$ is the minimum observed value, such that,

$$\phi'_{k,z} = \frac{\phi_{g,z} - \phi_{k,z}}{\phi_{g,z} - \phi_{u,z}}. \tag{65}$$

Following this scaling step, the new set of descriptive process planning statistics, $\underline{\boldsymbol{\Phi}'}$, and observed objective function values, $\underline{f}$, are used to generate the proxy measure $\hat{f}(\boldsymbol{\Phi}')$. The resulting proxy measure can be used to estimate the objective function for a given set of process planning decisions $\boldsymbol{P}$. The proxy measure can be used in conjunction with local search methods to improve overall process planning and scheduling performance.

5.2.2 Stepwise Regression Modeling

A modified stepwise regression approach is used to generate a multidimensional polynomial model to estimate the interaction between the independent variables (descriptive process planning statistics) and the dependent variable (objective function) with the objective of maximizing the adjusted-R^2, a metric of fit quality.

In addition to the pre-processing steps described in Section 5.2.1.1, interaction variables are calculated for the purpose of generating a regression based proxy measure. Interaction variables are calculated by multiplying single measure descriptive statistics together, such as $\rho_1^{total} * o_1^{total}$ or $\sigma_\rho * \sigma_\rho$. While the number of interaction terms is an input for the model, a 2-degree model is assumed sufficient for proxy measure fitting. For the purposes of regression modeling, these interaction terms are considered independent variables. This increases the dimensionality of the original input variable vector, $\boldsymbol{\Phi}_k$, from $|\boldsymbol{\Phi}_k|$ to $|\boldsymbol{\Phi}_k|^2$.

The dependent variable of the regression model is an estimate of the objective function. For the C&L problem the objective function is the sum of tardiness denoted by $f(\boldsymbol{P}_k)$, where $\boldsymbol{P}_k$ is the set of process planning decisions that result in a given schedule k. There are a number of feasible solutions to the scheduling problem for $\boldsymbol{P}_k$. Assuming a fixed deterministic heuristic for scheduling, there is a single objective value $f(\boldsymbol{P}_k)$ for a given $\boldsymbol{P}_k$.

For a given demand set, an initial set of schedules, $\boldsymbol{S}^T$, are generated which include process planning decisions for each job, $P_{k,i}$. Detailed schedules are generated using the OU-MW heuristic, described in Section 5.1.5. The resulting set of descriptive statistics, $\underline{\boldsymbol{\Phi}}$, and objective function values, $f(\underline{\boldsymbol{P}})$, are used to develop the regression based proxy measure model. A modified step-wise regression is used to determine which terms, $\phi_z \in \boldsymbol{\Phi}$, to include in the model. The modified step-wise regression iteratively adds and removes terms to the model with the goal of increasing the adjusted R^2 statistic, denoted as $\bar{R}^2$. The $\bar{R}^2$ statistic describes the strength of the model fit while taking into account the number of parameters included to prevent over fitting. The calculation for the R^2 statistic described in Equation (66), where $\hat{f}(\boldsymbol{P}_k)$ is the model estimate for inputs $\boldsymbol{P}_k$, $f(\boldsymbol{P}_k)$ is the actual observed value for $\boldsymbol{P}_k$, and $\bar{f}(\boldsymbol{P}_k)$ is the mean of the observed values, such that,

$$R^2 = 1 - \frac{\sum_k \left(f(P_k) - \hat{f}(P_k)\right)^2}{\sum_k \left(f(P_k) - \bar{f}(P_k)\right)^2} \tag{66}$$

Equation (67), shows the calculation for $\bar{R}^2$ which accounts for the number of observations in the training set, N^T, and the number of variables used in used in the model, p, such that,

$$\bar{R}^2 = 1 - \frac{(1 - R^2)(N - 1)}{N^T - p - 1} \tag{67}$$

The following steps summarize the modified stepwise regression:

Step 1: Let $\boldsymbol{\Phi}^*$ be the set of all parameters, ϕ, used in the proxy measure model

Step 2: Set $\boldsymbol{\Phi}^* = \emptyset$, thus $\bar{R}^2_{\boldsymbol{\Phi}^*} = 0$.

Step 3: Evaluate $\bar{R}^2_{\boldsymbol{\Phi}^* \cup \phi_z}$ for all $\phi_z \notin \boldsymbol{\Phi}^*$

Step 4: If $\bar{R}^2_{\boldsymbol{\Phi}^* \cup \phi_z} > \bar{R}^2_{\boldsymbol{\Phi}^*}$, let ϕ_z^* maximize $\bar{R}^2_{\boldsymbol{\Phi}^* \cup \phi_z}$

Step 5: Let $\boldsymbol{\Phi}^{*\prime} = \boldsymbol{\Phi}^* \cup \phi_z^*$

Step 6: Evaluate $\bar{R}^2_{\boldsymbol{\Phi}^* / \phi_z}$, for all $\phi_z \in \boldsymbol{\Phi}^{*\prime}$

Step 7: If $\bar{R}^2_{\mathbf{\Phi}^{*\prime}/\phi_z} > \bar{R}^2_{\mathbf{\Phi}^*}$, let ϕ_z' maximize $\bar{R}^2_{\mathbf{\Phi}^{*\prime}/\phi_z}$

Step 8: Let $\mathbf{\Phi}^{*\prime} = \mathbf{\Phi}^{*\prime}/\phi_z'$

Step 9: If $\mathbf{\Phi}^{*\prime} = \mathbf{\Phi}^*$ terminate, otherwise go to 3

Because Steps 4 and 7 select the term that maximizes $\bar{R}^2$, this modified step-wise regression can be considered a greedy heuristic and does not guarantee optimality in the case of local maximum. This greedy approach for parameter fitting is found to generate good solutions more quickly than a more complex search approach. The use of an adjusted R^2 measure is motivated by the goal of preventing over fitting and limiting the number of terms allowed in the model. From computational experience, a limit of ten terms has provided better solutions to the proxy based local search problem when compared to regression based proxy models that allow more than ten terms. For the purposes of this study, the proxy model is limited to ten terms. An over fit model often does not accurately predict future observations. Using $\bar{R}^2$ instead of R^2, along with limiting the number of terms allowed in the model, is designed to protect against over fitting the proxy model.

5.2.3 Neural Network Modeling

A neural network is trained using a back propagation algorithm to estimate the interaction between the input variables (descriptive process planning statistics) and the dependent variable (objective function values). Neural networks consist of an input layer, an output layer, and a series of hidden layers or processing units between the input and output layers. The nodes at each layer are connected by weights. Figure 5.2 provides an illustration of a neural network architecture.

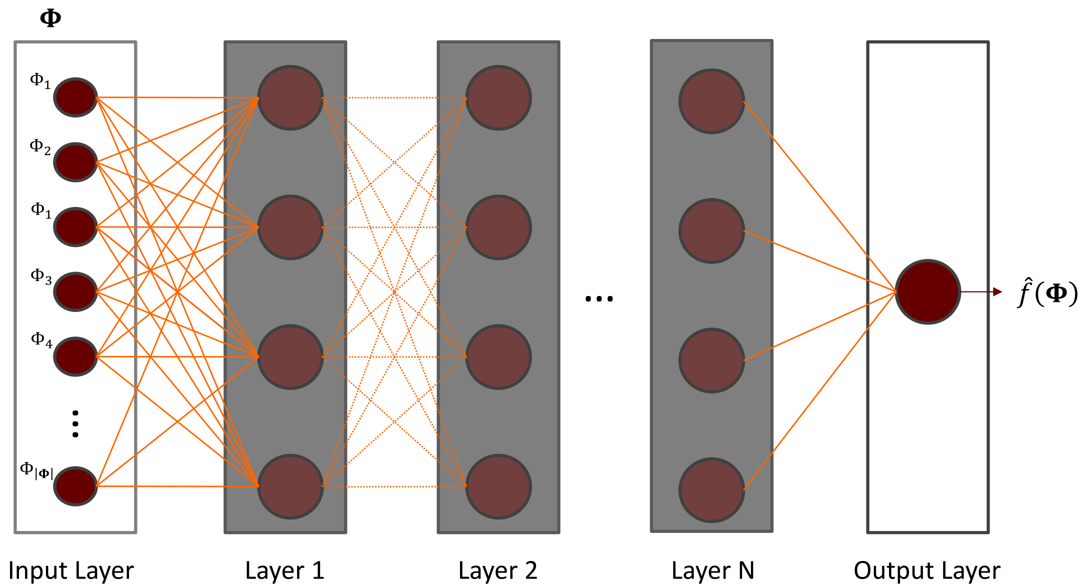


Figure 5.2 Neural Network Architecture

Neural networks have increased in popularity due to their ability to detect and model complex patterns not effectively detected by more basic linear or even polynomial models. Neural network models also present an advantage of reduced dimensionality which can lead to better models with less training data than polynomial regression models (Bishop, 2006). The ability of neural networks to detect complicated patterns and learn from a training set has made it popular in world of machine learning, image detection, as well as scheduling and dispatching problems (El-Bouri and Shah, 2006). Artificial neural networks have been applied in a variety of ways to solving scheduling problems (Akyol and Bayhan, 2007). Neural networks have also been used for process planning decision-making. Xu *et al.* (2011) provide a review of computer aided process planning, along with discussion of the application of neural networks. Similar to much of the computer aided production planning research, the work focuses predominantly on a single part or focused on establishing a sequence of required machining operations (Ding *et al.*, 2005, Deb *et al.*, 2006). In this research, in contrast to previous applications of neural networks to scheduling or IPPS, a neural network is trained to estimate the effect process planning changes have on the sum of tardiness in a job shop environment with a variety of complicating constraints. The network is then used to guide a local search as a part of an overall heuristic to generate a good solution to a complex process planning and scheduling problem.

The training data for the neural network consists of observed inputs (descriptive process planning statistics) and outputs (observed objective function values). Neural network training inherently considers multidimensional interactions and thus interaction terms do not need to be explicitly identified prior to training the neural network. Using a back propagation algorithm, the weights connecting the input layers, processing units in the hidden layers, and the output layer are modified to minimize mean squared error. The resulting network can be used to estimate output given a set of input variables. The number of hidden layers in the neural network can influence the quality of the network. In Section 6.2.4, the ideal number of layers for the proxy based local search problem is investigated.

The functionality to train and evaluate a neural network is included in Matlab (Mathworks, 2015). For this dissertation, the Matlab toolbox is accessed through a Java to Matlab interface. Improvements to computational processing time may be possible with an improved API or by leveraging a Java library for training and evaluating the neural network.

5.3 Local Search Methods

Process planning decisions affect the scheduling objective function. Therefore, it is advantageous to consider modifying process planning decisions with the goal of improving the objective function (such as reducing tardiness). Figure 5.3 illustrates the structure of process planning decisions for a representative material WF-A.

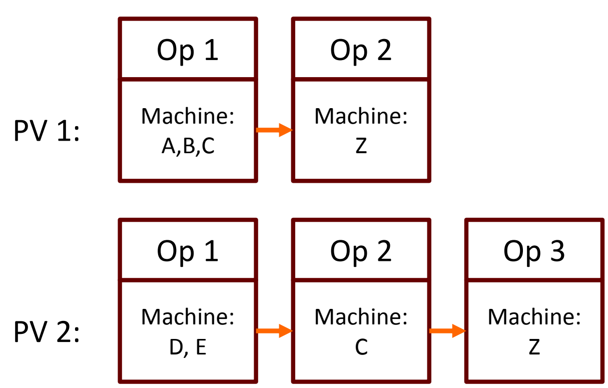


Figure 5.3 Processing plans for WF-A

WF-A can be produced through two different production versions, PV 1, which consists of two operations, and PV 2, which consists of three operations. Operation 1 in PV 1 can be processed on three different machines, while operation 1 of PV 2 can be processed on two different machines. Therefore, this material has five different processing plans.

The two local search methods presented here provide a structured approach for exploring the alternative processing plans for a given job. The first search method is based on machine swaps and, the second is based on production version swaps. The effect on the objective function of these local search moves can be quickly estimated using the proxy measure.

5.3.1 Machine Swap Search

For a given production version, the machine swap (MS) search considers all of the alternative machine options within that production version. Consider material WF-A (Figure 5.3) for which production version PV1 is selected and operation one is processed on machine A. The MS swap search would consider all alternative processing plans within PV 1, that includes performing operation Op1 on machine B or machine C for this example.

5.3.2 Production Version Search

For a given job, the production version (PV) search considers all production versions not currently active for this job. In the example of material WF-A for which PV 1 was initially selected, the PV search would switch the production version to PV 2 and identify the total number of processing plans (machine selections) within that production version. The PV search would then randomly make machine selection decisions resulting in a new process planning decision for WF-A.

5.4 Proxy Based Local Search for Integrated Process Planning and Scheduling

For a given demand set, detailed schedules are generated to populate the training set, $\boldsymbol{S}^T$. Detailed schedule, S_k, is generated using the modified Uzsoy heuristic, described in Section 5.1, with process planning decisions for each job, $P_{k,i}$, made *a priori*. The objective function value for each of these schedules $\underline{f}$ along with the descriptive process planning statistics $\underline{\boldsymbol{\Phi}}$ are used to generate a proxy measure $\hat{f}(\cdot)$.

Alternative process planning decisions for individual jobs are explored using a local search approach described in Section 5.3. Each modification to a processing plan, $\boldsymbol{P}_k \rightarrow \boldsymbol{P}'_k$, will result in a new schedule with a new objective function, $f(\boldsymbol{P}_k) \rightarrow f(\boldsymbol{P}'_k)$. The decision to keep a process planning change is based on the effect the change has on the objective function, estimated by the proxy measure $\hat{f}(\cdot)$. Because proxy measure evaluation is quick, multiple process plan changes can be considered before a new detailed schedule is generated. The steps for the proxy based local (PBLS) search are as follows:

- Step 1: Let N be the set of jobs in some demand set, and $\boldsymbol{P}_k^o$ be the set of original process planning decisions for each job $i \in N$, for some schedule S_k.
- Step 2: Let $\boldsymbol{J} \subseteq N$, be the set of jobs for which process planning changes will be considered. The size and the sequence of the search set $\boldsymbol{J}$ can be adjusted.
- Step 3: Select job i from the search set $\boldsymbol{J}$ whose original process plan is $P_{k,i}$.
- Step 4: Identify each possible process plan $P'_{k,i}$ that can be reached by applying Machine Swap Search to $P_{k,i}$.
- Step 5: Evaluate the proxy measure values of the current process plan set $\boldsymbol{P}_k$, as $\hat{f}(\boldsymbol{\Phi}_k)$.
- Step 6: Evaluate the proxy measure for each proposed process plan change for job , $P'_{k,i}$.
- Step 7: Let ${P'_{k,i}}^*$ be the process plan that results in the best proxy value $\hat{f}\left({\boldsymbol{\Phi}'_k}^*\right)$
- Step 8: If $\hat{f}\left({\boldsymbol{\Phi}'_k}^*\right) < \hat{f}(\boldsymbol{\Phi}_k)$, in the case of a minimization problem, accept process plan change ${P'_{k,i}}^*$, updating the overall process plan for the schedule $\boldsymbol{P}'_k$. If the proxy measure does not estimate an improvement no process plan changes are made for job i. Local search moves are censored, or ignored if $\hat{f}\left({\boldsymbol{\Phi}'_k}^*\right) < 0$.
- Step 9: Remove job i from the search set $\boldsymbol{J}$. If $|\boldsymbol{J}| > 0$, then return to Step 3, otherwise go to Step 10.
- Step 10: $|\boldsymbol{J}| = 0$, then generate a new detailed schedule with process plan $\boldsymbol{P}'_k$ and evaluate the objective function $f(\boldsymbol{P}'_k)$ compared to the original objective function $f(\boldsymbol{P}_k^o)$.
- Step 11: If $f(\boldsymbol{P}'_k) < f(\boldsymbol{P}_k^o)$, in the case of a minimization problem, accept the new schedule.
- Step 12: Repeat Steps 2-11, using the superior schedule S_k or S'_k, applying Production Version Search in Step 4.

All schedules in the training set $\boldsymbol{S}^T$ are used to train the proxy measure. After the proxy measure has been created, the local search approach can be applied to any schedule, S_k, from the training set. The outcome of applying PBLS to S_k is independent of applying the search being applied to another schedule, S_g. This independence allows for the search to be applied simultaneously to multiple schedules in $\boldsymbol{S}^T$. This ability to search in parallel improves the speed of this approach.

5.5 Summary of Solution Approach

The proxy based local search (PBLS) is a multi-phase approach to generate process planning and scheduling decisions for the C&L problem. First, for a given demand set, a process planning decision $P_{k,i}$ is made for each job i in the demand set. A scheduling heuristic is used to generate a detailed schedule S_k where process planning decisions $\boldsymbol{P}_k$, are inputs. This process is repeated to generate a training set, $\boldsymbol{S}^T$. The resulting objective functions $\underline{f}$ and process planning statistics, $\underline{\boldsymbol{\Phi}}$, are used to generate a proxy measure $\hat{f}(\cdot)$. Local searches are conducted to modify process plan decisions in a given schedule, S_k. The effect of each process plan change on the objective function is evaluated using the proxy measure. New process planning decisions are accepted if they result in an improvement in the proxy measure. The local search is performed on some number, N^S, of the initial $\boldsymbol{S}^T$ schedules. Following the local search, the new set of process planning decisions, $\boldsymbol{P}'_k$, are used to create a detailed schedule. The new schedule is retained if the realized objective function from the detailed schedule improves on the incumbent objective function for schedule S_k. This flow is graphically depicted in Figure 5.4 and Figure 5.5. In Figure 5.4, the steps required to generate a proxy measure are illustrated. Figure 5.5 shows the local search and rescheduling phases of PBLS.

