11-20-2006

Prediction of job completion times and optimal overtime allocation for satisfying production due dates

Olivia Liu

Rochester Institutes of Technology

PREDICTION OF JOB COMPLETION TIMES AND OPTIMAL OVERTIME ALLOCATION FOR SATISFYING PRODUCTION DUE DATES

A Thesis

Submitted in partial fulfillment of the

requirement for the degree of

Master of Science in Industrial Engineering

in the

Kate Gleason College of Engineering

Industrial and Systems Engineering Department

By

Olivia Liu

B.S.; Industrial Engineering, Cal Poly San Luis Obispo

November, 2006

DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING

KATE GLEASON COLLEGE OF ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Olivia Liu

has been examined and approved by the

thesis committee as satisfactory for the

thesis requirement for the

Master of Science degree

Approved by:

____

Dr. Michael E. Kuhl, Thesis Advisor

____

Dr. Moises Sudit

Abstract

One of the important aspects contributing to the competitiveness and success of a manufacturer is the efficient management for timely order delivery. After production orders are scheduled, there arises the need of a support tool to aid in the analysis with the available information, and to support managerial decision making which ultimately aims at on-time delivery. One way in which companies can meet due-dates of orders that are in jeopardy of being late, is to schedule overtime. This research presents a method used for 1) predict the completion times of scheduled jobs; and 2) optimizing overtime allocation when delays are foreseen. Mathematical mixed-integer linear program models are developed to represent the above problems for a tandem production line with single machine work stages. Non-operational downtime occurrences are considered in the production horizons which can be varied by work stage. Buffer areas (queues) are also included in the production system. These MILP models are solved using commercial optimizer ILOG-OPL studio. Using VBA script with OPL, a friendly interface is built in MS Excel for ease in user manipulation. The interface can also be used in production test to hypothetical "what if" questions. The models are verified using simulation. Runtime evaluation is also preformed to determine the capabilities and limitations of the models.

Table of Contents

1 Introduction

This research was initially inspired by real production scheduling issues encountered at a local machine manufacturing site. A Xerox production manager wanted to find a tool that would provide WIP (Work In Process) job completion times at each station. The desired tool would also be able to determine the best expedited production schedule, including overtime hours, to ensure on-time delivery when there were uncontrollable or unexpected delays preventing timely deliveries. The above application is an example of a resource and production planning situation in which information is needed to support managerial decision-making. A goal of a production manager is to achieve efficiency in production management and to be able to make effective decision with available production data.

However, computing prediction and overtime allocation scheduling information is currently a complicated task in the practical manufacturing environment. The manufacturing environment is highly competitive due to its nature of low technology, ease of duplication, and low profit margin. Manufacturing processes must also change to meet the customer and market needs. Due to the variety of demand requirements, high-mix-low-volume (HMLV) production in which small quantity orders are received is becoming one of the important types of production systems. RMA (Return Material Authorization) production which repairs customer returns products is a good example of HMLV production. Other examples include prototype manufacturing such as large printed circuit board designed for large automatic test equipment, and the aircraft assembly business. Make-to-Order (MTO) manufacturing also fits into the same category since only customized orders are taken. A leading MTO company is Dell, Inc.

(Reporter, 1998), which assembles computer systems based on customer needs. Another MTO example is Lands' End, Inc. (Parente, Venkataraman, Fizel, & Millet 2004), with the company's customized cloth order line. Dell and Lands' End have in common small quantity orders, and mixed order types where different parts and processes are required in production. Obviously, such production systems are challenges to their production managers. Lead time and job completion prediction dates are difficult to foresee even by experts with extensive work experience. It is difficult due to concurrent job processing stages and various related processing times. The difficulty is also increased when a job family is complicated, and frequent changeovers are needed. In these kinds of production system process flows, to balance the jobs evenly on the production line as a continuous flow line is almost impossible to achieve, therefore job completion time or lead time can not be done by simple computation. Consequently, this exaggerates the degree of difficulty for production managing, and computer guidance is needed to use to support production decisions.

Secondly, in addition to the difficulty of predicting job completion time and overtime expedition scheduling caused by the complexity of the job orders and process flow, uncontrollable issues also make the production managing more complicated. Uncontrollable issues include but are not limited to: unexpected absent employees or employee unavailability; various actual processing times due to different employee performance rates; and even unexpected machine breakdowns and accidents. All of these can change or delay job completion times and may require changes in overtime expedition scheduling. These issues can largely affect previously scheduled job completion dates, especially when there are long non-operational hours in the production

horizons. Due to the fact that many production floors are operated in one to two shifts a day, five days a week, thus leaving idle production hours. Such non-operational production hours should be included in production schedule horizons when conducting research on job completion time and overtime expedition methods to return better and more precise data. Not considering non-operational hours in the production horizon makes the implementation of a research method in the practical production floor difficult as well as unrealistic and overtime scheduling becomes impractical.

Depending on the industry, some production lines set up buffer areas. Buffer areas, also known as work in process (WIP) queuing areas, are used to temporarily hold unfinished incoming jobs and allow a previous stage to process the next job without blocking the line. In production floors where line balancing cannot be achieved, setting up buffer areas may be helpful. Buffer areas in an unbalanced line can helps to reduce unnecessary machine idle time, and thus reduce possible waiting time due to work stage blockings. However, buffer areas also make the determination of job completion time and overtime scheduling more complicated.

In practice, when attempting to utilize resources and maximize productivity management one may regularly ask "what if" questions. Some of questions are like "What if overtime is planned at that stage will the shipment be completed two days earlier?", or "What if additional WIP is allowed at in a the stage queue will it helps with the machine idle time?" Managers use these "What if" hypothetical questions to learn how resources interact by changing input data and observing results, and to create understanding of the floor activities and the resources interaction. In addition to the job

completion time prediction and overtime allocation, managers also need a tool to help them to get answers for those "what if" questions they have in their daily work.

This paper focuses on finding a method that can predict job completion times as well as optimal overtime allocation for satisfying production due dates for a buffered production line in which mixed orders are received. The research takes non-operational downtime in the production horizon into consideration. The research aids in answering various "what if" questions, which are valuable to production managers and planners. A user interface associated with the research method will be designed to provide output information to managers. This methodology will help managers to obtain necessary information to assist and support their decision-making, as well as aid in meeting timely deliveries of products to the customer as promised.

2 Problem Statement

Currently, it is not difficult to collect production data. New computer applications make shop floor tracking more accessible than ever before. Production data can be automatically collected during job processing. A task's standard process time can also be obtained with the popular time motion study method. These collected data are the essential input data needed and used by production management in their daily production data analyses.

The ability to predict job completion times and ability to schedule expedited production with overtime hours to meet production due dates gives production managers better foresight and provides strong support to their managerial decision-making. The research described in this paper has sought to find a solution to the problem of finding an efficient operations research method to obtain following goals for flow shop production systems: (i) to predict job completion times; (ii) to find optimal solutions for overtime allocation to meet production due dates.

The method under investigation uses a mathematical linear programming (MILP) model to represent the discrete production systems. Input data used in this research includes deterministic standard job processing times for fixed incoming sequence jobs at each stage, production operation horizons with non-operational downtimes, job completion due dates, and WIP buffer area sizes, if any.

Research is conducted on a flow shop type production system with single machined work stages in a tandem line which may or may not have a buffer area at each work stage. Incoming jobs are in fixed sequence order. Jobs are processed sequentially in

every work stage. Varied production horizons can be set for each work stage, and different non-operational downtime hours can be set up in each horizon.

The followings assumptions are made in this MILP model:

1. Deterministic standard job process times and non-operational downtimes are given as input data. For WIPs, process times are updated to the remaining process times. In addition, in order to obtain process times that are equal to the actual process, job process times can be modified with consideration of operator performance rate. Actual process time = Standard process time / Performance percentage rate. Process setup time, if any, can be included to the process time as well.
2. Production process blocking is permitted. A job can't start if the following three conditions are not met: a) The job must be available for processing; b) a server (and machine) must be available; c) the server may not hold up a finished job.
3. Non-operational down times pre-empt jobs. All jobs that are interrupted by downtime can continue its rest of the operation task without redoing the whole task.
4. Single jobs are processed on the production line. However, for batch process as well. A batch can be considered as a job when it comes to forecasting production activity.
5. Overtimes, if needed, are added at the end of the up time. All overtime will be scheduled during the non-operational downtimes.

This mathematical linear programming modeling method will be solved with the commercial optimizer ILOG OPL studio/ CPLEX. The method will be validated and

verified with simulation using the Arena simulation software. Evaluation of the method's performance will be discussed. Lastly, there is a user friendly MS Excel user interface set up to execute the models. The CPLEX program can be automatically launched and processed after input data is ready and a command on the spreadsheet is given. Output data will be presented to the user in the same spreadsheet for ease in editing and data analysis.

3 Literature Review

The ability to predict accurate job completion times greatly helps the production manager to make correct decisions leading to positive growth in business, especially in today's dynamic and highly competitive manufacturing environment. Managers rely on guidance of computer tools for information they need. Accurately predicting production job completion times is not new and much research has been conducted in this area.

3.1 Production Process Flows

Categorizing production system by its flow structure, a process can be briefly classified as either a job shop or flow shop. A job shop is highly flexible with its general purpose resources, compared to a flow shop where specialized resources follow a fixed path. More specifically, production system flow categories can be split into the following: Project, Job shop, Batch process, Assembly line, and Continuous flow. The following table summarizes the characteristics of each type.

Table 3-1 Production flow types & example

Process Flow	Job Shop	Batch Process	Assembly line	Continuous flow
Example	Machine Shop	Bakery	Vehicle assembly	Automatic PCB line
Flow	None --------------------------------→ Continuous			
No. of Products	High --- --------------------------------→ Low			
Volume	Low ------------------------------------→ High			

Batch process, assembly line and continuous flow productions are generally called "flow shop". Unlike project and job shop, flow shops generally have a fixed pace and fixed sequence of activities. The batch process produces the product in batches. The process may requires setup time, and allow mixed products. Production activities don't have to be connected. The assembly line process works in discrete steps with the

machine in-line while the continuous flow process work in pace and normally carried by a connected conveyor.

The choice of process is primarily based on the product variety and volume. It also depends on the marketing, business strategy, stages of product cycles, as well as the local economy, labor cost and equipments available.

3.2 Existing Completion Time Prediction

Computer simulation is very popular and has been wildly applied in predicting job completion times and in finding completion time probability distributions, as well as completion time analyses. There are many computer software programs for simulation available on the market. To name a few: *Arena, Promodel, Extend, AutoMod and Witness*. By correctly constructing models, computer simulations can imitate the operation of real-world taking into consideration process overtime, and also the non-operational hours in the production horizon. Another advantage of simulation is that it allows users to see how variables or resources interact by changing input data and observing results. A method was studied to simulate expected occurrence of the uncertainty variables, as well as to compute the completion time probability (Ahuja,1985). Due to the nature of program language and design, simulation also has disadvantages. It usually takes users a long time to set up production models, and it is inflexible in searching for optimal solutions, and may require a lot of manual manipulation when it comes to optimization problems. In this research, the simulation program *Arena* is used to build a model to generate verification test random data.

Job completion times can also be calculated for well balanced production lines using a spreadsheet. Obtaining job completion time probability distribution using a spreadsheet

method was introduced (Johnson, 2002). This method was studied on a paced production line where jobs at each stage are well balanced and processing times are consistent at each stage. By knowing the product service cycle time, coefficient of variation, and number of stations, the spreadsheet is formulated to calculate, and yield the last stage job completion time distribution as the output. The output from Johnson's spreadsheet calculation was very close to results from system simulation. However, this method is not beneficial to a complicated production line such as an assembly line or a batch process. The Excel formulation doesn't consider buffer area or non-operational hours in the production horizon. In addition, how to obtain accurate processing time distribution is still an issue when it comes to input data collection.

The neural networks method is also used in job completion prediction. Fuzzy duration in neural networks was used to study job completion time (Feygin,1985). RETE algorithm and RETE network, which implement a rule based knowledge system with memory nodes, was also be used in the study of production run-time prediction (Barachini, Mistelberger and Gupta, 1992). A hybrid method of back-propagation network combining with genetic network (Li, 2005) is applied to improve accuracy of the completion prediction for batch process production. Li also analyzed a batch production line with no buffer queue. Focusing on using genetic network to optimize the neural network's parameters, he gained some degree of improvement in prediction accuracy. The main advantage of the neural network is that it can automatically detect complex non-linear relations between the job set characteristics and the completion time, providing for more accurate prediction. However, parameter design in complex neural network requires trial and error.

3.3 Prediction with Linear Programming Representation Model

Another way to predict job completion times is by using an optimization method setting up linear programming models. Linear program constraints are used to represent production floor activity movements, with the objective of minimizing job completion times. Some research has been done using this method. Zhang, Yin, Liu and Linn (2003) proposed two optimization algorithms. They used the heuristic method to minimize completion time in a two-stage hybrid flow shop using identical machines at the first stage, and a single machine at the second stage to study scheduling problems. Ambuhl and Mastrolilli (2004) studied a method to minimize total processing time by using a pre-emptive algorithm. This model splits a production lot into sub-lots in order to accelerate the progress of an order in production. This split method was done on a two work stages production with multiple machines in each stage where the job can be equality split. Chen, Deng and Zang (2004) also used a linear programming optimization method for a batch process production scheduling problem. Using linear programming formulation to represent the production system is very powerful, due to the flexibility of constraint formulation and model setup. The linear programming model can also be used to answer the "what is" questions by changing input parameter data in some models.

Event graph is another method using linear programming by partitioning a system into events represented by vertices, and uses directed edges to represent relationships between events. Event graph models production system with a state transition diagram systematically (Ross, 1993). Professor Lee Schruben was the first person to apply event graphs to represent discrete event production systems in the study of simulation modeling. Most of the schedule literature constraints previously were generated using

intuitive arguments. (Chan, 2004). Schruben started the research using mathematical programming to represent discrete event relationship graphs in 1983. The event relationship graph was applied in representing the work information flow in networking (Liu, Fan & Li, 2003). Research done by Schruben and his student Chan includes applying the event graph in a model representing discrete event simulation(DES), and also mapping discrete event systems event graph relationship with mathematical programming. Their published papers show the research in applying linear programming to represent flow shop queuing networks with finite buffers for different blocking scenarios in a continuous time horizon (Schruben & Chan, 2003, 2004 & 2005). They also completed and improved a general approach for translating DES as mathematical programs by eliminating redundant constraints, and enabled a better model to be used in simulation (Chan, 2005). These studies were all done in the continuous production time horizon in which non-operational downtimes were not included nor considered.

The relevant works which also use linear programming formulations to represent systems can be categorized into the following: 1) performance analysis of timed Petri nets using linear or integer programming and 2) linear system theory of discrete-event system. The method using Petri net to formulate a model is different form using an event graph method. The first category, Petri-nets linear programming formulations, is based on the state equation and the incidence matrix of the nets A linear program model was developed to represent a timed window event using a negative token (Lee, Seo, and Park, 2002). With the second category, linear system theory, state equations are formulated intuitively, and issues like "found no literature that address on how to obtain the recursions systematically or methodically" were questioned (Chan, 2005). Both differ

from Chan and Schruben's event graph mathematical representation approach in the means of deriving equalities and inequalities, and the way the equalities and inequalities are being used. Representing systems with linear programming formulation using the event graph method allows the models to be analyzed using mathematical programming techniques. They also have the advantage of acting as a connection between the mathematical programming and simulation integrating optimization into simulation (Chan, 2005). The traditional method of using simulation to predict optimization is very different from this new formulation representation. There is very little literature related to the event graph mathematical programming representation approach used for production scheduling problems or production related studies that can be found beside Chan's research in cluster scheduling (Chan, 2004).

In practical production, the primary method employed to meet due dates is through the use of overtime. Many papers that include overtime variable in the research topic primarily used it as a variable to calculate the overall cost. Research done includes using overtime as an element for the purpose of capacity and lot size optimization for a single machine facility that associates setup time (Ozdamar & Birbil, 1997), and in a single machine make-to-order environment for a lot size problem (Dellaert & Melo, 1997). The overtime scheduling problem was conducted in one machine production in these papers. Overtime scheduling in packaging industry was studied (Lagodimos& Mihiotis, 2005). A full packing line was considered as a unit for overtime schedule with multiple lines paralleling in the facility. This problem actually is a multiple parallel lines scheduling problem which is NP-hard in a strong sense. The objective of this paper was to reduce the

cost of manpower in regular and overtime shifts. A linear program model was constructed with constraints and a heuristic method was applied in the research.

3.4 Decision Support Systems

Research results will only be beneficial to practical production after applying the methods, implementing to actual production, and generating information needed by the users. Though there is overwhelming research that has been developed, not all are applied in practice as a tool supporting in decision making. One of the primary reasons is lacking of a good interface. The important factors for a good application tool include, but are not limited to: input data generation and data input, output data viewing and analysis, solution accuracy and flexibility to adapt to production changes. An interface is normally set up for users to run the method and to collect data. The user system can be large to combine with many research methods in one user interface for different purposes or just use method. Support systems with user friendly interfaces are expected to be intelligent and dynamic with information displayed in appropriate way and the manipulation is easy and friendly. Some production planning and control production support decision system research was done based on simulation, artificial intelligence in batch manufacturing (Groenskov, 1996), constraints in aircraft scheduling (Esquirol, Lopez, Haudot & Sicard, 1997), and heuristics in nurse scheduling (Furuhashi, Maeda, & Takaba, 2003). Others are based on very hybrid systems. All of the interfaces have in common that information is displayed properly and easy to operate.

4 A Study of Job Completion Time Prediction & Overtime Allocation

As previously mentioned, job completion time prediction is important, as well as critical to production management. Prediction of a continuous production time horizon is difficult, and considering non-operational hours in the production horizon is even more complex and hard to achieve. In practical production non-operational hours are "musts" in planning for none production line can run continuously without downtimes.

4.1 Production System Description

This research is conducted on so called flow shop where multiple stages are lined with one machine at each stage. Every incoming job has one or less operation on each machine. Incoming jobs are in fixed order and are processing subsequently as they arrive at the production line. There are chain precedence constraints between operations, so that a job cannot skip any operation in a stage or move ahead of the other jobs even when there is no operation process time in a specific stage.

This is a G/G/1 queuing system in which the inter-arrival times and operation process times are governed with deterministic given values. Buffer areas have finite capacities. Each of the stages has one machine and also its own production horizon, which can be different from other stages. A process flow chart for a production layout with five functional stages and one machine in each stage is displayed below in figure 4-1.

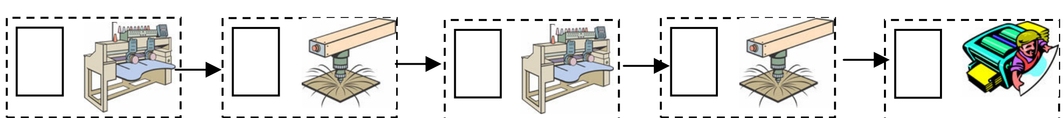


Figure 4-1 Five stage process flow diagram

The dash boxes indicate the differential functional stages in the production route. At the starting point, jobs either arrive to the first stage (receiving area) or wait in the buffer areas. All jobs or orders are processed in the stages in subsequent order until all tasks in all stages are completed. At each stage, jobs are completed in the fixed incoming order sequence, a job cannot start until the previous job has completed. The solid boxes inside the functional dash boxes mark the queuing buffer waiting areas (if applicable) in each stage. Every machine has capacity of one. We assume that each machine has its own production horizon and downtime schedule, which can be the same as the other stage schedule or it can be different.

The rules that describe how a job is moved in the production line is called the blocking policy. There are a few common production blockings, such as Kanban blocking, communication blocking, and production blocking. Due to the nature of this production layout and our research, only production blocking is discussed in this research. A job must meet the following three conditions to be moved with production blocking policy:

1. A job is available for processing; it completed the previous operation.
2. A server (operator and the machine) or a waiting queue space is available.
3. The server is not holding a finished job.

A production with no buffer areas will be first system to investigate in this research. A mathematical linear program model is set up to reflect the production activity movements and status; and then complexity is added to investigate the model for a production with buffer areas. At the next phase, we investigate an overtime allocation

model. Lastly, a simple system with friendly user interface is set up to allow flexible user manipulation, system modification, as well as data viewing and analysis.

In practical production, most production lines in a flow shop do not run continuously. Most small to medium size factories operate one or two shifts per day. In addition, production floor activities normally can be interrupted and stopped for scheduled maintenance, auditing and other unexpected issues, which is generally named as non-operational down times. Other possible non-operational time can be operator lunch breaks and changeover between shifts. In addition, non-operational hours can be caused by the lack of incoming jobs and resources. Depending on the industry and nature of the manufacturing, some productions may allow flexible working schedules where the non-operational downtimes are set up by the work stage.

Since a job's completion time depends on the job completion times in the previous stages and previous jobs, the non-operational hours in each of the stages usually needs to be considered. Non-operational hours in combination with the job processing information, determine the final job completion times, which can be dramatically varied compared to the prediction without consideration of downtime hours. When non-operational time should occur, it is possible that even a slight change in job processing time can result in a big delay in the final job completion time due to the downtime interruption. Therefore, it is important to include downtimes and non-operational hours in the production horizon when it comes to research for job completion time prediction.

4.2 Research Methods Conducted

Several methods were attempted in searching for the job completion time prediction methodology. The three methods investigated that failed to successfully establish MILP with include PERT, Event Graph, and Discrete Time Optimization.

4.2.1 PERT Method

PERT stands for Project Evaluation Review Technique. We setup PERT netowork and used “backward pass” to model the relationship between the job statuses at each stage. With the PERT network model and production data we would be able to define the critical path therefore the calculation of the job completion times. The idea was to find the rules of the critical path so that calculation the job completion times quick. Defining the relationship and rules are hard due to the randomness of process data. In addition, ,PERT network model is not flexible since new model required by each of system. Not being able to simplify PERT network construction, in addition that it is hard to achieve overtime optimization with the model, this PERT attempt did not turn out to have a positive result.

4.2.2 Event Graph with Non-operational Downtime

Following PERT, we emulated the method of Event Graph which set up linear program constraints for event occurrences, and attempted to include downtime occurrences as events. A variable ρ was used to indicate downtime status and another variable W*ij* was used to indicate if a downtime *j* interrupted a job *i*. However, when trying to define the event of downtime, we experienced difficulty in obtaining right linear constraints. Lacking an efficient way to define the location relationship between jobs and the downtime occurrences, we failed to set up linear constraints that can be used to set up

a model to be solved with an optimizer program. The event graph theory only developed for simplified systems. It doesn't work well for complex system with non-operational downtimes.

4.2.3 Discrete Time Optimization Approach

The third method that we tried was discrete production horizon linear programming modeling. Production horizon was indexed using t = 1 to T, where T is the end of the horizon. A binary variable X_{kit} ≡{1 if job i at stage k is running at time slot t, 0 otherwise} is used to indicate if a job is running in the up time slots. Constraints are set to limit the total production scheduled up times. A MILP model was setup with a set of linear constraints for the time element relationship in the system. However, due to the fact that production horizon time is indexed, the total number of the X binary variable is huge since every time slot is a variable in the model. The run time for such a model is extremely long for a small scale of production with small number of jobs.

In addition to a questionable issue raised with one of the constraints, we decided to search for another method to solve the problem. However, this was a great learning process in which we gained experience in setting up a linear program for a production system. At this stage, we realized that a linear program model can be method used for job completion time prediction. In addition, the linear program model provides optimization opportunities for overtime and other elements in the model. We needed to seek a new way to represent the non-operational hours in a production horizon. Therefore, a better method is researched, and it is described in the following section.

4.3 Mathematical Linear Programming Representation

Instead of using discrete production time horizon, we switched to consider a downtime occurrence as a dummy process job in a production horizon. When a non-operational downtime should occur and interrupt a job during it process, the downtime duration is added to a job's process time. In order to identify the location and relationship of a non-operational downtime occurrence with the all the jobs in each stage, linear programming constraints are to set up to represent the relationship. A more detailed explanation of this model is described below.

4.3.1 Notation

The notations used in this mixed integer linear programming (MILP) representation model are:

$k \equiv$ stages in the production line system. Stages are set up in a production line in fixed sequence order. k = stage (1, 2, …, m).

$i \equiv$ jobs waiting to process or in the production system in fixed sequence order. i= job (1, 2, ..., n)

$j \equiv$ non-operational downtime occurrences in a production scheduling horizon. j= downtime (1, 2, ..., J)

4.3.2 Input Parameters

The input parameters used in this integer linear programming representation model are:

$A_i \equiv$ job i arrival time at the first stage

$a_k \equiv$ Total stage queuing capacity at stage k

$d_{ki} \equiv$ Job i process time at stage k

$\overline{d}_{kj} \equiv$ Duration of the j^{th} downtime occurrence at stage k

$\overline{S}_{kj} \equiv$ Start time of j^{th} downtime occurrence at stage k

$Duedate_{ki} \equiv$ job i due date at stage k

$MaxOT_{kj} \equiv$ Maximum overtime hours at stage k for the j^{th} downtime occurrence

The input parameter A_i is the time job i arrives to the first stage in the production line. In this model, all arrival times must be equal to or larger than the system start time when the system resumes starting. For jobs currently running in a particular stage in the system, known Work In Process (WIP), their job arrival times should be same as the system start time.

The input parameter a_k is the stage buffer capacity at stage k including the one space in the machine. If a system has no buffer area, a_k value is equal to one which is the machine capacity. If the a_k is set to a number other than one, the buffer area at stage k has a size of (a_k -1).

The input parameter d_{ki} is the standard process time required to complete the task at stage k for job i. Performance allowance should be considered in the processing time. For a job that has been partially completed, the process time at that stage is the time that remains to complete the task.

The input parameter $\overline{d}_{kj}$ is the non-operational downtime duration for the j^{th} occurrence at stage k in its production schedule timeline.

The input parameter $\overline{S}_{kj}$ is the start time of the j^{th} non-operational downtime at stage k in its production schedule timeline.

The input parameter $Duedate_{ki}$ is the expected job finish time for job i at stage k.

The input parameter $MaxOT_{kj}$ is the maximum overtime total allowed to schedule at stage *k* in the j^{th} downtime occurrence.

4.3.3 Decision Variables

The decision variables used in the mathematical linear programming model are:

S_{ki} ≡ Job *i* start time at stage *k*

F_{ki} ≡ Job *i* finish time at stage *k*

SS_{ki} ≡ Job *i* arrival time at stage *k*

D_{ki} ≡ Job *i* departure time at stage *k*

Y_{kij} ≡ 1 if the start time of the j^{th} non-operational downtime occurrence at stage *k* ($\overline{S}_{kj}$) precedes job *i* finish time at stage *k* (F_{ki}), otherwise 0

W_{kij} ≡ 1 if job *i* start time at stage *k* (S_{ki}) precedes the finish time of j^{th} non-operational downtime at stage *k* ($\overline{S}_{kj} + \overline{d}_{kj}$) , otherwise 0

Z_{kij} ≡ 1 if j^{th} non-operational downtime occurs or interrupts the process of job *i* at stage *k*, otherwise, 0

OT_{kij} ≡ overtime hours Job i occupied in the j^{th} non-operational downtime at stage k

Variable S_{ki} is the start time of the job *i* processing at stage *k.* A job can only be started on or after its arrival.

Variable F_{ki} is the finish time of job *i* processing at stage *k.* A job can be finished on or before its departure.

Variable D_{ki} indicates the departure time when job *i* leaves stage *k.* A job can only be departed on or after the job is finished.

Variable Y_{kij} is a binary variable, its value is equal to one when the start time of j^{th} down occurrence at stage k ($\overline{S}_{kj}$) precedes job i finish time at stage k (F_{ki}).

Variable W_{kij} is a binary variable, its value equals to one when job i start time at stage k (S_{ki}) precedes the finish time of j^{th} downtime occurrence at stage k ($\overline{S}_{kj} + \overline{d}_{kj}$) . Both variable Y and W are dummies variables used to identify the location of a non-operational downtime occurrence.

Variable Z_{kij} are binary variables, its value equals to 1 only when downtime j interrupts job i processing at stage k, where Y_{kij} and W_{kij} both equal to one.

4.4 Production without Buffer Queuing – Model COM-N-Q

To start, we first build a model for a simple production G/G/1 system. There is no waiting queue in each stage. A completed job must wait at the current stage until the next server becomes available. Blocking idle time occurs if the next stage server is not available. The production horizon on each stage doesn't have to be consistent, and non-operational down time hours can be setup at each of the horizon. An example of a five stage production tandem line is shown in figure 4-2 below.

