Monitoring Air Cargo Shipments:

A Framework for

Detecting Potential Delays and Prescribing Corrective Measures

WAN, Yulai

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of

Master of Philosophy

in

Decision Sciences and Managerial Economics

June 2007

Thesis/Assessment Committee

Professor Linda Fung-yee Ng (Chair)

Professor Lawrence C Leung (Thesis Supervisor)

Professor Waiman Cheung (Committee Member)

Professor Yer Van Hui (External Examiner)

Abstract

This research studies the shipment planning problem confronted by air freight forwarders. The objective is to prevent delays caused by the randomness in shipping activity duration. The emergence of RFID enables real time monitoring of the shipping process. With the assistance of RFID, it is possible for forwarders to detect potential delays and conduct corrective action before delays actually take place. The determination of whether delay will happen is difficult, since many factors influence the chance of delay. Because controlling of the shipping process is similar to managing multiple projects at the same time, this feature increases the complexity of the problem and adds difficulty to the selection of corrective measures. This paper proposes a framework which consists of three phases: detecting potential delay, prescribing corrective measures, and validating corrective measures. This framework defines a set of delay indicators for detecting potential delays and a group of criteria for selecting corrective measures. The decision on tolerance level and the validation of corrective measures are based on simulation experiments.

摘 要

本論文對航空貨運代理面臨的裝運規劃問題進行了研究。研究目的在於預防由各項運輸任務所用時間的隨機性導致的貨物交付延誤。在物流領域新近出現並逐漸投入應用的射頻識別（RFID）技術能夠對運輸過程進行即時監控。通過RFID技術的協助，貨運代理能夠在延誤發生之前探測到潛在的延誤，並且採取補救措施。出現延誤的可能性受到許多因素的影響，因此，對延誤到底是否會發生做出正確的判斷是本研究所要攻克的難點之一。由於貨運過程的監控類似於對多個項目同時進行管理，這就增加了問題的複雜性和決定補救措施的困難性。本研究就此提出了一個包括探測潛在延誤、制定補救措施、以及驗證補救措施有效性的三階段監控框架。此框架定義了一組延誤指數來探測潛在延誤，同時，制定了一組選擇補救措施的標準。並且通過建立仿真模型，來確定延誤指數的容忍度，以及對補救措施的有效性進行檢測。

Acknowledgement

I would like to take this opportunity to express my gratitude to many individuals who have provided lots of supports during my two-year MPhil program. Without their advice and help, this thesis would not have been completed.

First of all, I would like to thank my thesis supervisor, Prof. Lawrence C. Leung, who introduces me to the real academic life and leads me through my first research. It is his encouragement and patient guidance that help me overcome all the obstacles encountered during the past two years. Without his valuable comments and advice, this thesis would not have been improved so much.

I would also like to appreciate the efforts from Prof. Waiman Cheung, Prof. Linda Ng, and my external examiner, Prof. Yer Van Hui, who kindly spare their valuable time reviewing this thesis and providing precious comments.

I am grateful to my fellow colleagues and friends in the Chinese University of Hong Kong who have shared their own experience with me both in research and in daily life. I have been inspired by all the interesting and informative discussions with them.

Last but not least, I would like to give my hearty thanks to my family, especially to my parents. Their full support is my source of strength, any time and any where.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In air cargo shipping, delivering a single shipment requires performing a sequence of activities, such as trucking, warehousing, consolidating, and air transporting. Figure 1 gives an example of a shipping process for a single shipment, where six activities need to be performed to complete the shipping process. The shipping process involves effort from many parties and coordination is important.

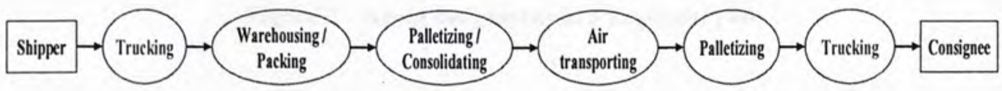


Figure 1 A shipping process for a single shipment

Air freight forwarders are third party logistics providers, responsible for managing the shipping process. Typically, forwarders design shipment plans to coordinate different agents during the shipping process. They must make sure that the target delivery date is met at the lowest cost.

When planning for a portfolio of shipments, the issues of freight consolidation and integration have to be addressed. Figure 2 shows a shipment plan for four shipments, four activities and four available agents. Consolidation happens when an agent is designated to perform the same shipping activity for two or more shipments. Consolidation results in cost savings by achieving higher resource utilization (Bookbinder and Higginson 2002) but increases the delivery time (Pooley and Stenger 1992). Integration happens when consecutive activities of a shipment are assigned to the same agent. Since a single agent is involved for a number of consecutive activities, the setup time and setup cost are reduced.

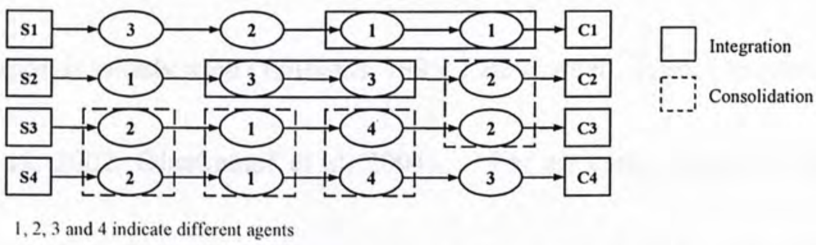


Figure 2 Agent assignment in a shipment plan

Deterministic models have been developed to design shipment plans. Leung, Hui and Wang (2003) minimize the total cost by considering the effects of consolidation and integration with a 0-1 LP model. Wong, Leung and Hui (2005) build a mixed 0-1 LP model to incorporate operational constraints, such as target delivery time, agent capacity and shipping budget.

A 2003 survey claims that reliability (defined as the probability of delivering the shipment on time), is the most important performance measurement for an air freight forwarder, while delivery time and cost are second and third respectively (Air cargo world, 2003). Time and cost in shipment planning have been addressed in the past using deterministic shipment planning models, but study on reliability has received little attention.

Reliability is influenced by uncertainties encountered in the shipping process. Shafer (1976) defines aleatory uncertainty (natural or unpredictable variation inherent to the system, also refers to randomness) and epistemic uncertainty (uncertainty due to incomplete knowledge of the behavior of the system). This classification is widely used (Williams, 1995; Pate-Cornell, 1996; Casman et al, 1999; Hofer et al, 2002; Oberkampf et al, 2004). For air cargo shipping, the aleatory uncertainties include randomness in activity duration and available shipping capacity. The epistemic uncertainties cover change in shipping destination, cancellation of flight due to extreme situations, such as bad weather, accidents, etc.

Deterministic shipment planning models do not incorporate uncertainties. Figure 3 shows an example of a deterministic shipment plan: the assigned agent and corresponding planned duration for each activity are shown in brackets. Under a deterministic assumption, both shipments can be delivered before target delivery times. However, after incorporating randomness in activity duration, the reliability

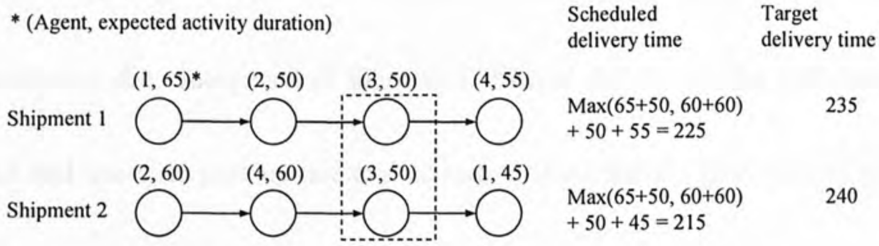


Figure 3 Agent assignment for the illustrative example

of Shipment 1 is around 84.2% (obtained by simulating the shipping process for 1000 times under random activity durations). To illustrate how delay happens, we note that Shipment 2 spends 80 units of time on its first activity. Then, even if durations of all the other activities are the same as planned, Shipment 1 will still be delayed, since its earliest delivery time becomes 245.

Is it possible to avoid such delay? Can delay be detected early? Can some corrective actions be taken when delay is detected? For the above example, if the forwarder is able to check the time spent on the first activity of Shipment 2, it will find that time spent on the first activity of Shipment 2 is too long and Shipment 1 has to wait for Shipment 2 for an extra amount of time which causes delay of Shipment 1. Then, the forwarder may give up the consolidation in activity 3, so that both Shipment 1 and Shipment 2 can still be delivered on time. The forwarder may also benefit from in-process shipment monitoring by having higher flexibility since corrective actions are employed if shipment delay is predicted.

Technologically, it is possible to achieve real time monitoring of the shipping process due to the emergence of internet, GPS and RFID. RFID facilitates more frequent and accurate data update and access so that timely adjustment of shipment plans at operational level is possible.

In this paper, we propose a framework to detect potential delays during the execution of a deterministic shipment plan and provide guidelines on corrective measures. The objective is to make sure that the reliability of each shipment meets the target level. The shipment plan is obtained by existing deterministic shipment planning models. Then, by simulation experiment, we test the reliability of this shipment plan with uncertain activity durations and investigate the shipment monitoring process. A set of indicators are defined and monitored to detect whether the shipment is able to achieve the target reliability level. The framework also identifies certain criteria which guide the selection of corrective measures or formation of rules.

Chapter 2 Literature Review

This research problem covers topics in project management with uncertainty and reactive job-shop scheduling.

Structurally, monitoring the shipping process under uncertainty shares some similar features with project management (Figure 4). The shipping process of an individual shipment can be regarded as a project in which all the tasks must be

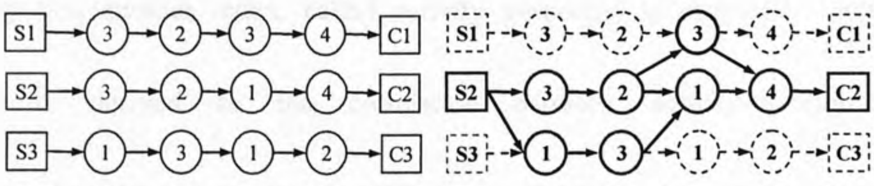


Figure 4 Example of regarding shipment 2 as a project

completed before a specified date. However, managing a shipping process is much more complicated than managing a project. Since a shipment plan includes a portfolio of shipments, the forwarder manages a set of projects at the same time. These projects have different starting times and target delivery times and, due to consolidation, they share many activities. Therefore, an activity of no importance to one shipment may cause serious delays for others.

A major goal in project management is to identify critical activities by introducing concepts of critical path, activity criticality index and activity cruciality index. Van Slyke (1963) defines activity criticality as the probability that the activity is on the critical path and uses Monte Carlo simulation to calculate this index. This index is widely accepted in the project management literature. Later researches on the activity criticality index try to approximate the index analytically (Dodin and Elmaghraby, 1985), or improve the accuracy of criticality estimation by conditional sampling approach (Bowman, 1995). Williams (1992) demonstrates that activity criticality index may give counterintuitive information about the risk in an activity and another index, called activity cruciality, is proposed. Activity cruciality is defined as the correlation between activity-duration and total-project-duration. According to Williams (1992), both criticality and cruciality of an activity should be measured, since the former one identifies activities whose expected duration may be reduced in order to reduce the duration of the project and the later one reflects the impact of uncertainty on the project duration.

Although these concepts are also helpful in monitoring a shipping process, they do not predict whether delay will happen in the future. Bowman (2006) develops a method to ensure on-time completion of a project by having control limit and actions for each activity. However, Bowman's method does not consider the structural feature of a project, but sets control limits arbitrarily based on expected durations of individual activities.

The concept of monitoring and controlling a shipping process is also similar to the predictive-reactive scheduling approach in job-shop scheduling, even though the structure of a job-shop scheduling problem is quite different from a shipment planning problem. The predictive-reactive scheduling approach is commonly used to generate, monitor and control a scheduling system with uncertainty in job-shop scheduling.

