Comparison of Scheduling Algorithms for a Multi-Product Batch-Chemical Plant with a Generalized Serial Network

Niem-Trung L. Tra

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

Master of Science

in

Industrial and Systems Engineering.

_____

Dr. John E. Kobza, Co-Chair

_____

Dr. C. Patrick Koelling, Co-Chair

_____

Dr. Ebru Bish

_____

Dr. Kimberly P. Ellis

January 24, 2000

Blacksburg, Virginia

Comparison of Scheduling Algorithms for a Multi-Product Batch-Chemical Plant with a Generalized Serial Network

Niem-Trung L. Tra

ABSTRACT

Despite recent advances in computer power and the development of better algorithms, theoretical scheduling methodologies developed for batch-chemical production are seldom applied in industry (Musier & Evans 1989 and Grossmann et al. 1992). Scheduling decisions may have significant impact on overall company profitability by defining how capital is utilized, the operating costs required, and the ability to meet due dates. The purpose of this research is to compare different production scheduling methods by applying them to a real-world multi-stage, multi-product, batch-chemical production line. This research addresses the problem that the theoretical algorithms are seldom applied in industry and allows for performance analysis of several theoretical algorithms.

The research presented in this thesis focuses on the development and comparison of several scheduling algorithms. The two objectives of this research are to:

1. modify different heuristic production scheduling algorithms to minimize tardiness for a multi-product batch plant involving multiple processing stages with several out-of-phase parallel machines in each stage; and
2. compare the robustness and performance of these production schedules using a stochastic discrete event simulation of a real-world production line.

The following three scheduling algorithms are compared:

1. a modified Musier and Evans scheduling algorithm (1989);
2. a modified Ku and Karimi Sequence Building Algorithm (1991); and
3. a greedy heuristic based on an earliest-due-date (EDD) policy.

Musier and Evans' heuristic improvement method (1989) is applied to the three algorithms. The computation times to determine the total tardiness of each schedule are compared. Finally, all the schedules are tested for robustness and performance in a stochastic setting with the use of a discrete event simulation (DES) model. Mignon, Honkomp, and Reklaitis' evaluation techniques (1995) and Multiple Comparison of the Best are used to help determine the best algorithm.

ACKNOWLEDGEMENTS

Many people have contributed a great deal of their time and effort so that I may finish this research and thesis. I would like to thank my committee members for their time, support, and the knowledge they shared with me. I appreciate the support and encouragement that Dr. Ellis provided me. Dr. Bish introduced me to the world of scheduling and her technical, personal, and optimistic support has been invaluable. Special thanks are given to Dr. Koelling and Dr. Kobza for serving as co-advisors and for providing guidance throughout this complex process. Dr. Koelling's technical support is much appreciated. Dr. Kobza introduced me to our industrial partner and his endless support, guidance, edits and friendship will always be cherished. I would also like to thank our industrial partner for their time and support on this research. Without their cooperation a significant part of this research would not have been completed. Finally, I would like to thank my family and friends for their support, advice, and encouragement throughout my studies.

TABLE OF CONTENTS

FIGURES

TABLES

CHAPTER 1: INTRODUCTION

A significant proportion of the world's chemical production by volume, and a much larger proportion by value, is still made in batch plants and it does not seem likely that this proportion will decline in value (Overturf, Reklaitis, and Woods 1978 and Rippin 1983b). The need for scheduling production operations originates from the competing alternatives available to utilize limited resources. Scheduling problems involve large numbers of tasks and resources subject to a diverse set of utilization constraints and objectives. The scheduling decisions may have significant impact on overall company profitability by defining how capital is utilized, the operating costs required, and the ability to meet due dates.

The scheduling methodologies presented in the literature laid a good theoretical foundation. However, despite recent advances in computer power and the development of better algorithms, they are seldom applied in industry (Musier & Evans 1989 and Grossmann et al. 1992). The purpose of this research is to compare different production scheduling methods by applying them to a real-world multi-stage, multi-product, batch-chemical production line. The following three scheduling algorithms were compared:

1. a modified Musier and Evans scheduling algorithm (1989);
2. a modified Ku and Karimi Sequence Building Algorithm (1991); and
3. a greedy heuristic based on an earliest-due-date (EDD) policy.

The objective of these algorithms is to minimize total tardiness. A discrete-event simulation model of a production line is used to determine how well they perform in a stochastic setting and how they compare to one another. Table 1 provides a summary of the research.

The remaining chapters are organized as follows. Chapter 2 presents an overview of the various methods used to address the production scheduling problem in the batch-chemical processing industry. Chapter 3 presents the problem description and the methodology of the proposed research. Chapter 4 summarizes the data requirements for the proposed research. Chapter 5 provides a detailed outline of Musier and Evans' modified scheduling algorithm and improvement method, along with Ku and Karimi's modified scheduling algorithm and the Earliest-Due-Date scheduling algorithm. Chapter 6 provides a detailed explanation of how the simulation model was developed and the type of production line it represents. Chapter 7 gives a detailed explanation of the comparison techniques to be used. Finally, Chapter 8 summarizes the experimental results and draws conclusions about them.

Description	Musier and Evans Heuristic (1989)	Ku and Karimi SB Algorithm (1991)	Greedy Heuristic
Initial Optimal Schedule	uses ordering by random number assignment	builds a good production sequence from partial sequences	uses earliest due-date
Objective Function	$\sum_{i=1}^{N} T_i$ sum of the tardiness penalties		
Improvement Method	Musier and Evans' heuristic improvement method: based on the idea that 1-optimal and 2-optimal schedules are close to the global optimum.		
Simulation Model	Discrete-event model of a multi-product batch plant involving multiple processing stages with several out-of-phase parallel machines in each stage.		
Comparison Techniques	Mignon, Honkomp, and Reklaitis' Evaluation Technique (1995) and Multiple Comparison of the Best		
Things to Compare	objective function value; robustness; performance; required computing time; required simulation time; and throughput		

Table 1: Research Summary

CHAPTER 2: LITERATURE SURVEY

Plants involving batch operations have received considerable attention in the literature and interest in designing and scheduling batch plants continues to grow (Rich and Prokopakis 1986). This literature review focuses on the production scheduling of batch-chemical plants. Batch processing is predominately used to produce multiple, low-volume, high-value-added products in a single plant. These products are made based on their own recipe by sharing available resources. This allows for a higher degree of flexibility in scheduling but adds to the complexity of operations.

