HIGH VOLUME CONVEYOR SORTATION SYSTEM ANALYSIS

A Dissertation
Presented to
The Academic Faculty

by

Ying Wang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and System Engineering

Georgia Institute of Technology
August, 2006

HIGH VOLUME CONVEYOR SORTATION SYSTEM ANALYSIS

Approved by:

Dr. Chen Zhou, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Gunter Sharp
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Leon F. McGinnis
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Yorai Wardi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Spiridon Reveliotis
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: May 3,2006

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Chen Zhou, for his numerous supports and mentorship. I learned a lot from him both academically and personally, which will become an invaluable asset in my future career life. I wish to thank Dr. Leon McGinnis, Dr. Spiridon Reveliotis, Dr. Gunter Sharp and Dr. Yorai Wardi for serving on my committee and for their careful reading of my dissertation. Their insightful comments and suggestions greatly helped improve this research.

I would like to deliver a special gratitude to Dr. Leon McGinnis and professors in Virtual Factory Laboratory, who have supported me during the course of my PhD study. I thank all the professors in Georgia Institute of Technology who taught me in the past five years. Their high quality courses will always be my treasure.

Finally, I would like to thank my parents, Youguo Wang and Lisheng Wang, and my husband, Rongqiang Qin. They have always been a tremendous source of encouragement in my life. I also thank to my lovely daughter, Tina for her understanding of my busy schedule during my PhD study. I am also grateful to all church members in Chinese Campus Fellowship for their prayers and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DC	Distribution Center
A/M	Accumulation/Merge
BPN	Batches Petri Nets
S-BPN	Stochastic Batches Petri Nets
DNFP	Dynamic Network Flow Problem
MFF	Maximum Firing Flow
IFF	Instantaneous Firing Flow
LP	Linear Programming
WIP	Work in Process
IPA	Infinitesimal Perturbation Analysis

LIST OF NOTATION

v	Conveyor Driving Speed
l	Conveyor Length
d	Conveyor Maximum Density
c	Conveyor Type
$\mathcal{I}(t)$	Input Flow Rate
$\mathcal{C}(t)$	Discharge Capacity
$I(t)$	Effective In-flow Rate
$O(t)$	Effective Out-flow Rate
$F(t)$	Overflow Rate
$v_c(t)$	Effective Speed of Non-Accumulation Conveyor
$a(t)$	Accumulation position
r	Maximum Transfer Capacity of a junction
b	Maximum Buffer Capacity of a Junction
σ	Transfer Control Logic of a Junction
$d_o(t)$	Output Batch Density
$d_i(t)$	Input Batch Density
$d_a(t)$	Accumulating Batch Density
τ	Segment Traversal Time
$\Lambda(t)$	Random Arrival Process
$\lambda(t)$	A Sample of $\Lambda(t)$
$\mathcal{L}(\Lambda,\theta)$	Random Loss Volume

$L(\lambda, \theta)$	Replication Loss Volume
$s(L)$	Subgradient of L
N	Number of Replications in Monte Carlo Sampling

SUMMARY

The design and operation of a high volume conveyor sortation system are important due to its high cost, large footprint and critical role in the system. In this thesis, we study the characteristics of the conveyor sortation system from performance evaluation and design perspectives employing continuous modeling approaches.

We present two continuous conveyor models ("Delay and Stock Model" and "Batch on Conveyor Model") with different representation accuracy in a unified mathematical framework. Based on the Batch on Conveyor Model, we develop a fast fluid simulation methodology. We address the feasibility of implementing fluid simulation from modeling capabilities, algorithm design and simulation performance in terms of accuracy and simulation time.

From a design perspective, we focus on rates determination and accumulation design in the accumulation and merge subsystem. The optimization problem is to find a minimum cost design that satisfies some predefined performance requirements under stochastic conditions. We first transform this stochastic programming problem into a deterministic nonlinear programming problem through sample path based optimization method. A gradient based method is adopted to solve the deterministic problem. Since there is no closed form for performance metric even for a deterministic input stream, we adopt continuous modeling to develop deterministic performance evaluation models and conduct sensitivity analysis on these models. We explore the prospects of using the two continuous conveyor models we presented.

First we model the performance evaluation problem as a class of dynamic network flow problems by using Delay and Stock Model. Not only does this model give us the performance metric but it also provides us with an analytical form of subgradient of performance metric with respect to the design parameters. Second, we adopt the Batch on Conveyor Model and employ fluid simulation as a performance evaluation tool. We

investigate the approach on a two-segment tandem conveyor system which is the building block of more complex systems. For this system, we derive a sensitivity estimator by applying infinitesimal perturbation analysis (IPA) and study the performance of this estimator through numerical experiments.

CHAPTER 1

INTRODUCTION

1.1 Overview

To support increasing throughput in the fierce world of retailing, logistics service providers and retailers are consolidating their distribution business by expanding or rebuilding large distribution centers (DCs). These DCs have large customer base and high order fulfillment capacity. The daily throughput can be hundreds of thousands of items. High speed sortation is an inevitable choice when company's order volume reaches certain level. It is stated in DC Velocity, June 2003 that "the outlook for sortation equipment sales remains relatively bright. Even with the dot-com meltdown and the feeble economy, demand for high-speed sorters has held its own, bolstered by DC managers who hope that sortation systems' fabled ability to increase productivity, reduce costs and improve customer satisfaction will help them rev up their operations."

Table 1.1 shows fifteen recently built DC profiles from different industries. All these DCs use conveyorized sortation systems. The system expands the pick and loading area throughout the facility to allow high rate of item induction. The inducted items are merged again and again under control to form a high speed flow for final sortation to the shipping lanes. These conveyor sortation systems are characterized by large footprint, complex configurations, high-throughput, high WIP and high cost. The design and control of these systems have significant impact on the system performance such as throughput, order response time and operating cost. The system is so important that a manager replied to a curious visitor that when the conveyor is down, the DC is dead.

This research is concerned with conveyor sortation system design. We especially focus on the systems for batch picking and sortation of full cases.

Table 1.1 Profiles of DC Employing Conveyor Sortation System

Company	Product	Size (Sq Meter)	Daily Throughput	Store Served	Conveyor Length (km)	Merge Hierarchy
Wal-Mart	grocery	111,480	240,000	87	37	14-4-2
Toys “R” US	toy	91,042	168,000 (Peak)	110	6.84	9-5-3-1
IDG Books	books	24,340	160,000			3-1
TechData	IT product	22,575	2,500 orders	9000		3-1
Target	grocery	125,415	250,000		13	
Walgreen	drug	65,030		600-800	23	
JCPenney stores	apparel	102,190	85,000	1,200	13.7	3-1
National Retail Consolidator	3PL Crossdocking	46,450	45,000	500	8	14-2-1
Crate&Barrel	home ware	41,805	70,000 (Peak)	75	4.8	5-1
Big Lot	grocery	111,480	200,000	350	8.4	13-1
Elizabeth Arden	cosmetics	37,160	6,000 orders	35,000		3-1
McKesson Pharmaceutical	drug	20,903	35,000 orders		3.2	
Rite Aid	drug	81,288	60,000 (peak)	400		16-2-1

In this chapter, we first give a general description of conveyor sortation operation and identify major design issues in section 1.2 and 1.3. A literature review on sortation operations research and general conveyor system research is given on section 1.4. We state our research objectives at section 1.5.

1.2 Conveyor Sortation System Operation

The major functions in a DC are shown in Fig 1.1.

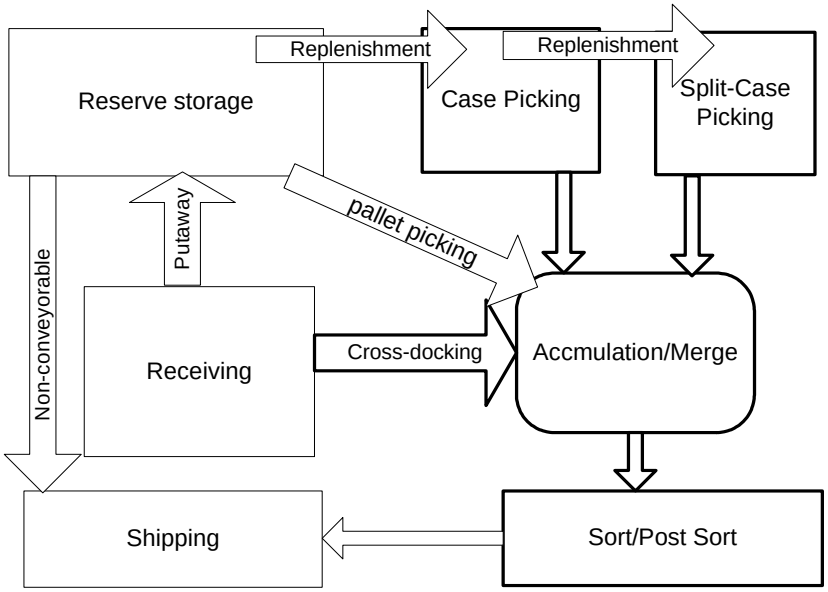


Figure 1.1 Distribution Center Material Flow

The four main operations of a DC are receiving, storage, picking and shipping. Figure 1.1 provides an overview of how product flows from receiving to shipping. In the receiving area, products arrive with low variety but high volume. Products are then either put in the Accumulation/Merge system, or stacked in pallets and put-away into storage. The order picking is normally the most demanding operation in a DC. High throughput DCs have tens of thousands of SKUs and many of them are picked frequently. Therefore, the pick area is very large. To improve the pick rate and keep the pick tours short, large pick area is often divided into several pick zones in high throughput DCs. Conveyors convert any area into a pick area by connecting it to the sorter. Some picking zones handle case picks while others handle split-case picks. For example, in a retail chain regional distribution center, there are 48 case zones; each zone is over 100 meters long. A picker can pick over 400 cases per hour to the conveyor. Picking modules are typically built over one another to improve space utilization. The picked items and cross-dock items are accumulated and merged and then inducted to the sorter where they are pushed

to the appropriate shipping doors. Non-conveyable products will be sent directly to shipping dock from reserve storage area. Case picking modules and Split-Cast picking modules are replenished with stock from reserve area and case storage area regularly with forklifts. In this research, we will not address non-conveyable items and replenishment.

In the graph, the functional blocks or arcs with bold frame are served by conveyor systems in high throughput DCs. The conveyor provides connectivity to large area and buffering to streamline the flow before sorting. The aggregation of all these modules is called a conveyor sortation system. Transporting, merging, identifying, inducting and separating products are main functions of a conveyor sortation system. From research point of view, we further decompose a sortation system into three main subsystems: (1) Order Picking (OP), (2) Accumulation/Merge (A/M), and (3) Sort/Post Sort (S/PS). An abstract illustration of three subsystems modified from Wagner 1994 is shown in Fig 1.2.

We will discuss the operations and the design issues of these three subsystems below in detail.

1.3 Operation and Design Issues of Subsystems

1.3.1 Order Picking Subsystem

The decision to use conveyorized sort implies *batch picking* in zones. In batch picking, a picker is responsible for many (batch) orders from a subset (zone) of SKUs. The batches from zones are merged then sorted to orders, destinations or specific carriers. A large-scale variant of batch picking is *wave picking*. Wave picking refers to the operation that the demands from large number of orders (50-100) are accumulated and picked and conveyed to the sorter as a batch. Wave picking is frequently adopted in high volume distribution system where there would be too many orders in progress without waves. Another example of wave picking is when a set of orders together forms a

truckload or set of trucks that occupy all shipping doors. Operations with high total number of SKUs and moderate to high picks per order may benefit from wave picking.

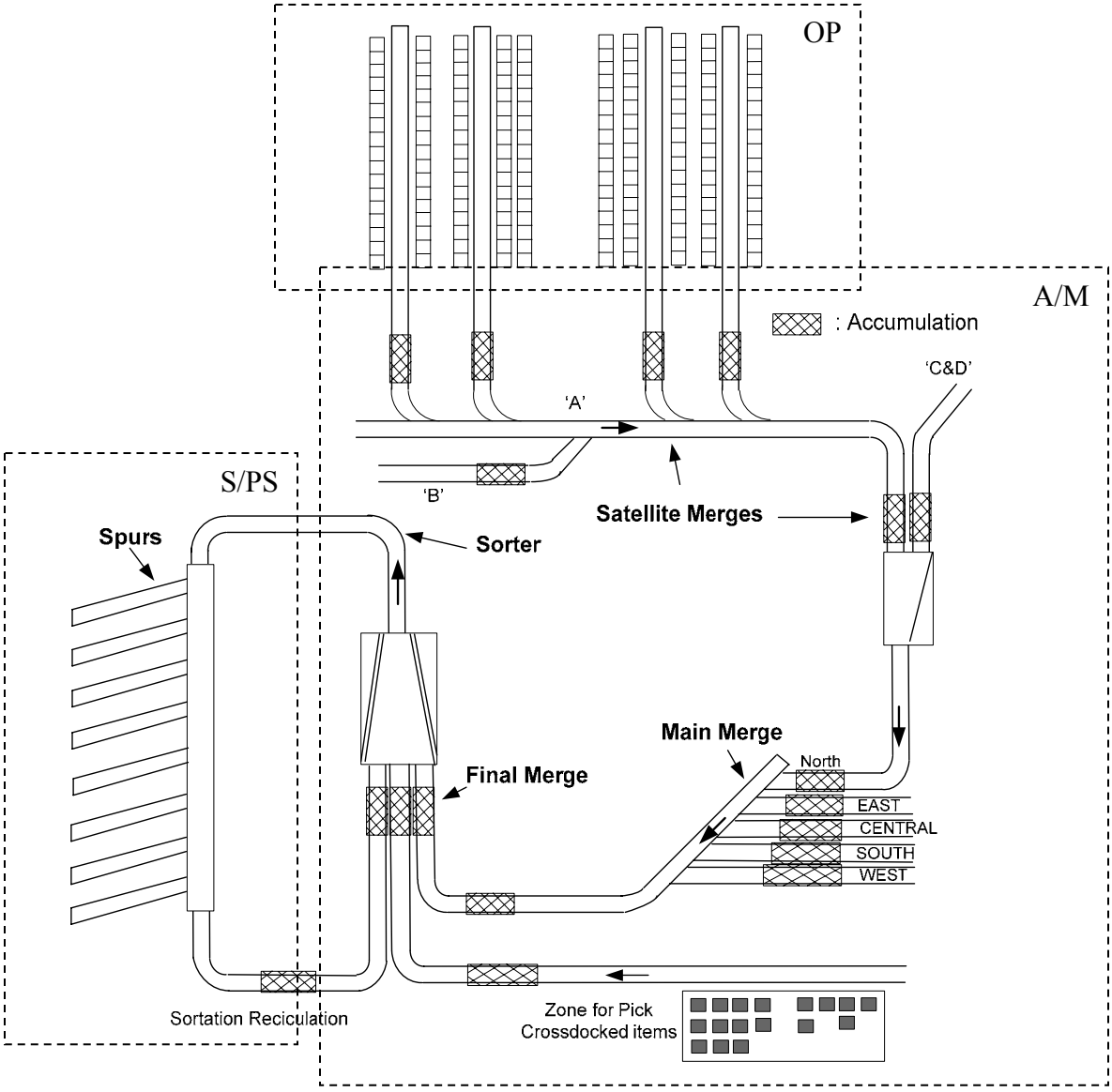


Figure 1.2 Subsystems of Conveyor Sortation System

The challenge to form a batch or wave is to synchronize the flow from the various picking zones (Apple, 1999). Because of varying travel distances, different pick rates and the imbalance in the number of cases for a batch from each zone, some zones will finish early and others will finish late. The implication is lower utilization of the sorter towards

the end of a wave and some pickers' inefficiency due to the idle time waiting for next wave to begin.

Synchronization issue should be taken into account in wave algorithm design. A wave algorithm determines how many orders to send out in a wave and the composition of the wave. On the other hand, good A/M system design and control can also ameliorate the unbalanced situation.

1.3.2 Accumulation/Merge (A/M) Subsystem

A/M subsystem is the conveyor system that collects items from picking and receiving, accumulates, merges, and prepares for sortation. Its objective is to provide a constant flow of items to the sortation system. The upstream merge points near the picking area or receiving are called satellite merges and the downstream merge point near the sorter induction point are called main merges. Accumulation is found before most merge points to buffer the peaks and valleys of the instantaneous input rates from different induction lines.

The hypothetical A/M system in Fig 1.2 represents four pick lines from Pick Module A, a delivery conveyor from an identical Pick Module B and a merge with a delivery conveyor from identical Pick Modules C & D. The NORTH Picking Area (comprised of the above four Pick Modules A, B, C, & D) is then merged with identical Picking Areas EAST, CENTRAL, SOUTH, and WEST. These picked items are then merged with a line from crossdocking area and the recirculation line from the shipping sortation function. Various merges and accumulations in A/M system are responsible for blending the incoming traffic from different sources into a main stream and take care of the rate changes of the incoming traffic to achieve high sorter utilization.

A/M subsystem has high impact on the overall performance of the conveyor sortation system. First, A/M system is responsible for the challenge task to achieve the uniform and high speed flow of items in the right sequence before sort. If the taking away

system never blocks the sorter, A/M subsystem governs the throughput of the system. Second, A/M has large footprint. The traversing and waiting time on this subsystem accounts for large proportion of total cycle time. So improving A/M subsystem efficiency has great potential in improving the order response time. Third, the motion and control of high speed flow are most technically demanding. The construction and operating cost incurred in this subsystem is very high with sophisticated accumulating and merging devices and control logic being in use.

The decisions related to A/M system design include:

- Conveyor path design
- Rate determination
- Accumulation design
- Equipment selection
- Merge/diverge policy design

Several considerations need to be take into account in conveyor path design such as directionality, connectivity, space requirement, shortest distance and number of specific merge or diverge devices needed etc. This is a topic addressed in the class of facility layout design problem.

By rate determination, we mean specifying the transferring capacity for every component of the system such as sorter, merging device and every conveyor segment. The objective is to achieve the minimum rate configuration while satisfying the throughput requirement. Accumulation design needs to determine where to put accumulation and how much the accumulation would be enough. Good accumulation design can provide buffering of peaks and valleys in the instantaneous induction rates so that maintain high sorter utilization. Accumulation conveyor is much more expensive than traditional motorized conveyor, so we want keep the accumulation level as low as possible provided that it satisfies the operation requirement. Rate design and accumulation design should be performed simultaneously to reach the best tradeoff

between them since in certain cases lower rate can be achieved by increasing accumulation and vice verse.

In A/M subsystem, there is a variety of equipment offered to accomplish the specific functionality. For instance, for accumulation, one can choose from wheel accumulation and live roller accumulation and for merge, one can choose from live roller merge, herringbone merge or sliding-shoe merge etc. Usually, there is a maximum rate limit for each equipment type. Cost effective equipment should be chosen based on the item profile and rate requirements.

Most A/M conveyor system can be controlled manually and automatically. In automatic operating model, some control logic needs to be set up. For example, the merge logic was used to prioritize the release sequence of the in-feed conveyors. Determination of the parameters used in these logics is also part of design problem.

1.3.3 Sort/Post Sort Subsystem

Sortation system sorts mixed orders or batches to planned shipping destinations. The sorting process involves identifying the item's destination, tracking the item along its conveyor path and then physically diverting the carton to the proper destination. The function of the takeaway conveyor is to accept product diverted from the sorter conveyor so the sorter is not blocked. Sort/Post Sort system may be classified (Bozer, 1988) for analysis purpose into two types: S/PS-1 and S/PS-2, depending on the relationship between number of lanes and number of orders.

In S/PS-1, the number of lanes is less than the number of orders. Direct customer order fulfillment systems, such as e-tailers (e.g. Amazon.com), catalog distributors (e.g. J.Crew), fall into this category. The items without lane assignment are recirculated. The major bottleneck of the S/PS-1 process is normally the recirculation loop.

In S/PS-2, the number of lanes is equal or more than the number of orders. Lanes are provided so that every incoming item will be destined to a particular lane. Most

retailer distribution centers serving stores within a certain region employ this type of system. Recirculation occurs only when the corresponding lane is full or identification fails.

Sometimes both system types are used in one facility. Type 1 is used to sort the items into orders, while type 2 is used to sort the orders into shipments depending on the type of shipment required (UPS, FedEx, LTL and so forth).

Key considerations in S/PS subsystem design include:

- How many diverting lanes needed?
- How long is the length of the diverting lane?
- How long is the recirculation conveyor?
- Determination of lane assignment strategy (for S/PS-1);
- Selection of order prioritizing rules (for S/PS-1).

As we will see in the literature review, the S/PS system analysis has received some attention from researchers and the above issues have been addressed in academic papers. We do not intend to include these issues in our research scope. Key results of these issues were included in literature review section along with its reference.

Conveyor sortation system design should be accounted for in the overall warehouse design procedure due to its interrelation with system requirement, warehouse layout, storage assignment and general flow pattern. The design would be performed after the establishment of operating policies, arrangement of departments and zone formation. Usually, several iterations may be involved between sortation system design and previous or subsequent steps before the design gets finalized. In this research, we only focus on the decisions made at conveyor sortation system design step while assuming the warehouse layout, order picking scheme and storage partition are already known.

1.4 Literature Review

We review the research from two perspectives, research on sortation system design and research on general conveyor theory.

1.4.1 Literature on Sortation System

Relative scarce prior research has been conducted on the design and operation of overall high volume conveyor sortation systems. Most researches focus on the Sort/Post Sort subsystem design. A brief summary of important papers is given on table 1.2.

Table 1.2 Sortation Literature Summary

Citation	Method	Scope	Performance Measure	Design Factors Concerned
Bozer and Sharp (1985)	simulation	S/PS-2	throughput	induction capacity, the number of lanes, the length of the lanes, the presence of a recirculation loop, and the control system
Bozer et al (1988)	simulation	S/PS-1	throughput	the number of sortation lanes; The distribution of items per order; order assignment rule; wave assignment
Choe (1990)	queueing model	OP & S/PS-1	batch sortation time	maximum zone pick time; batch profile, order lane assignment
Johnson and Lofgren (1994)	simulation	Sortation System	conveyor sortation system implementation at Hewlett-Packard ‘s new North American distribution center	
Johnson (1998)	analytical	S/PS-1	prove that incidental order lane assignment is a better strategy than any static fixed-assignment rule.	
Meller (1997)	analytical	S/PS-1	throughput	determine the optimal order-to-lane assignment for particular item arrival sequence
Wang (1997)	heuristic	Sortation System	presented an integrated approach that simultaneously considers the storage zones dimensions, storage zone layout and conveyor path problem for designing a conveyor sortation problem	
Johnson and Meller (2002)	analytical	Split-case sorting	throughput	manual induction configuration
Russell and Meller (2003)	descriptive model	Split-case sorting	throughput & cost	demand rates, labor rates, fixed and variable costs, order sizes, wave sizes and sorter capacity

Using simulation, Bozer and Sharp (1985) studied the S/PS-2 system. They analyzed the throughput performance of the system as a function of the induction capacity, the number of lanes, the length of the lanes, the presence of a recirculation loop, and the control system. Later, Bozer et al. (1988) used simulation to examine the S/PS-1

system. The paper examined the effects on productivity of the closed-loop conveyor sortation system S/PS-1, of the following factors:

(a) The number of sortation lanes

(b) The distribution of items per order.