In predictive-reactive scheduling, an initial schedule of the entire shop floor is generated by a priori. Then, during the execution of the schedule, this existing schedule is revised based on updated information on the shop floor. Aytug et al (2005) provide a thorough review on this approach. Church and Uzsoy (1992) claim that remedial actions can be taken any time an event (e.g. arrival of a new job) happens (continuous approach), between predefined time intervals (periodic approach), or after the occurrence of events which are potential causes of disruption (event-driven approach). Optimum seeking algorithm and simple heuristic rules are commonly used in the reactive scheduling part (Sabuncuoglu and Bayiz, 2000). Many researches (e.g. Lawrence and Sewell, 1997; Wan, 1995) demonstrate that global scheduling algorithm may not outperform or even perform poorer than dispatching rules when the processing time is random, especially when the uncertainty level is high. However, even this predictive-reactive approach does not include look-ahead detection of potential delays. Since job-shop scheduling does not allow consolidation of jobs, special heuristic rules or reactions are needed for shipment monitoring problems.

Chapter 3 Framework

To begin, a shipment plan is obtained under deterministic environment by some deterministic shipment planning model (e.g. Wong, Leung and Hui, 2005). This shipment plan is input to a simulation model. This simulation model mimics the shipping process and facilitates the implementation of the framework. It is assumed that randomness in activity duration is the only uncertainty considered and that sufficient amount of resource/capacity is always available (Section 4 gives the details of the simulation model). The framework of monitoring and controlling this shipment plan includes three phases.

The first phase is to detect potential delays by checking a predefined delay indicator each time an activity is completed. Since the shipping process is highly dynamic, the forwarder needs to track an indicator representing the chance of delay throughout the shipping process. Then, the forwarder has to set a tolerance level by simulation as the cutoff point to decide whether delay will occur or not. Potential delay is considered to be detected if the delay indicator drops below the tolerance level.

Once potential delay is detected, the second phase is invoked to decide whether corrective measures will be taken and which measures will be executed. A group of criteria are defined to facilitate these decisions. Forwarders may form rules which specify which measures should be taken based on the criteria.

Decisions on the tolerance level and corrective measure are difficult to make because whether the shipment will have delay depends on the combined effect of the tolerance level and the corrective measure. Increasing the tolerance level increases the amount of correction, and hence the possibility of achieving high reliability. However, if an inappropriate corrective measure is taken, increasing the tolerance level may not be helpful.

The third phase tests whether the measures selected are effective through the simulation experiments. If the selected measure does not improve the reliability, the validation algorithm will choose and evaluate another one. If none of the measures can achieve the target reliability, the whole shipment plan is considered failed. A new shipment plan should be generated through other deterministic methods, which is not the concern of this paper.

Figure 5 shows the three-phase framework of monitoring and controlling the shipping process.

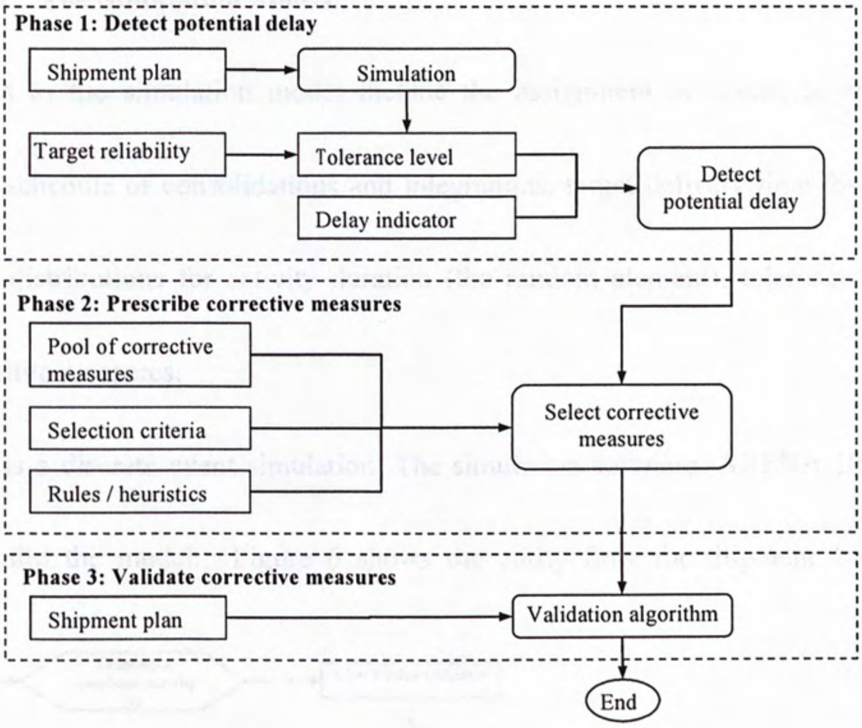


Figure 5 Framework of in-process monitoring and controlling the shipping process

Chapter 4 The Simulation Model

Inputs to the simulation model include the assignment of agents to all the activities, schedule of consolidations and integrations, target delivery time for each shipment, distributions for activity duration (the random element), tolerance level and corrective measures.

This is a discrete event simulation. The simulation software, ARENA 10.0, is used to build the model. Figure 6 shows the entity flow for shipment *k* when

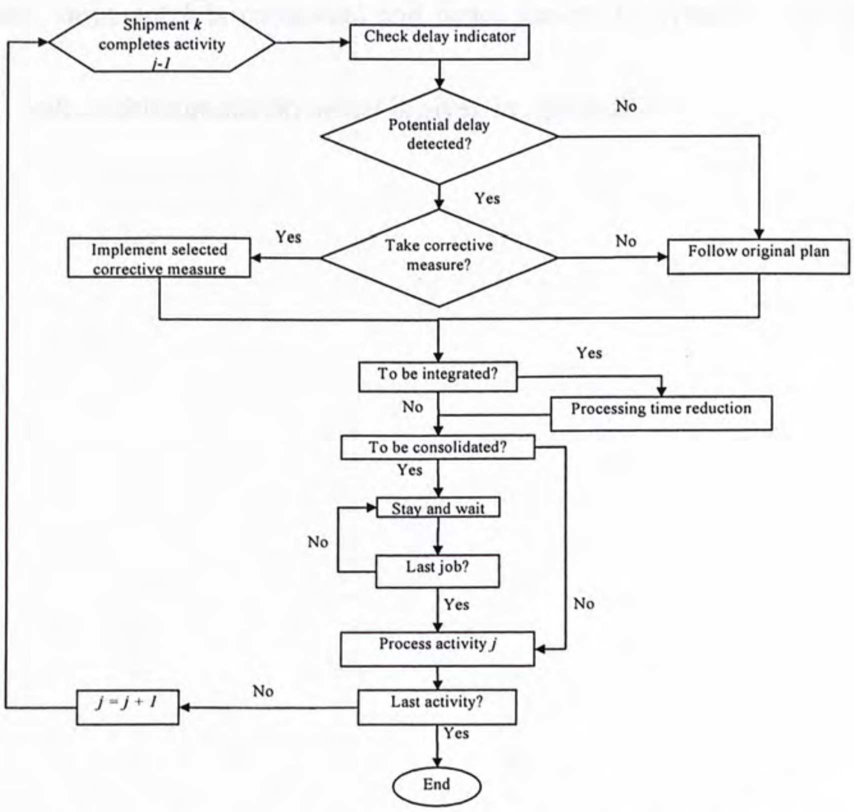


Figure 6 The entity flow chart for a single shipment *k* at activity *j*

activity *j-1* is just completed and activity *j* is going to be performed soon. Firstly, the delay indicator of shipment *k* is checked. If potential delay is detected, the corrective measure is selected and taken. If no correction is needed, shipment *k* follows the original plan. Then, integration and consolidation are identified. If integration is scheduled, the processing time of this activity is reduced by a predetermined amount. If consolidation is scheduled, shipment *k* must stay at the current consolidation site till all other shipments to be consolidated arrive. If shipment *k* is the last arrival shipment, no waiting is needed. All the consolidated shipments will be processed in a batch by the same agent. After the last activity is completed, shipment *k* is completed and hence leaves the system. The computer program code of this simulation model is given in Appendix.

Chapter 5 Phase 1: Detect Potential Delay

Detection of potential delay is based on tracking the state of all the shipments throughout the shipping process. Whether delay will occur depends on the cumulated effects of the randomness in a number of activities. It is possible that a smaller increase in the duration of each activity is significant enough to cause serious delay. It is also possible that some activity take longer time while others take shorter time than expected, resulting in no delay. Therefore, to detect delays, the forwarder must check the total amount of time used by the shipments each time an activity is completed. Indicators are needed to catch the changing state of shipments and predict the possibility of delay.

5.1 The delay indicator

The delay indicator is the key to drive the whole detecting and correcting process. It is tracked and updated throughout the shipping process. Each time one activity is completed, indicators for different shipments are checked for potential delays.

For shipments with consolidations, it is important to consider the position of other shipments when estimating the possibility of delay for a certain shipment. Considering the example in Section 1, it is Shipment 2 rather than Shipment 1 that causes the delay of Shipment 1. Thus, at the end of activity 2, if the forwarder does not consider the position of Shipment 2, it will not detect the delay of Shipment 1 until activity 3 is completed. In this case, it may be too late to take corrective measures. To detect delays caused by other shipments, the forwarder should give a delay indicator to each pair of shipments which shows whether the current status of shipment *k* will cause delay of shipment *m*. Therefore, when we check a particular shipment for potential delay, we should look at a set of delay indicators showing the influence of all the related shipments.

In designing the delay indicator, we try to combine all the factors which will influence the reliability, so that it will be much easier to make decision. After several simulation experiments, we find that the reliability is based on time already spent, target delivery time of the shipment, planned duration for remaining uncompleted activities, and randomness of remaining activities. The difference between target delivery time and time already spent represents the available time for remaining activities. The difference between available time and planned duration for remaining activities indicates the time slack to the target delivery time. We find that, at the end of each activity, the reliability has increasing monotonicity in the time slack (Figure 7). However, at the end of different activities, the same time slack predicts different level of reliability, because the amounts of randomness at the end of different activities are different. Therefore, we divide the time slack with the standard deviation of remaining activities to standardize the effects of different levels of randomness. Figure 8 shows that after standardization, indicators at the end of different activities give similar prediction on reliability.

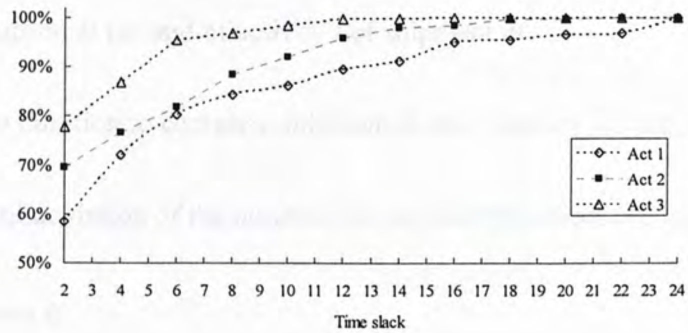


Figure 7 Reliability vs. time slack at the end of different activities

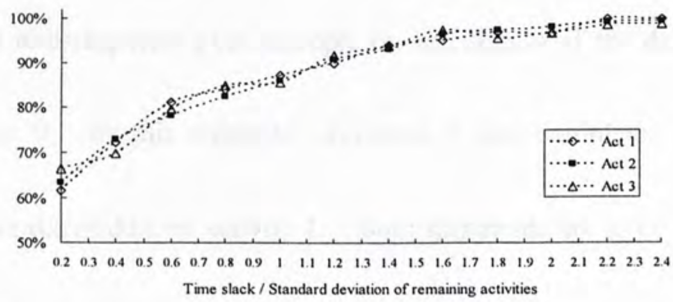


Figure 8 Reliability vs. Time slack/Standard deviation for remaining activities

Based on the above rationale, at the end of activity l of shipment m, the delay indicator NS_{jklm} predicting the possibility of delivering shipment m on time when shipment k is at the beginning or in the middle of activity j is shown below.