Scheduling techniques proposed to solve batch production scheduling problems are almost as complex and diverse as the problem itself. Batch scheduling is a method that determines the order in which products are to be processed to optimize a desired criterion under capacity constraints (Reklaitis 1992). Scheduling algorithms have been categorized as exact (mathematical programming), heuristic, and search (stochastic) methods (Dockx and Meert, 1993). There has been considerable work in applying mathematical programming approaches to production scheduling. Exact methods, such as mixed-integer linear programming (MILP) and non-linear programming (MINLP), are enumerative techniques that ensure that the schedule provides an optimal solution. However, Garey et al. (1976) and Papadimitrou and Kanellakis (1980) showed that most scheduling problems are NP-complete. “NP-completeness means that any algorithm intending to solve such problems in its full generality is likely to require exponential time in the worst case and thus be impractical except for relatively small inputs (Garey et al. 1976).” However, NP-completeness does not mean that simple cases of the scheduling problem cannot be solved in practice. Therefore, for larger and more complicated problems, it is recommended that approximation methods be used to obtain near-optimal results. Approximation methods, also known as heuristics, “must be mathematically proven to generate a solution within a certain percentage of optimality no matter what problem instance is solved (Sipper and Bulfin 1997).”

Finally, search methods, such as simulated annealing and tabu search, provide at least local optimal solutions (Dockx and Meert 1993 and Sangiovanni-Vincentelli 1991). These search methods are based on the hill climbing problem solving technique; these stochastic methods conduct a search in a statistically ordered manner.

> [The hill climbing technique makes] a guess at a solution and then move[s] to a better solution closer to the [optimal]. To determine whether another solution is better, some type of evaluation or reward function is needed. The solution is when the evaluation function is maximal (or minimal). In many instances the hill climbing solution is an algorithm. These 'algorithms have the ability to probabilistically accept candidates solutions with higher cost than that of the incumbent solution, in an effort to escape local optima (Johnson 1996).' Many hill climbing methods are "greedy" algorithms, which may not yield a best solution (Cook 1999).

The simulated annealing is a frequently used stochastic hill climbing algorithm (Johnson 1996). Jacobson and Yucesan (2000) summarizes the simulated annealing algorithm as follows:

> Simulated annealing starts with an initial solution. A neighboring solution is then generated (either randomly or using some prespecified rule). If the neighboring solution is greater/less than or equal to the initial solution, then it becomes the current solution. However, if the neighboring solution is less/greater than the initial solution, then it is accepted as the current solution with a certain probability, which is related to an annealing temperature that a nonincreasing at each iteration.

The annealing temperature dictates how high the acceptance probability will be; the higher the temperature the more likely that a bad move can be made, so that it can escape from local optima and hence move closer to the global optimal solution. A global optima is the optimal solution for all possible regions of the problem; whereas, a local optima is the optimal solution to only a small region of the entire solution space (Bazaraa et al. 1979 and Himmelblau 1972). The temperature can be controlled so that it is large enough to move the current solution off a local optima but small enough not to move it off a global optima. As the temperature approaches zero, simulated annealing behaves more like a hill climbing technique that is performing a local search, searching for an optimal solution in a small region of the possible solution space. Simulated annealing is very attractive for solving large-scale scheduling problems in batch processes because of its generality and simplicity (Ku and Karimi 1991a). However, simulated annealing is a very time consuming technique (Boissel and Kantor 1995). Tabu search uses a limited memory of past moves to help diversify the search and avoid becoming trapped in local optima. Each time a move is made, it is placed on a list called the "tabu-list." A move cannot be selected if it is on the tabu-list. Old moves are removed from the tabu-list after some predefined number of iterations.

The primary techniques used for solving the batch scheduling problem have just been introduced. The remainder of the chapter will discuss the characteristics of batch-processes, how

the various techniques introduced above have been used to solve various batch scheduling problems, suggest other literature surveys to review, and generalize what future work needs to be done in this area.

2.1 Characteristics of Batch-Process

Graham et al. (1979), Reklaitis (1982), Musier and Evans (1989), and Liu and McGreavy (1996) provide overviews of the role of scheduling by defining what the scheduling problem involves and the various network structures, i.e., the layout of machines and flow of material, that are prevalent to the chemical production industry. General scheduling problems are defined as having (Reklaitis 1982):

- a set of N products which must be processed,
- a set of M machines which are available for this purpose,
- a performance criterion used to optimize the schedule,
- a set of production times, and
- a set of rules which govern the production process, such as order of operations, intermediate storage capacities, allowances made for subdividing product runs.

The various network structures used to represent the production flow within a facility are parallel, serial, and generalized serial (see Figure 1). Parallel networks consist of several machines that can perform the same process on a job. Since material flows directly to the parallel machines, intermediate storage areas are not considered in a parallel network. Each parallel production line consists of a series of machines and is independent from other parallel production lines. Individually, these parallel production lines are called serial networks. The only relationship the parallel lines have with one another is satisfying demand, i.e., the lines are all operating in order to fill all the customer orders. Although parallel structures are quite common in the chemical production industry, serial networks are the most pertinent to the multi-product plants in the chemical production industry.

Unlike the parallel network, serial networks do consider intermediate storage areas because incoming material must wait for machines in the next area to finish up before it can be processed. Serial machines may have four different intermediate storage rules: unlimited intermediate storage, no intermediate storage, zero-wait, and finite intermediate storage. Under the unlimited intermediate storage rule the product is always removed from the machine upon process completion. The no intermediate storage rule only allows for products to be removed from processing machines if the next machine is available for processing. When intermediate

products are not allowed to wait between processing steps, the zero wait storage rule is being utilized. Finite intermediate storage is similar to unlimited intermediate, except that a limit is set on the number of intermediate products that can be stored between processing stages. If restrictions are placed on the availability of intermediate storage, there is a tendency to increase machine idle times and therefore increase the makespan. This happens because "intermediate storage tanks decouple the operation of the resulting upstream and downstream substrains and allows the independence of both subprocesses in terms of capacity constraints as well as cycle time (Graells, et al. 1995)."

Chemical plants usually consist of a series of one or more stages (or processing areas) with a set of parallel machines in each stage (Hindi and Toczylowski 1995 and Musier and Evans 1989). This is known as a generalized serial system. Similar to the standard serial network, restrictions may be placed on the type of intermediate storage that follows each processing area.