(c) When and how orders are assigned to sortation lanes.

(d) When the next wave is released into the system.

It is conjectured in this paper that incidental order lane assignment is a better strategy than any static fixed-assignment rule. Johnson (1998) provides a proof of this result by developing analytical expressions for the sorting time operating under two families of sorting rules based on the assumption that the locations of the boxes in each order remain independent and uniformed distributed throughout the sorting process.

Choe (1990) has examined questions related to the design of both the order picking system and its relationship to the S/PS. He developed approximate queueing models of the order picking and S/PS subsystems to find the sortation time of a batch with a finite input stream of items and incorporate those models into an overall analysis of the effect of picking schemes. His model for S/PS subsystem is for S/PS-1 type. The time spending at A/M system is ignored or set as a constant delay. Heuristic approximation methods are used to get the mean and variance of the total sortation time of a wave.

Johnson and Lofgren (1994) provided a good description of a conveyor sortation system implementation at Hewlett-Packard Company. In designing the Hewlett-Packard‘s new North American distribution center, they decomposed the system into loosely independent subsections and built simulation model of each section in parallel.

Meller (1997) considered a two-level sortation system, where in the first level the items of each order are reconstituted, and in the second level the orders are positioned in lanes to enable correct truck loading sequences. In contrast with previous approaches

they consider the current item arrival sequence at the recirculation conveyor is known because items are scanned before entering the accumulation conveyor. They developed a mathematical programming model to determine the optimal order-to-lane assignment for particular item arrival sequence. They showed that improvements in throughput are obtained by optimally assigning orders to lanes on the basis of the item-arrival sequence.

Wang (1997) presented an integrated approach that simultaneously considers the storage zones dimensions, storage zone layout and conveyor path problem for designing a conveyor sortation problem. Constructive algorithms were developed for zone dimensioning, zone placement and shortest path based conveyor path determination. For conveyor operating parameters design, it presents a simulated annealing search algorithm with a discrete simulation embedded within the method as a performance evaluation tool. Its decision parameters are also on S/PS system.

Johnson and Meller (2002) developed analytical performance models for a Split-Case Sorting system. They assume the manual induction is the bottleneck of the system. By modeling the induction attempts as a Bernoulli process, they developed throughput models for different systems configurations. They have established many interesting insights into system design, such as the negative impact of interference for multiple inductors, that split configurations can significantly outperform side-by-side systems and that side-by-side inductors should be place from faster to slowest.

Russell and Meller (2003) develop descriptive and prescriptive models for the Split-Case Sorting system described in Johnson and Meller (2002) to aid in the selection of an order sortation system by comparing the throughput and cost of different system configurations for both automated and manual systems. Using some formula in Johnson and Meller (2002), their descriptive model takes demand rates, labor rates, fixed and variable costs, order sizes, wave sizes and sorter capacity as an instance of input. Based on this deterministic input, the number of employees, packing stations, and induction

stations required to meet demand is determined and total annualize system cost is calculated.

Besides these papers, a number of articles that mainly describe the operation of sortation system have appeared in trade publications. Such articles include those presented by Apple 1999, Wagner 1994, Stubbs 1980 and Veldsma 1993. They offer general rules-of-thumb and accepted practice guidelines in determining the rates and accumulation length of the conveyor system and also equipment selection.

1.4.2 Literature on General Conveyor Theory

Conveyor research is classified into design and analysis of simple versus complex conveyor systems. A simple conveyor system typically refers to configuration of a line or closed-loop or is composed of accumulating or non-accumulating but not both. On the other hand, complex conveyor system is composed of accumulating and non-accumulating segments with branches and merge intersections. The layouts of such systems are usually network-like.

Closed-loop Conveyor System Analysis

Past research on general conveyor systems has concentrated on production systems serviced by conveyor systems. In these systems, conveyors serve as finite buffers with a transportation function. Most of these researches are about the discrete–flow close-loop conveyor system and investigate the steady-state performance of conveyors. Table 1.3 summarizes the literature on closed-loop conveyor system analysis.

Table 1.3 Literatures on Closed-Loop Conveyor System Analysis

Citation	Problem Setting				Objective
	Discrete or Continuous Placement	Recirculation allowed?	Arrival and Service pattern	Number of Loading (L),Workstation (W) and Unloading stations (U)	
Kwo(1958)	D	N	Deterministic	Single L and Single U	Feasible input and output rate functions
Mayer (1960)	D	N	Stochastic	Multiple L and single U	Rate of unsuccessful loading
Morris (1962)	D	N	Stochastic	Multiple L, Multiple U	Rate of unsuccessful loading
Disney (1962,1963)	D	N	Poisson arrival, negative exponential service time	Single L, 2 W, 1 U (Ordered Access)	Queue length distribution at each work station
Muth (1972)	C	N	Deterministic time-varying	Single L and Single U	Establish conditions under which compatibility exists.
Elsayed (1977)	C	Y	Poisson input Exponential Service Time	Multiple L and Multiple U (Random Access)	Evaluate the effect of multiple inputs, storage and recirculation on overall system behavior
Sonderman (1982)	C	Y	Stochastic	Single L and single U	Approximate the output process at the unloading station
Pourbabai and Sonderman (1985)	C	Y	Stochastic	Single L, multiple U (random access)	Steady state probability
Bastani (1986)	C	Y	Deterministic	Multiple closed-loop conveyor system each having a single L and a single U	Recirculation times of products
Xue and Proth (1987)	D	N	Stochastic	Single L, Single U	Expected WIP level
Coffman (1988)	D	N	Stochastic	Single L and single W	Determine the proper distance separating the input and output points of a workstation.
Bastani (1988)	C	Y	Deterministic	Single L, Multiple U	Load recirculation times
Bastani (1990)	C	Y	Poisson input, Exponential service (with breakdown)	Single L, multiple U	Steady-state probabilities of system being in different operating states
Atmaca (1994)	D	N	Poisson input Exponential Service Time	Single L and multiple W and one U	Expected waiting time at loading station

Schmidt and Jackman (2000)	C	Y	Stochastic	Single L, multiple W and single U (ordered access)	Stead-state probability
Bozer (2004)	D	N	Stochastic	Multiple L, Multiple U	Expected waiting time at loading stations

These papers differ in four aspects of problem settings:

(1) Placement of material on conveyors. Continuous placement or Discrete placement?

(2) Whether recirculation is allowed?

(3) Material flow in and out of stations. Deterministic or Stochastic?

(4) Number of loading and unloading stations. In the case of multiple unloading stations, in which order the items access an unloading station, ordered access or random access?

Although the closed-loop conveyor may look like the recirculation system in the S/PS subsystem, the results on conveyor theory do not apply to S/PS subsystem for variety of reasons. First most of the existing models are based on conveyors with discrete carriers. Except for tilt-tray sorters, a majority of high-throughput sortation systems utilize belt and roller conveyors. More importantly, these existing studies assume that the item is removed from the conveyor by first available unloading station. In S/PS systems, not only is the item assigned or destined to a specific lane but also it is diverted to a specific lane automatically; that is, the worker does not have to be available to remove the item from the conveyor.

Complex Conveyor Network Analysis

Network flow model has long been proposed for complex conveyor network analysis. Ravindran, Foote and Williams (1988) applied a network flow model approach to redesign a conveyor system. The model was restricted to constant input flow rates and accumulation capability was not considered in the model.

Maxwell and Wilson (1981) developed a time-expanded network flow model for analyzing the flow in a dynamic material handling system with fixed paths. The “micro-models” for different conveyor type component are provided. In their model, if the travel time of an accumulation segment is more than one time slice, the node-arc structure of

micro-model is replicated for each time-slice. So the number of nodes and arcs in time-expand network will increase as the accumulation arc transit time increases, which is an undesirable feature of this approach.

Lin (1994) developed an approach using a time-and-intersection expanded (TIE) network flow model to optimize the operation parameters given a user defined objectives. Processing, assembly, grouping and batching operation at stations were considered in the model. A transformation methodology was used to convert a complex conveyor system into an abstract network. Micro models were designed for the conversion and for the representation of the geometric relationship, the time factor, and the physical characteristics of the conveyor system.

The common point of Maxwell and Lin‘s approaches is that they use the same abstraction of conveyor transportation function. It essentially models the conveyor transportation as a fixed time delay and also provides a storage buffer at the discharge end in the accumulation segment case. We derive a formal conveyor continuous model, “Delay and Stock Model”, based on this abstraction in chapter 4 where it is utilized in our A/M network parametric design model.

Two papers address the design and control problem of a conveyor network with merging configuration (CNMC). In such a system, several induction conveyor lines connect into the main conveyor line at consecutive places. Cargo is loaded at the up ends of the induction lines, transported into the mainline. Since the output of this CNMC could be the input to an induction line of another CNMC, several CNMCs can form a complicated network. In advanced system the merging operations of induction lines are under control to balance the throughput among induction lines. Performance of a CNMC is primarily measured by its main-line throughput and utilization. Artantes and Deng (1996) devised an algorithm (called QTM) based on queueing theory to design the system so that different induction lines can reach a balance while maintaining high throughput. The QTM can identify proper buffer sizes based on the number of induction lines, arrival

rates, conveyor speeds and parcel size etc. Jing, Kelton and Arantes (1998) use simulation to realize the logic in QTM and to analyze the behavior of CNMCs under various conditions. Their finding is that QTM is quick and conservative in finding a reasonably good initial design for CNMCs to reach a balanced throughput. The approximation error in QTM is small at high mainline speed but is large at low mainline speed and QTM approximation also deteriorates at high mainline utilization.

Zrnic and Cupric (1992) studied the material flow system in a high-bay warehouse. The system they dealt with consists of one or three devices for joining and dividing. They point out the capacity of the whole transportation system depends on the capacity of these joining and dividing devices where the possibilities for bottleneck occur. The objective of their research is to determine the accumulation positions. In the special case where there is only one joining device and the main flow and slave flow are all exponential, there is analytical formula to calculate the average number of items in system. Simulation modeling is the only choice for detail study of the behavior of systems that consist of more than one device for joining or dividing. These results were applied in the design of two new distribution centers in Belgrade.

Conveyor Simulation Modeling Research

Most simulation literature on conveyor system use commercial simulation software to study a specific system. Among the few papers concerning about the conveyor representation in simulation studies, Geinzer (1990) addressed the high volume conveyor modeling difficulties through a Federal Express distribution facility simulation application. The paper discusses alternative simulation modeling approach for high volume conveyor system. Several key issues, such as computer memory, run time and model accuracy were evaluated for each modeling approach. The first three approaches required the use of the conveyor constructs that are provided by most state-of-art simulation software. These options are eliminated because of the extreme length of the

conveyors and the number of cells that would have been required. In the fourth option, conveyor segment is modeled by a fixed delay. Obviously, this is an extremely approximate approach where the capacity of the conveyor and the contention for conveyor space is ignored. In the fifth option, a conveyor segment is modeled by a constrained resource at the entry point of the conveyor, in conjunction with a delay for the object to arrive at the destination point. The amount of time an object controls the entry resource determines the rate at which the objects can enter the conveyor, but accumulation and blocking are also still ignored in this modeling.

Demongodin and Prunet (1993) proposed an extension of Hybrid Petri Nets, Batches Petri Nets, to model hybrid production system with continuous transfer elements. It captures the accumulation as well as the delay characteristics of the conveyors with a mathematical formalism. In their model, the conveyor evolution is embedded in the Petri Net representation. They did not provide mathematical expressions to describe the relationships between input and output and other parameters of a conveyor segment. Thus, their approach is hard to extend or to use in deep level analyses, such as sensitivity analysis.

Literature Summary

We can summarize our literature review results into following observations:

1) Sortation system design is important. On one hand, the sortation system represents the heart of the operation of centralized DC, efficient sorting and shipping of orders is one key to survival in the hypercompetitive environment of retailing. On the other hand, construction cost of an automated DC varies widely in the 10-50 million dollar range, and hourly operating expenses often exceeding 1000 dollars (Johnson, 1998). Cost-effective sortation systems are required to enable today's just-in-time distribution service.

2) Past research on sortation system focuses on type 1 S/PS subsystem design. Research on Accumulation/Merge system design is limited to simulation study or aggregated analysis in which the whole A/M subsystem was treated as a large static buffer. However the design and operation of A/M system is important due to its high cost, large footprint and serving as the bloodline in the system.

3) There is rich body on general conveyor theory, but they mainly focus on the production system served by a close-loop simple conveyor system.

4) Stochastic conveyor network analysis has been restricted to very simple structure with one or two junction nodes.

5) In large scale conveyor system the discrete item-level of conveyor representation can be too slow for simulation and leads to intractable models for design optimization. However, the continuous delay and stock abstraction essentially ignores the spatial material flow evolution. There has been no study done to evaluate the accuracy of this representation. Batches Petri Net is another effort along this line of research. However, there is a lack of rigorous mathematical abstraction of the conveyor continuous model which limits its usage beyond simple conveyor system simulation.

1.5 Research Objectives

Based on our literature summary, we study the sortation system from performance evaluation and design perspectives. Because of the prospects of reducing computational effort in simulation and leading to tractable models in optimization, we take continuous modeling approach throughout the study.

First, we want to create a rigorous and uniform mathematical framework to present various continuous conveyor models. It will serve as a concrete base to develop simulation and optimization models and to conduct sensitivity analysis. We also want to investigate differences on representation accuracy of these models and analyze the rationale of generating this difference.

In the performance evaluation, we attempt to develop a fast fluid simulation methodology suited for high volume conveyor system simulation so that the simulation model run time and construction time can be greatly reduced while maintaining satisfactory simulation accuracy.

In the design optimization, we focus on rate determination and accumulation design of A/M system of a type 2 sortation system. Since direct stochastic analytical of A/M subsystems is intractable due to queueing characteristics of the system such as dynamic arrival and transient overload conditions presenting in wave picking environment, we propose to integrate the stochastic optimization techniques with deterministic system evaluation tool or simulation in the process of searching near optimal design with minimum cost and performance bounds.

The remainder of this thesis is organized in two main parts followed by conclusions. The first main part is in Chapter 2. It introduces an "exact" continuous conveyor model –"Batch on Conveyor Model". Based on this model we developed a fluid simulation methodology for high volume conveyor system's transient and steady-state simulation. The second main part is in Chapters 3, 4 and 5. All three chapters address the A/M system parametric optimization. Chapter3 describes the general optimization problem modeling and simulation optimization solution framework. Chapter 4 and 5 discuss the detail solution techniques of combining this framework with two deterministic performance evaluation tools. In chapter 4, "Delay and Stock Model" is adopted. We model the performance evaluation problem into a class of dynamic network flow problem and we obtain the gradient of performance metric with respect to design parameters from the dual problem. In chapter 5, we adopt the "Batch on Conveyor Model" and employ fluid simulation as performance evaluation tool. We investigate the approach on a two-segment tandem conveyor system which is the build block of more complex system.

CHAPTER 2

FLUID SIMULATION MODEL FOR CONVEYOR SYSTEM

We present a fluid approach for simulating high volume conveyor transportation system in this chapter. The chapter is organized as follows: The motivation of this approach is discussed in section 2.1. The continuous flow model for a single segment operation is presented in section 2.2. We use an extension of Batches Petri Net as basic architecture to present the conveyor network structure. Section 2.3 defines the Petri Net model and its evolution rules. Section 2.4 shows the Petri Net model can be easily transformed into a discrete event fluid simulator. Performance measure collection is discussed in section 2.5. A case example and some computational experiment results will be discussed in section 2.6 and 2.7 respectively.

2.1 Motivation

We address the motivation of fluid simulation approach from the following perspectives.

2.1.1 Issues of Current Conveyor Simulation Approach

The conveyor simulation constructs used in the state-of-art simulation software, such as Arena, are cell-based representation of conveyor systems at item-level. Conveyors are divided into cells and system tracks whether each cell is occupied or not. Each time a package moves from one cell to another an event is triggered. First, we observe that this representation inherently introduces errors in some common scenarios to model the continuous movement type conveyor, such as chain conveyor or belt conveyor. This is first illustrated through the following example. Suppose we have two belt conveyors connected to each other as shown in Fig 2.1. The first conveyor can move at a speed v_1 = 50 length units per minute and the second one has higher speed than the first one which is v_2 =70 length units per minute. The item is 1 length unit long. We have a

constant input of 50 items per minute. Intuitively, we should expect no blocking since conveyor transportation capacity is larger than the input rate.

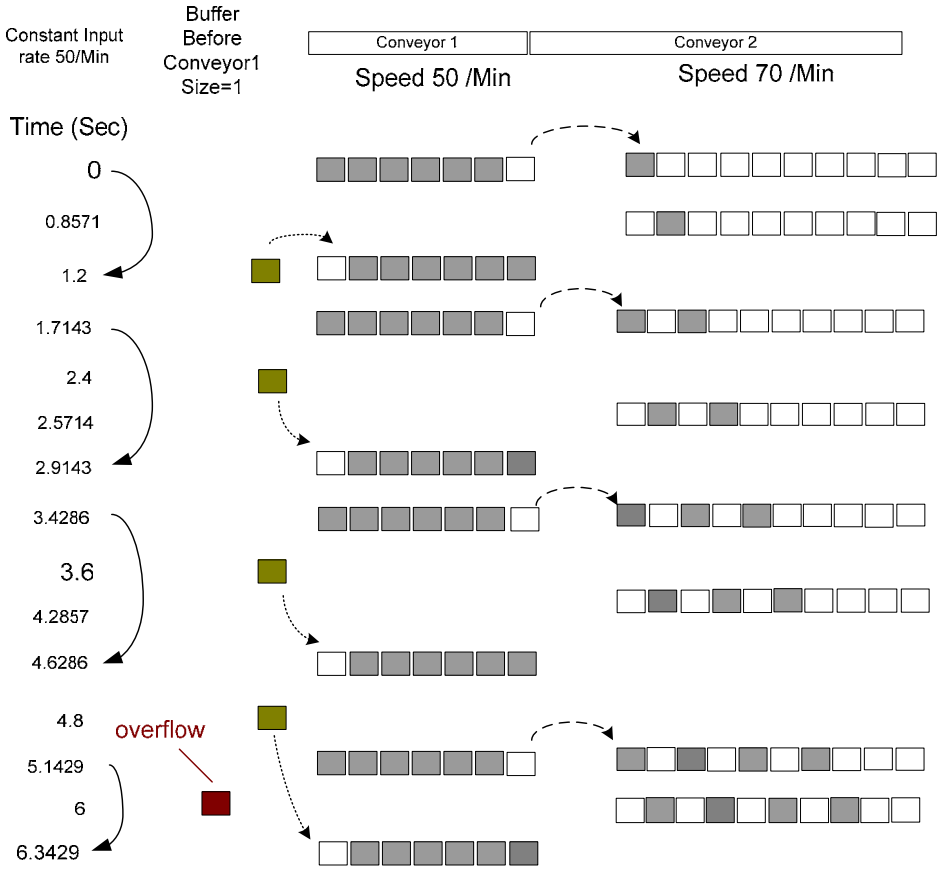


Figure 2.1 Example of Inaccuracy of Cell-Based Conveyor Simulation

However, if we simulate this simple system using cell size *cl* = 1 length unit in Arena we would see overflows. Figure 2.1 illustrates the dynamics of the simulation progression. We observe that there is a blank cell between two consecutive items on the conveyor 2 material stream. In fact, the output capacity of conveyor 2 can only achieve 70, 70/2, 70/3.., discrete values which correspond to the cases that there are 0, 1, 2.., blank cells between two items in the conveyor 2 material flow. The system throughput is the minimum of output capacity of conveyor 2 and the output rate from conveyor 1. This leads to a reduced system throughput 35 items per minute in this case. So we could see first overflow after six minutes. Actually, we can derive a generic form for the system

achievable throughput in this example as $\frac{1}{\left\lceil \frac{v_2}{v_1} * \frac{1}{cl} \right\rceil * cl} * v_2$. We can see that system achievable throughput depends on the cell length and ratio of two conveyor speeds. Figure 2.2 listed the achieved system throughputs with different cell lengths when v_1 equals 50 and v_2 equals 70.

Cell Length (length unit)	Output Rate (Item/Min)
1	**35**
1/2	**46.7**
1/3	**42**
1/4	**46.7**
1/5	**50**
1/6	**46.7**
1/7	**49**
1/8	**46.7**
1/9	**48.5**
1/10	**50**
1/11	**48.1**
1/12	**49.4**
1/13	**47.9**
1/14	**49.0**
1/15	**50**

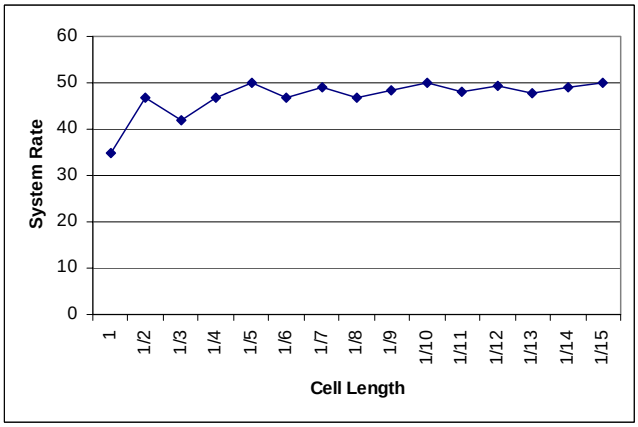


Figure 2.2 System Throughput vs. Cell Length

In this example, system throughput can achieve 50 if v_2 equals 50. Higher conveyor 2 speed leads to a lower system throughput. Scenarios like this are common in a sortation conveyor network. For example, a suitable gap between items is required before inducting them to the sorter. A much more common method is to pull a gap across a speed change of two connecting conveyors. A merging junction can be another example where the downstream conveyor usually has higher speed then upstream conveyors. One way to overcome this error is to use smaller cell size. In the above example, system throughput rises to 46.7 when cell length is 0.5 length unit. It can reach 50 so that no blocking would occur if cell length is 0.2 length unit. In general, simulation results will converge to the real performance measure as cell length becomes smaller and smaller. In high volume transportation conveyor system (HVTCS), hundreds or thousands of items move along tens of kilometers of complex conveyor system; the trigger events are very

frequent. The small cell size would elongate the execution time of already lengthy simulation execution time and the simulation becomes less useful.