Let $NS_{jklm} = \dfrac{D_m - X_{lm} - R_{jkm}}{\sigma_{R_{jkm}}}$ $\quad \forall k \in \pi_m$ and $k = m$, where

j = The activity which shipment k just completes or is about to complete

l = The activity which shipment m just completes or is about to complete

NS_{jklm} =The delay indicator checking the possibility of on-time for shipment m at activity l under the influence of shipment k which is at activity j

D_m =Target delivery time of shipment m

X_{lm} =Time elapsed at the end of activity l of shipment m

R_{jkm} =Planned duration to complete shipment m after activity j of shipment k

$\sigma_{R_{jkm}}$ =Standard deviation of the duration to complete shipment m from activity j of shipment k

π_m =The set of shipments which influence the delivery time of shipment m

A detailed two-shipment plan example on calculation of the delay indicator is given in Figure 9. In this example, shipment k just completes activity 2 and shipment m is in the middle of activity 1. Both shipments are to be consolidated at the third activity. T_{jk} represents the planned duration for activity j shipment k.

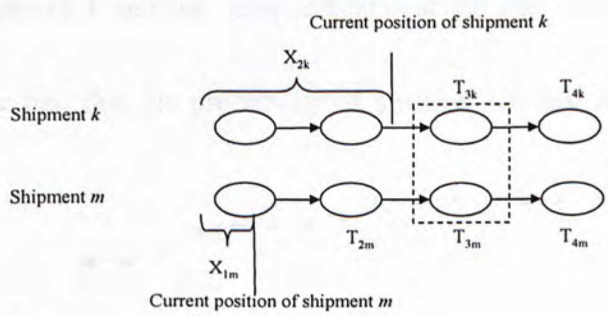


Figure 9 An example of a two-shipment plan for delay indicator calculation

The delay indicator of shipment k to the target delivery time of shipment k:

$$NS_{2k2k} = \frac{D_k - X_{2k} - R_{2kk}}{\sigma_{R_{2kk}}}, \text{ where}$$

$$R_{2kk} = \max(T_{3m}, T_{3k}) + T_{4k} \text{ and } \sigma^2_{R_{2kk}} = \max(\sigma^2_{T_{3m}}, \sigma^2_{T_{3k}}) + \sigma^2_{T_{4k}}$$

The delay indicator of shipment *m* to the target delivery time of shipment *k*:

$$NS_{1m2k} = \frac{D_k - X_{2k} - R_{1mk}}{\sigma_{R_{1mk}}}, \text{ where}$$

$$R_{1mk} = T_{2m} + \max(T_{3m}, T_{3k}) + T_{4k} \text{ and } \sigma^2_{R_{1mk}} = \sigma^2_{T_{1m}} + \sigma^2_{T_{2m}} + \max(\sigma^2_{T_{3m}}, \sigma^2_{T_{3k}}) + \sigma^2_{T_{4k}}$$

The delay indicator of shipment *k* to the target delivery time of shipment *m*:

$$NS_{2k1m} = \frac{D_m - X_{1m} - R_{2km}}{\sigma_{R_{2km}}}, \text{ where}$$

$$R_{2km} = \max(T_{3m}, T_{3k}) + T_{4m} \text{ and } \sigma^2_{R_{2km}} = \max(\sigma^2_{T_{3m}}, \sigma^2_{T_{3k}}) + \sigma^2_{T_{4m}}$$

The delay indicator of shipment *m* to the target delivery time of shipment *m*:

$$NS_{2k1m} = \frac{D_m - X_{1m} - R_{1km}}{\sigma_{R_{1km}}}, \text{ where}$$

$$R_{1km} = T_{2m} + \max(T_{3m}, T_{3k}) + T_{4m} \text{ and } \sigma^2_{R_{1km}} = \sigma^2_{T_{1m}} + \sigma^2_{T_{2m}} + \max(\sigma^2_{T_{3m}}, \sigma^2_{T_{3k}}) + \sigma^2_{T_{4m}}$$

Using the example in Section 1, Figure 10 shows the relationship between the reliability of Shipment 1 and the delay indicator at the end of the first activity of Shipment 1. We find that the probability of on-time delivery increases when the

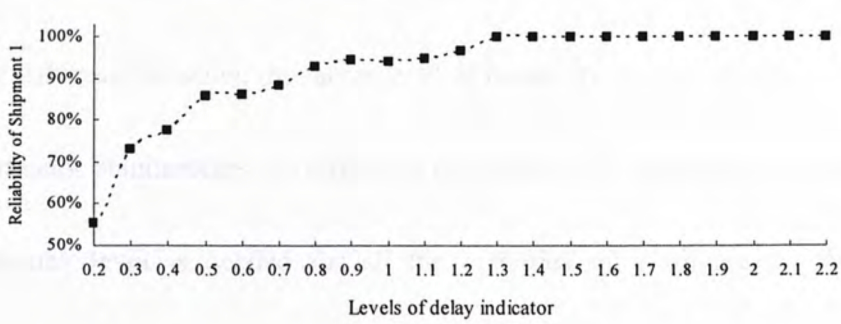


Figure 10 Reliability of Shipment 1 vs. different levels of delay indicators

value of delay indicator increases. Increase in reliability becomes slower with increase in the delay indicator. This pattern exists in all the examples we have tried, suggesting that the value of delay indicator can be used to predict the reliability of a shipment plan.

Moreover, if Shipment 2 spends 80 units of time on its first activity and $\sigma_{R_{121}}$ = 7.91, at the end of this activity NS_{1221} = (235-80-60-50-55)/7.91 = -1.26. Then, the forwarder should be aware that Shipment 1 will be delayed if no corrective action is taken.

5.2 Setting tolerance level

The potential delay of shipment m is identified if any NS_{jklm} is smaller than the tolerance level (TL). Therefore, it is important to set an appropriate tolerance level. One way is to construct the graph showing the relationship between reliability and delay indicator, such as Figure 10. After the target reliability is determined by the forwarder, the tolerance level can be set as the lowest delay indicator value under which the target level of reliability can be reached. Since the delay indicator standardizes the effects of randomness for remaining activities, only one tolerance level is needed for all the activities of a shipment. While, for activities of the other shipments which may also cause delay of this shipment, a different tolerance level may be set.

The relationship between reliability and delay indicator can be obtained by simulation. First, we observe a value of the delay indicator, $NS_{jklm}(i)$. Second, we simulate the shipping process n times with this level of delay indicator and record the probability of on-time delivery p_i. After repeating this process for N times, we obtain N pairs of delay indicators and probabilities. Then, we can draw the mapping of delay indicator and reliability based on these simulation outcomes.

Chapter 6 Phase 2: Prescribe Corrective Measures

When the delay indicator shows that potential delay is expected, the forwarder has to decide whether and which corrective measures should be taken. Whether the selected measure will make the shipment plan achieve its target reliability depends on the tolerance level and the features of the plan itself. Thus, it is difficult to find out the best measure for all situations, but it is possible to create some rules based on some criteria introduced here.

6.1 Corrective measures

The corrective measure aims to increase the value of the delay indicator to or above the tolerance level. This can be achieved by shortening the expected activity duration or reducing the standard deviation of subsequent activities. If a shipment has to wait for other shipments to be consolidated at the next activity, giving up the consolidation can also increase the delay indicator. Therefore, three corrective measures are proposed when NS_{jklm} is below the tolerance level.

Measure 1 Shorten the expected duration or reassign a faster agent for the next activity of the shipment (i.e. activity $j+1$ of shipment k)

Measure 2 Reassign the next activity of the shipment (i.e. activity $j+1$ of shipment k) to an agent with a lower level of randomness

Measure 3 Give up the consolidation for some or all of the shipments to be consolidated for the next activity (i.e. activity $j+1$ of shipment k) and reassign agents to these activities

Again, we use the example in Section 1 to illustrate these three measures. Figure 11 shows that Shipment 2 just completes its first activity and 80 units of time

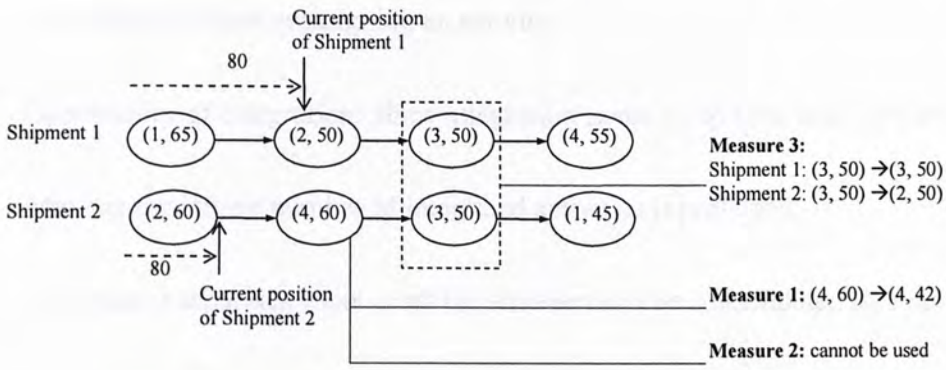


Figure 11 Use three measures for the example in Section 1

have been spent, while Shipment 1 is being processed for its second activity. Assume that the tolerance level is 1. If measure 1 is used, the expected duration of the second activity of Shipment 2 must be shortened so that NS_{1221} can be increased to at least 1. By simple calculation, we find that the expected duration for this activity should be shortened to 42 units of time. Since the delay indicator, NS_{1221}, is negative, it is impossible to increase it over the tolerance level by using measure 2 only. If measure 3 is taken, the consolidation will be given up by reassigning the third activity of Shipment 2 to agent 2 whose expected duration is also 50.

There may be a number of alternative agents who are able to increase the delay indicator to the desired level. Therefore, several criteria may be considered when selecting a new agent.

- Cost: select the agent which incurs least cost increase.
- Opportunity of consolidation: to limit the risk of delay, avoid creating consolidation when reassigning an activity.
- Opportunity of integration: since integration saves setup time and cost, the agent who has the largest number of integrated activities is preferred.

Measure 3 separates some or all the shipments to be consolidated by reassigning those shipments different agents. When three or more shipments are included in the consolidation, the forwarder has to decide which shipments should be kept consolidated and which ones should be processed separately (or form a new consolidation with a new agent). These decisions are situation specific and may be based on some rules.

6.2 Criteria of selecting measures

Taking corrective measures each time the delay indicator drops below the tolerance level may not be reasonable. At the early stage of the shipping process, a considerable amount of uncertainty still exits since many activities have yet to be performed. It is possible that even though a correction is made, potential delay will still be detected later. Taking corrective measures early could result in unnecessarily high cost. Taking corrective measures later may be more effective, since the forwarder faces less uncertainty. However, the forwarder is exposed to the risk of not being able to improve the reliability to the desired level. Thus, we propose that the decision of whether to take a corrective measure should be based on three criteria: remainder ratio, activity cruciality and improvement potential.

(1) Remainder ratio: "Remainder ratio" shows how late the activity is performed. Remainder ratio for activity *j* of shipment *k* is defined as the number of activities not yet performed since activity *j* over total number of activities required by shipment *k*. That is, AL_{jk} can be expressed by

$$\frac{Total\ Number\ of\ Activities\ for\ Shipment\ k-(j-1)}{Total\ Number\ of\ Activities\ for\ Shipment\ k}.$$

Later activities have relatively less uncertainty to be encountered. Thus, information obtained at later activities gives better prediction on the possibility of delay. Corrective measures carried out at later activities may be more effective than those taken at earlier activities.

(2) Activity cruciality: In project management, Williams (1992) defines activity cruciality as the correlation between activity-duration and total-project-duration. It reflects the impact of uncertainty on the project duration. Similarly, we define that activity cruciality measures the impact of uncertainty of a single activity on the delivery time of an individual shipment. Activity cruciality is the correlation of time spent on activity *j* of shipment *m* to the total delivery time of shipment *k*. That is, AU_{jmk} is defined as $Corr(t_{jm}, T_k)$, where T_k is the duration of delivering shipment *k* and t_{jm} is the duration of activity *j* of shipment *m*.