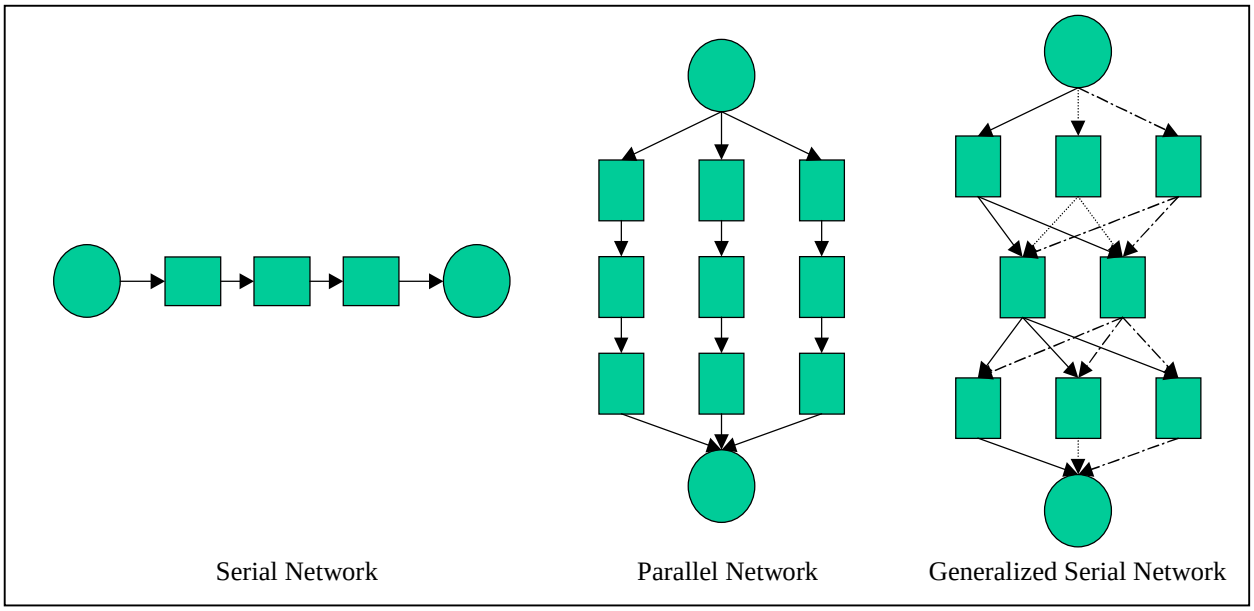

Figure 1: Network Structures for Batch Processes

Other characteristics of a scheduling problem include open vs. closed shops, preemptive vs. non-preemptive production runs, multi-product vs. multi-purpose plants, and production in campaigns. An open shop is one where all production orders are by customer request and no inventory is stocked. Whereas in a closed shop all customer requests are serviced from inventory and production tasks are generally a result of inventory replenishment decisions (Graves 1981). Under preemptive production, a batch may be split-up or interrupted during its run; with non-preemptive production once processing begins on the batch it continues uninterrupted in the

machine until it is finished. Multi-product plants have jobs that follow a fixed routing through the plant one batch at a time and commonly have single-stage, i.e., one group of similar machines, serial production lines. Multi-purpose plants produce ordered jobs simultaneously throughout the plant and sequences for successive production runs may vary (Rich and Prokopakis 1986). A campaign is a period in which identical/non-identical "batches of the same order are successively processed in the same machine (Creda and Henning 1997)."

As in any processing plant, batch processes have variations caused by operator response times, fluctuations in utility availability, equipment failures, variation in process times, recipe inaccuracies, and changes in raw material quality. These variations contribute to the difficulty of production scheduling. The ability to react to variations during the execution of a schedule is an important part of a batch plant operating strategy since real plants exist in a dynamic environment.

2.2 Mathematical Programming Approaches

The previous section discussed the characteristics of batch-processing. This section focuses on exact techniques (mathematical programming) used to find the optimal solution for the batch-process scheduling problem. Mathematical programming approaches have been developed using MILP and MINLP.

2.2.1 Mixed Integer Linear Programming Algorithms

Kondili et al. (1993) presented a general framework for handling a wide range of scheduling problems for multi-purpose/product batch-chemical plants. They introduced a MILP maximizing profit for a short-term scheduling problem that involved sharing raw material and intermediates, batch splitting and mixing, recycling material, flexible equipment allocation, variable batch sizes, and mixed intermediate storage policies. Prabhakar (1974) presented a MILP for optimizing, sequencing, and scheduling; it was solved using a branch and bound approach. Mauderli and Rippin (1979) developed a computer program based on MILP (enumeration of alternatives) for solving general production planning and scheduling in a multi-purpose, parallel batch-chemical plant without intermediate storage. Rich and Prokopakis (1986) developed an approach to scheduling and sequencing multiple production runs over a finite time horizon.

Coulman (1989) presented an optimal design and scheduling approach for a multi-purpose batch plant. The production line analyzed allowed for parallel machines of each type of equipment to exist and operate out-of-phase, machines start and stop at different times, but all machines of one type can only be used for one product at a given time. Rajagopalan and Karimi (1989) presented a method for determining the completion times of all machines in a serial flowshop, with a single machine in each processing stage, considering transfer and set-up times. Patsidou and Kantor (1991) derived a MILP algorithm to graphically solve scheduling problems in multi-purpose batch plants, with consideration given to the various intermediate storage policies. Penky et al. (1993) developed an exact algorithm for resource constrained sequencing based on the Resource Constrained Traveling Salesman Problem, and solved using branch and bound. They also presented a relaxation technique that combines LP-based branch and bound with disjunctive programming. Gooding et al. (1994) developed an algorithm, solved with a branch and bound algorithm, based on a transformation from a parallel flowshop problem to a constrained traveling salesman problem. Their algorithm solved a parallel flowshop scheduling problem. Dedopoulos and Shah (1995) provided a MILP model maximizing profit, solved with a branch and bound method, for simultaneously scheduling production and maintenance activities within a multi-purpose batch/semi-continuous plant.

Graells et al. (1995) introduced a comprehensive method for scheduling multi-purpose batch-chemical processes while considering intermediate storage and in-phase and out-of-phase operation modes. Daniel et al. (1996) developed a mathematical formulation and both an optimal and a heuristic solution for a parallel manufacturing area. Both static and dynamic resources were analyzed. Pinto and Grossmann (1997) developed a MILP scheduling algorithm modeling discrete resource constraints in continuous time. This accommodated different storage modes, multi-purpose, parallel, and preemptive processing, along with limited availability of utilities and manpower. Cerda et al. (1997) presented a formulation for the batch-scheduling problem involving a single processing stage with an objective to minimize tardiness. They considered sequence-dependent changeover time, non-identical parallel machines, finite release times for orders, restrictions on the types of orders that can be manufactured by each equipment, and variations in processing rates.

2.2.2 Mixed Integer Non-Linear Programming Algorithms

Suhami and Mah (1982) presented a generalized reduced gradient code, with the aid of a branch and bound search, to solve the MINLP for scheduling a multi-purpose batch plant. Wellons and Reklaitis (1989) formulated a MINLP based on maximizing the plant's profitability or productivity for a single-product production line. Dessouky and Kijowski (1997) developed a MINLP model for a single-stage, multi-product batch process. They assumed that all process times were identical regardless of batch size and that set-up and cleaning times were sequence-independent.