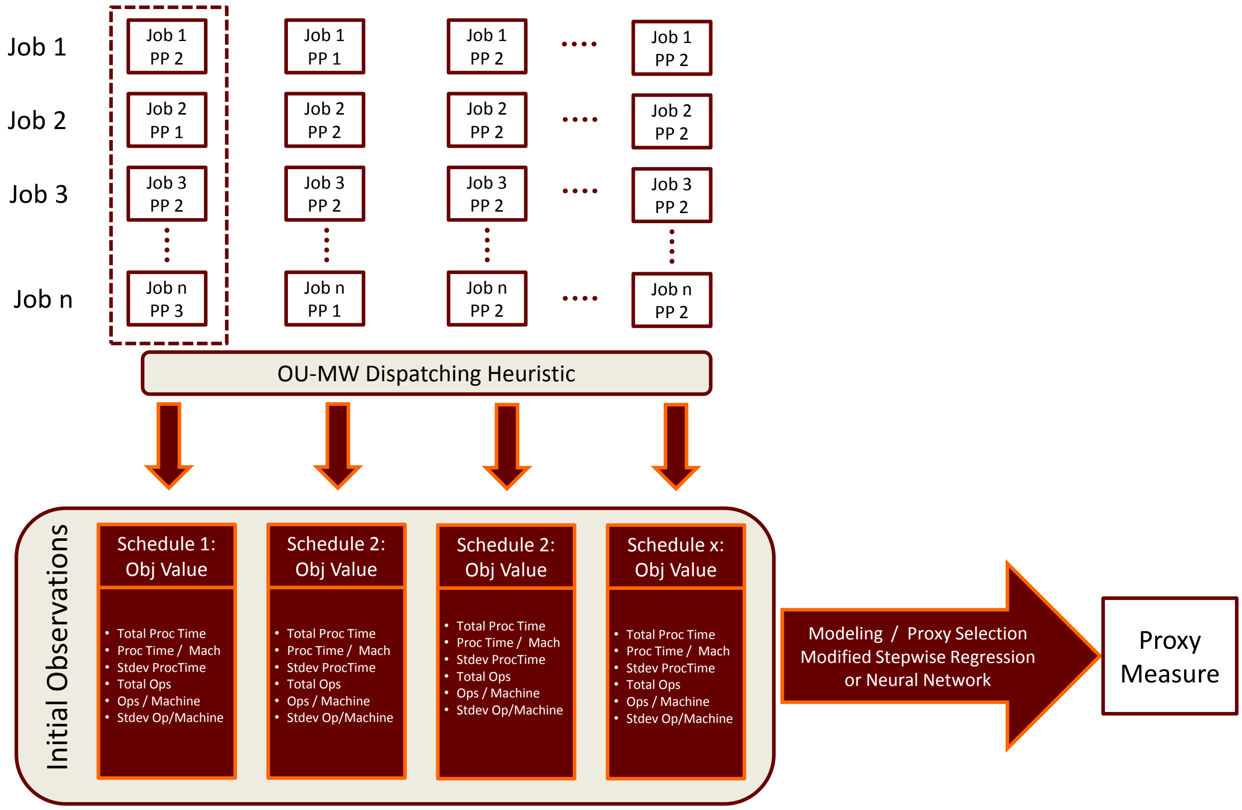


Figure 5.4 Concept Flow Chart – Proxy Measure Generation

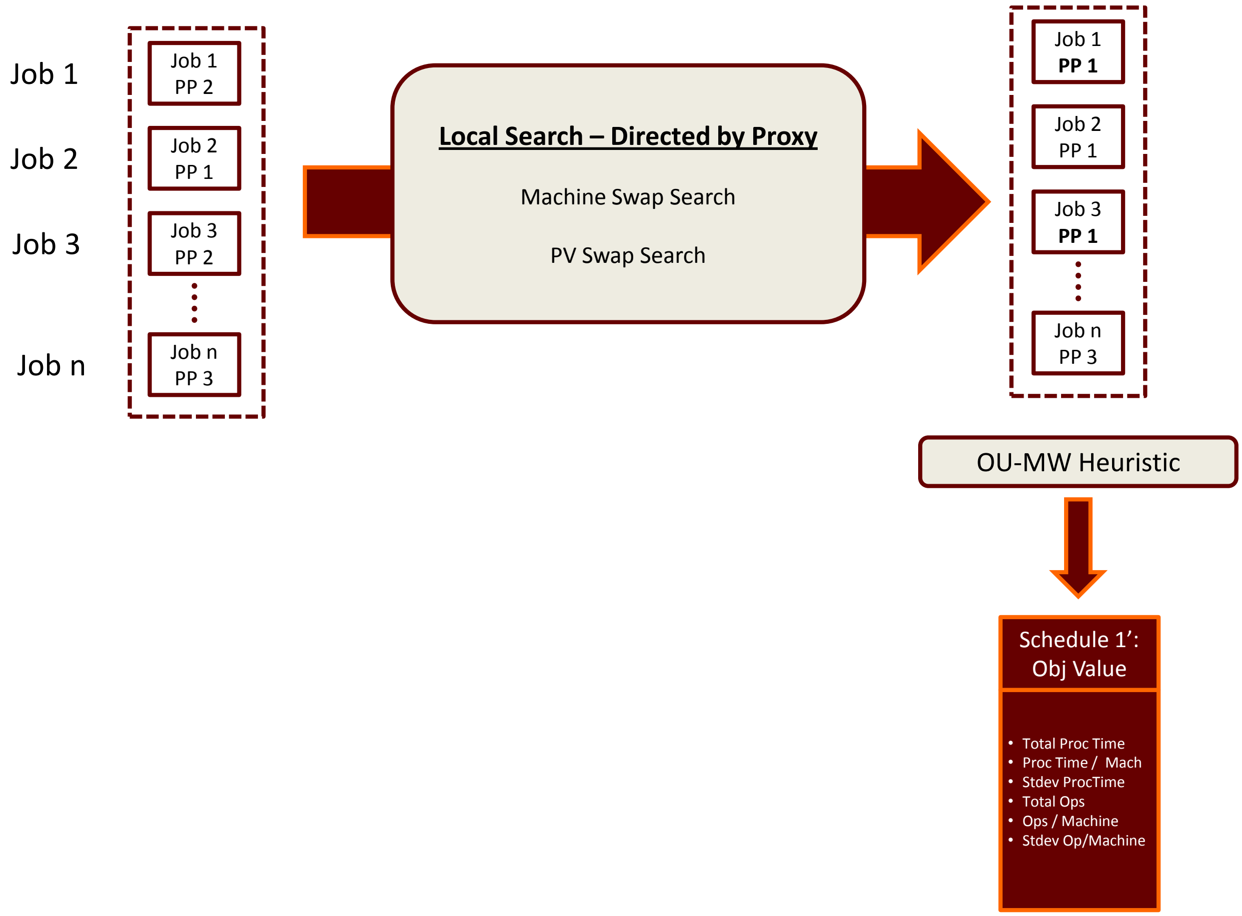


Figure 5.5 Concept Flow Chart – Proxy Based Local Search

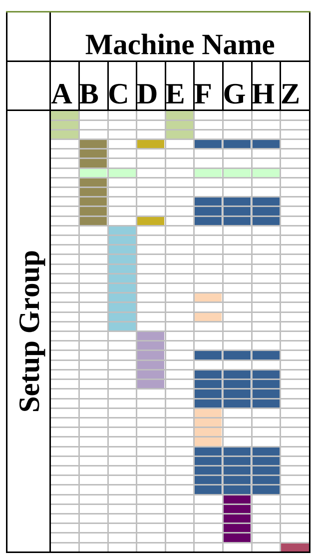
6. Applying Proxy Based Local Search

To evaluate the proxy based local search (PBLS) described in the previous section, four industry representative demand sets are evaluated that include nine representative machines with varying capabilities. PBLS, using both regression and neural network based proxy measures, is applied to each demand set. The results of PBLS are compared to random walk based local search. Finally, PBLS is applied to a small data set for which the optimal solution can be found.

6.1 Machines, Setups, and Demand Sets

Each demand set is scheduled using the same representative factory resources of nine machines. Each machine has its own unique characteristics and processing capabilities. The scrap rate and feed rate on a given machine is operation dependent. Due to the nature of production, sequence dependent setups exist on all machines. The setup time required between processing two operations depends on the setup group of each operation. Table 6.1 identifies the setup groups each machine is capable of processing. The capability for a machine to process a particular setup group does not directly imply the ability to process any operation of that setup group. For example, a given operation may only be approved for processing on machine A, even though other operations in the same setup group are approved for processing on machines A and E.

Table 6.1 Setup Groups by Machine



6.1.1 Parallel Machine Capabilities

Large-scale production facilities often have more than one machine station capable of preforming a given operation. Table 6.2 presents a representative operation (Op 1) that can be processed on four different machines. Scrap and feed rates differ across machines, resulting in machine dependent processing times. In the case of jobs with more than one operation, the machine selection decisions for one operation influence the processing time of the other operations for that job. Scheduling production for a given demand set requires selecting an allowed machine as part of the process planning.

Table 6.2 Process Time Comparison Across Four Different Process Plans

	Op 1 (Mach B)	**Op 1 (Mach F)**	**Op 1 (Mach G)**	**Op 1 (Mach H)**
Final Req Length (ft)	10,000	10,000	10,000	10,000
Scrap Rate	5%	5%	5%	5%
Initial Req Length (ft)	10,500	10,500	10,500	10,500
Feed Rate (ft/hr)	7,200	6,000	9,000	7,800
Process Time (hr)	1.46	1.75	1.17	1.35

6.1.2 Advanced Machine Stations

The representative machine set used in this research includes several machines that can accomplish in a single operation what otherwise would take two or more operations. Table 6.3 presents two process plans for material WF-B. Process plan 1 leverages less advanced machining stations and requires 2 operations and 3.77 hours to produce 10,000 feet of WF-B. Process plan 2 leverages a more advanced machine H. This processing plan requires less initial raw film and takes only 1.47 hours to produce 10,000 feet of WF-B. An integrated process planning and scheduling solution to the C&L problem must consider the effect of these types of process plan decisions on the desired objective function.

Table 6.3 Influence of Advanced Machine Stations on Process Plan

WF-B Process Plan 1			
	Final	**Op 2 (Mach C)**	**Op 1 (Mach D)**
Final Req Length (ft)	10,000	10,000	11,000
Scrap Rate	X	10%	10%
Initial Req Length (ft)	10,000	11,000	12,100
Feed Rate (ft/hr)	X	7,200	5,400
Process Time (hr)	3.77	1.53	2.24

WF-B Process Plan 2		
	Final	**Op 1 (Mach H)**
Final Req Length (ft)	10,000	10,000
Scrap Rate	X	15%
Initial Req Length (ft)	10,000	11,500
Feed Rate (ft/hr)	X	7,800
Process Time (hr)	1.47	1.47

6.1.3 Setup Groups

Setup groups are both operation and machine dependent. The number of different setup groups that can be processed on the nine industry representative machines is derived from the process plans for each of the four demand sets. A summary of this information is displayed in Table 6.4.

When examining the materials in the demand sets and all possible processing plans, approximately 50 unique setup groups emerge across all the machines. Setup times range from 0.5 hours to as much as 8 hours.

Table 6.4 Unique Setup Group by Machine

Machine	Number Unique Setup Groups
A	3
B	11
C	12
D	8
E	3
F	21
G	21
H	15
Z	1

6.1.4 Demand Sets

Four representative demand sets are considered to evaluate the performance of the proxy based local search (PBLS). A given demand set contains a list of jobs which specify a material type, the job length, a due date, and the potential process plans. The specific operations required to complete the job depend on the processing plan selected for that job as apart of the PBLS procedure.

Details such as the number of jobs, number of unique material types, information about requirement length, and due dates for each of the four demand sets is summarized in Table 6.5. Demand Sets 1-3 include approximately one week of production volume. The due dates for jobs in Demand Sets 1-3 were generated from a uniform distribution between 6 hours and 144 hours (6 days) to simulate a manufacturing environment with a 24 hour per day and 6 days per week production cycle. Demand Set 4 includes approximately four weeks of production volume and the due dates are generated accordingly.

Table 6.5 Summary of Demand Sets

	Demand Set 1	Demand Set 2	Demand Set 3	Demand Set 4
Number of Jobs	110	165	98	200
Unique Materials	102	151	94	102
Total Req. Length (ft)	2,210,418	3,947,696	2,345,615	10,777,200
Avg. Req. Length (ft)	20,094	23,925	23,934	53,886
Min Req. Length (ft)	625	625	1,465	4,730
Max. Req. Length (ft)	174,991	136,344	184,299	98,744
Avg. Due Date (hours)	87.3	87.6	86.5	392
Standard Dev. Due Date (hours)	46	48	50	202
Avg. Process Plans per job	3.3	3.2	3.9	3.4

6.1.5 Process Time Calculation

Process time is both operation and machine dependent. Additionally, the scrap and feed rate of upstream machines influence the process time of a given operation. The operations to be performed and their processing times depend on the process plan selected for each job. The final product length requirement is specified for each job. Processing times for each operation are based on a machine's feed and scrap rate as well as the scrap rate for all upstream machine selections for that process plan. Table 6.6 summarizes the requirement lengths and processing times for two process plans of a sample job WF-C with a final requirement of 10,000 ft of film.

The initial requirement length for an operation serves as the final requirement for its predecessor operation. In Process Plan 1, WF-C is processed on machines H, F and B. Each operation has an associated scrap rate. As shown, a total of 10,000 ft is required for WF-C, and 11,000 ft is needed for Op 1 on machine B to account for the scrap rate in each operation. The processing time for a given operation is calculated based on the initial requirement length and the machines feed rate for that operation. In Process Plan 2, WF-C is processed on machines H, E, and B. Machine F, used in Process Plan 1, has a scrap rate of 3% and a feed rate of 7,210 feet/hour. While, machine E, used in Process Plan 2, has a scrap rate of 6% and a feed rate of 9,300 feet/hour. As a result, the raw material requirement and processing times differ for operations 1 and 2.

Table 6.6 Requirement Length and Processing Time Calculations

WF-C Process Plan 1				
	Final	**Op 3 (Mach H)**	**Op 2 (Mach F)**	**Op 1 (Mach B)**
Final Req Length (ft)	10,000	10,000	10,500	10,815
Scrap Rate	-	5%	3%	1.7%
Initial Req Length (ft)	10,000	10,500	10,815	11,000
Feed Rate (ft/hr)	X	9,500	7,210	10,000
Process Time (hr)	3.7	1.1	1.5	1.1

WF-C Process Plan 2				
	Final	**Op 3 (Mach H)**	**Op 2 (Mach E)**	**Op1 (Mach B)**
Final Req Length (ft)	10,000	10,000	10,500	11,130
Scrap Rate	-	5%	6%	1.7%
Initial Req Length (ft)	10,000	10,500	11,130	11,319
Feed Rate (ft/hr)	-	9,500	9,300	10,000
Process Time (hr)	3.4	1.1	1.2	1.1

6.1.6 Resource Profile by Demand Set

The demand for each machine depends on the specific process plan selected for the jobs in a demand set. However, the possible processing plans for all jobs in a demand set provide some insight into the demand set and resource profile. Table 6.7 summarizes the number of unique setup groups per machine for each of the four demand sets. As shown, there is a similar profile across the demand sets.

Table 6.7 Unique Setup Group by Machine

	Unique Setup Groups by Machine			
Machine	**Demand Set 1**	**Demand Set 2**	**Demand Set 3**	**Demand Set 4**
A	2	3	3	2
B	5	8	6	5
C	6	7	6	8
D	5	4	2	7
E	3	3	3	3
F	13	14	11	10
G	14	14	13	10
H	10	11	9	7
Z	1	1	1	1

6.2 Proxy Based Local Search Parameters

A variety of parameters are considered when applying a proxy based local search (PBLS) to an integrated process planning and scheduling problem. Key parameters are described in Table 6.8. In the following sections, the effects of these parameters on the sum of total tardiness is investigated.

Table 6.8 Key Parameters for Proxy Based Local Search

Parameter Name	**Definition**	**Values Considered**
Proxy Type	The method for generating the proxy measure.	Regression, Neural Network, Random Walk
Size of training set, N^T	The number of schedules used in training the proxy measure.	100, 196, 324, 400
Search size, N^S	The number of schedules to perform the local search on.	100
Search approach	The sequence in which jobs are considered during the local search.	Sequential or Random
Search depth, S^D	The percentage of jobs from the demand set to include during a local search.	25%, 50%, 75%, 100%
Network Layers	The number of layers used in the neural network. Relevant only when Proxy Type is neural Network.	1, 2, 3, 4, 5, 10

The effectiveness of PBLS is evaluated based on its ability to return a solution with a favorable objective function value, in this case, low total tardiness. Given the desire to apply this local search heuristic to large data sets in industrial settings, the ability for the heuristic to return timely solutions is also a key component of overall effectiveness. Thus, the effects of parameters on both objective function values and processing time are evaluated.

A robust design is used for evaluating the effect parameters have on performance across the four demand sets. Table 6.9 and Table 6.10 detail the parameter combinations explored for PBLS with a regression based proxy measure (PBLS-R) and PBLS with a neural network based proxy measure (PBLS-NN), along with the average objective function values across a number of runs.

Because the proxy measure generation is not deterministic, multiple runs are conducted as a part of the analysis and the average and variance of key metrics are recorded. The values presented in Table 6.9 represent the average objective function value from applying PBLS-R 100 times. Table 6.10 presents the average objective function value from applying PBLS-NN 50 times. The number of layers used in training the neural network is represented in parentheses in front of the objective function value in Table 6.10. The averages and standard deviations of the objective function values and other scheduling statistics, such as maximum tardiness, makespan and percentage of tardy job,s for the trial runs are provided in Appendix A.3.

While the average value across multiple runs provides a good point measure, a statistical confidence interval provides a more meaningful description of the influence a set of parameters have on heuristic performance. A 95% confidence interval can be calculated using the confidence interval equations in (68) and (69), where $\bar{X}$ and σ are the average and standard deviation, respectively, of the n trials, and 1.96 is the 95% confidence interval. The following confidence intervals are used in several cases to identify the significance of parameter settings:

$$UB = \bar{X} + \frac{1.96\sigma}{\sqrt{n}} \tag{68}$$

$$LB = \bar{X} - \frac{1.96\sigma}{\sqrt{n}}. \tag{69}$$

Table 6.9 PBLS-R Average Objective Function Value*

	N^T	Search Sequence and Depth				
		Sequential 100%	**Random 100%**	**Random 75%**	**Random 50%**	**Random 25%**
Demand Set 1	**100**		1,758	1,863	2,073	2,403
	196		1,667	1,815	1,954	2,352
	324	1,614	1,563	1,697	1,959	2,302
	400		**1,557**	1,699	1,912	2,280
Demand Set 2	**100**		5,677	5,573	5,610	5,960
	196		5,499	5,512	5,591	5,830
	324	5,398	5,390	5,406	5,432	5,807
	400		5,438	**5,381**	5,411	5,733
Demand Set 3	**100**		1,640	1,623	1,752	2,056
	196		1,601	1,597	1,676	2,034
	324	1614	1,601	**1,541**	1,647	1,979
	400		1,601	1,536	1,642	1,964
Demand Set 4	**100**		455	430	416	549
	196		439	395	398	520
	324	462	418	399	397	504
	400		403	391	**382**	507

*Average from 100 trials

Table 6.10 PBLS-NN: Average Objective Function Value*

	N^T	Search Sequence and Depth Random 100%	 Random 75%	 Random 50%	 Random 25%
Demand Set 1	**100**		(10) 2,304		
	196		(10) 2,245		
	324		(10) 2,226		
	400	(1) **1,941** (2) 2,037 (3) 1,956 (4) 2,019 (5) 2,055 (10) 2,111	(10) 2,114		
Demand Set 2	**100**				
	196				
	324	(10) **5,513**	(10) 5,601	(10) 5,844	(10) 5,925
	400				
Demand Set 3	**100**				
	196				
	324	(1) **1,560** (2) 1,614 (3) 1,634 (4) 1,712 (5) 1,700 (10) 1,813			
	400				
Demand Set 4	**100**	(10) 420			
	196	(10) 398			
	324	(10) 392	(1) **338** (2) 426 (3) 367 (4) 367 (5) 385 (10) 451		
	400	(10) 359			

6.2.1 Effect of Search Order on Proxy Based Local Search Performance

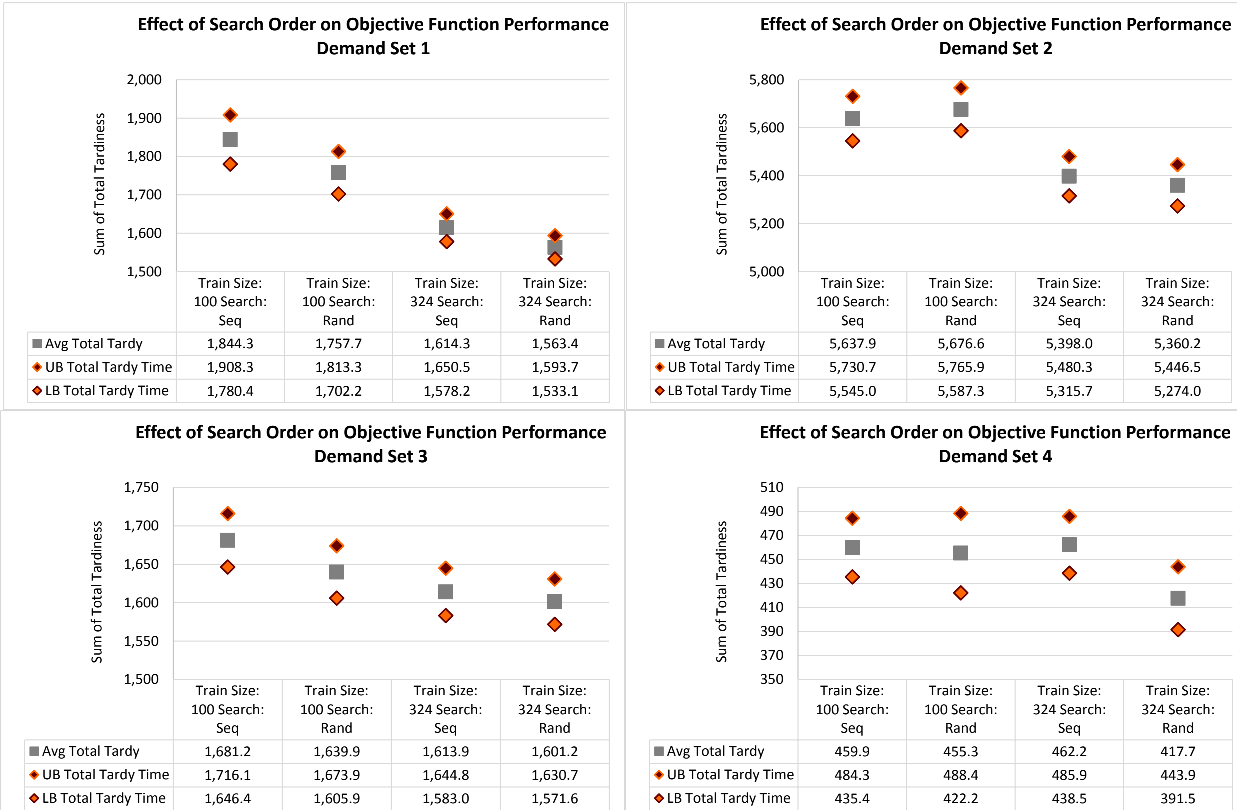
When selecting the sequence of jobs on which to search, two approaches are considered. For this analysis the jobs are either searched in a random order or searched in the order they appear in the demand list. This random search provides a more diverse exploration of the solution space by not always focusing first on job 1 and following the same sequence 1, …, N.

Objective function performance is compared to search sequence for four demand sets. For each demand set, the sum of total tardiness is evaluated for the parameter values described in Table 6.11

Table 6.11 Search Order Parameters

Training Size	**Search Approach**
100	Sequential
100	Random
324	Sequential
324	Random

The values presented in Figure 6.1 include the average total tardiness based on 100 PBLS-R runs and the upper and lower bounds based on a 95% confidence interval. Though not statistically significant, the random search sequence provides better results in all but one case. The average objective function performance for Demand Set 2 with a training size of 100 improves when a sequential search is applied. The improvement associated with random search order is most pronounced in Demand Set 4, with a training size of 324. Based on these results, a random search approach is used for all remaining runs for both PBLS-R and PBLS-NN.



Demand Set 1

	Train Size: 100 Search: Seq	Train Size: 100 Search: Rand	Train Size: 324 Search: Seq	Train Size: 324 Search: Rand
Avg Total Tardy	1,844.3	1,757.7	1,614.3	1,563.4
UB Total Tardy Time	1,908.3	1,813.3	1,650.5	1,593.7
LB Total Tardy Time	1,780.4	1,702.2	1,578.2	1,533.1

Demand Set 2

	Train Size: 100 Search: Seq	Train Size: 100 Search: Rand	Train Size: 324 Search: Seq	Train Size: 324 Search: Rand
Avg Total Tardy	5,637.9	5,676.6	5,398.0	5,360.2
UB Total Tardy Time	5,730.7	5,765.9	5,480.3	5,446.5
LB Total Tardy Time	5,545.0	5,587.3	5,315.7	5,274.0

Demand Set 3

	Train Size: 100 Search: Seq	Train Size: 100 Search: Rand	Train Size: 324 Search: Seq	Train Size: 324 Search: Rand
Avg Total Tardy	1,681.2	1,639.9	1,613.9	1,601.2
UB Total Tardy Time	1,716.1	1,673.9	1,644.8	1,630.7
LB Total Tardy Time	1,646.4	1,605.9	1,583.0	1,571.6

Demand Set 4

	Train Size: 100 Search: Seq	Train Size: 100 Search: Rand	Train Size: 324 Search: Seq	Train Size: 324 Search: Rand
Avg Total Tardy	459.9	455.3	462.2	417.7
UB Total Tardy Time	484.3	488.4	485.9	443.9
LB Total Tardy Time	435.4	422.2	438.5	391.5

Figure 6.1 Effect of Search Order on Objective Function: PBLS-R

6.2.2 Effect of Training Size on Proxy Based Local Search Performance

The training size is the amount of information that is provided to the proxy measure generation method to develop the initial proxy. Every schedule, S_k, in the training set has a set of process planning decisions, $\boldsymbol{P}^k$, and an objective function value, $f(\boldsymbol{P}^k)$. From this training set, the fit method will establish a proxy measure, $\hat{f}(\boldsymbol{P}^k)$, to be used in the local search.