2.1.2 Flow Characteristics of High Volume Transportation Conveyor System

Closer observation of HVTCS reveals that different behaviors in the system are on different time scales. For example, the time for item interarrival is in seconds, while the time for rate change in loading or unloading is usually in minutes or tens of minutes. Uncertainty and variability exist on all time scales, but the ones associated with the longer time scale are significant and dominant for system design purpose.

This type of system can be modeled as a network operating in a slowly changing environment - the rate of change in environment (start unloading a truck) is much less frequent than the rate of changes of the system state (loading an item on the conveyor). The ratio of rates is at least an order of magnitude. We describe the longer time scale behavior by states of environment. The change in the state of environment triggers the change of interarrival rate or routing control parameters. In certain environment states, the system appears unstable in the sense that arriving rate exceeds the service rate. However, conveyors can provide the buffering and stability. The queueing network analysis of such system is touched by Choudhury and Mendelbarm et al. (1997) and Chang (2004). They show that the queueing systems in a random slowly changing environment can be approximated by stochastic fluid model. This motivates us to use fluid simulation to analyze high volume conveyor network in hope to reduce the computational burden of the simulation.

2.1.3 Related Research

Our modeling approach is rooted in two lines of research: the fluid simulation in telecom domain and the Batches Petri Nets in manufacturing domain to model the continuous transfer equipment.

Fluid simulation has been developed in computer network paradigm to cope with today's network growth in size and complexity (Kesidis et al., 1996; Kumaran et al., 1998). A fluid simulator models an Asynchronous Transfer Mode (ATM) network as some fluid sources followed by a set of fluid bandwidth schedulers linked with constant propagation delay. The fluid emitting rates from these sources are modeled as piece-wise constant functions. At any time a set of input rates, a set of output rates and current buffer content can completely describe the status of a bandwidth switch. A fluid-based scheduling policy is used at each switch to determine the output rates of different sources from input rates. Comparison between efficiencies of fluid simulation and packet-level simulation has been done by several researchers (Liu et al., 1999; Liu et al., 2001) and their primary conclusion is that the fluid simulation will generally outperform packet-level simulation for the simple network. As the network size and complexity grow, the fluid simulation suffers from the so-called "ripple effect" which makes the fluid simulation less efficient. Some corresponding approaches also have been proposed to overcome it.

Batches Petri Nets, an extension of Hybrid Petri Nets, has been developed (Demongodin and Prunet, 1993) to model hybrid production system with continuous transfer elements. It captures the accumulation as well as the delay characteristics of the conveyors with a mathematical formalism. Their model does not include mathematical expressions to describe the relationships between input and output and other parameters of a conveyor segment. There are three extensions of Batches Petri Nets so far: Controlled BPN (Audry and Prunet, 1995), Colored BPN (Caradec and Prunet, 1997), and Generalized BPN (Demongodin, 1999). However, they all lack the ability to model more complicated conveyor network structures, such as merging, diverging and stochastic elements.

In this chapter, we develop the fluid simulation methodology for conveyor network simulation. Like the fluid simulation in telecom network, we use loading and

unloading rates to describe the arrival and departure. Random arrivals and interruptions are captured by stochastic model of rate changing. As the result of these continuous stochastic arrival trajectories, batches of items with different length, density and position are generated, circulated and merged within the network. We first present a continuous conveyor model that describes batch evolution. We then use an extension of Batches Petri Nets as a basic architecture to describe the interaction among system elements. We call the new model Stochastic Batches Petri Nets (S-BPN) since stochastic transition is added to the model.

2.2 Continuous Flow Conveyor Model

In this section, we present a continuous flow conveyor model that will be used in the fluid simulation and later sample path analysis. We called it "Batch on Conveyor Model" since we describe the aggregate items as continuous batches. We first introduce the general model setting and then discuss the Batch on Conveyor Model for two types of conveyor segment separately.

2.2.1 Batch on Conveyor Model Setting

Conveyor segments, either accumulative or non-accumulative, are building blocks of transportation conveyor network. A conveyor segment is a one-directional transportation and buffering device, which receives items from one end and sends them to another end and preserves the sequence of items during transportation (see Figure 2.3). We call the two ends ENTER and EXIT respectively. A conveyor segment associates with four design parameters:

v	driving speed, length unit/time unit;
l	length, length unit;
d	maximum density, items/length unit ;

$c \in \{'A','N'\}$ conveyor segment type, 'A': accumulation and 'N' : non-accumulation.

As mentioned in the motivation section, we will use stochastic fluid processes to model the input and output function in fluid simulation. We also assume these functions have piece-wise constant sample paths, and discrete time events in fluid simulation constitute the rate changes in these sample paths. The characteristic quantities to describe the input and output flow of a segment at an instant *t* are defined as follows:

$\mathcal{I}(t)$: input flow rate to the ENTER side of the segment; this could be external arrival rate or the rate at which items can be transferred from upstream conveyors provided that no blocking occurs in this segment;

$\mathcal{O}(t)$: discharge capacity at the EXIT end of the segment; this is the maximum rate at which items can be removed from this segment;

$I(t)$: effective in-flow rate; this is the real achieved rate at which items will be loaded to this segment;

$O(t)$: effective out-flow rate; this is the real achieved rate at which items will be unloaded from this segment;

$F(t)$: overflow rate, $F(t) = \mathcal{I}(t) - I(t)$.

Since overflow is undesirable, maximum transportation capacity of a conveyor segment, *vd*, should be designed higher than the input flow rate or,

$$\mathcal{I}(t) \le vd \ . \qquad (2\text{-}1)$$

The piece-wise constant continuous input forms batches on the segment and batches are transported to the EXIT end of the segment where it is transferred to output flow. A *batch* represents a set of items with the same density of repartition on a segment.

At a fixed instant *t*, we denote the ordered sequence of batches on a segment as $\left\{B^1(t), B^2(t), .., B^K(t)\right\}$; the batch index increases from EXIT end to ENTER end. We also refer this ordered sequence of batches as the *batch condition* at time *t*. The *k*th batch in

this sequence is characterized by $B^k(t):[l^k(t),d^k(t),x^k(t)]$, where [ ] denotes a list of characteristics which have meanings as follows:

k: sequence number of the batch;

$l^k(t)$: length of batch;

$d^k(t)$: density ;

$x^k(t)$: beginning position of the batch. The position is measured such that when the beginning of a batch is at EXIT, its position is *l*.

An *output batch* $B_o(t)$ is the first batch $B^1(t)$ if its beginning position is equal to the length of the segment; otherwise we set it to NULL. Output batch is the batch that has reached the EXIT end of the conveyor and is being transferred or is ready to be transferred. We use $d_o(t)$ to denote the density of the output batch; if there is no output batch we let $d_o(t)=0$. An *input batch* $B_i(t)$ is the batch that will be formed during a time interval starting at time *t*. We use $d_i(t)$ to denote the density of the input batch; if there is no input batch that will be formed because of zero in-flow rate, we let $d_i(t)=0$.

The above definitions and notations are illustrated in the following figure, which is a snapshot of a conveyor segment at time *t*.

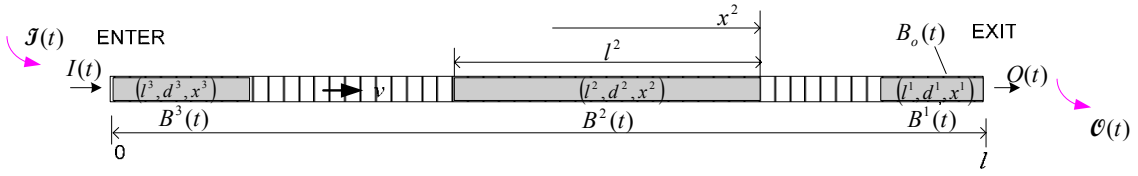


Figure 2.3 A Snapshot of a Segment at Time t

We omit the segment index in the notations used in this section. When a segment index is needed, we add it as the first or second subscript. To summarize, we use the notation convention $a^d_{b,c}$, where *a* can be *B* which stands for a batch or some characteristic quantities like *l* ,*v*, *d*; *b* indicates either in(i) or out(o); *d* is for batch count; *c* is for segment, when clear from context, we will drop it. For example, d_1 is the

maximum density of conveyor segment 1, $d_{o,1}(t)$ is the output batch density of segment 1 at time *t* and $d_2^2(t)$ is the density of second batch on segment 2 at time *t*.

The flow of Batch on Conveyor Model is illustrated in Figure 2.4.

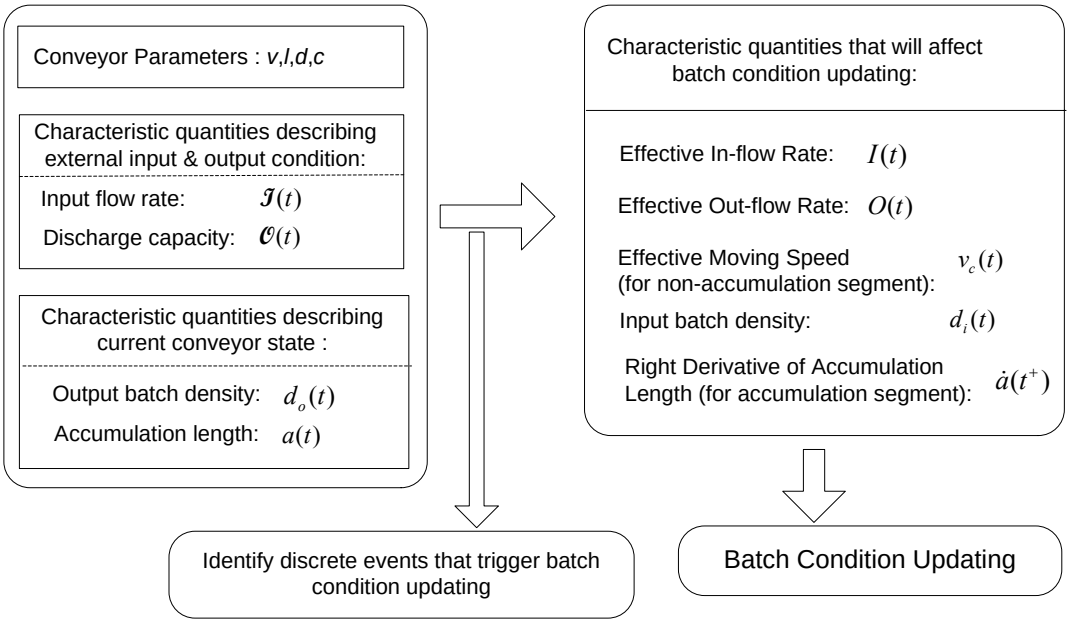


Figure 2.4 Flow of Batch on Conveyor Model

The main task of Batch on Conveyor Model is to develop mathematical relationships between two sets of characteristic quantities. The first set of quantities include conveyor design parameters, characteristic quantities that describe the external input and output conditions and characteristic quantities that describe the current conveyor state. The other set of quantities are ones that will be used to guide the batch condition updating. The discrete event types that will trigger batch condition updating are also determined by these relationships. In the following, we show how to develop these relationships and how to update the batch condition of two types of conveyor segments separately.

2.2.2 Batch on Conveyor Model of Non-Accumulation Conveyor Segment

In the item-level description, non-accumulation conveyor moves in a stop-and-go format. In fluid simulation, since we describe the aggregate items as continuous batches,

we consider the non-accumulation conveyor segment is moving continuously at a certain speed at any instant. We call it *effective moving speed* denoted as $v_c(t)$. It is equal to the average moving speed in a time interval Δt in the item-level context. This is illustrated through Fig 2.5.

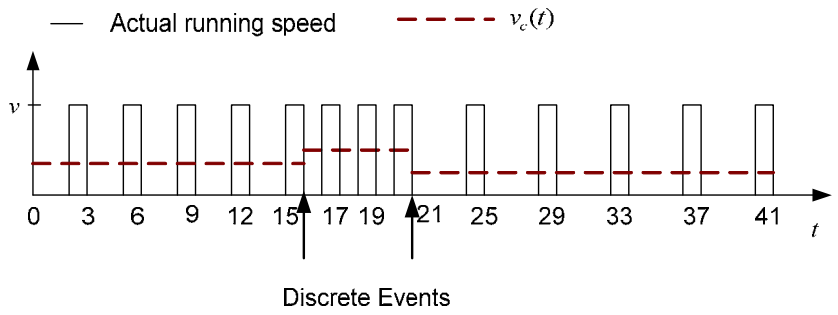


Figure 2.5 Illustration of Effective Speed

During time 0-15, conveyor stops for two time units between every two moves; each move lasts for one time unit. So the effective speed $v_c(t)$ equals $\frac{1}{3}v$. During interval 15-21, conveyor runs in alternate time units and $v_c(t)$ equals $\frac{1}{2}v$. As we can see, $v_c(t)$ is a piece-wise constant function. The jumping points of the function correspond to some discrete event times which we summarize below.

Batch condition of a non-accumulation segment is updated in discrete times on which three types of events occur:

- $\mathcal{I}(t)$ change
- $\mathcal{O}(t)$ change
- Output batch density $d_o(t)$ change

In the following discussion, we show how conveyor state updates during a time period $[t, t+\Delta t]$ in which $\mathcal{I}(t), \mathcal{O}(t)$ and $d_o(t)$ remain constant. Figure 2.6 depicts the relationships among different types of characteristic quantities. The text that follows provides detail derivation of these relationships.

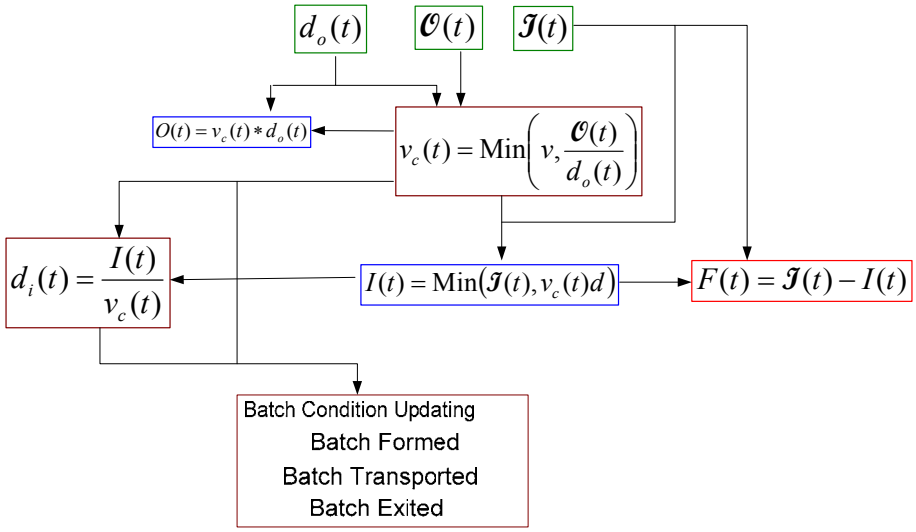


Figure 2.6 Batch on Conveyor Model of Non-Accumulation Segment

First the segment will always move at its driving speed as long as the discharge rate allows, i.e., $v_c(t)d_o(t) \le \mathcal{O}(t)$. So the effective moving speed can be expressed as

$$v_c(t) = \text{Min}\left(v, \frac{\mathcal{O}(t)}{d_o(t)}\right). \tag{2-2}$$

Once $v_c(t)$ is determined, the effective admitting rate of the segment is $v_c(t)d$ since d is the maximum density of the segment. So the effective in-flow rate $I(t)$ is the minimum between the input flow rate and the maximum admitting rate which expressed as

$$I(t) = \text{Min}(\mathcal{I}(t), v_c(t)d) = \text{Min}\left(\mathcal{I}(t), \frac{\mathcal{O}(t)}{d_o(t)}d\right). \tag{2-3}$$

The equivalence between the two expressions is established under the assumption stated in (2-1). Based on this, the overflow rate is

$$F(t) = \mathcal{I}(t) - I(t) = \text{Max}\left(0, \mathcal{I}(t) - \frac{\mathcal{O}(t)}{d_o(t)}d\right). \tag{2-4}$$

Effective out-flow rate is described as

$$O(t) = v_c(t) * d_o(t) = \text{Min}(\mathcal{O}(t), vd_o(t)). \tag{2-5}$$

Input batch density equals the effective in-flow rate divided by effective conveyor moving speed. Using equations (2-2) and (2-3) we can derive input batch density depending on $\mathcal{I}(t)$, $\mathcal{O}(t)$, $d_o(t)$, v and d as

$$d_i(t) = \frac{I(t)}{v_c(t)} = \begin{cases} \dfrac{\mathcal{I}(t)}{v} & \text{if } \dfrac{\mathcal{O}(t)}{d_o(t)} \geq v \\ \dfrac{\mathcal{I}(t) d_o(t)}{\mathcal{O}(t)} & \text{if } \dfrac{\mathcal{I}(t)}{d} < \dfrac{\mathcal{O}(t)}{d_o(t)} < v \\ d & \text{if } \dfrac{\mathcal{O}(t)}{d_o(t)} \leq \dfrac{\mathcal{I}(t)}{d} \end{cases}. \qquad (2\text{-}6)$$

Batches evolve during a time interval $[t, t+\Delta t]$ in the following three ways.

Input Batch Formed

An input batch is formed if $I(t) > 0$ during $[t, t+\Delta t]$. The input batch is characterized by $B_i(t) : \left[\Delta t v_c(t), d_i(t), \Delta t v_c(t)\right]$. It is possible for the input batch to merge with the batch $B^K(t) : [l^K(t), d^K(t), x^K(t)]$ whose back end is just the ENTER side of the segment and has the same density as the input batch, i.e., $l^K(t) = x^K(t)$ and $d^K(t) = d_i(t)$.

Batch Transported

If $v_c(t) > 0$, a batch $B^n(t) : [l^n(t), d^n(t), x^n(t)]$ other than the output batch is transported forward into a batch $B^m(t+\Delta t) : [l^n(t), d^n(t), x^n(t) + \Delta t v_c(t)]$. The batch sequence number could possibly change because the output batch might exit from the segment.

Output Batch Consumed

If $O(t) > 0$, output batch $B_o(t) : [l_o(t), d_o(t), l]$ with $d_o(t) > 0$ will become shorter after a delay Δt as $B_o(t+\Delta t) : [l_o(t) - \Delta t v_c(t), d_o(t), l]$.

2.2.3 Batch on Conveyor Model of Accumulation Conveyor Segment

When there is enough output capacity, accumulation conveyor segment just behaves like the non-accumulation conveyor segment. Otherwise, accumulating starts from the EXIT end of the segment. An output batch of maximum density *d* is formed gradually. We call the end of the maximum density output batch the *accumulation interface*. The length of the maximum density accumulation is denoted as $a(t)$. We denote the density of the batch on the other side of the accumulation interface as $d_a(t)$ which is specified for difference cases as

$$d_a(t) = \begin{cases} d_o(t) & \text{if } a(t) = 0 \\ d^2(t) & \text{if } 0 < a(t) < l \text{ and } x^2(t) + a(t) = l \\ 0 & \text{if } 0 < a(t) < l \text{ and } x^2(t) + a(t) < l \\ \dfrac{\mathcal{I}(t)}{v} & \text{if } a(t) = l \end{cases} . \qquad (2\text{-}7)$$

In the accumulative case, the discrete events that trigger the batch condition updating consist of

- Rate change in $\mathcal{I}(t)$
- Rate change in $\mathcal{O}(t)$
- $d_a(t)$ change
- $a(t) = 0$ or $a(t) = l$

Figure 2.7 summarizes the relationship among quantities in Batch on Conveyor Model of accumulation segment. The text that follows provides detail derivation of these relationships.

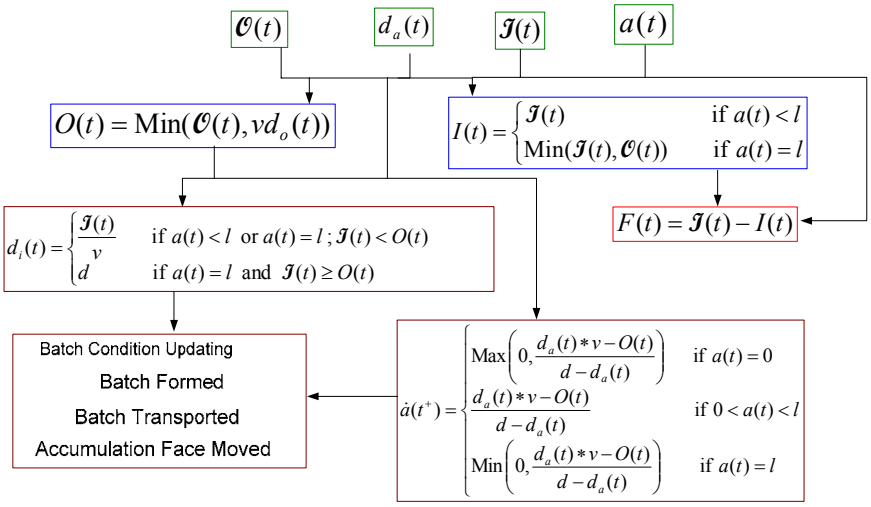


Figure 2.7 Batch on Conveyor Model of Accumulation Segment

When accumulation is not full, accumulation segment can always admit the current input rate; in the full accumulation case, conveyor segment becomes a single batch with maximum density, real admitting rate depends on the discharge capacity. We can write effective in-flow rate as

$$I(t) = \begin{cases} \mathcal{I}(t) & \text{if } a(t) < l \\ \text{Min}(\mathcal{I}(t), \mathcal{O}(t)) & \text{if } a(t) = l \end{cases}. \tag{2-8}$$

Effective out-flow rate is the minimum between attempting out-flow rate $vd_o(t)$ and discharge capacity,

$$O(t) = \text{Min}(\mathcal{O}(t), vd_o(t)). \tag{2-9}$$

Accumulation segment can be considered always moving at its driving speed provided that full accumulation has not reached. So input density is

$$d_i(t) = \begin{cases} \frac{\mathcal{I}(t)}{v} & \text{if } a(t) < l \text{ or } a(t) = l; \mathcal{I}(t) < O(t) \\ d & \text{if } a(t) = l \text{ and } \mathcal{I}(t) \geq O(t) \end{cases}. \tag{2-10}$$

$a(t)$ is a continuous function but not differentiable. There are points where the left derivative and right derivative are not equal. We only need to know the right

derivative of $a(t)$ for batch condition updating during the interval $[t, t+\Delta t]$. It is captured in the following theorem.

Theorem 2-1: $$\dot{a}(t^+) = \begin{cases} \text{Max}\left(0, \dfrac{d_a(t) * v - O(t)}{d - d_a(t)}\right) & \text{if } a(t) = 0 \\ \dfrac{d_a(t) * v - O(t)}{d - d_a(t)} & \text{if } 0 < a(t) < l \\ \text{Min}\left(0, \dfrac{d_a(t) * v - O(t)}{d - d_a(t)}\right) & \text{if } a(t) = l \end{cases} \quad (2\text{-}11)$$

Proof. Figure 2.8 illustrates the change on the interval $[t, t+\Delta t]$ in three cases.