2.3 Heuristic Approaches

The majority of scheduling problems have been shown to be NP-complete, therefore it is not productive to try to enumerate all the possible scheduling combinations. For this reason, heuristics have been used to approximate the optimal solution to scheduling problems. Suhami and Mah (1981) introduced a branch and bound procedure to minimize the makespan of a batch flowshop scheduling problem with no intermediate storage. They used a heuristic rule to obtain an initial solution, which serves as an upper bound to the optimum. Ku and Karimi (1990) proposed a method for a serial multi-stage batch process with unlimited intermediate storage, minimizing the due date penalties. Roba and Teghem (1989) introduced a dynamic programming approach and a heuristic method to solve a scheduling problem involving machine pairs. They considered several non-identical machines to be used at a time. Elkamel et al. (1997) decomposed the multi-purpose scheduling problem, for a parallel machine, into smaller problems prior to heuristically solving the proposed MILP. They assumed that transfer of material was instantaneous, and if that was not the case, transport lines were defined as separate processing machines. Tandon, Cummings, and LeVan (1991) research is based on a non-permutation schedule, which is a unique contribution because most of the research has been based on permutation schedules. Permutation schedules are variations of the original schedule, i.e., jobs are rearranged and then follow the same sequence throughout the processing stages. They proposed a heuristic method for solving a serial, multi-product, batch scheduling problem under a non-permutation schedule, a single schedule "where the product sequence can be different through each of the [machines]."

Musier and Evans (1989) introduced a method that requires an initial solution. From there a heuristic improvement method (HIM) is applied and terminated when the solution cannot be further improved by local modifications. To increase the probability of finding the global-optimal or near optimal solution, the initial solution should be varied so that other local optima are considered. Musier and Evans obtained solutions for non-identical machines, sequence-dependent cleaning requirements, and process restrictions; then compared their approximation approach to an intuitive list scheduling technique. The technique lists customer orders by earliest-due-date, then orders them to obtain a schedule with the minimum number of tardy deliveries. They then used their HIM method to minimize stockout for multi-machine plants where customer orders may not constitute integral batch multiples (1991). This was a new idea for lot sizing research. Previously, sequencing decisions were made after lot sizes had been determined and trade-offs between changeover costs and inventory holding costs had been investigated (Graves 1981).

2.4 Stochastic Approaches

This section will review simulated annealing and tabu techniques used in chemical batch scheduling. Simulated annealing is an algorithm for single-objective multivariable optimization of a physical system (Metropolis et al. 1953) that provides an approximate solution to NP-complete problems. Ku and Karimi (1991a) and Das et al. (1990) presented evaluations of a potential simulated annealing method for solving multi-product batch process scheduling problems, which minimizes the makespan of a serial flow-shop with unlimited intermediate storage space. These results were then compared to other strategies, such as complete enumeration. Ku and Karimi (1991b) presented two algorithms that use an iterative improvement strategy to solve the serial multi-product process with a single batch machine in each stage with arbitrary intermediate storage polices. The first algorithm used simulated annealing, and the other built a good sequence from partial sequences; both minimized the weighted tardiness criterion. The second algorithm involves N iterations, where N is the total number of customer orders. At the end of n iterations a partial sequence of n products is produced. With each iteration, another order is inserted into the partial sequence minimizing the tardiness penalty. When all the orders have been inserted into the partial sequence, a good sequence (a feasible job sequence that minimizes tardiness penalty) has been formed. It was shown that simulated

annealing provides the best solution, but the partial sequence method is more computationally efficient.

Kurbel and Ruppel (1996) used simulated annealing for scheduling jobs on groups of machines for short-term production. They also implemented shift schedules for a more realistic analysis. Fortemps et al. (1996) presented a scheduler for a multi-stage, multi-product plant based on a priority list using a simulated annealing or tabu search method. Graells et al. (1996) defined a multi-purpose production schedule that used simulated annealing for matching deliveries with due dates. They simplify the scheduling problem by decomposing long recipes into consecutive mini-jobs to yield stable intermediaries and applied it to an academic scenario. Following this work, Graells et al. (1998) presented a flexible modeling framework, implemented into a basic general scheduling tool using simulated annealing, for both the study of scheduling problems and as an efficient way to find practical solutions in complex industrial applications.

Hindi and Toczylowski (1995) introduced a methodology for scheduling production in a cell containing several parallel facilities, which considered changeover times. The first step of the proposed method was to develop a number of elementary production schedules by ignoring changeover times. These were then ordered with the use of tabu search to minimize the total number of changeover operations. Finally, all operations were scheduled and the overall makespan was minimized. Tandon et al. (1995) presented a methodology to obtain a near-optimal solution for the problem of scheduling multiple products on a network of single-stage, unrelated parallel machines using simulated annealing with a performance criterion based on tardiness minimization. They compared their simulated annealing approach with the heuristic improvement method (HIM) of Musier and Evans (1989).

2.5 Simulation and On-line Approaches

In the past, batch and semi-continuous processes have been neglected in the computer-oriented scheduling literature, i.e., simulations and on-line scheduling techniques, despite the fact that they represent a considerable portion of the chemical industry and are used to produce some of the highest unit price products (Overturf et al. 1978). They were primarily neglected because of their large scale, which made them difficult to analyze and develop models for. However, more recently, scheduling problems have been solved with simulation models and on-line

scheduling techniques. The main advantage to using a simulation model to solve scheduling problems is that both the discrete and the continuous phenomena can be included in the simulation model and performance of the schedules can be estimated. On-line scheduling allows for detection of schedule deviations and modification of the schedule during production.

Von Watzdor et al. (1994) discussed different approaches to model and simulate integrated batch operations while Watson (1997) showed how to use discrete-event simulation (DES) to effectively analyze batch-processing systems. They developed a model that was successfully used in a real-world study to evaluate designs and operating policies for new facility proposals. Baker (1981) presented a generalized interactive-network formulation where the user was allowed to change the parameters of the model to obtain other production schedules if the first schedule produced was not desirable. The formulation was used in combination with a specialized branch and bound algorithm to minimize total costs for single-stage productions. Knoff (1985) presented a branch and bound procedure that uses a simulation model to evaluate optimal sequences for problems consisting of a single processor and its input queue, followed by N parallel intermediate storage units and then by a stage of P parallel machines. A modified Johnson's algorithm (heuristic rule) was used to determine the initial solution.

Parsons et al. (1992) proposed a strategy for recursively integrating discrete-event simulation (DES) with traditional predictive production scheduling tools to produce schedules that more closely model the actual production process. Van De Pol (1993) outlined MINLP Models and Constraint Satisfaction Programming Models in simulation format for a number of parallel machines at each processing stage. Baudet et al. (1995) developed a detailed job-shop DES model of a multi-purpose, multi-product batch plant with consideration given to human constraints and bi-product production. Wohlhalf et al. (1996) designed the Batch Process Simulation (BaSiP) model for recipe-driven production to reduce the load on the operators and to allow them to focus on the high-level tasks of assignment of equipment, sequencing, and scheduling. Music and Matko (1998) presented another model for recipe-driven production. Their model combines the procedures defined by the recipes with coordination and resource allocation functions. Finally, Valenzuela and Williams (1983) discussed a simulation for an economically-based scheduling heuristic.

A formal approach to on-line schedule modification was presented by Cott and Macchietto (1989) and applied to the computer-aided operation of a simple, batch flowshop.