For a meaningful comparison, results are normalized with respect to demand set and search depth. The training size that provides the best objective function performance for that demand set and search depth combination is assigned a normalized value of 1. The normalized values for a given training size are averaged across the search depths for a given run. This results in an aggregated measure for each training size and demand set. For uniformity, these results are scaled to one. A description of the normalization and aggregation process is provided in the following section.

6.2.2.1 Normalizing Objective Function Quality

Let $f_{i,j}^{D}$ be the average objective function found for demand set D with training size i and search depth j. Then $\left|f_i^D\right|_j$ is the normalized objective function value with respect to depth j, where,

$$|f_i^D|_j = 1 - \frac{f_{i,j}^D - \min_i\{f_{i,j}^D\}}{\min_i\{f_{i,j}^D\}}. \tag{70}$$

An example calculation for Demand Set 4 is provided in Table 6.12.

Table 6.12 Normalized Objective Function Value for Training Size

Training Size (i)	**Search Depth 100%** (j)	**Normalize Value** $|f_i^D|_j$
100	455	0.871
196	439	0.912
324	418	0.964
400	403	1.000
$\min_i\{f_{i,j}^D\}$	403	

The averaged normalized objective function value for demand set D across all search depths j is represented by $\widehat{|f_i^D|_J}$. An example calculation for a given training size ($i = 100$) is shown in Table 6.13, where the value in the cells are the normalized values, for example $|f_{100}^D|_{75\%} = 0.910$.

Table 6.13 Average Normalized Objective Function Value

	Search Depth				
	100%	**75%**	**50%**	**25%**	$\widehat{|f_i^D|_J}$
Training Size 100	0.871	0.910	0.910	0.912	0.901

The normalized objective function value with respect to search depth and demand set is $\left|\left|f^D\right|_j\right|_i$, where

$$\left| \left| f^D \right|_j \right|_i = 1 - \frac{\widehat{|f_i^D|}_J - \min_j\{\widehat{|f_i^D|}_J\}}{\min_j\{\widehat{|f_i^D|}_J\}}. \tag{71}$$

Table 6.14 provides the final normalized values for Demand Set 4. These values can be compared to other demand sets such that the effect of training size on objective function performance can be evaluated.

Table 6.14 Normalization with Respect to Search Depth and Demand Set

Training Size (i)	**Average Normalized Value** $\widehat{\lvert f_i^D \rvert}_J$	**Normalized Value with Respect to Search Depth and Demand Set** $\left\lvert \left\lvert f^D \right\rvert_j \right\rvert_i$
100	0.901	0.905
196	0.959	0.964
324	0.978	0.984
400	0.995	1.000
$\min_i\{f_{i,j}^D\}$	1,557	

6.2.2.2 *Results for Regression Based Proxy Measure*

Four different training set sizes (100, 196, 324, 400) are evaluated for generating the regression based proxy measure used in PBLS-R. The normalized relative objective function performance, $\left| \left| f^D \right|_j \right|_i$, is plotted against the training set size in Figure 6.2

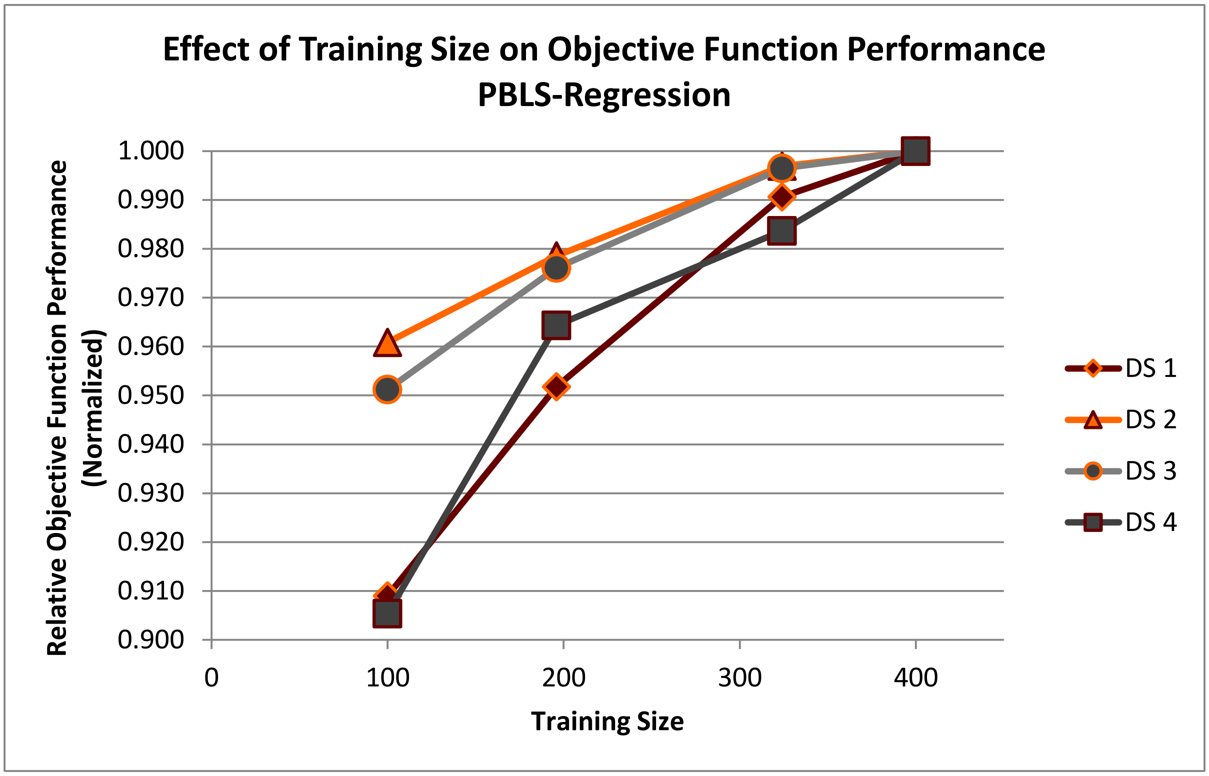


Figure 6.2 Effect of Training Size on Objective Function: PBLS-R

As show in Figure 6.2, objective function performance on average improves as the training size increases. As the training size increases, additional detailed schedules are required. Generating these schedules may be time consuming depending on the scheduling heuristic used. Because these schedules are independent, however, they can be generated in parallel to mitigate impact on overall processing time. Based on these results, a training size of at least 400 is advised when applying PBLS-R to demand sets with these characteristics. Exploring the effect of a training size larger than 400 on performance and establishing an analytic based recommendation for training size for a given demand set are subjects for future work.

6.2.2.3 Results for Neural Network based Proxy Measure

Four different training set sizes (100, 196, 324, 400) for two demand sets (DS 1, DS 4) are evaluated for generating the neural network based proxy measure. The normalized relative objective function performance, $\left|\left|f^D\right|_j\right|_i$, is plotted against the training set size in Figure 6.2.

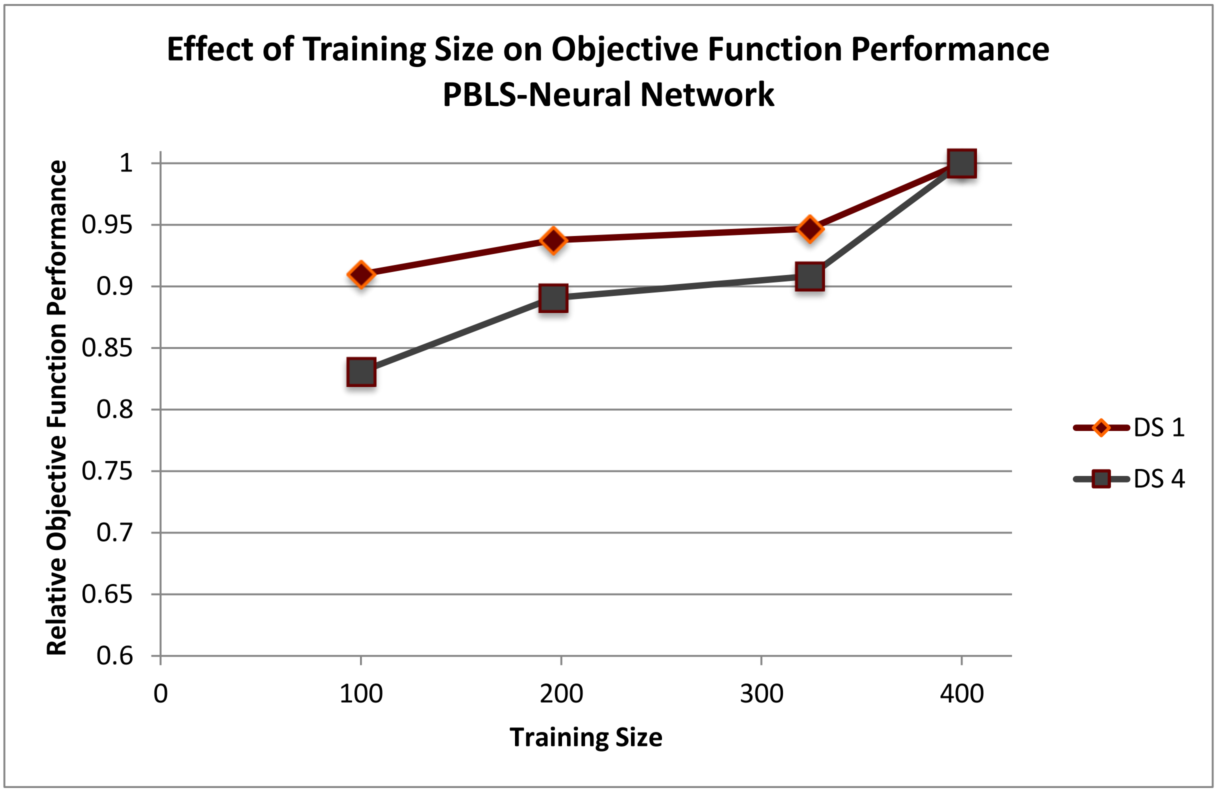


Figure 6.3 Effect of Training Size on Objective Function: PBLS-NN

For Demand Set 1, increasing training size from 100 to 400 resulted in a 9% improvement in objective function performance. For Demand Set 4, the increase from a training size of 100 to a training size of 400 represents a 17% improvement in objective function performance.

The neural network training method in Matlab employs a multi-stage training approach to protect against over fitting. The input data set, $\boldsymbol{S}^T$, is divided into three sub sets: Set A, Set B and Set C. Set A is used directly to train the model. The final model is that which minimizes mean squared error when compared to sets B and C. Future work is required to understand the effect of a training size larger than 400 on performance and to establish an analytic basis for recommending a training size for a given demand set.

6.2.3 Effect of Search Depth on Proxy Based Local Search Performance

Search depth dictates the percentage of jobs from the demand set that are examined during the local search portion of the heuristic. If the search depth is set to 100%, then Machine Swap and Production Version Swap searches, guided by the proxy measure, are applied to every job in the demand set. For any percentage less than 100%, the search method is applied to the specific percentage of randomly selected jobs in the demand set.

For a meaningful comparison of the effect of search depth on PBLS, results are normalized with respect to demand set and training set size. For a given demand set and training, there is one search depth with a normalized value of 1 representing the search depth that provides the best objective function performance for that demand set, and training size combination. The normalized values for a given search depth are averaged across the different training set sizes for a given run. This results in an aggregated measure for each search depth and demand set. For uniformity these results are scaled to one. A description of the normalization and aggregation process is presented in the following sections.

6.2.3.1 Normalizing Objective Function Quality

Let $f_{i,j}^{D}$ be the average objective function found for demand set D with training size i and search depth j. The normalized objective function value for training size i, $\left|f_j^D\right|_i$, is expressed as follows:

$$\left|f_j^D\right|_i = 1 - \frac{f_{i,j}^D - \min_j\{f_{i,j}^D\}}{\min_j\{f_{i,j}^D\}} \tag{72}$$

In Equation (72), $\widehat{\left|f_j^D\right|_\iota}$ is the averaged normalized objective function value for demand set *D* across all search depths j. The normalized objective function value with respect to search depth, $\left|\left|f^D\right|_i\right|_j$, is expressed as follows:

$$\left|\left|f^D\right|_i\right|_j = 1 - \frac{\widehat{\left|f_j^D\right|_\iota} - \min_i\left\{\widehat{\left|f_j^D\right|_\iota}\right\}}{\min_i\left\{\widehat{\left|f_j^D\right|_\iota}\right\}} \tag{73}$$

6.2.3.2 Results for Regression based Proxy Measure

Four different search depths (25%, 50%, 75%, 100%) are evaluated for PBLS-R with respect to normalized relative objective function performance. The results are displayed in Figure 6.4, where the vertical axis is the value of the normalized objective function, $\left\| f^D \right|_i \Big|_j$ and the horizontal axis represents the search depth, j, used by PBLS-R.

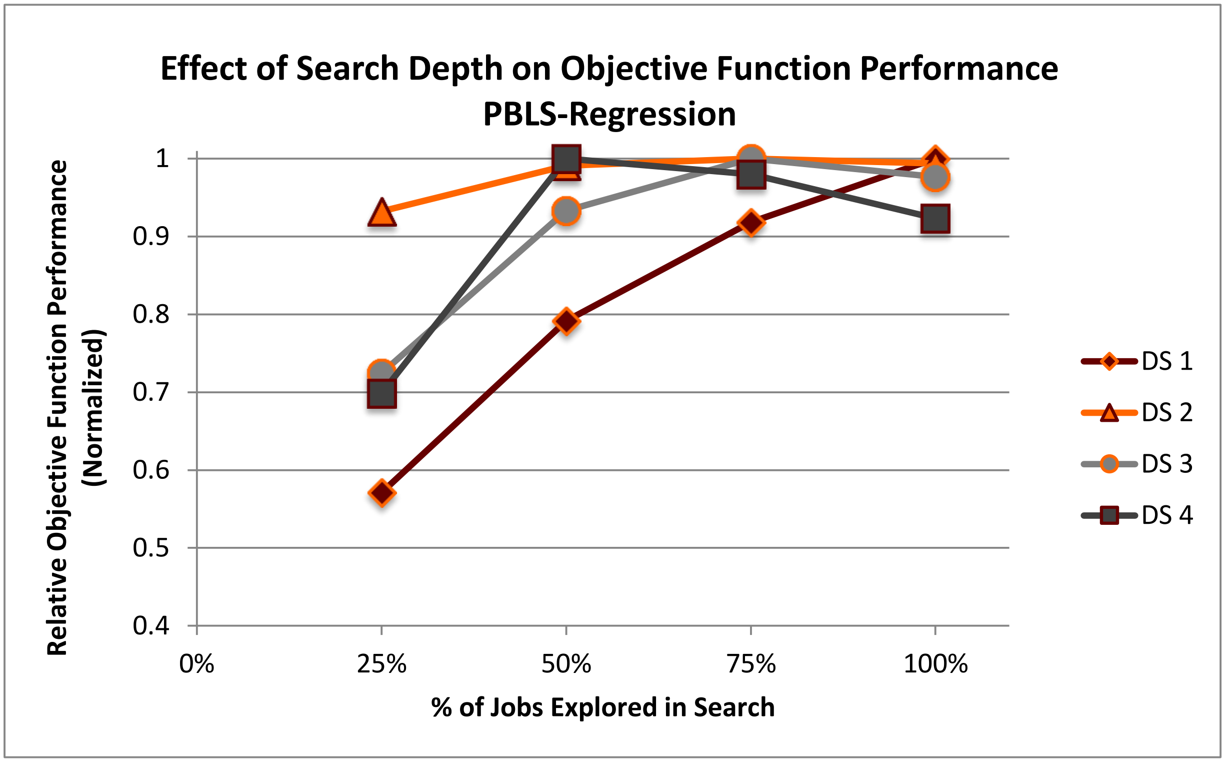


Figure 6.4 Effect of Search Depth on Objective Function: PBLS-R

Objective function performance generally improved as search depth increased, as illustrated in Figure 6.4. For Demand Set 4, however, the best average performance resulted from a search depth of 50%. Demand Set 3 also saw a minor decrease in performance beyond a 75% search depth.

Demand Set 4 is the largest demand set with respect to number of jobs. The decreased performance with search depths greater than 50% may be related to over applying the proxy measure. Ideally, the proxy measure search should provide directionally accurate measurements and not have an issue of decreased performance when applied to a large number of jobs in a demand set. Three of the four demand sets demonstrated increased objective function performance with a search depth greater than 50%.

Search depth does influence overall run time by increasing the number of times the proxy measure is evaluated. For proxy measures that are not computationally intensive, however, the effect should be minimal. From the results in Figure 6.4, a search depth of 75% to 100% is suggested when applying PBLS-R to demand sets with similar characteristics.

6.2.3.3 Results for Neural Network Based Proxy Measure

For PBLS-NN, the effect of search depth on objective function performance is evaluated and results are presented in Figure 6.4. The vertical axis is the value of the normalized objective function, $\left||f^D|_i\right|_j$, and the horizontal axis represents the search depth, j, used in PBLS-NN.

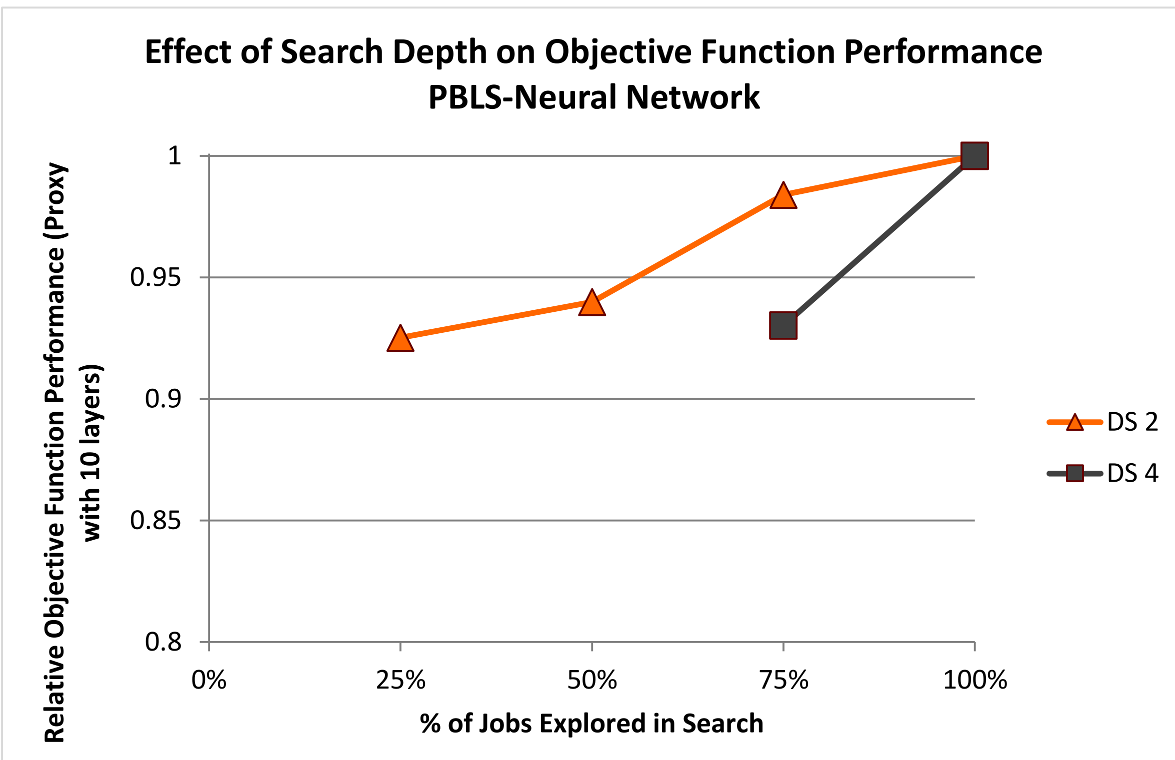


Figure 6.5 Effect of Search Depth on Objective Function: PBLS-NN

As shown in Figure 6.5, a positive relationship emerges between search depth and objective function performance. Increasing the search depth does require additional proxy measure evaluations. The evaluation time of the neural network based proxy measure used in PBLS-NN is larger than that of the regression based proxy used in PBLS-R (see Section 6.2.5 for more details). Although the relative improvement from increased search depth was less than 10%, increasing the search depth used in PBLS-NN at the expense of processing time is advised. Developing an analytic model to estimate the tradeoff between processing time and expected quality of PBLS-NN results is the subject of future research.

6.2.4 Effect of Layer Count on Performance for Neural Network Proxy

The number of hidden layers in a neural network describes the complexity of the network. Each layer applies some function to the input values from nodes in the previous layer as shown in Figure 6.6. With more layers the neural network is capable of modeling increasingly complex structures. As with traditional polynomial regression modeling the increased flexibility can result in over fitting. Over fitting occurs when the model conforms too closely to the training set. An over fit model provides poor prediction of future events.

Six different neural network architectures (1, 2, 3, 4, 5 and 10 hidden layers) are evaluated for generating the neural network based proxy measures used in PBLS-NN. The average objective function values and 95% confidence intervals are presented in Figure 6.6 for three demand sets. While not statistically significant in all cases, the results suggest better performance is obtained when the proxy measure is generated with fewer layers. Based on the results of the three demand sets examined here, a single layer neural network model is advised for PBLS-NN. Exploring alternative network architectures is the focus of future work.