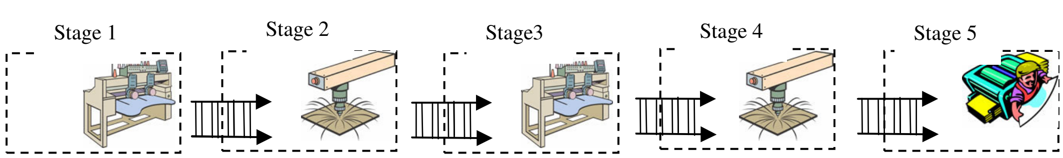


Figure 4-2 Five stage production process diagram - without buffer queue

The model we developed for this system is listed as:

$$Min \quad z = \sum_{i=1}^{n} \sum_{k=1}^{m} F_{ki}$$

s. t.

$$F_{ki} - \overline{S}_{kj} \leq M \ast Y_{kij} \quad \forall i, j, k \tag{1}$$

$$\overline{S}_{kj} - F_{ki} \leq M \ast (1 - Y_{kij}) \quad \forall i, j, k \tag{2}$$

$$\overline{S}_{kj} + \overline{d_{kj}} - S_{ki} \leq M \ast W_{kij} \quad \forall i, j, k \tag{3}$$

$$S_{ki} - \overline{S}_{kj} - \overline{d_{kj}} \leq M \ast (1 - W_{kij}) \quad \forall i, j, k \tag{4}$$

$$W_{kij} + Y_{kij} - Z_{kij} \leq 1 \quad \forall i, j, k \tag{5}$$

$$Z_{kij} - Y_{kij} \leq 0 \quad \forall i, j, k \tag{6}$$

$$Z_{kij} - W_{kij} \leq 0 \quad \forall i, j, k \tag{7}$$

$$F_{ki} - S_{ki} >= d_{ki} + \sum_{j=1}^{J} (\overline{d_{kj}} \ast Z_{kij}) \quad \forall i, j, k \tag{8}$$

$$SS_{ki} \leq S_{ki} \quad \forall k, i \tag{9}$$

$$F_{ki} \leq D_{k,i} \quad \forall k, i \tag{10}$$

$$SS_{ki} - D_{k,(i-1)} \geq 0 \quad \forall k, i = 2, \ldots, n \tag{11}$$

$$SS_{ki} - D_{(k-1),i} \geq 0 \quad \forall i, \quad k = 2, \ldots, m \tag{12}$$

$$D_{ki} - D_{k+1,i-1} \geq 0 \quad k = 1, \ldots, m-1, \; i = 2, \ldots, n \tag{13}$$

$$S_{1i} - A_i \geq 0 \quad \forall i, k = 1 \tag{14}$$

$$W_{kij}, Y_{kij}, Z_{kij} \in \{0,1\} \quad \forall i, j, k \tag{15}$$

$$A_i, SS_{ki}, S_{ki}, F_{ki}, D_{ki} \geq 0 \quad \forall k, i \tag{16}$$

4.4.1 Objective Function

The objective of this linear program model is to find the minimum total job completion times so to reduce the blocking idle time between the stages minimally. Idle time between stages in job processing can be minimized when minimizing the sum of a job's completion time at each of the stages. For a minimum job completion time at the last stage for each of the jobs, objective function need to be set to minimize the sum of all jobs' completion times in each stage. Therefore, we can write the objective function as:

$$Min \quad z = \sum_{i=1}^{n} \sum_{k=1}^{m} F_{ki} \ .$$

4.4.2 Constraints

The constraints in this model are grouped in two major sets for easy explanation and understanding. Each of the sets represents different constraint functions in this MILP model.

The first set of constraints is c*onstraints defining the relationship between the non-operational downtimes and job start and finish times.* Job processing times can be varied at each stage. The incoming job type doesn't need to be consistent. The non-operational downtime start time and duration can also vary at each stage. To find the shortest job completion time could be, identification of which jobs are interrupted during the process is needed. We look the relationship between processing job start time S_{ki} and finish time F_{ki}, also the j^{th} downtime occurrence start time $\overline{S}_{kj}$ and downtime ending time $\overline{F}_{kj}$.

We need to know if a non-operational time interrupts a job. There can only be two situations in a stage's schedule horizon: i^{th} job in the critical path is interrupted by j^{th} downtime occurrence, or the i^{th} job is not interrupted by the j^{th} downtime. A binary variable Z is introduced for these situations: Z_{kij} =1 when the i^{th} job is interrupted by the j^{th} downtime in the process at stage k; otherwise, $Z_{kij} = 0$.

We try to retrieve Z value with two dummy binary variables Y and W. The exclusion method is used to identify the location and relationship between the non-operational downtime and a job's processing beginning and ending times. Value of binary variable Y is used to identify the relationship between job finish time F_{ki} and the downtime start time $\overline{S}_{kj}$; and binary variable W is used to identify the relationship between job start time S_{ki} and the downtime finish time($\overline{F}_{kj}$). It is easy to understand that the down period finish time is $\overline{F}_{kj} = \overline{S}_{kj} + \overline{d}_{kj}$.

We set variable Y_{kij} value to 1 when j^{th} downtime occurs before job i finished. Constant M has a value larger than all the time horizon times. With M constant, we can write: $F_{ki} - \overline{S}_{kj} \le M * Y_{kij}$ (1). If j^{th} downtime begins after the i^{th} job is finished, Y_{kij} then equal to 0. We can then write $\overline{S}_{kj} - F_{ki} \le M * (1 - Y_{kij})$ (2).

The same method is applied to the constraints to define W_{kij}, which defines the relationship between a job start time S_{ki} and a non-operational downtime finish time $\overline{S}_{kj} + \overline{d}_{kj}$. We emulate the above equation and acquire $\overline{S}_{kj} + \overline{d_{kj}} - S_{ki} \le M * W_{kij}$ (3) and $S_{ki} - \overline{S}_{kj} - \overline{d_{kj}} \le M * (1 - W_{kij})$ (4). Figure 4-3 draws the above relationships in a graphic view.

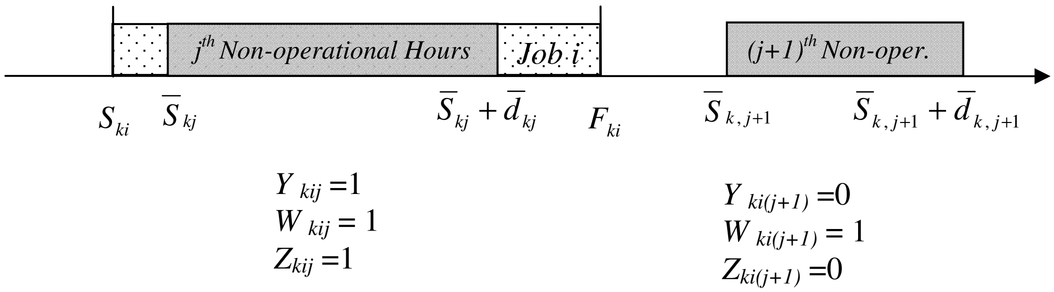


Figure 4-3 Non-operational downtime vs. job processing

A non-operational downtime does not actual interrupt a job process unless both relationships in the above example are true, where Y_{kij} and W_{kij} both are equal to one. Z_{kij} is true (Z_{kij} =1) only when both Y_{kij} and W_{kij} are true. The constraints of *WYZ* relationship are defined and grouped in constraints (5), (6) & (7) below. Therefore, we can summarize the constraints set up in this sub-constraint set listed below.

s.t.

$$F_{ki} - \overline{S}_{kj} \leq M * Y_{kij} \quad \forall i, j, k \tag{1}$$

$$\overline{S}_{kj} - F_{ki} \leq M * (1 - Y_{kij}) \quad \forall i, j, k \tag{2}$$

$$\overline{S}_{kj} + \overline{d_{kj}} - S_{ki} \leq M * W_{kij} \quad \forall i, j, k \tag{3}$$

$$S_{ki} - \overline{S}_{kj} - \overline{d_{kj}} \leq M * (1 - W_{kij}) \quad \forall i, j, k \tag{4}$$

$$W_{kij} + Y_{kij} - Z_{kij} \leq 1 \quad \forall i, j, k \tag{5}$$

$$Z_{kij} - Y_{kij} \leq 0 \quad \forall i, j, k \tag{6}$$

$$Z_{kij} - W_{kij} \leq 0 \quad \forall i, j, k \tag{7}$$

The second set of constraints is: *Constraints defining the precedence of the jobs.* We assume that a job can be preempted by non-operational downtimes. That is, after a job is interrupted, the uncompleted operation can be performed in the next system up time without restarting the whole process. Therefore, the total time required to process a job is equal to the standard job processing time plus the non-operational downtime that interrupts the job. No every job is interrupted, but it is possible a job is being interrupted by more than one downtime occurrence during its process. The sum of total non-operational downtime interrupting a job can be $\sum_{j=1}^{J} (\overline{d_{kj}} * Z_{kij})$, where Z_{kij} identifies if the downtime occurs in the job process. The actual process time between the start time and finish time includes non-operational hours which can be represented with a constraint in the following example:

$$F_{ki} - S_{ki} \geq d_{ki} + \sum_{j=1}^{J} (\overline{d_{kj}} * Z_{kij}) \text{ (8).}$$

Due to the nature of system blocking, blocking idle time in some stages is inevitable. The idle time is the gap time between finish time and departure time. Therefore, besides the two time elements job start time and job finish time, we need to introduce two more time elements to represent the situation associated with job processing in the horizon: arrival time SS_{ki} and departure time $D_{k,i}$. The event elements in job processing in sequential order are: arrival, start, finish, and departure times. If a job arrives and then starts immediately, the arrival time can be the same as the start time. This also applies to the job finish time and job departure time. In addition, in each a stage, a previous job departure time is the arrival time of the next waiting job. Time elements may or may not overlap dependant upon the job status at the moment. Figure

4-4 demonstrates the time elements for two jobs in two work stage schedule horizons.

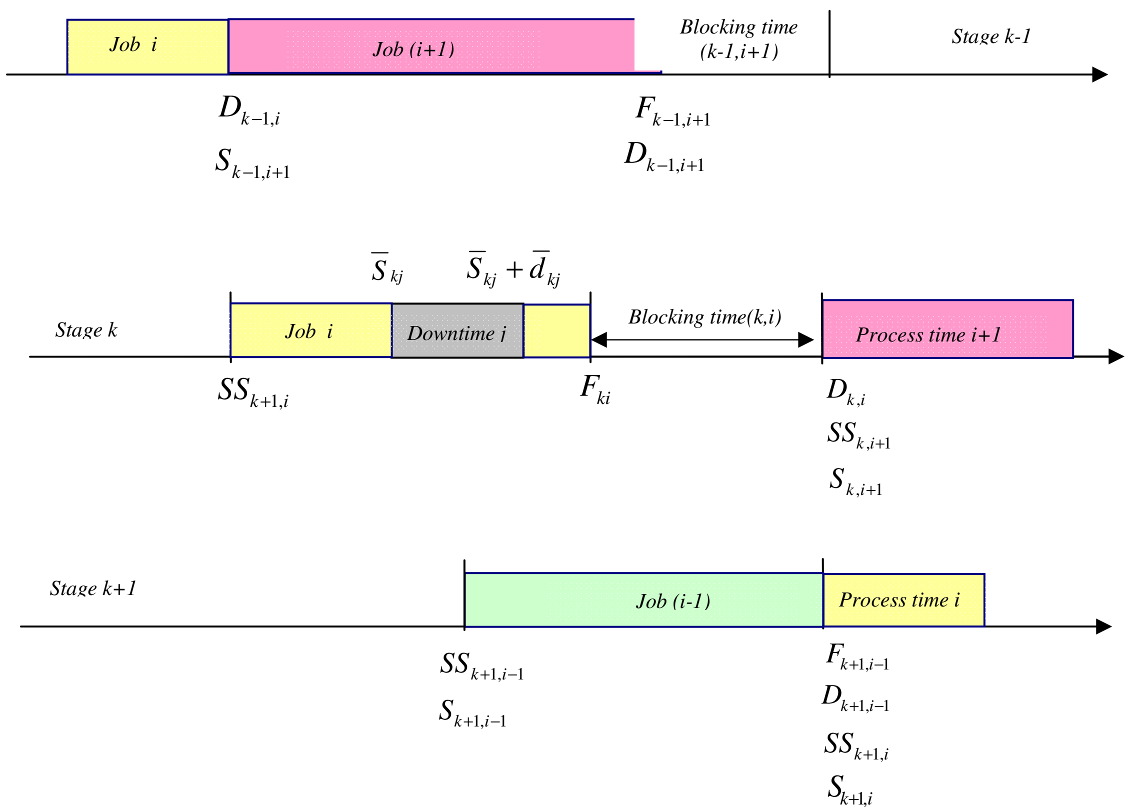


Figure 4-4 Time elements in two stage schedule horizon

We need to define the relationship between the newly created time elements arrival time and departure time with the start time and finish time. It is not difficult to set up the relationship, since a job's start time and completion time should occur between the arrival time and departure time. The two constraints to define such a relationship are:

$$SS_{ki} \le S_{ki} \quad \forall k,i \ (9)$$

$$F_{ki} \le D_{k,i} \quad \forall k,i \ (10) .$$

From the previous study of job completion time we learn that it is proved (Buzacott & Shanthikumar,1993) that in a continuous timeline horizon without downtime breaks, the relationship of the job departure times is in recursion where departure time

$$D_{ki} = \max\{D_{k-1,i} + d_{ki}, D_{k,i-1} + d_{ki}, D_{k+1,i-1}\}.$$

This verified statement was also proved using event graph method to construct linear program constraints for a discrete event system (Chan, 2005). Departure time D_{ki} equals the maximum value of another three departure values. Since in this research we consider non-operational downtime as a dummy job process in a job there will not be break time in the schedule horizons needs. The above recursion statement should be applicable in our research. We rewrite the above recursion into three linear program constraints as the following:

$$D_{ki} \geq D_{k-1,i} + d_{ki}$$

$$D_{ki} \geq D_{k,i-1} + d_{ki}$$

$$D_{ki} \geq D_{k+1,i-1}.$$

Considering the non-operational downtimes that possibly occurred in a job processing, the first two constraints with d_{ki} in it may be inaccurate in some cases. Elimination of the use of processing time d_{ki} in the constraints is needed. Under the production blocking policy, a job can not depart if it is not done, therefore we can say $F_{ki} \geq S_{ki} + d_{ki}$. Since $S_{ki} \geq SS_{ki}$ and $D_{ki} \geq F_{ki}$, substituting these two constraints in $F_{ki} \geq S_{ki} + d_{ki}$ we derived this statement: $D_{ki} \geq SS_{ki} + d_{ki}$. We again substitute this with the above three LP constraints and they can be simplified to:

$$SS_{ki} - D_{k,(i-1)} \geq 0 \qquad \forall k,\, i = 2,\ldots,n$$

$$SS_{ki} - D_{(k-1),i} \geq 0 \quad \forall i, \quad k = 2,...,m$$

$$D_{ki} - D_{k+1,i-1} \geq 0 \qquad k = 1,...,m-1, \; i = 2,...n \, .$$

The above constraints can be interpreted with arrival time SS_{ki}: job *i* cannot move or arrive to stage *k* at time SS_{ki} unless both $D_{k,i-1}$ and $D_{k-1,i}$ departures occurred. Job *i* departure times at stage *k* must be equal to or larger than the departure time of last job *(i-1)* at the next stage.

In the final step, we add a constraint to establish connection with the first stage arrival times A_i. A_i is given values, and A_i = SS_{1i} literally. In summary, the constraints setup in the second set of constraints in this model are listed below.

$$F_{ki} - S_{ki} >= d_{ki} + \sum_{j=1}^{J} (\overline{d_{kj}} * Z_{kij}) \quad \forall i, j, k \tag{8}$$

$$SS_{ki} \leq S_{ki} \quad \forall k, i \tag{9}$$

$$F_{ki} \leq D_{k,i} \quad \forall k, i \tag{10}$$

$$SS_{ki} - D_{k,(i-1)} \geq 0 \qquad \forall k, \; i = 2,...,n \tag{11}$$

$$SS_{ki} - D_{(k-1),i} \geq 0 \quad \forall i, \quad k = 2,..., m \tag{12}$$

$$D_{ki} - D_{k+1,i-1} \geq 0 \qquad k = 1,...,m-1, \; i = 2,...,n \tag{13}$$

$$S_{1i} - A_i \geq 0 \quad \forall i, k = 1. \tag{14}$$

4.5 Production with Buffer Areas – Model COM-W-Q

Now we include buffer areas in the production system. The buffer area is a location used to temporarily hold incoming jobs and allows the preceding stage to

process the next available job without blocking it. A completed job can be moved into an available buffer space at the next stage when a space is available. Blocking can still occur when all the buffer spaces in the next stage are fully occupied with no more room available. Below figure 4-5 is a graphic of a five stage flow line with a buffer area in front of each stage.

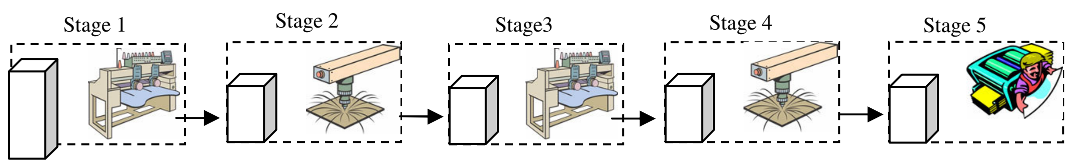


Figure 4-5 Five stage production line with buffer queue

4.5.1 Objective Function and Constraints

The objective function remains the same when buffer queues are included in the production. We want to minimize the job completion times at each of the stages and the idle times between stages.

First, we exam if the production system buffer queues makes any difference to the existing set one constraints in the previous section. The constraints in set one involve only job processing times on the machine and the machine timeline horizons. Adding buffer queues does not change the constraints in the set, therefore all the constraints in the set are valid.

Intuitively we know that less machine resource blocking occurs in the system with the buffer queues. Idle time caused by the blocking is reduced because a completed part can be moved to the next waiting queue and free the machine resource to start the next job immediately. Machine resource idle time prolongs job completion time. Looking closely into the process of production with buffers queues, production blocking only

occurs only when the buffer queue in the next stage is fully occupied. While job i is working at a stage j, and the stage $j+1$ buffer area is full, the job at stage $j+1$is the $(a_{k+1})^{th}$ job before the job i, where a_{k+1} is the next stage's stage capacity.

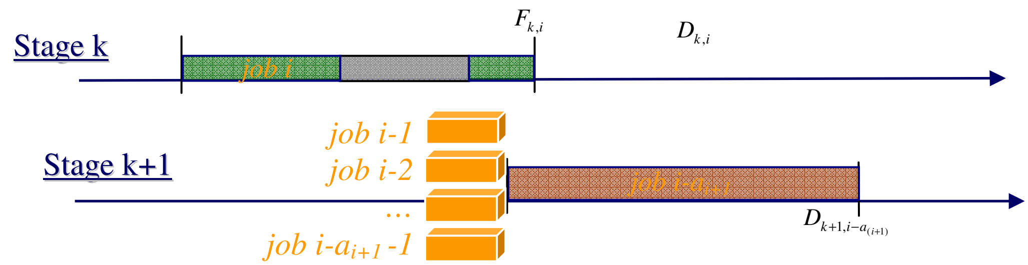


Figure 4-6 Job and buffer area

Now we exam the constraints set two in the previous no buffer model. We want to see if the existence of buffer queues affects the departure time recursion and other constraints. Constraints (8), (9), (10), and (14) are not affected because of the buffer area. Constraints 11 and 12 set limits only between departure times and the next arrival times therefore they are also true with or without buffer areas. All the constraints but one in the constraints set two needs to change when buffer areas are set up. It is constraint (13), which defines the relationship between two departure times and needs to be modified. Departure time D_{ki} is determined by the occupancy of the next stage. When a job is ready to depart, the buffer queue is full and the machine resource is busy in the next stage, based on production blocking policy, the job needs to be held until the next stage machine available. That means, departure time D_{ki} must be greater than or equal to the job departure time at the next stage machine. The job at stage k+1 is the $(i-a_{k+1})^{th}$ job, and the job departure time for that job is $D_{k+1,i-a_{k+1}}$. Therefore, the constraint can be rewritten

as $D_{ki} - D_{k+1,i-a_{k+1}} \geq 0$ (13-B) when buffer areas are setup in the line. The departure time of the job at [*k,i*] must be larger or equal to the departure time of job at the next stage machine [*k+1, i-a*$_{k+1}$]. This formulation is also verified with event graph on a continuous time horizon (Chan, 2005). All the other constraint remain unchanged since they are not affected by the setup of buffer area.

A stage without a buffer area in the first model can be viewed as a special case for stage capacity a_k=1 where the waiting buffer is none. We can also express the the model COM-N-Q with a_k. so that a new model that can be applied to both systems with and without buffer queues . The formulation of a job completion time prediction model with buffer queue is listed below as model COM-W-Q.

$$Min \quad z = \sum_{i=1}^{n} \sum_{k=1}^{m} F_{ki}$$

s.t.

$$F_{ki} - \overline{S}_{kj} \leq M * Y_{kij} \quad \forall i, j, k \tag{1}$$

$$\overline{S}_{kj} - F_{ki} \leq M * (1 - Y_{kij}) \quad \forall i, j, k \tag{2}$$

$$\overline{S}_{kj} + \overline{d_{kj}} - S_{ki} \leq M * W_{kij} \quad \forall i, j, k \tag{3}$$

$$S_{ki} - \overline{S}_{kj} - \overline{d_{kj}} \leq M * (1 - W_{kij}) \quad \forall i, j, k \tag{4}$$

$$W_{kij} + Y_{kij} - Z_{kij} \leq 1 \quad \forall i, j, k \tag{5}$$

$$Z_{kij} - Y_{kij} \leq 0 \quad \forall i, j, k \tag{6}$$

$$Z_{kij} - W_{kij} \leq 0 \quad \forall i, j, k \tag{7}$$

$$F_{ki} - S_{ki} >= d_{ki} + \sum_{j=1}^{J} (\overline{d_{kj}} * Z_{kij}) \quad \forall i, j, k \tag{8}$$

$$SS_{ki} \le S_{ki} \quad \forall k, i \tag{9}$$

$$F_{ki} \le D_{k,i} \quad \forall k, i \tag{10}$$

$$SS_{ki} - D_{k,(i-1)} \ge 0 \qquad \forall k,\ i = 2,\ldots,n \tag{11}$$

$$SS_{ki} - D_{(k-1),i} \ge 0 \quad \forall i, \quad k = 2,\ldots,m \tag{12}$$

$$D_{ki} - D_{k+1,i-a_{k+1}} \ge 0 \qquad k = 1,\ldots,m-1,\ i = a_{k+1}+1,\ldots,n \tag{13-B}$$

$$S_{1i} - A_i \ge 0 \quad \forall i,\ k = 1 \tag{14}$$

$$W_{kij}, Y_{kij}, Z_{kij} \in \{0,1\} \quad \forall i, j, k \tag{15}$$

$$A_i, SS_{ki}, S_{ki}, F_{ki}, D_{ki} \ge 0 \quad \forall k, i \tag{16}$$

4.6 Production Overtime Optimization – Model OT-OPT

Including overtime hours in the production horizon is the action of changing part of the non-operational hours into operation hours. Non-operational downtime data is deterministic in the above MILP models. Overtime extends the machine resource up hours, thus it directly changes the downtime start time ($\overline{S}_{kj}$) and the downtime duration $\overline{d_{kj}}$. Variable OT_{kij} is introduced to indicate the overtime hours job i occupies in j^{th} non-production downtime at stage k.

4.6.1 Objective Function

The objective of this model is to find the least the total overtime needed to meet all the due dates:

$$Min \quad z = \sum_{j=1}^{J} \sum_{i=1}^{n} \sum_{k=1}^{m} OT_{kij}\ .$$

4.6.2 Constraints

To write the constraints with overtime variable, we first set the relationship of overtime and the non-operational hours start time ($\overline{S}_{kj}$), downtime duration $\overline{d_{kj}}$, and the non-operational hours end time ($\overline{F}_{kj}$). See the following table.

Table 4-1 Non-operational down time hour notations with overtime

Description	Without Overtime	With Overtime
Non-operational time duration	$\overline{d}_{kj}$	$\overline{d}_{kj} - \sum_{i=1}^{n} OT_{kij}$
Non-operational start time	$\overline{S}_{kj}$	$\overline{S}_{kj} + \sum_{i=1}^{n} OT_{kij}$
Non-operational end time	$\overline{F}_{kj}$	$\overline{F}_{kj}$

To build the overtime allocation model, we substitute the variables in original constraints of the model COM-W-Q with the variables associate with overtime OT_{kij} in above table. This can be easily done for constraints #1 to #4. However, the substitution of constraint with the relationship between a job start time and completion time in constraint 8 needs to be modified. After the OT_{kij} variable substitution, this constraint is changed to

$$F_{ki} - S_{ki} >= d_{ki} + \sum_{j=1}^{J} (\overline{d_{kj}} * Z_{kij} - OT_{kij} * Z_{kij}) \quad \forall i, j, k$$

in which $OT_{kij} * Z_{kij}$ is a non-linear statement. To solve this problem, we try to split the constraint into linear one. The variable Z is a binary variable values either 1 or 0. Overtime $OT_{kij} * Z_{kij}$ can only be none-zero when non-operational downtime hour value ($\overline{d}_{kj} * Z_{kij}$) is a none-zero, in addition to that the sum of non-operational hours in the job

$\sum_{j=1}^{J}(\overline{d_{kj}} * Z_{kij} - OT_{kij} * Z_{kij})$ cannot be a negative number, we are able to split this constraint into following two constraints:

$$F_{ki} - S_{ki} >= d_{ki} + \sum_{j=1}^{J} (\overline{d_{kj}} * Z_{kij} - OT_{kij}) \quad \forall i, j, k \quad \&$$

$$\overline{d_{kj}} * Z_{kij} \geq OT_{kij} \quad \forall i, j, k \ .$$

In addition, we add a constraint to express the overtime length limitation. Overtime at the j^{th} non-operational downtime occurrence at stage k cannot exceed the length of the non-operational hours

$$\overline{d_{kj}} \geq \sum_{i=1}^{n} OT_{kij} \quad \forall i, j, k \ .$$

Lastly, we write a constraint to make sure all jobs are completed before the expected due dates. Job i finish time at stage k should occurs before or on the expected due dates

$$F_{k,i} \leq Duedate_{k,i} \quad \forall i, k \ .$$

In summary, the mathematical linear program model for overtime optimization is named as model OT-OPT and listed below.

Model OT-OPT

$$Min \quad z = \sum_{j=1}^{J} \sum_{i=1}^{n} \sum_{k=1}^{m} OT_{kij}$$

s.t .

$$F_{ki} - \overline{S}_{kj} - \sum_{i=1}^{n} OT_{kij} \leq M * Y_{kij} \quad \forall i, j, k \tag{1}$$

$$\overline{S}_{kj} + \sum_{i=1}^{n} OT_{kij} - F_{ki} \leq M * (1 - Y_{kij}) \quad \forall i, j, k \tag{2}$$

$$\overline{S}_{kj} + \overline{d_{kj}} - S_{ki} \leq M * W_{kij} \quad \forall i, j, k \tag{3}$$

$$S_{ki} - \overline{S}_{kj} - \overline{d_{kj}} \leq M * (1 - W_{kij}) \quad \forall i, j, k \tag{4}$$

$$W_{kij} + Y_{kij} - Z_{kij} \leq 1 \quad \forall i, j, k \tag{5}$$

$$Z_{kij} - Y_{kij} \leq 0 \quad \forall i, j, k \tag{6}$$

$$Z_{kij} - W_{kij} \leq 0 \quad \forall i, j, k \tag{7}$$

$$F_{ki} - S_{ki} >= d_{ki} + \sum_{j=1}^{J} (\overline{d_{kj}} * Z_{kij} - OT_{kij}) \quad \forall i, j, k \tag{8}$$

$$SS_{ki} \leq S_{ki} \quad \forall k, i \tag{9}$$

$$F_{ki} \leq D_{k,i} \quad \forall k, i \tag{10}$$

$$SS_{ki} - D_{k,(i-1)} \geq 0 \qquad \forall k, i = 2..n \tag{11}$$

$$SS_{ki} - D_{(k-1),i} \geq 0 \quad \forall i, \quad k = 2..m \tag{12}$$

$$D_{ki} - D_{k+1, i-a_{k+1}} \geq 0 \qquad k = 1..m-1, i = a_{k+1} + 1..n \tag{13}$$

$$S_{1i} - A_i \geq 0 \quad \forall i, k = 1 \tag{14}$$

$$\overline{d_{kj}} * Z_{kij} \geq OT_{kij} \quad \forall i, j, k \tag{15}$$

$$\overline{d_{kj}} \geq \sum_{i=1}^{n} OT_{kij} \quad \forall i, j, k \tag{16}$$

$$F_{k,i} \leq Duedate_{k,i} \quad \forall i, k \tag{17}$$

$$W_{kij}, Y_{kij}, Z_{kij} \in \{0,1\} \quad \forall i, j, k \tag{18}$$

$$A_i, SS_{ki}, S_{ki}, F_{ki}, D_{ki}, OT_{kij} \geq 0 \quad \forall i, j, k \tag{19}$$

4.7 Alternative Model for Production Overtime Allocation

The above mixed integer linear programming model returns with the solution for the minimum production overtime required to complete all jobs before their due dates. However, because the model objective does not optimize all job completion times, therefore job finish times returned from the overtime solution may or may not be optimal minimum job completion times.

In the situation in which production overtime is a must to meet the due dates, if the total available overtime (OTsum) is known, an alternative model can be set up with objective set to job completion time minimization. A constraint needs to be added for the total available overtime is:

$$\sum_{j=1}^{J} \sum_{i=1}^{n} \sum_{k=1}^{m} OT_{kij} \leq OTsum \qquad \forall k, i, j$$

This model objective function changes to minimize total job completion times since the total overtime hours is known. The remaining part of the model doesn't need any change. The model used for overtime allocation with a known total overtime is listed below.