Crucial activities have a large impact on uncertainty in total processing time. Before the crucial activities are completed, corrective actions may have little overall effect, because time spent on crucial activities is likely to be lengthened by a large amount and overrides the correction efforts taken before.

(3) Improvement potential: "Improvement potential" is defined as the largest amount of correction allowed for all activities not yet performed. If the forwarder takes corrective measures only at the later stage of a shipping process, only a small number of activities can be revised. Then, the forwarder encounters the risk that the allowable amount of correction is not large enough to increase the delay indicator to the desired level. Therefore, it is necessary to check whether there is a large room for improvement in later activities. If not, the forwarder should take corrective measures early.

We use an example of a single-shipment plan (Figure 12) to show the importance of considering the above indicators when deciding whether corrective measures will be taken. Four actions are compared: (1) Shorten expected duration for the last activity; (2) Shorten expected duration after the most crucial activity; (3)

Shorten activity duration before the most crucial activity; (4) Shorten activity duration at the end of any activities when potential delay is detected.

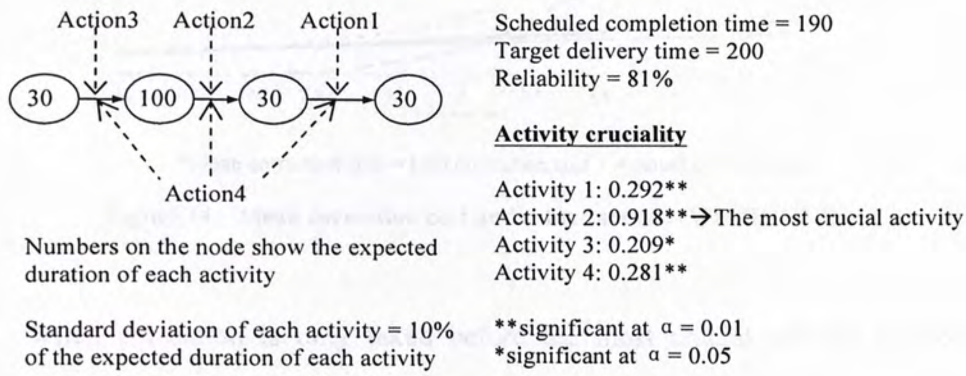


Figure 12 An example of a single-shipment plan

When there is no limit on the amount of duration to be shortened (Figure 13 and Figure 14), revising only the last activity (Action 1) achieves similar reliability levels to revising all the activities (Action 4). However, Action 1 incurs lower mean correction cost than Action 4, suggesting that taking corrective measures on activities closer to the end of the shipping process is more efficient.

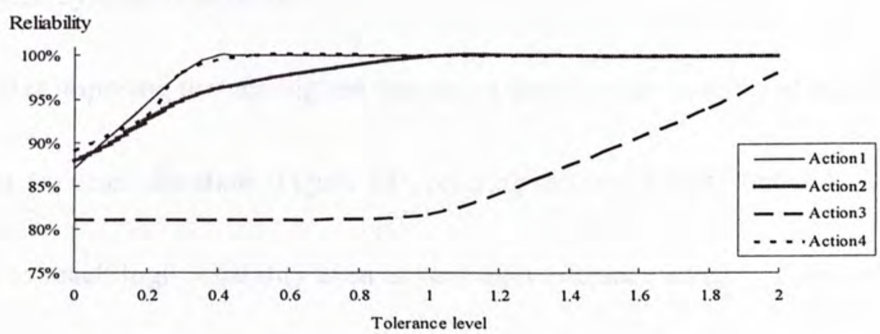
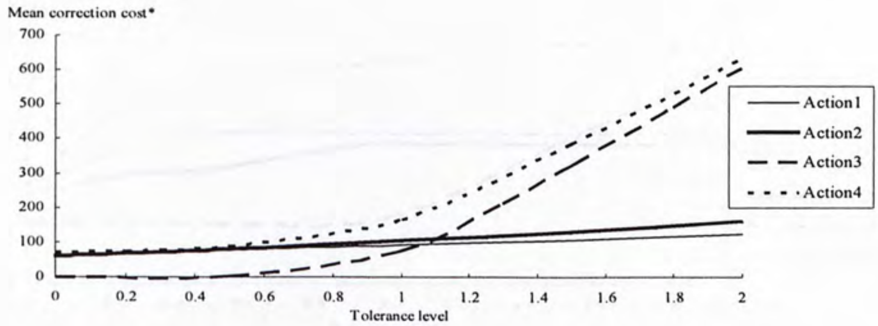


Figure 13 Reliability under four actions (WITHOUT limit)



*Mean correction cost = Unit correction cost × Amount of correction

Figure 14 Mean correction cost under four actions (WITHOUT limit)

When correction is only taken before the most crucial activity (Action 3), increase in reliability becomes faster when the tolerance level increases. Since prediction of delay is less accurate before crucial activities are performed, high tolerance level has to be set so as to achieve high reliability. Therefore, at the same reliability level, Action 3 incurs the highest mean correction cost. Moreover, only revising the activity after the most crucial activity (Action 2) also performs better than Action 3, which shows that the effectiveness of corrective measures is highly influenced by crucial activities.

After imposing that the highest amount of duration can be reduced is 20% of the original expected duration (Figure 15), revising the last activity only (Action 1) is unable to reach high reliability even at very high tolerance levels. This means that when the improvement potential decreases, corrective measures should be taken earlier.

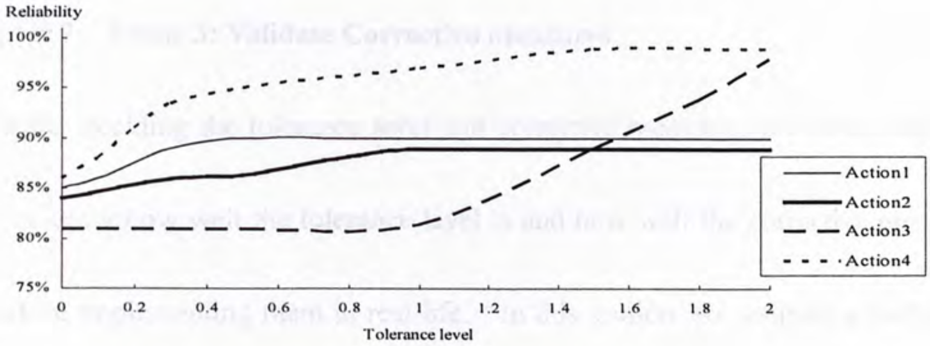


Figure 15 Reliability under four actions (WITH 20% limit)

If the forwarder decides to take corrective measures, decision on which measure to take relies on two criteria: consolidation at the next activity and activity criticality.

Consolidation at the next activity refers to whether the shipment is to be consolidated at the next activity. If the shipment is to be consolidated, giving up the consolidation should be considered.

In project management, activity criticality index identifies activities whose expected duration may be reduced in order to reduce the duration of the project (Van Slyke, 1963; Williams, 1992). Here, activity criticality is defined as the probability that activity j of shipment m is critical to shipment k. That is, we define AC_{jmk} as $P(Activity(j,m) \in A_k)$, where $Activity(j,m)$ is the activity j of shipment m and $A_k = \{Activity(j,m) | T_k \text{ will be shortened if } t_{jm} \text{ is shortened, } \forall j, m\}$.

We observe that in order to reduce the influence of an activity with high criticality, shortening its expected duration or choosing a faster agent works better than reducing the standard deviation. This will be discussed in detail in Section 8.

Chapter 7 Phase 3: Validate Corrective Measures

After deciding the tolerance level and corrective measures, the forwarder may need to know how well the tolerance level is and how well the corrective measures are before implementing them in real life. In this section, we propose a method to decide whether a particular combination of tolerance level and corrective measure works well to improve reliability. The effectiveness of the respective corrective measures can be validated by performing a simulation experiment.

As mentioned in Section 3, whether a particular corrective measure can induce the shipment to reach the target reliability level depends on the tolerance level. Therefore, when a particular combination of corrective measure and tolerance level does not work well, it is possible to improve the reliability by either increasing the tolerance level or trying another corrective measure. The shipment plan would not work if neither way can improve the reliability further.

Figure 16 shows the logic of the validation algorithm. By simulation, the reliability of shipments is examined. If the reliability does not achieve the target level, the possibility of improvement is checked which tells whether it is possible to reduce the delivery time for replications with delayed shipments. If it is impossible to improve the shipment further, validation fails and the whole shipment plan has to be re-planed by other methods. Otherwise, if some delay indicators are still below the tolerance level, another measure can be tried. If all the delay indicators are higher than the tolerance level, the reliability of delayed shipments may only be improved by increasing the tolerance level.

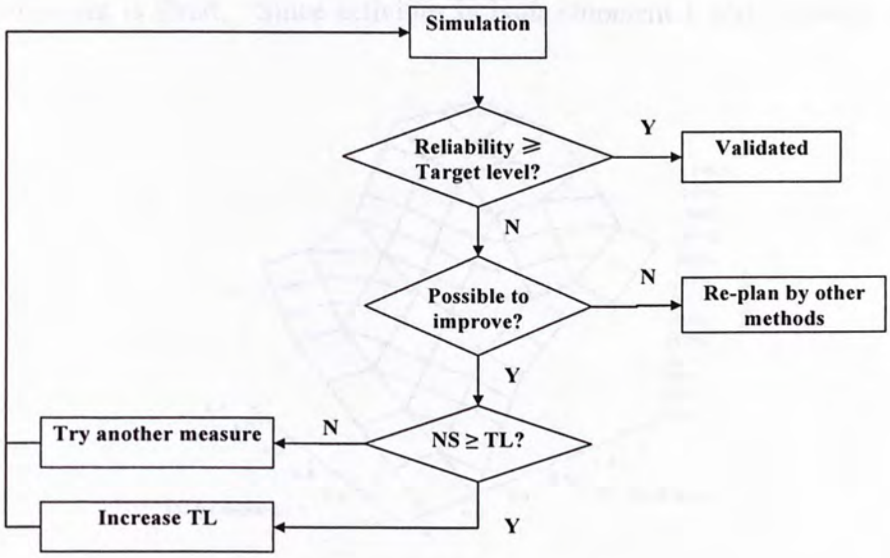


Figure 16 Validation algorithm

To show how different tolerance levels influence the reliability, an experiment is conducted on the example given in Section 1. This experiment examines the change in the reliability of Shipment 1 under the different tolerance levels for activities of Shipment 1 and Shipment 2 respectively (Table 1). The corrective measure is to shorten the expected duration of the next activity each time the potential delay is detected.