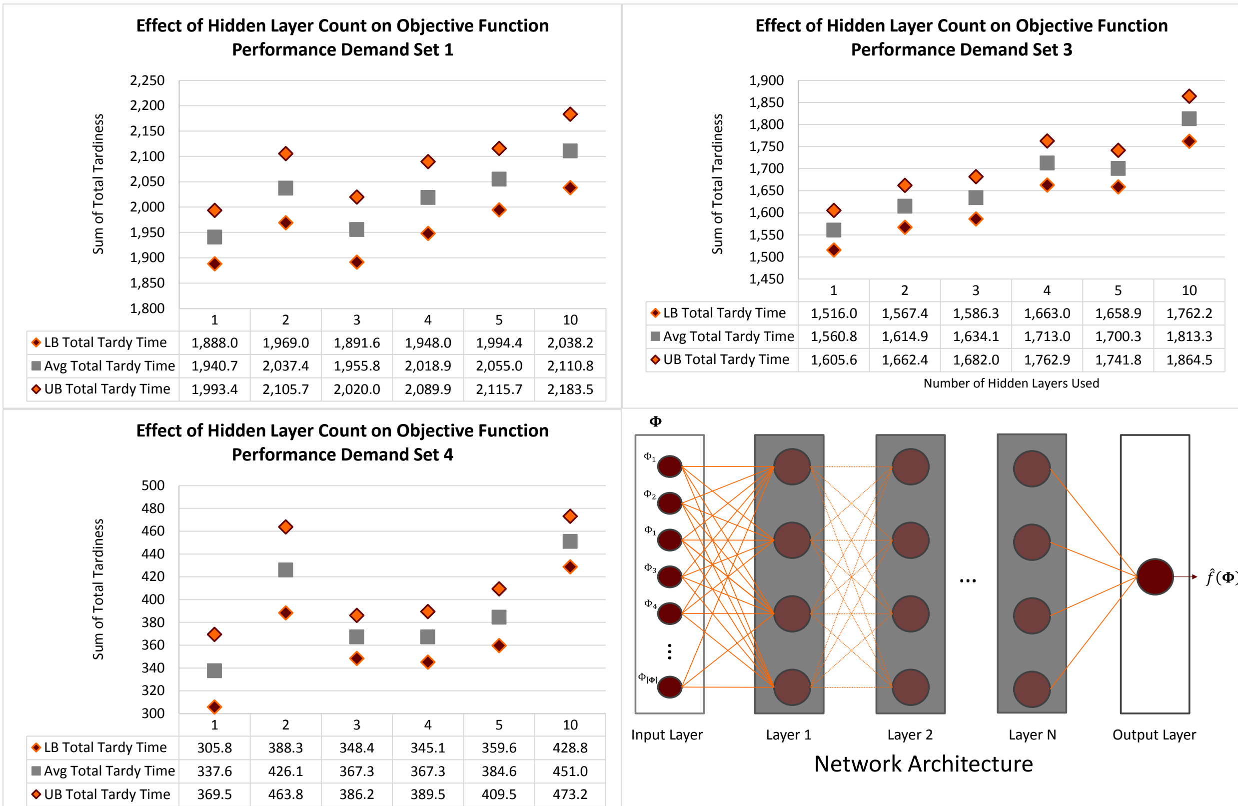


Figure 6.6 Effect of Layer Count on Objective Function: PBLS-NN

6.2.5 Evaluating Processing Time

The complexity and processing time required by PBLS depends on a variety of factors. Parameters such as the number of jobs in the demand set N^J, the training size N^T, the search count N^S, and the search depth S^D influence processing time. The time to generate a detailed schedule, T^S, depends on the scheduling heuristic and the quantity of operations to be scheduled. The time required to train the proxy measure, T^T, depends on the proxy measure fit method, the number of descriptive statistics, and the size of the training set. The time required to perform a proxy measure evaluation, T^E, depends on the final structure of the proxy measure. Considering the structure of PBLS, an estimate of time required execute the search is provided as follows:

$$Time = N^T T^s + T^T + 2N^S[N^J S^D(T^E) + T^S]. \tag{74}$$

From Equation (74), the largest multiplier, $2N^S N^J S^D$ is applied to the proxy measure evaluation time, T^E, and the least multiplier is placed on training time, T^T. More sophisticated training methods that still yield easy to evaluate proxy measures is an appealing direction for future work because additional time in training does not have the same multiplicative effect that an increase in proxy measure evaluation time has. PBLS leverages the proxy measure to make process planning swaps without the need to generate a detailed schedule. Even with this approach, a detailed schedule is generated $N^T + 2N^S$ times, thus highlighting the importance of a computationally efficient yet effective dispatching heuristic. For example, with a training size of 400 and a search count of 100, a dispatching heuristic that takes even a single minute to generate a detailed schedule, would result in a PBLS run of over eight hours.

Table 6.15 provides sample execution times for a single demand set captured using Java 1.7 run on a desktop computer equipped 32GB of RAM and an Intel Xeon E5-2620-v3 2.5GHz CPU. Each step is influenced by elements in the specific demand set that are not fully captured here. However, these results provide a relative magnitude of each step as well as a comparison of computational times for regression based proxy and neural network based proxy. Training the neural network based proxy measure took on average three times longer than training the regression based proxy measure. The time required to evaluate the neural network based proxy measure was 380 times that of the regression based proxy measure on average.

Table 6.15 Execution Time in Milliseconds for Proxy Based Local Search Steps

Step	**Regression**	**Neural Network**
Detailed Schedule, T^S	19	19
Training T^T	214	664
Evaluation T^E	0.05	19

Assuming a training size of 400, demand set of 100 jobs, search depth of 100%, and a search count of 100, the estimated processing time for both search approaches can be estimated using Equation (74), and the values in Table 6.15.

Table 6.16 Comparison of Estimated Run Time

	Regression	**Neural Network**
Estimated Time	12,614 ms	392,064 ms

6.2.5.1 Effect of Implementation on Processing Time

Training and evaluation of the neural network is inherently more complicated than that of the regression model. However, the effect on processing time is confounded by the implementation method. Using a third party API to interface between Java and Matlab as implemented in this research requires overhead and increases processing time. A more efficient means to train and evaluate the neural network may provide a notable improvement on processing time.

Each schedule in the training set is independent and could be generated in parallel. Additionally, each of the N^S searches could be conducted in parallel because they are performed on independent schedules and the results are independent. Therefore, the complexity of the heuristic can be greatly reduced by parallel processing. Assuming N^P parallel processers are available, the estimated time from Equation (74), can be reduced to

$$Time = \frac{N^T T^s}{N^P} + T^T + \frac{2N^S[N^J S^D(T^E) + T^S]}{N^P}. \tag{75}$$

Additionally, a repository of historic schedules for the same facility with similar demand mix may be stored and used to train the network. The effect of using historic results to train the proxy measure is the subject of future research.

6.3 Comparison of Proxy Based Local Search to Random Walk

The proxy based local search (PBLS) determines whether to keep or reject each proposed production plan change based on the estimated effect the move will have on the objective function. Following a series of proposed process planning changes, a detailed schedule is generated and the set of process planning decisions are kept if the resulting detailed schedule improves the objective function value. Because a new process plan is only accepted if it improves the objective function of the detailed schedule, this search approach is non-decreasing where no worse solution will be accepted. In this section, a random walk based local search is applied to the demand sets from Section 6.1.4. The random walk based local search employs the same local search moves as the PBLS, but new process planning decisions are accepted based on a uniform random variable with probability 0.5, instead of the value from a proxy measure. The results of the random walk are compared to those from PBLS-R and PBLS-NN.

6.3.1 Summary of Results for Local Search Decision Approaches

The PBLS-R and PBLS-NN out performed local search guided by random walk for all four demand sets compared. The average objective function value along with upper and lower bounds based on a 95% confidence interval are provided for each of the three local search decision approaches. The 95% confidence interval shows PBLS-R and PBLS-NN significantly out perform the local search guided by random walk. The results presented in Table 6.17 demonstrate that both regression and neural network based proxy measures are capable of capturing the underlying relationship between process planning decisions and scheduling performance. Across these four demand sets, PBLS provided as much as 44% improvement in total tardiness objective when compared to results obtained from guiding process planning changes with random walk.

Table 6.17 Summary of Results for Local Search Decision Approaches

	PBLS-R[1]	**PBLS-NN**	**Random Walk**[1]
DS 1	UB: 1,593 Avg: **1,563** LB: 1,533	UB: 1,993 Avg: 1,940 LB: 1,888	UB: 2,833 Avg: 2,802 LB: 2,772
DS 2	UB: 5,474 Avg: **5,390** LB: 5,305	UB: 5,531 Avg: 5,414 LB: 5,297	UB: 6,250 Avg: 6,213 LB: 6,177
DS 3	UB: 1,631 Avg: 1,601 LB: 1,571	UB: 1,606 Avg: **1,561** LB: 1,516	UB: 2,505 Avg: 2,482 LB: 2,458
DS 4	UB: 444 Avg: 418 LB: 391	UB: 369 Avg: **338** LB: 307	UB: 600 Avg: 580 LB: 561

[1] Average for 100 runs: $N^T = 324$, Depth = 100%

[2] Average for 50 runs: $N^T = 324$, Depth DS 1, 2 = 100%. Depth DS 3, 4 = 75%.

6.4 Comparison to Optimal for Small Demand Sets

The main application domain for the proxy based local search (PBLS) is large demand sets for which finding optimal solutions is intractable. To demonstrate the effectiveness of the solution approach, PBLS is applied to smaller integrated process planning and scheduling problems for which finding an optimal solution is feasible, though time consuming. An overview of the two demand sets is provided in the next section. Results from PBLS and local search guided by random walk are compared to the optimal solution for each demand set. The results in Section 6.4.6, demonstrate that PBLS is capable of generating solutions within 14% of optimality for small data sets on average. The comparison to optimal also provides insight into the ability of the OU-MW heuristic to generate good solutions to the scheduling sub-problem. Individual solutions generated from the dispatching heuristic were within 3% of optimal.

6.4.1 Summary of Small Demand Sets

Finding optimal solutions for job shop scheduling problems with sequence dependent setups is difficult even for problems with 15 or 20 jobs (Artigues *et al.*, 2004). Process planning decisions further complicate the already NP-hard job shop scheduling problem. In order to compare the proposed PBLS approaches to an optimal approach, two demand sets are created that captures the main characteristics of the IPPS problem but are sized such that the optimal solution can be determined. Details of each demand set are summarized in

Table 6.18. The requirement length for each job was generated using a uniform random distribution, $U{\sim}(5{,}000\ ,150{,}00)$. Due dates are generated based using a uniform random distribution, $U{\sim}(6, X)$, where X is calculated based on the total demand length for the data set, such that

$$X = \frac{\sum demand}{2{,}300{,}000\ ft} * 144\ hours \tag{76}$$

Table 6.18 Summary of Demand Sets for Comparison to Optimal

	Demand Set O1	**Demand Set O2**
Number of Jobs	9	13
Unique Materials	5	4
Total Req. Length (ft)	686,716	737,193
Avg. Req. Length (ft)	76,302	56,707
Min Req. Length (ft)	27,044	11,090
Max. Req. Length (ft)	81,387	102,719
Avg. Due Date (hours)	24.3	28.2
Standard Dev. Due Date (hours)	10.5	9.6
Avg. Process Plans per Job	5	4.6

The proxy based local search was applied to each demand set using both a regression and neural network based proxy measure. A search count of 25 is used for the small demand sets to ensure the solution quality is a measure of the proxy search and not brute force.

6.4.2 Setup Group Matrix for Small Demand Sets

The setup group matrix describes the time required to prepare a machine between two successive operations. For the demand sets in this dissertation a setup group is defined for each operation in a job. The matrix used for this problem is asymmetric, meaning $s_{s4,s18} \neq s_{s18,s4}$. The full setup matrix is provided in Table 6.19.

Table 6.19 Setup Matrix for Small Data Sets

To Setup Group

From Setup Group	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19
S1	0.5	2.5	2.5	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	X
S2	2.5	0.5	2.5	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	X
S3	2.5	2.5	0.5	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	X
S4	8	8	8	0.5	3	8	8	8	8	8	8	8	8	8	8	4	4	4	X
S5	8	8	8	3	0.5	8	8	8	8	8	8	8	8	8	8	4	4	4	X
S6	8	8	8	8	8	0.5	3	8	8	8	8	8	8	8	8	8	8	8	X
S7	8	8	8	8	8	3	0.5	8	8	8	8	8	8	8	8	8	8	8	X
S8	8	8	8	8	8	8	8	0.5	2	8	8	8	8	8	8	8	8	8	X
S9	8	8	8	8	8	8	8	2	0.5	8	8	8	8	8	8	8	8	8	X
S10	8	8	8	8	8	8	8	8	8	0.5	1.5	2	2	2	8	2	2	2	X
S11	8	8	8	8	8	8	8	8	8	1.5	0.5	2	2	2	8	2	2	2	X
S12	8	8	8	8	8	8	8	8	8	2	2	0.5	1.5	1.5	8	8	8	8	X
S13	8	8	8	8	8	8	8	8	8	2	2	1.5	0.5	1.5	8	8	8	8	X
S14	8	8	8	8	8	8	8	8	8	2	2	1.5	1.5	0.5	8	8	8	8	X
S15	8	8	8	8	8	8	8	8	8	8	8	8	8	8	0.5	8	8	8	X
S16	8	8	8	2	2	8	8	8	8	2	2	8	8	8	8	0.5	2	2	X
S17	8	8	8	2	2	8	8	8	8	2	2	8	8	8	8	2	0.5	2	X
S18	8	8	8	2	2	8	8	8	8	2	2	8	8	8	8	2	2	0.5	X
S19	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0.5

6.4.3 Process Plan Flexibility Network for Small Demand Sets

Six unique material types appear in the two small demand sets (DS O1 and DS O2) presented in this section. Each material type WF-1 to W-7 has a pre-determined set of valid processing plans. Process plan flexibility networks are commonly used to communicate a job's operation and process flexibility (Ho and Moodie, 1996). To account for minimum and maximum wait time constraints additional arcs are drawn between operations of the same job. A dashed arc between two operations indicates a minimum wait time constraint. A double solid line between two operations indicates the presence of a maximum wait time constraint. Each operation in the C&L problem belongs to a specific setup group, and each node in the flexibility network includes the setup group the operation belongs to. Due to varying scrap rates between machines, process time depends on upstream machine selections. Therefore, processing times have been omitted from the flexibility networks. The process plan flexibility network for WF-1, Figure 6.7 describes three different production versions. PV 2 for WF-1 requires four completion of four operations, represented by the four nodes in the flexibility network. The first operation of PV 2 can be processed on either machine A or machine B. The dashed line between the operation 1 node and operation 2 node in PV 2 indicates a minimum wait constraint between the two operations. The double solid line between the operation 2 node and operation 3 node indicates a maximum wait constraint between the two operations.

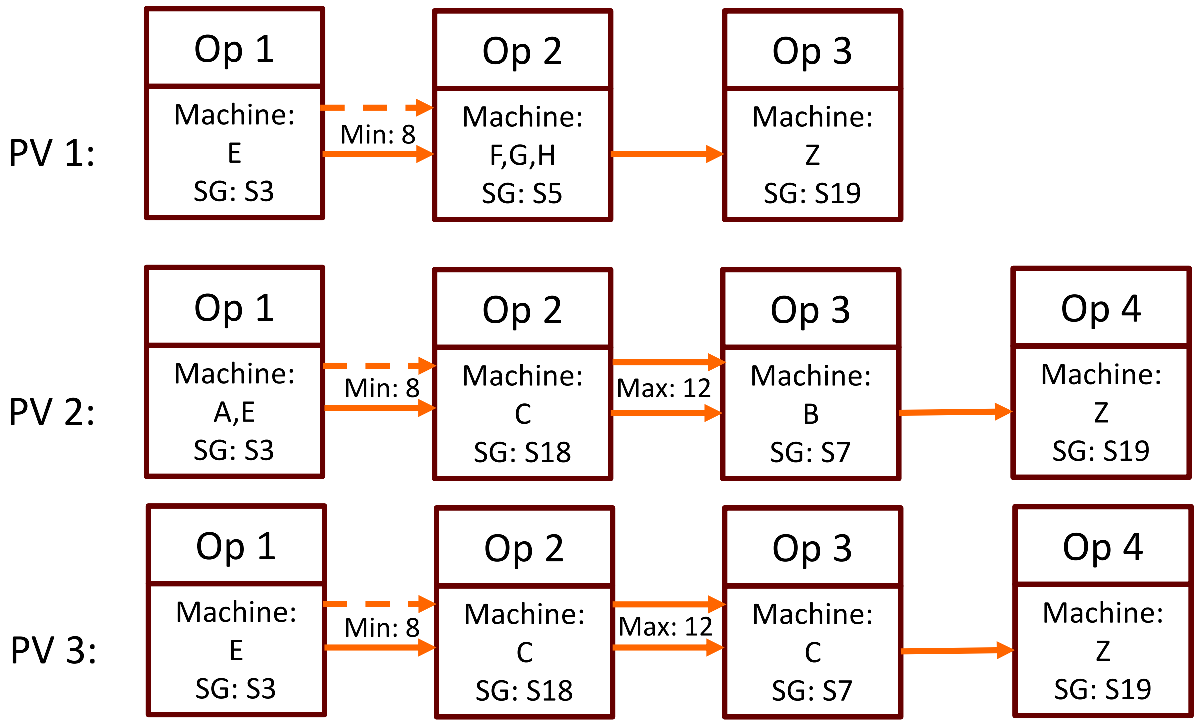


Figure 6.7 WF-1 Process Plan Flexibility Network

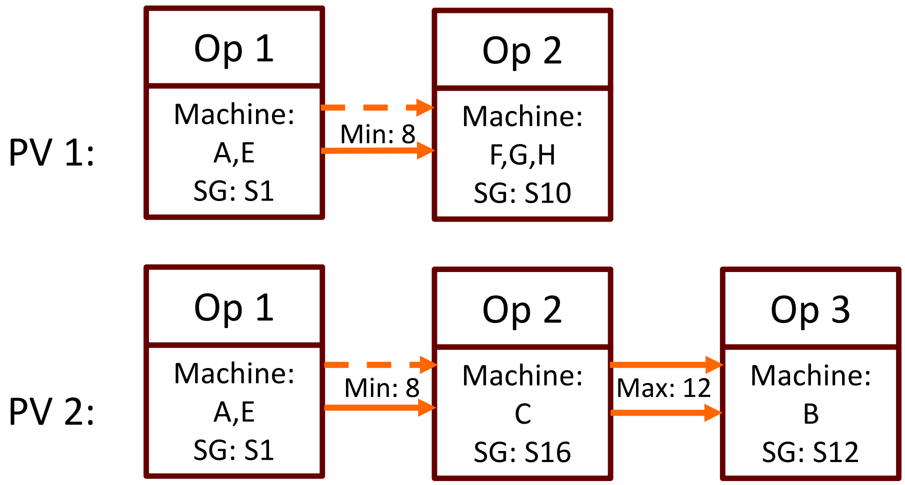


Figure 6.8 WF-2 Process Plan Flexibility Network

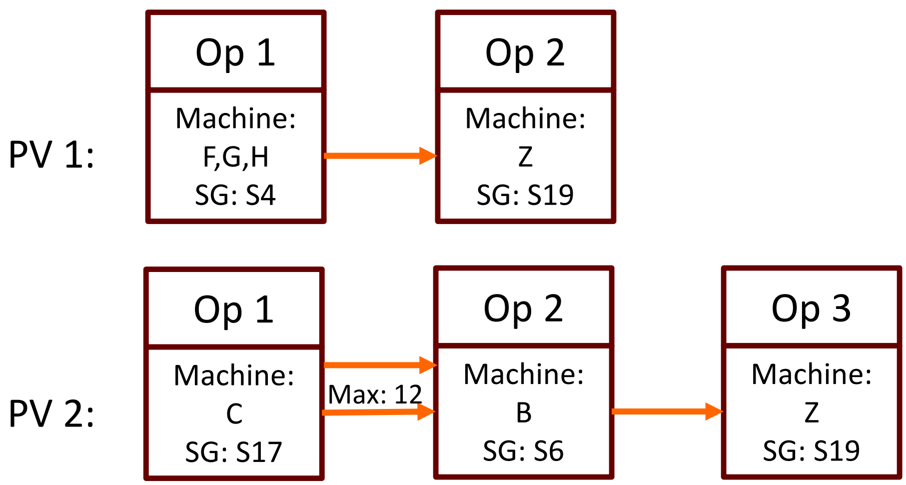


Figure 6.9 WF-3 Process Plan Flexibility Network

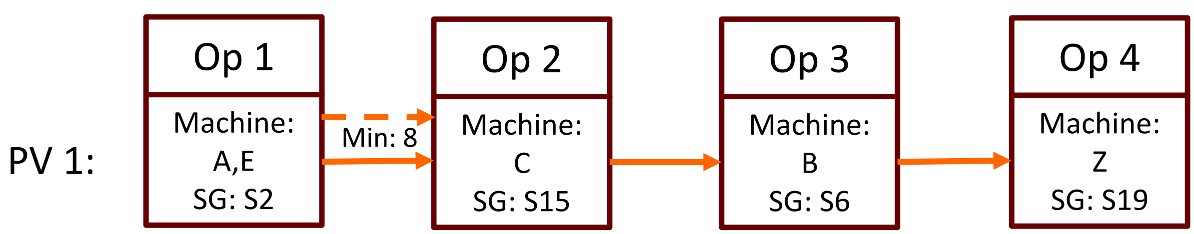


Figure 6.10 WF-4 Process Plan Flexibility Network

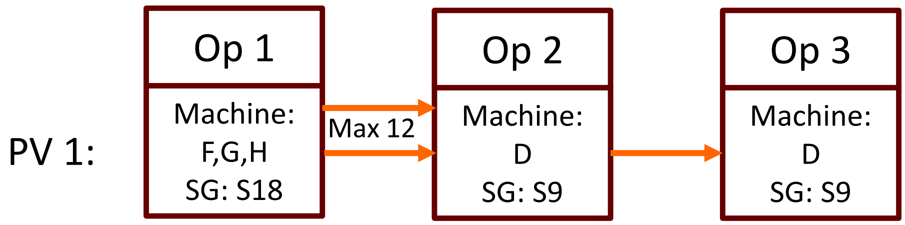


Figure 6.11 WF-5 Process Plan Flexibility Network

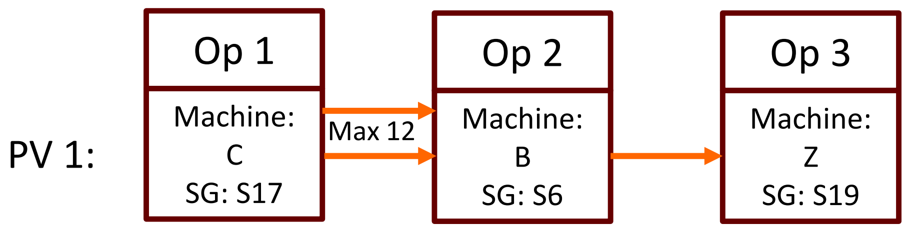


Figure 6.12 WF-6 Process Plan Flexibility Network

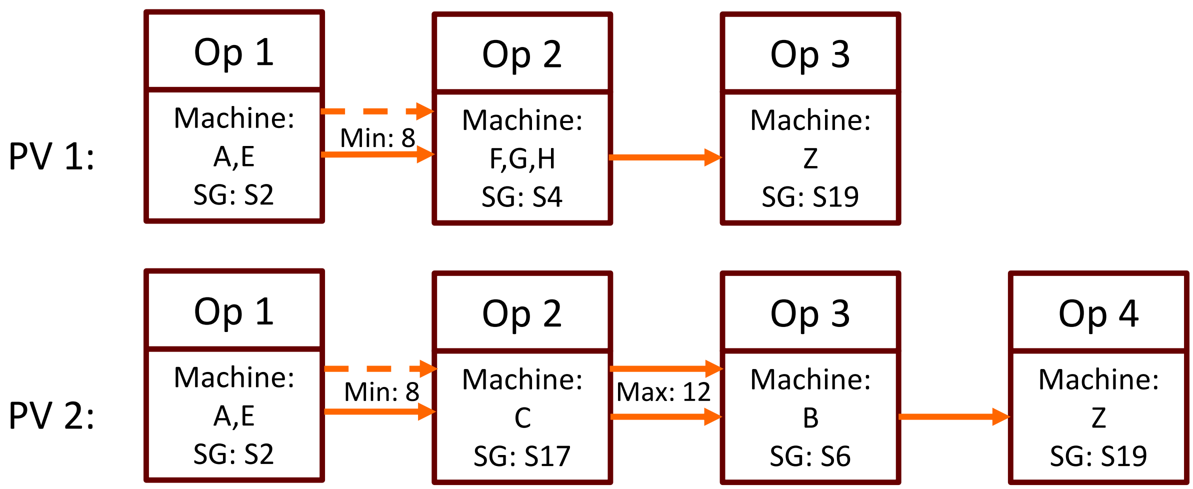


Figure 6.13 WF-7 Process Plan Flexibility Network

6.4.4 Demand Set O1

Demand Set O1 (DS O1) is designed to target the complexity of the C&L problem. Material types were chosen to ensure a representative level of setup complexity for the C&L problem. Five unique material types are required across the nine jobs. Table 6.20 provides a detailed description of the demand set.