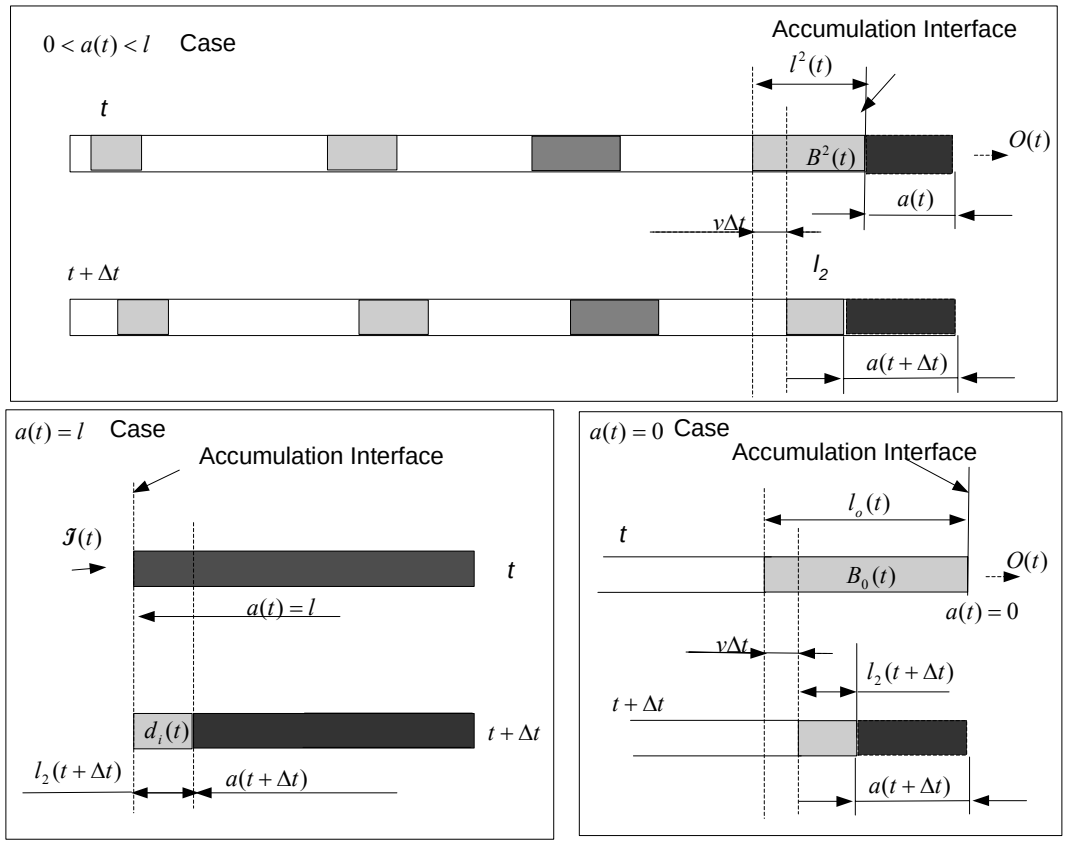


Figure 2.8 Illustration of Theorem 2-1 Proof

When $0 < a(t) < l$, we have $d_a(t) = d^2(t)$, if the second batch already reached at the accumulation interface, i.e., $x^2(t) + a(t) = l$. Using material conservation law, we have

$$l^2(t)d^2(t) + a(t)d = l^2(t+\Delta t)d^2(t) + a(t+\Delta t)d + O(t)\Delta t\,,$$

$l^2(t+\Delta t)$ can be expressed as $l^2(t+\Delta t) = l^2(t) + a(t) - v\Delta t - a(t+\Delta t)$, plugging this into the above equation and rearranging the items, we get

$$\dot{a}(t^+) = \lim_{\Delta t\to 0}\frac{a(t+\Delta t)-a(t)}{\Delta t} = \frac{d^2(t)v - O(t)}{d - d^2(t)} = \frac{d_a(t)v - O(t)}{d - d_a(t)}.$$

In the case $x^2(t) + a(t) < l$, we have $d_a(t) = 0$. The above equation is still valid since $\dot{a}(t^+)$ will indeed be zero in the case that there is no batch on the left hand side of accumulation interface.

In case $a(t) = 0$, we have $d_a(t) = d_o(t)$. If $d_o(t)v \le O(t)$, we have enough discharge capacity so $a(t)$ will remain 0, i.e., $\dot{a}(t^+) = 0$. Otherwise we have material conservation equation

$$l_o(t)d_o(t) = l^2(t+\Delta t)d_o(t) + a(t+\Delta t)d + O(t)\Delta t\,.$$

Substitute $l^2(t+\Delta t)$ with $l_o(t) - a(t+\Delta t) - v\Delta t$ in the above equation, we get

$$\dot{a}(t^+) = \lim_{\Delta t\to 0}\frac{a(t+\Delta t)-0}{\Delta t} = \frac{d_o(t)v - O(t)}{d - d_o(t)} = \frac{d_a(t)v - O(t)}{d - d_a(t)}.$$

In case $a(t) = l$, we have $d_a(t) = \dfrac{\mathcal{I}(t)}{v}$. If $\mathcal{I}(t) \ge O(t)$, the discharge capacity is lower than input rate. Therefore, $a(t)$ will remain at *l*, i.e., $\dot{a}(t^+) = 0$. Otherwise we have the following material conservation equation:

$$\mathcal{I}(t)\Delta t + dl = l^2(t+\Delta t)d_i(t) + a(t+\Delta t)d + O(t)\Delta t\,.$$

Substitute $l^2(t+\Delta t)$ with $l - a(t+\Delta t)$ and $d_i(t)$ with $\dfrac{\mathcal{I}(t)}{v}$ in the above equation, we get

$$\dot{a}(t^+) = \lim_{\Delta t\to 0}\frac{a(t+\Delta t)-l}{\Delta t} = \frac{\mathcal{I}(t) - O(t)}{d - \dfrac{\mathcal{I}(t)}{v}} = \frac{d_a(t)v - O(t)}{d - d_a(t)}.$$ □

In order to keep $\dot{a}(t^+)$ finite, for accumulation segment, condition (2-1) $\mathcal{I}(t) \le vd$ becomes

$$\mathcal{I}(t) < vd \ . \qquad (2\text{-}12)$$

The evolution of batches is abstracted into three moves as in non-accumulation conveyors. All the batches that have not reached the accumulation interface move at the conveyor driving speed *v*; only the batch that has contact with accumulation interface changes batch length as a result of moving of accumulation interface.

Input Batch Formed

When $a(t) \neq l$, an input batch is formed if $I(t) > 0$ during $[t, t+\Delta t]$. The input batch is characterized by $B_i(t) : [\Delta tv, d_i(t), \Delta tv]$. It is possible for the input batch to merge with the batch $B^K(t) : [l^K(t), d^K(t), x^K(t)]$ whose back end is just the ENTER side of segment and has the same density as the input batch, i.e., $l^K(t) = x^K(t)$ and $d^K(t) = d_i(t)$.

Batch Transported

A batch $B^n(t) : [l^n(t), d^n(t), x^n(t)]$ with $x^n(t) < l - a(t)$ is transported forward at speed *v* and becomes $B^m(t+\Delta t) : [l^m(t), d^m(t), x^m(t) + \Delta tv]$.

Accumulation Face Moved

For known $\dot{a}(t^+)$, the characteristics of the batches at the both sides of the accumulation interface can be derived from the moving of accumulation interface.

Batch on Conveyor Model addresses the dynamics of the continuous flow of two types of basic conveyor segments. For the conveyor network system, we need an infrastructure to model the connection and interaction among segments and interaction between a segment and external elements. We developed an extension of Batches Petri Nets for this purpose in the next section.

2.3 Stochastic Batches Petri Nets (S-BPN)

S-BPN was tailored from original BPN definition presented by Demongolin and Prunet (1993) to fit a fluid simulation framework.

2.3.1 Definition

A S-BPN is defined by a structure $N = (P, T, \text{Pre}, \text{Post}, C, \pi, mc, dc, M_0)$. The elements in the tuple are defined in the following.

1) $P = P_d \cup P_c \cup P_b$ is a set of places that are partitioned into a set of discrete places P_d, a set of continuous places P_c and a set of batch places P_b. Batch Places are used to model conveyor segments; continuous places are used to model physical buffers or virtual sources/sinks; discrete places are used in representing input models and control logics.

2) $T = T_d \cup T_b$ is a set of transitions that are partitioned into a set of discrete transitions T_d and a set of batch transitions T_b. The set of discrete transitions $T_d = T_I \cup T_D \cup T_S$ is further partitioned into a set of immediate transitions T_I, a set of deterministic timed transitions T_D and a set of stochastic timed transitions T_S. The set of all timed transition is denoted as $T_t = T_D \cup T_S$. There is a probability distribution function associated with each stochastic timed transition to specify the firing delay. It returns a random sample every time when the transition becomes enabled. There is a maximum firing flow (MFF) function $\Phi_j(t)$ expressed in items/unit time associated with each batch transition $t_j \in T_b$ which represents the handling rate, the arrival rate, and so on. An instantaneous firing flow (IFF) $\phi_j(t)$ is also associated with every batch transition having $0 \le \phi_j(t) \le \Phi_j(t)$.

3) $\text{Pre}(p_i, t_j)$ is a function defining the weight of arc from a place *i* to a transition *j* and $\text{Post}(p_i, t_j)$ is a function defining the weight of arc from a transition *j* to a place *i*.

Weights represent different meanings depending on type of place and transition. Table 2.2 gives weight condition definition for S-BPN.

4) *C* is the "characteristic function". It is associated with every continuous place a maximum buffer capacity, i.e., $C(p_i) = b_i$, if $p_i \in P_c$. It is associated with every batch place, the characteristics of underlying conveyor segment, i.e., $C(p_i) = \{c_i, l_i, v_i, d_i, Cap(i)\}$, where c_i, l_i, v_i, d_i are type, length, speed and maximum density respectively. $Cap(i) = \{Cap_1(i),..., Cap_n(i)\}$ is a finite set of sensors associated with the batch place p_i.

Sensors are used to model some control logics that depend on current state of a conveyor segment, such as whether there is positive out-flow in the segment or whether current accumulation exceeds 75% of the segment length. We define three types of sensors: accumulation sensor, detection sensor and empty sensor. Each sensor is defined by its type and the installation position on the conveyor segment. The weight of an arc linking a batch place to a discrete transition is a specific sensor of this batch place. We also call it a *sensor arc*. The sensor value on this arc will participate on deciding the enable condition of this transition. Mathematically, we define sensors and sensor value functions as follows.

$Cap_j(i) = \{t_j, x_j\}$ is the sensor *j* associated with the batch place p_i with:

x_j is the position of the sensor ($x_j \le l_i$)

t_j is the type of the sensor and belongs to the set {a, d, e}, that is {accumulation sensor, detection sensor, empty sensor}

Sensor value function is defined for accumulation sensor as: $V(Cap_j(i)) = 1$ if the accumulation passed the sensor position and $V(Cap_j(i)) = 0$ otherwise.

Sensor value function is defined for detective sensor as: $V(Cap_j(i)) = 1$ if there is a batch at the sensor position and $V(Cap_j(i)) = 0$ otherwise.

For empty sensor, it is defined as: $V(Cap_j(i)) = 1$ if there is no output batch in the batch place and $V(Cap_j(i)) = 0$ otherwise.

5) $\pi : T_d \to \mathbf{N}$ is the priority function. Timed transitions have priority level 0, immediate transitions have a priority level larger than zero. Highest numbers have highest priority. These values are used to control the sequence of firing in a marking with more than one enabled discrete transitions.

Before we define the elements mc and dc, we need to introduce the concepts of pre/post place, pre/post transition and merge/diverge place.

We call p_i a *pre-place* of transition t_j if $\mathrm{Pre}(p_i, t_j)$ is not null; p_i is a post-place of transition t_j if $\mathrm{Post}(p_i, t_j)$ is not null. We call t_j a *pre-transition* of place p_i if $\mathrm{Post}(p_i, t_j)$ is not null; t_j is a *post-transition* of place p_i if $\mathrm{Pre}(p_i, t_j)$ is not null.

We call ${}^\circ t_j$ ($t_j{}^\circ$) the set of pre (post) places of a transition t_j and ${}^\circ p_i$ ($p_i{}^\circ$) the set of pre (post) transitions of a place p_i.

We call a batch place p_i a *merge batch place* if it can possibly have more than one enabled pre batch-transitions at some time during the simulation run. We note P_{MB} the set of all merge batch places.

We call a batch place p_i a *diverge batch place* if it can possibly have more than one post batch-transitions at some time during the simulation run. We call P_{DB} the set of all diverge batch places. With the above auxiliary definitions, we can now define

6) $mc : P_{MB} \to \{\mathrm{PM}, \mathrm{SM}\}$, a function indicates the merge logic for every merge batch place, if it follows Priority Merge (PM) or Shared Processor Merge (SM). The definitions for these logics are given at Appendix A.

7) $dc : P_{DB} \rightarrow \{\text{PD}, \text{BD}, \text{RD}\}$, a function describes the diverge logic for every diverge batch place, if it follows Priority Diverge (PD), Blocking Diverge (BD) or Reroute Diverge (RD).

8) M_0 denotes the initial marking.

The graphic symbolism of S-BPN is the following:

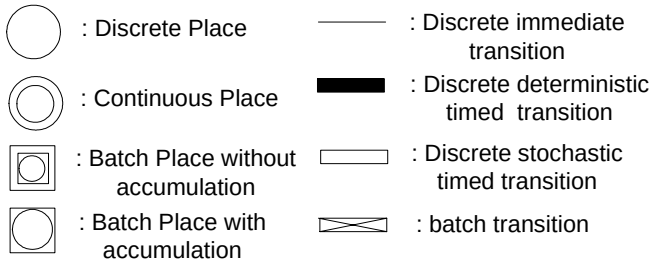


Figure 2.9 The Nodes of S-BPN

We further require that the Petri Net network has feed-forward structure. That is no cycle exists in the network. This is because a loop will pose difficulty on the iterative algorithm (will be described in section 2.3.5) to calculate the instantaneous firing flow.

Marking of a S-BPN

The marking definition of S-BPN is the same as BPN, which depends on the kind of place:

If $p_i \in P_d$, then $M(p_i) \in N$.

If $p_i \in P_c$, then $M(p_i) \in \Re^+$.

If $p_i \in P_b$, then $M(p_i) = \{B^1, ..., B^K\}$, it is an ordered set of batches.

Weight conditions:

In S-BPN, weights are used to indicate flow intensity or fractional flow from different sources. The weight conditions are summarized in the Table 2.1.

Table 2.1 Arc Weight Condition of S-BPN

	$t_j \in T_d$		$t_j \in T_b$	
	$\mathrm{Pre}(p_i, t_j)$	$\mathrm{Post}(p_i, t_j)$	$\mathrm{Pre}(p_i, t_j)$	$\mathrm{Post}(p_i, t_j)$
$p_i \in P_d$	Z^+	Z^+	Z^+	NULL
$p_i \in P_c$	$\Re^+$	$\Re^+$	$\Re^+$	$\Re^+$
$p_i \in P_b$	$Cap_n(i)$	NULL	$\Re^+$	$\Re^+$

The arc linking a pre-place to a discrete transition can be a regular arc (with arrow end) or an inhibit arc (with small circle end). The weight of an arc linking a batch place to a discrete transition is a specific sensor of this batch place. The sensor value on this arc will participate on deciding the enable condition of this transition. There is no link leading a discrete transition to a batch place or a batch transition to a discrete place. Beside these constraints, S-BPN also requires that there is at most one batch place in the sets of pre-places and post-places of a batch transition.

2.3.2 Example of S-BPN Modeling

S-BPN are made of a "continuous part" (batch places, continuous places and batch transitions) and a "discrete part" (discrete places and discrete transitions). The continuous part models system continuous flows and the discrete part models the control logic functioning.

We show how net components introduced in above section are used to model a conveyor system through a small example. Figure 2.10 shows the S-BPN of the Time Sliced Merge release.

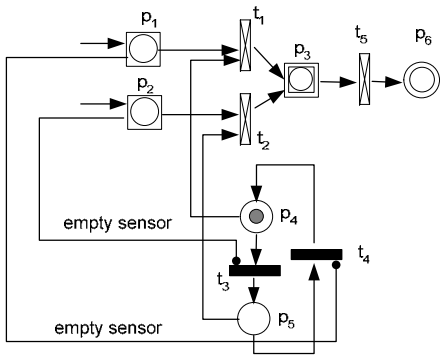


Figure 2.10 S-BPN for a Time Sliced Merge

The flows merge from two accumulation conveyor segments modeled by batch places p_1 and p_2 to a non-accumulation segment modeled as place p_3. Merging sequence is under the control of discrete part of the system which made up of discrete place p_4, p_5 and discrete time transition t_3 and t_4. Specifically, batch transition t_1 and t_2 becomes enabled alternatively according to the time delay set through timed transition t_3 and t_4. The inhibit arc linking from p_1 to t_4 will disable transition t_4 when p_1 becomes empty, so that the token will be kept at place p_5; thus flow from p_2 can be transferred to p_3 uninterruptedly through transition t_2. Another sensor arc achieves the same result when p_2 becomes empty. Batch transition t_5 models a sorter. Sorter capacity can be set as the maximum firing flow function of the transition t_5. Continuous place p_6 models a virtual sink that cumulates the flow that is out of the system.

2.3.3 Main differences between S-BPN and BPN

In the Table 2.2, we summarize some main features that differentiate the S-BPN from original BPN definition and its potential benefits. These features enhanced BPN's modeling capacity, which allows S-BPN to model fairly complex conveyor networks and control logics. By relaxing structure constraints that are not needed in fluid simulation, S-BPN can model the system structure in a more simplified and transparent manner than original BPN.

Table 2.2 BPN and S-BPN Comparison

Features	BPN	S-BPN	Benefit/Disadvantage
Type of discrete transition	Deterministic Timed Transition	Immediate Transition Timed Transition Deterministic Timed Transition Stochastic Timed Transition	Stochastic arrival and interruption model can be integrated into the model.
Type of continuous transition	Batch Transition Continuous Transition	Batch Transition	Unified continuous flow transition rule.
Type of batch place	Batch place	Normal batch place Merge batch place Diverge batch place	Be able to model more complex network structure then tandem network.
Discrete arc type	Regular arc	Regular arc Inhibit arc	Facilitate control logic modeling
Arc weight condition (Linking from batch place to discrete transition)	No Arc	Sensor Arc	Be able to model accumulation sensitive control logic
Co-continuous place requirement*	Enforced	Relaxed	Model simplification/ more macro events
Arc connectivity restriction *	Enforced	Relaxed	Model simplification/ change certain firing rules
Capacity of continuous place	Infinity	Finite buffer capacity	Easy to model finite continuous buffer /more macro events
MFF of batch transition	Fixed value	Time-varying function	Facilitate the integrating of fluid simulation with other external simulation units

*: Co-continuous place requirement refers to the restriction that BPN requires every batch place must have a continuous place, which limits the capacity. This condition is relaxed in S-BPN; instead we define the full condition of an accumulation batch place as an event.

Arc connectivity restriction refers to the following constraint. BPN requires that if an arc joins a discrete place at a batch transition, it must exist a reciprocal arc linking this batch transition at this discrete place. This is relaxed in S-BPN by modifying the enable and firing rules. Detail about the relaxation is found in Appendix B.

2.3.4 Evolution Rules

The S-BPN evolution is realized by firing transitions and by updating batch condition on batch places according to Batch on Conveyor Model. It evolves through a sequence of macro-states upon the occurrence of macro-events. Macro-state and Macro event have been used to describe the net behavior of first-order hybrid Petri Nets (Balduzzi and Giua, 2001).

The *macro-state* of a S-BPN corresponds to a time interval such as:

1) the marking (including reserved and non-reserved marks) of the discrete places is constant.

2) the reserved marking of the continuous places is constant.

3) the instantaneous firing flows associated with batch transitions are constant.

The passage from a macro-state to another macro state is carried out when an event occurs which is called a *macro-event*. The interval of time between the occurrences of two consecutive macro-events is called *macro-period*. Note that the above three macro-state conditions imply that the set of enabled transitions is kept unchanged during a macro-period.

The macro-events in S-BPN are listed below exclusively:

1) A discrete transition fires.

2) A batch in a batch place becomes an output batch.

3) The output batch in a batch place becomes null.

4) The value of a sensor in a sensor arc changed.

5) An accumulation batch place becomes full.

6) A continuous place becomes empty or full.

7) A marking of continuous place reaches the corresponding arc weight thus enabling some discrete transitions.

8) Maximum firing flow associated with a batch transition is modified

The end time of the current macro-period is determined by the nearest macro-event time. A macro-period may have zero duration due to the firing of immediate discrete transitions. In stochastic Petri Nets terminology, these periods are called *vanishing* periods, and the periods with positive duration are called *tangible* periods. In a vanishing period, conflicting enabled discrete transitions will fire according to their priority function.

The firing of discrete transition occurs at the end of a macro period. The firing of batch transitions always associates with a tangible macro-period. The IFF vector is calculated at the beginning of the tangible period. Within a macro-period, the fluid is drained from the input continuous and batch places of a batch transition and pumped into the output continuous and batch places. The conditions of enabling and firing of S-BPN were included in Appendix B.

2.3.5 Calculation of Instantaneous Firing Flows (IFF)

IFF calculation is the most complex step in S-BPN. We first introduce two more definitions, merging unit and diverging unit.

If there is more than one enabled batch transition in the set of pre-transitions of a merge batch place in a tangible macro-period, the batch place and its associated enabled pre-batch-transitions is called a *merging unit* in this macro-period. If there is more than one enabled batch transition in the set of post-transitions of a diverge batch place, the batch place and its associated enabled post-batch-transitions is called a *diverging unit*. A transition that is not in a merging unit or a diverging unit is called a *normal batch transition* at a macro-period.

Unlike merge or diverge batch place, the merging unit or diverging unit is a dynamic concept. That means all merging units and diverging units should be identified at the beginning of each macro-period.

There are two reasons that make the IFF calculation the most complex step in this fluid simulation framework. First, an empty continuous place or a full batch place results in the dependency among batch transitions. Hence, the general procedure for computing IFFs is an iterative process. The convergence of the algorithm is guaranteed since an IFF is increasing over the iterations and it always has an upper limit that could be its MFF or the upstream or downstream theoretical flow. Second, the batch transitions in a merging unit or a diverging unit should be calculated simultaneously using a single algorithm at each iteration step. Detail IFF calculation algorithm is included in Appendix B.

2.4 Discrete Event Simulator Based on S-BPN Approach

The dynamic of S-BPN is based on a discrete event continuous time behavior. With macro-events and macro-states, the net behavior of S-BPN can be described by a discrete-event model, which can be implemented to construct an efficient and general simulation tool. We built a prototype fluid simulator FluidSim from scratch using Java. Fig 2.11 shows the flow of simulation algorithm of our simulator.