Table 1 Experiment 1

Delay indicator	Tolerance level	Corrective measure
NS_{j1l1}	0.0, 0.2, 0.4, 0.6, 0.8, 1.0	Shorten the expected duration of the next activity
NS_{j2l1}	0.4, 0.6, 0.8, 1.0, 1.2, 1.4	

Figure 17 and Figure 18 show the reliability of Shipment 1 and corresponding mean correction cost under different tolerance levels. Increase in the reliability becomes slower when the tolerance level of one shipment increases and that of the other shipment is fixed. Since activities in both Shipment 1 and Shipment 2 have

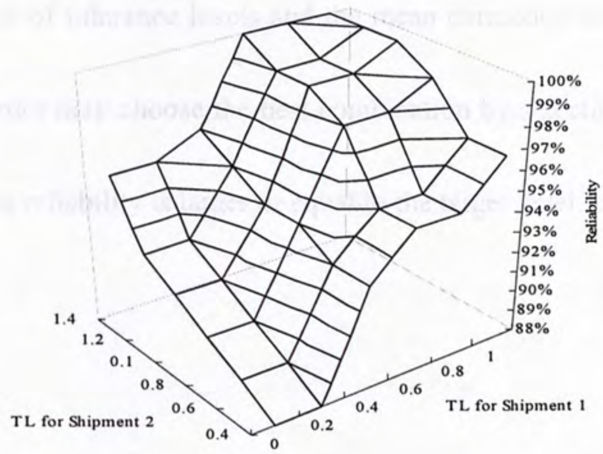


Figure 17 Reliability of Shipment 1 vs. TL's for Shipment 1 and Shipment 2

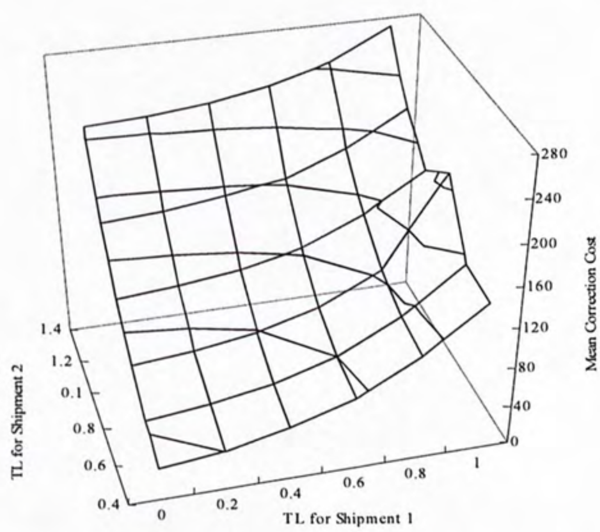


Figure 18 Mean correction cost vs. TL's for Shipment 1 and Shipment 2

influence on the reliability of Shipment 1, increasing the tolerance level of either shipment alone cannot achieve high reliability. Figure 17 shows that the reliability will be closer to 100% only when the tolerance levels of both Shipment 1 and Shipment 2 increase. Moreover, when the tolerance level increases, increase in correction cost becomes faster. Since the same reliability level can be achieved by many combinations of tolerance levels and the mean correction cost has the similar feature, the forwarder may choose the best combination by selecting the lowest cost combination whose reliability is larger or equal to the target level.

Chapter 8 Managerial Insights

The shipment monitoring and controlling process is highly dynamic and complicated, since the outcome depends on many factors, such as the tolerance level, the corrective measure, structure of the shipment plan, etc. This section aims to obtain some insights on this complex problem by doing some experiments.

8.1 Improvement potential, tolerance level and lateness of correction

The experiment conducted in Section 6.2 shows that the combined effect of improvement potential, tolerance level and whether corrective measures are taken early influence both the reliability and the efficiency of the control process.

After further experiment based on a two-shipment plan (Figure 3), we find that reliability increases when the improvement potential increases (Figure 19).

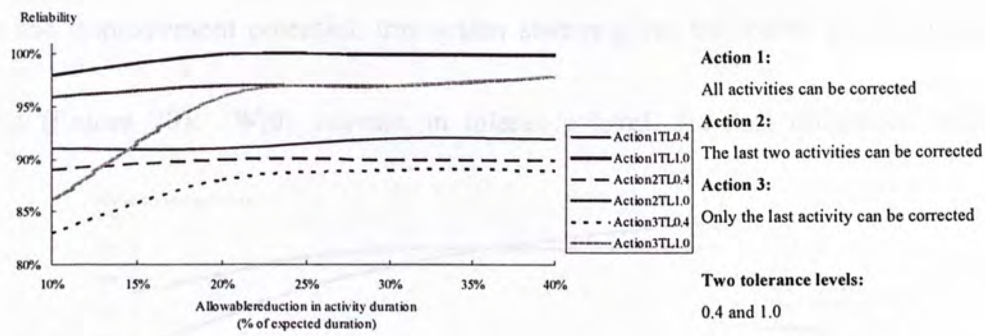


Figure 19 Reliability vs. improvement potential under two tolerance levels

However, if the tolerance level is too low, even very high improvement potential does not help increase the reliability. There are two reasons: low tolerance level has high chance to mistakenly regard cases of potential delay as on-time; additionally, low tolerance level limits the amount of correction to a very low level which may not be enough for achieving high reliability. In other words, when the improvement potential is low, its effect overwhelms the effect of tolerance level; while, when the improvement potential is high, the effect of the tolerance level is overwhelming.

The effectiveness of taking corrective actions only at later activities is very sensitive to the improvement potential. Figure 19 shows that when the improvement potential decreases, the reliability drops dramatically by taking Action 3. However, when the improvement potential increases, taking corrective action one activity earlier (Action 2) does not achieve higher reliability than Action 3, provided that the tolerance level is high enough.

Although the effectiveness of correcting only later activities is highly influenced by the improvement potential, this action always gives the lowest mean correction cost (Figure 20). With increase in tolerance level, the cost difference between

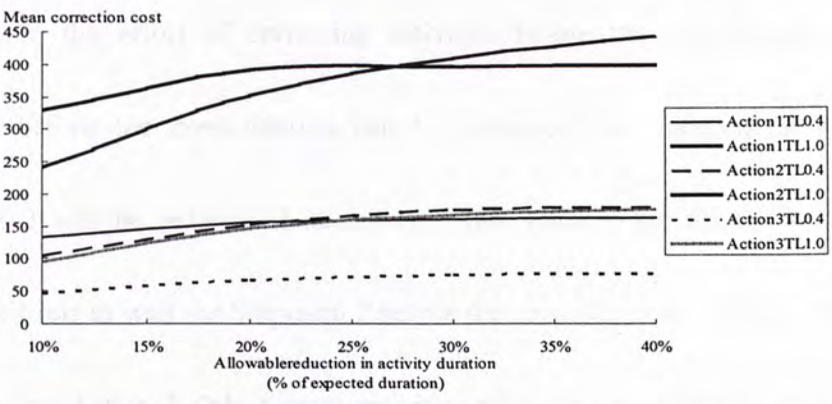


Figure 20 Mean correction cost vs. improvement potential under two tolerance levels

Action 3 and the other actions increases. Thus, only correcting the later activities should be preferred both in terms of reliability and cost, when both the tolerance level and improvement potential are high. More specifically, when the improvement potential is high, the forwarder is able to both improve the reliability and reduce the correction cost by increasing the tolerance level and deferring the corrective actions until later activities are being performed. In Figure 19 and Figure 20, to achieve this, the forwarder can switch from Action 1 with TL=0.4 to Action 3 with TL=1.0.

8.2 Taking corrective measure before consolidation is helpful

Figure 13 shows that correcting all the activities and only the last activity give similar reliability, but Figure 19 shows that taking corrective action at all the activities gives significantly higher reliability than correcting only the last activity (Action 3) and correcting only the last two activities (Action 2). This difference comes from the effort of correcting activities before the consolidation. When activities before the consolidation can be corrected, the duration of activity in Shipment 2 will be reduced if necessary. This reduces the amount of time that Shipment 1 has to wait for Shipment 2 before the consolidation. On the other hand, Action 2 and Action 3 only correct activities after the consolidation, which cannot remove the influence of Shipment 2 on Shipment 1. Therefore, the forwarder should take corrective measures for activities before consolidations, especially when the other shipments have much higher activity criticality than the shipment to be corrected.

8.3 Reducing activity duration is a better way to lower activity criticality

When the shipment is to be consolidated with some other shipments, it is highly possible that delay is caused by waiting for other shipments if these shipments have much higher activity criticality than the delayed shipment. The example in Section 1 belongs to this case. Before consolidation, the activity criticality of Shipment 1 is 29%, while that of Shipment 2 is 71%. As shown in Figure 17, the reliability of Shipment 1 is improved a lot by shortening the activity durations of Shipment 2 before consolidation. In other words, we can improve the reliability by reducing the criticality levels of other shipments' activities before the consolidation happens.

However, which measure, shortening the activity duration or reducing the randomness, is a better way to lower the activity criticality? An experiment is conducted by changing the values of expected durations and standard deviations for the example in Section 1 (Table 2).

Table 2 Experiment 2

Activity changed	Levels of expected duration	Levels of standard deviation
Activity 2 of Shipment 1	46, 48, 50, 52, 54, 56	5%, 10%, 15%, 20%, 25%, 30%
Activity 2 of Shipment 2	54, 56, 58, 60, 62, 64	5%, 10%, 15%, 20%, 25%, 30%

Figure 21 shows that either increasing the expected duration of activity in

Shipment 1 or decreasing the expected duration of activity in Shipment 2 helps increase the criticality of activity in Shipment 1, AC_{211}. That is, AC_{211} depends on the expected duration of activity in of Shipment 1 relative to that in Shipment 2.

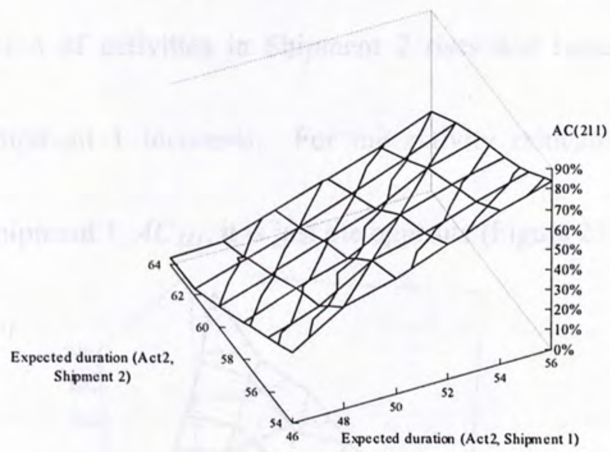


Figure 21 AC_{211} **vs. expected durations of activity 2 in Shipment 1 and Shipment 2**

Counter intuitively, as shown in Figure 22, increasing the standard deviation of the activity in Shipment 2 helps to increase the criticality of the activity in Shipment 1. This is because the expected duration of the activity in Shipment 2 is longer than

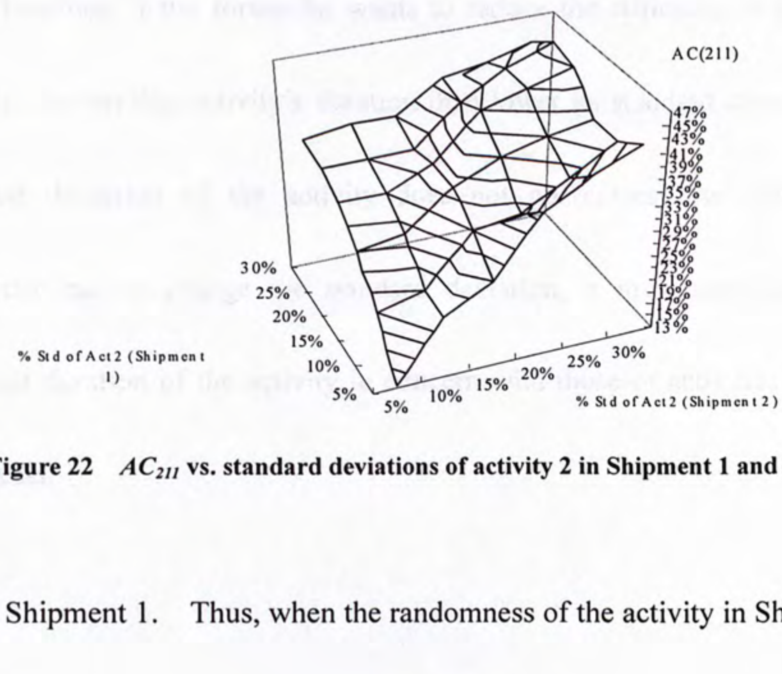


Figure 22 AC_{211} **vs. standard deviations of activity 2 in Shipment 1 and Shipment 2**

that in Shipment 1. Thus, when the randomness of the activity in Shipment 2 is low, the duration of activities in Shipment 2 is close to the expected value which is longer than the duration of activity in Shipment 1. When the standard deviation randomness of the activity in Shipment 2 is high, the number of cases which have very short duration of activities in Shipment 2 rises and hence the criticality of activity 2 in Shipment 1 increases. For the activity criticality of activity 2 in Shipment 2 to Shipment 1, AC_{221}, it is just the opposite (Figure 23).