Model OPT-OT-ALT

$$Min \qquad z = \sum_{i=1}^{n} \sum_{k=1}^{m} F_{ki}$$

s.t.

$$F_{ki} - \overline{S}_{kj} - \sum_{i=1}^{n} OT_{kij} \leq M \ {*Y_{kij}} \qquad \forall i, j, k \tag{1}$$

$$\overline{S}_{kj} + \sum_{i=1}^{n} OT_{kij} - F_{ki} \leq M \ * (1 - Y_{kij}) \qquad \forall i, j, k \tag{2}$$

$$\overline{S}_{kj} + \overline{d_{kj}} - S_{ki} \le M * W_{kij} \quad \forall i, j, k \tag{3}$$

$$S_{ki} - \overline{S}_{kj} - \overline{d_{kj}} \le M * (1 - W_{kij}) \quad \forall i, j, k \tag{4}$$

$$W_{kij} + Y_{kij} - Z_{kij} \le 1 \quad \forall i, j, k \tag{5}$$

$$Z_{kij} - Y_{kij} \le 0 \quad \forall i, j, k \tag{6}$$

$$Z_{kij} - W_{kij} \le 0 \quad \forall i, j, k \tag{7}$$

$$F_{ki} - S_{ki} >= d_{ki} + \sum_{j=1}^{J} (\overline{d_{kj}} * Z_{kij} - OT_{kij}) \quad \forall i, j, k \tag{8}$$

$$SS_{ki} \le S_{ki} \quad \forall k, i \tag{9}$$

$$F_{ki} \le D_{k,i} \quad \forall k, i \tag{10}$$

$$SS_{ki} - D_{k,(i-1)} \ge 0 \qquad \forall k, i = 2..n \tag{11}$$

$$SS_{ki} - D_{(k-1),i} \ge 0 \quad \forall i, \quad k = 2..m \tag{12}$$

$$D_{ki} - D_{k+1, i-a_{k+1}} \ge 0 \qquad k = 1..m-1, i = a_{k+1} + 1..n \tag{13}$$

$$S_{1i} - A_i \ge 0 \quad \forall i, k = 1 \tag{14}$$

$$\overline{d_{kj}} * Z_{kij} \ge OT_{kij} \quad \forall i, j, k \tag{15}$$

$$\overline{d_{kj}} \ge \sum_{i=1}^{n} OT_{kij} \quad \forall i, j, k \tag{16}$$

$$F_{k,i} \le Duedate_{k,i} \tag{17}$$

$$\sum_{j=1}^{J} \sum_{i=1}^{n} \sum_{k=1}^{m} OT_{kij} \le OTsum \quad \forall i, j, k \tag{18}$$

$$W_{kij}, Y_{kij}, Z_{kij} \in \{0,1\} \quad \forall i, j, k \tag{19}$$

$$A_i, SS_{ki}, S_{ki}, F_{ki}, D_{ki}, OT_{kij} \ge 0 \quad \forall i, j, k \tag{20}$$

4.8 Linear Program Model Applications

Not every production floor is alike and information expected by management users is varied. The above COM-W-Q and OPT-OT models represent the basic models that we can use to generate the job completion time and production overtime optimization. The OPT-OT-ALT is an example of enhancement. Enhancements can be done by changing or adding additional constraints to reflect the actual requirements. A linear programming model is more flexible in model modification compared with simulation model. It allows additional new constraints for other extra situations or limitation. It must be noted that one requires knowledge of linear programming, CPLEX language and thorough understanding of concepts being used in this model to be capable of making changes. Additional new constraints may or may not change the optimal solution, but if an improper constraint is formulated in the model the results could be wrong. Some constraints can be added without misrepresenting the system; they are not limited to new constraints that regulate a particular job start time or job departure time caused by raw material issue, for examples. The capability for easy enhancement is a very useful and powerful feature in practical production systems because different situations and requirements apply in varied workplaces. Therefore, the linear programming model that represents the production also needs to be modified. The flexibility of the change and application of the model makes it easy to use. It is strongly recommended that experimental verification should be performed for every new change in the model to ensure the accuracy of the result.

4.9 Solving Linear Programming Models

ILOG OPL Studio CPLEX program is used to solve the mathematical linear program representation models in this research. Only CPLEX programming part named “model” is need to setup. The CPLEX model is coded so that input data can be retrieved from a data source, not by manual input in the CPLEX. Listed in appendix A are the codes for models constructed in the previous sections.

In this research, CPLEX model is coded to retrieve production data from an Excel spreadsheet. CPLEX program contains only executive lines without hard coding data so that modification of code is not necessary when production input data has to be updated. Data only need to be updated in the spreadsheet. The CPLEX program can be launched directly by a command line written in Excel VBA. After the CPLEX program run is completed, solution results from CPLEX program are exported and posted to the spreadsheet. This setup allows easy data collection execution, solution viewing and understanding in a user interface. The user interface will be discussed in a later section.

5 Validation and Verification

The mathematical LP representation models are verified using simulation method. Rockwell Software's Arena simulation program is used. Computer simulation is one of the most popular operation research tools. It allows the system to be tested without commitment of the resources, and also can compress run time and system time greatly. In addition, it allows hypothetical tests for what-if questions. The above advantages allow us to use it to run experiments and evaluate the LP models that we proposed in the previous section.

The Arena program can be set up to automatically generate input data, this makes it possible to use the same data in CPLEX, and allows us to compare the output results from both Simulation Arena and the Optimization program CPLEX.

5.1 Simulation Models

Computer simulation models are set up to duplicate the exact production system we modeled in the CPLEX model. The following models were set up for test scenarios:

1. Five stage production flow shop without buffer queue
2. Five stage production flow shop with buffer queue
3. Ten stage production flow with buffer queue

5.1.1 Simulation Model Setup & Data Generation

A simulation model is built using the following simulation process to reflect the production floor activities:

- Hold and unhold processes: the hold and unhold conditions are set to the production blocking policy.

- Store and unstore processes: the store and unstore conditions are set for buffer area.
- Input Data – Random arrival times are created in the "Creation" process. Uniform distribution is used for the time between arrivals. Process time is also randomly generated by the system using uniform. All input data are rounded to two digits after the decimal point.
- Input Variables – Number of jobs and buffer queue size are set up with variable for easy data input and update.
- Schedule - the default non-operational hours is set to 16 hours. The system is up for 8 hours and then down for 16 hours repeatedly emulating shift hours
- Data collection and result export – ReadWrite process is used to collect the data generated in the running process and also the results. The following information is imported to Excel files: Arrival time, process time, job start time and finish time at each of stage.

After the simulation models were constructed, we ran a few verification tests to confirm that these simulation models were working correctly and represented the production line systems. Fabricated data were used and output was compared with manually calculated results. Review also was done with simulation experts. The Simulation model was confirmed to be correct. The chart below is a drawing of the Arena model set up for a five stage production line. See appendix B and CD in appendix E for detail of the Arena model setup.

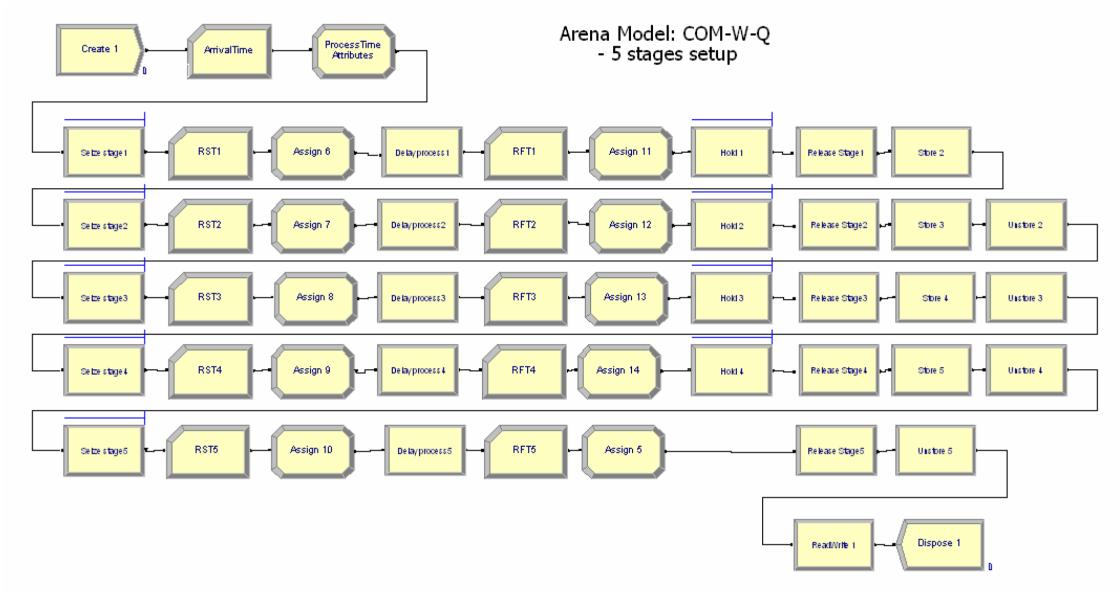


Figure 5-1 Arena model for five stage COM-W-Q model problem

5.2 Verification and Validation Scenarios

The performance verification process for the above MILP models is split into two verification scenario steps: the first verifies the performance and accuracy of the deterministic job completion time prediction model COM-W-Q, and the second verifies the overtime optimization model OT-OPT.

In addition, verification for the following special cases was also completed with fabricated data, results can also be verified with manual calculation in Excel.

1. A job is finished at the moment a downtime of next stage is started
2. A job is finished at the moment a downtime of next stage is completed
3. A job is finished in the middle of a next stage's non-operational hour
4. A job interrupted by more than one non-operational downtime
5. Long idle time caused by resources blocking

5.2.1 Scenario 1: Completion Time Prediction Verification

Verification flow process setup for model COM-W-Q is listed figure 5-2 below. A simulation Arena model is built to generate random data. The model is run. Input data generated by Arena are export to spreadsheet to be used in the CPLEX model. At the same time, the output data of job start time and completion time also is exported to be used for comparison.

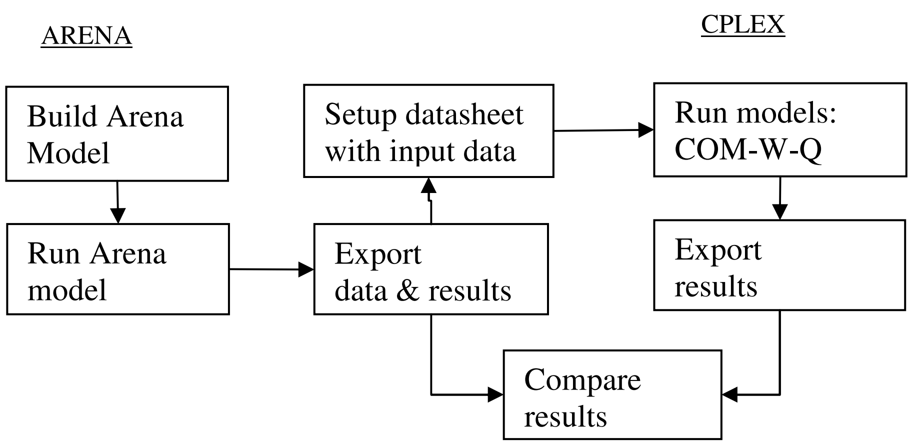


Figure 5-2 Scenario 1 test process flow

5.2.1.1 Test Set Up

Arena models are run and data is recorded and exported to the Excel spreadsheet. The following data are imported: arrival time, process service times, job start times and job finish times. Listed below are examples of the Arena outputs for a five stages ten jobs eight non-operational downtimes with buffer queue. Table 5-1 is the Arena generated arrival times and process times, and table 5-2 is the output of the entire job start times and finish times. All test experiment data is listed in Appendix C.

Table 5-1 Simulation input data

		Process Time (in hrs)				
		Stage				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	1.46	2.43	1.94	1.23	1.52
2	2.17	4.28	3.18	4.41	4.36	1.52
3	6.93	1.75	1.45	2.87	3.14	2.18
4	11.75	4.36	4.89	4.48	4.55	4.98
5	13.47	1.91	3.97	1.98	2.84	1.07
6	17.44	4.41	2.64	3.89	3.63	1.65
7	21.53	2.79	1.52	4.89	4.18	3.67
8	25.57	2.21	1.33	2.28	4.81	4.50
9	27.93	4.76	4.60	2.77	2.12	3.87
10	31.12	3.54	3.21	3.05	2.28	3.58

Table 5-2 Simulation output data

	Stage Start Time				
Job	**1**	**2**	**3**	**4**	**5**
1	0.00	1.46	3.89	5.83	7.06
2	2.17	6.45	25.63	30.04	50.40
3	6.93	25.63	30.04	50.40	53.54
4	25.63	30.04	50.93	55.41	75.96
5	30.04	50.93	55.41	75.96	96.94
6	50.93	55.41	75.96	96.94	100.60
7	55.41	75.96	96.94	101.80	122.00
8	75.96	96.94	101.80	122.00	126.80
9	96.94	101.80	122.40	126.80	147.30
10	101.80	122.40	126.80	147.30	151.20
	Stage Finish Time				
Job	**1**	**2**	**3**	**4**	**5**
1	1.46	3.89	5.83	7.06	24.58
2	6.45	25.63	30.04	50.40	51.92
3	24.68	27.08	48.91	53.54	55.72
4	29.99	50.93	55.41	75.96	96.94
5	31.95	54.90	73.39	78.80	98.01
6	55.34	74.05	79.85	100.60	102.20
7	74.20	77.48	101.80	122.00	125.70
8	78.17	98.27	120.10	126.80	147.30
9	101.70	122.40	125.20	144.90	151.20
10	121.40	125.60	145.90	149.60	170.80

Input data listed in table 5-1 is used in the CPLEX model as production input data. Data in table 5-2 is used for comparison. The non-operational downtime data set

up in Arena schedule is added to the CPLEX model. The default non-operational downtime starts from the end of the eighth hour everyday and lasts for sixteen hours. Non-operational data for all runs are provided in the appendix C.

5.2.1.2 Scenario 1 Verification Results

With the above input data MILP model is run in the CPLEX for the job completion time prediction data. The CPLEX solution is returned and exported to an Excel spreadsheet for comparison. The results are compared and show that the outputs from CPLEX and Arena are perfectly matched. Thus the MILP model returned accurate job completion time prediction.

Using the above method five replicates were run. The discrepancy between Arena and CPLEX was set to four decimal places to observe any slight differences. No discrepancy is found in the entire results from both MILP model and Arena model for all three different production setups. The bellowing table summarize of the comparisons. The numbers in the table is the total discrepancies found in the output comparison. Zeros in the table indicate that the data form both Arena and CPLEX are perfectly matched for all the job start times and job finish times in every stage.

Table 5-3 Test Scenario 1 result

Description\ Test Run No.	1st	2nd	3rd	4th	5th
5 stages 10 jobs with 8 down with no waiting buffer setup	0	0	0	0	0
5 stages 10 jobs with 8 down with buffer(capacity=3) setup	0	0	0	0	0
10 stages 15 jobs with 15 down with buffer(capacity=2) setup	0	0	0	0	0

The above verification results confirm that the MILP model COM-W-Q is correctly developed. It can be used to predict jobs completion times for production jobs and orders.

5.2.2 Scenario 2: Overtime Allocation Model Verification

The second scenario is setup to verify the overtime optimization model OT-OPT. In order to verify this overtime optimization model, we first must confirm that the MILP completion time prediction model is true and correct. Since the completion time model also is an optimization model with which minimum job finish times are returned, we can use it to verify the result returned from overtime optimization model OT-OPT. Howver, the OT-OPT model does not minimize the job completion times when all due dates are met, the completion times output data generated from OT-OPT may or may not meet the minimum values. If the completion time outputs from the two models are not perfectly matched, a second objective verification would be performed. The minimum job completion times will then be used as due dates in the OPT-OT overtime model. The sum of all overtime output from the overtime model in this second run should be equal to the sum of all overtime from the first run. The following flow chart is the process procedure used to set up verification scenario two to investigate the output of the overtime optimization model.

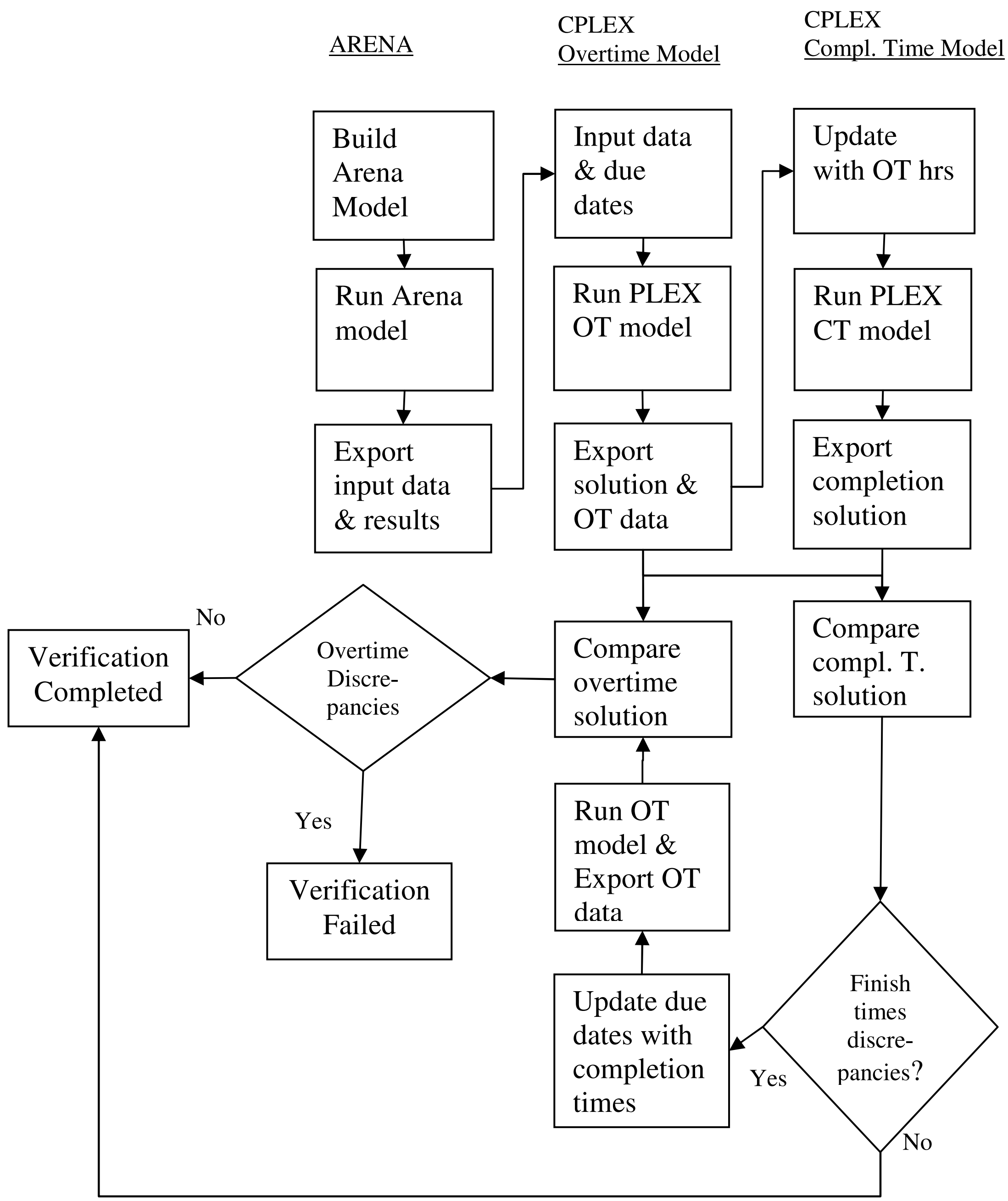


Figure 5-3 Scenario 2 verification process flow

5.2.2.1 Test Set Up

The first part of this verification process is similar to the first part of the verification scenario one. Arena simulation is used to generate input data.

In addition to the random input data obtained from Arena models, due date data is also set up to be used in CPLEX program for OT-OPT model. We set up a table for due dates. The due dates can be any numbers. To reduce the runtime it needs for an optimal solution, the smaller and the closer it is to the actual finish time is better. In our test, the due date table is set up with the finish times generated in the previous job completion model with some necessary modifications. For example, a normal job completion time is 145 hours. The due date set to 128 hour so that overtime is required to meet the due date. In this scenario the stage capacity size is set to three.

We run the CPLEX model OT-OPT with due dates and process it with input data generated from the simulation model. Results of overtime schedule from the CPLEX model OT-OPT are imported into excel spreadsheet. Table 5-4 below is the overtime optimal solution. Table 5-5 is the actual job start time and completion time with the overtime hours. The minimal total overtime is 5.99 hours.

Table 5-4 Model OPT-OT output data

	Job Start Time									
	Job									
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0	4.44	5.81	25.71	30.69	49.81	50.97	54.01	74.74	77.45
2	4.44	8.47	27.39	31.13	49.81	54.13	72.1	75.72	77.45	97.39
3	24	28.52	49.02	52.32	54.13	73.38	75.72	79.37	98.35	101.3
4	28.52	49.02	52.36	56	73.64	76.31	96	98.35	102.2	120.5
5	31.22	52.36	56	75.59	78.68	98.38	102.1	103.5	123.1	125.2
	Job Finish Time									
1	4.44	5.81	25.71	30.69	49.81	50.97	54.01	74.74	77.45	97.39
2	8.47	27.39	31.13	49.81	54.13	72.1	75.72	77.45	97.39	101.3
3	28.52	49.02	52.32	54.11	73.38	75.72	79.37	98.35	101.3	120.5
4	31.22	52.36	56	73.64	76.31	78.97	98.35	102.2	120.5	125.2
5	52.36	56	75.59	78.68	98.38	102.1	103.5	123.1	125.2	128

Table 5-5 Overtime allocation solution – general due dates

	Non-Operational downtime occurrence							
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0	**0.43**	0	0	0	0	0	0
2	**1.94**	0	**1.04**	0	0	0	0	0
3	0	**0.27**	0	**1.09**	**0.16**	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	**1.06**	0	0	0

The next step is to verify overtime solution. We accumulate the overtime hours into the regular production schedule timeline and obtain the new working production schedule as shown in Table 5-6.

Table 5-6 New production schedule with overtimes

Down Start								
	Non-operational downtime occurrence							
Stage	1	2	3	4	5	6	7	8
1	8	32.43	56	80	104	128	152	176
2	9.94	32	57.04	80	104	128	152	176
3	8	32.27	56	81.09	104.2	128	152	176
4	8	32	56	80	104	128	152	176
5	8	32	56	80	105.1	128	152	176
Down Duration								
1	16	15.57	16	16	16	16	16	16
2	14.06	16	14.96	16	16	16	16	16
3	16	15.73	16	14.91	15.84	16	16	16
4	16	16	16	16	16	16	16	16
5	16	16	16	16	14.94	16	16	16

The above non-operational hour data and process time from Arena, job completion time model COM-W-Q is run to confirm the minimum job completion time under the new schedule with overtime. Table 5-7 below is output data returned for the completion time model.

The result from CPELX model is not perfectly matched to the job completion time solutions from completion time model COM-W-Q as we expected. It is due to the fact that overtime model doesn't necessarily minimize the job start time or finish times as

long as all due dates given are met. Discrepancies listed in the below table 5-8 are the results of the completion time model output minus the overtime time model output for this experiment.

Table 5-7 Scenario 2 model COM-W-Q output

	Job Start Time									
Stage	1	2	3	4	5	6	7	8	9	10
1	0	4.44	5.81	25.71	30.69	49.81	50.97	54.01	74.74	77.45
2	4.44	8.47	27.39	31.13	49.81	54.13	72.1	75.72	77.45	97.39
3	24	28.52	49.02	51.24	54.13	72.1	75.72	79.37	98.35	101.3
4	28.52	49.02	52.36	55.44	72.56	75.23	79.37	98.35	102.2	120.5
5	31.22	52.36	56	75.59	78.68	98.38	102.1	103.5	123.1	125.2
	Job Finish Time									
1	4.44	5.81	25.71	30.69	49.81	50.97	54.01	74.74	77.45	96.96
2	8.47	27.39	31.13	49.81	54.13	72.1	75.72	77.45	97.39	101.3
3	28.52	49.02	51.24	53.03	55.44	73.86	79.37	98.35	99.58	120.5
4	31.22	52.36	55.44	72.56	75.23	77.36	96.75	102.2	120.5	125.2
5	52.03	56	75.59	78.68	98.38	102.1	103.5	123.1	125.2	128

Table 5-8 Scenario 2 comparison discrepancies

	Job Start Time					**Job Finish Time**				
Job	**St1**	**St2**	**St3**	**St4**	**St5**	**St1**	**St2**	**St3**	**St4**	**St5**
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.33
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-1.08	-0.56	0.00
4	0.00	0.00	-1.08	-0.56	0.00	0.00	0.00	-1.08	-1.08	0.00
5	0.00	0.00	0.00	-1.08	0.00	0.00	0.00	-17.94	-1.08	0.00
6	0.00	0.00	-1.28	-1.08	0.00	0.00	0.00	-1.86	-1.61	0.00
7	0.00	0.00	0.00	-16.63	0.00	0.00	0.00	0.00	-1.60	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-1.70	0.00	0.00
10	0.00	0.00	0.00	0.00	0.00	-0.43	0.00	0.00	0.00	0.00

All the discrepancy numbers are either zero or a negative number. Those negative numbers are relatively small negative numbers considering the sixteen non-operational downtime hours as part o the discrepancy. This can be interpreted as the model OT-OPT returning the best minimum overtime scheduling to meet the due date requirements, however, since the model OT-OPT doesn't have the objective function in minimizing the

job start and completion times, the output completion data is not necessarily the minimized completion times. The negativity of discrepancy reflects this fact.

In order to verify that the overtime results are the minimal total overtime allocation, we update the due dates in OPT-OT model with the minimal job completion times generated from the verified COM-W-Q model (table 5-7). We rerun the OPT-OT model and find that the output, the minimal total overtime hour, in this case, 5.99 hours, is matched with the result from the first run.

Table 5-9 Overtime solution example table

	Non-operational down occurrence							
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0	0	0	0	0	0	0	0
2	**0.81**	0	**3.26**	0	0	0	0	0
3	0	0	0	0	**0.59**	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	**1.33**	0	0	0

5.2.2.2 Scenario 2 Verification Results

Model OPT-OT was run in both the 5x10x8 and 10x15x15 systems setup in test scenario one. All five replicates are assigned with new production due dates. We are able to find optimal solution for the 5x10x8 system in seconds. All replicates return results as expected. There is no discrepancy between the overtime output totals generated from using the overtime OPT-OT model and the verified job completion time COM-W-Q model. Please see table 5.2-1 for result summary. “OK” in the tables indicates that it has completed the verification process and the result is positive. The model is valid. These test results proved and validated the overtime model OPT-OT’s correctness.

Table 5.2-1 Test Scenario one result

Description\ Test Run No.	1st	2nd	3rd	4th	5th
5 stages 10 jobs with 8 down with buffer(capacity=3) setup	OK	OK	OK	OK	OK

We are unable to obtain an optimal solution for 10x15x15 system after running the model for 10+ days. More discussions about run time evaluation are explained in the next section.

5.2.3 Scenario 3: Overtime Allocation Alternative Model Verification

This overtime allocation alternative OP-OPT-ALT model is used when the total overtime number is known, and wants to know the completion time of the job. It is a model combines of job completion time and overtime allocation model.

Model OP-OPT-ALT yields the minimum job completion time for a known overtime allocation total hours. We examine this model using the results from the overtime allocation model. We collect the total minimum overtime data, and used it as the value of OTsum. We also use the over time due date in overtime model. If infeasible solution presents, that could be possible due to the round up of the due date number in the data. For example, the due date read 4.5hrs. It actually is 4.54 hrs, but due to the round up it reads only 4.5 hrs. In this case, we added 0.1hrs to each of the due date in the process except the requirement due dates to eliminate such mathematical errors. The results turn out from this OT-OPT-ALT model will be compared to the results from the job completion model (COM-OPT).

The results from OP-OPT-ALT must be smaller or equal than results from COM-OPT model with the updated production horizon with overtime included in. It is because the OT-OPT-ALT can find the best overtime allocation which yields the minimum of job completion times. But the overtime allocation used in the COM-OPT is the minimum total overtimes but not necessary the overtime hours, in the stages and downtime

occurrences, not necessary the best allocation for the minimum of completion times. The process flow chart is as below in figure 5-4.