In this simulator, the flow rate change is simulated and tracked at any point of the network. Liu (1999) called this type an exact fluid simulation in the literature. It has the same accuracy as cell-based simulation in term of all summary level performance measures, such as throughput, average accumulation length and utilization etc. From the definition of macro events, the number of events is expected to be reduced drastically compared with cell-based simulation. However, each step in the simulation flow can take much longer time than a single event handling in cell-based simulation. For example "compute IFFs" step involves an iterative computing procedure.

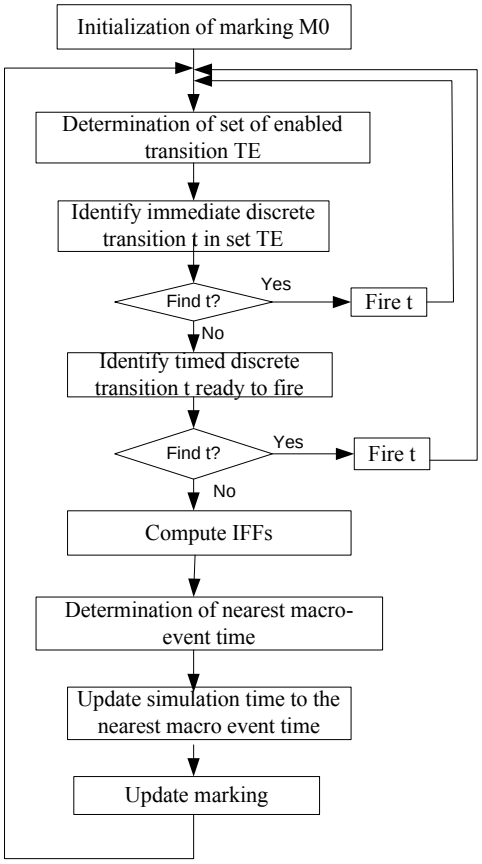


Figure 2.11 Fluid Simulation Algorithm

2.5 Performance Measure

In conveyor simulation studies, we are interested in system throughput, operator and buffer utilization, conveyor segment density as well as item sojourn time. Fluid simulation generates three types of function of time: IFF function $\phi(t)$ for every batch transition, the buffer content function $b(t)$ for every continuous place and accumulation position function $a(t)$ for every batch place. Various node, segment and network performance measures of interest can be derived using these functions. For instance,

The utilization of a particular machine or operator modeled as a transition t_j is given by

$$\overline{u}_i = \frac{\int_0^T \phi_i(t)dt}{\int_0^T \Phi_i(t)dt}.$$

The utilization of a conveyor segment modeled as batch place p_i is given by

$$\overline{u}_i = \frac{\int_0^T \sum_{\{t_j \in T_b : t_j \in p_i^o\}} \phi_j(t)dt}{\int_0^T (v_i * d_i)dt}.$$

Average workload of a buffer and average accumulation length of conveyor segment are given by

$$\overline{w}_i = \frac{1}{T}\int_0^T b_i(t)dt$$

$\overline{a}_i = \frac{1}{T}\int_0^T a_i(t)dt$ respectively.

To find network level performance measure, we can add two continuous places with infinite buffer capacity p_k and p_l to denote a virtual source and a virtual sink. Let all flow originate from p_k and terminate at p_l. System throughput is given by the total amount of fluid into sink during [0,*T*] divided by *T*. Average system sojourn time is a little harder to derive since the individual identity is lost in fluid simulation. An approximate method using the input-output diagram approach can be done as follows. We draw a function of cumulative amount of fluid out of source and a function of cumulative amount of fluid into the sink as functions of time. The average sojourn time can be approximated by the area enclosed by these two curves and any two horizontal lines divided by the vertical coordinator difference of these two horizontal lines.

2.6. Case Study

We apply the S-BPN modeling to a regional distribution center of a retailer chain. The conveyor network layout is shown in Figure 2.12. The DC has an area of 101,700 square meters and operates as both distribution center and warehouse where cross-docking goods are forwarded and staple goods are stored. It has a capacity to serve to a maximum of 105 stores. The operation time is 24x7 with 3 shifts. Most of its cross-dock

goods are handled in cases. The incoming staple goods can be in pallets or in cases. The outgoing staple goods are mostly cases. The total length of conveyor belt is around 40 kilometers. The conveyor belt links up the products from the receiving docks and staple product picking modules to the sorter and to shipping docks. Its accumulation and merge system consists of 21 crossdocking product receiving lines D1 to D21, six case picking modules P1 to P6 and three break case picking modules B1 to B3. R1 is a return line linked to the sorter. In total, these 31 input lines first go through some satellite merges and then lead to the main merge area where 16 lanes are merged into 4 lanes by sawtooth merge. In addition, before conveyor belt passing the sorting area, they are merged again at the mini merging area into 2 lanes which leads to two high-speed one side sliding shoe sorters. More than 50 docks are located at either side of two sorters.

The S-BPN model of the example system is shown in Fig 2.13. The S-BPN model uses accumulation and non-accumulation batch places to model conveyor segments. Conveyor segment connectivity is modeled through batch transitions. In order to compare the simulation performance between fluid simulator and the traditional cased-based item-level simulator, we also built a simulation model in Arena for this case. Table 2.3 shows the number of modules and objects used in the Arena model and FluidSim respectively.

Comparing to Arena model, FluidSim model uses fewer objects and it is much more straightforward to construct. Next we conduct simulation performance comparison in terms of accuracy and simulation speed.

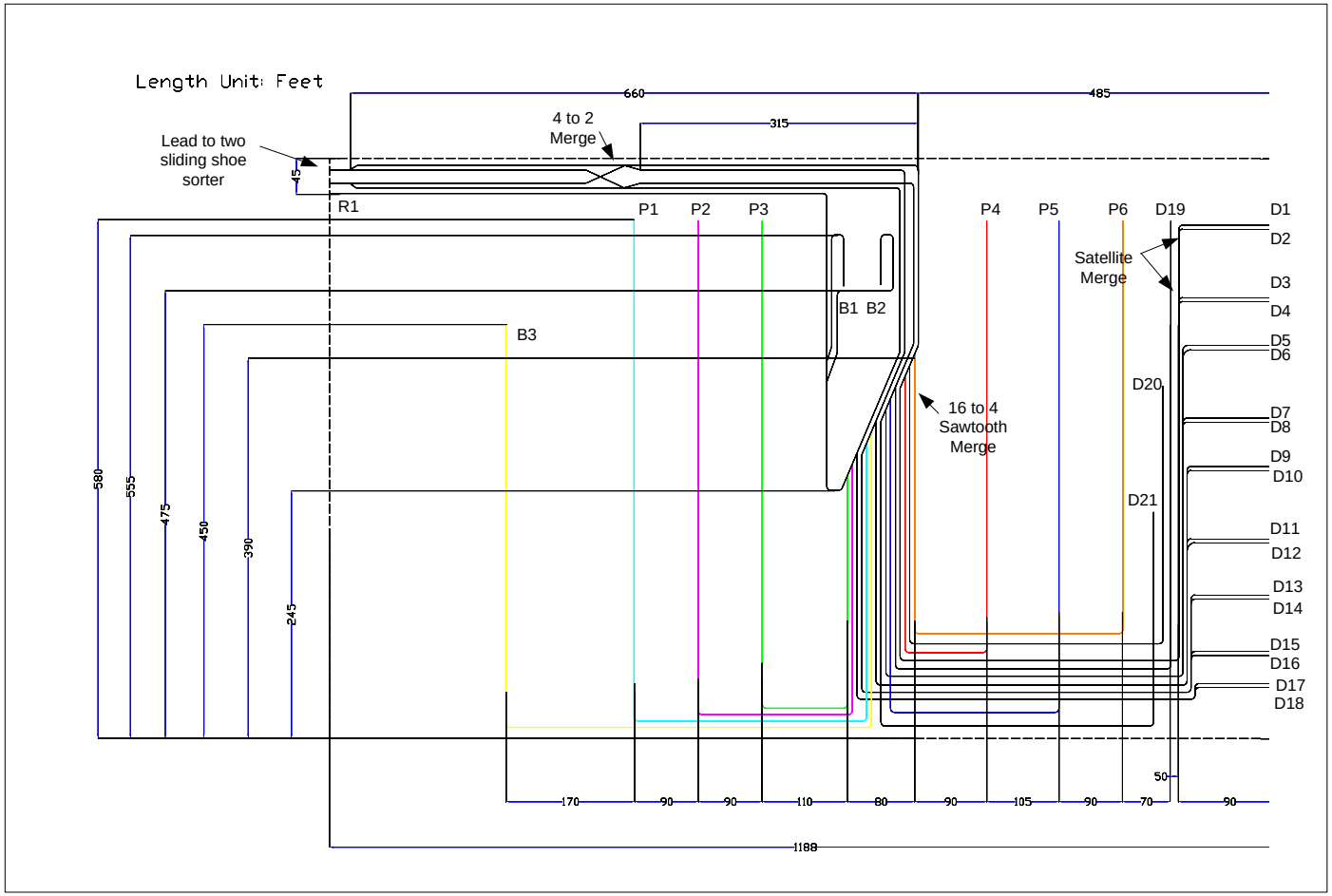


Figure 2.12 Case Conveyor Transportation Network

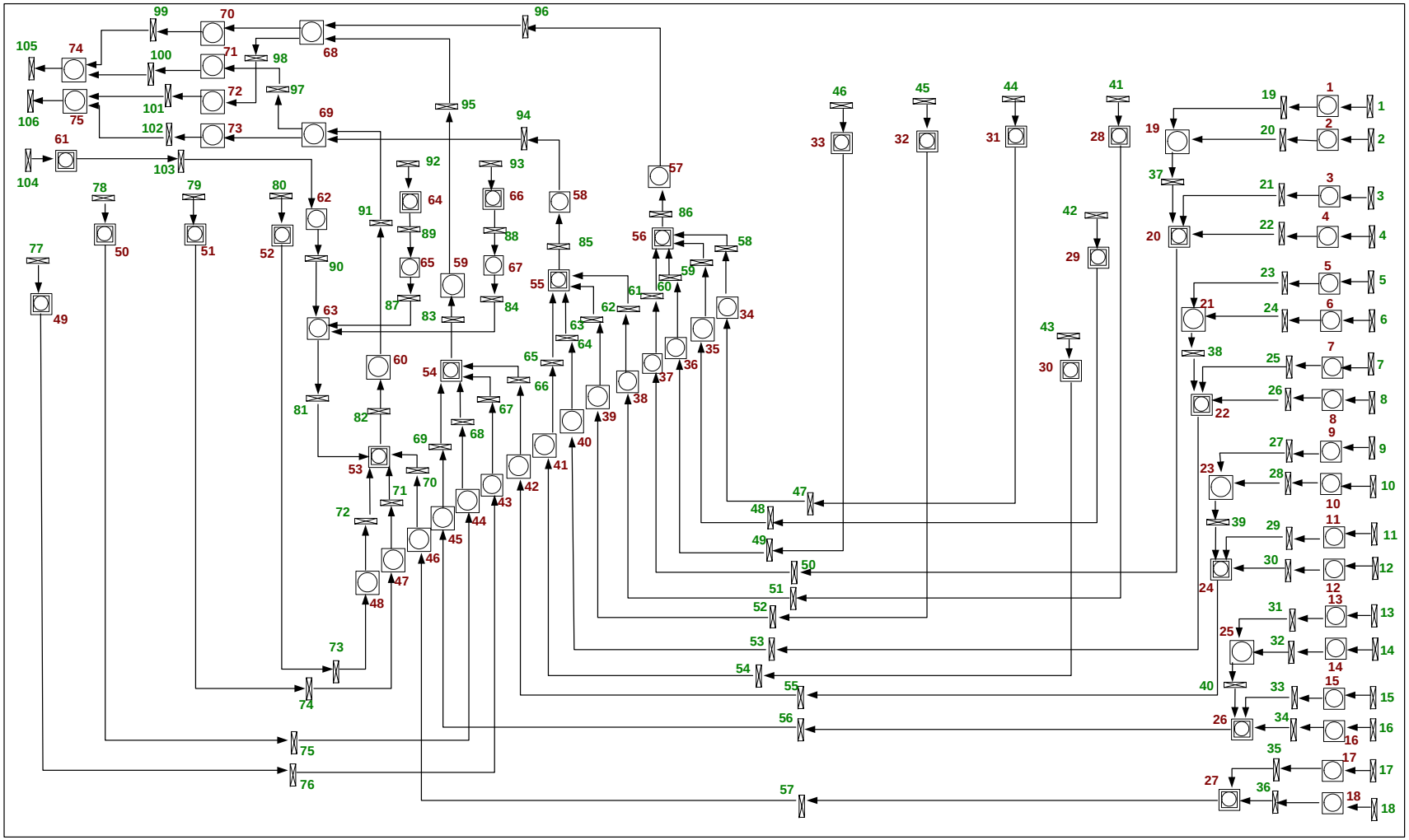


Figure 2.13 S-BPN Model of Case Conveyor Network

Table 2.3 Modeling Objects Used in Two Models

Arena Model		FluidSim	
Module or Block	Number	Object	Number
Conveyor	76	Place	76
Segment	76	Transition	110
Access	76	Arc	189
Exit	76		
Convey	76		
Station	74		
Assign	51		
Queue	31		
Decide	21		
Arrive	31		
Total	588		375

In the experiment, we use the same external random input generator for both models. This program generates the random fluid input streams according to some underlying Markov Modulated Fluid Process model at 31 sources. These streams are recorded in maximum firing flow table in FluidSim and schedule module in Arena. We generate the input stream for 2 hours and record the corresponding system throughput and simulation run time for both models. The data and graph are provided as Table 2.4 and Figure 2.14.

Table 2.4 Simulation Accuracy Comparison between Arena Model and FluidSim

	Total Accumulation Length = 1403 meter			Total Accumulation Length = 2810 meter		
	2-hour throughput	System Empty Time (Min)	Simulation Run Time (Min)	2-hour throughput	System Empty Time (Min)	Simulation Run Time (Min)
FluidSim	45939	180	0.48	47098	180	0.47
Cell Len (item length)						
1	34784	212	0.65	36340	219	1.35
1/2	39392	190	0.82	40466	193	1.57
1/4	41232	186	0.92	42977	184	1.85
1/8	43315	181	1.13	44410	182	2.10
1/16	43522	181	1.50	44501	181	2.48
1/32	44038	180	2.18	45074	180	3.42
1/64	44090	180	3.53	45298	180	5.27

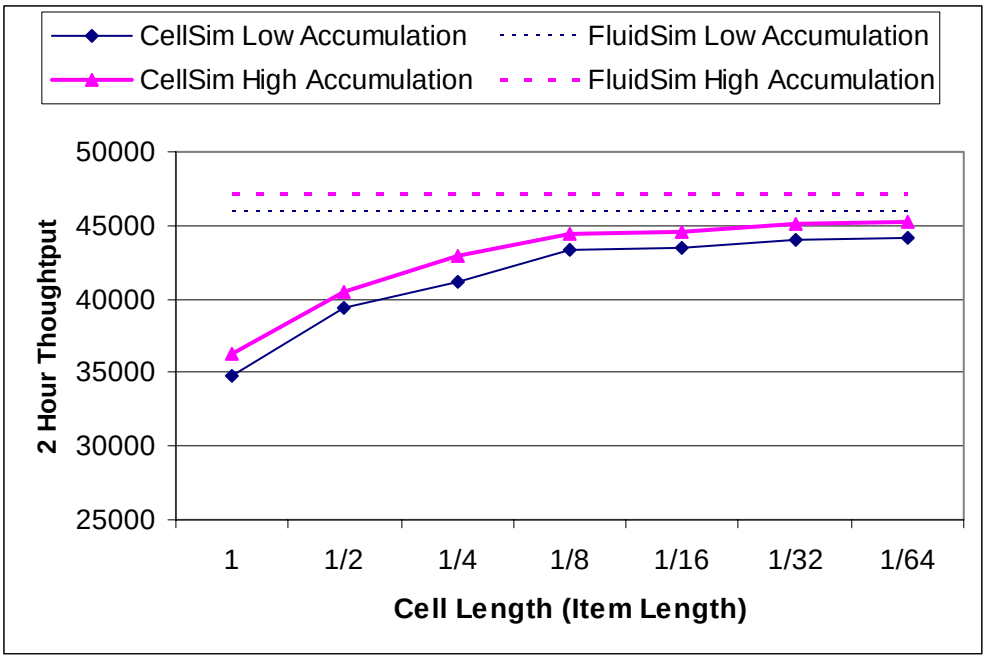


Figure 2.14 Case Simulation Throughput Comparison

In this experiment, we assume an item has a length of 0.3 meters. We try different cell sizes on the Arena model. The result appears that the throughput of cell based simulation increases as we reduce the cell size. Based on the analysis we presented at section 2.1, the converged value should approach the real system throughput. The throughput obtained from FluidSim is within 4% of the converged value from Arena. The gap we believe is because of the continuously adding up the fractional values that will not be admitted as input in the item level simulation. We compared two cases with relatively low and high accumulation. Throughput shows an increase in higher accumulation case, but the overall trend and the gap illustrated by the two cases are the same. This indicates FluidSim gives fairly accurate performance estimation. Figure 2.15 compares the simulation run time of the two models.

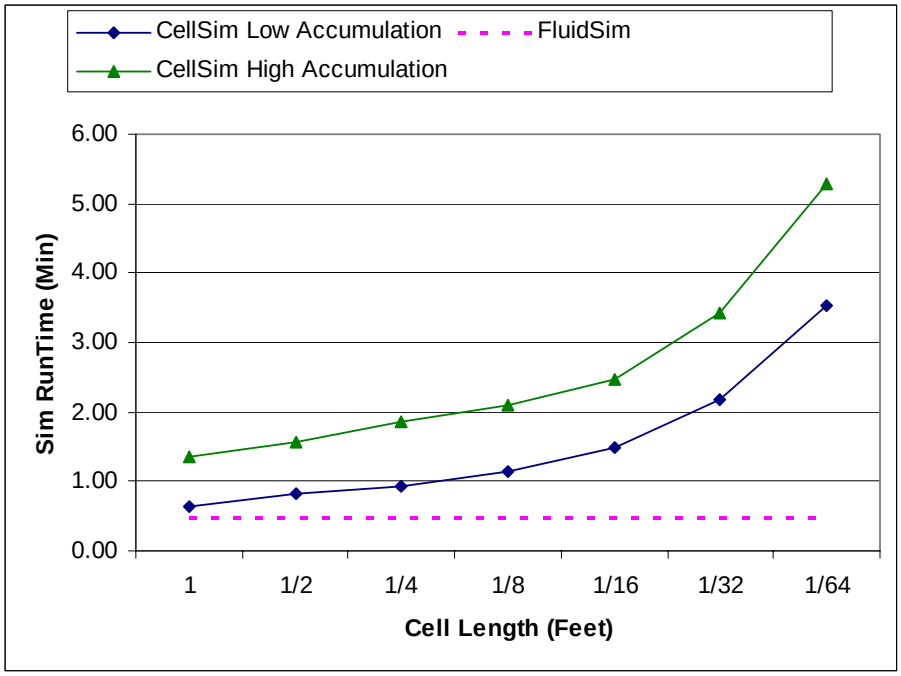


Figure 2.15 Case Simulation Runtime Comparison

Smaller cell size greatly increases the computational effort of the item level simulation. This effect is more significant in high accumulation case. In general, this case example illustrated that we can construct a complex fluid simulation model fairly quickly and it can return satisfactory performance measure results with much less computation effort than item-level simulation model.

2.7 Computational Comparison between Cell-based Simulation and Fluid Simulation

We performed computational comparison of cell based simulation approach and fluid simulation approach on several network configurations and different operational parameters. The detail network configuration data is provided in Appendix C. Experiments are designed to facilitate our investigation on major factors that could affect the simulation time of two approaches. In this section experiments, all simulation times are collected under steady state simulation. We ensure this by specifying a delay before

simulation time is recorded. The delay time is determined by observing a time after which the WIP level is stabilized.

WIP vs. Simulation Speed

In the 15-segment configuration, we gradually increase the input intensity and keep the constant transportation capacity so that the WIP level will increase accordingly. Table 2.5 and Figure 2.16 shows the relationship between the simulation run time of two hours wall clock time period and the WIP level of two approaches. We use log scale of run time in the vertical axis because otherwise the scale difference is too great.

Table 2.5 WIP vs. Simulation Time Experiment

15-Segment Example											
Num of Conveyor Segments	15										
Total Conveyor Length (meter)	4877										
RealTime (Minutes)	120										
Warm-up time (Minutes)	120										
Total AccConveyor Length (meter)	4877										
Average WIP	598	622	1428	3417	4002	4777	5663	6308	8627	10353	14577
Sim Time (Seconds)											
Arena	0.6	0.6	0.6	4.2	6.0	69.0	168.0	207.0	254.4	245.4	250.2
FluidSim	2.6	2.9	2.5	3.7	3.0	3.8	3.0	3.0	1.8	4.8	2.0

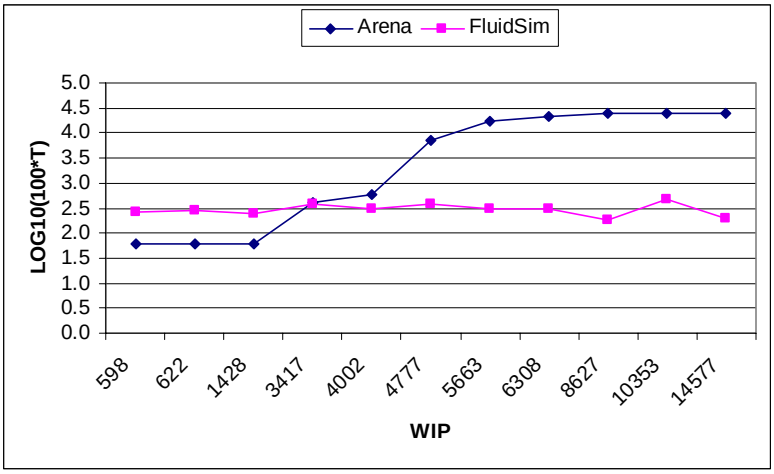


Figure 2.16 Simulation Runtime vs. WIP

We can see the cell-based simulation time increases exponentially as the WIP increases while the fluid based simulation is much faster and is not sensitive to the number of packages on the conveyor. This demonstrates the advantage of fluid simulation on high-volume system simulation. The Arena curve becomes flat after certain

WIP level, this is because the system becomes saturated and the increased WIP are just stocked at different queues.