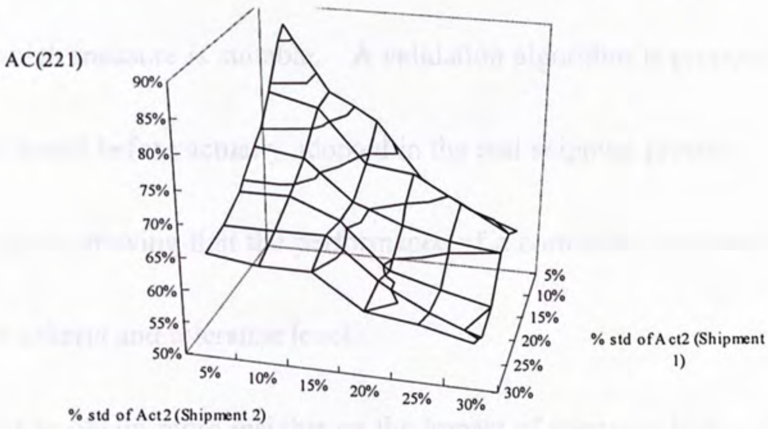


Figure 23 AC_{221} vs. standard deviations of activity 2 of Shipment 1 and Shipment 2

Therefore, if the forwarder wants to reduce the criticality of an activity, it is safer to shorten this activity's duration than lower its standard deviation, since low standard deviation of the activity does not guarantees low criticality. If the forwarder has to change the standard deviation, it must carefully compare the expected duration of the activity in concern with those of activities in consolidated shipments.

Chapter 9 Conclusion and Future Research

This research proposes a framework of monitoring the cargo shipping process, so that each shipment can achieve its target reliability level by taking some corrective measures in the middle of the shipping process. The delay indicator is introduced to detect potential delay. Criteria are the basis to decide whether any corrective measure should be carried out when the delay indicator drops below the tolerance level and which measure is suitable. A validation algorithm is proposed to test the measures selected before actually adopted in the real shipping process. We provide some examples, showing that the performance of a corrective measure does depend on selection criteria and tolerance level.

In order to obtain more insights on the impact of tolerance level and corrective measures on the reliability of the shipment, more experiments are conducted to investigate the relationship among improvement potential, tolerance level and lateness of correction. We also find that taking corrective measures for activities before the consolidation helps improve the reliability.

In this study, we assume that different activities have the same importance, but in reality it may not be true. The consequence of delay in some activities may be much more serious than in other activities. For example, flight delay due to snow storm may have worse consequence than the truck delay caused by a small accident on the highway. Actually, the different levels of seriousness can be modeled into the randomness of different agents. It is possible to be included in future research.

In the future, further research can be conducted on testing the effectiveness of proposed indicator, criteria and measures. We can also incorporate other types of uncertainties so as to investigate a more complicated and real situation.

References

1. Aytug, H., Lawley, M., McKay, K., Mohan, S. and Uzsoy, R. (2005), "Executing Production Schedules in the Face of Uncertainties: A Review and Some Future Directions", *European Journal of Operational Research*, 161(1), pp.86-110
2. Bookbinder J.H., Higginson J.K. (2002), "Probabilistic Modeling of Freight Consolidation by Private Carriage" *Transportation Research Part E: Logistics and Transportation Review*, 38(5), pp. 305-318
3. Bowman, R. A. (1995), "Efficient Estimation of Arc Criticalities in Stochastic Activity Networks", *Management Science*, 41(1), pp.58-67
4. Bowman, R. A. (2006), "Developing Activity Duration Specification Limits for Effective Project Control", *European Journal of Operational Research*, 174, pp.1191-1204
5. Casman, E. A., Morgan, M. G. and Dowlatabadi, H. (1999), "Mixed Levels of Uncertainty in Complex Policy Models", *Risk Analysis*, 19(1), pp.33-42
6. Church, L. K. and Uzsoy, R. (1992), "Analysis of Periodic and Event-driven Rescheduling Policies in Dynamic Shops", *International Journal of Computer Integrated Manufacturing*, 5, pp.153-163
7. Dodin, B. M. and Elmaghraby, S. E. (1985), "Approximating the Criticality Indices of the Activities in PERT Networks", *Management Science*, 31(2), pp.207-223
8. Herroelen, W. and Leus, R. (2004), "Robust and Reactive Project Scheduling: A Review and Classification of Procedures", *International Journal of Production Research*, 42(8), pp.1599-1620
9. Hofer, E., Kloos, M., Krzykacz-Hausmann, B., Peschke, J., and Woltereck, M. (2002), "An Approximate Epistemic Uncertainty Analysis Approach in the

Presence of Epistemic and Aleatory Uncertainties", *Reliability Engineering and System Safety*, 77, pp.229-238

10. Lawrence, S. R. and Sewell, S. E. (1997), "Heuristic, Optimal, Static and Dynamic Schedules When Processing Times Are Uncertain", *Journal of Operations Management*, 15, pp. 71-82
11. Leung, L. C., Hui, Y. V., Wang, Y. (2003), "A 0-1 LP Model for the Integration and Consolidation of Air Cargo Shipments", *Working Paper,* Dept of Decision Sciences and Managerial Economics, Chinese University of Hong Kong.
12. Oberkampf, W. L., Helton, J. C, Joslyn, C. A. Wojtkiewicz, S. F. and Ferson, S. (2004), "Challenge Prolems: Uncertainty in System Response Given Uncertain Parameters", *Reliability Engineering and System Safety*, 85, pp. 11-19
13. Pate-Cornell, M. E. (1996), "Uncertainties in Risk Analysis: Six Levels of Treatment", *Reliability Engineering and System Safety*, 54, pp. 95-111
14. Pooley, J., Stenger, A. J. (1992), "Modeling and Evaluating Shipment Consolidation in a Logistics System", *Journal of Business Logistics*, 13(2), pp.153-164
15. Sabuncuoglu, I. and Bayiz, M. (2000), "Analysis of Reactive Scheduling Problems in A Job Shop Environment", *European Journal of Operational Research*, 126, pp.567-586
16. Shafer, G. (1976), *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, NY.
17. "The 2003 Air Cargo Quality Survey", *Air Cargo World*, Oct, 2003
18. Van Slyke, R. M. (1963), "Monte Carlo Methods and the PERT Problem", *Operations Research*, 11(5), pp.839-860
19. Wan, Y. W. (1995), "Which Is Better, Off-line or Real-time Scheduling",

International Journal of Production Research, 33, pp.2053-2059

20. Williams, T. M. (1992), "Criticality in Stochastic Networks", *Operational Research Society*, 43(4), pp.353-357
21. Williams, T. (1995), "A Classified Bibliography of Recent Research Relating to Project Risk Management", *European Journal of Operational Research*, 85(1), pp. 18-38
22. Wong, W.H., Leung, L.C., Hui, Y.V. (2005), "Airfreight Forwarder Shipment Planning: A Mixed 0-1 Model and Managerial Issues in the Integration and Consolidation of Shipments", *Working Paper*, Dept of Decision Sciences and Managerial Economics, Chinese University of Hong Kong.
23. Wynne, B. (1992), "Science and Social Responsibility", in: J. Ansell and F. Wharton (eds.), *Risk: Analysis, Assessment and Management,* Wiley, New York, pp. 137-152

Appendix: Program code for the simulation model

Part A gives the SIMAN code generated automatically by ARENA when using the modules given by the software's own templates. When a VBA module is used, the customized VBA code is inserted. These VBA codes are given in Part B and written.

A. The SIMAN code generated by ARENA 10.0

(1) Create shipment entities, assign shipment specific values and check which activity is to be performed

```
;       Model statements for module:   Create 3

125$            CREATE,           1,DaysToBaseTime(0),Job1:DaysToBaseTime(1),1:NEXT(126$);

126$            ASSIGN:           Enter Job1.NumberOut=Enter Job1.NumberOut + 1:NEXT(0$);

;       Model statements for module:   Assign 2
;
0$              ASSIGN:           volume=0:
                                  JobIndex=1:
                                  ActIndex=1:
                                  Targettime=0:
                                  weight=0:
                                  Totaltime=0:NEXT(23$);

23$             VBA:              1,vba:NEXT(67$);

                                             → CONNECT TO VBA CODE FOR VBA BLOCK 1

;       Model statements for module:   Assign 74
;
67$             ASSIGN:           IndActTime=TNOW:NEXT(29$);

;       Model statements for module:   Decide 13
;
29$             BRANCH,           1:
                                  If,AgentAssign(JobIndex,ActIndex)==0,129$,Yes:
                                  Else,130$,Yes;
129$            ASSIGN:           Dummy activity?.NumberOut True=Dummy activity?.NumberOut True +
1:NEXT(66$);

130$            ASSIGN:           Dummy activity?.NumberOut False=Dummy activity?.NumberOut False +
1:NEXT(114$);
```

(2) Choose corrective measures

```
;       Model statements for module:   Decide 22
;
114$            BRANCH,           1:
                                  If,CorrType==1,115$,Yes:
                                  If,CorrType==2,119$,Yes:
                                  Else,124$,Yes;
124$            VBA:              13,vba:NEXT(15$);

                                             → CONNECT TO VBA CODE FOR VBA BLOCK 13

115$            VBA:              6,vba:NEXT(15$);

                                             → CONNECT TO VBA CODE FOR VBA BLOCK 6

119$            VBA:              8,vba:NEXT(15$);

                                             → CONNECT TO VBA CODE FOR VBA BLOCK 8
```

(3) Make adjustment for integrations and consolidations and guide shipments to assigned agents

```
;       Model statements for module:   Decide 8
;
15$              BRANCH,          1:
                                 If,IntegraAssign(JobIndex,ActIndex)==1,136$,Yes:
                                 Else,137$,Yes;
136$             ASSIGN:          Integration?.NumberOut True=Integration?.NumberOut True + 1:NEXT(3$);

137$             ASSIGN:          Integration?.NumberOut False=Integration?.NumberOut False + 1:NEXT(71$);

;       Model statements for module:   Decide 1
;
3$               BRANCH,          1:
                                 If,ConsoAssign(JobIndex,ActIndex)==1,138$,Yes:
                                 Else,139$,Yes;
138$             ASSIGN:          Cosolidation?.NumberOut True=Cosolidation?.NumberOut True + 1:NEXT(71$);

139$             ASSIGN:          Cosolidation?.NumberOut False=Cosolidation?.NumberOut False +
1:NEXT(70$);

;       Model statements for module:   Assign 78
;
70$              ASSIGN:
TempActTime(JobIndex,ActIndex)=ActTime(JobIndex,ActIndex)-TimeSaveIntegra(JobIndex,ActIndex)
                                 :NEXT(11$);

;       Model statements for module:   Assign 79
;
71$              ASSIGN:          TempActTime(JobIndex,ActIndex)=ActTime(JobIndex,ActIndex):NEXT(11$);

;       Model statements for module:   Decide 6
;
11$              BRANCH,          1:
                                 If,AgentAssign(JobIndex,ActIndex)==1,77$,Yes:
                                 If,AgentAssign(JobIndex,ActIndex)==2,82$,Yes:
                                 Else,87$,Yes;
```

(4) Set consolidation

```
;       Model statements for module:   Assign 80
;
77$              ASSIGN:          Counter(1,ActIndex)=Counter(1,ActIndex)+1:NEXT(76$);

;       Model statements for module:   Decide 14
;
76$              BRANCH,          1:
                                 If,Counter(1,ActIndex)==BatchSize(1,ActIndex),251$,Yes:
                                 Else,252$,Yes;
251$             ASSIGN:          Last to be consolidated 1?.NumberOut True=Last to be consolidated
1?.NumberOut True + 1:NEXT(75$);

252$             ASSIGN:          Last to be consolidated 1?.NumberOut False=Last to be consolidated
1?.NumberOut False + 1:NEXT(73$);

;       Model statements for module:   Signal 1
;
75$              SIGNAL:          1:NEXT(17$);

;       Model statements for module:   Hold 1
;
73$              QUEUE,           Hold 1.Queue;
                 WAIT:            1:NEXT(17$);
```