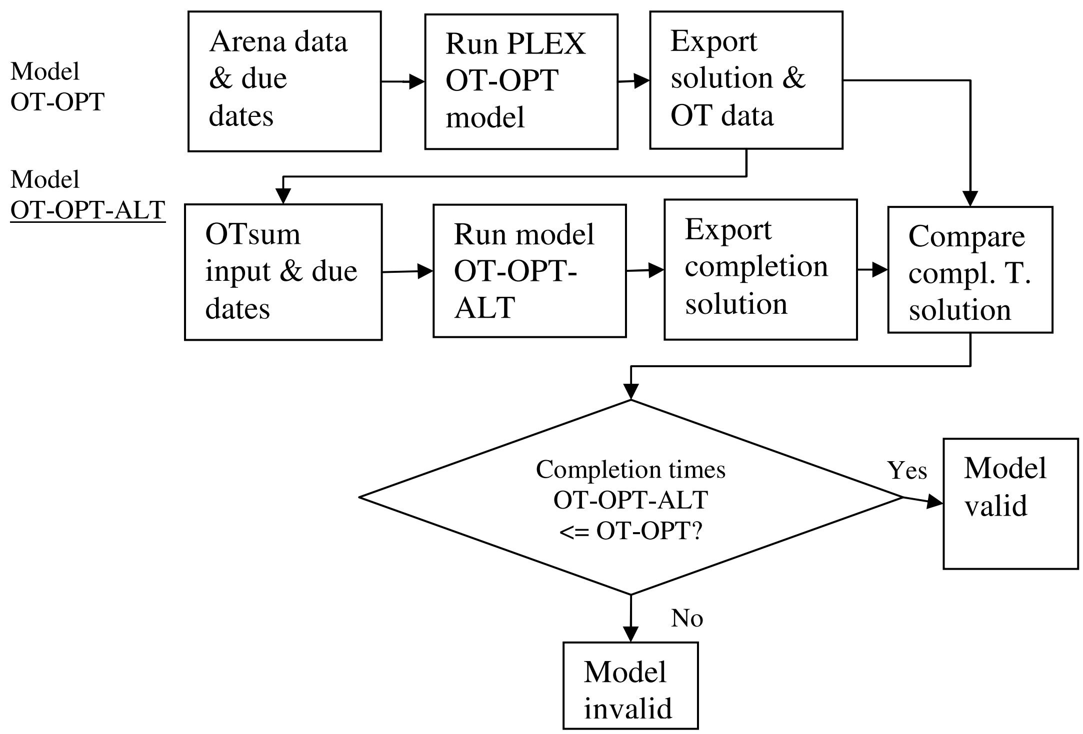


Figure 5-4 Model OT-OPT-ALT validation process flow

5.2.3.1 Test Set Up

We used the data and setup in the previous two test scenarios for this model verification. With due dates $duedate_{ki}$ and *OTsum* as input for the OT-OPT-ALT model, output data can be obtained, and used to compared with output from COM-OPT in which overtimes are updated in the schedule horizon. We are able to get the discrepancies between the two output data using the following formula

Discrepancy = COM-OPT output – OT-OPT-ALT output.

Table 5-10 time discrepancies between two outputs

	Start time discrepancies									
Stage	1	2	3	4	5	6	7	8	9	10
1	0.00	0.00	0.00	-0.59	-0.59	-0.16	-0.16	-0.16	-0.16	-0.16
2	0.00	0.00	-0.16	-0.16	-0.16	-0.16	-15.12	-1.09	-0.16	-0.16
3	0.00	0.00	-0.33	-0.33	-0.16	-0.10	-1.09	-1.09	0.00	-0.18
4	0.00	-0.33	-0.33	-0.33	-0.33	-0.33	-1.09	0.00	0.00	-0.02
5	0.00	-0.33	-0.33	-1.06	-1.06	-1.06	-1.02	-1.08	0.03	0.00
	Finish time discrepancies									
1	0.00	0.00	-0.59	-0.59	-0.16	-0.16	-0.16	-0.16	-0.16	-0.16
2	0.00	-0.16	-0.16	-0.16	-0.16	-15.12	-1.09	-1.09	-0.16	-0.18
3	0.00	-0.33	-0.33	-0.33	-0.16	-0.10	-1.09	0.00	0.00	-0.02
4	0.00	-0.33	-0.33	-0.33	-0.33	-0.33	-17.09	0.00	-0.02	0.00
5	0.00	-0.33	-1.06	-1.06	-1.06	-1.02	-1.08	0.03	0.00	0.00

All the discrepancies in the table above are less or equal to the zero. It demonstrates that the OT-OPT-ALT model, having a same total overtime, can generate a better overtime allocation with better minimal job completion times than the OT-OPT model. Table5-11 shows the new overtime allocation output (total = 5.99 hours) from using this model.

Table 5-11 Overtime allocation using OT-OPT-ALT model

	Non-Operational downtime occurrence							
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	**0.59**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	**2.10**	0.00	**1.97**	0.00	0.00	0.00	0.00	0.00
3	0.00	**0.60**	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	**0.73**	0.00	0.00	0.00	0.00	0.00

5.2.3.2 Scenario 3 Verification Results

Five replicates are setup to verify this model. The result is listed in below.

Table 5-12 Verification result for model OT-OPT-ALT

Description\ Test Run No.	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
5 stages 10 jobs with 8 down with buffer(capacity=3) setup	OK	OK	OK	OK	OK

The all five replicates show positive results for better minimal job completion times with same total overtime hours. It is a valid model for find a minimal job completion time with over time allocation to meet all the due dates.

The runtime for this model, with a narrow due dates obtained from job completion time model, ran from 2.05 seconds to 658 seconds in the above five replicates. With a wider production due dates, the time takes to solve the model is very long. It is suggested that to use this model with only narrow due dates or when heuristic can be applied.

Experimental Performance Evaluation

The MILP models developed in this research paper are flexible enough to be applied to many production flow system. They have the following advantages:

1. Mixed order type - mixed incoming orders with varied orders type is allowed as long as they have the same work route.
2. Varied process times - process times don't need to be consistent at a stage
3. Varied schedule horizon at each stage - schedule horizons do not need to be consistent at all the work stages.
4. Non-operational down time occurrences – Downtime can setup in each stage horizon. The occurrences can be varied from stage to stage.
5. Due dates set for each stage – Due dates can be set for each stage of every job

The ability to predict mixed and small quantity job completion times as well as overtime allocation provides flexibility to the production manager for a better forecast of production activity on the floor. In addition, production horizon by stage instead of whole production line also creates an opportunity to set up a flexible work schedule at each stage. Most work shops plan their production work schedule using eight hour work shifts. All machines start and stop at the same time. During job process up time, idle time is created at the beginning of the shift while waiting for a job to move from the first stage. One of the reasons that varied work schedules are not established in the workplace is due to the fact that they have no effective and valid tool to generate necessary data for analyses and support such decisions. This situation can be changed by using the support data generated from the above models. The varied work horizon may be used to support

the decision in manpower allocation, where those idled work forces can be relocated to other work functions for better productivity of production output.

These MILP models can also be used for hypothetical situations to get answers for “what if” questions raised regularly by production managers. For example “What if the process time for a specific job has to be changed?” “What is two hours overtime for a shift two weeks later?” Question like these can be answered using the model by changing input data.

The best feature of this MILP optimization model is that it generates overtime optimal solution specifically for a non-operational downtime at a work stage. Only required minimum manpower needs to schedule. In practical production without support decision tool, overtime scheduling usually happens at the last minute when a job has been moved to the last stage, where the delay problem becomes very obvious. Without an effective predicting tool, it is hard for a busy manager to know exactly when a job can be completed. Due to the concurrence of jobs, it is hard to know how long time needs to schedule and at what stages it needs to be scheduled in ahead of time when planning. Usually, it is already too late when a delay is obvious. Production overtime scheduling normally is set up at the last few work stages after a delay has become obvious and inevitable. The last few stages are not the best places for overtime scheduling. Overtime scheduling based on an experts’ intuition and experience is not scientifically proven to be efficient or correct. An optimal overtime allocation solution which can be obtained when a delay is foreseen is important to production decision making. The model optimally boosts production efficiency, and helps in cost reduction by reducing unnecessary work forces and idle time. At best, it helps efficient production controlling.

5.3 CPLEX Run Time Evaluation

Both models we developed in the previous sections are solved with the ILOG commercial software OPL studio using CPLEX program. CPLEX run time is not consistent for production systems. Varied input data may take different length of run times to reach the optimal solutions under CPLEX program's optimality criteria. When a MILP model is run, CPLEX starts to build a node tree in which each sub-problem is a node. The root of the tree is the continuous relaxation of the original MIP problem. If the solution to the relaxation has one or more fractional variables, CPLEX will try to find cuts to cut away areas of the feasible region of the relaxation that contain fractional solutions. After cutting, if the solution to the relaxation still has one or more fractional-valued integer variables, CPLEX branches on a fractional variable to generate two new sub-problems, each with more restrictive bounds on the branching variable. With binary variables, one node will fix the variable at zero, the other, at one (Bienstock, Internet source). This branch and cut algorithm is used in searching for an optimal solution, therefore, run time for an optimal solution depends greatly on how the branch and cut search is performed in the CPLEX program and its strategy setting (Bienstock, Internet manual).

5.3.1 Optimality Tolerance

Experiments are setup for run time testing. At first, we run experiment for different process time lengths. With a same production system (10 stages, 15 jobs, and 15 downtimes) four set of proportioned process time of 0.3, 0.5, 0.7 and 1 are prepared respectively. The experiments are run. Run times for optimal solutions under default

setting of CPLEX program for job completion time prediction using COM-W-Q model are collected in table 6-1.

Table 0-1 Run time result for proportioned process times

Process Time Length	*0.3 x*	*0.5 x*	*0.7 x*	*1*
Run time	*4.75sec*	*6.27sec*	*200.01sec*	*367.5 sec*
Optimal Value	*5618.17s*	*10158.29s*	*14697.57s*	*21530.57s*
Total Downtime interrupted	*16*	*26*	*33*	*53*

Run times increase while the optimal solution value and number of interrupted downtime increases. The larger the optimal value is the longer the production horizon is; and the more downtime occurrences in the production horizon. When large amount of downtime interruptions is involved, the CPLEX program is unlikely to be able to perform cutting out an infeasible large area. It may have to search a large area of the tree nodes and look into the sub-problem branches. This could possible be the reason for long solving time for an optimal solution. In addition, the long run time could be caused by the searching that CPLEX conducted in all the possible nodes to prove the solution optimality. Log files (Appendix D) for above experiments show that CPLEX can find good integer solutions early, at the very beginning of the search after preprocess and cut are done. Log file D.3 shows that at the 16.3 seconds the best solution is found but the optimality was not proved until time at 200 seconds. According to Daniel Bienstock's ILOG user resource, it states "sometime CPLEX finds a good integer solution early, but it must examine many additional nodes to prove the solution is optimal. In such a case, the additional computation is a waste of time, and an optimal solution can be speeded up by changing the optimality tolerance". The model COM-W-Q falls into this case where a good integer solution can be found early. We may consider modifing the optimality tolerance to reduce the runtime.

5.3.2 Optimality Tolerance Tradeoff

In general, the larger the optimality tolerance the greater possibility of not having a true optimal solution returned. The CPLEX's default optimality relative MIP gap tolerance is $1e^{-4}$, which means the final integer solution is guaranteed to be within 0.01% of the best node, the optimal value. Setting the tolerance to a higher number, for example $2e^{-3}$ causes CPLEX to skip any potential solution with objective value that is not at least 0.2% better than the best node solution. Theoretically, an objective difference setting can weaken the warranty of an optimal solution. There is tradeoff between run time and optimality. However, many formulations of integer or maxed integer programs can find the best integer solution quickly; they do not require such a tight tolerance, so meeting this tight default tolerance in those cases is wasting of computation. (Bienstock). In these cases, a reasonable optimal tolerance can still help the system obtain an optimal solution, or a feasible solution that is very close to the optimal solution. The nature of this research's completion time prediction model setup is not a NP-complete hard problem in which a model needs non-polynomial time to solve. Therefore, after initialed, CPLEX node tree can perform many cuts in the preprocess, and help to find the best integer solution quickly. Reviewing many completion time model log files, we can see the optimal solution appears in the very early stage of the log after a large number of cuts are performed, so we can assume that the model falls into this category. Therefore we set the MIP relative tolerance to a higher value of $2e^{-3}$, which is a number that the best integer solutions can be found be in all logs files from the experiments we ran.

5.4 Models Run Time Analysis

CPLEX model run time is reviewed in this section. We look into the four major factors in the system which may affect the CPLEX run time: number of stage, number of incoming jobs, number of stage capacity, and number of non-operational downtime occurrences.

5.4.1.1 Completion Time Prediction Model

A factorial analysis is conducted with all four main factors in a system to determine if the factors are significant to the run time. A factorial design is set up below and data is collected. We set the high and low range target to the small (5 stage, 10 jobs and 8 downtimes) to medium size (10 stages, 15 jobs, and 15 downtimes) production. Two set of data are collected. One uses the default CPLEX program setting, another one uses all the default setting except it changes the relative MIP gap tolerance to 0.2%. Table 6-2 shows the run times in the two settings. Time unit is in seconds in the table.

Table 0-2 Run time factorial analysis data

Stages	Jobs	Downs	Buffers	Default gap 0.01%	0.2% gap	Same Solution
10	10	8	5	3.25	2.81	Yes
5	15	8	5	2.64	2.5	Yes
5	10	15	5	2.38	2.27	Yes
5	10	8	1	1.69	1.63	Yes
10	15	15	5	200.01	11.89	Yes
10	15	8	1	1000+	4.33	Yes
10	10	15	1	3.44	3.44	Yes
5	15	15	1	2.74	2.67	Yes

Under the default setting of optimal tolerance some experiments take a very long time to search for the optimal value while the best integer can be found in the early of time. The optimal solution returned from setting with 0.2% relative optimal tolerance are the same with the default CPLEX settings. The best integer solutions are found at the

time before the 0.2% tolerance is reached. Combing the information we learn about the cut and branch search method using CPLEX, we find that run times seem to be unpredictable with CPLEX when searching and proving optimality,

Factorial analyses are run with MINITAB 14 program. Shown in Figure 6-1 and Figure 6-2 are the analyses run time using default setting and 0.2% tolerance setting, respectively.

Factorial Fit: Run Time (Default) versus Stages, Jobs, Downs, Buffers

Estimated Effects and Coefficients for defaultTime (coded units)

Term	Effect	Coef	SE Coef	T	P
Constant		152.0	118.7	1.28	0.290
Stages	299.1	149.5	118.7	1.26	0.297
Jobs	298.9	149.5	118.7	1.26	0.297
Downs	-199.5	-99.7	118.7	-0.84	0.462
Buffers	-200.2	-100.1	118.7	-0.84	0.461

S = 335.601 R-Sq = 60.49% R-Sq(adj) = 7.81%

Analysis of Variance for defaultTime (coded units)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	4	517287	517287	129322	1.15	0.474
Residual Error	3	337884	337884	112628		
Total	7	855171				

Figure 0-1 ANOVA run time analysis for default CPLEX Setting

Factorial Fit: Run Time(0.2% tolerance) versus Stages, Jobs, Downs, Buffers

Estimated Effects and Coefficients for RunTime (coded units)

Term	Effect	Coef	SE Coef	T	P
Constant		3.9425	0.9462	4.17	0.025
Stages	3.2800	1.6400	0.9462	1.73	0.181
Jobs	2.8800	1.4400	0.9462	1.52	0.225
Downs	2.3200	1.1600	0.9462	1.23	0.308
Buffers	1.7800	0.8900	0.9462	0.94	0.416

S = 2.67618 R-Sq = 71.98% R-Sq(adj) = 34.63%

Analysis of Variance for RunTime (coded units)

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Main Effects	4	55.21	55.21	13.802	1.93	0.308
Residual Error	3	21.49	21.49	7.162		
Total	7	76.69				

Figure 0-2 Factorial run time analysis for 0.2% MIP gap tolerance

The analyses for both default optimal setting and the 0.2% tolerance setting in above tables show that within the ranges of ten stages, fifteen jobs, and between eight to fifteen downtimes, all the factors have some degree of significance with a comparatively large p-value of 0.3 and 0.47. Among the factors, the stage is the most significant factor to run times with both default setting and the 0.2% relative gap tolerance.

Even though run times don't show strong significance with the four main factors in the above small and medium size of production systems in the factorial analyses, the size of those main factors do matter to the run time. We ran the following experiment increasing the stage number and jobs number alone to see how the run times change by the number of the factor changes.

Table 0-3 Runtime table - change in job number

Stage:	5	5	5	5
Job:	10	50	100	150
Down:	8	8	8	8
System size:	400	2000	4000	6000
Buffer Capacity:	3	3	3	3
Default	4.58	6.91	45.44	1278.25
Runtime(sec) with 0.2% gap	2.64	3.58	25.23	100.2

Table 0-4 Runtime table - change in number of stages

Stage:	10	15	20	25
Job:	15	15	15	15
Down:	15	15	15	15
System size:	2250	3375	4500	5625
Buffer Capacity:	3	3	3	3
Runtime(sec) with 0.2% gap	10.75	16.92	24.39	32.28
Gap % first reach optimal best integer	0.21%	0.27%	0.47%	0.73%

The above experiments show that the job completion prediction model under the 0.2% relative MIP gap tolerance works efficiently for small to medium scale of production within a minute of run time. To test the limit, we also ran tests in a large scale of production to see the performance of the model, below is the runtime summary:

Table 0-5 Large scale system runtimes summary

Stage:	20	20	20
Job:	30	50	50
Down:	30	30	30
System Size:	18000	30000	30000
Buffer Capacity:	1	1	1
Runtime(sec) with 0.2% gap	740	11175	122132++/ 313300++
Gap % first reach optimal best integer	0.57%	68%	No integer solution found yet

We tried a larger production system with 20 stages, 50 jobs and 30 downtimes with no waiting buffer the solution within 740 sections under the 0.2% gap tolerance. When experiment is run for a 20 stages, 50 jobs and 30 downtimes system with no waiting buffer and 0.2% gap tolerance, the run time was 11175.11 seconds. It is a bit more than 3.1 hours to get the answer. From the log of the executive file (Appendix D.3) it is found that many cuts are performed in the early stage of the execution, the best integer solution was found at the time before the 2005 seconds, where the gap was about 68%. In order to prove the optimality, the CPLEX program takes 9000 and more seconds to search all the other nodes for gaps from 0.2% to 68% to prove the optimum. This again shows that the best integer solution can be found in the very early time when running this completion time model. For the user who don't need a very tight optimal tolerance, changing the optimal tolerance with CPLEX can help to reduce the run time, especially for a large system that generally takes a comparatively long time to run. An expertise

using this system may be able to suggest what tolerance to be. However, when we run the model using the different set of input data for the same system, the runtime seems to take forever to find an integer solution. By the time it was stopped after three and half days, the best integer had not yet been found. This test is further evidence of run time unpredictability using CPLEX, especially for a larger scale system. Optimal solution may or may not be obtained within a reasonable time for a large scale production.

5.4.1.2 Overtime Allocation Optimization Model

We also tested the runtime for the overtime allocation optimization model. The model is run under tight due date constraints using the output from the job completion prediction model. Due dates in overtime allocation optimization model should be smaller than the normal job completion dates. In this experiment, the last job's due date is set to the end of the previous day of the regular job completion date. Five sets of data are setup for both smaller and medium size productions for experiment. Run time information is collected as in below.

Table 0-6 Overtime Optimization model runtimes

Stage:	*5*	*10*
Job:	*10*	*15*
Down:	*8*	*15*
Capacity:	*3*	*2*
1	16.67s	10+ days
2	1.55s	10+ days
3	13.92s	10+ days
4	5.32s	10+ days
5	4.86s	10+ days

Under the default CPLEX program setting and default MIP relative gap, the optimal solution returned for a small production system is fast, within 20 seconds. However, optimal solution for a medium size of production cannot be obtained in a reasonable time

under the default CPLEX. Research for run time reduction for the overtime allocation optimization needs to be explored.

6 Application and User Interface

6.1.1 Application Practicality

Theoretically an optimal solution is the best solution to be found under the program's optimality setting criteria. Even though it is commonly said that there are a lot of interrelated factors influencing and complicating production floor activities, an optimal solution that generated from available existing production data still can give strong support to management decision making. It at least answers this question: at this moment with the information we have, what is the best solution to my problem. In practical production planning and management, the use of this research method can be implemented by combining with shop floor tracking systems. Shop floor tracking system closely monitors the shop activities and provides actual production data that can be used and updated in the completion time prediction and overtime allocation models. For any unexpected issues or situations that shift away from what was planned and scheduled, Production data is reviewed. Previous decision may need to be re-verified and revised, prediction models to be re-run, and new decision may be made. Using the optimization models empowers shop floor activity predictability and supports production planning, and it ensures that the production is under managers' control.

6.1.2 User Interface Design

To demonstrate how this method can be applied a simple application is built based on the methods we studied. We designed a basic but functional application with Microsoft Excel. Contained in the application are two basic files: An MS Excel file, the user interface; and a complied CPLEX OPL file.

In general, there are two types of users for this type of production application tool: regular users and administrators. Regular users manipulate the application by setting up and retrieving input data datasheets, also running the program without changing any source codes or programming. Administrators have higher authority to modify the codes in the application. The administrator is responsible for maintaining the application usability. When it comes to necessary changes, for example, production or process layout change or engineering change in processes, codes in the application may need to be modified to align with the changes in production. MS Excel is chosen to build the user interface primarily because Excel and its attached program visual basic for application (VBA) are well known by most office users. MS Excel is commonly used and basic learning and training are not difficult for people with basic computer skill.

For a complex or large scale production, the interface can be modified and advanced coding can be done for more program execution flexibility and automation. Other backend data management applications such as MS Access and SQL database can be used as input data sources. It isn't necessary change the CPLEX coding except the modification of a few coding lines for data source connection.

When working with Excel interface, direct export data from CPLEX file to an Excel file when the file is opened is not possible. It is a Microsoft design issue. In order to post result data to the same Excel sheet, we use VBA codes to retrieve the solution from CPLEX after the model is executed. OPLserver type library must be installed in the excel VBA script so that codes can be run. After a command is given in the interface, VBA script in Excel calls the complied CPLEX model to run. After CPLEX execution,

VBA script retrieves the result data and exports the data value to the Excel file' designated cells for viewing and analysis.

To make it more convenient to users, this interface is built with automation of due date checking and overtime optimization model execution. Overtime optimization model is automatically launched if not all the given due dates at every stage are met. The VBA script code performs the following process flow:

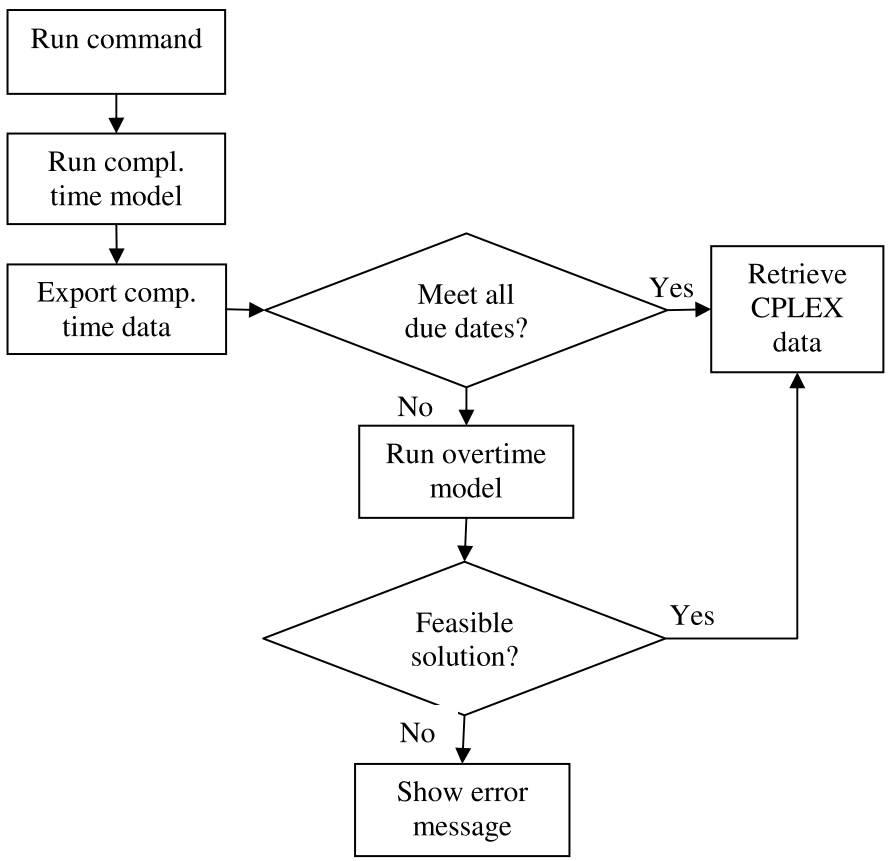
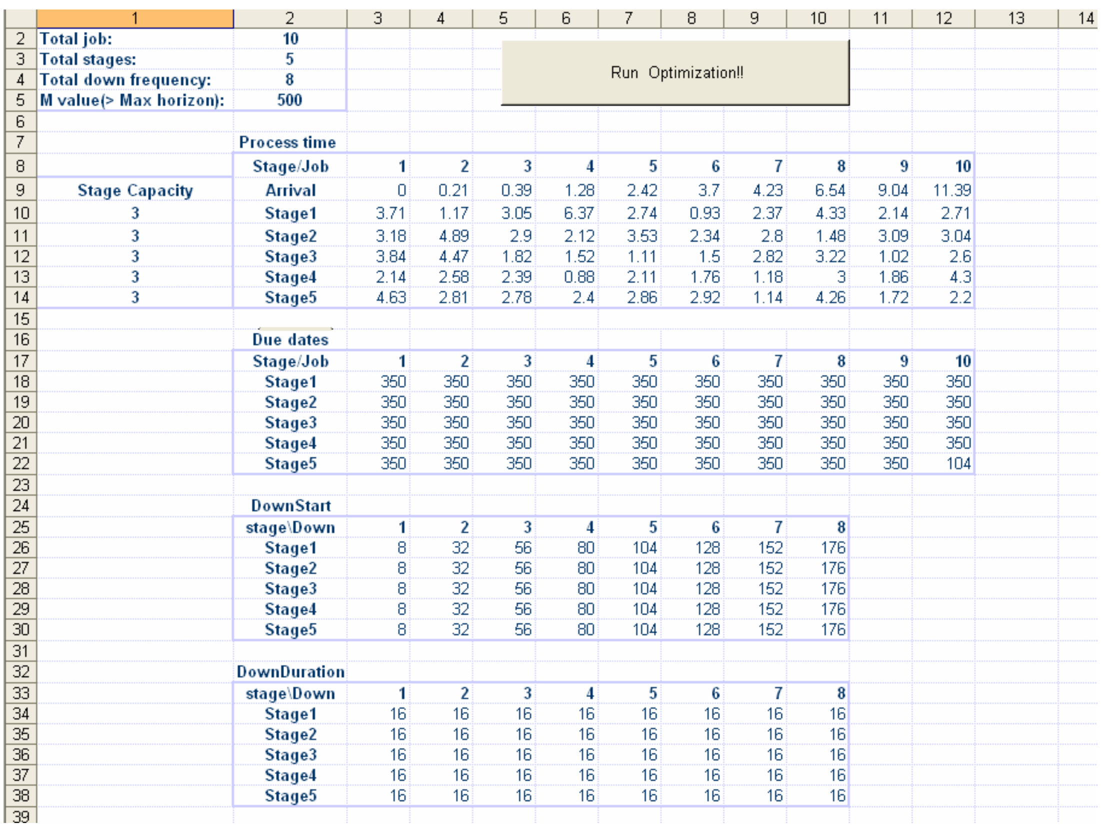


Figure 6-1 User interface VBA script process flow

The interface design aims to be simple and brief. It tries to present the essential information in an appropriate way, and make the manipulation to be user friendly. OPL executive file and the VBA codes need not to be seen by regular users. The design is

flexible to allow users to make data change in the interface, for example, changes in arrival time, in due date for a particular job at a particular stage, or a long production non-operation down time for a particular work stage. This function permits hypothesis, and allows "what-if" questions to be answered. The administrator, on the other hand, can setup or modify the design of interface to set limits or authority for what input data can be modified, in order to prevent data misuse and abuse. The user interface designed in this research demonstrates an interface used for a small production with 5 stages, 10 jobs and 8 non-operational down time occurrences. Input data includes job arrival and process times at each work stage, due dates, production non-operation down time start times and down time duration. VBA codes are assigned to the "Run Optimization" button not seen and used by regular users.



Total job:	10
Total stages:	5
Total down frequency:	8
M value(> Max horizon):	500

Run Optimization!!