For this demand set, an optimal sequence is shown in Table 6.21. In addition, the sequence with the least total tardiness from PBLS-R and PBLS-NN are shown for comparison.

Table 6.20 Job Details for DS O1

Job	Material Type	Length	Due Date
1	WF-1	72,677	8
2	WF-2	128,365	29
3	WF-3	63,637	23
4	WF-4	58,075	41
5	WF-5	84,183	7
6	WF-2	27,044	20
7	WF-3	142,835	30
8	WF-4	56,413	30
9	WF-5	53,487	31

Table 6.21 Comparison of Process Planning Solutions for DS O1

Job	Material Type	Machine Sequence Optimal	Machine Sequence PBLS-R	Machine Sequence PBLS-NN
1	WF-1	E,H,Z (29.49)	E,G,Z (25.28)	E,G,Z (30.54)
2	WF-2	A,G (27.09)	E,H (28.44)	E,G (32.78)
3	WF-3	F,Z (0.0)	C,B,Z (6.25)	H,Z (0.0)
4	WF-4	E,C,B,Z (25.47)	A,C,B,Z (11.41)	E,C,B,Z (39.43)
5	WF-5	G,D,D (79.96)	F,D,D (79.96)	F,D,D (79.69)
6	WF-2	E,F (5.73)	A,H (10.74)	A,G (0.0)
7	WF-3	H,Z (0.0)	H,Z (0.0)	H,Z (4.51)
8	WF-4	A,C,B,Z (16.40)	E,C,B,Z (49.29)	A,C,B,Z (22.63)
9	WF-5	G,D,D (33.80)	G,D,D (8.24)	G,D,D (8.24)
Total Tardiness		**217.95**	**219.62**	**218.09**

The sequence and schedule results from PBLS-R and PLBS-NN in Table 6.21 represent the best solutions found from 100 and 50 runs respectively. Although the results may not represent the average solution quality, they demonstrate the capability of the OU-MW heuristic to generate schedules with good total tardiness. Detailed scheduling results for Demand Set O1 are provided in Appendix A.1.

6.4.5 Demand Set O2

Demand Set O2 (DS O2) is a larger demand set than Demand Set O1, with an additional four jobs. The thirteen jobs in Demand Set O2 represent four unique materials. While Demand Set O2 is larger than Demand Set O1 less setup complexity is present than in Demand Set O1. Table 6.20 provides a detailed description of Demand Set O2.

Table 6.22 Job Details for DS O2

Job	Material Type	Length	Due Date
1	WF-6	72,945	21
2	WF-7	68,145	15
3	WF-3	33,127	34
4	WF-1	84,409	39
5	WF-3	11,090	21
6	WF-3	34,399	48
7	WF-7	74,235	18
8	WF-7	32,338	20
9	WF-3	41,853	35
10	WF-6	54,103	31
11	WF-3	30,420	33
12	WF-6	102,719	18
13	WF-3	97,410	34

For this demand set, an optimal sequence is shown in Table 6.23. In addition, the sequences with the least total tardiness from PBLS-R and PBLS-NN are shown for comparison.

The sequence and schedule results from PBLS-R and PLBS-NN in Table 6.23 represent the best solutions found from 100 and 50 runs, respectively. While the results may not represent the average solution quality, they demonstrate the capability of the OU-MW heuristic to generate schedules with good total tardiness. Detailed scheduling results for Demand Set O2 are provided in Appendix A.2.

Table 6.23 Comparison of Process Planning Solutions for DS O2

Job	Material Type	Machine Sequence Optimal	Machine Sequence PBLS-R	Machine Sequence PBLS-NN
1	WF-6	C,B,Z (12.86)	C, B, Z (10.77)	C,B,Z (10.33)
2	WF-7	A,H,Z (14.82)	A,H Z (20.58)	A,F,Z (21.12)
3	WF-3	G,Z (0.0)	G, Z (0.0)	H,Z (0.0)
4	WF-1	E,G,Z (9.48)	E, G, Z (9.48)	E,H,Z (9.48)
5	WF-3	F,Z (0.0)	F, Z (0.0)	G,Z (0.0)
6	WF-3	F,Z (0.0)	F, Z (0.0)	G,Z (0.0)
7	WF-7	A,H,Z (25.88)	A, G, Z (25.88)	A,G,Z (25.88)
8	WF-7	E,G,Z (0.0)	H, E, Z (0.0)	E,F,Z (0.47)
9	WF-3	F,Z (0.0)	F, Z (0.0)	G,Z (0.0)
10	WF-6	C,B,Z (8.78)	C, B, Z (8.78)	C,B,Z (8.78)
11	WF-3	G,Z (0.0)	H, Z (0.0)	H,Z (0.0)
12	WF-6	C,B,Z (55.05)	C, B, Z (55.05)	C,B,Z (55.05)
13	WF-3	H, Z (0.0)	H, Z (0.0)	H,Z (0.0)
Total Tardiness		**126.87**	**130.54**	**131.11**

6.4.6 Results from PBLS-R, PBLS-NN, and Optimal Solution Approaches

The best solution found solving for optimality and the average results from PBLS-R, PBLS-NN, and random walk are presented in Table 6.24. The value presented for the regression based proxy and neural network based proxy approaches, represents the average of the results from 100, and 50 runs respectively. The average results for both PBLS-R and PBLS-NN are within 37% of optimal for DS O2. These results are less impressive than the results for PBLS-R and PBLS-NN applied to DS O1, which are within 13.3% of optimal. The random walk provided the worst performance for each of the demand sets, with the average objective function value returned for DS O2 being 91% above the optimal. The comparison to optimal highlights the drawbacks of a heuristic approach, which does not guarantee a convergence to optimal. However, the PBLS approach does generate good feasible schedules in a fraction of the time required to generate an optimal solution.

Table 6.24 Comparison of Results for Small Demand Set

	Optimal[1]	Regression[2]	Neural Network[3]	Random Walk[4]
DS O1	217.9	247.7 (13.6%)	247.0 (13.3%)	302.5 (38.8%)
DS O2	123.9	160.0 (29.7%)	168.8 (36.9%)	235.3 (90.8%)

[1]Solving DS O2 terminated after 100 hours, optimality gap 2.8%
[2] Average of 100 Runs, $N^T = 324$, Depth = 100%
[3] Average of 50 Runs, $N^T = 324$, Depth = 100%, Layers = 1
[4] Average of 100 Runs, Depth = 100%, Accept Probability = 0.5

Optimal solutions were found using IBM CPLEX V12.5 run on a system with two 3.1GHz Intel Xeon E5-2687 CPUs and 32 GB of RAM. Solving DS O1 to optimality took 112 minutes. Solving DS O2 required multiple iterations with the addition of a lower bound constraint after each iteration. CPLEX ran out of memory multiple times while trying to solve DS O2. The best integer solution and lower bound were captured for each model at the time of failure. The lower bound was converted into a constraint and added to LP model from Section 4.2, and the best integer solution was provided to CPLEX as a warm start for the next iterations. Attempts to obtain a proven optimal solution were terminated after 50 hours. The best integer solution found for DS O2, 126.9, was within 2.8% of the established theoretical lower bound 123.3. For comparison purposes the theoretical lower bound is used as a conservative estimate of the optimal objective function value in Table 6.24

6.5 Summary of Findings

The proxy based local search (PBLS) is capable of generating good solutions to realistic integrated process planning and scheduling problems. The results presented in Section 6.2 provide insights on the influence that parameters play on overall solution quality. The findings suggest increasing the number of schedules used to train the proxy measure and increasing the search depth can improve overall performance as measured by the sum of total tardiness with tolerable effect on processing time for both PBLS-R and PBLS-NN. Both the neural network based proxy measure in PBLS-NN and the regression based proxy measure used in PBLS-R significantly outperformed a local search guided by random walk, as discussed in Section 6.3. The comparison to a random walk provides statistically significant evidence that both the neural network proxy measure and regression proxy measure are capable of capturing the relationship between process planning decisions and the sum of total tardiness objective in the scheduling problem. A summarized view of the average objective function obtained with PBLS applied to industry representative demand sets (DS 1 – DS 4) is presented in Table 6.25.

Table 6.25 Summary of Average Solutions Found for Each Industry Demand Set

	Total Tardiness		
	PBLS-R Average	**PBLS-NN Average**	**Random Walk Average**
DS 1	**1,557**	1,941	2,802
DS 2	**5,381**	5,513	6,213
DS 3	**1,541**	1,560	2,482
DS 4	382	**338**	580

Due to the complexity of the integrated process planning and scheduling problem, solving an industry representative problem to optimality was not feasible. To evaluate the quality of the proxy based local search to optimal, two smaller demand sets were generated. Comparison of the solutions obtained from PBLS-R and PBLS-NN to the optimal solution, as shown in Table 6.26, provides two important benchmarks. First, PBLS is capable of generating good solutions (13% - 37% from optimal). Second, the OU-MW dispatching heuristic developed in Section 5.1.5 is capable of generating near optimal solutions to the single route scheduling problem for small demand sets.

Table 6.26 Summary of Results for Small Demand Set

	Total Tardiness						
	PBLS-R		**PBLS-NN**		**Random Walk**		**Optimal**
	Average	Best	Average	Best	Average	Best	Best
DS O1	252 (13.6%)	219 (0.76%)	**247 (13.3%)**	**218 (0.06%)**	303 (38.8%)	247 (13.3%)	218
DS O2	**160 (29.7%)**	**130 (5.4%)**	169 (36.9%)	131 (6.24%)	235 (43.5%)	177 (90.5%)	123

Both methods for generating the proxy measure returned similar overall results. The largest disparity in average solution quality between the two proxy fit methods was seen on DS 1, where PBLS-R out performed PBLS-NN by 24%. The neural network based proxy measure used in

PBLS-NN requires more training time and significantly more time for each proxy measure evaluation when compared to the regression based proxy measure used in PBLS-R. Given the similar objective function performance between the two methods, the additional processing time required by the neural network based proxy measure is not justified and PBLS-R is recommended.

6.6 Implementation of Solution Approach

The PBLS solution methods were developed and executed using Java 1.7 and the Eclipse (Luna 4.4.1) development environment. Standard Java libraries were used throughout the implementation. Regression modeling was done using the OLSMultipleLinearRegression class from the Apache Commons math library version 3.1 (Apache, 2015). To overcome shallow cloning issues in Java, Deep-Cloning Library version 1.8.5 was used (Cloning, 2012). Integration with MatLab for neural network training and evaluation was done through the MalabControl Java API (MatlabControl, 2013). IBM CPLEX v12.5 was used for generating optimal solutions. The final execution of runs for this dissertation were carried out across a number of different systems.

Parallel processing was only used when generating optimal solutions using CPLEX. Utilizing parallel processing could improve runtimes for both PBLS-R and PBLS-NN. Building the training set involves generating a number of independent schedules that are then used to train the proxy measure model. This generation could be easily handled using threading to reduce overall run time. Additionally, the local searches applied to a number of initial schedules could be conducted in parallel to further reduce processing time.

7. Conclusions

Within manufacturing systems, process planning and scheduling are two interrelated problems that are often treated independently. Process planning involves deciding which operations are required to produce a finished product and which resources should perform each operation. Scheduling involves deciding the sequence that operations should be processed by each resource, where process planning decisions are known *a priori*. Integrating process planning and scheduling offers significant opportunities to reduce bottlenecks and improve plant performance, particularly for complex job shops.

This research is motivated by the coating and laminating (C&L) system of a film manufacturing facility, where more than 1,000 product types are regularly produced monthly. The C&L system can be described as a complex job shop with sequence dependent setups, operation re-entry, minimum and maximum wait time constraints, and a due date based performance measure. In addition to the complex scheduling environment, products produced in the C&L system have multiple feasible process plans. The C&L system experiences significant issues with both schedule generation and due date performance. Thus, an integrated process planning and scheduling approach is needed to address large industry representative problems.

In this research, a novel proxy measure based local search (PBLS) approach is proposed to address the integrated process planning and scheduling for a complex job shop. PBLS uses a proxy measure in conjunction with local search procedures to adjust process planning decisions with the goal of reducing sum of total tardiness. A new dispatching heuristic, OU-MW, is developed to generate feasible schedules for complex job shop scheduling problems with maximum wait time constraints. Two local search methods are proposed to explore process and operation flexibility.

A proxy measure is developed to estimate the effect a process planning change has on the overall objective function of minimizing total tardiness. The proxy measure is used to guide local search decisions. Modified stepwise regression and back propagating neural network fitting are investigated as methods for generating the proxy measure used in PBLS-R and PLBS-NN respectively. In both cases, descriptive statistics about the active process planning set are used as independent variables in the proxy measure model. These descriptive statistics are easily determined without generating a detailed schedule. Using a proxy measure to guide the process planning decisions reduces the number of times a detailed schedule has to be generated and thus reduces overall runtime.

PBLS-R, PBLS-NN, and local search guided by random walk are applied to four industry representative integrated process planning and scheduling problems. In all cases, PBLS-R and PBLS-NN showed significantly better performance than the local search guided by random walk, with respect to a sum of total tardiness objective. The comparison to random walk provides statistically significant evidence that the proxy measure is capable of capturing the relationship between process planning decisions and objective function performance. Additionally, PBLS-R and PBLS-NN results were compared to results from an optimal solution approach for smaller demand sets. While the PBLS is designed specifically for large demand sets, the comparison to optimal demonstrates the ability of both PBLS-R and PBLS-NN to return good solutions to integrated process planning and scheduling problems for which optimal solutions are known. Run time for the optimal solution approach applied to small data sets of 9-13 jobs was between 112 minutes and 50 hours. Run times for PBLS-R and PBLS-NN are less than 1 hour even for large data sets of 98-200 jobs. PBLS-R provides comparable solution quality to PBLS-NN. For the implementation used in this study computational time for PBLS-NN is 30 times that of PBLS-R. Considering computational times and case study results, PBLS-R is the preferred approach explored for solving large scale IPPS problems in a job shop environment with sequence dependent setups, operation re-entry, minimum and maximum wait time constraints, and due date performance measure like that presented by the C&L problem.

7.1 Future Work

This research establishes the efficacy of a proxy based local search (PBLS) framework for generating good solutions to large scale integrated process planning and scheduling (IPPS) problems. Two different proxy measure approaches are proposed and the effects of a number of PBLS parameters on objective function performance are explored. The PBLS was applied to a complex job shop scheduling problem for which a dispatching heuristic, OU-MW, was developed for generating solutions to the scheduling sub-problem. The remainder of this section highlights five areas of future work that could contribute to improving the performance and applicability of PBLS.

7.1.1 Dispatching Heuristic for Job Shop Scheduling with Maximum Wait Time

The development of the OU-MW dispatching heuristic allowed for the generation of good solutions to the job shop scheduling problem with sequence dependent setups, operation re-entry, minimum and maximum wait times, and a due date based objective function. The heuristic is capable of solving problems with more than one maximum wait time constraint per job, as long as there is no more than one successive maximum wait time constraint on a given job. The development of a quick dispatching heuristic for the generalized job shop scheduling problem with any number of maximum wait time constraints remains an open research question.

7.1.2 Method for Informing Parameter Selection for PBLS

The application of PBLS-R and PBLS-NN is influenced by a number of parameters as discussed in 6.2. In the case of training size, N^T, and search depth, S^D, objective function performance increased at the expense of processing time. Developing an analytic approach to access the tradeoff between PBLS parameters, processing time, and objective function performance would provide guidance to practitioners seeking to apply PBLS to real world IPPS problems.

7.1.3 Increasing Training Size for Proxy Measure

PBLS relies on the proxy measure to accurately predict the effects of process planning changes on objective function performance. For both PBLS-R and PBLS-NN, increasing the training size improved overall objective function performance. Four training sizes from 100 to 400 were explored as a part of this research. The effect of training sizes above 400 remains an interesting direction for future work.

Generating detailed schedules for training the proxy measure can be costly. It is possible that including previously generated detailed schedules for demand sets with similar dynamics may improve the performance of PBLS with minimal impact to processing time. Exploring the effect of incorporating the results from similar demand sets when training the proxy measure on PBLS performance could be the focus of future work.

7.1.4 Improving Proxy Measure Generation

PBLS relies on the proxy measure to accurately predict the effects that process planning changes have on objective function performance. The more accurate the proxy measure, the more effective the PBLS approach will be.

For PBLS-R, the independent variables used in the regression based proxy measure are determined using a modified stepwise regression. Alternative approaches, such as Lasso, exist for solving the variable selection problem (Osborne *et al.*, 2000). The effect of these variable selection methods on PBLS-R performance is an interesting direction for future work.

For PBLS-NN, a neural network based proxy measure is trained and evaluated using the Neural Network Toolbox in Matlab. The number of hidden layers used in the network was shown to have a significant effect on PBLS-NN performance. In addition to the number of hidden layers, adjustments to the equations used at each node and the node-arch structure can be incorporated. Further exploration of the effect of neural network architecture on PBLS-NN performance is an interesting direction for future work.

For the PBLS framework presented in this dissertation, a single proxy measure is generated from a training set with size N^T. The proxy measure is then applied repeatedly to estimate the effects of process planning changes on objective function performance. Throughout the PBLS an additional $2N^S$ detailed schedules are generated, but the detailed process planning statistics and observed objective function value from these schedules are not reflected by the proxy measure. An adaptive proxy based local search could use the results from each detailed schedule generated during the search to continually improve the proxy measure. Adaptive proxy based local search presents an interesting area of future research.

7.1.5 Descriptive Process Planning Statistics

The regression model and neural network model used in PBLS-R and PBLS-NN rely on descriptive process planning statistics as inputs to estimate the sum of total tardiness in the scheduling sub-problem. The ability to calculate meaningful descriptive statistics without the generation of a detailed schedule is key to the proposed PBLS. The base process planning statistics used in this work are processing time on a given machine, number of operations on a given machine, total processing time, total operations, the standard deviation of processing time across machines, and the standard deviation of operation count across machines. A subset of these parameters is selected to build the proxy measures used in PBLS. The use of other easy to calculate descriptive statistics, such as number of unique setup groups per machine, could be considered for future work. Additionally, examining the descriptive process planning statistics used in a proxy measure may provide insight into plant floor dynamics not otherwise apparent.

8. References

Adams, J., E. Balas and D. Zawack (1988). "The Shifting Bottleneck Procedure for Job Shop Scheduling." Management Science **34**(3): 391-401.

Akyol, D. E. and G. M. Bayhan (2007). "A Review on Evolution of Production Scheduling with Neural Networks." Computers & Industrial Engineering **53**(1): 95-122.

Allahverdi, A., C. Ng, T. Cheng and M. Y. Kovalyov (2008). "A Survey of Scheduling Problems with Setup Times or Costs." European Journal of Operational Research **187**(3): 985-1032.