(5) Ship or process by the agent

```
;       Model statements for module:   Assign 25
;
17$              ASSIGN:          TimeBFbatch(JobIndex,ActIndex)=TNOW:
                                 Temp shipping time(1,ActIndex)=
```

```
                         MX(Temp shipping time (1, ActIndex), TempActTime(JobIndex,
ActIndex)):NEXT(4$);

;     Model statements for module:   Batch 5
;
4$            QUEUE,         consolidate set1.Queue;
253$          GROUP,         ActIndex,Temporary:BatchSize(1,
ActIndex),First,Job_consolidated:NEXT(254$);

254$          ASSIGN:        consolidate set1.NumberOut=consolidate set1.NumberOut + 1:NEXT(16$);

;     Model statements for module:   Process 12
;
16$           ASSIGN:        Delay due to consolidation 1.NumberIn=Delay due to consolidation 1.NumberIn +
1:
                             Delay due to consolidation 1.WIP=Delay due to consolidation 1.WIP+1;
284$          STACK,         1:Save:NEXT(256$);

256$          DELAY:         ConsoTime(1, ActIndex),,NVA:NEXT(265$);

265$          TALLY:         Delay due to consolidation 1.TotalTimePerEntity,Diff.StartTime,1;
289$          ASSIGN:        Delay due to consolidation 1.NVATime=Delay due to consolidation 1.NVATime +
Diff.NVATime;
290$          TALLY:         Delay due to consolidation 1.NVATimePerEntity,Diff.NVATime,1;
304$          STACK,         1:Destroy:NEXT(303$);

303$          ASSIGN:        Delay due to consolidation 1.NumberOut=Delay due to consolidation
1.NumberOut + 1:
                             Delay due to consolidation 1.WIP=Delay due to consolidation 1.WIP-1:NEXT(50$);

;     Model statements for module:   Assign 65
;
50$           ASSIGN:        TimeAFbatch(1,ActIndex)=TNOW:NEXT(7$);

;     Model statements for module:   Process 6
;
7$            ASSIGN:        Ship by Agent1.NumberIn=Ship by Agent1.NumberIn + 1:
                             Ship by Agent1.WIP=Ship by Agent1.WIP+1;
335$          STACK,         1:Save:NEXT(309$);

309$          QUEUE,         Ship by Agent1.Queue;
308$          SEIZE,         2,VA:
                             Agents,1:NEXT(307$);

307$          DELAY:         GAMM(Temp shipping time (1,
ActIndex)/alpha(1,ActIndex),alpha(1,ActIndex)),,VA:NEXT(350$);

350$          ASSIGN:        Ship by Agent1.WaitTime=Ship by Agent1.WaitTime + Diff.WaitTime;
314$          TALLY:         Ship by Agent1.WaitTimePerEntity,Diff.WaitTime,1;
316$          TALLY:         Ship by Agent1.TotalTimePerEntity,Diff.StartTime,1;
340$          ASSIGN:        Ship by Agent1.VATime=Ship by Agent1.VATime + Diff.VATime;
341$          TALLY:         Ship by Agent1.VATimePerEntity,Diff.VATime,1;
306$          RELEASE:       Agents,1;
355$          STACK,         1:Destroy:NEXT(354$);

354$          ASSIGN:        Ship by Agent1.NumberOut=Ship by Agent1.NumberOut + 1:
                             Ship by Agent1.WIP=Ship by Agent1.WIP-1:NEXT(10$);

;     Model statements for module:   Separate 2
;
10$           SPLIT::NEXT(357$);

357$          ASSIGN:        Separate 1.NumberOut Orig=Separate 1.NumberOut Orig + 1:NEXT(66$);
```

(6) Update information after completing an activity

```
;     Model statements for module:   Assign 73
;
66$           ASSIGN:        IndActTime=TNOW-IndActTime:NEXT(69$);

;     Model statements for module:   Assign 76
;
69$           ASSIGN:        Totaltime=Totaltime+IndActTime:NEXT(68$);
```

```
68$          VBA:          3,vba:NEXT(13$);
```

→ **CONNECT TO VBA CODE FOR VBA BLOCK 3**

```
;     Model statements for module:   Assign 15
;
13$          ASSIGN:          CurrentAct(JobIndex)=CurrentAct(JobIndex) + 1:
                        ActIndex=ActIndex+1:NEXT(14$);
```

(7) Check for uncompleted activities, take record and exit the system if the shipping process is completed

```
;     Model statements for module:   Decide 7
;
14$          BRANCH,          1:
                        If,ActIndex>Num activities,131$,Yes:
                        Else,132$,Yes;
131$         ASSIGN:          Another activity?.NumberOut True=Another activity?.NumberOut True +
1:NEXT(22$);

132$         ASSIGN:          Another activity?.NumberOut False=Another activity?.NumberOut False +
1:NEXT(67$);

;;     Model statements for module:   Assign 36
;
22$          ASSIGN:          System time=MX(System time, Totaltime):NEXT(58$);

58$          VBA:          2,vba:NEXT(9$);
```

→ **CONNECT TO VBA CODE FOR VBA BLOCK 2**

```
;;     Model statements for module:   Dispose 1
;
9$           ASSIGN:          Finish job.NumberOut=Finish job.NumberOut + 1;
133$         DISPOSE:          Yes;
```

B. The VBA code incorporated in SIMAN

(1) Declare variables

```
Option Explicit

Dim m As Model

Dim FileToOpen As String

Dim ArenaDir As String

Dim s As SIMAN

Dim XL As Object

Dim NumJob As Integer
Dim NumAct As Integer
Dim NumAgent As Integer
Dim RepCounter As Integer

'declare time and cost array, agent*act*job
Dim TimeArray(1 To 8, 1 To 8, 1 To 8) As Double
Dim CostArray(1 To 8, 1 To 8, 1 To 8) As Double

'declare integraion and consolidation array, job*act, 0-1 variable array
Dim IntegraArray(1 To 8, 1 To 8) As Integer
Dim ConsoArray(1 To 8, 1 To 8) As Integer
'declare assigned time saving for integration and consolidation, job*act
Dim TsaveIntegra(1 To 8, 1 To 8) As Double
Dim TsaveConso(1 To 8, 1 To 8) As Double
'declare AgentAssignment array, job*act
Dim AgentArray(1 To 8, 1 To 8) As Integer
'declare BatchArray, agent*act
Dim BatchArray(1 To 8, 1 To 8) As Integer
```

```
'declare AlphaArray, agent*act*job
Dim AlphaArray(1 To 8, 1 To 8, 1 To 8) As Integer
```

(2) Initialize each simulation replication

```
Private Sub ModelLogic_RunBeginReplication()

'************ Restore Initial values for variables in simulation model **********************
'restore AgentAssign, ActTime, IntegraAssign, ConsoAssign, TimeSaveIntegra, ConsoTime -- job*act
Dim i As Integer
Dim j As Integer
Dim k As Integer

For k = 1 To NumJob
    For j = 1 To NumAct
        s.VariableArrayValue(s.SymbolNumber("AgentAssign", k, j)) = AgentArray(k, j)
        s.VariableArrayValue(s.SymbolNumber("ActTime", k, j)) = TimeArray(AgentArray(k, j), j, k)
        s.VariableArrayValue(s.SymbolNumber("IntegraAssign", k, j)) = IntegraArray(k, j)
        s.VariableArrayValue(s.SymbolNumber("ConsoAssign", k, j)) = ConsoArray(k, j)
        s.VariableArrayValue(s.SymbolNumber("TimeSaveIntegra", k, j)) = TsaveIntegra(k, j)
        s.VariableArrayValue(s.SymbolNumber("ConsoTime", k, j)) = TsaveConso(k, j)
    Next j
Next k

'restore alpha
For i = 1 To NumAgent
    For j = 1 To NumAct
        For k = 1 To NumJob
            If AgentArray(k, j) = i Then
                s.VariableArrayValue(s.SymbolNumber("alpha", i, j)) = AlphaArray(i, j, k)
            End If
        Next k
    Next j
Next i

'restore BatchSize and alpha matrix for new replication, agent*act
For i = 1 To NumAgent
    For j = 1 To NumAct
        s.VariableArrayValue(s.SymbolNumber("BatchSize", i, j)) = BatchArray(i, j)

    Next j
Next i

End Sub

Private Sub ModelLogic_RunBeginSimulation()

Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim l As Integer

Set m = ThisDocument.Model
Set s = ThisDocument.Model.SIMAN

'Start Excel and get a "handle" to the Excel application, using some variable

'of your own, like XL. For information on Excel's Object Model and how to

'automate Excel you will need to refer to Excel's online help and

'documentation.

Set XL = GetObject(, "Excel.Application")

'Open the first file (Assignment file--workbooks(1))
FileToOpen = "C:\Documents and Settings\zhangmin_dse\Desktop\Sarah Research\Thesis experiments\Two jobs\Case244bcor.xls"
XL.workbooks.Open FileToOpen

'open the second file (data file--workbooks(2))
FileToOpen = "C:\Documents and Settings\zhangmin_dse\Desktop\Sarah Research\Thesis experiments\Two jobs\Case244bdataTSP.xls"
XL.workbooks.Open FileToOpen
```

```
'read number of jobs, acts, and agents
NumJob = XL.workbooks(1).worksheets(1).range("a2")
NumAct = XL.workbooks(1).worksheets(1).range("a4")
NumAgent = XL.workbooks(1).worksheets(1).range("a6")
s.VariableArrayValue(s.SymbolNumber("Num activities")) = NumAct

'set time and cost array
With XL.workbooks(2).worksheets(1).range("d1")
    For k = 1 To NumJob
        For j = 1 To NumAct
            For i = 1 To NumAgent
                TimeArray(i, j, k) = .offset(i + NumAgent * (j - 1) + NumAct * NumAgent * (k - 1), 0)
                CostArray(i, j, k) = .offset(i + NumAgent * (j - 1) + NumAct * NumAgent * (k - 1), 1)
                AlphaArray(i, j, k) = .offset(i + NumAgent * (j - 1) + NumAct * NumAgent * (k - 1), 3)
            Next i
        Next j
    Next k

End With

'initialize replication counter
RepCounter = 1


'************************************ Read Assignments and Initial Parameters ************************************

'AgentAssign -- agent assignment table, job by activity
For i = 1 To NumJob
    For j = 1 To NumAct
        AgentArray(i, j) = XL.workbooks(1).worksheets(2).range("a1").offset(i, j)
    Next j
Next i


'IntegraAssign -- calculate the IntegraAssign table from the excel AgentAssign table (0=no integration, 1=integration,
do not identify agents)
'TimeSaveIntegra -- calculate time savings for each integration
'Job by activity
Dim inteCount As Integer
Dim inteSet As Integer

inteCount = 0
inteSet = 0

With XL.workbooks(1).worksheets(2).range("a1")
For i = 1 To NumJob
    For j = 1 To NumAct - 1
        If .offset(i, j) = .offset(i, j + 1) Then
            IntegraArray(i, j + 1) = 1
        End If
    Next j
Next i
End With

With XL.workbooks(1).worksheets(4).range("b3")
For k = 1 To NumJob
    For i = 1 To NumAgent
        For j = 1 To NumAct
            If AgentArray(k, j) = i And IntegraArray(k, j) = 1 Then
                inteCount = inteCount + 1
                If j = NumAct Then
                    inteSet = inteSet + 1
                    For l = j - inteCount + 1 To j
                        TsaveIntegra(k, l) = .offset(inteSet, 0) / inteCount
                    Next l
                    inteCount = 0
                End If
                If j < NumAct Then
                    If IntegraArray(k, j + 1) = 0 Then
                        inteSet = inteSet + 1
                        For l = j - inteCount + 1 To j
                        TsaveIntegra(k, l) = .offset(inteSet, 0) / inteCount
                        Next l
                        inteCount = 0
```

```
                    End If
                End If
            End If
        Next j
    Next i
Next k
End With

'ConsoAssign -- calculate the ConsoAssign table from the excel AgentAssign table (0=no consolitation,
1=consolitation, do not identify agents)
'Job by activity

With XL.workbooks(1).worksheets(2).range("a1")
For j = 1 To NumAct
    For i = 1 To NumJob
        For k = i + 1 To NumJob
            If .offset(i, j) = .offset(k, j) Then
                ConsoArray(i, j) = 1
                ConsoArray(k, j) = 1
            End If
        Next k
    Next i
Next j
End With

'Batchsize -- calculate the Batchsize table for consolidation from the excel AgentAssign table
'Agent by activity
Dim n As Integer
With XL.workbooks(1).worksheets(2).range("a1")
For j = 1 To NumAct
    For n = 1 To NumAgent
        For i = 1 To NumJob
            If .offset(i, j) = n Then
                s.VariableArrayValue(s.SymbolNumber("Batchsize", n, j)) =
s.VariableArrayValue(s.SymbolNumber("Batchsize", n, j)) + 1
                BatchArray(n, j) = s.VariableArrayValue(s.SymbolNumber("Batchsize", n, j))
            End If
        Next i
    Next n
Next j
End With


'ConsoTime -- assign the time for consolidation for each consolidation set
'Agent by activity
Dim consoSet As Integer
Dim consoCount As Integer

consoSet = 0

With XL.workbooks(1).worksheets(4).range("e3")
For j = 1 To NumAct
    For i = 1 To NumAgent
        consoCount = 0
        For k = 1 To NumJob
            If ConsoArray(k, j) = 1 And AgentArray(k, j) = i Then
                consoCount = consoCount + 1
                If consoCount = BatchArray(i, j) Then
                    consoSet = consoSet + 1
                    TsaveConso(i, j) = .offset(consoSet, 0)
                End If
            End If
        Next k
    Next i
Next j
End With

'Read scheduled time and standard deviation (no correction) for remaining activities, job*act, 1st value is from the
beginning of act1
With XL.workbooks(1).worksheets(3).range("a14")
```

```
    For k = 1 To NumJob
        For j = 1 To NumAct
            s.VariableArrayValue(s.SymbolNumber("RemainTime", k, j)) = .offset(k, j)
            'job1 wait job2
            s.VariableArrayValue(s.SymbolNumber("stdRact", k, j)) = .offset(k + 12, j)
            'job2 wait job1
            s.VariableArrayValue(s.SymbolNumber("stdRact2", k, j)) = .offset(k + 12 + 11, j)

        Next j
    Next k
End With

End Sub


Private Sub ModelLogic_RunEndReplication()

Dim k As Integer
Dim j As Integer

'record total system cost and time for each replication
Dim resFailure As Integer

With XL.workbooks(1).worksheets(5).range("a2")
    .offset(RepCounter, 9) = s.VariableArrayValue(s.SymbolNumber("System time"))
End With


'record integration time saving
With XL.workbooks(1).worksheets(4).range("g3")
For k = 1 To NumJob
    For j = 1 To NumAct
        .offset(k, j) = s.VariableArrayValue(s.SymbolNumber("TimeSaveIntegra", k, j))
    Next j
Next k
End With


'count for starting another replication
RepCounter = RepCounter + 1


End Sub
```