Process time

	Stage/Job	1	2	3	4	5	6	7	8	9	10
Stage Capacity	Arrival	0	0.21	0.39	1.28	2.42	3.7	4.23	6.54	9.04	11.39
3	Stage1	3.71	1.17	3.05	6.37	2.74	0.93	2.37	4.33	2.14	2.71
3	Stage2	3.18	4.89	2.9	2.12	3.53	2.34	2.8	1.48	3.09	3.04
3	Stage3	3.84	4.47	1.82	1.52	1.11	1.5	2.82	3.22	1.02	2.6
3	Stage4	2.14	2.58	2.39	0.88	2.11	1.76	1.18	3	1.86	4.3
3	Stage5	4.63	2.81	2.78	2.4	2.86	2.92	1.14	4.26	1.72	2.2

Due dates

Stage/Job	1	2	3	4	5	6	7	8	9	10
Stage1	350	350	350	350	350	350	350	350	350	350
Stage2	350	350	350	350	350	350	350	350	350	350
Stage3	350	350	350	350	350	350	350	350	350	350
Stage4	350	350	350	350	350	350	350	350	350	350
Stage5	350	350	350	350	350	350	350	350	350	104

DownStart

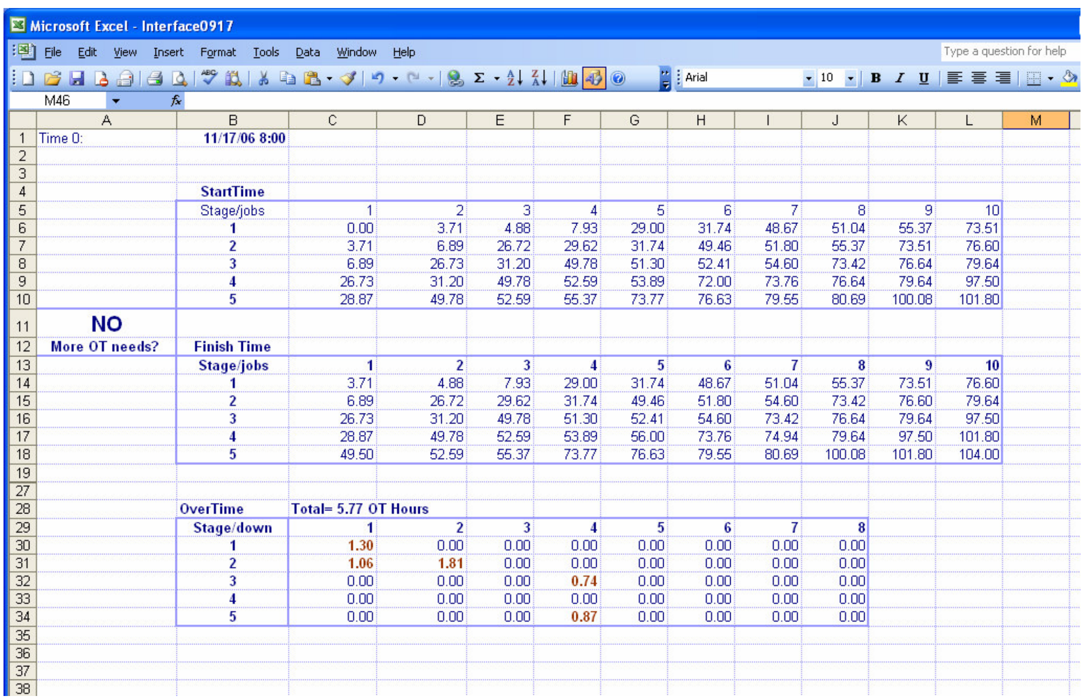
stage\Down	1	2	3	4	5	6	7	8
Stage1	8	32	56	80	104	128	152	176
Stage2	8	32	56	80	104	128	152	176
Stage3	8	32	56	80	104	128	152	176
Stage4	8	32	56	80	104	128	152	176
Stage5	8	32	56	80	104	128	152	176

DownDuration

stage\Down	1	2	3	4	5	6	7	8
Stage1	16	16	16	16	16	16	16	16
Stage2	16	16	16	16	16	16	16	16
Stage3	16	16	16	16	16	16	16	16
Stage4	16	16	16	16	16	16	16	16
Stage5	16	16	16	16	16	16	16	16

Figure 6-2 User Interface - Input data sheet

After the VBA script execution, the following output data are retrieved and posted to output sheet: job start times, job completion times, and optimal overtime allocation for satisfying all given due dates in the input sheet. Depending on the nature of the production and data requirement from management, values of some other data and variables also can be easily retrieved by an administrator modifying the VBA codes. The other data that do not show in this interface but can be retrieved from the CPLEX by additional coding includes but is not limited to: job arrival times, job departure times, the binary variable *WYZ* values, overtime allocation detailed by job at each stage, and more.


Microsoft Excel - Interface0917

Time 0: 11/17/06 8:00

NO

More OT needs?

StartTime

Stage/jobs	1	2	3	4	5	6	7	8	9	10
1	0.00	3.71	4.88	7.93	29.00	31.74	48.67	51.04	55.37	73.51
2	3.71	6.89	26.72	29.62	31.74	49.46	51.80	55.37	73.51	76.60
3	6.89	26.73	31.20	49.78	51.30	52.41	54.60	73.42	76.64	79.64
4	26.73	31.20	49.78	52.59	53.89	72.00	73.76	76.64	79.64	97.50
5	28.87	49.78	52.59	55.37	73.77	76.63	79.55	80.69	100.08	101.80

Finish Time

Stage/jobs	1	2	3	4	5	6	7	8	9	10
1	3.71	4.88	7.93	29.00	31.74	48.67	51.04	55.37	73.51	76.60
2	6.89	26.72	29.62	31.74	49.46	51.80	54.60	73.42	76.60	79.64
3	26.73	31.20	49.78	51.30	52.41	54.60	73.42	76.64	79.64	97.50
4	28.87	49.78	52.59	53.89	56.00	73.76	74.94	79.64	97.50	101.80
5	49.50	52.59	55.37	73.77	76.63	79.55	80.69	100.08	101.80	104.00

OverTime Total= 5.77 OT Hours

Stage/down	1	2	3	4	5	6	7	8
1	1.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	1.06	1.81	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.74	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.87	0.00	0.00	0.00	0.00

Figure 6-3 User Interface - Output Sheet

In summary, a simple user friendly interface is designed for the application of the mathematical linear program models we studied in this research. It demonstrates that the research method can be applied to practical production and be implemented as a support system tool in assisting in production managing and decision-making.

A user manual describes how to use this simple decision support system and also the VB script developed for automatically data posting is attached in the appendix (Appendix E). The user application user interface and the CPLEX complied OPL files are listed in the attached CD (Appendix F) for reference.

7 Conclusion & Recommendations for Future Research

On time delivery is one of the key factors in efficient managing production aims for customer satisfactions. Therefore, the ability to foresee and forecast job completion times as well as the ability to ensure production on time delivery by scheduling best production overtime allocation empowers the competitiveness of a business.

An operations research method has been studied for job completion time forecast and production expedition scheduling with overtime allocation. It uses mathematical linear programming model to represent a discrete production system which includes the downtime in the production horizon. This method applies for tandem production lines with single machined work stages. There can be buffer areas at each work stage. Each work stage has its production horizon with non-operational downtimes which don't have to be consistent with production horizons of other stages. The MILP models is efficient and work good for a small size of production system to foresee completion times for coming jobs and jobs in the production line. These models allow users to change input parameters including non-operational hour by stage to observe how the input resources interact with job completion times and overtime allocation. It is used in answering those common "what if" questions raised by the managers. A user interface is also designed to provide friendly use and easy data input and retrieval with Microsoft Excel.

The method developed from this research provides opportunity for future research and applications aiming for efficient production management support. Future study is suggested to include research for multiple parallel machines in each work stage, thus to explore an executable method for a more complex production system.

Since the run time in this research is somewhat unpredictable for medium to large size production systems, a method to obtain achievable run time for above systems is also needed. An optimal solution is good, but may not be necessary since there are so many other realistic factors such as labor cost and product quality are interrelating in a practical production. Research of heuristic is suggested when optimal solution cannot be reached in a reasonable time.

In addition, there lie the opportunities for applied research. Managerial reports can be developed from the output data, for example, idle time distribution and analysis and buffer queue space utilization report. Lastly, implementation in real manufacturing floor is needed to validate the efficiency and accuracy of the performance, and obtain a better idea of credibility of this method and support tool.

8 References

Ahuja, H. (1985). Simulation model to forecast project completion time. *Journal of construction engineering and management, 111, No.4,* 325-342.

Ambuhl, C. & Mastrolilli, M. (2005). On-line scheduling to minimize max flow time: an optimal preemptive algorithm. *Operations Research Letters, 33,* 597-602.

Bienstock, D. (Unknown date) ILOG CPLEX 9.1. Retrieved from *http://www.columbia.edu/~dano/resources/cplex91_man/usrcplex/solveMIP33.html#647126* .

Chan, W. K. & Schruben, L. W. (2004). Generating scheduling constraints for discrete event dynamic systems. *Winter Simulation Conference.*

Chan, W. K. & Schruben, L. W. (2003). Properties of discrete event systems from their mathematical programming representations. *Winter Simulation Conference.*

Chan, W. K. & Schruben, L. W. (2003). Advanced event scheduling methodology. *Winter Simulation Conference.*

Chang, G., Aganagic, M., Waight, J., Medina, J., Burton, T., Reeves, S., & Christoforidis, M. (2001). Experiences with mixed integer linear programming based approaches on short-term hydro scheduling. *IEEE Transactions on Power Systems 16, No. 4,* 743-749.

Deng, Y., Wang, J. & Zhou, M. (2003). Consistency verification in modeling of real-time system. *IEEE Transactions on Robotics and Automation, 19, No.6,* 136 -142.

Evans, G.W., Mollaghasemi, M., Russell, E.C., & Biles, W.E.(1993). The use of event graphs in simulation modeling instruction. *Winter Simulation conference.*

Esquirol, P., Lopez, P., Haudot, L., & Sicard, M. (1997). Constraint-oriented cooperative scheduling for aircraft manufacturing. *IEEE* Vol.12, Issue 1, 32 – 39.

Feygin, L.I. (1985). Forecasting completion times for operations with fuzzy durations. *Soviet Journal of Computer and Systems Sciences, 23, no.2 128-131.*

French, R.S., Lam, M.S., Levitt, J. R. & Olukotun, K. (1995). A general Method for Compiling Event-Driven Simulations. *32nd ACM/IEEE Design Automation Conferenc*e, *internet source*.

Furuhashi, T, Maeda, H, & Takaba, M(2003) A proposal of combined method of evolutionary algorithm and heuristics for nurse scheduling support. *IEEE Transactions on Industrial Electronics*, v50, n5, 833-838.

Hendrickson, C., & Au,T. (1989). Fundamental scheduling procedures. *Project Management for Construction* (Chapter 10). Prentice Hall, Retrieved from http:// www.ce.cmu.edu/pmbook/.

Hodgson, T., King, R., Thoney, K., Stanislaw, N., & et al (2000). On satisfying due-dates in large job shops: idle time insertion. *IIE Transactions, 32, No.2,* 177.

Johnson, D. (2002). A spreadsheet method for calculating work completion time probability distribution. *International Journal of Production research, 40, No.5,* 1131-1153.

Lagodimos, A.G. & Mihiotis, A.N. (2006). Overtime vs. Regular shift planning decisions in packing shops. *International Journal of production Economics,* v101,n2 246-258.

Lara, J.D. (2005). Distributed Event Graphs: Formalizing component-based modeling and simulation. *Theoretical Computer Science, 127,*145-162.

Lee, T.E., Seo, J.W., & Park, S.H.(2003). An extended event graph with negative places and negative tokens for time window constraints. *The sixth International Workshop on Discrete Event Systems.*

Li, S.J., Li, V., Liu,Y., & Liu, Z.G. (2005). Hybrid method of BPN and genetic algorithm for completion time prediction. *International conference on Machine Learning and Cybernetics, ICMLC,* 4625-4630.

Liu, S.G., Fan, H., & Li, J.J. (2003). Application of event graph in network performance analysis. *Second International Conference on Machine Learning and Cybernetic.*

Ozdamar, L., & Birbil, S.I. (1998) Hybrid heuristics for the capacitated lot sizing and loading problem with setup times and overtime decisions. *European Journal of Operational Research, v 110, n 3,* 525-547.

Parente, D.H., Venkataraman, R., Fizel, J., & Millet, I.(2004). A conceptual research framework for analyzing online auctions in a B2B environment. *Supply Chain Management, 5,* 287-294.

Reporter (2003, July 3, 1998). Manufacturing and Technology News. *Dell's make-to-order system leaves competitors in the dust. 5*, #13. Retrieved from: http://www.manufacturingnews.com/news/98/0703/art1.html

Schruben, L. W.(1993). Simulation modeling with event graphs *Communications of the ACM, l26, No.11,* 957 -963.

Schruben, L.W. (2000). Mathematical programming models of discrete event system dynamics. W*inter Simulation Conference.*

Swisher, J., Hyden, P., Jacobson, S. & Schruben, L.W. (2000). A survey of simulation optimization techniques and procedures. *Winter Simulation Conference.*

Taner, M. & Hodgson, T., King, R., & Thoney, K. (2003). Satisfying due-dates in a jobshop with sequence-dependent family set-ups. *International Journal of Production Research, 41, No.17,* 4153-4169.

Thoney, K., Hodgson, T., King, R., Taner, M.& Wilson, A.(2002). Satisfying due-dates in large multi-factory supply chains. *IIE Transaction, 34,* 803-811.

Xia, W.J. & Wu, Z.M. (2005). An effective hybrid optimization approach for multi-object flexible job-shop scheduling problems. *Computer & Industrial Engineering*, *48,* 409- 425.

APPENDIES

Appendix A: ILOG Program Codes

A.1 Code for production without buffer area: Model COM-N-Q

```
/////// 4 variables - no queue06/28/2006
int job = ...;
int stage = ...;
int down = ...;
float M = 5000;

range jobs 1..job;
range stages 1..stage;
range downs 1..down;

float+ Arrive[jobs] = ...;
float+ sT[stages,jobs]= ...;        //service Time
float+ DStart[stages, downs]=...;   //downtime start time
float+ DR[stages, downs]=...;       //downtime duration

var int Y[stages, jobs, downs] in 0..1;
var int W[stages, jobs, downs] in 0..1;
var float Z[stages, jobs, downs] in 0..1;
var float+ SST[stages,jobs];    // Arrival time
var float+ DT[stages, jobs];    // Depature Time --- in 0..450
var float+ ST[stages,jobs];    //Start time
var float+ FT[stages,jobs];  //Finish time

minimize
sum(k in stages,i in jobs) FT[k,i]

subject to {


forall (k in stages, j in downs, i in jobs)
FT[k,i] - DStart[k, j] <= M*Y[k,i,j];

forall (k in stages, j in downs, i in jobs)
DStart[k, j] - FT[k,i]  <= M* (1- Y[k,i,j]);

forall (k in stages, i in jobs, j in downs)
DStart[k,j]+DR[k,j]-ST[k,i]  <= M*W[k,i,j];

forall (k in stages, i in jobs, j in downs)
ST[k,i] - DStart[k,j]-DR[k,j]<= M* (1-W[k,i,j]);
```

```
forall (k in stages, j in downs, i in jobs)
Y[k,i,j]+W[k,i,j]-Z[k,i,j] <= 1;
forall (k in stages, j in downs, i in jobs)
Z[k,i,j]-Y[k,i,j] <= 0;
forall (k in stages, j in downs, i in jobs)
Z[k,i,j]-W[k,i,j] <= 0;

forall (k in stages, i in jobs)
FT[k,i] - ST[k,i]>= sT[k,i] + sum (j in downs) DR[k,j]*(Z[k,i,j]);

forall(k in stages, i in 2..job)
  SST[k,i] - DT[k,i-1] >=0;

forall(k in 2.. stage, i in jobs)
  SST[k, i] - DT[k-1,i] >=0;

forall (k in stages, i in jobs)
  SST[k,i]<=ST[k,i];

forall (k in stages, i in jobs)
  ST[k,i]<=DT[k,i];

forall (k in stages, i in jobs)
  SST[k,i]<=FT[k,i];

forall (k in stages, i in jobs)
  FT[k,i]<=DT[k,i];


forall (k in 1..stage-1,i in jobs: i>=2)
  DT[k,i] - DT[k+1, (i-1)] >=0;

forall(i in jobs)
  SST[1, i]- Arrive[i] >= 0;

};
display (k in stages, i in jobs) DT[k,i];
display (k in stages, i in jobs) SST[k,i];
display (k in stages, i in jobs) ST[k,i];
display (k in stages, i in jobs) FT[k,i];
display (k in stages, i in jobs, j in downs: Y[k,i,j]>0) Y[k,i,j];
display (k in stages, i in jobs, j in downs: W[k,i,j]>0) W[k,i,j];
display (k in stages, i in jobs, j in downs: Z[k,i,j]>0) Z[k,i,j];
```

A.2 Code for production with Buffer area: Model COM-W-Q

```
//Completion Time prediction model
//No hard coded data in the model
//Data controlled by the ranges in Excel in this model

string location = ...;
SheetConnection sheet(location,1);
int temp[0..3] from SheetRead(sheet,"BaseData");
int job  = temp[0];
int stage =temp[1];
int down = temp[2];
float M = temp[3];
range jobs 1..job;
range stages 1..stage;
range downs 1..down;

float+ Arrive[jobs] from SheetRead(sheet,"Arrival");
float+ sT[stages,jobs] from SheetRead (sheet,"ProcessTime");      //service Time
float+ DStart[stages, downs]from SheetRead (sheet,"DownStart");   //downtime start time
float+ DR[stages, downs]from SheetRead (sheet,"DownDuration");    //downtime
duration
int+ Buffer[stages] from SheetRead (sheet,"Stage_Capacity");      //Stage capacity

var int Y[stages, jobs, downs] in 0..1;
var int W[stages, jobs, downs] in 0..1;
var float Z[stages, jobs, downs] in 0..1;
var float+ SST[stages,jobs];    // Arrival time
var float+ DT[stages, jobs];    // Depature Time
var float+ ST[stages,jobs];     //Start time
var float+ FT[stages,jobs];     //Finish time

//* Objective
minimize
sum(k in stages,i in jobs) FT[k,i]

//* Constraints
subject to {

forall (k in stages, j in downs, i in jobs)
FT[k,i] - DStart[k, j] <= M*Y[k,i,j];

forall (k in stages, j in downs, i in jobs)
DStart[k, j] - FT[k,i]  <= M* (1- Y[k,i,j]);

forall (k in stages, i in jobs, j in downs)
```

```
DStart[k,j]+DR[k,j]-ST[k,i]  <= M*W[k,i,j];

forall (k in stages, i in jobs, j in downs)
ST[k,i] - DStart[k,j]-DR[k,j]<= M* (1-W[k,i,j]);


forall (k in stages, j in downs, i in jobs)
Y[k,i,j]+W[k,i,j]-Z[k,i,j] <= 1;
forall (k in stages, j in downs, i in jobs)
Z[k,i,j]-Y[k,i,j] <= 0;
forall (k in stages, j in downs, i in jobs)
Z[k,i,j]-W[k,i,j] <= 0;

forall (k in stages, i in jobs)
FT[k,i] - ST[k,i]>= sT[k,i] + sum (j in downs) DR[k,j]*(Z[k,i,j]);

forall(k in stages, i in 2..job)
  SST[k,i] - DT[k,i-1] >=0;

forall(k in 2.. stage, i in jobs)
  SST[k, i] - DT[k-1,i] >=0;

forall (k in stages, i in jobs)
  SST[k,i]<=ST[k,i];

forall (k in stages, i in jobs)
  ST[k,i]<=DT[k,i];

forall (k in stages, i in jobs)
  SST[k,i]<=FT[k,i];

forall (k in stages, i in jobs)
  FT[k,i]<=DT[k,i];

forall (k in 1..stage-1,i in jobs: i>=Buffer[k+1]+1)
  DT[k,i] - DT[k+1, (i-Buffer[k+1])] >=0;

forall(i in jobs)
  SST[1, i]- Arrive[i] >= 0;
```

A.3 Code for overtime allocation with buffer area: model OT-OPT

```
//OverTime Optimization model
//No hard coded data in the model
//Data controlled by the ranges in Excel in this model

string location = ...;
SheetConnection sheet(location,1);
int temp[0..3] from SheetRead(sheet,"BaseData");
int job  = temp[0];
int stage =temp[1];
int down = temp[2];
float M = temp[3];
range jobs 1..job;
range stages 1..stage;
range downs 1..down;

float+ Arrive[jobs] from SheetRead(sheet,"Arrival");
float+ sT[stages,jobs] from SheetRead (sheet,"ProcessTime");      //service Time
float+ DStart[stages, downs]from SheetRead (sheet,"DownStart");   //downtime start time
float+ DR[stages, downs]from SheetRead (sheet,"DownDuration");    //downtime
duration
int+ Buffer[stages] from SheetRead (sheet,"Stage_Capacity");      //Stage capacity
float+ Duedate[stages, jobs] from SheetRead (sheet,"OTDue");

var int Y[stages, jobs, downs] in 0..1;
var int W[stages, jobs, downs] in 0..1;
var float Z[stages, jobs, downs] in 0..1;

var float+ SST[stages,jobs];    // Arrival time
var float+ DT[stages, jobs];    // Depature Time --- in 0..450
var float+ ST[stages,jobs];    //Start time
var float+ FT[stages,jobs];  //Finish time
var float+ OT[stages,jobs,downs];
var float+ DOT[stages, downs]; // for output purpose


//Objective
minimize
sum(k in stages,j in downs) DOT[k,j]

//Constraints
subject to {

forall (k in stages, j in downs, i in jobs)
FT[k,i] - DStart[k, j] - sum (i in jobs) OT[k,i,j]<= M*Y[k,i,j];
```

```
forall (k in stages, j in downs, i in jobs)
DStart[k, j] +sum (i in jobs) OT[k,i,j]- FT[k,i]  <= M* (1- Y[k,i,j]);

forall (k in stages, i in jobs, j in downs)
DStart[k,j]+DR[k,j]-ST[k,i]  <= M*W[k,i,j];

forall (k in stages, i in jobs, j in downs)
ST[k,i] - DStart[k,j]-DR[k,j]<= M* (1-W[k,i,j]);

forall (k in stages, j in downs, i in jobs)
Y[k,i,j]+W[k,i,j]-Z[k,i,j] <= 1;
forall (k in stages, j in downs, i in jobs)
Z[k,i,j]-Y[k,i,j] <= 0;
forall (k in stages, j in downs, i in jobs)
Z[k,i,j]-W[k,i,j] <= 0;

forall (k in stages, i in jobs)
FT[k,i] - ST[k,i]>= sT[k,i] + sum (j in downs)(DR[k,j]*Z[k,i,j]-OT[k,i,j]);

forall(k in stages, i in 2..job)
  SST[k,i] - DT[k,i-1] >=0;

forall(k in 2.. stage, i in jobs)
  SST[k, i] - DT[k-1,i] >=0;

forall (k in stages, i in jobs)
  SST[k,i]<=ST[k,i];

forall (k in stages, i in jobs)
  ST[k,i]<=DT[k,i];

forall (k in stages, i in jobs)
  SST[k,i]<=FT[k,i];

forall (k in stages, i in jobs)
  FT[k,i]<=DT[k,i];

forall (k in 1..stage-1,i in jobs: i>=Buffer[k+1]+1)
  DT[k,i] - DT[k+1, (i-Buffer[k+1])] >=0;

forall(i in jobs)
  SST[1, i]- Arrive[i] >= 0;

forall(k in stages, i in jobs,j in downs)
  DR[k,j]*Z[k,i,j]>=OT[k,i,j];
```

```
forall(k in stages,j in downs)
  DR[k,j]>= sum(i in jobs) OT[k,i,j];

forall(k in stages, i in jobs)
FT[k,i] <= Duedate[k,i];

forall (k in stages,j in downs)
DOT[k,j]>= sum(i in jobs)OT[k,i,j];

};
```

A.4 Code for overtime allocation alternative model: model OT-OPT-ALT

```
/*****************   5 stages & 10 jobs & 8 downtimes*****************/
string sheetname="\\Gozer\home directories\Grad.
Students\OLiu\thesis\Arena\Verification\ArenaOut-S5J10-OT.xls";
SheetConnection OTS5J10(sheetname);
int job = 10;
int stage =5;
int down = 8;
float M = 500;//20000;
range jobs 1..job;
range stages 1..stage;
range downs 1..down;
float+ Arrive[jobs] from SheetRead(OTS5J10,"B2:k2");
float+ sT[stages,jobs] from SheetRead (OTS5J10,"B3:K7");       //service Time
float+ DStart[stages, downs]from SheetRead (OTS5J10,"B48:I52");  //downtime start
time
float+ DR[stages, downs]from SheetRead (OTS5J10,"B55:I59");      //downtime duration
int+ Buffer[stages] from SheetRead (OTS5J10,"Q48:Q52");
float+ Duedate[stages, jobs] from SheetRead (OTS5J10,"B62:K66");

float+ MaxOT = 5.78; // # given in when total OT value is known

///////---- Model content-----
var int Y[stages, jobs, downs] in 0..1;
var int W[stages, jobs, downs] in 0..1;
var float Z[stages, jobs, downs] in 0..1;
var float+ SST[stages,jobs];   // Arrival time
var float+ DT[stages, jobs];   // Depature Time --- in 0..450
var float+ ST[stages,jobs];   //Start time
var float+ FT[stages,jobs];  //Finish time
var float+ OT[stages,jobs,downs];
var float+ DOT[stages, downs]; // for output purpose
```

```
minimize
sum(k in stages,i in jobs) FT[k,i]

subject to {

forall (k in stages, j in downs, i in jobs)
FT[k,i] - DStart[k, j] - sum (i in jobs) OT[k,i,j]<= M*Y[k,i,j];

forall (k in stages, j in downs, i in jobs)
DStart[k, j] +sum (i in jobs) OT[k,i,j]- FT[k,i]  <= M* (1- Y[k,i,j]);

forall (k in stages, i in jobs, j in downs)
DStart[k,j]+DR[k,j]-ST[k,i]  <= M*W[k,i,j];

forall (k in stages, i in jobs, j in downs)
ST[k,i] - DStart[k,j]-DR[k,j]<= M* (1-W[k,i,j]);

forall (k in stages, j in downs, i in jobs)
Y[k,i,j]+W[k,i,j]-Z[k,i,j] <= 1;
forall (k in stages, j in downs, i in jobs)
Z[k,i,j]-Y[k,i,j] <= 0;
forall (k in stages, j in downs, i in jobs)
Z[k,i,j]-W[k,i,j] <= 0;

forall (k in stages, i in jobs)
FT[k,i] - ST[k,i]>= sT[k,i] + sum (j in downs)(DR[k,j]*Z[k,i,j]-OT[k,i,j]);

forall(k in stages, i in 2..job)
  SST[k,i] - DT[k,i-1] >=0;

forall(k in 2.. stage, i in jobs)
  SST[k, i] - DT[k-1,i] >=0;

forall (k in stages, i in jobs)
  SST[k,i]<=ST[k,i];

forall (k in stages, i in jobs)
  ST[k,i]<=DT[k,i];

forall (k in stages, i in jobs)
  SST[k,i]<=FT[k,i];

forall (k in stages, i in jobs)
  FT[k,i]<=DT[k,i];
```

```
forall (k in 1..stage-1,i in jobs: i>=Buffer[k+1]+1)
  DT[k,i] - DT[k+1, (i-Buffer[k+1])] >=0;

forall(i in jobs)
  SST[1, i]- Arrive[i] >= 0;

forall(k in stages, i in jobs,j in downs)
  DR[k,j]*Z[k,i,j]>=OT[k,i,j];

forall(k in stages,j in downs)
  DR[k,j]>= sum(i in jobs) OT[k,i,j];

forall(k in stages, i in jobs)
FT[k,i] <= Duedate[k,i];

forall (k in stages,j in downs)
DOT[k,j]>= sum(i in jobs)OT[k,i,j];

sum (k in stages, j in downs)DOT[k,j]<=MaxOT;

};

/****  S5J10-OT Output Display *******/
SheetWrite (OTS5J10, "C27:L31" )(ST);
SheetWrite (OTS5J10, "C33:L37" )(FT);
SheetWrite (OTS5J10, "C39:J43" )(DOT);
```

Appendix B: Simulation Models

Ten stage simulation model used for CPLEX model verification and validation. A soft copy is saved in the CD.

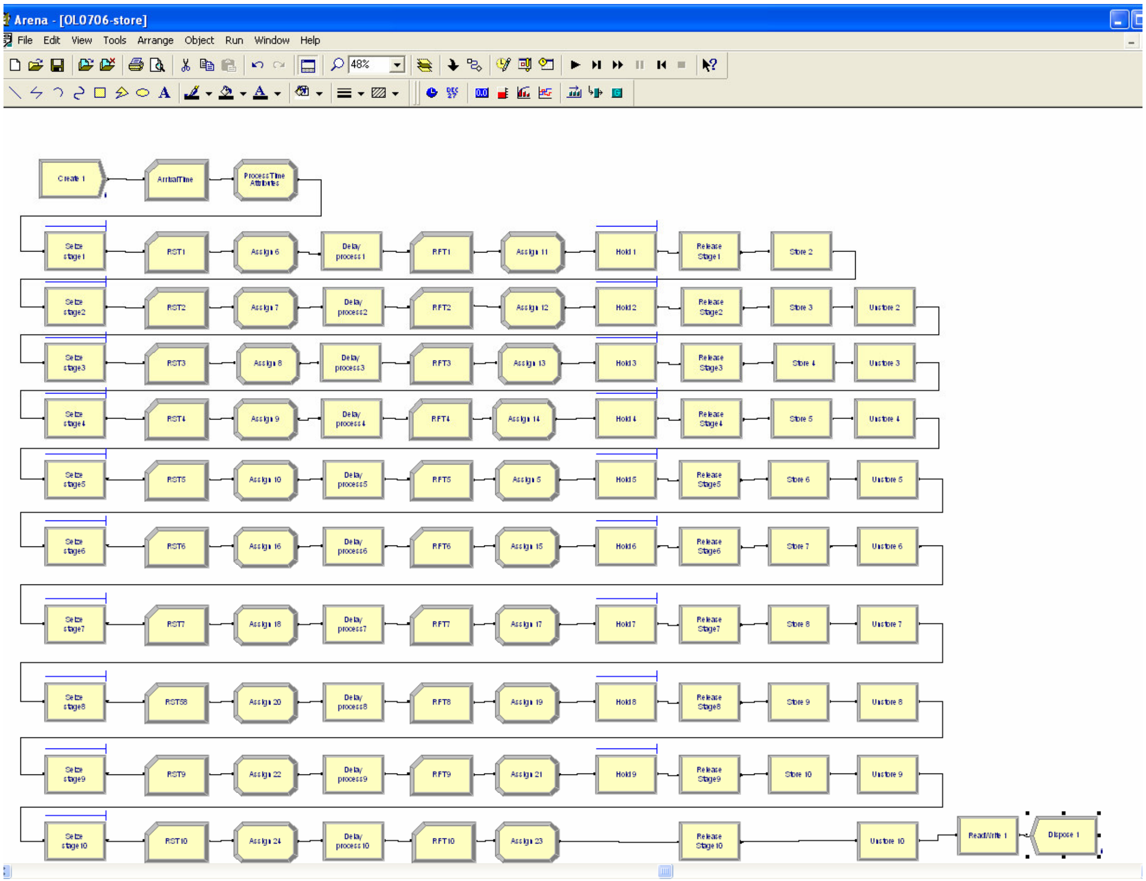


Figure B-1 Arena model setup for 10 stage production

Appendix C: Test Data and Outputs

The below tables are the test data and outputs from Arena and CPLEX models in the following categories. Each category was run for five experiments with individual dataset.