Anderson, E. J., C. A. Glass and C. N. Potts (1997). Machine Scheduling. Local Search Algorithms in Combinatorial Optimization. E. H. L. Aarts and J. K. Lenstra, Wiley: 361-414.

Apache. (2015). "Class Olsmultiplelinearregression." From http://commons.apache.org/proper/commons-math/apidocs/org/apache/commons/math3/stat/regression/OLSMultipleLinearRegression.html.

Artigues, C., S. Belmokhtar and D. Feillet (2004). "A New Exact Solution Algorithm for the Job Shop Problem with Sequence-Dependent Setup Times." Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems 37-49.

Balas, E. (1969). "Machine Sequencing Via Disjunctive Graphs: An Implicit Enumeration Algorithm." Operations Research **17**(6): 941-957.

Balas, E. and N. Simonetti (2001). "Linear Time Dynamic-Programming Algorithms for New Classes of Restricted Tsps: A Computational Study." INFORMS Journal on Computing **13**(1): 56-75.

Balas, E., N. Simonetti and A. Vazacopoulos (2008). "Job Shop Scheduling with Setup Times, Deadlines and Precedence Constraints." Journal of Scheduling **11**(4): 253-262.

Barnes, J. W. and J. B. Chambers (1995). "Solving the Job Shop Scheduling Problem with Tabu Search." IIE Transactions **27**(2): 257-263.

Benjaafar, S. and R. Ramakrishnan (1996). "Modelling, Measurement and Evaluation of Sequencing Flexibility in Manufacturing Systems." International Journal of Production Research **34**(5): 1195-1220.

Binsse, H. L. (1887). "A Short Way to Keep Time and Cost." Transactions of the American Society of Mechanical Engineers **9**: 380-386.

Bishop, C. M. (2006). Pattern Recognition and Machine Learning, Springer New York.

Blazewicz, J., W. Domschke and E. Pesch (2003). "The Job Shop Scheduling Problem: Conventional and New Solution Techniques." European Journal of Operational Research **93**(1): 1-33.

Brucker, P., B. Jurisch and B. Sievers (1994). "A Branch and Bound Algorithm for the Job-Shop Scheduling Problem." Discrete Applied Mathematics **49**(1–3): 107-127.

Brucker, P. and O. Thiele (1996). "A Branch and Bound Method for the General-Shop Problem with Sequence Dependent Setup-Times." OR Spectrum **18**(3): 145-161.

Cai, L., Q. Wu and Z. Yong (2000). A Genetic Algorithm with Local Search for Solving Job Problems. Real-World Applications of Evolutionary Computing 363-365.

Candido, M., S. Khator and R. Barcia (1998). "A Genetic Algorithm Based Procedure for More Realistic Job Shop Scheduling Problems." International Journal of Production Research **36**(12): 3437-3457.

Carlier, J. (1982). "The One-Machine Sequencing Problem." European Journal of Operational Research **11**(1): 42-47.

Cheng, R., M. Gen and Y. Tsujimura (1996). "A Tutorial Survey of Job-Shop Scheduling Problems Using Genetic Algorithms—I. Representation." Computers & Industrial Engineering **30**(4): 983-997.

Cheung, W. and H. Zhou (2001). "Using Genetic Algorithms and Heuristics for Job Shop Scheduling with Sequence-Dependent Setup Times." Annals of Operations Research **107**(1): 65-81.

Chryssolouris, G., S. Chan and W. Cobb (1984). "Decision Making on the Factory Floor: An Integrated Approach to Process Planning and Scheduling." Robotics and Computer-Integrated Manufacturing **1**(3–4): 315-319.

Chryssolouris, G., S. Chan and N. P. Suh (1985). "An Integrated Approach to Process Planning and Scheduling." CIRP Annals - Manufacturing Technology **34**(1): 413-417.

Cloning. (2012). "Java Deep-Cloning Library." from https://github.com/kostaskougios/cloning.

Deb, S., K. Ghosh and S. Paul (2006). "A Neural Network Based Methodology for Machining Operations Selection in Computer-Aided Process Planning for Rotationally Symmetrical Parts." Journal of Intelligent Manufacturing **17**(5): 557-569.

Dell'Amico, M. and M. Trubian (1993). "Applying Tabu Search to the Job-Shop Scheduling Problem." Annals of Operations Research **41**(3): 231-252.

Ding, L., Y. Yue*, K. Ahmet, M. Jackson and R. Parkin (2005). "Global Optimization of a Feature-Based Process Sequence Using Ga and Ann Techniques." International Journal of Production Research **43**(15): 3247-3272.

Dorndorf, U. and E. Pesch (1995). "Evolution Based Learning in a Job Shop Scheduling Environment." Computers & Operations Research **22**(1): 25-40.

Du, J. and J. Y.-T. Leung (1990). "Minimizing Total Tardiness on One Machine Is NP-Hard." Mathematics of operations research **15**(3): 483-495.

El-Bouri, A. and P. Shah (2006). "A Neural Network for Dispatching Rule Selection in a Job Shop." The International Journal of Advanced Manufacturing Technology **31**(3-4): 342-349.

Ellis, K., Y. Lu and E. Bish (2004). "Scheduling of Wafer Test Processes in Semiconductor Manufacturing." International Journal of Production Research **42**(2): 215-242.

Falkenauer, E. and S. Bouffouix (1991). A Genetic Algorithm for Job Shop. International Conference on Robotics and Automation, Sacramento, California, USA, IEEE.

Fujii, N., R. Inoue and K. Ueda (2008). Integration of Process Planning and Scheduling Using Multi-Agent Learning. Manufacturing Systems and Technologies for the New Frontier. M. Mitsuishi, K. Ueda and F. Kimura, Springer London: 297-300.

Gantt, H. L. (1913). Work, Wages, and Profits, Engineering Magazine Co.

Giffler, B. and G. L. Thompson (1960). "Algorithms for Solving Production-Scheduling Problems." Operations Research **8**(4): 487-503.

González, M., C. Vela and R. Varela (2010). A Tabu Search Algorithm to Minimize Lateness in Scheduling Problems with Setup Times. Current Topics in Artificial Intelligence: 212-221.

Graham, R. L., E. L. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan (1979). "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey." Annals of Discrete Mathematics **5**(2): 287-326.

Gu, P., S. Balasubramanian and D. H. Norrie (1997). "Bidding-Based Process Planning and Scheduling in a Multi-Agent System." Computers & Industrial Engineering **32**(2): 477-496.

Guo, Y. W., W. D. Li, A. R. Mileham and G. W. Owen (2009). "Applications of Particle Swarm Optimisation in Integrated Process Planning and Scheduling." Robotics and Computer-Integrated Manufacturing **25**(2): 280-288.

Gupta, A. K. and A. I. Sivakumar (2006). "Job Shop Scheduling Techniques in Semiconductor Manufacturing." The International Journal of Advanced Manufacturing Technology **27**(11): 1163-1169.

Hart, E., P. Ross and D. Corne (2005). "Evolutionary Scheduling: A Review." Genetic Programming and Evolvable Machines **6**(2): 191-220.

Hax, A. C. and H. C. Meal (1975). Hierarchical Integration of Production Planning and Scheduling. Studies in the Management Sciences Amsterdam: 53-69.

He, Z., T. Yang and A. Tiger (1996). "An Exchange Heuristic Imbedded with Simulated Annealing for Due-Dates Job-Shop Scheduling." European Journal of Operational Research **91**(1): 99-117.

Ho, Y. C. and C. L. Moodie (1996). "Solving Cell Formation Problems in a Manufacturing Environment with Flexible Processing." International Journal of Production Research **34**(10): 2901.

Hodgson, T. J., D. Cormier, A. J. Weintraub and A. Zozom (1998). "Note. Satisfying Due Dates in Large Job Shops." Management Science **44**(10): 1442-1446.

Holland, J. (1975). Adaptation in Artificial and Natural Systems. University of Michigan Press.

Holtsclaw, H. H. and R. Uzsoy (1996). "Machine Criticality Measures and Subproblem Solution Procedures in Shifting Bottleneck Methods: A Computational Study." The Journal of the Operational Research Society **47**(5): 666-677.

Hopp, W. J. and M. L. Spearman (2008). Factory Physics, McGraw-Hill/Irwin New York.

Iwata, K., Y. Murotsu, F. Oba and T. Uemura (1978). "Optimization of Selection of Machine Tools, Loading Sequence of Parts and Machining Conditions in Job-Shop Type Machining Systems." Annals of the CIRP **27**(1): 447-451.

Jain, A. S. and S. Meeran (1999). "Deterministic Job-Shop Scheduling: Past, Present and Future." European Journal of Operational Research **113**(2): 390-434.

Jones, A., L. C. Rabelo and A. T. Sharawi (1999). "Survey of Job Shop Scheduling Techniques." Wiley Encyclopedia of Electrical and Electronics Engineering.

Kennedy, J. (2010). Particle Swarm Optimization. Encyclopedia of Machine Learning, Springer New York: 760-766.

Kim, K. H. and P. J. Egbelu (1999). "Scheduling in a Production Environment with Multiple Process Plans Per Job." International Journal of Production Research **37**(12): 2725-2753.

Kim, Y. K., K. Park and J. Ko (2003). "A Symbiotic Evolutionary Algorithm for the Integration of Process Planning and Job Shop Scheduling." Computers & Operations Research **30**(8): 1151-1171.

Kis, T. (2003). "Job-Shop Scheduling with Processing Alternatives." European Journal of Operational Research **151**(2): 307-332.

Kolonko, M. (1999). "Some New Results on Simulated Annealing Applied to the Job Shop Scheduling Problem." European Journal of Operational Research **113**(1): 123-136.

Lee, Y. H., K. Bhaskaran and M. Pinedo (1997). "A Heuristic to Minimize the Total Weighted Tardiness with Sequence-Dependent Setups." IIE Transactions **29**(1): 45-52.

Lee, Y. H. and M. Pinedo (1997). "Scheduling Jobs on Parallel Machines with Sequence-Dependent Setup Times." European Journal of Operational Research **100**(3): 464-474.

Lenstra, J. and A. Rinnooy Kan (1979). "Computational Complexity of Discrete Optimization Problems." Annals of Discrete Mathematics **4**: 121-140.

Lenstra, J. K., A. Rinnooy Kan and P. Brucker (1977). "Complexity of Machine Scheduling Problems." Annals of Discrete Mathematics **1**: 343-362.

Li, W. D. and C. A. McMahon (2007). "A Simulated Annealing-Based Optimization Approach for Integrated Process Planning and Scheduling." International Journal of Computer Integrated Manufacturing **20**(1): 80-95.

Li, X., L. Gao, C. Zhang and X. Shao (2010). "A Review on Integrated Process Planning and Scheduling." International Journal of Manufacturing Research **5**(2): 161-180.

Lim, M. K. and D. Z. Zhang (2004). "An Integrated Agent-Based Approach for Responsive Control of Manufacturing Resources." Computers & Industrial Engineering **46**(2): 221-232.

Lin, G. Y.-J. and J. J. Solberg (1991). "Effectiveness of Flexible Routing Control." International Journal of Flexible Manufacturing Systems **3**(3): 189-211.

Manikas, A. and Y.-L. Chang (2009). "Multi-Criteria Sequence-Dependent Job Shop Scheduling Using Genetic Algorithms." Computers & Industrial Engineering **56**(1): 179-185.

Manikas, A. and Y.-L. Chang (2009). "A Scatter Search Approach to Sequence-Dependent Setup Times Job Shop Scheduling." International Journal of Production Research **47**(18): 5217-5236.

Mason, S. J., J. W. Fowler and W. Matthew Carlyle (2002). "A Modified Shifting Bottleneck Heuristic for Minimizing Total Weighted Tardiness in Complex Job Shops." Journal of Scheduling **5**(3): 247-262.

Mathworks. (2015). “Neural Network Toolbox.” from http://www.mathworks.com/products/neural-network/index.html

MatlabControl. (2013). "A Java Api to Interact with Matlab." From https://code.google.com/p/matlabcontrol/downloads/detail?name=matlabcontrol-4.1.0.jar&can=2&q=.

Mattfeld, D. C. and C. Bierwirth (2004). "An Efficient Genetic Algorithm for Job Shop Scheduling with Tardiness Objectives." European Journal of Operational Research **155**(3): 616-630.

Naderi, B., S. M. T. F. Ghomi and M. Aminnayeri (2010). "A High Performing Metaheuristic for Job Shop Scheduling with Sequence-Dependent Setup Times." Applied Soft Computing **10**(3): 703-710.

Nwana, H. and D. Ndumu (1997). An Introduction to Agent Technology Software Agents and Soft Computing Towards Enhancing Machine Intelligence. H. Nwana and N. Azarmi, Springer Berlin **1198:** 1-26.

Oddi, A., R. Rasconi, A. Cesta and S. F. Smith (2011). "Solving Job Shop Scheduling with Setup Times through Constraint-Based Iterative Sampling: An Experimental Analysis." Annals of Mathematics and Artificial Intelligence **62**(3-4): 371-402.

Oracle. (2015). "Java Platform, Standard Edition Hotspot Virtual Machine Garbage Collection Tuning Guide." Retrieved 6/6/2015, from http://docs.oracle.com/javase/8/docs/technotes/guides/vm/gctuning/index.html.

Osborne, M. R., B. Presnell and B. A. Turlach (2000). "A New Approach to Variable Selection in Least Squares Problems." IMA Journal of Numerical Analysis-Institute of Mathematics and its Applications **20**(3): 389-404.

Ovacik, I. M. and R. Uzsoy (1992). "A Shifting Bottleneck Algorithm for Scheduling Semiconductor Testing Operations." Journal of Electronics Manufacturing **2**(3): 119-134.

Ovacik, I. M. and R. Uzsoy (1994). "Exploiting Shop Floor Status Information to Schedule Complex Job Shops." Journal of manufacturing systems **13**(2): 73-84.

Ovacik, I. M. and R. Uzsoy (1995). "Rolling Horizon Procedures for Dynamic Parallel Machine Scheduling with Sequence-Dependent Setup Times." International Journal of Production Research **33**(11): 3173-3192.

Ovacik, I. M. and R. Uzsoy (1996). "Decomposition Methods for Scheduling Semiconductor Testing Facilities." International Journal of Flexible Manufacturing Systems **8**(4): 357-387.

Ovacik, I. M. and R. Uzsoy (1994). "Rolling Horizon Algorithms for a Single-Machine Dynamic Scheduling Problem with Sequence-Dependent Setup Times." The International Journal of Production Research **32**(6): 1243-1263.

Palmer, G. J. (1996). "A Simulated Annealing Approach to Integrated Production Scheduling." Journal of Intelligent Manufacturing **7**(3): 163-176.

Panwalkar, S., R. Dudek and M. Smith (1973). Sequencing Research and the Industrial Scheduling Problem. Symposium on the Theory of Scheduling and its Applications, Springer.

Pezzella, F. and E. Merelli (2000). "A Tabu Search Method Guided by Shifting Bottleneck for the Job Shop Scheduling Problem." European Journal of Operational Research **120**(2): 297-310.

Pfund, M., J. W. Fowler, A. Gadkari and Y. Chen (2008). "Scheduling Jobs on Parallel Machines with Setup Times and Ready Times." Computers & Industrial Engineering **54**(4): 764-782.

Pfund, M. E., H. Balasubramanian, J. W. Fowler, S. J. Mason and O. Rose (2008). "A Multi-Criteria Approach for Scheduling Semiconductor Wafer Fabrication Facilities." Journal of Scheduling **11**(1): 29-47.

Pinedo, M. (2012). Scheduling: Theory, Algorithms, and Systems, Springer New York.

Pinedo, M. and X. Chao (1999). Operations Scheduling with Applications in Manufacturing and Servies, McGraw Hill New York.

Pinedo, M. and M. Singer (1999). "A Shifting Bottleneck Heuristic for Minimizing the Total Weighted Tardiness in a Job Shop." Naval Research Logistics (NRL) **46**(1): 1-17.

Rachamadugu, R. and T. Morton (1982). Myopic Heuristics for the Single Machine Weighted Tardiness Problem. Working Paper, Carnigie Mellon University.

Rossi, A. and G. Dini (2007). "Flexible Job-Shop Scheduling with Routing Flexibility and Separable Setup Times Using Ant Colony Optimisation Method." Robotics and Computer-Integrated Manufacturing **23**(5): 503-516.

Schmidt, K. (2001). Using Tabu Search to Solve the Job Shop Scheduling Problem with Sequence Dependent Setups Masters Thesis, Brown University.

Shukla, S., M. Tiwari and Y. Son (2008). "Bidding-Based Multi-Agent System for Integrated Process Planning and Scheduling: A Data-Mining and Hybrid Tabu-Sa Algorithm-Oriented Approach." The International Journal of Advanced Manufacturing Technology **38**(1): 163-175.

Singer, M. and M. Pinedo (1998). "A Computational Study of Branch and Bound Techniques for Minimizing the Total Weighted Tardiness in Job Shops." IIE Transactions **30**(2): 109-118.

Sun, X. and J. S. Noble (1999). "An Approach to Job Shop Scheduling with Sequence-Dependent Setups." Journal of Manufacturing Systems **18**(6): 416-430.

Tan, W. and B. Khoshnevis (2000). "Integration of Process Planning and Scheduling—a Review." Journal of Intelligent Manufacturing **11**(1): 51-63.

Uzsoy, R., C.-Y. Lee and L. A. Martin-Vega (1994). "A Review of Production Planning and Scheduling Models in the Semiconductor Industry Part II: Shop-Floor Control." IIE Transactions **26**(5): 44-55.

Vaessens, R. J. M., E. H. L. Aarts and J. K. Lenstra (1996). "Job Shop Scheduling by Local Search." INFORMS Journal on Computing **8**(3): 302-317.

Valls, V., M. Angeles Perez and M. Sacramento Quintanilla (1998). "A Tabu Search Approach to Machine Scheduling." European Journal of Operational Research **106**(2): 277-300.

Van Laarhoven, P. J. M., E. H. L. Aarts and J. K. Lenstra (1992). "Job Shop Scheduling by Simulated Annealing." Operations Research **40**(1): 113-125.

Vázquez, M. and D. Whitley (2000). A Comparison of Genetic Algorithms for the Static Job Shop Scheduling Problem. Parallel Problem Solving from Nature PPSN VI. M. Schoenauer, K. Deb, G. Rudolph et al., Springer Berlin / Heidelberg. **1917:** 303-312.

Vepsalainen, A. P. J. and T. E. Morton (1987). "Priority Rules for Job Shops with Weighted Tardiness Costs." Management Science **33**(8): 1035-1047.

Wang, L. and W. Shen (2003). "Dpp: An Agent-Based Approach for Distributed Process Planning." Journal of Intelligent Manufacturing **14**(5): 429-439.

Wang, L. and M. Wang (1997). A Hybrid Algorithm for Earliness-Tardiness Scheduling Problem with Sequence Dependent Setup Time. Proceddings of the 36th IEEE Conference on Decision and Control, San Diego, California.

Watson, J.-P., J. C. Beck, A. E. Howe and L. D. Whitley (2003). "Problem Difficulty for Tabu Search in Job-Shop Scheduling." Artificial Intelligence **143**(2): 189-217.

Weiming, S., W. Lihui and H. Qi (2006). "Agent-Based Distributed Manufacturing Process Planning and Scheduling: A State-of-the-Art Survey." IEEE Transaction on Systems, Man, and Cybernetics **36**(4): 563-577.

Weintraub, A., D. Cormier, T. Hodgson, R. King, J. Wilson and A. Zozom (1999). "Scheduling with Alternatives: A Link between Process Planning and Scheduling." IIE Transactions **31**(11): 1093-1102.

Wilbrecht, J. K. and W. B. Prescott (1969). "The Influence of Setup Time on Job Shop Performance." Management Science **16**(4): B-274-B-280.

Xu, X., L. Wang and S. T. Newman (2011). "Computer-Aided Process Planning–a Critical Review of Recent Developments and Future Trends." International Journal of Computer Integrated Manufacturing **24**(1): 1-31.

Zhang, W. and S. Xie (2007). "Agent Technology for Collaborative Process Planning: A Review." The International Journal of Advanced Manufacturing Technology **32**(3): 315-325.

Zhou, H., W. Cheung and L. C. Leung (2009). "Minimizing Weighted Tardiness of Job-Shop Scheduling Using a Hybrid Genetic Algorithm." European Journal of Operational Research **194**(3): 637-649.