(3) Code for VBA Block 1 (Read entity attributes from Excel files)

```
Private Sub VBA_Block_1_Fire()

'************************ Initialize Entity attributes ****************************
Set s = ThisDocument.Model.SIMAN

Dim i As Integer

With XL.workbooks(1).worksheets(1).range("a8")
For i = 1 To NumJob
    If s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("JobIndex")) = i Then
        s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("Targettime")) = .offset(1, i)
        s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("weight")) = .offset(2, i)
        s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("volume")) = .offset(3, i)
    End If
Next i
End With

End Sub
```

(4) Code for VBA Block 13 (Correction Method 1)

```
Private Sub VBA_Block_13_Fire()

'Correction method1 -- Shorten ActTime, NSTD, lowerbound=0.8*ActTime, check latest conso slack of job 2 for job 1
at the end of act 1
Dim k As Integer
```

```
Dim j As Integer
Dim i As Integer

k = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("JobIndex"))
j = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ActIndex"))

Dim slack As Double
Dim deadline As Double
Dim remain As Double
Dim limit As Double
Dim extracost As Double
Dim unicost As Double
Dim allowlimit As Double
Dim newActTime As Double

allowlimit = 0.4
deadline = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("Targettime"))
remain = s.VariableArrayValue(s.SymbolNumber("RemainTime", k, j))
limit = s.VariableArrayValue(s.SymbolNumber("Limit", k, j))

If k = 1 Then
    slack = deadline - s.RunCurrentTime - remain
End If

If k = 2 Then
    slack = 235 - s.RunCurrentTime - remain
End If

If slack < limit Then
    unicost = CostArray(s.VariableArrayValue(s.SymbolNumber("AgentAssign", k, j)), j, k) /
s.VariableArrayValue(s.SymbolNumber("ActTime", k, j))
    If limit - slack <= allowlimit * s.VariableArrayValue(s.SymbolNumber("ActTime", k, j)) Then
        newActTime = s.VariableArrayValue(s.SymbolNumber("ActTime", k, j)) - (limit - slack)
        s.VariableArrayValue(s.SymbolNumber("ActTime", k, j)) = newActTime
        extracost = unicost * (limit - slack)
        With XL.workbooks(1).worksheets(7).range("a3")
            .offset(RepCounter, 8 * (j - 1) + k) = extracost
        End With

    Else
        extracost = unicost * allowlimit * s.VariableArrayValue(s.SymbolNumber("ActTime", k, j))
        newActTime = (1 - allowlimit) * s.VariableArrayValue(s.SymbolNumber("ActTime", k, j))
        s.VariableArrayValue(s.SymbolNumber("ActTime", k, j)) = newActTime
        With XL.workbooks(1).worksheets(7).range("a3")
            .offset(RepCounter, 8 * (j - 1) + k) = extracost
        End With
    End If

    With XL.workbooks(1).worksheets(8).range("a3")
        .offset(RepCounter, 8 * (j - 1) + k) = 1
    End With

    'Make the duration of the consolidation activity the same for two shipments
    If j = 3 Then
        s.VariableArrayValue(s.SymbolNumber("ActTime", 1, j)) = newActTime
        s.VariableArrayValue(s.SymbolNumber("ActTime", 2, j)) = newActTime
    End If

Else
    With XL.workbooks(1).worksheets(7).range("a3")
        .offset(RepCounter, 8 * (j - 1) + k) = 0
    End With
    With XL.workbooks(1).worksheets(8).range("a3")
        .offset(RepCounter, 8 * (j - 1) + k) = 0
    End With

End If
End Sub
```

(5) Code for VBA Block 2 (Record total duration to Excel files)

```
Private Sub VBA_Block_2_Fire()
```

```
'record total time for each job
With XL.workbooks(1).worksheets(5).range("a2")
      .offset(RepCounter, s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("JobIndex"))) =
s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("Totaltime"))
End With

End Sub
```

(6) Code for VBA Block 3 (Record individual activity durations to Excel files)

```
Private Sub VBA_Block_3_Fire()

'record act time for each job each activity

Dim k As Integer
Dim j As Integer

k = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("JobIndex"))
j = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ActIndex"))

With XL.workbooks(1).worksheets(6).range("a3")
      .offset(RepCounter, k + 8 * (j - 1)) = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("IndActTime"))
End With

End Sub
```

(7) Code for VBA Block 6 (Correction Method 2)

```
Private Sub VBA_Block_6_Fire()

'Correction method2--Shorten ActTime, If conso, break consolidation, NSTD
Dim k As Integer
Dim j As Integer
Dim i As Integer

k = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("JobIndex"))
j = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ActIndex"))

Dim slack As Double
Dim deadline As Double
Dim remain As Double
Dim limit As Double
Dim extracost As Double
Dim unicost As Double

deadline = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("Targettime"))
remain = s.VariableArrayValue(s.SymbolNumber("RemainTime", k, j))
limit = s.VariableArrayValue(s.SymbolNumber("Limit", k, j))
slack = deadline - s.RunCurrentTime - remain
extracost = 0

If k = 1 Then
      slack = deadline - s.RunCurrentTime - remain
End If

If k = 2 Then
      slack = 235 - s.RunCurrentTime - remain
End If

If slack < limit Then
      unicost = CostArray(s.VariableArrayValue(s.SymbolNumber("AgentAssign", k, j)), j, k) /
s.VariableArrayValue(s.SymbolNumber("ActTime", k, j))
      s.VariableArrayValue(s.SymbolNumber("ActTime", k, j)) = s.VariableArrayValue(s.SymbolNumber("ActTime", k,
j)) - (limit - slack)
      extracost = unicost * (limit - slack)

      If j = 3 Then
            If s.VariableArrayValue(s.SymbolNumber("Counter", 3, j)) = 1 Then
                  s.VariableArrayValue(s.SymbolNumber("Breakconso", 3, j)) = 1
            End If
            If s.VariableArrayValue(s.SymbolNumber("Counter", 3, j)) = 0 Then
                  s.VariableArrayValue(s.SymbolNumber("Breakconso", 3, j)) = 1
                  s.VariableArrayValue(s.SymbolNumber("Batchsize", 3, j)) = 1
```

```
                s.VariableArrayValue(s.SymbolNumber("ConsoAssign", 1, j)) = 0
                s.VariableArrayValue(s.SymbolNumber("ConsoAssign", 2, j)) = 0
            End If
            extracost = extracost + 200
        End If

        With XL.workbooks(1).worksheets(7).range("a3")
            .offset(RepCounter, 8 * (j - 1) + k) = extracost
        End With
Else
        With XL.workbooks(1).worksheets(7).range("a3")
            .offset(RepCounter, 8 * (j - 1) + k) = 0
        End With

End If

End Sub
```

(8) Code for VBA Block 8 (Correction Method 3)

```
Private Sub VBA_Block_8_Fire()

'Correction method3--Shorten ActTime, STD, Lowerbound=0.8 of the ActTime
Dim k As Integer
Dim j As Integer
Dim i As Integer

k = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("JobIndex"))
j = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("ActIndex"))

Dim slack As Double
Dim deadline As Double
Dim remain As Double
Dim limit As Double
Dim extracost As Double
Dim unicost As Double
Dim stdRact As Double
Dim stdslack As Double

deadline = s.EntityAttribute(s.ActiveEntity, s.SymbolNumber("Targettime"))
remain = s.VariableArrayValue(s.SymbolNumber("RemainTime", k, j))
limit = s.VariableArrayValue(s.SymbolNumber("Limit", k, j))
slack = deadline - s.RunCurrentTime - remain
stdRact = s.VariableArrayValue(s.SymbolNumber("stdRact", k, j))
stdslack = slack / (stdRact / 10)

If stdslack < limit Then
        unicost = CostArray(s.VariableArrayValue(s.SymbolNumber("AgentAssign", k, j)), j, k) /
s.VariableArrayValue(s.SymbolNumber("ActTime", k, j))
        If limit - slack <= 0.2 * s.VariableArrayValue(s.SymbolNumber("ActTime", k, j)) Then
            s.VariableArrayValue(s.SymbolNumber("ActTime", k, j)) = s.VariableArrayValue(s.SymbolNumber("ActTime",
k, j)) - (limit - slack)
            extracost = unicost * (limit * stdRact / 10 - slack)
            With XL.workbooks(1).worksheets(7).range("a3")
                .offset(RepCounter, 8 * (j - 1) + k) = extracost
            End With

        Else
            extracost = unicost * 0.2 * s.VariableArrayValue(s.SymbolNumber("ActTime", k, j))
            s.VariableArrayValue(s.SymbolNumber("ActTime", k, j)) = 0.8 *
s.VariableArrayValue(s.SymbolNumber("ActTime", k, j))
            With XL.workbooks(1).worksheets(7).range("a3")
                .offset(RepCounter, 8 * (j - 1) + k) = extracost
            End With
        End If

        With XL.workbooks(1).worksheets(8).range("a3")
            .offset(RepCounter, 8 * (j - 1) + k) = 1
        End With

 Else
        With XL.workbooks(1).worksheets(7).range("a3")
            .offset(RepCounter, 8 * (j - 1) + k) = 0
        End With
```

```
        With XL.workbooks(1).worksheets(8).range("a3")
            .offset(RepCounter, 8 * (j - 1) + k) = 0
        End With

    End If

End Sub
```