1. Five experiments for system with five stages ten jobs eight downtimes with stage capacity of 1(no waiting queue).
2. Five experiments for system with five stages ten jobs eight downtimes with stage capacity of 3(waiting queue size is 2).
3. Five experiments for system with ten stages fifteen jobs fifteen downtimes with stage capacity of 2 (waiting queue size is 1).

C.1 Dataset A- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 1

Table C-1 Input data - from Arena

		Stage				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	1.46	2.43	1.94	1.23	1.52
2	2.17	4.28	3.18	4.41	4.36	1.52
3	6.93	1.75	1.45	2.87	3.14	2.18
4	11.75	4.36	4.89	4.48	4.55	4.98
5	13.47	1.91	3.97	1.98	2.84	1.07
6	17.44	4.41	2.64	3.89	3.63	1.65
7	21.53	2.79	1.52	4.89	4.18	3.67
8	25.57	2.21	1.33	2.28	4.81	4.50
9	27.93	4.76	4.60	2.77	2.12	3.87
10	31.12	3.54	3.21	3.05	2.28	3.58

Table C-2 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-3 Arena output data

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	1.46	3.89	5.83	7.06	1.46	3.89	5.83	7.06	24.58
2	2.17	6.45	25.63	30.04	50.40	6.45	25.63	30.04	50.40	51.92
3	6.93	25.63	30.04	50.40	53.54	24.68	27.08	48.91	53.54	55.72
4	24.68	29.04	49.93	54.41	74.96	29.04	49.93	54.41	74.96	79.94
5	29.04	49.93	54.41	74.96	79.94	30.95	53.90	72.39	77.80	97.01
6	30.95	53.90	72.54	77.80	97.43	51.36	72.54	76.43	97.43	99.08
7	51.36	72.54	76.43	97.43	101.61	54.15	74.06	97.32	101.61	121.28
8	54.15	74.06	97.32	101.61	122.42	72.36	75.39	99.60	122.42	126.92
9	72.36	77.12	99.60	122.42	126.92	77.12	97.72	102.37	124.54	146.79
10	77.12	97.72	102.37	124.54	146.79	96.66	100.93	121.42	126.82	150.37

CPLEX model COM-W-Q output is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

C.2 Dataset B- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 1

Table C-4 Input data - from Arena

		Stages				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	4.44	4.03	4.52	2.70	4.81
2	1.40	1.37	4.86	4.77	3.34	3.64
3	2.75	3.90	3.74	2.22	3.08	3.59
4	5.19	4.98	2.68	1.79	1.12	3.09
5	7.93	3.55	4.32	1.31	2.67	3.70
6	10.82	1.16	3.01	1.76	2.13	3.76
7	12.75	3.04	3.62	3.65	1.38	1.34
8	16.49	4.73	1.73	4.07	3.85	4.71
9	20.34	2.71	3.94	1.23	2.28	2.07
10	24.10	3.51	3.89	3.36	4.72	2.80

Table C-5 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-6 Arena output data

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	4.44	24.47	28.99	31.69	4.44	24.47	28.99	31.69	52.50
2	4.44	24.47	29.33	50.10	53.44	5.81	29.33	50.10	53.44	73.08
3	24.47	29.33	50.10	53.44	73.08	28.37	49.07	52.32	72.52	76.67
4	29.33	50.31	53.44	73.08	76.67	50.31	52.99	55.23	74.20	79.76
5	50.31	53.86	74.18	76.67	79.76	53.86	74.18	75.49	79.34	99.46
6	53.86	74.18	77.19	79.76	99.46	55.02	77.19	78.95	97.89	103.22
7	74.18	77.22	96.84	100.49	103.22	77.22	96.84	100.49	101.87	120.56
8	77.22	97.95	100.49	120.56	124.41	97.95	99.68	120.56	124.41	145.12
9	97.95	100.66	120.60	124.41	145.12	100.66	120.60	121.83	126.69	147.19
10	100.66	120.60	124.49	145.12	149.84	120.17	124.49	127.85	149.84	168.64

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

C.3 Dataset C- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 1

Table C-7 Input data - from Arena

		Stages				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	3.41	1.01	2.81	3.03	1.97
2	1.51	4.13	4.21	1.56	3.05	3.32
3	3.91	2.82	3.05	3.85	1.32	3.79
4	8.76	3.31	4.09	1.99	4.47	3.90
5	12.68	3.76	4.72	4.15	1.20	2.36
6	13.72	1.25	3.31	2.47	3.85	1.50
7	18.44	1.13	1.28	3.11	3.47	2.94
8	22.39	4.10	2.85	4.69	3.95	1.79
9	25.14	1.34	2.16	1.61	4.86	1.04
10	26.34	3.24	2.27	3.87	1.33	1.88

Table C-8 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-9 Arena output data

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	3.41	4.42	7.23	26.26	3.41	4.42	7.23	26.26	28.23
2	3.41	7.54	27.75	29.31	48.36	7.54	27.75	29.31	48.36	51.68
3	7.54	27.75	30.80	50.65	51.97	26.36	30.80	50.65	51.97	55.76
4	27.75	31.06	51.15	53.14	73.61	31.06	51.15	53.14	73.61	77.51
5	31.06	51.15	55.87	76.02	77.51	50.82	55.87	76.02	77.22	79.87
6	51.15	55.87	76.02	78.49	98.34	52.40	75.18	78.49	98.34	99.84
7	55.87	76.02	78.49	98.34	101.81	73.00	77.30	97.60	101.81	120.75
8	76.02	96.12	98.97	103.66	123.61	96.12	98.97	103.66	123.61	125.40
9	96.12	98.97	103.66	123.61	144.47	97.46	101.13	121.27	144.47	145.51
10	98.97	103.66	123.61	144.47	145.80	102.21	121.93	127.48	145.80	147.68

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

C.4 Dataset D- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 1

Table C-10 Input data - from Arena

		Stages				
Job	Arrival	1	2	3	4	5
1	0.00	1.26	1.96	1.63	1.02	1.31
2	0.69	3.47	2.47	3.65	3.58	1.31
3	6.40	1.49	1.25	2.25	2.44	1.79
4	12.60	3.58	5.07	3.78	3.90	6.73
5	12.99	1.61	3.12	1.65	2.23	0.77
6	15.71	3.66	2.10	3.04	2.81	1.41
7	18.68	2.20	1.31	5.09	3.35	2.84
8	21.53	1.82	1.13	1.86	4.62	3.80
9	22.36	4.42	4.00	2.18	1.75	3.02
10	23.95	2.74	2.48	2.37	1.86	2.77

Table C-11 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	2	3	4	5	6	7	8
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-12 Arena output data

	StartTime					FinishTime				
Job	S1	S2	S3	S4	S5	F1	F2	F3	F4	F5
1	0.00	1.26	3.22	4.85	5.87	1.26	3.22	4.85	5.87	7.18
2	1.26	4.73	7.20	26.85	30.43	4.73	7.20	26.85	30.43	31.74
3	6.40	7.89	26.85	30.43	48.87	7.89	25.14	29.10	48.87	50.66
4	24.00	27.58	48.65	52.43	72.33	27.58	48.65	52.43	72.33	79.06
5	27.58	48.65	52.43	72.33	79.06	29.19	51.77	54.08	74.56	79.83
6	48.65	52.43	72.33	79.06	97.87	52.31	54.53	75.37	97.87	99.28
7	52.43	72.33	79.06	100.15	103.50	54.63	73.64	100.15	103.50	122.34
8	72.33	79.06	100.15	103.50	124.12	74.15	96.19	102.01	124.12	127.92
9	79.06	100.15	120.15	124.12	127.92	99.48	120.15	122.33	125.87	146.94
10	100.15	120.15	124.12	127.92	146.94	102.89	122.63	126.49	145.78	149.71

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

C.5 Dataset E- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 1

Table C-13 Input data - from Arena

		Stages				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	3.71	3.18	3.84	2.14	4.63
2	0.21	1.17	4.89	4.47	2.58	2.81
3	0.39	3.05	2.90	1.82	2.39	2.78
4	1.28	6.37	2.12	1.52	0.88	2.40
5	2.42	2.74	3.53	1.11	2.11	2.86
6	3.70	0.93	2.34	1.50	1.76	2.92
7	4.23	2.37	2.80	2.82	1.18	1.14
8	6.54	4.33	1.48	3.22	3.00	4.26
9	9.04	2.14	3.09	1.02	1.86	1.72
10	11.39	2.71	3.04	2.60	4.30	2.20

Table C-14 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-15 Arena output data

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	3.71	6.89	26.73	28.87	3.71	6.89	26.73	28.87	49.50
2	3.71	6.89	27.78	48.25	50.83	4.88	27.78	48.25	50.83	53.64
3	6.89	27.78	48.25	50.83	53.64	25.94	30.68	50.07	53.22	72.42
4	27.78	50.15	52.27	53.79	72.42	50.15	52.27	53.79	54.67	74.82
5	50.15	52.89	72.42	73.53	75.64	52.89	72.42	73.53	75.64	78.50
6	52.89	72.42	74.76	76.26	78.50	53.82	74.76	76.26	78.02	97.42
7	72.42	74.79	77.59	96.41	97.59	74.79	77.59	96.41	97.59	98.73
8	74.79	79.12	96.60	99.82	102.82	79.12	96.60	99.82	102.82	123.08
9	79.12	97.26	100.35	102.82	123.08	97.26	100.35	101.37	120.68	124.80
10	97.26	100.35	103.39	123.08	127.38	99.97	103.39	121.99	127.38	145.58

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

C.6 Dataset A- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 3

This experiment setting has stage capacity of 3.

Table C-16 Input data - from Arena

		Stages				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	4.44	4.03	4.52	2.70	4.81
2	1.40	1.37	4.86	4.77	3.34	3.64
3	2.75	3.90	3.74	2.22	3.08	3.59
4	5.19	4.98	2.68	1.79	1.12	3.09
5	7.93	3.55	4.32	1.31	2.67	3.70
6	10.82	1.16	3.01	1.76	2.13	3.76
7	12.75	3.04	3.62	3.65	1.38	1.34
8	16.49	4.73	1.73	4.07	3.85	4.71
9	20.34	2.71	3.94	1.23	2.28	2.07
10	24.10	3.51	3.89	3.36	4.72	2.80

Table C-17 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-18 Arena output data

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	4.44	24.47	28.99	31.69	4.44	24.47	28.99	31.69	52.50
2	4.44	24.47	29.33	50.10	53.44	5.81	29.33	50.10	53.44	73.08
3	5.81	29.33	50.10	53.44	73.08	25.71	49.07	52.32	72.52	76.67
4	25.71	49.07	52.32	72.52	76.67	30.69	51.75	54.11	73.64	79.76
5	30.69	51.75	72.07	73.64	79.76	50.24	72.07	73.38	76.31	99.46
6	50.24	72.07	75.08	76.84	99.46	51.40	75.08	76.84	78.97	103.22
7	51.40	75.08	78.70	98.35	103.22	54.44	78.70	98.35	99.73	120.56
8	54.44	78.70	98.35	102.42	122.27	75.17	96.43	102.42	122.27	126.98
9	75.17	96.43	102.42	122.27	126.98	77.88	100.37	103.65	124.55	145.05
10	77.88	100.37	120.26	124.55	145.27	97.39	120.26	123.62	145.27	148.07

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

The last job is completed at time 148.07 in above data set. By setting the due date for the last job (job 10) at the last stage (stage 5) to 124, the overtime optimal solution is returned from the CPLEX model OPT-OT model as below.

Table C-18.5 Overtime for due date at time 128 for the last job

	Downtime occurrence							
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0	0.43	0	0	0	0	0	0
2	1.94	0	1.04	0	0	0	0	0
3	0	0.27	0	1.09	0.16	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1.06	0	0	0

Table C-19 CPLEX overtime model OPT-OT output

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	4.44	24.00	28.52	31.22	4.44	8.47	28.52	31.22	52.03
2	4.44	8.47	28.52	49.02	52.36	5.81	27.39	49.02	52.36	56.00
3	5.81	27.39	49.02	52.36	56.00	25.71	31.13	51.24	55.44	75.59
4	25.71	31.13	51.24	55.44	75.59	30.69	49.81	53.03	72.56	78.68
5	30.69	49.81	54.13	72.56	78.68	49.81	54.13	55.44	75.23	98.38
6	49.81	54.13	72.10	75.23	98.38	50.97	72.10	73.86	77.36	102.14
7	50.97	72.10	75.72	79.37	102.14	54.01	75.72	79.37	96.75	103.48
8	54.01	75.72	79.37	98.35	103.48	74.74	77.45	98.35	102.20	123.13
9	74.74	77.45	98.35	102.20	123.13	77.45	97.39	99.58	120.48	125.20
10	77.45	97.39	101.28	120.48	125.20	96.96	101.28	120.48	125.20	128.00

Table C-20 Completion time output for CPLEX completion time model with overtime updated

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	4.44	24.00	28.52	31.22	4.44	8.47	28.52	31.22	52.03
2	4.44	8.47	28.52	49.02	52.36	5.81	27.39	49.02	52.36	56.00
3	5.81	27.39	49.02	52.36	56.00	25.71	31.13	51.24	55.44	75.59
4	25.71	31.13	51.24	55.44	75.59	30.69	49.81	53.03	72.56	78.68
5	30.69	49.81	54.13	72.56	78.68	49.81	54.13	55.44	75.23	98.38
6	49.81	54.13	72.10	75.23	98.38	50.97	72.10	73.86	77.36	102.14
7	50.97	72.10	75.72	79.37	102.14	54.01	75.72	79.37	96.75	103.48
8	54.01	75.72	79.37	98.35	103.48	74.74	77.45	98.35	102.20	123.13
9	74.74	77.45	98.35	102.20	123.13	77.45	97.39	99.58	120.48	125.20
10	77.45	97.39	101.28	120.48	125.20	96.96	101.28	120.48	125.20	128.00

Table C-21 OT output for CPLEX overtime model with due dates updated

	Downtime occurrence							
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0	0.43	0	0	0	0	0	0
2	1.94	0	1.04	0	0	0	0	0
3	0	0.27	0	1.09	0.16	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	1.06	0	0	0

There is no discrepancy between the sum of all the OT hours from OT model with and OT model without due dates update.

Table C-22 Discrepancies between OPT-OT-ALT & COM-OPT

	Start time discrepancies									
Stage	1	2	3	4	5	6	7	8	9	10
1	0.00	0.00	0.00	-0.59	-0.59	-0.16	-0.16	-0.16	-0.16	-0.16
2	0.00	0.00	-0.16	-0.16	-0.16	-0.16	-15.12	-1.09	-0.16	-0.16
3	0.00	0.00	-0.33	-0.33	-0.16	-0.10	-1.09	-1.09	0.00	-0.16
4	0.00	-0.33	-0.33	-0.33	-0.33	-0.33	-1.09	0.00	0.00	0.00
5	0.00	-0.33	-0.33	-1.06	-1.06	-1.06	-1.06	-1.06	0.00	0.00
	Finish time discrepancies									
1	0.00	0.00	-0.59	-0.59	-0.16	-0.16	-0.16	-0.16	-0.16	-0.16
2	0.00	-0.16	-0.16	-0.16	-0.16	-15.12	-1.09	-1.09	-0.16	-0.16
3	0.00	-0.33	-0.33	-0.33	-0.16	-0.10	-1.09	0.00	0.00	0.00
4	0.00	-0.33	-0.33	-0.33	-0.33	-0.33	-17.09	0.00	0.00	0.00
5	0.00	-0.33	-1.06	-1.06	-1.06	-1.06	-1.06	0.00	0.00	0.00

C.7 Dataset B- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 3

This experiment setting has stage capacity of 3.

Table C-23 Input data - from Arena

		Stages				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	3.41	1.01	2.81	3.03	1.97
2	1.51	4.13	4.21	1.56	3.05	3.32
3	3.91	2.82	3.05	3.85	1.32	3.79
4	8.76	3.31	4.09	1.99	4.47	3.90
5	12.68	3.76	4.72	4.15	1.20	2.36
6	13.72	1.25	3.31	2.47	3.85	1.50
7	18.44	1.13	1.28	3.11	3.47	2.94
8	22.39	4.10	2.85	4.69	3.95	1.79
9	25.14	1.34	2.16	1.61	4.86	1.04
10	26.34	3.24	2.27	3.87	1.33	1.88

Table C-24 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-25 Arena output data

	StartTime					FinishTime				
Job	S1	S2	S3	S4	S5	F1	F2	F3	F4	F5
1	0.00	3.41	4.42	7.23	26.26	3.41	4.42	7.23	26.26	28.23
2	3.41	7.54	27.75	29.31	48.36	7.54	27.75	29.31	48.36	51.68
3	7.54	27.75	30.80	50.65	51.97	26.36	30.80	50.65	51.97	55.76
4	26.36	30.80	50.89	52.88	73.35	29.67	50.89	52.88	73.35	77.25
5	29.67	50.89	55.61	75.76	77.25	49.43	55.61	75.76	76.96	79.61
6	49.43	55.61	75.76	78.23	98.08	50.68	74.92	78.23	98.08	99.58
7	50.68	74.92	78.23	98.08	101.55	51.81	76.20	97.34	101.55	120.49
8	51.81	76.20	97.34	102.03	121.98	55.91	79.05	102.03	121.98	123.77
9	55.91	79.05	102.03	121.98	126.84	73.25	97.21	103.64	126.84	127.88
10	74.92	97.21	103.64	126.84	144.17	78.16	99.48	123.51	144.17	146.05

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

Table C-26 Overtime for due date at time 128 for the last job

	Downtime occurrence							
Stage	1	2	3	4	5	6	7	8
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	1.13	0.21	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.71	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table C-27 CPLEX overtime model OPT-OT output

	StartTime					FinishTime				
Job	S1	S2	S3	S4	S5	F1	F2	F3	F4	F5
1	0.00	3.41	4.42	7.23	26.26	3.41	4.42	7.23	26.26	28.23
2	3.41	7.54	26.62	28.18	48.36	7.54	26.62	28.18	48.36	51.68
3	7.54	26.62	29.67	50.65	51.97	26.36	29.67	50.65	51.97	55.76
4	26.36	29.67	50.65	52.64	73.35	29.67	49.55	52.64	73.35	77.25
5	29.67	49.55	54.27	74.42	77.25	49.43	54.27	74.42	76.96	79.61
6	49.43	54.27	74.42	76.96	96.81	50.68	74.42	76.89	96.81	99.58
7	50.68	74.92	76.89	96.81	100.69	51.81	76.20	80.00	100.69	103.63
8	51.81	76.20	96.00	100.69	120.00	55.91	79.05	100.69	104.64	121.79
9	55.91	79.05	100.69	104.64	124.79	73.25	97.21	102.30	124.79	125.83
10	74.92	97.21	102.30	124.79	126.12	78.16	99.48	122.17	126.12	128.00

Table C-28 Completion time output for CPLEX completion time model with overtime updated

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	3.41	4.42	7.23	26.26	3.41	4.42	7.23	26.26	28.23
2	3.41	7.54	26.62	28.18	31.23	7.54	26.62	28.18	31.23	50.55
3	7.54	26.62	29.67	49.52	50.84	26.36	29.67	49.52	50.84	54.63
4	26.36	29.67	49.55	51.54	72.01	29.67	49.55	51.54	72.01	75.91
5	29.67	49.55	54.27	74.42	75.91	49.43	54.27	74.42	75.62	78.27
6	49.43	54.27	74.42	76.89	96.74	50.68	73.58	76.89	96.74	98.24
7	50.68	73.58	76.89	96.74	100.21	51.81	74.86	80.00	100.21	103.15
8	51.81	74.86	80.00	100.69	120.00	55.91	77.71	100.69	104.64	121.79
9	55.91	77.71	100.69	104.64	124.79	73.25	79.87	102.30	124.79	125.83
10	73.58	79.87	102.30	124.79	126.12	76.82	98.14	122.17	126.12	128.00

Table C-29 OT output for CPLEX overtime model with due dates updated

	Downtime occurrence							
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	**1.13**	**0.21**	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	**0.16**	**0.48**	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	**0.07**	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

There is no discrepancy between the sum of all the OT hours from OT model with and OT model without due dates update.

Table C-30 Discrepancies between OPT-OT-ALT & COM-OPT

	Start time discrepancies									
Stage	1	2	3	4	5	6	7	8	9	10
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.12
2	0.00	0.00	0.00	0.00	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12
3	0.00	0.00	0.00	-0.03	-0.12	-0.71	-0.71	-0.71	-0.71	-0.71
4	0.00	0.00	0.00	-0.03	-0.71	-0.71	-0.71	-0.71	-0.71	0.00
5	0.00	0.00	0.00	-16.03	-0.03	-0.71	-0.71	-16.07	0.00	0.00
	Finish time discrepancies									
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.12
2	0.00	0.00	0.00	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12	-0.12
3	0.00	0.00	0.00	-0.03	-0.71	-0.71	-0.71	-0.71	-0.71	-0.71
4	0.00	0.00	0.00	-16.03	-0.71	-0.71	-0.71	-0.71	0.00	0.00
5	0.00	0.00	0.00	-0.03	-0.03	-0.71	-0.71	-0.07	0.00	0.00

C.8 Dataset C- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 3

This experiment setting has stage capacity of 3.

Table C-31 Input data - from Arena

		Stages				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	1.46	2.43	1.94	1.23	1.52
2	2.17	4.28	3.18	4.41	4.36	1.52
3	6.93	1.75	1.45	2.87	3.14	2.18
4	11.75	4.36	4.89	4.48	4.55	4.98
5	13.47	1.91	3.97	1.98	2.84	1.07
6	17.44	4.41	2.64	3.89	3.63	1.65
7	21.53	2.79	1.52	4.89	4.18	3.67
8	25.57	2.21	1.33	2.28	4.81	4.50
9	27.93	4.76	4.60	2.77	2.12	3.87
10	31.12	3.54	3.21	3.05	2.28	3.58

Table C-32 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-33 Arena output data

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	1.46	3.89	5.83	7.06	1.46	3.89	5.83	7.06	24.58
2	2.17	6.45	25.63	30.04	50.40	6.45	25.63	30.04	50.40	51.92
3	6.93	25.63	30.04	50.40	53.54	24.68	27.08	48.91	53.54	55.72
4	24.68	29.04	49.93	54.41	74.96	29.04	49.93	54.41	74.96	79.94
5	29.04	49.93	54.41	74.96	79.94	30.95	53.90	72.39	77.80	97.01
6	30.95	53.90	72.54	77.80	97.43	51.36	72.54	76.43	97.43	99.08
7	51.36	72.54	76.43	97.43	101.61	54.15	74.06	97.32	101.61	121.28
8	54.15	74.06	97.32	101.61	122.42	72.36	75.39	99.60	122.42	126.92
9	72.36	77.12	99.60	122.42	126.92	77.12	97.72	102.37	124.54	146.79
10	77.12	97.72	102.37	124.54	146.79	96.66	100.93	121.42	126.82	150.37

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

Table C-34 Overtime for due date at time 128 for the last job

	Downtime occurrence							
Stage	1	2	3	4	5	6	7	8
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	1.02	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00
4	0.00	0.00	0.14	2.98	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	2.38	0.00	0.00	0.00

Table C-35 CPLEX overtime model OPT-OT output

	StartTime					FinishTime				
Job	S1	S2	S3	S4	S5	F1	F2	F3	F4	F5
1	0.00	1.46	3.89	5.83	7.06	1.46	3.89	5.83	7.06	24.58
2	2.17	6.45	25.63	30.04	50.40	6.45	25.63	30.04	50.40	51.92
3	6.93	25.63	30.04	50.40	53.54	24.68	27.08	48.91	53.54	55.72
4	24.68	29.04	48.91	53.54	73.95	29.04	48.91	53.54	73.95	78.93
5	29.04	48.91	53.54	73.95	78.93	30.95	52.88	55.52	76.79	80.00
6	30.95	52.88	55.52	76.79	80.42	51.36	55.52	75.41	80.42	98.76
7	51.36	55.52	75.41	80.42	98.76	54.15	73.04	80.30	97.62	102.43
8	54.15	73.04	80.30	97.62	102.43	72.36	74.37	97.62	102.43	120.55
9	72.36	77.12	97.72	102.43	120.55	77.12	97.72	102.37	120.55	124.42
10	77.12	97.72	102.37	121.42	124.42	96.66	100.93	121.42	124.42	128.00

Table C-36 Completion time output for CPLEX completion time model with overtime updated

	StartTime					FinishTime				
Job	S1	S2	S3	S4	S5	F1	F2	F3	F4	F5
1	0.00	1.46	3.89	5.83	7.06	1.46	3.89	5.83	7.06	24.58
2	2.17	6.45	25.63	30.04	50.40	6.45	25.63	30.04	50.40	51.92
3	6.93	25.63	30.04	50.40	53.54	24.68	27.08	48.91	53.54	55.72
4	24.68	29.04	48.91	53.54	73.95	29.04	48.91	53.39	73.95	78.93
5	29.04	48.91	53.39	73.95	78.93	30.95	52.88	55.37	76.79	80.00
6	30.95	52.88	55.52	76.79	96.00	51.36	55.52	75.41	80.42	97.65
7	51.36	55.52	75.41	80.42	97.65	54.15	73.04	80.30	97.62	101.32
8	54.15	73.04	80.30	97.62	102.43	72.36	74.37	97.62	102.43	120.55
9	72.36	77.12	97.72	102.43	120.55	77.12	97.72	100.49	120.55	124.42
10	77.12	97.72	100.93	120.55	124.42	96.66	100.93	103.98	122.83	128.00

TableC-37 OT output for CPLEX overtime model with due dates updated

	Downtime occurrence							
Stage	1	2	3	4	5	6	7	8
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	**1.02**	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	**0.30**	**0.66**	0.00	0.00	0.00	0.00
4	0.00	0.00	**0.56**	**2.56**	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	**2.38**	0.00	0.00	0.00

There is no discrepancy between the sum of all the OT hours from OT model with and OT model without due dates update.

Table C-38 Discrepancies between OPT-OT-ALT & COM-OPT

	Start time discrepancies									
Stage	1	2	3	4	5	6	7	8	9	10
1	0.00	0.00	0.00	-0.68	-0.68	-0.68	-0.68	-0.68	-16.68	-0.68
2	0.00	0.00	-1.06	-0.68	-15.66	-0.96	-0.96	-0.96	-0.68	-0.68
3	0.00	-1.06	-1.06	-0.91	-0.91	-0.96	-0.96	-0.96	-0.10	-0.54
4	0.00	-1.06	-1.06	-1.06	-1.08	-1.08	-1.08	0.00	0.00	0.00
5	0.00	-1.06	-1.06	-1.08	-1.08	-16.66	-0.03	0.00	0.00	0.00
	Finish time discrepancies									
1	0.00	0.00	-0.68	-0.68	-0.68	-0.68	-0.68	-16.68	-0.68	-16.68
2	0.00	-1.06	-1.06	-15.66	-0.96	-0.96	-0.96	-0.96	-0.68	-0.68
3	0.00	-1.06	-17.06	-0.91	-0.91	-0.96	-0.96	0.00	-0.10	-0.54
4	0.00	-1.06	-1.06	-1.08	-1.08	-1.08	0.00	0.00	0.00	0.00
5	0.00	-1.06	-1.06	-1.08	-1.08	-0.66	-0.03	0.00	0.00	0.00

C.9 Dataset D- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 3

This experiment setting has stage capacity of 3.