They highlight the detrimental effects caused by varying processing times for fixed, off-line scheduling. Furthermore, they developed several schedule modification algorithms that improve the schedule performance by keeping batch times close to nominal values while recovering all possible machine idle time introduced by early completion. Clarke (1989) presented an on-line scheduling program based on the bottleneck of the system that uses just-in-time and optimized-production-technology concepts. In addition, Huercio et al. (1995) introduced a reactive methodology, based on a decision tree method, for multi-production and multi-purpose batch plants.

2.6 Other Approaches

Other approaches to the batch scheduling problem use a combination of the techniques discussed thus far and other scheduling methodologies. Methods for checking schedules for robustness and improvement techniques have also been developed.

Some of the combined techniques introduced are multi-model systems and the coupling of discrete event simulation with genetic algorithms and MILP. Artiba and Riane (1998) presented an architecture for a multi-model system that integrates expert system techniques, discrete-event simulation, optimization algorithms, and heuristics. Their system supported decision-making for complex production planning and scheduling problems. The scheduling technique was applied to a hybrid flowshop problem consisting of a series of production stages, each of which has several facilities operating in parallel, where the production environment was subjected to uncertainties, delays, and interference. Azzaro-Pantel et al. (1998) solved the job-shop scheduling problem with a two-stage methodology that involved coupling discrete event simulation with a generic algorithm. Gonzalez and Realff (1998) presented another coupling approach where MILP was used with discrete-event simulation to allow for variations in the production operations.

Several authors have introduced new scheduling methodologies. Doll and Whybark (1973) introduced an iterative procedure for directly determining near optimal frequencies of production for products and their associated fundamental cycle times. They were concerned with scheduling the production of several different products on a single machine. Salvador (1973) presented a dynamic programming approach that used branch and bound to search over permutation schedules, where the “product processing sequence is the same for all [machines]”

(Tandon, Cummings, and LeVan 1991). The objective was to minimize makespan of a production line with no intermediate storage. This algorithm was applied to a system with identical parallel processors in a given stage, without preemption. Henning and Cerda (1996) discussed, analyzed, stressed, and exploited the similarities between the plant design and the multi-product batch plant scheduling problem for their task-oriented knowledge-based approach. They designed their system so that new information extracted from manufacturing experience can be easily incorporated.

Sanmarti et al. (1998) solved a scheduling problem in a chemical multi-purpose batch plant, with consideration of transfer policies and changeovers, using a graph representation and branch and a bound algorithm. Martinez and Perez (1998) presented a project-oriented framework for interactive analysis of feasible plans and schedules that can provide an effective/efficient flow of orders within a batch process. Their method is based on a hierarchical project-oriented production model that integrates production planning with short-term scheduling and resource leveling. Kelle and Peak (1996) proposed an adaptive production scheduling policy for a pull-type system. The policy determined production runs by considering random demands and on-hand inventory. They determined the frequency of production and required safety stock to minimize set-up and holding costs subject to service level. Mauderli and Rippin (1980) developed the Batchman process for assigning equipment for scheduling multi-purpose production.

The following work was done to improve, evaluate, and make more robust schedules. Schumacher et al. (1997) introduced a method for improving performance of batch process operations with the presence of disturbances. Frauendorfer and Konigsperger (1996) introduced concepts that help evaluate production schedules in chemical processing environments. They also discuss how to determine the quality of feasible schedules and offer a decision support concept for schedule evaluation and improvement based on a special multi-criteria analysis approach. They evaluated non-preemptive, multi-product and multi-purpose production schedules, with a no wait policy.

A two-step schedule modification, for a multi-purpose batch-chemical plant within a single campaign, using least-impact heuristic beam search, was introduced by Kanakamedala et al. (1994). First, a decision tree was created to help identify possible rerouting of the product causing conflicts, then heuristic pruning of the search space was used to contain the

combinatorial complexity. Sanmarti et al. (1995 and 1997) presented a methodology for making a more robust schedule based on machine reliability. Their scheduling method does not guarantee an optimal solution, it only meets demand and due dates. Finally, Mignon, Honkomp, and Reklaitis (1995) presented a Monte-Carlo based simulation model to evaluate the robustness of MILP schedules. This model provides an estimate of how well the schedules will perform in reality. They compared individual terms of the objective function values of different simulation runs to identify the area that contributed to the degradation of the process performance. This type of evaluation allows us to examine the performance and robustness of a schedule that has been obtained via deterministic methods when it is executed in a stochastic plant.

2.7 Survey Conclusions and Future Directions

The literature survey presented in this chapter is by no means an exhaustive evaluation of the research that has been completed in addressing the chemical batch-process scheduling problem. Other surveys should be referenced for more topics relating to those presented here. Reklaitis (1982), Ku et al. (1987), and Rodammer (1988) presented reviews of scheduling problems studied in the chemical industry. Panwalker and Iskander (1977) presented a survey of rules that have been used to solve scheduling problems. They stated that a combination of simple priority rules or a combination of heuristics with a simple priority rule works better than individual priority rules. A comprehensive review of resource constrained problems was given by Blazewicz et al. (1991) and Reklaitis (1992). Rippin (1983a) provided a classification of scheduling problems that have been reported in the literature and the methods used for their solution. Roy and Meikle (1995) examined the role of discrete event simulation scheduling in improving business performance and suggest possible considerations in the development and implementation of a finite capacity scheduler.

Despite all the research that has been done in an attempt to solve the chemical batch scheduling problem, there still exists a need for new research to fill in the holes in batch process design and operational methodologies. New avenues for closing the gap between theory and practice are also needed (Rippin 1983b). Reklaitis (1982) and Ku et al. (1987) state that currently, there exists no single algorithm that can solve a general scheduling problem. Little is known about how the theoretical scheduling algorithms will perform in a real-world batch

processing environment. Therefore, the research presented in this thesis explores the real-world operations of several scheduling algorithms.

CHAPTER 3: PROBLEM DESCRIPTION

The previous chapter documented the literature related to scheduling algorithms. The research problem presented in this chapter was motivated by the results of Musier and Evans' (1989) survey of chemical industries and Reklaitis's (1982) review of scheduling operations. Musier and Evans' survey indicated that the most important similarity among the chemical industries was their objective to meet customers' due dates. The second most important similarity was in the plant process structures, which consisted of "a series of one or more processing stages with parallel [machines] (or production lines) at each stage." Reklaitis stated in his review that "such systems have received very little attention." He goes on to say that "the generalized flowshop problem can be expected to be quite difficult…[but] good approximate solutions are surely obtainable and should be found." A batch production plant was studied because they represent a large proportion of the chemical production facilities (Overturf, Reklaitis, and Woods 1978 and Rippin 1983). In summary, the research problem presented here focuses on production scheduling for a batch-chemical plant with a multi-staged, generalized serial network, minimizing total tardiness. Table 2 summarizes the focus and motivation for the research.