A. Appendix

A.1 Solutions for Demand Set O1

A.1.1 Optimal Solution for DS O1

Table A.1 Job View of Optimal Solution for DS 01

Job	Op	Machine	SG	ST	CT	Due	Tardy
1	1	E	S3	6.23	16.43		
1	2	H	S5	24.43	33.96		
1	3	Z	S19	33.96	37.49	8	29.49
2	1	A	S1	11.36	29.04		
2	2	G	S10	37.04	56.09	29	27.09
3	1	F	S4	0.00	12.52		
3	2	Z	S19	12.52	15.61	23	0.00
4	1	E	S2	18.93	28.05		
4	2	C	S15	36.05	54.29		
4	3	B	S6	54.29	63.65		
4	4	Z	S19	63.65	66.47	41	25.47
5	1	G	S18	9.56	23.81		
5	2	D	S9	25.48	50.55		
5	3	D	S9	65.30	86.96	7	79.96
6	1	E	S1	0.00	3.73		
6	2	F	S10	20.52	25.73	20	5.73
7	1	H	S4	0.00	18.73		
7	2	Z	S19	18.73	25.67	30	0.00
8	1	A	S2	0.00	8.86		
8	2	C	S15	16.86	34.58		
8	3	B	S6	34.58	43.66		
8	4	Z	S19	43.66	46.40	30	16.40
9	1	G	S18	0.00	9.06		
9	2	D	S9	9.06	24.98		
9	3	D	S9	51.05	64.80	31	33.80
						Total	217.95

Table A.2 Machine View of Optimal Solution for DS O1

Machine	Job	Op	SG	Setup	ST	CT
A	8	1	S2		0.00	8.86
A	2	1	S1	2.8	11.36	29.04
B	8	3	S6		34.58	43.66
B	4	3	S6	0.5	54.29	63.65
C	8	2	S15		16.86	34.58
C	4	2	S15	0.5	36.05	54.29
D	9	2	S9		9.06	24.98
D	5	2	S9	0.5	25.48	50.55
D	9	3	S9	0.5	51.05	64.80
D	5	3	S9	0.5	65.30	86.96
E	6	1	S1		0.00	3.73
E	1	1	S3	2.5	6.23	16.43
E	4	1	S2	2.5	18.93	28.05
F	3	1	S4		0.00	12.52
F	6	2	S10	8	20.52	25.73
G	9	1	S18		0.00	9.06
G	5	1	S18	0.5	9.56	23.81
G	2	2	S10	2	37.04	56.09
H	7	1	S4		0.00	18.73
H	1	2	S5	3	24.43	33.96
Z	3	2	S19		12.52	15.61
Z	7	2	S19	0.5	18.73	25.67
Z	1	3	S19	0.5	33.96	37.49
Z	8	4	S19	0.5	43.66	46.40
Z	4	4	S19	0.5	63.65	66.47

A.1.2 Proxy Based Local Search with Regression for DS O1

Table A.3 Job View of Best Solution for DS O1: PBLS-R

Job	Op	Machine	SG	ST	CT	MinWait	MaxWait	Due	Tardy
1	0	E	S1	0.00	10.21	8			
1	1	G	S5	18.21	29.20				
1	2	Z	S19	29.75	33.28			8	25.28
2	3	E	S1	12.71	30.39	8			
2	4	H	S10	38.39	57.44			29	28.44
3	5	C	S18	0.00	13.99		12		
3	6	B	S6	13.99	24.24				
3	7	Z	S19	26.17	29.25			23	6.25
4	8	A	S2	0.00	9.12	8			
4	9	C	S15	21.99	40.23				
4	10	B	S6	40.23	49.59				
4	11	Z	S19	49.59	52.41			41	11.41
5	12	F	S18	0.00	28.51		12		
5	13	D	S9	39.74	64.80				
5	14	D	S9	65.30	86.96			7	79.96
6	15	A	S1	11.62	15.34	8			
6	16	H	S10	26.73	30.74			20	10.74
7	17	H	S4	0.00	18.73				
7	18	Z	S19	18.73	25.67			30	0.00
8	19	E	S2	32.89	41.75	8			
8	20	C	S15	49.75	67.47				
8	21	B	S6	67.47	76.55				
8	22	Z	S19	76.55	79.29			30	49.29
9	23	G	S18	0.00	9.06		12		
9	24	D	S9	9.06	24.98				
9	25	D	S9	25.48	39.24			31	8.24
								Total	219.62

Table A.4 Machine View of Best Solution for DS O1: PBLS-R

Machine	Job	Op	Machine	SG	Setup	ST	CT
A	4	1	A	S2		0.00	9.12
A	6	1	A	S1	2.5	11.62	15.34
B	3	2	B	S6		13.99	24.24
B	4	3	B	S6	0.5	40.23	49.59
B	8	3	B	S6	0.5	67.47	76.55
C	3	1	C	S18		0.00	13.99
C	4	2	C	S15	8	21.99	40.23
C	8	2	C	S15	0.5	49.75	67.47
D	9	2	D	S9		9.06	24.98
D	9	3	D	S9	0.5	25.48	39.24
D	5	2	D	S9	0.5	39.74	64.80
D	5	3	D	S9	0.5	65.30	86.96
E	1	1	E	S1		0.00	10.21
E	2	1	E	S1	0.5	12.71	30.39
E	8	1	E	S2	2.5	32.89	41.75
F	5	1	F	S18		0.00	28.51
G	9	1	G	S18		0.00	9.06
G	1	2	G	S5	2	18.21	29.20
H	7	1	H	S4		0.00	18.73
H	6	2	H	S10	8	26.73	30.74
H	2	2	H	S10	0.5	38.39	57.44
Z	7	2	Z	S19		18.73	25.67
Z	3	3	Z	S19	0.5	26.17	29.25
Z	1	3	Z	S19	0.5	29.75	33.28
Z	4	4	Z	S19	0.5	49.59	52.41
Z	8	4	Z	S19	0.5	76.55	79.29

A.1.3 Proxy Based Local Search with Neural Network for DS O1

Table A.5 Job View of Best Solution for DS O1: PBLS-NN

Job	Op	Machine	SG	ST	CT	MinWait	MaxWait	Due	Tardy
1	1	E	S3	0.00	10.21	8		8	
1	2	G	S5	23.74	34.74			8	
1	3	Z	S19	35.01	38.54			8	30.54
2	1	E	S1	12.71	30.39	8		29	
2	2	G	S10	42.74	61.78			29	32.78
3	1	H	S4	0.00	8.34			23	
3	2	Z	S19	8.34	11.43			23	0.00
4	1	E	S2	32.89	42.01	8		41	
4	2	C	S15	50.01	68.25			41	
4	3	B	S6	68.25	77.61			41	
4	4	Z	S19	77.61	80.43			41	39.43
5	1	F	S18	0.00	28.51		12	7	
5	2	D	S9	39.74	64.80			7	
5	3	D	S9	65.30	86.96			7	79.96
6	1	A	S1	0.00	3.73	8		20	
6	2	G	S10	11.73	15.74			20	0.00
7	1	H	S4	8.84	27.58			30	
7	2	Z	S19	27.58	34.51			30	4.51
8	1	A	S2	6.23	15.08	8		30	
8	2	C	S15	23.08	40.80			30	
8	3	B	S6	40.80	49.89			30	
8	4	Z	S19	49.89	52.63			30	22.63
9	1	G	S18	0.00	9.06		12	31	
9	2	D	S9	9.06	24.98			31	
9	3	D	S9	25.48	39.24			31	8.24
								Total	218.09

Table A.6 Machine View Best Solution for DS O1: PBLS-NN

Machine	Job	Op	SG	Setup	ST	CT
A	6	1	S1		0.00	3.73
A	8	1	S2	2.5	6.23	15.08
B	8	3	S6		40.80	49.89
B	4	3	S6	0.5	68.25	77.61
C	8	2	S15		23.08	40.80
C	4	2	S15	0.5	50.01	68.25
D	9	2	S9		9.06	24.98
D	9	3	S9	0.5	25.48	39.24
D	5	2	S9	0.5	39.74	64.80
D	5	3	S9	0.5	65.30	86.96
E	1	1	S3		0.00	10.21
E	2	1	S1	2.5	12.71	30.39
E	4	1	S2	2.5	32.89	42.01
F	5	1	S18		0.00	28.51
G	9	1	S18		0.00	9.06
G	6	2	S10	2	11.73	15.74
G	1	2	S5	8	23.74	34.74
G	2	2	S10	2	42.74	61.78
H	3	1	S4		0.00	8.34
H	7	1	S4	0.5	8.84	27.58
Z	3	2	S19		8.34	11.43
Z	7	2	S19	0.5	27.58	34.51
Z	1	3	S19	0.5	35.01	38.54
Z	8	4	S19	0.5	49.89	52.63
Z	4	4	S19	0.5	77.61	80.43

A.2 Solutions for Demand Set O2

A.2.1 Optimal Solution for DS O2

Table A.7 Job View Optimal Solution for DS O2

Job	Op	Machine	SG	ST	CT	MinWait	MaxWait	Due	Tardy
1	1	C	S17	0.00	16.04		12		
1	2	B	S6	16.04	27.79				
1	3	Z	S19	30.32	33.86			21	12.86
2	1	A	S2	0.00	9.57	8			
2	2	H	S4	17.57	26.51				
2	3	Z	S119	26.51	29.82			15	14.82
3	1	G	S4	0.00	5.01				
3	2	Z	S19	23.03	24.64			34	0.00
4	1	E	S3	7.04	18.90				
4	2	G	S5	26.90	39.67				
4	3	Z	S19	44.38	48.48			39	9.48
5	1	F	S4	9.59	11.77				
5	2	Z	S19	12.16	12.70			21	0.00
6	1	F	S4	0.00	6.77				
6	2	Z	S19	8.01	9.69			48	0.00
7	1	A	S2	10.07	20.50	8			
7	2	H	S4	28.50	38.23				
7	3	Z	S119	40.28	43.88			18	25.88
8	1	E	S2	0.00	4.54	8			
8	2	G	S4	12.54	17.43				
8	3	Z	S119	18.43	20.00			20	0.00
9	1	F	S4	12.27	20.50				
9	2	Z	S19	20.50	22.53			35	0.00
10	1	C	S17	16.54	28.44		12		
10	2	B	S6	28.44	37.15				
10	3	Z	S19	37.15	39.78			31	8.78
11	1	G	S4	5.58	10.19				
11	2	Z	S19	10.19	11.66			33	0.00
12	1	C	S17	28.94	51.52		12		
12	2	B	S6	51.52	68.07				
12	3	Z	S19	68.07	73.05			18	55.05
13	1	H	S4	0.00	12.77				
13	2	Z	S19	13.20	17.93			34	0.00
								Total	126.87

Table A.8 Machine View Optimal Solution for DS O2

Machine	Job	Op	SG	Setup	ST	CT
A	2	1	S2		0.00	9.57
A	7	1	S2	0.5	10.07	20.50
B	1	2	S6	8	16.04	27.79
B	10	2	S6	0.5	28.44	37.15
B	12	2	S6	0.5	51.52	68.07
C	1	1	S17		0.00	16.04
C	10	1	S17	0.5	16.54	28.44
C	12	1	S17	0.5	28.94	51.52
E	8	1	S2		0.00	4.54
E	4	1	S3	2.5	7.04	18.90
F	6	1	S4		0.00	6.77
F	5	1	S4	0.5	9.59	11.77
F	9	1	S4	0.5	12.27	20.50
G	3	1	S4	0.5	0.00	5.01
G	11	1	S4	0.5	5.58	10.19
G	8	2	S4	0.5	12.54	17.43
G	4	2	S5	3	26.90	39.67
H	13	1	S4		0.00	12.77
H	2	2	S4	0.5	17.57	26.51
H	7	2	S4	0.5	28.50	38.23
Z	6	2	S19		8.01	9.69
Z	11	2	S19	0.5	10.19	11.66
Z	5	2	S19	0.5	12.16	12.70
Z	13	2	S19	0.5	13.20	17.93
Z	8	3	S19	0.5	18.43	20.00
Z	9	2	S19	0.5	20.50	22.53
Z	3	2	S19	0.5	23.03	24.64
Z	2	3	S19	0.5	26.51	29.82
Z	1	3	S19	0.5	30.32	33.86
Z	10	3	S19	0.5	37.15	39.78
Z	7	3	S19	0.5	40.28	43.88
Z	4	3	S19	0.5	44.38	48.48
Z	12	3	S19	0.5	68.07	73.05

A.2.2 Proxy Based Local Search with Regression for DS O2

Table A.9 Job View Best Solution for DS O: PBLS-R

Job	Op	Machine	SG	ST	CT	MinWait	MaxWait	Due	Tardy
1	1	C	S17	0.00	16.04	0	12		
1	2	B	S6	16.04	27.79				
1	3	Z	S19	28.23	31.77			21	10.77
2	1	A	S2	0.00	9.57	8			
2	2	H	S4	17.57	26.51				
2	3	Z	S19	32.27	35.58			15	20.58
3	1	G	S4	0.00	5.01				
3	2	Z	S19	5.01	6.62			34	0
4	1	E	S3	7.04	18.90	8			
4	2	H	S5	29.51	40.58				
4	3	Z	S19	44.38	48.48			39	9.48
5	1	F	S4	0.00	2.18				
5	2	Z	S19	2.18	2.72			21	0
6	1	F	S4	17.90	24.66				
6	2	Z	S19	26.06	27.73			48	0
7	1	A	S2	10.07	20.50	8			
7	2	G	S4	28.50	39.73				
7	3	Z	S19	40.28	43.88			18	25.88
8	1	E	S2	0.00	4.54	8			
8	2	H	S4	12.54	16.78				
8	3	Z	S19	16.78	18.35			20	0
9	1	F	S4	2.68	10.91				
9	2	Z	S19	10.91	12.95			35	0
10	1	C	S17	16.54	28.44		12		
10	2	B	S6	28.44	37.15				
10	3	Z	S19	37.15	39.78			31	8.78
11	1	F	S4	11.41	17.40				
11	2	Z	S19	18.85	20.33			33	0
12	1	C	S17	28.94	51.52		12		
12	2	B	S6	51.52	68.07				
12	3	Z	S19	68.07	73.05			18	55.05
13	1	G	S4	5.51	20.25				
13	2	Z	S19	20.83	25.56			34	0
								Total	130.54

Table A.10 Machine View Best Solution for DS O2: PBLS-R

Machine	Job	Op	SG	Setup	ST	CT
A	2	1	S2		0.00	9.57
A	7	1	S2	0.5	10.07	20.50
B	1	2	S6		16.04	27.79
B	10	2	S6	0.5	28.44	37.15
B	12	2	S6	0.5	51.52	68.07
C	1	1	S17		0.00	16.04
C	10	1	S17	0.5	16.54	28.44
C	12	1	S17	0.5	28.94	51.52
E	8	1	S2		0.00	4.54
E	4	1	S3	2.5	7.04	18.90
F	5	1	S4		0.00	2.18
F	9	1	S4	0.5	2.68	10.91
F	11	1	S4	0.5	11.41	17.40
F	6	1	S4	0.5	17.90	24.66
G	3	1	S4		0.00	5.01
G	13	1	S4	0.5	5.51	20.25
G	7	2	S4	0.5	28.50	39.73
H	8	2	S4		12.54	16.78
H	2	2	S4	0.5	17.57	26.51
H	4	2	S5	3	29.51	40.58
Z	5	2	S19		2.18	2.72
Z	3	2	S19	0.5	5.01	6.62
Z	9	2	S19	0.5	10.91	12.95
Z	8	3	S19	0.5	16.78	18.35
Z	11	2	S19	0.5	18.85	20.33
Z	13	2	S19	0.5	20.83	25.56
Z	6	2	S19	0.5	26.06	27.73
Z	1	3	S19	0.5	28.23	31.77
Z	2	3	S19	0.5	32.27	35.58
Z	10	3	S19	0.5	37.15	39.78
Z	7	3	S19	0.5	40.28	43.88
Z	4	3	S19	0.5	44.38	48.48
Z	12	3	S19	0.5	68.07	73.05

A.2.3 Proxy Based Local Search with Neural Network for DS O2

Table A.11 Job View Best Solution for DS O2: PBLS-NN

Job	Op	Machine	SG	ST	CT	MinWait	MaxWait	Due	Tardy
1	1	C	S17	0.00	16.04		12		
1	2	B	S6	16.04	27.79				
1	3	Z	S19	27.79	31.33			21	10.33
2	1	A	S2	0.00	9.57	8			
2	2	F	S4	19.40	32.81				
2	3	Z	S19	32.81	36.12			15	21.12
3	1	H	S4	0.00	4.34				
3	2	Z	S19	4.34	5.95			34	0.00
4	1	E	S3	7.04	18.90	8			
4	2	H	S5	26.90	37.97				
4	3	Z	S19	44.38	48.48			39	9.48
5	1	G	S4	0.00	1.68				
5	2	Z	S19	1.68	2.22			21	0.00
6	1	G	S4	2.18	7.38				
6	2	Z	S19	10.81	12.48			48	0.00
7	1	A	S2	10.07	20.50	8			
7	2	G	S4	28.50	39.73				
7	3	Z	S19	40.28	43.88			18	25.88
8	1	E	S2	0.00	4.54	8			
8	2	F	S4	12.54	18.90				
8	3	Z	S19	18.90	20.47			20	0.47
9	1	G	S4	7.88	14.22				
9	2	Z	S19	14.22	16.25			35	0.00
10	1	C	S17	16.54	28.44		12		
10	2	B	S6	28.44	37.15				
10	3	Z	S19	37.15	39.78			31	8.78
11	1	H	S4	4.84	8.83				
11	2	Z	S19	8.83	10.31			33	0.00
12	1	C	S17	28.94	51.52		12		
12	2	B	S6	51.52	68.07				
12	3	Z	S19	68.07	73.05			18	55.05
13	1	H	S4	9.33	22.11				
13	2	Z	S19	22.11	26.84			34	0.00
								Total	131.11

Table A.12 Machine View Best Solution DS O2: PBLS-NN

Machine	Job	Op	SG	Setup	ST	CT
A	2	1	S2		0.00	9.57
A	7	1	S2	0.5	10.07	20.50
B	1	2	S6		16.04	27.79
B	10	2	S6	0.5	28.44	37.15
B	12	2	S6	0.5	51.52	68.07
C	1	1	S17		0.00	16.04
C	10	1	S17	0.5	16.54	28.44
C	12	1	S17	0.5	28.94	51.52
E	8	1	S2		0.00	4.54
E	4	1	S3	2.5	7.04	18.90
F	8	2	S4		12.54	18.90
F	2	2	S4	0.5	19.40	32.81
G	5	1	S4		0.00	1.68
G	6	1	S4	0.5	2.18	7.38
G	9	1	S4	0.5	7.88	14.22
G	7	2	S4	0.5	28.50	39.73
H	3	1	S4		0.00	4.34
H	11	1	S4	0.5	4.84	8.83
H	13	1	S4	0.5	9.33	22.11
H	4	2	S5	3	26.90	37.97
Z	5	2	S19		1.68	2.22
Z	3	2	S19	0.5	4.34	5.95
Z	11	2	S19	0.5	8.83	10.31
Z	6	2	S19	0.5	10.81	12.48
Z	9	2	S19	0.5	14.22	16.25
Z	8	3	S19	0.5	18.90	20.47
Z	13	2	S19	0.5	22.11	26.84
Z	1	3	S19	0.5	27.79	31.33
Z	2	3	S19	0.5	32.81	36.12
Z	10	3	S19	0.5	37.15	39.78
Z	7	3	S19	0.5	40.28	43.88
Z	4	3	S19	0.5	44.38	48.48
Z	12	3	S19	0.5	68.07	73.05

A.3 Summary of All Trial Runs

A.3.1 Summary of Runs for DS 1

Table A.13 Run Results for DS 1 (Average)

Width	Method	Random	Search Depth	Layers	Runs	Time	Avg Total Tardy	Avg Max Tardy	Avg Total Flow	Avg Make Span	Avg % Tardy Jobs
10	LR	FALSE	100%		100	616.62	1,844.34	179.40	135.82	302.24	49%
10	LR	TRUE	100%		100	667.76	1,757.74	180.41	134.27	302.04	48%
10	LR	TRUE	75%		100	672.77	1,862.85	179.10	136.12	307.16	49%
10	NN	TRUE	75%	10	50	14,557.74	2,304.87	173.59	144.61	317.29	53%
10	LR	TRUE	50%		100	679.29	2,073.01	178.82	140.13	315.52	52%
10	LR	TRUE	25%		100	681.70	2,402.67	175.42	148.73	324.54	55%
12	LR	FALSE	100%		100	1,628.31	1,734.42	180.13	133.96	299.88	47%
14	LR	TRUE	100%		100	1,140.14	1,666.54	181.89	131.37	298.64	46%
14	LR	TRUE	75%		100	1,175.83	1,814.80	178.56	135.48	307.50	48%
14	NN	TRUE	75%	10	50	21,092.84	2,245.83	175.50	145.09	319.19	53%
14	LR	TRUE	50%		100	1,187.14	1,954.31	178.60	138.29	311.93	50%
14	LR	TRUE	25%		100	1,132.24	2,352.38	177.69	146.56	324.09	54%
18	LR	FALSE	100%		100	2,017.99	1,614.34	183.97	131.97	297.01	46%
18	LR	TRUE	100%		100	1,863.61	1,563.39	179.40	129.59	295.85	45%
18	NN	TRUE	100%	1	5	204,624.99	1,869.43	188.26	134.00	312.62	50%
18	LR	TRUE	75%		100	1,924.29	1,696.96	183.27	132.52	304.10	47%
18	NN	TRUE	75%	10	50	16,619.87	2,226.24	174.42	144.43	319.76	53%
18	RW	TRUE	75%		100	2,100.41	2,802.78	178.27	156.36	330.98	58%
18	NN	TRUE	75%	1	50	168,773.41	1,940.43	182.83	139.58	311.08	51%
18	LR	TRUE	50%		100	1,792.05	1,958.53	179.49	137.81	312.84	50%
18	NN	TRUE	50%	1	50	127,712.60	2,065.77	174.34	146.38	311.51	51%
18	LR	TRUE	25%		100	1,678.22	2,302.31	179.48	145.02	320.51	54%
18	NN	TRUE	25%	1	50	82,570.19	2,438.27	186.58	152.30	321.07	55%
20	LR	TRUE	100%		100	3,007.17	1,556.53	182.13	128.70	298.03	44%
20	NN	TRUE	100%	10	50	9,689.69	2,110.85	175.71	141.34	317.14	51%
20	NN	TRUE	100%	1	50	9,562.81	1,940.68	183.44	137.59	313.46	48%
20	NN	TRUE	100%	2	50	9,624.63	2,037.38	183.20	139.98	316.57	50%
20	NN	TRUE	100%	3	50	9,595.01	1,955.77	183.39	137.68	311.09	51%
20	NN	TRUE	100%	4	50	9,706.20	2,018.94	179.30	139.71	311.93	49%
20	NN	TRUE	100%	5	50	9,680.21	2,055.03	174.91	139.67	315.48	50%
20	LR	TRUE	75%		100	3,245.40	1,698.62	184.39	132.14	303.51	47%
20	NN	TRUE	75%	10	50	17,775.38	2,114.08	176.50	141.45	319.17	51%
20	LR	TRUE	50%		100	3,185.56	1,911.93	178.48	137.37	310.97	49%
20	LR	TRUE	25%		100	2,793.99	2,280.48	175.42	144.79	319.55	54%

Table A.14 Summary of Run Results DS 1 (Variance)