Table C-39 Input data - from Arena

		Stages				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	1.26	1.96	1.63	1.02	1.31
2	0.69	3.47	2.47	3.65	3.58	1.31
3	6.40	1.49	1.25	2.25	2.44	1.79
4	12.60	3.58	5.07	3.78	3.90	6.73
5	12.99	1.61	3.12	1.65	2.23	0.77
6	15.71	3.66	2.10	3.04	2.81	1.41
7	18.68	2.20	1.31	5.09	3.35	2.84
8	21.53	1.82	1.13	1.86	4.62	3.80
9	22.36	4.42	4.00	2.18	1.75	3.02
10	23.95	2.74	2.48	2.37	1.86	2.77

Table C-40 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-41 Arena output data

	StartTime					FinishTime				
Job	S1	S2	S3	S4	S5	F1	F2	F3	F4	F5
1	0.00	1.26	3.22	4.85	5.87	1.26	3.22	4.85	5.87	7.18
2	1.26	4.73	7.20	26.85	30.43	4.73	7.20	26.85	30.43	31.74
3	6.40	7.89	26.85	30.43	48.87	7.89	25.14	29.10	48.87	50.66
4	24.00	27.58	48.65	52.43	72.33	27.58	48.65	52.43	72.33	79.06
5	27.58	48.65	52.43	72.33	79.06	29.19	51.77	54.08	74.56	79.83
6	29.19	51.77	54.08	74.56	79.83	48.85	53.87	73.12	77.37	97.24
7	48.85	53.87	73.12	78.21	97.56	51.05	55.18	78.21	97.56	100.40
8	51.05	55.18	78.21	97.56	102.18	52.87	72.31	96.07	102.18	121.98
9	52.87	73.29	96.07	102.18	121.98	73.29	77.29	98.25	103.93	125.00
10	73.29	77.29	98.25	103.93	125.00	76.03	79.77	100.62	121.79	127.77

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

Table C-42 Overtime for due date at time 104 for the last job

	Downtime occurrence							
Stage	1	2	3	4	5	6	7	8
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.65	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	1.60	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.76	3.93	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	5.52	0.00	0.00	0.00	0.00

Table C-43 CPLEX overtime model OPT-OT output

	StartTime					FinishTime				
Job	S1	S2	S3	S4	S5	F1	F2	F3	F4	F5
1	0.00	1.26	3.22	4.85	5.87	1.26	3.22	4.85	5.87	7.18
2	1.26	4.73	7.20	26.85	30.43	4.73	7.20	26.85	30.43	31.74
3	6.40	7.89	26.85	30.43	48.87	7.89	25.14	29.10	48.87	50.66
4	24.00	27.58	48.00	51.78	72.18	27.58	48.00	51.78	55.68	78.91
5	27.58	48.00	51.78	55.68	78.91	29.19	51.33	53.43	73.15	79.68
6	29.19	51.33	53.43	73.15	79.68	48.85	53.43	56.47	75.96	81.09
7	48.85	53.43	56.47	75.96	81.09	51.05	55.18	75.96	79.31	83.93
8	51.05	55.18	75.96	79.31	83.93	52.87	72.31	77.82	83.93	98.21
9	52.87	73.29	77.82	83.93	98.21	73.29	77.29	80.00	98.21	101.23
10	73.29	77.29	80.00	99.37	101.23	76.03	79.77	99.37	101.23	104.00

Table C-44 Completion time output for CPLEX completion time model with overtime updated

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	1.26	3.22	4.85	5.87	1.26	3.22	4.85	5.87	7.18
2	1.26	4.73	7.20	26.85	30.43	4.73	7.20	26.85	30.43	31.74
3	6.40	7.89	26.85	30.43	48.87	7.89	25.14	29.10	48.87	50.66
4	24.00	27.58	48.00	51.78	55.68	27.58	32.65	51.78	55.68	78.41
5	27.58	32.65	51.78	55.68	78.41	29.19	51.12	53.43	73.15	79.18
6	29.19	51.12	53.43	73.15	79.18	48.85	53.22	56.47	75.96	80.59
7	48.85	53.22	56.47	75.96	80.59	51.05	54.53	75.96	79.31	83.43
8	51.05	54.53	75.96	79.31	83.93	52.87	55.66	77.82	83.93	98.21
9	52.87	73.29	77.82	83.93	98.21	73.29	77.29	80.00	97.75	101.23
10	73.29	77.29	80.00	98.37	101.23	76.03	79.77	98.37	100.23	104.00

Table C-45 OT output for CPLEX overtime model with due dates updated

	Downtime occurrence							
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	**0.65**	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	**1.60**	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	**0.76**	**3.93**	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	**5.52**	0.00	0.00	0.00	0.00

There is no discrepancy between the sum of all the OT hours from OT model with and OT model without due dates update.

Table C-46 Discrepancies between OPT-OT-ALT & COM-OPT

	Start time discrepancies									
Stage	1	2	3	4	5	6	7	8	9	10
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.50	-0.50	-0.23
4	0.00	0.00	0.00	0.00	0.00	-0.50	-0.50	-0.50	-0.50	-0.23
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	-0.50	0.00	0.00
	Finish time discrepancies									
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	-0.50	-0.50	-0.50	-0.23
4	0.00	0.00	0.00	0.00	-0.50	-0.50	-0.50	-0.50	0.00	-0.23
5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

C.10 Dataset E- 5 Stages, 10 Jobs, 8 Downtimes, Stage Capacity of 3

This experiment setting has stage capacity of 3.

Table C-47 Input data - from Arena

		Stages				
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**
1	0.00	3.71	3.18	3.84	2.14	4.63
2	0.21	1.17	4.89	4.47	2.58	2.81
3	0.39	3.05	2.90	1.82	2.39	2.78
4	1.28	6.37	2.12	1.52	0.88	2.40
5	2.42	2.74	3.53	1.11	2.11	2.86
6	3.70	0.93	2.34	1.50	1.76	2.92
7	4.23	2.37	2.80	2.82	1.18	1.14
8	6.54	4.33	1.48	3.22	3.00	4.26
9	9.04	2.14	3.09	1.02	1.86	1.72
10	11.39	2.71	3.04	2.60	4.30	2.20

Table C-48 Downtime data for all the down occurrences

	Downtime Occurrences							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Start time	8	32	56	80	104	128	152	176
Duration	16	16	16	16	16	16	16	16

Table C-49 Arena output data

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	3.71	6.89	26.73	28.87	3.71	6.89	26.73	28.87	49.50
2	3.71	6.89	27.78	48.25	50.83	4.88	27.78	48.25	50.83	53.64
3	4.88	27.78	48.25	50.83	53.64	7.93	30.68	50.07	53.22	72.42
4	7.93	30.68	50.07	53.22	72.42	30.30	48.80	51.59	54.10	74.82
5	30.30	49.04	52.57	54.10	74.82	49.04	52.57	53.68	72.21	77.68
6	49.04	52.57	54.91	72.41	77.68	49.97	54.91	72.41	74.17	96.60
7	49.97	54.91	73.71	76.53	96.60	52.34	73.71	76.53	77.71	97.74
8	52.34	73.71	76.53	79.75	98.75	72.67	75.19	79.75	98.75	103.01
9	72.67	75.19	79.75	98.75	103.01	74.81	78.28	96.77	100.61	120.73
10	74.81	78.28	97.32	100.61	120.91	77.52	97.32	99.92	120.91	123.11

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

Table C-50 Overtime for due date at time 124 for the last job

	Downtime occurrence							
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	2.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	1.05	1.82	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.87	0.00	0.00	0.00	0.00

Table C-51 CPLEX overtime model OPT-OT output

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	3.71	6.89	26.73	28.87	3.71	6.89	26.73	28.87	49.50
2	3.71	6.89	26.73	31.20	49.78	4.88	26.73	31.20	49.78	52.59
3	4.88	26.73	48.00	50.07	52.59	7.93	29.63	50.07	52.59	55.37
4	7.93	29.63	50.07	52.59	55.37	28.26	31.75	51.59	53.47	73.77
5	28.26	31.75	51.59	53.47	73.77	31.00	49.46	53.10	55.58	76.63
6	31.00	49.46	53.10	55.58	76.63	31.93	51.80	54.60	74.17	79.55
7	31.93	51.80	54.60	74.17	79.55	50.30	54.60	73.42	76.64	80.69
8	50.30	54.63	73.42	76.64	80.69	54.63	72.77	76.64	79.64	100.08
9	54.63	72.77	76.64	79.64	100.08	72.77	75.86	78.90	97.50	101.80
10	72.77	75.86	78.90	97.50	101.80	75.86	78.90	97.50	101.80	104.00

Table C-52 Completion time output for CPLEX completion time model with overtime updated

	StartTime					FinishTime				
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**F1**	**F2**	**F3**	**F4**	**F5**
1	0.00	3.71	6.89	26.73	28.87	3.71	6.89	26.73	28.87	49.50
2	3.71	6.89	26.73	31.20	49.78	4.88	26.73	31.20	49.78	52.59
3	4.88	26.73	31.20	49.78	52.59	7.93	29.63	49.02	52.17	55.37
4	7.93	29.63	49.02	52.17	55.37	28.26	31.75	50.54	53.05	73.77
5	28.26	31.75	50.54	53.05	73.77	31.00	49.46	51.65	55.16	76.63
6	31.00	49.46	51.80	55.16	76.63	31.93	51.80	53.30	72.92	79.55
7	31.93	51.80	54.60	73.42	79.55	50.30	54.60	73.42	74.60	80.69
8	50.30	54.63	73.42	76.64	80.69	54.63	72.11	76.64	79.64	100.08
9	54.63	72.77	76.64	79.64	100.08	72.77	75.86	77.66	97.50	101.80
10	72.77	75.86	78.90	97.50	101.80	75.48	78.90	97.50	101.80	104.00

Table C-53 OT output for CPLEX overtime model with due dates updated

	Downtime occurrence							
Stage	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	**2.04**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	**1.80**	**1.07**	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	**0.87**	0.00	0.00	0.00	0.00	0.00

There is no discrepancy between the sum of all the OT hours from OT model with and OT model without due dates update.

TableC-54 Discrepancies between OPT-OT-ALT & COM-OPT

	Start time discrepancies									
Stage	1	2	3	4	5	6	7	8	9	10
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	-0.75	-0.75	-0.75	0.00	0.00	0.00	0.00	0.00
3	0.00	-0.75	-0.75	-0.75	-0.75	0.00	0.00	0.00	0.00	0.00
4	-0.75	-0.75	-0.87	-0.87	-0.87	-0.87	0.00	0.00	0.00	0.00
5	-0.75	-0.87	-0.87	-0.87	-0.87	-0.87	-0.87	-0.87	0.00	0.00
	Finish time discrepancies									
1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	-0.75	-0.75	-0.75	0.00	0.00	0.00	0.00	0.00	0.00
3	-0.75	-0.75	-0.75	-0.75	-0.75	0.00	0.00	0.00	0.00	0.00
4	-0.75	-0.87	-0.87	-0.87	-0.87	-0.87	0.00	0.00	0.00	0.00
5	-0.75	-0.87	-0.87	-0.87	-0.87	-0.87	-0.87	0.00	0.00	0.00

C.11 Dataset A- 10 Stages, 15 Jobs, 15 Downtimes, Stage Capacity of 2

Table C-55 Input data - from Arena

		Stages									
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.00	1.52	4.76	3.18	4.41	4.36	2.43	1.94	1.23	1.52	1.46
2	2.17	4.55	1.75	1.72	4.89	4.48	4.82	1.45	2.87	3.14	2.18
3	6.45	3.89	1.07	1.91	4.09	2.64	4.36	3.97	3.97	1.98	2.84
4	11.43	1.33	4.18	3.67	2.79	2.36	1.65	4.41	4.04	1.52	4.89
5	15.06	3.38	2.77	2.12	3.87	4.76	4.81	4.50	2.21	3.19	4.60
6	17.34	2.19	4.74	3.49	4.71	1.05	3.05	2.28	3.58	3.54	4.59
7	20.55	2.50	2.23	1.59	1.01	2.68	2.84	3.12	3.39	2.42	1.96
8	22.60	2.57	3.77	1.93	4.82	4.95	4.72	3.24	3.36	4.83	1.64
9	23.86	3.42	3.89	2.13	4.99	4.26	1.86	3.25	4.57	2.09	2.39
10	28.55	4.00	3.40	2.12	2.78	2.49	1.04	4.97	1.36	2.11	1.69
11	32.19	1.49	2.30	4.06	2.90	2.68	1.01	2.19	2.46	2.78	3.77
12	36.94	2.83	2.67	2.47	4.33	4.78	2.57	2.83	3.55	4.43	4.65
13	41.89	4.75	3.94	4.69	1.58	4.91	3.20	1.85	4.74	4.12	1.05
14	43.69	3.35	3.49	4.32	1.47	1.42	3.88	2.72	2.07	4.17	2.86
15	44.80	1.72	1.91	1.40	4.66	2.70	1.80	1.74	2.58	1.78	2.44

Table C-56 Downtime data for all the down occurrences

	Downtime Occurrences														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Start time	8	32	56	80	104	128	152	176	200	224	248	272	296	320	344
Duration	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16

Table C-57 Arena output - job start times

	StartTime									
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
1	0.00	1.52	6.28	25.46	29.87	50.23	52.66	54.60	55.83	73.35
2	2.17	6.72	25.46	29.87	50.76	55.24	76.06	77.51	96.38	99.52
3	6.72	26.61	27.68	50.76	55.24	76.06	96.42	100.39	120.36	122.34
4	26.61	27.94	48.12	54.85	73.88	96.42	100.39	120.80	124.84	126.36
5	27.94	48.12	51.79	73.64	77.51	98.27	120.80	125.30	127.51	147.25
6	31.32	50.89	55.63	77.51	98.27	103.08	125.30	127.58	147.16	151.85
7	49.51	55.63	75.12	98.22	99.32	122.13	127.58	147.16	150.70	172.44
8	52.01	73.86	77.63	99.23	120.05	125.30	146.70	150.55	169.91	174.74
9	55.63	77.63	98.22	120.05	125.04	146.02	149.94	169.91	174.74	192.83
10	75.05	97.52	100.92	125.04	145.30	147.88	169.19	174.48	192.83	195.22
11	79.05	100.92	120.05	127.82	147.79	150.47	174.16	192.35	194.94	197.72
12	97.52	103.22	125.04	146.72	151.05	171.83	192.35	195.18	198.73	219.16
13	100.92	121.89	127.82	151.05	171.83	192.74	195.94	198.73	219.47	223.81
14	121.67	125.83	148.51	168.83	192.74	195.94	199.82	219.47	223.59	243.76
15	125.02	145.32	168.83	171.83	194.16	199.82	218.54	221.54	243.76	246.62

Table C-58 Arena output - job finish times

	Finish Time									
Job	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**
1	1.52	6.28	25.46	29.87	50.23	52.66	54.60	55.83	73.35	74.81
2	6.72	24.47	27.18	50.76	55.24	76.06	77.51	96.38	99.52	101.70
3	26.61	27.68	29.59	54.85	73.88	96.42	100.39	120.36	122.34	125.18
4	27.94	48.12	51.79	73.64	76.24	98.07	120.80	124.84	126.36	147.25
5	31.32	50.89	53.91	77.51	98.27	103.08	125.30	127.51	146.70	151.85
6	49.51	55.63	75.12	98.22	99.32	122.13	127.58	147.16	150.70	172.44
7	52.01	73.86	76.71	99.23	102.00	124.97	146.70	150.55	169.12	174.40
8	54.58	77.63	79.56	120.05	125.00	146.02	149.94	169.91	174.74	192.38
9	75.05	97.52	100.35	125.04	145.30	147.88	169.19	174.48	192.83	195.22
10	79.05	100.92	103.04	127.82	147.79	148.92	174.16	175.84	194.94	196.91
11	96.54	103.22	124.11	146.72	150.47	151.48	192.35	194.81	197.72	217.49
12	100.35	121.89	127.51	151.05	171.83	174.40	195.18	198.73	219.16	223.81
13	121.67	125.83	148.51	168.63	192.74	195.94	197.79	219.47	223.59	240.86
14	125.02	145.32	168.83	170.30	194.16	199.82	218.54	221.54	243.76	246.62
15	126.74	147.23	170.23	192.49	196.86	217.62	220.28	240.12	245.54	265.06

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

C.12 Dataset B- 10 Stages, 15 Jobs, 15 Downtimes, Stage Capacity of 2

Table C-59 Input data - from Arena

		Stages									
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.00	3.64	1.35	4.86	4.77	3.34	4.03	4.52	2.70	4.81	4.44
2	1.40	1.12	3.90	2.74	2.68	1.79	2.44	3.74	2.22	3.08	3.59
3	2.77	1.76	3.70	3.55	1.93	3.01	4.98	2.89	4.32	1.31	2.67
4	5.86	1.73	1.38	1.34	3.04	3.85	3.76	1.16	3.74	3.62	3.65
5	7.99	1.20	1.23	2.28	2.07	2.71	3.85	4.71	4.73	3.76	3.94
6	12.06	2.83	2.58	3.77	4.18	3.02	3.36	4.72	2.80	3.51	1.22
7	15.95	1.74	3.75	4.33	3.91	3.13	1.84	3.31	4.07	3.30	3.69
8	19.62	3.07	3.45	2.24	1.07	1.38	2.33	4.99	3.34	4.22	1.43
9	21.12	1.35	3.93	1.55	4.83	4.10	4.33	2.02	3.82	3.95	1.09
10	25.19	2.71	4.46	3.43	2.48	2.89	4.92	2.32	3.24	4.45	2.25
11	27.80	3.73	2.31	4.11	1.58	1.77	2.06	1.09	3.63	3.75	1.10
12	30.76	1.99	1.59	4.76	1.46	3.25	1.39	2.36	2.02	4.31	1.64
13	33.19	4.93	3.17	4.04	4.87	4.77	2.71	2.61	4.00	3.93	2.15
14	35.78	2.85	1.67	3.52	2.78	3.00	1.35	3.52	2.53	2.78	4.45
15	39.70	2.61	2.28	2.06	2.60	3.56	4.52	2.06	3.56	2.37	1.95

Table C-60 Downtime data for all the down occurrences

	Downtime Occurrences														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Start time	8	32	56	80	104	128	152	176	200	224	248	272	296	320	344
Duration	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16

Table C-61 Arena output - job start times

	StartTime									
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
1	0.00	3.64	4.99	25.85	30.62	49.96	53.99	74.51	77.21	98.02
2	3.64	4.99	25.85	30.62	49.96	53.99	74.51	78.25	98.02	102.46
3	4.76	24.89	28.59	49.30	51.75	72.43	78.25	97.14	101.46	122.05
4	6.52	28.59	48.14	51.23	54.76	77.41	97.17	101.46	121.20	124.82
5	24.89	29.97	49.48	54.27	74.61	97.17	101.02	121.73	126.46	146.22
6	28.59	48.14	51.76	72.34	77.41	101.02	121.73	126.46	146.22	150.16
7	31.42	50.72	55.53	76.52	97.17	120.38	126.45	145.76	149.83	169.13
8	49.16	54.47	75.86	96.43	101.02	122.22	145.76	150.75	170.09	174.31
9	52.23	73.92	78.10	97.50	120.38	126.45	150.75	170.09	174.31	194.26
10	54.47	77.85	98.31	102.33	124.48	146.78	168.77	173.91	194.26	198.71
11	73.92	98.31	101.74	121.85	127.37	151.70	171.09	193.15	198.71	218.46
12	77.85	100.62	121.85	126.61	146.78	169.76	173.91	196.78	218.46	222.77
13	98.31	103.24	126.61	146.65	151.70	172.47	193.15	198.80	222.77	242.70
14	103.24	122.41	146.65	151.52	172.47	175.47	196.78	218.80	242.70	245.48
15	122.09	126.61	150.17	170.30	175.47	195.03	216.30	222.77	245.48	265.93

Table C-62 Arena output - job finish times

	Finish Time									
Job	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**
1	3.64	4.99	25.85	30.62	49.96	53.99	74.51	77.21	98.02	102.46
2	4.76	24.89	28.59	49.30	51.75	72.43	78.25	96.47	101.10	122.05
3	6.52	28.59	48.14	51.23	54.76	77.41	97.14	101.46	102.77	124.72
4	24.25	29.97	49.48	54.27	74.61	97.17	98.33	121.20	124.82	144.47
5	26.09	31.20	51.76	72.34	77.32	101.02	121.73	126.46	146.22	150.16
6	31.42	50.72	55.53	76.52	96.43	120.38	126.45	145.26	149.73	151.38
7	49.16	54.47	75.86	96.43	100.30	122.22	145.76	149.83	169.13	172.82
8	52.23	73.92	78.10	97.50	102.40	124.55	150.75	170.09	174.31	175.74
9	53.58	77.85	79.65	102.33	124.48	146.78	168.77	173.91	194.26	195.35
10	73.18	98.31	101.74	120.81	127.37	151.70	171.09	193.15	198.71	216.96
11	77.65	100.62	121.85	123.43	145.14	169.76	172.18	196.78	218.46	219.56
12	79.84	102.21	126.61	144.07	150.03	171.15	192.27	198.80	222.77	240.41
13	103.24	122.41	146.65	151.52	172.47	175.18	195.76	218.80	242.70	244.85
14	122.09	124.08	150.17	170.30	175.47	192.82	216.30	221.33	245.48	265.93
15	124.70	144.89	168.23	172.90	195.03	199.55	218.36	242.33	247.85	267.88

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

C.13 Dataset C- 10 Stages, 15 Jobs, 15 Downtimes, Stage Capacity of 2

Table C-63 Input data - from Arena

		Stages									
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.00	3.32	2.40	4.21	1.56	3.05	1.01	2.81	3.03	1.97	3.41
2	1.51	4.47	2.82	3.92	4.09	1.99	4.85	3.05	3.85	1.32	3.79
3	5.64	2.47	2.36	3.76	4.72	3.31	3.31	1.04	4.72	4.15	1.20
4	9.54	2.85	3.47	2.94	1.13	2.75	1.50	1.25	3.95	1.28	3.11
5	13.39	2.49	1.61	4.86	1.04	1.34	3.95	1.79	4.10	1.20	2.16
6	18.08	4.08	1.05	1.77	4.23	4.88	3.87	1.33	1.88	3.24	3.04
7	20.35	3.81	2.98	4.38	3.28	4.42	4.55	2.67	2.74	4.05	3.16
8	21.71	4.61	4.02	1.50	4.40	4.59	4.23	2.90	4.24	4.05	2.11
9	24.35	3.56	1.62	1.47	4.40	3.68	1.87	2.05	3.95	1.06	1.82
10	26.16	4.42	2.27	4.29	2.05	4.26	3.31	1.05	1.32	3.06	2.86
11	30.05	2.41	3.58	4.28	2.10	2.98	2.84	1.83	4.54	2.20	1.17
12	32.85	4.93	3.14	2.46	2.71	2.95	2.02	1.92	4.40	2.16	1.20
13	35.15	2.84	1.41	4.31	3.57	1.83	3.05	3.50	3.66	3.52	1.56
14	38.40	3.78	1.52	3.08	3.52	4.87	3.34	2.25	4.60	3.73	1.28
15	43.29	2.35	2.95	3.89	3.11	4.16	4.99	3.91	1.84	1.14	1.84

Table C-64 Downtime data for all the down occurrences

	Downtime Occurrences														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Start time	8	32	56	80	104	128	152	176	200	224	248	272	296	320	344
Duration	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16

Table C-65 Arena output - job start times

	StartTime									
Job	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
1	0.00	3.32	5.72	25.93	27.49	30.54	31.55	50.36	53.39	55.36
2	3.32	7.79	26.61	30.53	50.62	52.61	73.46	76.51	96.36	97.68
3	7.79	26.61	30.53	50.62	55.34	74.65	77.96	96.36	101.08	121.23
4	26.26	29.11	50.29	55.34	74.65	77.96	79.46	101.08	121.23	122.51
5	29.11	48.58	53.23	74.09	77.40	79.46	99.41	121.03	125.13	126.33
6	31.60	51.68	74.09	75.86	96.09	100.97	120.84	125.13	127.01	146.25
7	51.68	55.49	75.86	96.24	100.97	121.39	125.94	144.61	147.35	151.40
8	55.49	76.10	96.24	99.52	121.39	125.98	146.21	149.11	169.35	173.40
9	76.10	96.12	97.74	103.92	125.98	146.21	149.11	169.35	173.40	175.51
10	79.66	100.08	102.35	124.32	145.66	149.92	169.23	173.30	174.62	193.68
11	100.08	102.49	122.64	126.92	149.92	169.23	172.07	174.62	195.16	197.36
12	102.49	123.42	126.92	145.66	168.90	172.07	174.09	195.16	199.56	217.72
13	123.42	126.56	145.38	149.92	171.85	174.09	193.14	199.56	219.22	222.74
14	126.26	146.04	149.69	169.49	173.68	194.55	197.89	219.22	223.82	243.55
15	146.04	148.39	168.77	173.01	194.55	198.71	219.70	223.82	243.55	244.83

Table C-66 Arena output - job finish times

	Finish Time									
Job	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
1	3.32	5.72	25.93	27.49	30.54	31.55	50.36	53.39	55.36	74.77
2	7.79	26.61	30.53	50.62	52.61	73.46	76.51	96.36	97.68	101.47
3	26.26	28.97	50.29	55.34	74.65	77.96	79.00	101.08	121.23	122.43
4	29.11	48.58	53.23	72.47	77.40	79.46	96.71	121.03	122.51	125.62
5	31.60	50.19	74.09	75.13	78.74	99.41	101.20	125.13	126.33	144.49
6	51.68	52.73	75.86	96.09	100.97	120.84	122.17	127.01	146.25	149.29
7	55.49	74.47	96.24	99.52	121.39	125.94	144.61	147.35	151.40	170.56
8	76.10	96.12	97.74	103.92	125.98	146.21	149.11	169.35	173.40	175.51
9	79.66	97.74	99.21	124.32	145.66	148.08	151.16	173.30	174.46	193.33
10	100.08	102.35	122.64	126.37	149.92	169.23	170.28	174.62	193.68	196.54
11	102.49	122.07	126.92	145.02	168.90	172.07	173.90	195.16	197.36	198.53
12	123.42	126.56	145.38	148.37	171.85	174.09	192.01	199.56	217.72	218.92
13	126.26	127.97	149.69	169.49	173.68	193.14	196.64	219.22	222.74	240.30
14	146.04	147.56	168.77	173.01	194.55	197.89	216.14	223.82	243.55	244.83
15	148.39	151.34	172.66	192.12	198.71	219.70	223.61	241.66	244.69	246.67

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

C.14 Dataset D- 10 Stages, 15 Jobs, 15 Downtimes, Stage Capacity of 2

Table C-67 Input data - from Arena

		Stages									
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.00	3.75	3.90	4.20	2.70	3.31	4.97	4.92	4.37	1.33	4.78
2	2.56	4.51	2.33	2.28	3.61	1.00	1.75	4.13	1.29	4.31	3.81
3	4.02	3.82	4.58	4.31	3.18	2.47	4.97	4.38	4.56	1.69	3.69
4	5.49	1.91	3.67	1.17	3.77	1.47	3.68	3.95	3.63	2.24	2.41
5	7.81	4.53	4.50	2.16	2.19	2.02	1.63	3.52	1.97	1.81	4.14
6	11.40	3.65	1.53	2.89	4.20	3.77	3.41	1.67	2.84	4.82	1.03
7	15.31	1.60	4.18	2.27	4.35	1.12	1.53	2.46	4.25	3.23	4.06
8	17.08	4.94	4.10	3.72	3.43	4.18	4.14	2.79	2.39	1.09	3.50
9	21.24	1.01	1.76	1.97	2.14	3.94	1.05	4.88	2.93	4.48	3.94
10	24.02	3.62	4.19	2.27	3.00	3.96	2.51	1.97	2.76	4.73	4.19
11	27.12	4.44	3.65	1.45	1.46	2.50	2.91	3.13	4.26	2.77	3.82
12	30.71	1.30	4.07	2.20	3.92	4.95	2.76	1.80	2.15	3.40	1.93
13	34.99	1.45	2.19	2.16	4.74	2.14	3.69	1.81	3.67	4.00	2.12
14	36.79	3.39	2.43	2.92	2.85	3.42	2.77	4.09	4.52	3.55	4.95
15	40.02	3.24	4.20	1.10	3.18	2.45	2.16	2.63	3.72	1.01	4.06

Table C-68 Downtime data for all the down occurrences

	Downtime Occurrences														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Start time	8	32	56	80	104	128	152	176	200	224	248	272	296	320	344
Duration	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16

Table C-69 Arena output - job start times

	StartTime									
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
1	0.00	3.75	7.65	27.85	30.55	49.86	54.83	75.75	96.12	97.45
2	3.75	24.26	27.85	30.55	50.16	54.83	75.75	96.12	97.45	102.23
3	24.26	28.08	48.66	52.97	72.15	74.62	79.88	100.26	120.82	122.51
4	28.08	48.66	52.97	72.15	75.92	79.59	100.26	120.82	124.45	126.69
5	29.99	52.33	72.83	75.92	78.11	99.27	120.21	124.45	126.69	145.10
6	50.52	72.83	74.99	78.11	98.31	102.08	123.73	126.42	145.26	150.08
7	54.17	74.36	78.54	98.31	102.66	121.49	125.40	145.26	150.08	169.31
8	72.83	78.54	98.64	102.66	122.09	126.27	146.41	149.51	169.31	173.37
9	77.77	98.64	102.36	122.09	126.27	146.41	149.20	170.08	173.01	193.49
10	78.78	100.40	120.59	124.23	146.21	150.17	170.08	173.01	193.49	198.22
11	98.64	120.59	124.24	127.23	150.17	168.68	172.05	175.77	198.22	218.41
12	103.08	124.24	144.31	146.51	168.67	173.62	192.38	196.03	216.99	222.23
13	120.59	144.31	146.51	150.43	173.62	192.38	196.07	198.22	220.39	240.39
14	124.24	146.50	148.93	171.17	175.76	196.07	198.84	218.93	240.39	243.94
15	144.31	148.93	169.13	174.02	195.18	198.84	218.93	223.45	243.94	264.89