Accumulation Length vs. Simulation Time

In the above case, we also found that non-accumulation conveyor does not increase much computational burden to cell-based simulation while the simulation time increases drastically as the total length of accumulation conveyor increases. Table 2.6 and Fig 2.17 illustrate this observation. This is because the items on non-accumulation conveyor changes status uniformly so that the discrete events generated can be aggregated. Fluid simulation is not sensitive to the accumulation length.

Table 2.6 Accumulation Conveyor Length vs. Simulation Time Experiment

77-Segment case						
Num of Conveyor Segments	77					
Total Conveyor Length (meter)	7710					
RealTime (Minutes)	240					
Warm-up time (Minutes)	120					
Total Accumulation Conveyor Length(meter)	1404	2681	4189	5713	6637	7710
Average WIP	14403	15763	17985	18535	18465	21393
Sim Time (Seconds)						
Arena	56	232	375	455	591	725
FluidSim	55	46	32	29	27	24

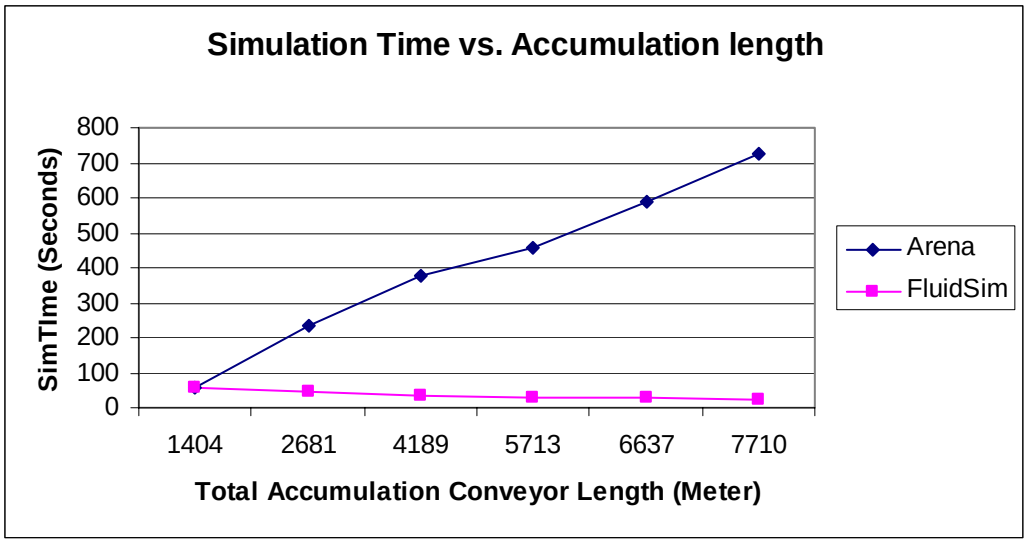


Figure 2.17 Simulation Runtime vs. Accumulation Conveyor Length

Input Rate Change Frequency vs. Simulation Time

In the 15-segment configuration, we fixed the WIP level around 4500 and gradually decrease the average arrival rate change interval of one input source. The relationship between runtime and this interval value is shown in Figure 2.18.

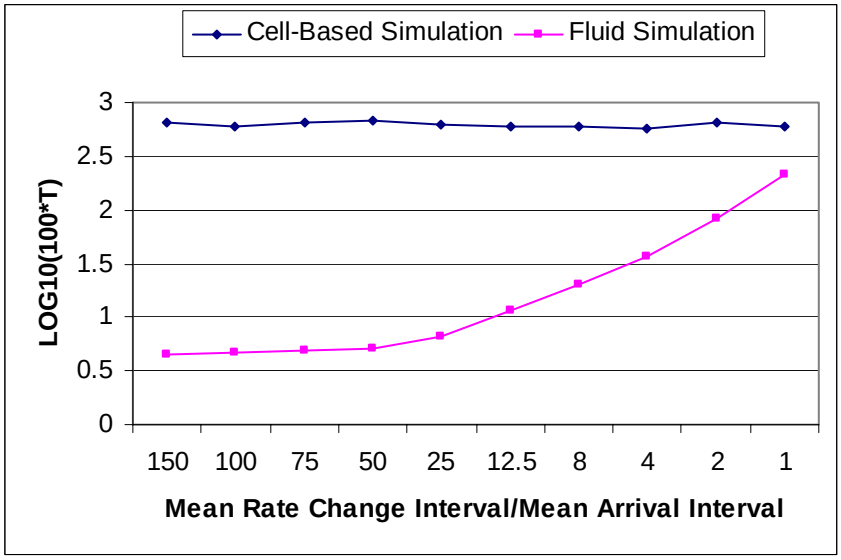


Figure 2.18 Simulation Runtime vs. Release Interval

We use the ratio of mean rate change interval to mean arrival interval as the horizontal axis label. The ratio has little impact to cell-based simulation runtime, while fluid simulation runtime increases drastically as the ratio approaches one. This trend is expected since the number of events in fluid simulation will converge to the number of events in cell-based simulation as the batches becomes smaller and smaller. So the potential for computational savings of fluid simulation is greatest when rates remain constant for significant periods of time; hence, a slowly changing environment is the key factor to justify using of fluid simulation.

Network Complexity vs. Simulation Time

The computational savings of fluid simulation also decreases as the network complexity increases. This can be shown from Table 2.7.

Table 2.7 Simulation Time vs. Network Complexity

Total Accumulation Conveyor Length (Meter) around			2743
Total WIP level			10000
Simulation Time (Min)			120
	2-Seg	15-Seg	76-Seg
Number of Sources: *m*	1	4	17
Number of levels in Merge Hierarchy :*H*	1	2	4
SimRunTime(Min)			
FluidSim	0.02	0.05	0.47
CellSim	4.95	5.08	5.27
CellSim/FluidSim	297	102	11

In this experiment, we examine three network configurations from simple to complex structure. For each configuration, we fix the total accumulation conveyor length at 2743 meters and control the WIP level around 10000 items. The simulation run time for 2 hour wall clock time are recorded for both approaches. We can see the ratio of cell based simulation run time to fluid simulation run time decreases sharply as the network becomes more and more complex. This is because the upstream large batches are often truncated into smaller batches at merge junctions. A rate change event in any upstream segment could trigger the generation of a new batch in the downstream segment. Figure 2.19 illustrated this effect in a 3-to-1 merge.

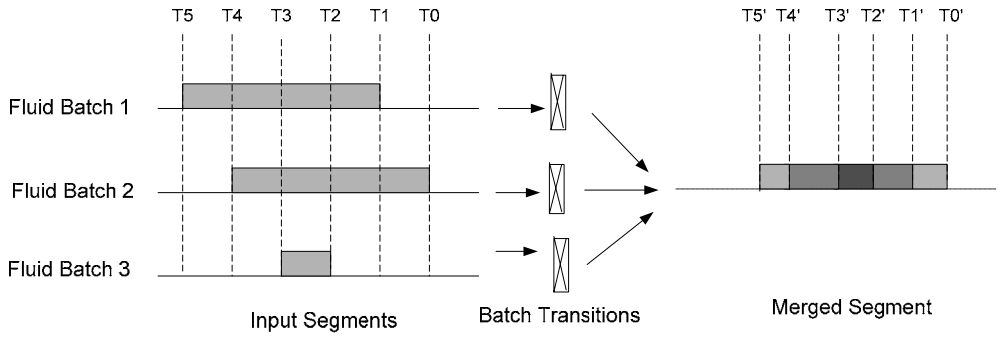


Figure 2.19 Number of Batches in a Merge Junction

As a result of interweaving effect between the arrival and departure event of each batch, the original three batches in the three upstream segments generate five batches in the post merge segment. In the following theorem, we attempt to quantify this effect by considering the extreme case.

Theorem 2-2: In an in-tree structure Accumulation/Merge system, the number of batches simulated for a period time T is $O(2^H m\mu)$, where H is the number of levels in merge hierarchy in the network, m is number of sources and $u = \max\{u_i : i = 1,...,m\}$. Here we use u_i to denote the expected number of batches observed in the source i for a period time T.

Proof: First suppose in the first level of merge hierarchy, m sources were partitioned into groups to participate in merge at s junctions. Without loss of generality, let $\{1,..,m_1\},\{m_1+1,..,m_2\},\ldots,\{m_{s-1}+1,..,m\}$ denote the sources participated in merge at junction 1,2,..,s respectively.

Now consider the number of batches after merge at junction 1. Starting with the expected number of batches in source 1, we consider the effect of adding the batches in source 2, 3... m_1 one by one. Since each batch at most can add two more batches to the original batch sequence by splitting a batch into two with its arrival and departure events, the number of batches after merge at junction 1 is at most $u_1 + 2u_2 + ... + 2u_{m_1}$. This can be bounded as $O(2m_1u)$. By the same derivation, we can say the number of batches after merge at junction 2,…,s are bounded as $O(2(m_2 - m_1)u),\ldots,O(2(m - m_{s-1})u)$ respectively. So the total number of batches after first level merge hierarchy is bounded as $O(2mu)$. Next, we prove the theorem result through mathematical induction method.

Suppose p conveyors attempt to merge in the h+1 layer of merge. Denote the number of batches in these p conveyors as x_1, x_2,…, x_p and suppose we have $\sum_{i=1}^{p} x_i < 2^h mu$. Since each batch at most adds two more batches to the original batch sequence, the number of batches after h+1 merge layer is bounded by $x_1 + 2x_2 + ... + 2x_p$. This quantity is further bounded by

$x_1 + 2x_2 + ... + 2x_p < 2 * \sum_{i=1}^{p} x_i \le 2 * 2^h mu = 2^{h+1} mu$. This completes the proof. □

This theorem reveals the three main factors that can affect the computational effort of fluid simulation. They are the number of input sources, the number of levels in merge hierarchy and expected number of batches in a source which relates to the input rate change frequency. Notice, although the computational savings decreases when the complexity of the network increasing, the so-called "ripple effect" is less pronounced in conveyor network simulation than in the telecommunication network simulation. This can be made clear by comparing Figure 2.19 with Figure 2.20 below.

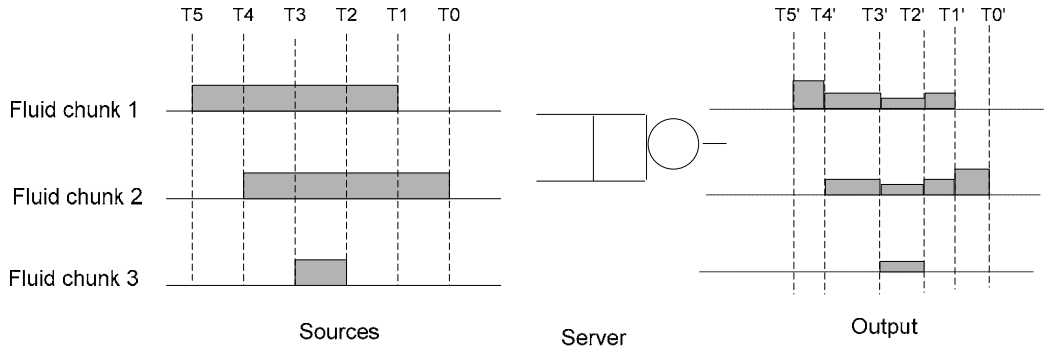


Figure 2.20 Ripple Effect in Telecommunication Network Fluid Simulation

In telecommunication network, the input streams are not merged together after going through a server. Instead there are three output streams after passing the server. So the original three fluid chunks end up with nine fluid chunks in three output streams. This batch number increase will further propagate to the downstream server nodes. The batch number growth rate is faster than that of conveyor network simulation where all dividing effects are aggregated into one output stream. So even for a 76 segment complex network simulation, FluidSim is still 11 times faster than the item level simulation.

2.8. Summary

In this chapter, we present a fluid simulation model for feed-forward conveyor network. We first developed continuous flow model for single conveyor segment and then introduced the S-BPN model which serves as a architecture framework that holds

all system elements together. S-BPN combines the advantages of formalism in discrete Petri nets for modeling controls and of macro feature of fluid model for reduced events in dynamic system. From Petri Nets modeling perspective, we extended and modified the Batch Petri Nets in several ways to enhance its modeling capability while reduced the model complexity at the same time. This Petri Net framework makes the transformation from a continuous model to a discrete event simulator fairly easy through object oriented programming implementation. Extensive experiments are designed to compare the accuracy and simulation speed between the fluid simulation and traditional cell based item-level simulation. Primary experimental result is that fluid simulator can achieve the similar accuracy (<5% percentage difference) in much shorter simulation time. The computational savings is especially significant when simulating a network that comprises of many accumulation conveyor segments. We also identified three main factors that can affect the computational effort of fluid simulation. They are the number of input sources, the number of merge hierarchies and expected number of batches in a source which relates to the input rate change frequency. Although the computational savings decreases when the complexity of the network increasing, the so-called “ripple effect” of fluid simulation is less pronounced in conveyor network simulation than in the telecommunication network simulation.

CHAPTER 3

ACCUMULATION AND MERGER CONVEYOR NETWORK PARAMETRIC OPTIMIZATION FRAMEWORK

This is the first of three chapters addressing the parametric optimization of accumulation and merge (A/M) conveyor network. The objective is to find a minimum cost design that satisfies certain performance requirement under stochastic work environment. In this chapter, we present a formal A/M network representation and its stochastic non-linear parametric optimization model. Since the model is analytically non-tractable, we propose a gradient-based simulation optimization solution procedure to solve it. The solution procedure is further detailed in Chapters 4 and 5.

3.1 System Description

Accumulation and merge (A/M) conveyor network, shown in Figure 1.2, collects items from order picking, receiving stations, and other collecting conveyors. It attempts to merge and assemble a high density and evenly spaced item stream for the downstream sorter. The abstract representation of A/M conveyor is shown in Fig 3.1.

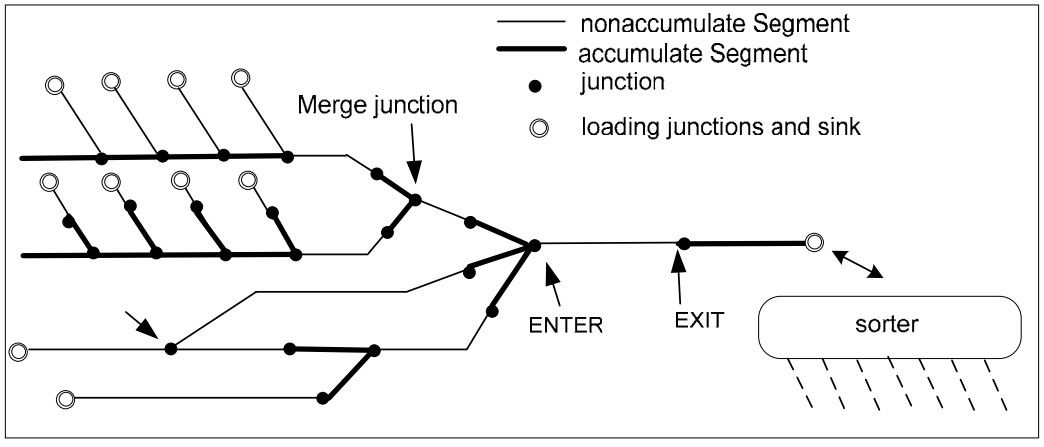


Figure 3.1 A/M Conveyor Network Illustration

3.1.1 A/M Network Representation

The elements of this conveyor network abstraction are conveyor segments and junctions. Mathematically, we represent a segment by the structure $sm = \{v, l, d, c, IN, OUT\}$, where v, l, d and c are segment parameters as defined in section 2.2. IN and OUT are set of junctions that link with sm at ENTER and EXIT end respectively.

We use junction to model the pure connection between two segments, a merge/diverge control point or a station with or without a buffer. We also define a virtual ENTER side and EXIT side for junction even though there is no physical distance between ENTER and EXIT of a junction. All upstream elements of the junction connect to ENTER side while EXIT side connects with all downstream elements. It transfers items from upstream elements to downstream elements according to some transfer control logic. We represent a junction formally by $jn = \{r, b, \sigma, IN, OUT\}$, where

r	maximum transfer capacity, items/time unit
b	maximum buffer capacity in number of items
σ	transfer control logic
IN	set of upstream conveyor segments and junctions
OUT	set of downstream conveyor segments and junctions

Notice that connection among elements is defined through *IN* and *OUT* structure. That means if an element *i* belongs to the *IN* set of element *j*, i.e., $i \in IN_j$, this implies ENTER of j overlaps with the EXIT of *i*.

To model pure connection of two consecutive conveyor segments, *b* is simply set to zero since there is no physical buffer between these two segments. *r* is set to infinity since there is no additional control over the connection and the transportation capacity realized is purely determined by the speeds of two segments themselves.

Using segment and junction definition, an A/M conveyor network can be defined by the structure $CN = (SM, JN, SIN, s)$.

SM	the set of conveyor segments
JN	the set of junctions
$SIN \subset JN$	the subset of junctions that receive external input
m	the cardinality of subset *SIN*.
$s \in SM \cup JN$	the sink with OUT(s) = $\varnothing$.

A junction in *SIN* might not represent a physical loading station. External arrival can be just to the ENTER side of a conveyor, but we create a virtual junction for modeling purpose. For the same reason, if the external arrival is to certain middle point of a conveyor segment, this segment will be split into two segments in the model with a junction inserted in between.

3.1.2 Model Assumptions

Our conveyor network parametric design model is based on the following assumptions:

1. Sortation subsystem is of type S/PS-2. Basically, the number of shipment destinations of all orders in a batch will always be no more than the number of take-away lanes of the sorter. So an item will be diverted to its corresponding destination when it reaches the lane for the first time, and no recirculation will occur unless the corresponding take-away lane is full or an item misread occurs. In the later case the item will be re-circulated to the induction point. Since this full lane situation is much less frequent and it can also be treated as a separate problem within the S/PS subsystem, we will not take this into consideration in the following analysis. That means we assume that except for sorter speed no other operational issues in S/PS will have influence on the system throughput.

2. Uniform product length. Very often the products circulating on the conveyor system are of different length. For analysis simplification, we use a nominal product length for all quantity transferring.

3. The layout of conveyor network is specified. Specifically, for each $sm_k \in SM$, its type c_k, input links IN_k and output links OUT_k are known. We also assume the speed of each conveyor segment has also determined. The general guild to construct a feasible A/M conveyor network is to use a network of minimum total conveyor length that satisfies space and connectivity constraints. Conveyor speed can be initially designed to satisfy the flow capacity constraints that specified through average arrival rates. Both decisions are important steps in A/M network design and usually can be done without taking the stochastic work conditions into consideration. While in this research, we focus on the next level decisions that need to incorporate the stochastic nature of the system into the model. For example, accumulation design closely depends on the variability of system input. Another reason to assume known conveyor speed is that we can not estimate the cost coefficients of conveyor segments without knowledge of conveyor speeds.

3.1.3 Arrival Process

We consider arrivals over a continuous time horizon [0, T] to the system. T is the duration of orders to be filled in a wave, a shift or a day. The stochastic nature of arrivals comes from the differences between orders, the order picking process and the interruptions. In discrete modeling, Poisson arrival is commonly used. Closer observation reveals that items tend to arrive at A/M in chunks of different densities. The high input rate can temporarily overload within each wave. This bursty nature does not agree with Poisson arrival assumptions. Also as mentioned in chapter 2, different behaviors in the system are on different time scales, the queuing effect contributed from variability of

larger time scale is significant and dominant compared to small time scale effect. We need a stochastic process that can better capture this larger time scale variability.

We adopt stochastic fluid model in describing the arrival process. Let $\Lambda_i(t), i = 1,..m; t \in [0,T]$ be a stochastic process taking values in $\Re^+$ and $\lambda_i(t), i = 1,..m; t \in [0,T]$ a sample point of $\Lambda_i(t)$ from its sample space. $\lambda_i(t)$ represents the arrival rate immediately after time *t* at junction $i \in SIN$. We also assume $\Lambda_i(t)$ has a piece-wise constant sample path and the arrival rate of two sources $\Lambda_i(t)$ and $\Lambda_j(t)$ are independent. We use $\Lambda = (\Lambda_i(t),..,\Lambda_m(t))$ and $\lambda = (\lambda_i(t),..,\lambda_m(t))$ to denote the vector form of random input process and its sample path. Although fitting a stochastic input model for Λ to drive the simulation is itself a challenging problem, we will assume that a good fitting and generating mechanism has already been developed so that we can focus on the network optimization.

3.1.4 Performance Measure

One of the most important design objectives for high volume order fulfillment system is to achieve its required throughput. High sorter throughput requires A/M subsystem to present a uniform and high density item stream. The length, speed, accumulation and merge control are important decision variables to ensure the buffering and streamlining the imbalances of arrival and order picking processes. A cost effective design will achieve minimum blocking of arrival process and minimum voids in the item stream to the sorter simultaneously.

In real operation, the items that can not be loaded to the system because of temporary blocking might attempt to reload at some later time. However it is hard to incorporate this operation into an analytical model. So we assume that arrival process Λ will not be affected by blocking of arrival to A/M system. In other words the overflow volume is considered lost. This assumption is acceptable since this model does not serve

as an accurate performance evaluation tool while it is designed to be used in the design phase to evaluate the feasibility of a system configuration.

In the optimization model we use loss volume $\mathcal{L}$ defined below as a performance measure to evaluate the feasibility of design. Higher sorter utilization is achieved automatically in the course of searching for a minimum cost feasible design. Let $\Psi(t)$ be the cumulative arrival process to the sink, the loss volume is defined as

$$\mathcal{L} = \sum_{i\in[1,..m]} \int_0^T \Lambda_i(t)dt - \int_0^T \Psi(t)dt \,. \qquad (3\text{-}1)$$

In the model T is chosen much longer than the time period during which there is positive input flow so that the system WIP level at time T dropped to zero. $\mathcal{L}$ is a random variable since Λ and $\Psi(t)$ are stochastic.

3.1.5 Cost

The system cost considered in the model consists of fixed and variable cost associated with conveyor segments, merge devices and sorter.

Conveyor segment fixed cost depends on equipment type and accumulation type. For example, powered live roller conveyor ranges from $200-$250/ft; powered slider bed conveyor ranges from $180-$250/ft; while powered accumulation conveyor ranges from $250-$350/ft. Conveyor speed will affect selecting of equipment type and segment variable cost. In our model, segment cost only depends on accumulation type and rate since we do not specify equipment type information in the model.

For merge devices, lower rates directly lead to lower variable cost and indirectly lead to lower fixed cost by selecting slower and less sophisticated merge devices. For example, live roller merge and herringbone merge are considered low-to-medium rate merge in the range of 10-80 cartons/min; while sliding shoe merge and sawtooth merge is capable of rates in excess of 80 cartons/min.

Sorters can also be classified by three rate categories:

Low rate 0 - 40 cartons per minute

Medium rate 40 - 80 cartons per minute

High rate 80 - 200 cartons per minute

Example low rate sorters are 90 degree transfers, barrier-type diverters and pusher sorters; example medium rate sorters are pop-up wheel and roller sorter; while sliding shoe sorters and tilt tray sorters are considered as high rate sorters. Usually, higher rate sorters are much expensive than lower rate sorters.