Width	Method	Random	Search Depth	Layers	Runs	Var Total Tardy	Var Max Tardy	Var Total Flow	Var Make Span	Var % Tardy Jobs
10	LR	FALSE	100%		100	388.87	22.92	8.89	17.09	0.06
10	LR	TRUE	100%		100	337.75	21.08	7.64	14.08	0.06
10	LR	TRUE	75%		100	274.09	19.59	7.27	12.36	0.05
10	NN	TRUE	75%	10	50	309.46	18.33	7.82	12.88	0.05
10	LR	TRUE	50%		100	231.62	19.34	6.28	11.63	0.05
10	LR	TRUE	25%		100	186.47	21.28	5.86	11.50	0.05
12	LR	FALSE	100%		100	305.06	19.61	7.85	12.68	0.06
14	LR	TRUE	100%		100	300.58	17.54	7.44	12.73	0.06
14	LR	TRUE	75%		100	209.50	19.15	6.13	11.06	0.05
14	NN	TRUE	75%	10	50	381.79	18.63	8.98	14.39	0.05
14	LR	TRUE	50%		100	142.37	21.02	4.92	10.98	0.05
14	LR	TRUE	25%		100	162.57	20.87	4.38	12.54	0.04
18	LR	FALSE	100%		100	220.00	16.55	6.77	10.36	0.05
18	LR	TRUE	100%		100	184.25	17.92	5.05	11.17	0.04
18	NN	TRUE	100%	1	5	646.60	8.93	14.37	12.46	0.09
18	LR	TRUE	75%		100	155.17	18.95	4.77	11.11	0.05
18	NN	TRUE	75%	10	50	281.98	22.02	7.34	13.40	0.05
18	RW	TRUE	75%		100	186.59	20.67	5.67	12.11	0.04
18	NN	TRUE	75%	1	50	107.57	15.38	4.66	7.68	0.05
18	LR	TRUE	50%		100	171.72	18.17	5.72	10.66	0.04
18	NN	TRUE	50%	1	50	123.76	22.93	4.01	14.14	0.02
18	LR	TRUE	25%		100	145.71	20.42	4.51	9.49	0.04
18	NN	TRUE	25%	1	50	291.08	15.37	4.66	9.47	0.04
20	LR	TRUE	100%		100	212.88	16.76	5.66	10.65	0.04
20	NN	TRUE	100%	10	50	312.12	21.06	6.30	14.37	0.05
20	NN	TRUE	100%	1	50	226.56	18.53	6.42	10.04	0.04
20	NN	TRUE	100%	2	50	293.73	22.25	8.33	13.83	0.05
20	NN	TRUE	100%	3	50	276.01	21.43	7.42	13.70	0.05
20	NN	TRUE	100%	4	50	304.93	16.34	7.82	12.43	0.05
20	NN	TRUE	100%	5	50	260.58	20.17	7.68	14.10	0.04
20	LR	TRUE	75%		100	159.23	17.21	5.13	11.20	0.04
20	NN	TRUE	75%	10	50	262.13	21.13	6.15	12.01	0.04
20	LR	TRUE	50%		100	139.71	19.67	5.01	9.92	0.04
20	LR	TRUE	25%		100	388.87	22.92	8.89	17.09	0.06

A.3.2 Summary of Runs for DS 2

Table A.15 Run Results DS 2 (Average)

Width	Method	Random	Search Depth	Layers	Runs	Time	Avg Total Tardy	Avg Max Tardy	Avg Total Flow	Avg Make Span	Avg % Tardy Jobs
10	LR	FALSE	100%		100	938.71	5,637.88	170.78	202.09	305.00	71.4%
10	LR	TRUE	100%		100	956.43	5,676.60	167.52	202.73	304.21	71.7%
10	LR	TRUE	75%		100	970.58	5,573.30	163.80	201.79	307.64	70.8%
10	LR	TRUE	50%		100	1,040.72	5,609.81	161.39	202.45	312.38	70.2%
10	LR	TRUE	25%		100	940.59	5,959.98	166.13	205.52	324.02	67.4%
12	LR	FALSE	100%		100	1,619.84	5,605.42	165.73	202.88	306.17	73.5%
14	LR	TRUE	100%		100	1,574.65	5,498.74	166.81	200.98	300.57	72.1%
14	LR	TRUE	75%		100	1,644.32	5,511.73	162.08	201.49	303.01	71.5%
14	LR	TRUE	50%		100	1,523.50	5,590.53	159.03	201.82	311.68	69.2%
14	LR	TRUE	25%		100	1,450.54	5,829.60	165.85	204.85	321.69	68.4%
18	LR	FALSE	100%		100	3,147.57	5,397.99	164.50	200.32	303.07	71.0%
18	LR	TRUE	100%		100	3,802.53	5,360.24	160.71	200.90	302.91	71.9%
18	LR	TRUE	100%		100	3,082.24	5,390.01	157.94	201.12	295.59	72.7%
18	NN	TRUE	100%	10	50	24,313.48	5,512.72	160.19	202.01	309.69	70.0%
18	NN	TRUE	100%	1	50		5,414.37				
18	LR	TRUE	75%		100	3,444.14	5,426.90	160.66	200.77	302.69	71.2%
18	LR	TRUE	75%		100	3,072.66	5,405.66	159.10	200.52	302.68	69.7%
18	NN	TRUE	75%	10	50	24,621.31	5,600.66	165.33	202.65	313.39	68.9%
18	RW	TRUE	75%		100	3,085.57	6,213.67	170.89	208.27	329.28	68.2%
18	LR	TRUE	50%		100	3,489.22	5,504.22	161.54	200.64	310.30	68.1%
18	LR	TRUE	50%		100	2,531.31	5,431.94	159.55	201.10	310.30	69.9%
18	NN	TRUE	50%	10	50	18,164.73	5,843.94	162.79	204.41	320.19	68.4%
18	LR	TRUE	25%		100	3,554.57	5,807.23	165.17	204.37	322.78	68.2%
18	NN	TRUE	25%	10	50	11,320.77	5,924.67	169.31	206.01	327.42	67.8%
20	LR	TRUE	100%		100	36,159.20	5,363.83	163.78	200.58	300.77	72.0%
20	LR	TRUE	100%		100	3,998.82	5,438.28	161.41	201.62	304.39	73.2%
20	LR	TRUE	75%		100	29,816.08	5,420.39	154.21	200.91	300.23	71.3%
20	LR	TRUE	75%		100	3,474.09	5,381.26	156.21	200.50	303.37	71.0%
20	LR	TRUE	50%		100	3,154.46	5,411.10	158.20	200.66	307.08	71.3%
20	LR	TRUE	25%		100	3,125.94	5,733.20	161.90	203.58	319.58	68.8%

Table A.16 Run Results DS 2 (Variance)

Width	Method	Random	Search Depth	Layers	Runs	Time	Var Total Tardy	Var Max Tardy	Var Total Flow	Var Make Span	Var % Tardy Jobs
10	LR	FALSE	100%		100	938.71	564.49	35.31	6.55	27.29	0.10
10	LR	TRUE	100%		100	956.43	542.95	31.65	6.63	26.67	0.09
10	LR	TRUE	75%		100	970.58	479.51	26.17	5.65	21.86	0.09
10	LR	TRUE	50%		100	1,040.72	413.30	19.21	5.33	18.23	0.09
10	LR	TRUE	25%		100	940.59	333.07	15.24	4.64	16.31	0.06
12	LR	FALSE	100%		100	1,619.84	544.18	27.73	6.53	25.59	0.10
14	LR	TRUE	100%		100	1,574.65	465.11	31.81	5.78	26.32	0.09
14	LR	TRUE	75%		100	1,644.32	491.64	25.68	5.67	21.10	0.08
14	LR	TRUE	50%		100	1,523.50	379.93	19.94	4.92	16.46	0.08
14	LR	TRUE	25%		100	1,450.54	305.44	16.09	4.07	13.98	0.06
18	LR	FALSE	100%		100	3,147.57	500.51	31.34	6.10	25.65	0.10
18	LR	TRUE	100%		100	3,802.53	524.20	29.28	5.87	24.89	0.08
18	LR	TRUE	100%		100	3,082.24	515.19	26.26	6.11	25.96	0.10
18	NN	TRUE	100%	10	50	24,313.48	446.92	21.20	5.27	21.16	0.09
18	NN	TRUE	100%	1	50		503.85				
18	LR	TRUE	75%		100	3,444.14	411.07	29.49	5.28	20.46	0.08
18	LR	TRUE	75%		100	3,072.66	434.88	23.53	5.46	21.16	0.08
18	NN	TRUE	75%	10	50	24,621.31	347.86	24.34	4.25	18.38	0.07
18	RW	TRUE	75%		100	3,085.57	222.32	14.21	4.27	12.42	0.08
18	LR	TRUE	50%		100	3,489.22	356.44	22.35	4.89	16.44	0.08
18	LR	TRUE	50%		100	2,531.31	361.27	20.77	4.75	17.01	0.07
18	NN	TRUE	50%	10	50	18,164.73	341.74	16.76	5.33	14.50	0.09
18	LR	TRUE	25%		100	3,554.57	278.47	16.48	4.21	13.48	0.07
18	NN	TRUE	25%	10	50	11,320.77	283.85	13.91	3.95	13.01	0.07
20	LR	TRUE	100%		100	36,159.20	455.78	34.45	5.52	23.03	0.09
20	LR	TRUE	100%		100	3,998.82	501.27	29.24	5.66	23.94	0.10
20	LR	TRUE	75%		100	29,816.08	409.67	23.14	5.07	18.83	0.09
20	LR	TRUE	75%		100	3,474.09	391.16	22.01	4.74	20.90	0.10
20	LR	TRUE	50%		100	3,154.46	318.69	19.69	4.50	15.24	0.09
20	LR	TRUE	25%		100	3,125.94	247.58	16.12	4.60	13.48	0.08

A.3.3 Summary of Runs for DS 3

Table A.17 Run Results DS 3 (Average)

Width	Method	Random	Search Depth	Layers	Runs	Time	Avg Total Tardy	Avg Max Tardy	Avg Total Flow	Avg Make Span	Avg % Tardy Jobs
10	LR	FALSE	100%		100	473.69	1,681.24	127.14	142.52	224.97	58.7%
10	LR	TRUE	100%		100	546.54	1,639.89	125.47	142.02	225.26	58.7%
10	LR	TRUE	75%		100	516.96	1,623.25	127.74	143.16	225.24	59.1%
10	LR	TRUE	50%		100	526.44	1,752.38	127.59	145.85	231.12	60.2%
10	LR	TRUE	25%		100	524.67	2,055.86	138.30	151.63	240.67	61.8%
12	LR	FALSE	100%		100	967.90	1,664.14	130.31	142.74	226.61	59.0%
14	LR	TRUE	100%		100	874.01	1,600.78	124.46	141.01	226.29	58.6%
14	LR	TRUE	75%		100	900.67	1,597.28	127.09	142.15	226.00	58.0%
14	LR	TRUE	50%		100	894.62	1,675.50	131.53	144.22	228.19	58.7%
14	LR	TRUE	25%		100	949.72	2,033.66	132.58	151.17	239.85	62.0%
18	LR	FALSE	100%		100	1,802.67	1,613.92	130.56	142.02	222.20	58.6%
18	LR	TRUE	100%		100	1,563.37	1,601.17	130.93	142.41	225.47	59.3%
18	NN	TRUE	100%	10	50	23,583.41	1,817.85	134.84	147.24	234.54	60.7%
18	NN	TRUE	100%	10	50	24,939.71	1,813.33	135.39	145.75	233.30	60.1%
18	NN	TRUE	100%	1	50	25,637.08	1,560.79	127.70	140.99	227.76	56.5%
18	NN	TRUE	100%	2	50	25,044.47	1,614.86	123.64	141.42	227.15	58.2%
18	NN	TRUE	100%	3	50	24,540.06	1,634.13	132.16	142.47	229.96	57.0%
18	NN	TRUE	100%	4	50	24,842.03	1,712.97	132.15	144.54	229.65	58.7%
18	NN	TRUE	100%	5	50	24,848.56	1,700.34	132.28	143.80	232.54	58.7%
18	RW	TRUE	100%		100	1,827.05	2,481.93	142.85	160.72	251.34	64.6%
18	LR	TRUE	75%		100	1,543.58	1,541.18	125.00	140.47	224.79	57.6%
18	LR	TRUE	50%		100	1,512.40	1,646.94	128.89	144.21	231.04	59.1%
18	LR	TRUE	25%		100	1,443.34	1,979.10	136.20	150.36	239.33	61.8%
20	LR	TRUE	100%		100	2,364.49	1,601.09	125.15	141.38	223.99	58.8%
20	LR	TRUE	75%		100	2,441.73	1,535.90	126.01	140.56	223.96	57.3%
20	LR	TRUE	50%		100	2,318.24	1,642.25	129.47	143.34	228.41	58.5%
20	LR	TRUE	25%		100	2,335.08	1,963.64	135.84	150.21	239.43	61.5%

Table A.18 Run Results DS 3 (Variance)

Width	Method	Random	Search Depth	Layers	Runs	Avg Total Tardy	Avg Max Tardy	Avg Total Flow	Avg Make Span	Avg % Tardy Jobs
10	LR	FALSE	100%		100	211.94	25.74	7.12	15.63	0.04
10	LR	TRUE	100%		100	206.71	22.24	5.93	13.69	0.04
10	LR	TRUE	75%		100	201.91	22.97	5.96	13.43	0.04
10	LR	TRUE	50%		100	164.35	17.79	5.36	14.25	0.05
10	LR	TRUE	25%		100	176.36	22.79	5.87	15.97	0.04
12	LR	FALSE	100%		100	210.36	24.23	5.46	14.68	0.05
14	LR	TRUE	100%		100	193.00	20.40	5.93	14.00	0.05
14	LR	TRUE	75%		100	183.34	19.28	6.23	12.93	0.04
14	LR	TRUE	50%		100	157.23	17.60	5.53	13.15	0.05
14	LR	TRUE	25%		100	150.17	17.45	4.68	14.34	0.04
18	LR	FALSE	100%		100	187.80	30.22	5.59	12.43	0.04
18	LR	TRUE	100%		100	179.60	27.25	6.09	14.22	0.05
18	NN	TRUE	100%	10	50	287.50	19.91	6.61	16.92	0.05
18	NN	TRUE	100%	10	50	219.90	23.38	5.27	12.23	0.05
18	NN	TRUE	100%	1	50	192.45	17.31	6.51	15.42	0.04
18	NN	TRUE	100%	2	50	204.16	14.50	6.07	13.33	0.04
18	NN	TRUE	100%	3	50	205.80	20.72	6.74	12.15	0.05
18	NN	TRUE	100%	4	50	214.79	18.87	6.03	15.73	0.04
18	NN	TRUE	100%	5	50	178.08	16.21	5.76	13.78	0.05
18	RW	TRUE	100%		100	143.56	25.07	4.98	15.49	0.04
18	LR	TRUE	75%		100	122.31	16.63	4.68	10.62	0.04
18	LR	TRUE	50%		100	121.62	14.12	5.37	12.37	0.04
18	LR	TRUE	25%		100	123.57	18.21	5.19	14.76	0.04
20	LR	TRUE	100%		100	166.94	19.73	5.40	11.54	0.05
20	LR	TRUE	75%		100	106.55	20.35	4.93	9.97	0.04
20	LR	TRUE	50%		100	116.60	18.31	4.80	11.66	0.04
20	LR	TRUE	25%		100	148.11	21.64	5.14	13.33	0.04

A.3.4 Summary of Runs for DS 4

Table A.19 Run Results DS 4 (Average)

Width	Method	Random	Search Depth	Layers	Runs	Time	Avg Total Tardy	Avg Max Tardy	Avg Total Flow	Avg Make Span	Avg % Tardy Jobs
10	LR	FALSE	100%		100	1,033.05	459.88	119.23	406.14	627.88	7.2%
10	LR	TRUE	100%		100	1,137.48	455.28	114.04	405.19	637.83	7.0%
10	NN	TRUE	100%	10	50	42,198.22	419.83	101.11	406.22	628.69	6.8%
10	LR	TRUE	75%		100	1,159.02	430.26	105.92	401.69	631.08	6.8%
10	LR	TRUE	50%		100	1,158.51	416.04	102.55	403.44	620.54	6.7%
10	LR	TRUE	25%		100	997.06	549.01	124.91	409.83	638.97	7.9%
14	LR	TRUE	100%		100	2,249.73	438.75	106.33	403.56	636.30	6.9%
14	NN	TRUE	100%	10	50	29,571.01	398.34	100.98	402.79	623.24	6.5%
14	LR	TRUE	75%		100	2,097.24	394.67	99.94	401.39	619.43	6.7%
14	LR	TRUE	50%		100	2,086.72	398.19	95.28	403.62	616.94	6.7%
14	LR	TRUE	25%		100	2,022.61	520.40	120.17	405.90	624.60	7.6%
18	LR	FALSE	100%		100	6,844.95	462.20	111.35	402.32	623.91	7.1%
18	LR	TRUE	100%		100	5,684.58	417.68	108.89	403.94	628.70	6.6%
18	NN	TRUE	100%	10	50	45,375.23	392.41	99.35	404.14	629.54	7.0%
18	RW	TRUE	100%		100	3,970.95	580.31	121.59	414.57	644.89	8.4%
18	LR	TRUE	75%		100	5,520.14	398.80	96.21	401.86	625.74	6.5%
18	NN	TRUE	75%	10	50	14,549.22	451.00	125.96	402.29	626.40	6.8%
18	NN	TRUE	75%	1	50	14,745.58	337.63	80.64	399.69	624.51	6.1%
18	NN	TRUE	75%	2	50	14,512.69	426.08	123.52	402.89	641.12	6.4%
18	NN	TRUE	75%	3	50	14,650.80	367.30	91.81	398.82	620.99	6.3%
18	NN	TRUE	75%	4	50	14,915.95	367.34	95.13	399.54	621.74	6.0%
18	NN	TRUE	75%	5	50	14,769.73	384.57	95.64	399.21	614.47	6.6%
18	LR	TRUE	50%		100	6,820.73	396.88	94.17	398.82	609.70	6.5%
18	LR	TRUE	25%		100	5,757.04	504.45	118.17	408.03	628.42	7.5%
20	LR	TRUE	100%		100	8,011.55	403.10	97.63	401.96	628.05	6.5%
20	NN	TRUE	100%	10	50	47,009.88	359.10	95.15	403.74	619.84	5.9%
20	LR	TRUE	100%		100	3,674.73	419.71	109.63	404.43	630.32	6.7%
20	LR	TRUE	75%		100	7,838.01	400.91	96.81	399.35	611.41	6.6%
20	LR	TRUE	75%		100	3,645.75	391.46	91.84	400.93	620.67	6.6%
20	LR	TRUE	50%		100	8,757.55	381.80	85.68	401.06	612.99	6.6%
20	LR	TRUE	50%		100	3,733.70	389.36	96.87	399.09	610.56	6.5%
20	LR	TRUE	25%		100	3,739.88	506.73	114.71	403.80	625.14	7.5%

Table A.20 Run Results DS 4 (Variance)

Width	Method	Random	Search Depth	Layers	Runs	Var Total Tardy	Var Max Tardy	Var Total Flow	Var Make Span	Var % Tardy Jobs
10	LR	FALSE	100%		100	58.05	19.45	49.62	0.02	0.15
10	LR	TRUE	100%		100	71.19	19.42	44.24	0.02	0.17
10	NN	TRUE	100%	10	50	46.91	17.54	46.63	0.02	0.10
10	LR	TRUE	75%		100	45.83	17.61	49.96	0.02	0.12
10	LR	TRUE	50%		100	43.11	17.28	40.28	0.02	0.09
10	LR	TRUE	25%		100	58.04	13.56	38.07	0.02	0.04
14	LR	TRUE	100%		100	61.30	17.28	52.47	0.02	0.16
14	NN	TRUE	100%	10	50	85.70	15.85	47.83	0.02	0.14
14	LR	TRUE	75%		100	46.96	17.15	44.94	0.02	0.12
14	LR	TRUE	50%		100	37.68	16.18	41.79	0.02	0.07
14	LR	TRUE	25%		100	54.05	15.16	40.28	0.02	0.04
18	LR	FALSE	100%		100	56.30	16.50	52.00	0.02	0.19
18	LR	TRUE	100%		100	72.38	19.04	48.72	0.02	0.16
18	NN	TRUE	100%	10	50	49.10	13.56	36.31	0.02	0.10
18	RW	TRUE	100%		100	59.45	11.57	33.01	0.02	0.03
18	LR	TRUE	75%		100	40.04	16.51	42.74	0.01	0.12
18	NN	TRUE	75%	10	50	62.50	13.15	41.00	0.02	0.07
18	NN	TRUE	75%	1	50	32.61	11.14	45.77	0.02	0.14
18	NN	TRUE	75%	2	50	68.86	14.42	47.16	0.02	0.17
18	NN	TRUE	75%	3	50	25.60	11.15	34.81	0.01	0.11
18	NN	TRUE	75%	4	50	35.88	10.55	41.23	0.01	0.08
18	NN	TRUE	75%	5	50	40.63	14.47	43.46	0.02	0.09
18	LR	TRUE	50%		100	36.96	13.58	36.51	0.01	0.07
18	LR	TRUE	25%		100	51.85	13.33	33.10	0.01	0.03
20	LR	TRUE	100%		100	48.34	16.60	46.66	0.02	0.14
20	NN	TRUE	100%	10	50	31.75	14.57	33.90	0.01	0.09
20	LR	TRUE	100%		100	57.46	18.30	49.94	0.02	0.17
20	LR	TRUE	75%		100	40.67	14.57	40.72	0.01	0.12
20	LR	TRUE	75%		100	33.74	14.69	41.93	0.02	0.13
20	LR	TRUE	50%		100	30.05	15.42	42.21	0.02	0.07
20	LR	TRUE	50%		100	44.79	14.93	34.38	0.02	0.07
20	LR	TRUE	25%		100	57.03	13.94	33.41	0.01	0.03