Batch Plant
large proportion of chemical production
(Overturf, Reklaitis, and Woods 1978 and Rippin 1983)
Generalized Serial Network
most common plant process structure
(Musier and Evans 1989)
Minimize Tardiness
most common objective in chemical industry
(Musier and Evans 1989)
Multi-stage
representative of how production really operates

Table 2: Research Focus and Motivation Summary

The industrial partner of this research is a primary manufacturer of defense products. A change from directed cost-reimbursable to competitively awarded firm fixed-price contracting by the U.S. Army meant that the company had to reduce costs to effectively compete for future contracts and retain jobs. The objective of the original simulation project was to reduce production costs, improve product quality and increase production volume of the primary solvent-based production line. Enhancements were added to the original simulation model in order to use it as a comparison tool for the research presented here.

Three scheduling algorithms were used to schedule a given set of customer orders and then the average total tardiness of each of the schedules was determined. Next the schedules were improved upon and the average total tardiness was once again determined. The schedules developed were used as input to the simulation model to determine how well they performed in the "real-world." Finally the results from the various schedules were compared to determine the best scheduling algorithm under investigation.

3.1 Problem Statement

The scheduling algorithm comparison problem can be stated as follows:

Given:

- an open flowshop with a generalized serial network with fixed intermediate storage (see Figure 2);
- a multi-product batch plant with fixed batch sizes dependent on transport vehicle size;
- 5 multiple processing stages with several out-of-phase parallel machines in each stage
    - first stage has 4 identical parallel machines (dehydration machines);
    - second stage has 3 identical parallel machines (mixers);
    - third stage has 2 identical parallel machines (blockers);
    - fourth stage has 3 identical parallel machines (extruders);
    - fifth stage has 4 identical parallel machines (cutters);
- the production line operates in a non-preemptive mode;
- products are produced in the same order for all stages;
- at any time, all machines of one type can only be used for one product;
- the machine and load, unload, transfer (LUT) time distributions for each stage;
    - LUT time is all the worker time, such as time to load, unload, and transport;
    - the process time is the sum of machining time and LUT time;
- product-dependent changeover times (deterministic);
- a set of customer orders and due dates; and
- a stochastic discrete event simulation model of the production line.

Assume:

- raw material is always available;
- all workers work continuously;
- the production line runs continuously;
- no machine break-downs or wear-and-tear;
- a static model, where all jobs are available at time zero;
- LUT times are product independent;
- the same type of distribution is used for the machine time for all products (the mean machine time is product-dependent but the form of the distribution is the same);
- each customer order only involves a single product;
- each order can be manufactured by the available equipment;

- each order may be processed at different times, i.e., no campaign requirement;
- each order comprises an integer number of full-size batches, all of the same size;
- the machine time for an order is product dependent; and
- a changeover period for cleaning and equipment setup is needed before producing a new product.

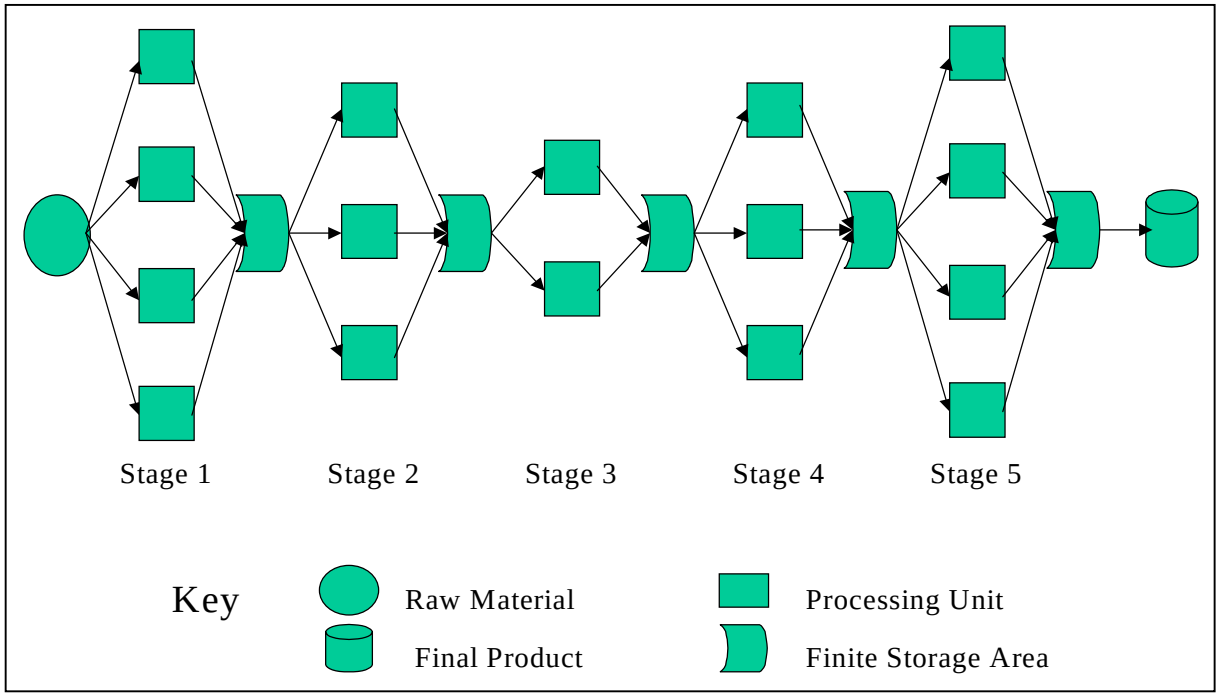

Figure 2: Generalized Serial Network Under Analysis

Objective:

Determine the best scheduling algorithm for a system with the given set of information and assumptions.

3.2 Research Strategy for Addressing the Problem

The research strategy for addressing the previously defined scheduling algorithm comparison problem involves the following activities:

- Gather necessary data for the scheduling problem.
- Modify or develop algorithms to suit the scheduling problem.
- Develop production schedules and improve them.
- Treat changeovers as dummy customer orders (Customer 0) and add them to the schedules as needed.
- Make enhancements to the discrete event simulation model.
  - Logic was added for determining which set of production times to use.
  - Logic was added for reading in schedules.
  - Calculation for determining the production schedule's objective function was added.

- Conduct simulation experiments.
- Determine the robustness and performance of each schedule.
- Compare the results using Multiple Comparison with the Best.
- Summarize the findings.

Data was collected from the company that the original discrete event simulation model was developed for, so the scheduling algorithms may be applied to a real-world industrial scheduling problem. The data was also used to maintain the accuracy of the simulation model. More detail about this data may be found in Chapter 4.

Two heuristic scheduling algorithms from the literature were modified and one greedy heuristic algorithm was developed specifically for the generalized flowshop under study. Musier and Evans' scheduling algorithm (1989) was developed for a single-stage batch production area with non-identical parallel machines. This algorithm was modified to fit a multi-stage batch production line with identical parallel machines. Ku and Karimi's Sequence Building Algorithm (1991) was designed for a serial multi-product process with only one processing machine in each stage. They assigned cost penalties as a function of the order tardiness, which is not considered in this research. However, since they assumed that the processing order for all stages was the same, their algorithm, for the most part, has been applied directly to the generalized serial network under consideration. The greedy heuristic algorithm developed was based on an earliest-due-date approach. Changeovers were treated as dummy customer orders and were placed in the production schedule as needed.