3.2 A/M Network Parametric Optimization Model

3.2.1 The Model

The objective of the model is to determine the set of parameters of a minimum cost design that satisfies the maximum loss percentage requirement in the stochastic work environment. By assumption the layout structure and conveyor speed are already determined by outer loop design functions. The decision variables in this model are conveyor length, junction capacity and buffer size, which mathematically we can denote as $\{l_k \mid k \in SM\} \cup \{r_l, b_l \mid l \in JN \setminus SIN\}$. An important design parameter sorter speed is also represented as one of the junction capacity. We let $\theta \in \Re^D, D = |SM| + 2 * (|JN \setminus SIN|)$ denote the complete set of decision variables.

The design problem can be formulated into the following optimization problem.

Problem 3-1: Minimize $C(\theta)$

s.t. $$g \equiv E[\mathcal{L}(\Lambda, \theta)] \le E\left(\sum_{i \in [1,..,m]} \int_0^T \Lambda_i(t) dt \right) * p\%$$

$$\theta \in F$$

Where C is some convex cost function, $\mathcal{L}(\Lambda, \theta)$ is the random variable representing the loss volume for a given θ; g is the expected loss volume and p is pre-

selected system percentage loss limit. F represents the feasible set for parameter l, r, b that satisfies the facility dimension and equipment capacity constraints.

Problem 3-1 is a constrained stochastic optimization problem with linear or nonlinear objective function and nonlinear stochastic constraints. Furthermore, there is no closed form expression for the loss volume even for a deterministic input stream, it can only be approximated by another optimization problem or in detail by simulation. Problems of this type lead themselves to a simulation optimization method. Various simulation optimization techniques can be used, with the choice based on the assumption of the feasible region. See Olafssion (2002) for a recent survey of simulation optimization methods. Our decision variables are the length of each conveyor segment and merging, sorting speed at each junction. If we treat these variables as discrete, the feasible region is finite but combinatorially large. Only adapted metaheuristics methods such as simulated annealing, tabu search, genetic algorithms are capable. However, these metaheuristics often suffer from slow convergence. If we treat decision variables as continuous, we can use some gradient-based search method such as stochastic approximation. Stochastic approximation schemes generally require convex and differentiable objective functions. We will show later that simulated performance function could possibly be non-differentiable. This poses difficulties in employing stochastic approximation method. Furthermore, stochastic approximation is often considered slow compared with deterministic optimization algorithms.

Another approach is sample path optimization presented and analyzed by Gurkan (1994), and Plambeck, Fu and Robinson (1996). It is also called sample average approximation in Shapiro (2003). The basic idea of such methods is to approximate the expectation function through Monte Carlo sampling, the consequent sample average approximation problem becomes deterministic and the powerful machinery of mathematical optimization can be applied directly. The optimal solution of this sample average approximation problem is taken as an estimator of the "true" optimizer. Under

certain conditions, this approach can be shown to converge almost surely as the sample size of Monte Carlo sampling increases. We adopt this approach in the following solution procedure because of the non-smooth nature of our problem and the desire of fast convergence.

3.2.2. Solution Procedure Overview

In problem 3-1, we have two types of constraints. The feasible set F is given by explicit, simple linear constraints. They represent the physical bound of the system and parameters outside these constraints are invalid. So they are considered as *hard* constraints. These constraints are better handled by projection method in numerical solution procedure. The constraint imposed by loss volume performance measure is stochastic and often nonlinear, non-differentiable. This constraint is considered *soft* which means we allow parameters to take on values violating this constraint during optimization. This type of constraint can be incorporated into objective function in a form of penalty. Problem 3-1 will then be transformed to a sequence of mildly constrained problem:

Problem 3-2: Minimize $C(\theta_k) + r_k P\left(E\left[\mathfrak{L}(\Lambda, \theta_k)\right] - E\left[\sum_{i\in[1,..m]} \int_0^T \Lambda_i(t)dt \right] p\% \right)$

s.t. $\theta_k \in F$

Penalty function $P(x)$ should be defined satisfying the following conditions:

(1) $P(x)$ is differentiable on R

(2) $P(x) \geq 0$ for all $x \in R$

(3) $P(x)$=0 if and only if $x \leq 0$

(4) $P(\cdot)$ is strictly increasing on $[0, \infty)$

The r_k are a sequence of penalty weights with $r_k > 0$ and $r_k \to \infty$.

We solve the problem 3-2 by sample path optimization, that is, by generating random samples $\lambda^1, \lambda^2, .., \lambda^N$ of Λ and approximating problem 3-2 by its corresponding sample average problem:

Problem 3-3: Minimize $U(\theta_k) \equiv C(\theta_k) + r_k \cdot P(g_N(\theta_k) - M)$

s.t. $\theta_k \in F$

Where $g_N(\theta_k) = N^{-1} \sum_{j=1}^{N} L(\lambda^j, \theta_k)$ and $M = \left(N^{-1} \sum_{j=1}^{N} \sum_{i \in [1,..m]} \int_0^T \lambda_i^j(t) dt \right) * p\%$. M is a constant for a fix set of samples.

We assume that C and g are convex with respect to θ. Due to the possibility that g is non-smooth, we employ projection sub-gradient optimization procedure to solve problem 3-3. The iterative scheme of this approach has the form

$$\theta_{k+1} = \Pi_F(\theta_k + a_k \cdot s(U)) \tag{3-2}$$

Where $s(U)$ is a subgradient of U at θ_k and calculated by

$$s(U) = \nabla C(\theta_k) + r_k \cdot \nabla P(g_N(\theta_k) - M) \cdot s(g) \tag{3-3}$$

∇ indicates the gradient operator and $s(g)$ is a subgradient vector of $g_N(\theta_k)$.

$\{a_k\}$ is a positive sequence of numbers converging to 0, and Π_F denotes a projection operator that maps points in $\Re^D$ to their nearest neighbor in F. In the following two chapters, we investigate on two methods that evaluate $g_N(\theta_k)$ and $s(g)$.

3.3 Summary

In this chapter, we give a formal network representation of A/M network. We present its parametric optimization model and sketch the simulation optimization solution procedure. The success of this solution procedure depends on having fast performance evaluation tools that not only can determine the loss volume but also can provide the loss gradient with respect to design parameters quickly. We investigate two such methods in

the following two chapters. In chapter 4, “Delay and Stock Model” is chosen and performance evaluation is done through an analytical deterministic optimization model. In chapter 5, we explore the prospect of using fluid simulation proposed at chapter 2 in conjunction with the above simulation optimization framework.

CHAPTER 4

OPTIMIZATION VIA DYNAMIC NETWORK FLOW MODEL

In this chapter, we present another conveyor continuous model - "Delay and Stock Model". Based on this model, we propose an analytical approach for evaluating the loss volume for a single replication of Monte Carlo sampling in the accumulation and merge (A/M) network parametric optimization problem. The resulting model is a class of dynamic network flow problem that we call the maximum flow over time problem with piece-wise constant arc capacity and constant finite node capacity. For the model solution, we transfer the problem to a linear programming problem through uniform discretization. Not only does this model give us the loss volume but it also provides us with a subgradient of loss volume with respect to the design parameters. Thus, we can incorporate this model into our parametric optimization solution framework sketched in the chapter 3. The pros and cons of this approach are discussed at last.

4.1 Delay and Stock Model

In chapter 2, we introduced a continuous conveyor model - Batch on Conveyor Model, which is used in the fluid simulation. We consider that model an "exact" continuous model since the model keeps track of conveyor state evolution in time and space two dimensions. In this chapter, we present an approximate continuous abstraction called "Delay and Stock Model". Essentially, Delay and Stock Model abstracts the conveyor transportation as a fixed delay τ from its ENTER end to EXIT end and providing a fixed size buffer at the EXIT end of the conveyor. If it is a non-accumulation conveyor, the buffer size is zero. In the following, we refer readers to list of symbols section for notation meaning of $v, l, d, c, \mathcal{I}(t), \mathcal{O}(t), I(t), O(t), a(t)$.

Delay and Stock Model expresses conveyor capacity and accumulation constraints mathematically through following equations and inequalities.

$$\tau = \frac{l}{v} \tag{4-1}$$

$$\int_0^t I(\theta - \tau)d\theta = \int_0^t O(\theta)d\theta + a(t) \quad \forall t \in [0,T] \tag{4-2}$$

$$I(t) \le \mathcal{I}(t) \quad \forall t \in [0,T] \tag{4-3}$$

$$O(t) \le \mathcal{O}(t) \quad \forall t \in [0,T] \tag{4-4}$$

$$I(t) \le vd \quad \forall t \in [0,T] \tag{4-5}$$

$$O(t) \le vd \quad \forall t \in [0,T] \tag{4-6}$$

$$\begin{cases} a(t) = 0 & \text{if } c = \text{'N'} \\ a(t) \le \dfrac{l}{d} & \text{if } c = \text{'A'} \end{cases} \qquad \forall t \in [0,T] \tag{4-7}$$

Equation 4-1 states the fixed traverse time is equal to the segment length divided by its speed. Equation 4-2 is the flow balance equation followed by material conservation law. (4-3) and (4-4) limit the effective in-flow and out-flow rate to the input flow rate and output capacity. (4-5) and (4-6) describe the flow rate limits imposed by transportation capacity of segment. (4-7) requires accumulation position will never exceed the maximum buffer size. At each instant, $I(t)$ and $O(t)$ attain maximum values that satisfy conditions stated in (4-1) to (4-7).

4.2 Maximum Flow Over Time Model

In this section, we will make use of the above Delay and Stock Model to construct an analytical model for performance evaluation of an A/M network with a given deterministic input stream. We first introduce the model and then give a transformation procedure for transforming the data in an A/M network to the input data of the analytical network model. Solution technique is stated at last.

4.2.1 Definition

A *maximum flow over time* problem with piece-wise constant arc capacity and constant node capacity is defined on a networks $\mathcal{N} = (V, A)$ with n:=|V| nodes and m:=|A| arcs. Each arc $e \in A$ has an associated integral transit time or length τ_e and a non-negative, left-continuous, piece-wise constant capacity function u_e: (0,T]$\rightarrow \Re^+ \cup \{0\}$. We can scale time such that the break points of all u functions are integral multiple of time unit. Each node $v \in V$ has an associated integral capacity z_v. Let $S^+ \subseteq V$ be a subset of sources and $S^- \subseteq V$ a single sink. The problem is to find a feasible flow over time that sends as much flow as possible from sources to the sink in time T.

A flow over time f on $\mathcal{N}$ with time horizon T is given by a collection of Lebesgue-measurable functions f_e:[0,T)$\rightarrow \Re^+$ where $f_e(\theta)$ determines the rate of flow (per time unit) entering arc e at time θ. The problem can be formulated as follows.

$$\text{Max} \int_0^T \sum_{e \in \delta^-(\text{S}^-)} f_e(\theta - \tau_e) d\theta$$

$$0 \le \int_0^t \left[\sum_{e \in \delta^-(v)} f_e(\theta - \tau_e) - \sum_{e \in \delta^+(v)} f_e(\theta) \right] d\theta \le z_v \quad \text{for all } t \in [0,T),\ v \in V \setminus S^+ \qquad (4\text{-}8)$$

$$f_e(\theta) \le u_e(\theta) \quad \text{for all } \theta \in [0,T) \text{ and } e \in A \qquad (4\text{-}9)$$

The objective is to maximize total flow to the sink during the time horizon. (4-8) is the flow balance constraint which states that for any node v other than source, at any time instant t, the total inflow to a node v minus total outflow of node v must be non-negative but less than the node capacity. Here $\delta^+(v)$ and $\delta^-(v)$ denote the set of arcs leaving node v and entering node v, respectively. Constraints (4-9) require a feasible f obey the capacity constraints through the time horizon.

4.2.2 Transformation Procedure

We describe the procedure to transform a problem that finds the loss volume in an A/M network $CN = (SM, JN, SIN, s)$ with a given input stream λ to an instance of maximum flow over time problem in three steps:

1) Select a base time unit Δ.

The choice of Δ should satisfy that all breakpoints of arc capacity functions in maximum flow over time problem and all arc transit time are integral multiple of Δ. The first condition means $\omega \geq \Delta$, where ω denotes the greatest common divisor of all breakpoints of λ functions. Generally, we consider the case when the capacities change over larger time scale than individual item movement. That means ω is relatively large. For the second condition we need to round up the real transit time l_k/v_k for all $k \in SM$ to the nearest multiple of Δ. The smaller Δ we choose, the less error will be in the solution of maximum flow over time problem. However, a small value of Δ can also leads to prohibitive computational costs in solving the problem. Since this model will be used in the preliminary network design phase, a relatively larger Δ may be appropriate.

2) Determine the arc set and node set.

The node set consists of all ENTER and EXIT point of all segments and junctions plus an auxiliary ACC point for each accumulation type segment. The construction procedure is as follows:

a. For any $sm \in SM$ and $c(sm)$='A', add an auxiliary position ACC between ENTER and EXIT of k.
b. V=ENTER(k)$\cup$ACC(k) $\cup$EXIT(k) for all $k \in SM \cup JN$
c. $e=(i,j) \in A$ if {ENTER(k) = i and EXIT(k) = j for $k \in JN$ or $k \in SM$ and $c(k)$ ='N'}
   or {ENTER(k) = i and ACC(k) = j for $k \in SM$ and $c(k)$ ='A'}
   or { ACC(k) = i and EXIT(k) = j for $k \in SM$ and $c(k)$ ='A'}

3) Determine the arc capacity, arc transit time and node capacity. The formulation is as follows:

a. S^+ = {ENTER(k):$k \in SIN$}; S^- = EXIT(s)

$$\text{b.}\ \forall\ (i,j) \in A\ ,\ \tau_{(i,j)} = \begin{cases} \lfloor l_k / v_k \rfloor & i = \text{ENTER}(k) \quad j = \text{EXIT}(k) \quad k \in SM \quad c(k) = \text{'N'} \\ \lfloor l_k / v_k \rfloor & i = \text{ENTER}(k) \quad j = \text{ACC}(k) \quad k \in SM \quad c(k) = \text{'A'} \\ 0 & i = \text{ACC}(k) \quad j = \text{EXIT}(k) \quad k \in SM \quad c(k) = \text{'A'} \\ 0 & i = \text{ENTER}(k) \quad j = \text{EXIT}(k) \quad k \in JN \end{cases}$$

$$u_{(i,j)}(t) = \begin{cases} v_k d_k & i = \text{ENTER}(k) \quad j = \text{EXIT}(k) \quad k \in SM \quad c(k) = \text{'N'} \\ v_k d_k & i = \text{ENTER}(k) \quad j = \text{ACC}(k) \quad k \in SM \quad c(k) = \text{'A'} \\ v_k d_k & i = \text{ACC}(k) \quad j = \text{EXIT}(k) \quad k \in SM \quad c(k) = \text{'A'} \\ r_k & i = \text{ENTER}(k) \quad j = \text{EXIT}(k) \quad k \in JN \setminus SIN \\ \lambda_k(t) & i = \text{ENTER}(k) \quad j = \text{EXIT}(k) \quad k \in SIN \end{cases}$$

$$\text{c.}\ \forall\ v \in V\ ,\ z_v = \begin{cases} l_k d_k & v = \text{ACC}(k) \quad k \in SM \quad c(k) = \text{'A'} \\ b_k & v = \text{EXIT}(k) \quad k \in JN \\ \infty & v = \text{ENTER}(k) \quad k \in SIN \ \text{ or } \ v = \text{EXIT}(s) \\ 0 & \text{OW} \end{cases}$$

It can be verified that this transformation procedure enforced the solution of the maximum flow over time problem satisfies all capacity and accumulation constraints in the original network expressed through (4-1) to (4-7).

4.2.3 Related Model in Literature

Maximum flow over time model falls into the class of dynamic network flow problem (DNFP). Dynamic network flow problems generalize standard network flow problem by introducing an element of time to model problems where travel are not instantaneous or when network capacities restrict the quantity of flow that can be sent at any one time, and thus necessitate sending flows in phases.

Within DNFP problem class, *maximum dynamic flow problem* was defined similarly as in our problem, but it has constant arc capacity and infinite node capacity. Ford and Fulkerson (1962) showed that the maximum dynamic flow problem could be solved in polynomial time via one minimum-cost flow computation. *Quickest transshipment problem* is defined in a dynamic network with a set of sources and sinks; each source has a specified supply of flow and each sink has a specified demand. The problem is to send exactly the right amount of flow out of each source and into each sink in the minimum overall time. Hoppe and Tardos (2000) give the first polynomial-time algorithm for this problem. All of these algorithms are based on the concepts of chain

decomposable flows. A *chain flow* $\gamma =< w, P >$ sends fixed units of flow *w* along a path P that obeys capacity constraints and length of the chain flow $\tau(\gamma)$ is less than the planning horizon *T*. $\tau(\gamma)$ equals the sum of the transit time of arcs in the path *P*. A feasible dynamic flow can be induced from any chain flow by sending *w* units of flow along P from time 0 until time $T - \tau(\gamma)$. A static flow can be decomposed into a set of chain flows and the corresponding dynamic flows induced by these chain flows are called chain decomposable flow.

Their approach is not applicable to our maximum flow over time problem since the time-varying arc capacity makes the chain decomposition of static flow invalid. Hoppe and Tardos (2000) also showed that there always exists an optimal flow over time that does not require the node storage except for sources and sinks, however, once the time-varying capacities are introduced, node capacity may need to be utilized in an optimal solution. Some DNFP problems with time-varying arc capacity have also been considered by several researchers (Ogier, 1988; Fleischer, 2001), but zero-transit times were assumed in their model. When all transit times are zero, the flow in an arc at any moment of time is independent of the flow in this arc in any previous moment. This independence allows for more flexible decomposition of the time horizon into time intervals.

In summary, although there is substantial body of research on DNFP, the time-varying capacity and non-zero transit time of maximum flow over time problem prevents the use of any existing fast algorithm.

4.2.4 Solution Technique

In this section, we will pursue the uniform discretization approach to solve the maximum flow over time problem with piece wise constant arc capacity and constant finite node capacity. We discretize the continuous time into steps of unit length Δ. Denote

τ_e and T as integral multiples of Δ, $\tau_e = m_e \cdot \Delta$ $m_e \in \{0,1,2,...\}$ and $T = p \cdot \Delta$. Construct a linear programming problem DP as follows.

DP $\max\ Q \equiv \sum_{k=0}^{p-1} \sum_{e \in \delta^+(S^-)} \hat{f}_e(k - m_e) \cdot \Delta$

S.T. $\Delta[\sum_{e \in \delta^+(v)} \hat{f}_e(k - m_e) - \sum_{e \in \delta^-(v)} \hat{f}_e(k)] = \hat{y}_v(k+1) - \hat{y}_v(k) \quad \forall v \in V \setminus S^+ \quad k = 0,..,p-1$

$\hat{f}_e(k) \le u_e(k\Delta) \quad k = 0,..,p-1; \forall e \in A$

$\hat{y}_v(k) \le z_v \quad k = 0,...,p; \forall v \in V$

$\hat{f}_e(k) \ge 0 \quad k = 0,1,...,p-1; \forall e \in A$

$\hat{y}_v(k) \ge 0 \quad k = 0,1,...,p; \forall v \in V$

$\hat{f}_e(k) = 0; \hat{y}_v(k) = 0 \quad \forall k < 0; \forall e \in A; \forall v \in V$

Fleischer and Tardos (1998) pointed out a strong connection between the continuous and discretized models. They show that if a continuous problem is feasible, there is a solution f that changes only at times in {1Δ, 2Δ,…, pΔ}. So an optimal discrete-time solution can be transformed into an optimal continuous-time solution by sending flow at rate $\hat{f}(k)$ in the interval [kΔ, (k+1)Δ). We call this *standard transformation procedure*. Thus, we can solve the maximum flow over time problem by solving linear programming problem DP, which can be solved by commercial mathematical programming solver like CPLEX.

4.3 Parametric Optimization

We incorporate the above maximum flow over time model into our chapter 3 parametric optimization framework as a tool to evaluate the loss volume and loss volume subgradient with respect to design parameters for single replication of Monte Carlo sampling. Recall optimization framework in chapter 3, in each iteration we need to

calculate $s(g)$ which is a subgradient vector of average loss $g_N(\theta_k) = N^{-1}\sum_{j=1}^{N} L^j(\lambda^j, \theta_k)$, where θ_k is the vector of decision variable at iteration k, N is the number of replications in Monte Carlo sampling and L denotes loss volume of a replication. If we know the subgradient vector for each replication $s(L)$, $s(g)$ can be calculate by averaging.

$$s(g) = N^{-1}\sum_{j=1}^{N} s(L^j) \tag{4-10}$$

In next section, we show how to determine L^j and $s(L^j)$ through maximum flow over time model. To simplify notation, we omit the replication index in the following.

4.3.1 Loss Subgradient Derivation

Loss volume L is the difference between total input stream and total flow to the sink. Total flow to the sink is the objective function value of problem DP. So L is

$$L = \sum_{i\in[1,..,|SIN|]} \int_0^T \lambda_i(t)dt - Q\,. \tag{4-11}$$

The transformation procedure indicates the decision variable l, r, b of original network are represented as parameters u_e, z_v and m_e in problem DP. To achieve the analytical form, we ignore the small change of m_e due to small change of l, so that we can say the decision variables appear purely on the right hand side of constraints in problem DP. Only a subset of node capacities and arc capacities of network $\mathcal{N}$ corresponds to the decision variables. We denote them as $\{u_e : e \in A' \subset A\}$ and $\{z_v : v \in V' \subset V\}$. We first formulate the dual of problem DP as follows.

Dual of DP $\quad$ Min $\sum_{e\in A}\sum_{k=0,..,p-1} u_e(k\Delta)\cdot w_e(k) + \sum_{v\in V}\sum_{k=0,..,p} z_v \cdot q_v(k)$

s.t. $\quad \Delta h_i(k+m_e) - \Delta h_j(k) + w_e(k) \geq 0 \quad \forall e(i,j) \in A \setminus \delta^+(\mathrm{S}^-); k = 0,..,p-1$

$$\Delta h_i(k+m_e) - \Delta h_j(k) + w_e(k) \geq \Delta \quad \forall e(i,j) \in \delta^+(S^-); k = 0,..,p-1$$

$$h_v(k+1) - h_v(k) + q_v(k) \geq 0 \quad \forall v \in V; k = 0,..,p$$

$$w_e(k) \geq 0 \quad \forall e \in A; k = 0,..,p-1$$

$$q_v(k) \geq 0 \quad \forall v \in V; k = 0,..,p$$

Let $\{u_e^{\,0} : e \in A'\}, \{z_v^{\,0} : v \in V'\}$ be arc and node capacities that correspond to the parameter values at current iteration and $\{u_e^{\,*}(k\Delta) : e \in A; k = 1,..,p-1\}, \{z_v^{\,*} : v \in V\}$ the current right hand side value of problem DP. This implies $u_e^{\,*}(k\Delta) = u_e^{\,0}$ for $e \in A'; k = 1,..,p-1$ and $z_v^{\,*} = z_v^{\,0}$ for $v \in V'$. We use $\{w_e^{\,*}(k) : e \in A; k = 1,..,p-1\}$ and $\{q_v^{\,*}(k) : v \in V; k = 1,..,p\}$ to denote the optimal dual solution of problem DP corresponding to current right hand side value. We can obtain the loss subgradients with respect to decision variables from these dual variables. This is stated in the following theorem.