Table C-70 Arena output - job finish times

	Finish Time									
Job	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**
1	3.75	7.65	27.85	30.55	49.86	54.83	75.75	96.12	97.45	102.23
2	24.26	26.59	30.13	50.16	51.16	72.58	79.88	97.41	101.76	122.04
3	28.08	48.66	52.97	72.15	74.62	79.59	100.26	120.82	122.51	126.20
4	29.99	52.33	54.14	75.92	77.39	99.27	120.21	124.45	126.69	145.10
5	50.52	72.83	74.99	78.11	96.13	100.90	123.73	126.42	144.50	149.24
6	54.17	74.36	77.88	98.31	102.08	121.49	125.40	145.26	150.08	151.11
7	55.77	78.54	96.81	102.66	103.78	123.02	127.86	149.51	169.31	173.37
8	77.77	98.64	102.36	122.09	126.27	146.41	149.20	151.90	170.40	192.87
9	78.78	100.40	120.33	124.23	146.21	147.46	170.08	173.01	193.49	197.43
10	98.40	120.59	122.86	127.23	150.17	168.68	172.05	175.77	198.22	218.41
11	103.08	124.24	125.69	144.69	168.67	171.59	175.18	196.03	216.99	222.23
12	120.38	144.31	146.51	150.43	173.62	192.38	194.18	198.18	220.39	240.16
13	122.04	146.50	148.67	171.17	175.76	196.07	197.88	217.89	240.39	242.51
14	127.63	148.93	151.85	174.02	195.18	198.84	218.93	223.45	243.94	264.89
15	147.55	169.13	170.23	193.20	197.63	217.00	221.56	243.17	244.95	268.95

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

C.15 Dataset E- 10 Stages, 15 Jobs, 15 Downtimes, Stage Capacity of 2

Table C-71 Input data - from Arena

		Stages									
Job	**Arrival**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.00	3.98	1.25	2.07	1.80	2.13	4.71	3.76	3.84	3.20	3.80
2	4.31	1.41	3.63	1.27	2.54	2.97	3.53	1.79	4.04	4.35	3.66
3	5.35	3.60	2.67	2.16	1.06	1.92	2.64	1.20	2.23	4.08	3.20
4	8.42	1.51	1.39	2.43	2.67	4.94	4.14	3.75	1.10	3.79	2.45
5	10.10	4.58	2.35	1.96	1.49	1.37	4.87	4.90	3.21	1.46	1.93
6	14.31	2.91	3.69	1.09	4.63	3.63	3.28	2.34	3.33	4.91	1.27
7	16.81	4.71	3.80	1.02	1.32	1.67	2.52	2.40	4.94	3.21	2.24
8	19.82	1.01	2.59	4.56	3.76	2.28	3.91	2.53	4.20	4.64	1.37
9	23.89	2.84	3.95	1.63	3.05	3.10	2.20	2.88	2.39	3.71	1.06
10	25.19	2.31	2.56	4.44	1.27	3.34	1.89	1.65	3.12	1.43	2.58
11	26.88	2.61	1.93	4.66	4.14	1.58	1.04	4.07	1.83	2.48	3.34
12	28.08	3.48	4.16	3.53	1.63	2.40	4.11	1.45	2.74	3.45	3.83
13	29.61	2.95	2.22	2.72	4.80	2.62	3.38	1.00	4.22	3.25	1.57
14	30.80	4.78	3.07	2.77	2.00	2.92	3.28	2.44	3.79	2.05	1.42
15	35.72	3.99	1.57	1.67	1.79	3.72	1.13	1.21	2.83	3.80	4.89

Table C-72 Downtime data for all the down occurrences

	Downtime Occurrences														
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
Start time	8	32	56	80	104	128	152	176	200	224	248	272	296	320	344
Duration	16	16	16	16	16	16	16	16	16	16	16	16	16	16	16

Table C-73 Arena output - job start times

	StartTime									
Job	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**
1	0.00	3.98	5.23	7.30	25.10	27.23	31.94	51.70	55.54	74.74
2	4.31	5.72	25.35	26.62	29.16	48.13	51.70	55.54	75.58	79.93
3	5.72	25.35	28.02	30.18	48.13	51.66	54.30	75.58	79.93	100.01
4	25.32	28.02	30.18	48.61	51.28	72.22	76.36	96.11	100.01	103.80
5	26.83	31.41	49.76	51.72	72.22	76.36	97.23	102.13	121.34	122.80
6	31.41	50.32	54.01	55.10	75.73	97.23	102.13	121.34	124.67	145.58
7	50.32	55.03	74.83	75.85	79.36	100.51	120.47	124.67	145.61	148.82
8	55.03	74.83	77.42	97.98	101.74	120.02	123.93	145.61	149.81	170.45
9	72.04	77.42	97.98	101.74	120.79	123.93	126.46	149.81	170.45	174.16
10	74.88	97.37	99.93	120.79	123.89	127.23	145.61	168.20	174.16	175.59
11	77.42	99.93	120.37	125.03	145.17	146.75	149.81	171.32	175.59	194.17
12	97.37	101.86	125.03	145.17	146.80	149.20	169.88	174.16	194.07	197.52
13	100.85	122.02	144.56	147.28	168.08	170.70	174.08	192.90	197.52	217.35
14	103.80	125.03	147.28	168.08	170.70	174.08	193.36	197.12	216.91	218.96
15	124.58	144.57	150.05	170.08	173.62	193.36	195.80	216.91	219.74	223.54

Table C-74 Arena output - job finish times

	Finish Time									
Job	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**
1	3.98	5.23	7.30	25.10	27.23	31.94	51.70	55.54	74.74	78.54
2	5.72	25.35	26.62	29.16	48.13	51.66	53.49	75.58	79.93	99.59
3	25.32	28.02	30.18	31.24	50.05	54.30	55.50	77.81	100.01	103.21
4	26.83	29.41	48.61	51.28	72.22	76.36	96.11	97.21	103.80	122.25
5	31.41	49.76	51.72	53.21	73.59	97.23	102.13	121.34	122.80	124.73
6	50.32	54.01	55.10	75.73	79.36	100.51	120.47	124.67	145.58	146.85
7	55.03	74.83	75.85	77.17	97.03	103.03	122.87	145.61	148.82	151.06
8	72.04	77.42	97.98	101.74	120.02	123.93	126.46	149.81	170.45	171.82
9	74.88	97.37	99.61	120.79	123.89	126.13	145.34	168.20	174.16	175.22
10	77.19	99.93	120.37	122.06	127.23	145.12	147.26	171.32	175.59	194.17
11	96.03	101.86	125.03	145.17	146.75	147.79	169.88	173.15	194.07	197.51
12	100.85	122.02	144.56	146.80	149.20	169.31	171.33	192.90	197.52	217.35
13	103.80	124.24	147.28	168.08	170.70	174.08	175.08	197.12	216.77	218.92
14	124.58	144.10	150.05	170.08	173.62	193.36	195.80	216.91	218.96	220.38
15	144.57	146.14	151.72	171.87	193.34	194.49	197.01	219.74	223.54	244.43

CPLEX model COM-W-Q output data is exactly matched with the Arena outputs in above. There is no discrepancy up to four digits after decimal points in the return data.

Appendix D: ILOG CPLEX Log Files

D.1 Log file for COM-W-Q with 0.5 process times

Tried aggregator 1 time.
MIP Presolve eliminated 887 rows and 296 columns.
MIP Presolve modified 186555 coefficients.
Aggregator did 141 substitutions.
Reduced MIP has 15853 rows, 6914 columns, and 35974 nonzeros.
Presolve time = 0.20 sec.
MIP emphasis: balance optimality and feasibility
Root relaxation solution time = 0.66 sec.

	Nodes Node	Left	Objective	IInf	Best Integer	Cuts/ Best Node	ItCnt	Gap	Variable	B	Parent	Depth
	0	0	5691.8478	179		5691.8478	3205					
			8896.3941	79		Cuts: 148	3921					
			9062.3185	90		Cuts: 268	4108					
*	0+	0	0		9122.6550	9062.3185	4108	0.66%				
			9085.3476	90	9122.6550	Cuts: 51	4153	0.41%				
*	0+	0	0		9106.6550	9085.3476	4153	0.23%				
			9091.8454	91	9106.6550	Cuts: 10	4206	0.16%				
			9091.8454	91	9106.6550	Impl Bds: 2	4210	0.16%				

Clique cuts applied: 32
Implied bound cuts applied: 53
Flow cuts applied: 4
Gomory fractional cuts applied: 26

D.2 Log file – COM-W-Q with 0.7 x process times

Tried aggregator 1 time.
MIP Presolve eliminated 1253 rows and 450 columns.
MIP Presolve modified 183755 coefficients.
Aggregator did 145 substitutions.
Reduced MIP has 15483 rows, 6756 columns, and 35130 nonzeros.
Presolve time = 0.20 sec.
MIP emphasis: balance optimality and feasibility
Root relaxation solution time = 0.66 sec.

	Nodes Node	Left	Objective	IInf	Best Integer	Cuts/ Best Node	ItCnt	Gap	Variable	B	Parent	Depth
	0	0	7125.2524	198		7125.2524	3145					
			12696.6031	81		Cuts: 224	4402					
			12881.4089	102		Cuts: 181	4613					
			12897.8243	98		Cuts: 16	4634					
			12904.1176	100		Impl Bds: 6	4650					
*	0+	0	0		13000.1510	12904.1176	4650	0.74%				
*	0+	0	0		12996.3590	12904.1176	4650	0.71%				
	100	51	12972.6287	23	12996.3590	12945.9674	5219	0.39%	x4414	U	99	28

200	127	12976.8599	19	12996.3590	12950.1146	5498	0.36%	x3969 U	199	30
300	177	12987.0006	40	12996.3590	12964.5945	5969	0.24%	x1249 U	299	19
400	203	12992.9371	16	12996.3590	12972.3857	6337	0.18%	x3085 U	399	31
500	240	12981.5946	5	12996.3590	12974.3167	6577	0.17%	x2822 U	499	42
600	288	12995.1750	11	12996.3590	12975.5372	6808	0.16%	x2354 U	599	33
700	323	12979.2372	8	12996.3590	12976.5722	7015	0.15%	x3759 U	699	39
800	351	12985.0462	2	12996.3590	12977.7103	7193	0.14%	x4385 U	799	45
900	385	12982.6860	3	12996.3590	12978.3682	7371	0.14%	x4012 U	899	44
1000	435	12989.9542	2	12996.3590	12978.8642	7507	0.13%	x4385 U	999	45

Elapsed time = 16.38 sec. (tree size = 1.85 MB)

1100	485	12984.1152	2	12996.3590	12979.1460	7631	0.13%	x4385 D	1098	45
1200	530	12979.3775	7	12996.3590	12979.3775	7770	0.13%	x4012 U	1093	40
1300	571	infeasible		12996.3590	12979.6846	7897	0.13%	x2822 D	374	42
1400	602	12981.9911	8	12996.3590	12980.0319	8027	0.13%	x4012 D	1399	39
1500	642	12986.7942	2	12996.3590	12980.2591	8149	0.12%	x4385 D	1498	45
1600	678	12984.0706	4	12996.3590	12980.3887	8286	0.12%	x2822 U	1599	43
1700	713	12987.7850	3	12996.3590	12980.5522	8429	0.12%	x3409 U	1699	44
1800	754	12982.3648	2	12996.3590	12980.6768	8581	0.12%	x4385 U	1799	45
1900	784	12984.8881	6	12996.3590	12980.8052	8722	0.12%	x3759 U	1899	41
2000	798	12982.4212	2	12996.3590	12980.9858	8850	0.12%	x4385 U	1999	45

Elapsed time = 28.14 sec. (tree size = 3.40 MB)

2100	824	12982.5191	5	12996.3590	12981.0877	8993	0.12%	x4324 U	2099	42
2200	859	12983.6421	6	12996.3590	12981.2230	9134	0.12%	x4178 U	2199	41

…

…

Elapsed time = 166.05 sec. (tree size = 2.84 MB)

13100	646	infeasible		12996.3590	12989.5383	21487	0.05%	x2822 D	13098	45
13200	614	12990.0552	2	12996.3590	12989.6487	21585	0.05%	x2822 U	13199	45
13300	595	12990.1187	3	12996.3590	12989.7397	21695	0.05%	x4385 D	10178	44
13400	577	12991.4553	3	12996.3590	12989.8362	21798	0.05%	x4385 U	13399	44
13500	561	infeasible		12996.3590	12989.9193	21900	0.05%	x2822 D	13498	45
13600	529	12990.3913	3	12996.3590	12990.0123	22005	0.05%	x4385 D	10640	44
13700	497	12990.5832	2	12996.3590	12990.1702	22099	0.05%	x2822 U	13699	45
13800	481	infeasible		12996.3590	12990.2602	22210	0.05%	x2822 D	13798	45
13900	459	12990.7283	3	12996.3590	12990.3493	22311	0.05%	x4385 D	11226	44
14000	438	12990.8642	2	12996.3590	12990.4512	22416	0.05%	x2822 U	13999	45

Elapsed time = 179.00 sec. (tree size = 1.89 MB)

14100	410	12991.6600	4	12996.3590	12990.6162	22513	0.04%	x4324 D	9422	43
14200	394	12991.0753	3	12996.3590	12990.6963	22610	0.04%	x4385 D	11728	44
14300	367	infeasible		12996.3590	12990.8733	22732	0.04%	x2822 D	14298	45
14400	338	12992.0360	4	12996.3590	12990.9922	22848	0.04%	x4324 D	11413	43
14500	322	infeasible		12996.3590	12991.1527	22948	0.04%	x2822 D	14498	45
14600	305	12991.9302	2	12996.3590	12991.3253	23075	0.04%	x2822 U	14599	45
14700	275	12992.1268	2	12996.3590	12991.5177	23184	0.04%	x2822 U	14699	45
14800	247	12992.1032	2	12996.3590	12991.7013	23274	0.04%	x2822 U	14799	45
14900	219	12992.3103	3	12996.3590	12991.9313	23377	0.03%	x4385 D	13249	44
15000	187	12992.5317	3	12996.3590	12992.1527	23475	0.03%	x4385 D	12531	44

Elapsed time = 191.55 sec. (tree size = 0.83 MB)

15100	158	12992.9032	2	12996.3590	12992.4953	23586	0.03%	x2822 U	15099	45
15200	130	infeasible		12996.3590	12992.8657	23698	0.03%	x2822 D	15198	45
15300	107	infeasible		12996.3590	12993.3007	23815	0.02%	x2822 D	15298	45
15400	77	12996.1409	5	12996.3590	12993.8842	23948	0.02%	x3409 U	15399	41

Clique cuts applied: 15

Implied bound cuts applied: 82

Flow cuts applied: 2

Gomory fractional cuts applied: 39

D.3 Log file for 20x50x30 COM-W-Q model

```
Presolve has eliminated 16901 rows and 7011 columns...
Presolve has improved bounds 387175 times...
Tried aggregator 2 times.
MIP Presolve eliminated 19816 rows and 7936 columns.
MIP Presolve modified 8953019 coefficients.
Aggregator did 825 substitutions.
Reduced MIP has 197271 rows, 85240 columns, and 449007 nonzeros.
Presolve time =    9.70 sec.
MIP emphasis: balance optimality and feasibility
Root relaxation solution time =   25.88 sec.

      Nodes                                 Cuts/
   Node  Left     Objective  IInf  Best Integer     Best Node    ItCnt     Gap         Variable B Parent  Depth

      0     0    88737.7307  1412                 88737.7307    34689
             190869.1628   656                Cuts:  1102    54758
             195517.6383   532                Cuts:  702    56646
             196100.8888   544                Cuts:  202    56928
             196791.0654   544                Cuts:  611    57267
             199165.7162   440                Cuts:  736    58374
             200174.0426   486                Cuts:  648    58939
             200190.0567   519                Cuts:  101    59009
             200261.0084   516                Cuts:  77    59077
             200292.9117   540                Cuts:  67    59133
             200392.0119   565                Cuts:  64    59214
             200392.7288   584                Cuts:  545    59258
Elapsed time = 235.03 sec. (tree size =  0.00 MB)
    100    86   200629.2439   395             200486.4245    60137               x47849 D     99     42
    200   167   200722.8396   450             200596.6250    61818               x31762 U    199     57
    300   229   200850.8956   437             200618.7501    85396               x51352 D     98     41
    400   275   323959.0036  1834             200674.8242   101142               x51005 U    399     29
    500   343   201824.9155   430             200691.5400   105231               x35228 U    499     73
    600   413   201236.6786   424             200706.5868   106811               x48563 U    598     48
    700   481   404002.8993  1471             200716.4366   124797               x50186 D    699     32
    800   539   201972.2036   330             200723.2390   136088               x55580 U    799     83
    900   591   201978.5783   311             200730.0648   137506               x52430 D    899     85
   1000   653   201971.8712   333             200741.7951   138749               x51553 D    999     84
Elapsed time = 873.72 sec. (tree size = 34.77 MB)
   1100   711   201864.4245   337             200749.9337   146548               x48906 D   1099     83
   1200   757   201892.4620   334             200780.9431   148185               x55580 U   1199     76
…
…
…
Elapsed time = 2437.86 sec. (tree size = 213.61 MB)
Nodefile size = 86.39 MB (29.12 MB after compression)
   7100  4065   203293.2422   415   631940.3060   201127.9046   367918   68.17%        x47740 D   7099
72
   7200  4129    infeasible         631940.3060   201128.9976   370751   68.17%        x49046 D   7199     86
   7300  4186   207135.9440   390   631940.3060   201129.6798   373708   68.17%        x24276 U   7299
104
   7400  4229   209652.0625   324   631940.3060   201129.6798   374592   68.17%         x2856 U   7399
203
```

7500 4247 211949.5468 342 631940.3060 201129.6798 375229 68.17% x33612 D 7499
303
7600 4317 202153.3218 467 631940.3060 201131.6834 377273 68.17% x26942 U 7599
75
7700 4372 204641.1733 330 631940.3060 201135.2327 382010 68.17% x46295 D 7698
82
7800 4429 201930.3474 397 631940.3060 201135.6693 383421 68.17% x31762 U 7799
61
7900 4483 infeasible 631940.3060 201140.2167 386290 68.17% x55693 D 934 62
8000 4538 201390.8125 355 631940.3060 201143.8390 389028 68.17% x52955 U 7999
69
Elapsed time = 2637.31 sec. (tree size = 241.99 MB)
Nodefile size = 114.24 MB (38.28 MB after compression)
8100 4594 204442.6721 370 631940.3060 201144.7496 393091 68.17% x52927 U 8099
80
8200 4644 203430.3004 398 631940.3060 201151.3690 396925 68.17% x54467 D 8198
66
8300 4701 206336.4870 431 631940.3060 201152.6460 401362 68.17% x17828 U 8299
99
8400 4775 infeasible 631940.3060 201156.0477 404493 68.17% x54438 D 8399 76
8500 4821 201377.1175 471 631940.3060 201158.2685 407422 68.17% x50175 D 8499
60
8600 4873 201588.9091 382 631940.3060 201160.2556 410475 68.17% x51440 D 8599
67
8700 4933 201287.5256 335 631940.3060 201160.8613 413815 68.17% x39178 U 8698
69
8800 4977 205050.2195 477 631940.3060 201163.2726 417988 68.17% x12150 U 8799
56
8900 5027 201501.1608 453 631940.3060 201168.8887 421176 68.17% x54438 D 8898
63
9000 5090 204760.6844 319 631940.3060 201169.2464 423755 68.17% x52400 U 8999
88
Elapsed time = 2859.06 sec. (tree size = 271.43 MB)
Nodefile size = 144.00 MB (48.07 MB after compression)
9100 5155 203120.6971 477 631940.3060 201171.7637 426136 68.17% x39260 D 9099
80

Appendix E: Support System Manual

For regular production users:

1. Open the Excel file Interface.xls.
2. Update/ Import production data (only change data ranges within green background)
3. Option step: Click "Clear Outputs" button to clear out the output sheet data
4. Click "Run Optimization !" button to see output data. The output sheet should be automatically selected at the end of run.
5. For any error message, please copy and report to system administrator for debug.

For system administer:

1. System setup requirements:
    a. MS office Excel program
    b. ILOG OPL Studio 3.7 IDE

2. Files required:
    a. Excel user interface – Interface.xls
    b. Complied job completion time prediction model OPL file - COMMOD.opl
    c. Complied overtime allocation model OPL file - OTMOD.opl

3. Error debug: if error OPL message about the log file occurs, try to comply the two original model files again to see if the problem can be solved.

4. Program setup:
    a. Make sure all three required files in the list 1 above are in a same directory (prefer local & network is OK)
    b. For Excel: make sure OPLserver type library 1.9 reference is toggled in VBA. (to confirm: Excel spreadsheet → tools → macros → Visual Basic Editor → Tools → References)
    c. Make sure OPL studio has valid access key

5. System setup Visual basic code check on the "OPLSolverMacro" script:
    a. Make sure the Excel file name in the VB " 'define data source" line 21 is the correct name.
    b. Make sure the completion time model is a complied file and named COMMOD.opl (hard-coded in VB. Can update the VB to the new name)
    c. Make sure the overtime allocation model is a complied file and named OTMOD.opl (hard-coded in VB. Can update the VB to the new name)
    d. Outputsheet sheet name "output5-10" is hard coded in VB as well. It can be updated in the VB to a new name.
    e. Sheet "Output5-10time" is an option sheet that doesn't involve with VB code. In order to show data correctly, update the data link formula in some of the data ranges are necessary.

f. Sheet “HideDummySheet” is used in identifying if overtime is needed. This sheet needs to update when change of system.

6. All name ranges must be exactly defined the corresponding data location in the Excel spreadsheets. Name ranges are coded in the OPL CPLEX files. Name ranges defined in the excel spreadsheets are:
   g. Input data ranges (input sheet):
      i. JobQTY – total quantity of jobs, value of *n*
      ii. StageQTY – total quantity of work stages, value of *m*
      iii. DownQTY – total number of down time occurrences, value of *J*
      iv. Arrival – data source for system arrival time data for A_i
      v. ProcessTime - data source for job process time d_{ki}
      vi. Duedate – data source for duedates $duedate_{ki}$
      vii. DownStart – data source for downtime start times $S(bar)_{kj}$
      viii. DownDuration - data source for downtime durations $d(bar)_{kj}$
      ix. Stage_Capacity – data source for stage capacity a_k
   h. Output data ranges (output sheet):
      i. OTSignal – define if the overtime model should run
      ii. Output_Starts – job start times S_{ki}
      iii. Output_Finishes - job finish times F_{ki}
      iv. Output_Departures - job departure times D_{ki}
      v. Output_OT- overtime by downtimes sum(OT_{kij}) for all *i*

7. New system setup:
   a. Modify and redefine all the name ranges (as listed in #5 in above) corresponding to new system in both input and output spreadsheets.
   b. Update all table titles and data cell names.
   c. Double check list in #4 in above.
   d.

Note for programmers:

This user interface is simply developed with this research to give a picture how it can work with OPL files. This interface does not automatically update input or output name ranges nor update the worksheets to reflect the changes. VB codes can be done for the automation but not in the design yet. The VB script code used in the interface is listed below.

```
Sub OPLsolverMacro()  ' Macro recorded 9/15/2006 by ofl

On Error GoTo theEnd
Application.ScreenUpdating = False
ActiveWorkbook.Save
'Application.ScreenUpdating = True
'range().clear

Dim i, j, k As Integer
```

```
'define the path
Dim path As String
path = ActiveWorkbook.path

'define data source
Dim doublequote
doublequote = String(1, 34)
Dim dataBuff As String
dataBuff = "location = " & doublequote & path & "\Interface.xls" & doublequote & ";"

'''''' Completion time model ''''''''''''''''''''

'Call OPL and solve
Dim opl As COPLsolver
Set opl = New COPLsolver
Call opl.loadCompiledModelFileAndDataBuff(path & "/COMMOD.opl", dataBuff, 1)

Dim result As Long
result = opl.solve
MsgBox ("Com-result=" & result)

If (result = 1) Then

'get results
Sheets("output5-10").Select
Dim jobs  As IOPLintRange
Set jobs = opl.getIntRange("jobs")
Dim stages As IOPLintRange
Set stages = opl.getIntRange("stages")
Dim downs As IOPLintRange
Set downs = opl.getIntRange("downs")
'(above ok )

' input ST
Dim ST As IOPLarray
Set ST = opl.getArray("ST")
Dim STi As IOPLarray
Dim STVar As IOPLfloatVar
For k = 1 To (Range("StageQTY").Value) 'ok
'MsgBox ("k=" & k)
Set STi = ST.eltArray(k)
    For i = 1 To (Range("JobQTY").Value) 'ok
    Set STVar = STi.eltFloatVar(i)
    'MsgBox ("i=" & i)
```

```
    Cells(k + 5, i + 2).Value = STVar.getValue() '10'
ST.getArray(k).getFloatVar(k).getValue()
    Next i
Next k

' input FT
Dim FT As IOPLarray
Set FT = opl.getArray("FT")
Dim FTi As IOPLarray
Dim FTVar As IOPLfloatVar
For k = 1 To (Range("StageQTY").Value) 'ok
'MsgBox ("k=" & k)
Set FTi = FT.eltArray(k)
    For i = 1 To (Range("JobQTY").Value) 'ok
    Set FTVar = FTi.eltFloatVar(i)
    'MsgBox ("i=" & i)
    Cells(k + 13, i + 2).Value = FTVar.getValue()
    Next i
Next k

' input DT
Dim DT As IOPLarray
Set DT = opl.getArray("DT")
Dim DTi As IOPLarray
Dim DTVar As IOPLfloatVar
For k = 1 To (Range("StageQTY").Value) 'ok
'MsgBox ("k=" & k)
Set DTi = DT.eltArray(k)
    For i = 1 To (Range("JobQTY").Value) 'ok
    Set DTVar = DTi.eltFloatVar(i)
    'MsgBox ("i=" & i)
    Cells(k + 21, i + 2).Value = DTVar.getValue()
    Next i
Next k

Else
      MsgBox "No feasible completion time solution returned, please check input data or
see system administrator for help.", vbCritical
End If

Call opl.Close

'''''' Overtime Optimization model ''''''''''''''''''''
'MsgBox ("OT is " & Range("OTSignal").Value)

If Range("OTSignal").Value = "NO" Then
```

```
MsgBox "Good News! No over time required!"
   Else 'OT required
'Call OTOPL and solve
Dim OTopl As COPLsolver
Set OTopl = New COPLsolver
Call OTopl.loadCompiledModelFileAndDataBuff(path & "/OTMOD.opl", dataBuff, 1)

Dim OTresult As Long
OTresult = OTopl.solve
MsgBox ("OTresult=" & OTresult)

If (OTresult = 1) Then

'get results
Sheets("output5-10").Select

Dim OTjobs  As IOPLintRange
Set OTjobs = OTopl.getIntRange("jobs")
Dim OTstages As IOPLintRange
Set OTstages = OTopl.getIntRange("stages")
Dim OTdowns As IOPLintRange
Set OTdowns = OTopl.getIntRange("downs")
'(above ok )

' input ST
Dim OTST As IOPLarray
Set OTST = OTopl.getArray("ST")
Dim OTSTi As IOPLarray
Dim OTSTVar As IOPLfloatVar
For k = 1 To (Range("StageQTY").Value) 'ok
'MsgBox ("k=" & k)
Set OTSTi = OTST.eltArray(k)
   For i = 1 To (Range("JobQTY").Value) 'ok
   Set OTSTVar = OTSTi.eltFloatVar(i)
   'MsgBox ("i=" & i)
   Cells(k + 5, i + 2).Value = OTSTVar.getValue() '10'
ST.getArray(k).getFloatVar(k).getValue()
   Next i
Next k

' input FT
Dim OTFT As IOPLarray
Set OTFT = OTopl.getArray("FT")
Dim OTFTi As IOPLarray
Dim OPFTVar As IOPLfloatVar
For k = 1 To (Range("StageQTY").Value) 'ok
```

```
'''MsgBox ("k=" & k)
Set OTFTi = OTFT.eltArray(k)
   For i = 1 To (Range("JobQTY").Value) 'ok
   Set OTFTVar = OTFTi.eltFloatVar(i)
   'MsgBox ("i=" & i)
   Cells(k + 13, i + 2).Value = OTFTVar.getValue()
   Next i
Next k

''' input DT
Dim OTDT As IOPLarray
Set OTDT = OTopl.getArray("DT")
Dim OTDTi As IOPLarray
Dim OTDTVar As IOPLfloatVar
For k = 1 To (Range("StageQTY").Value) 'ok
'''MsgBox ("k=" & k)
Set OTDTi = OTDT.eltArray(k)
   For i = 1 To (Range("JobQTY").Value) 'ok
   Set OTDTVar = OTDTi.eltFloatVar(i)
   'MsgBox ("i=" & i)
   Cells(k + 21, i + 2).Value = OTDTVar.getValue()
   Next i
Next k

' input OT
Dim OT As IOPLarray
Set OT = OTopl.getArray("DOT")
Dim OTj As IOPLarray
Dim OTVar As IOPLfloatVar
For k = 1 To (Range("StageQTY").Value)
'MsgBox ("k=" & k)
Set OTj = OT.eltArray(k)
  For j = 1 To (Range("DownQTY").Value)
 'MsgBox ("down= " & j)
  Set OTVar = OTj.eltFloatVar(j)
   'MsgBox ("j=" & j)
   Cells(k + 29, j + 2).Value = OTVar.getValue()
   Next j
Next k
Else 'OTresult =0
     MsgBox "No feasible for Overtime optimization returned, please check input data or
see system administrator for help.", vbCritical
End If

Call OTopl.Close
```

```
End If

Application.ScreenUpdating = True
Exit Sub

theEnd: MsgBox "No data found. Please check with system administrator.", vbCritical
Application.ScreenUpdating = True

End Sub
```

Appendix F: CD Attachment