Once the algorithms were modified and developed, production schedules were constructed, then improved using Musier and Evans' heuristic improvement method (1989). The deterministic objective function values for all the production schedules were calculated and the computing time for developing each schedule was determined. Chapter 5 provides a detailed outline of the three scheduling algorithms and the improvement method applied.

The existing discrete event simulation model was designed for processing a single-product and did not have the ability to read-in production schedules. Therefore, the model was modified to allow for product-dependent machine and changeover times. A method for reading in production schedules was also added to the model. Chapter 6 provides a detailed explanation of how the simulation model was developed and the type of production line it represents. It also discusses the changes that were made in order to use it as a comparison tool. Upon the completion of the necessary additions to the simulation model, experimental runs were

conducted using the developed production schedules and the corresponding stochastic objective function values were determined, along with the required simulation time.

Mignon, Honkomp, and Reklaitis' evaluation technique (1995) and the Multiple Comparison with the Best method (Swisher & Jacobson 1999a) were used to determine the robustness and performance of the schedules. Chapter 7 gives a detailed explanation of how these two techniques work.

Finally, Chapter 8 provides a summary of which of the algorithms had the best performance, based on: objective function value, robustness, performance, required computing time, required simulation time, and throughput. Conclusions were drawn from the summarized results.

CHAPTER 4: DATA REQUIREMENTS

The data necessary for developing the production schedules and simulation model were obtained from the company that sponsored the original simulation model. The following data was collected for the research:

- LUT (load, unload, and transport, etc.) times and distributions;
- machine times and distribution for each stage and product type; and
- changeover times for each product.

4.1 Times and Customer Orders

The industrial partner provided average machine and changeover times but was unable to provide the distribution of these times. A time study was conducted for the largest volume product, which provided enough data to fit distributions for the machine times and LUT times. The shape of those distributions was used for all products after adjusting the mean. In other words, the given average machine times, for each of the product types, were used as the mean value for the fitted distributions in order to provide stochastic machine times. Unfortunately, no changeover time distributions currently exist; therefore the changeover times were assumed to be deterministic. Table 3 provides information about the average machine and the changeover times. Table 4 provides the average LUT times.

		Average Machine Time* per Batch**						Total Time* for
Product	Type	NIR	Dehy	Mixer	Blocker	Extruder	Cutter	Changeover***
1	1	3.53	36.83	18.00	3.80	3.57	1.86	240
	2	3.53	36.83	28.00	3.80	3.57	1.86	240
2	3	3.53	36.83	50.00	3.80	3.57	1.86	300

* Time in minutes. ** Dehy & Mixer Batch = 500 lbs; Blocker Batch = 250 lbs; Extruder & Cutter Batch = 83.33 lbs *** Fixed time for changeovers.

Table 3: Average Machine and Changeover Times

	Average LUT Time* per Batch**							
Operator	Pre-Load	Load	Unload	Data	Walk	Weigh	Transport	Direction
NIR	N/A	0.95	0.55	1.58	0.19	N/A	2.56	from Dehy
Dehy	4.39	4.39	3.40	N/A	N/A	N/A	3.15	to Mixer
Mixer	N/A	3.51	6.70	N/A	N/A	N/A	11.56	to Blocker
Blocker	0.25	1.18	1.16	N/A	N/A	N/A	3.17	to UpExtru
UpExtru	N/A	1.09	1.06	N/A	N/A	N/A	N/A	N/A
DwExtru	N/A	0.74	0.56	N/A	N/A	N/A	0.56	to Cutter
Cutter	N/A	1.59	1.23	N/A	N/A	0.27	1.62	to Badger Buggies

* Time in minutes. ** Dehy & Mixer Batch = 500 lbs; Blocker Batch = 250 lbs; Extruder & Cutter Batch = 83.33 lbs

Table 4: Average Load-Unload-Transport Times

Two sets of fictitious customer orders and due dates were developed. Table 5 contains fictitious information used for this research.

Customer #	Product Type	Due Date	Total Dehy Batches
1	1	16	20
2	3	20	4
3	1	13	24
4	2	56	56
5	2	68	80
6	3	55	4
7	1	150	68

Customer #	Product Type	Due Date	Total Dehy Batches
1	1	1	4
2	2	12	24
3	3	48	8
4	2	34	16
5	1	8	32
6	1	52	8
7	2	79	60
8	3	155	12
9	3	85	4
10	1	61	12
11	2	12	36
12	3	201	48
13	3	37	20
14	1	8	44
15	3	96	32
16	1	92	40
17	2	66	8
18	1	73	12
19	2	49	48
20	1	18	60

Table 5: Customer Orders

4.2 Processing Time for Deterministic Model

The first part of the research was to determine production schedules using average processing times (machine + LUT times for each stage). However to develop more accurate deterministic models, it was necessary to account for the waiting time prior to each stage caused by stochastic variations in run times at each machine. Queueing theory was applied to account for the time spent waiting to be processed at each machining area. It was assumed there is an infinite queue of raw material waiting at the first stage of production. The output rate at this stage is then the same as the service rate, μ_1, which is also the arrival rate at Stage 2, i.e., $\lambda_2 = \mu_1$. Waiting time is not considered at Stage 1.

It was assumed that both the interarrival and the service times followed an exponential distribution. Since there were finite intermediate storage areas for each stage, a M/M/s/k queueing model was used to model each stage with the following terminology and notation (Hillier and Lieberman 1995):

λ mean arrival rate (expected number of arrivals per unit time),

μ mean service rate per server for overall system (expected number of customers completing service per unit time),

P_n	probability of exactly *n* customers in queueing system,
L	expected number of customers in queueing system,
L_q	expected queue length (excludes customers being served),
ST	expected time in system (includes process time and wait time for each individual customer),
W_q	expected waiting time in queue (excludes process time) for each individual customer,
s	number of servers, and
k	maximum number of customers allowed in the system.

It is assumed that $s \leq k$. Performance measures for the M/M/s/k queueing system are given by the following equations:

$$P_0 = \frac{1}{\sum_{n=0}^{s} \frac{(\lambda/\mu)^n}{n!} + \frac{(\lambda/\mu)^s}{s!} \sum_{n=s+1}^{k} \left(\frac{\lambda}{s\mu} \right)^{n-s}} ,$$

$$P_n = \begin{cases} \frac{(\lambda/\mu)^n}{n!} P_0 & for\ n = 1,\ 2,\ ...,\ s, \\ \frac{(\lambda/\mu)^n}{s!\, s^{n-s}} P_0 & for\ n = s,\ s+1,\ ..., k, \\ 0 & for\ n > k, \end{cases}$$

$$L_q = \frac{P_0 (\lambda/\mu)^s \rho}{s!(1-\rho)^2} \left[1 - \rho^{K-s} - (K-s)\rho^{K-s}(1-\rho) \right] ,$$

$$L = \sum_{n=0}^{s-1} nP_n + L_q + s(1 - \sum_{n=0}^{s-1} P_n),$$

$$ST = \frac{L}{\lambda}, \qquad W_q = \frac{L_q}{\mu},$$

where $\rho = \lambda/(s\mu)$. Table 6 summarizes the process times used as input for the queueing model.

		Average Process Time* ** (Machine + LUT Times) (μ_i)				
Product	Type	Dehy/NIR	Mixer	Blocker	Extruder	Cutter
1	1	61.52	39.77	9.55	7.58	6.57
	2	61.52	49.77	9.55	7.58	6.57
2	3	61.52	71.77	9.55	7.58	6.57

* Time in minutes. ** Dehy & Mixer Batch = 500 lbs; Blocker Batch = 250 lbs; Extruder & Cutter Batch = 83.33 lbs

Table 6: Average Process Times Used for M/M/s/k Analysis

The arrival rates for the remaining stages, λ_i for i = 2, 3, 4, 5, were based on the effective service rate, $\overline{\mu_i}$, of the previous stage, where:

$$\overline{\mu_i} = \mu_{i-1} \sum_{n=0}^{\infty} P_n ,$$

μ_{i-1} service rate of previous stage for *n* customers,
P_n probability of exactly *n* customers in queueing system.

A sample calculation of the effective service rate for Stage 2 is shown on the next page.

The arrival rate for each stage was adjusted to account for the varying batch sizes. For example, one mixer batch is processed as two blocker batches. The resulting arrival rates are:

$$\lambda_2 = \mu_1 = 0.0650,$$
$$\lambda_3 = 2\overline{\mu}_2 = 0.1274,$$
$$\lambda_4 = 2\overline{\mu}_3 = 0.0857, \text{ and}$$
$$\lambda_5 = 6\overline{\mu}_4 = 0.2389.$$

The mathematical calculations resulted in the expected system times, the time required for a batch to wait to be processed and to be processed in a given stage, shown in Table 7.

M/M/s/k

			ST* per batch** for each product		
Stage	s	K	1	2	3
1: Dehy	4	N/A	61.52	61.52	61.52
2: Mixer	3	15	85.42	162.35	316.56
3: Blocker	2	3	20.35	19.75	17.68
4: Extruder	3	4	15.07	14.75	13.45
5: Cutter	4	8	6.78	6.76	6.68

* in minutes ** varies from stage to stage

Table 7: Expected System Times

The system times, ST, shown in Table 7 were then adjusted to represent the actual time required to process one dehy batch at a stage when all the available machines in each stage were operating

Stage 2: Mixer

lamda = 0.065021 batch/min
mu = 0.025145 batch/min
s = 3 Machines
roh 0.861962 (lamda/(mu*s))
k = 15

n	mu	Cn	sum(rohs)^(n-s)	Probability that exactly n customers in queueing system Pn	Expected queue length (excludes customers being served) Lq	L	Expected waiting time in queue (excludes service time for individual customer) Wq	Expected waiting time in queue (includes service time for individual customer) W	mu (bar)
0	0	1		0.040355512		0			0
1	0.025145	2.5858873		0.104354804		0.104354804			0.002624
2	0.050289	3.3434065		0.134924879		0.269849759			0.006785
3	0.075434	2.8818907		0.116300175					0.008773
4	0.075434	2.4840815	0.861962419	0.100246381					0.007562
5	0.075434	2.1411849	0.742979213	0.086408613		sum(n*Pn)			0.006518
6	0.075434	1.8456209	0.64042016	0.074480977					0.005618
7	0.075434	1.5908559	0.55201811	0.064199803					0.004843
8	0.075434	1.371258	0.475818866	0.055337818					0.004174
9	0.075434	1.1819728	0.410137981	0.047699119					0.003598
10	0.075434	1.0188162	0.353523526	0.041114848					0.003101
11	0.075434	0.8781813	0.304723994	0.035439454					0.002673
12	0.075434	0.7569592	0.262660631	0.030547478					0.002304
13	0.075434	0.6524704	0.226403593	0.026330778					0.001986
14	0.075434	0.562405	0.195151389	0.022696141					0.001712
15	0.075434	0.484772	0.168213163	0.01956322					0.001476
			5.194013047		2.9102	5.4455	45.6503	85.42	0.063749

These values are calculated based on the formulas given on the previous pages. The effective service rate for Stage 2 is $\overline{\mu}_2 = 0.0637$ batch/mix; so the arrival rate for Stage 3 is $\lambda_3 = 0.127\,4$ batch/mix.

in parallel. The required time to process a dehy batches of Type 1 for each machining area was determined as shown below and summarized in Table 8.

- Since there are 4 dehy machines, for every 61.52 minutes, 4 batches are produced. Hence, the processing time for one batch is 61.52/4 = 15.38 minutes.
- Since there are 3 mixers, for every 85.42 minutes, 3 dehy batches are produced. Hence, the processing time for one batch is 85.42/3. Note, each mixer processes one dehy batch.
- Each mixer batch is processed as two blocker batches. So, for every batch that is processed at the mixers, two batches arrive at the blockers. Therefore, in 20.35 minutes only half a dehy batch is processed by the blocker. Hence, in order to process a dehy batches it will take 2*20.35 = 40.70 minutes.
- Similarly, the extruders can only process half a dehy batch in 15.07 minutes (it requires all 3 extruders to process half a dehy batch). Therefore, 2*15.07 = 30.14 minutes are required to process one dehy batch. Note that for every dehy batch processed by the extruders, 6 cutter batches are produced.
- For every 6.78 minutes, 4 cutter batches are processed. But in order to process an entire dehy batch, it will require ⌈6 cutter batches/4 cutter batches per 6.78 min⌉*6.78 = 13.56 minutes.

Type 1

Stage	Batches / ST*	ST* / Batch	Formulation	ST* / Dehy Batch
1: Dehy	4	61.52	61.52 / 4	15.38
2: Mixer	3	85.42	85.42 / 3	28.47
3: Blocker	0.5	20.35	20.35 / 0.5	40.70
4: Extruder	0.5	15.07	15.07 / 0.5	30.14
5: Cutter	0.67	6.78	6.78 * 2	13.56

* in minutes

Table 8: Summary of Calculation for System Time for on Dehy Batch

This resulted in the deterministic processing times for one dehy batch in each machining area shown in Table 9.

	ST* per dehy batch		
Stage	1	2	3
1: Dehy	15.38	15.38	15.38
2: Mixer	28.47	54.12	105.52
3: Blocker	40.70	39.50	35.36
4: Extruder	30.14	29.50	26.90
5: Cutter	13.56	13.52	13.36
Total	128.25	152.02	196.52

* in minutes

Table 9: System Time for one Dehy Batch