Theorem 4-1: The component of subgradient vector $s(L)$ corresponding to the parameter $u_e, e \in A'$ is $-\sum_{k=0}^{p-1} w_e^{\,*}(k)$; The component of subgradient vector $s(L)$ corresponding to the parameter $z_v : v \in V'$ is $-\sum_{k=0}^{p} q_v^{\,*}(k)$.

Proof: We can write the objective value Q of problem DP as a function of right hand side value $Q(\{u_e(k\Delta) : e \in A, k = 0,..,p-1\}, \{z_v : v \in V\})$. From general LP theory, Q is a piece-wise linear and concave function of right hand side value. According to strong duality theorem, we have

$$\sum_{e\in A}\sum_{k=0,..,p-1} u_e^*(k\Delta)\cdot w_e^*(k)+\sum_{v\in V}\sum_{k=0,..,p} z_v^*\cdot q_v^*(k)$$
$$=Q(\{u_e^*(k\Delta):e\in A;k=0,..,p-1\},\{z_v^*:v\in V\}) \quad . \tag{4-12}$$

$\{w_e^*(k)\}$ and $\{q_v^*(k)\}$ are also a feasible solution to the dual problem corresponding to any $\{u_e(k\Delta)\},\{z_v\}$, so according to the weak duality theory we have

$$\sum_{e\in A}\sum_{k=0,..,p-1} u_e(k\Delta)\cdot w_e^*(k)+\sum_{v\in V}\sum_{k=0,..,p} z_v\cdot q_v^*(k)\geq$$
$$Q(\{u_e(k\Delta):e\in A;\forall k=0,..,p-1\},\{z_v:v\in V\}) \quad . \tag{4-13}$$

Subtract (4-12) from (4-13), we have

$$\sum_{e\in A}\sum_{k=0,..,p-1} w_e^*(k)(u_e(k\Delta)-u_e^*(k\Delta))+\sum_{v\in V}\sum_{k=0,..,p} q_v^*(k)(z_v-z_v^*)\geq$$
$$Q(\{u_e(k\Delta):e\in A;k=0,..,p-1\},\{z_v:v\in V\})-Q(\{u_e^*(k\Delta):e\in A;k=0,..,p-1\},\{z_v^*:v\in V\}) \tag{4-14}$$

Since (4-14) is valid for any $\{u_e(k\Delta)\},\{z_v\}$, we choose a particular one such that

$u_e(k\Delta)=u_e^*(k\Delta),\forall e\in A\setminus A';k=0,..,p-1$, $z_v=z_v^*,\forall v\in V\setminus V'$ and

$u_e(k\Delta)=u_e$ for $e\in A';k=1,..,p-1$, (4-14) reduced to

$$\sum_{e\in A'}\sum_{k=0,..,p-1} w_e^*(k)(u_e-u_e^*(k\Delta))+\sum_{v\in V'}\sum_{k=0,..,p} q_v^*(k)(z_v-z_v^*)\geq$$
$$Q(\{u_e:e\in A';k=0,..,p-1\},\{z_v:v\in V'\})-Q(\{u_e^*(k\Delta):e\in A';k=0,..,p-1\},\{z_v^*:v\in V'\}) \tag{4-15}$$

Since $u_e^*(k\Delta)=u_e^0$ for $e\in A';k=1,..,p-1$ and $z_v^*=z_v^0$ for $v\in V'$, (4-15) further reduced to

$$\sum_{e\in A'}\sum_{k=0,..,p-1} w_e^*(k)(u_e-u_e^0)+\sum_{v\in V'}\sum_{k} q_v^*(k)(z_v-z_v^0)\geq$$
$$Q(\{u_e:e\in A'\},\{z_v:v\in V'\})-Q(\{u_e^0:e\in A'\},\{z_v^0:v\in V'\}) \tag{4-16}$$

That means $\left\{\sum_{k=0}^{p-1} w_e^*(k)\right\}$ and $\left\{\sum_{k=0}^{p} q_v^*(k)\right\}$ are components of subgradient of Q with respect to parameters. Theorem conclusion is reached because of the relationship between Q and L stated in equation (4-11).□

4.3.2 Numerical Example

We apply the above methodology to an example conveyor network parametric design. We consider a two-segment tandem system with a non-accumulation segment connected with an accumulation segment. Network layout and S-BPN representation are shown in Fig 4.1.

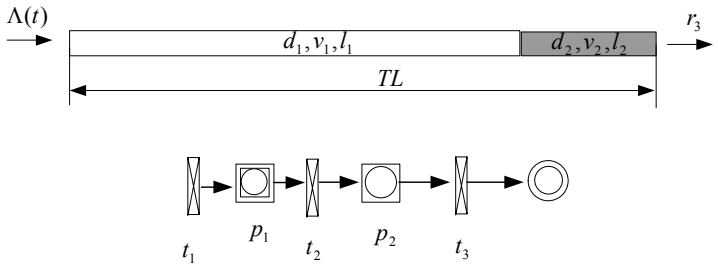


Figure 4.1 Two-Segment Tandem Conveyor System

The network comprises three junctions. External arrival $\Lambda(t)$ occurs at junction 1. Λ is modeled as Markov Modulated Fluid Process (MMFP). Junction 3 represents a sorter with capacity r_3. As described in chapter 3, the decision variables in the parametric optimization model are l_1, l_2 and r_3. Additional layout constraint requires the total length of two conveyor segments equal to a fixed value TL. So the decision variables reduced to two, l_2 and r_3. We denote vector form of decision variables as $\theta = [l_2, r_3]^T$.

For simplification, we use linear cost function in the experiment. The total system cost consists of one non-accumulation conveyor cost, one accumulation conveyor cost and the sorter cost. Since conveyor speed is given, conveyor cost purely depends on the type and length of the conveyor. Accumulation conveyor has much higher cost than non-accumulation conveyor. In the experiment, the total cost is $120l_1 + 300l_2 + 2500r_3$. The

layout constraint requires $l_1 + l_2 = 700$. We choose p=0.5%. Problem 4-1 is the real optimization problem.

Problem 4-1: Minimize $C(\theta) = 180l_2 + 2500r_3$

s.t. $E[\mathcal{L}(\Lambda, \theta)] \le E\left[\int_0^T \Lambda(t)dt\right] * 0.5\%$

$l_2 \le 700.$

As sketched in chapter 3, we construct an approximate sample average problem 4-2 as follows.

Problem 4-2: Minimize $C(\theta) = 180l_2 + 2500r_3$

s.t. $N^{-1}\sum_{j=1}^{N} L(\lambda^j, \theta) \le M$

$l_2 \le 700$

$\lambda^1, .., \lambda^N$ are random sample of Λ. $M = N^{-1}\left(\sum_{j=1}^{N}\int_0^T \lambda^j(t)dt\right) * 0.5\%$.

We move the first constraint to the objective function using penalty function $P(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } x > 0 \\ 0 & \text{OW} \end{cases}$ and penalty weight function $r_k = 1.2^k$. We apply subgradient method to update the decision variables at each iteration. Updating scheme is defined as $\theta_{k+1} = \theta_k + a_k \cdot \sigma \cdot s_k$ and $a_k = \frac{(U(\theta_k) - \overline{U}) * 0.9^k}{\|\sigma \cdot s_k\|^2}$, where $U(\theta_k) = C(\theta_k) + r^k \cdot P\left(N^{-1}\sum_{j=1}^{N} L^j\left(\lambda^j, \theta_k\right) - M\right)$ is the penalized objective function at iteration k; $\overline{U}$ is an estimated lower bound on the total cost; constant row vector σ is a normalization factor used to adjust the big difference of magnitude of the two parameters. We choose $\sigma = [1, 1/10]$ in the experiment; s_k is the subgradient vector of U with respect to θ_k, it is calculated using the following formula.

$$s_k = \begin{bmatrix} 180 + r^k \mathrm{Max}\left(N^{-1} \sum_{j=1}^{N} L^j\left(\lambda^j, \theta^k\right) - M, 0 \right) * N^{-1} \sum_{j=1}^{N} \dfrac{\partial L^j(\lambda^j, \theta^k)}{\partial l_2} \\ 2500 + r^k \mathrm{Max}\left(N^{-1} \sum_{j=1}^{N} L^j\left(\lambda^j, \theta^k\right) - M, 0 \right) * N^{-1} \sum_{j=1}^{N} \dfrac{\partial L^j(\lambda^j, \theta^k)}{\partial r_3} \end{bmatrix} \quad (4\text{-}17)$$

The key calculation in the solution procedure is to evaluate the $L(\lambda,\theta)$ and its subgradient with respect to the decision variables for each replication in the sample and at each iteration. $L^j(\lambda^j, \theta_k)$ is evaluated through maximum flow over time model and calculated through 4-11. $\dfrac{\partial L^j(\lambda^j, \theta^k)}{\partial l_2}$ and $\dfrac{\partial L^j(\lambda^j, \theta^k)}{\partial r_3}$ are obtained using formulas stated in the theorem 4-1.

Fig 4.2 shows the penalized objective value function changes over the iterations when N=5.

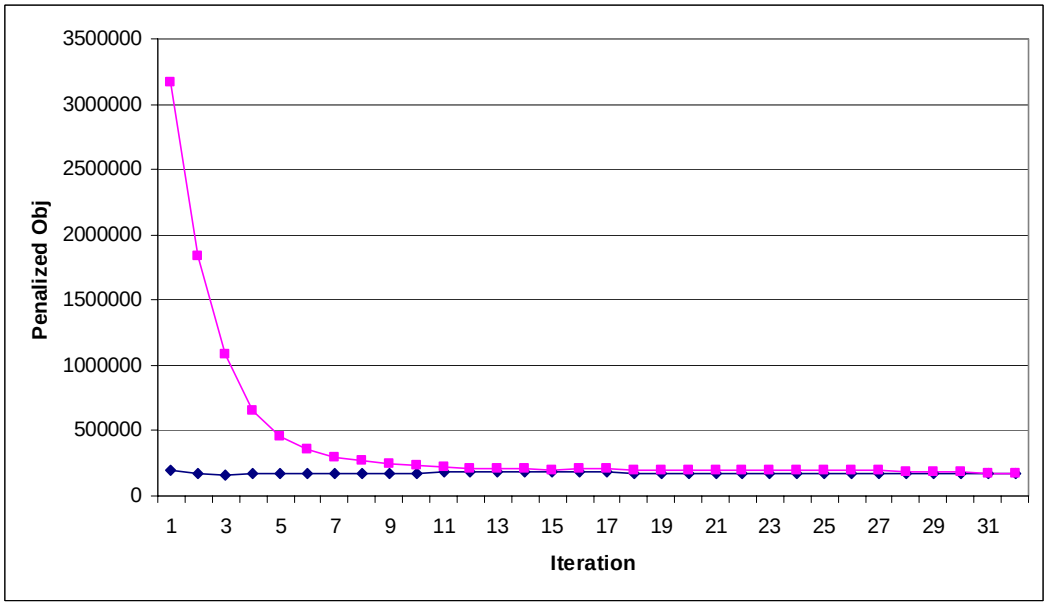


Figure 4.2 Penalized Objective Function When N=5

Two curves represent the iterations starting from two different initial points, one is a feasible solution and another is infeasible in the beginning. They converge relatively fast to very close objective function values. Discussion on objective function value and

optimal design parameters is postponed to chapter 5 where we study the same system using another method.

4.4 Limitations of Dynamic Network Flow Approach

Several aspects of the above approach make it amenable to analysis: 1) the performance measure is a strictly convex function with respect to the design parameters; and 2) performance evaluation can be done through commercial LP solver and exact subgradient information with respect to all parameters can be read out readily after solving the LP. However, there are some limitations of this approach, which we will discuss below from several perspectives.

4.4.1 Representation Accuracy of Delay and Stock Model

Delay and Stock Model is an approximate abstraction of the conveyor behavior. It ignores the spatial evolution of material flow on a conveyor. We show why this leads to inaccurate performance evaluation through two examples. As first example, consider an accumulation conveyor segment with v=50, l=75, d=1. External input arrives at time 0 at a rate 45 and lasts for 8 time units. The discharge rate at the other end of conveyor is 30 throughout the 8 time units. That is $\mathcal{I}(t) = 45 \quad \forall t \in [0,8)$ and $\mathcal{O}(t) = 30 \quad \forall t \in [0,8)$. First we use Batch on Conveyor Model to evaluate the loss volume during this 8 time units.

Since $a(0) = 0$, we have $I(0) = \mathcal{I}(0) = 45$ according to (2-8). The input batch density at time 0 is $d_i(0) = \frac{\mathcal{I}(0)}{v} = \frac{9}{10}$. We have $d_o(0) = 0$ since there is no output batch yet. By time $t = \frac{l}{v} = 1.5$, the input batch at time 0 becomes output batch. So output batch density changes to $d_o(1.5) = \frac{9}{10}$. At this time, we calculate the right derivative of $a(t)$ according to theorem 2-1. We get

$$\dot{a}(1.5) = \frac{d_o(t) * v - O(t)}{d - d_o(t)} = \frac{d_o(1.5) * 50 - 30}{1 - \frac{9}{10}} = 150.$$

Using $\dot{a}(1.5)$, we know $a(t)$ will change from 0 to l=75 in 0.5(=75/150) time units, i.e., it reaches full accumulation at time 2. Event $a(2) = 75$ will trigger the change of $I(t)$ according to (2-8). Effective in-flow rate becomes $I(2) = \text{Min}(\mathcal{I}(2), \mathcal{O}(2)) = 30$. Since no more events will occur until time unit 8, this value will maintain until time 8. In summary, we have

$$I(t) = \begin{cases} 45 & t = [0,2) \\ 30 & t = [2,8) \end{cases}.$$

So the loss volume is

$$L = \int_0^8 F(t)dt = \int_0^8 (\mathcal{I}(t) - I(t))dt = \int_2^8 (45 - 30)dt = 90.$$

Next, we evaluate the loss volume using Delay and Stock Model. We write conditions (4-1) to (4-7) as

$$\tau = \frac{l}{v} = 1.5$$

$$\int_0^t I(\theta - 1.5)d\theta = \int_0^t O(\theta)d\theta + a(\theta) \quad \forall t \in [1.5,8)$$

$$I(t) \le \mathcal{I}(t) = 45 \quad \forall t \in [0,8)$$

$$O(t) \le \mathcal{O}(t) = 30 \quad \forall t \in [0,8)$$

$$I(t) \le vd = 50 \quad \forall t \in [0,8)$$

$$O(t) \le vd = 50 \quad \forall t \in [0,8)$$

$$a(t) \le \frac{l}{d} = 75 \quad \forall t \in [0,8).$$

It is easy to figure out the maximum $I(t)$ will be

$$I(t) = \begin{cases} 45 & t = [0,5) \\ 30 & t = [5,8) \end{cases}.$$

So the loss volume is

$$L = \int_0^8 F(t)dt = \int_0^8 (\mathcal{I}(t) - I(t))dt = \int_5^8 (45-30)dt = 45 .$$

Figure 4.3 is the graphic illustration of the first example.

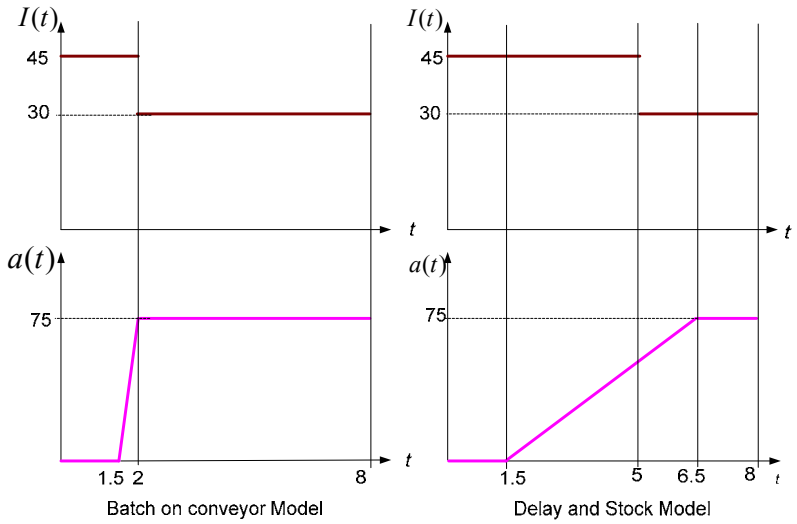


Figure 4.3 Illustration of Error Source 1 of Delay and Stock Model

We can see that the loss volume evaluated through Delay and Stock Model is 45 units less than the loss volume evaluated through Batch on Conveyor Model. We can summarize this error source as follows.

Error source 1: Because the Delay and Stock model does not consider that items take physical space on conveyor while accumulating, the actual blocking might happen earlier than computed time.

In second example, consider a non-accumulation conveyor segment with v=30, l=90, d=1. External input arrives at time 0 at a rate 15 and lasts for 3 time units. Then it changes the rate to 30 and maintains at 30 thereafter. The discharge rate at the EXIT end of conveyor is 15 throughout the investigation time period. As known input, we have

$$\mathcal{I}(t) = \begin{cases} 15 & \forall t \in [0,3) \\ 30 & \forall t \in [3,8) \end{cases} \text{ and}$$

$$\mathcal{O}(t) = 15 \quad \forall t \in [0,11) .$$

We first use Batch on Conveyor Model to evaluate the loss volume within first 8 time units. According to (2-2), we have $v_c(0) = v = 30$ and $I(0) = \text{Min}(\mathcal{I}(0), v_c(0)d) = 15$. Input density and output density at time 0 are $d_i(0) = \frac{I(0)}{v_c(0)} = \frac{1}{2}$ and $d_o(0) = 0$. By time $t = \frac{l}{v} = 3$, the input batch at time 0 becomes output batch. So output batch density changes to $d_o(3) = \frac{1}{2}$. Reevaluate the effective moving speed according to (2-2), we have $v_c(3) = \text{Min}\left(v, \frac{\mathcal{O}(3)}{d_o(3)}\right) = 30$. Using this rate, recalculate the effective in-flow rate and input density, we get $I(3) = \text{Min}(\mathcal{I}(3), v_c(3)d) = 30$ and $d_i(3) = \frac{I(3)}{v_c(3)} = 1$. By time $t = 6$, the input batch at time 3 becomes output batch, i.e., $d_o(6) = d_i(3) = 1$. This would reduce the effective moving speed to $v_c(6) = \text{Min}\left(v, \frac{\mathcal{O}(6)}{d_o(6)}\right) = 15$. Consequently, $I(t)$ also drops to $I(6) = \text{Min}(\mathcal{I}(6), v_c(6)d) = 15$. In summary, we have

$$I(t) = \begin{cases} 15 & t = [0,3) \\ 30 & t = [3,6) \\ 15 & t = [6,8) \end{cases}.$$

Thus, loss volume is

$$L = \int_0^8 F(t)dt = \int_0^8 (\mathcal{I}(t) - I(t))dt = \int_6^8 (30 - 15)dt = 30.$$

Next, we evaluate the loss volume using Delay and Stock Model. For this case, we write conditions (4-1) to (4-7) as

$$\tau = \frac{l}{v} = 3$$

$$I(t-3) = O(t) \quad \forall t \in [3,11)$$

$$I(t) \le \mathcal{I}(t) \quad \forall t \in [0,8)$$

$$O(t) \le \mathcal{O}(t) = 15 \quad \forall t \in [0,8)$$

$$I(t) \le vd = 30 \quad \forall t \in [0,8)$$

$$O(t) \le vd = 30 \quad \forall t \in [0,8).$$

It is easy to figure out the maximum $I(t)$ that satisfies the above conditions will be

$$I(t) = 15 \quad \forall t \in [0,8).$$

The loss volume is

$$L = \int_0^8 F(t)dt = \int_0^8 (\mathcal{I}(t) - I(t))dt = \int_3^8 (30-15)dt = 75.$$

Figure 4.4 is the graphic illustration of the second example.

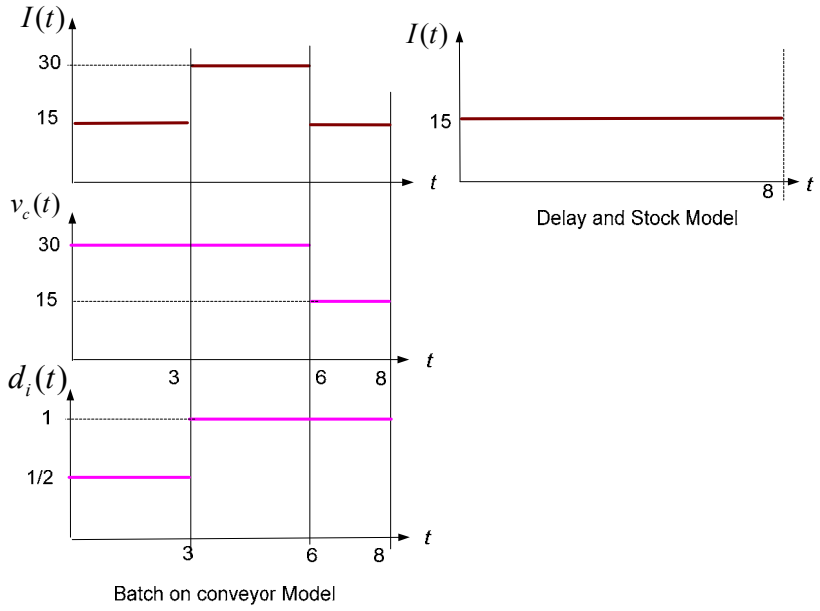


Figure 4.4 Illustration of Error Source 2 of Delay and Stock Model

We can see that the loss volume evaluated through Delay and Stock Model is 45 units more than the loss volume evaluated through Batch on Conveyor Model for the second example. We can state this error source as follows.

Error source 2: Without the material distribution information on the conveyor, Delay and Stock Model always transmits the blocking from EXIT end of non-

accumulation conveyor to ENTER end immediately, while many blockings are avoided through the material flow density change in real system.

These two examples show that Delay and Stock Model has both potentials to overestimate and to underestimate the performance metric which is loss volume in our examples. Fig 4.5 shows the loss volume differences in a series of two-segment experiments where we fix the sorter speed and increase the accumulation position gradually.

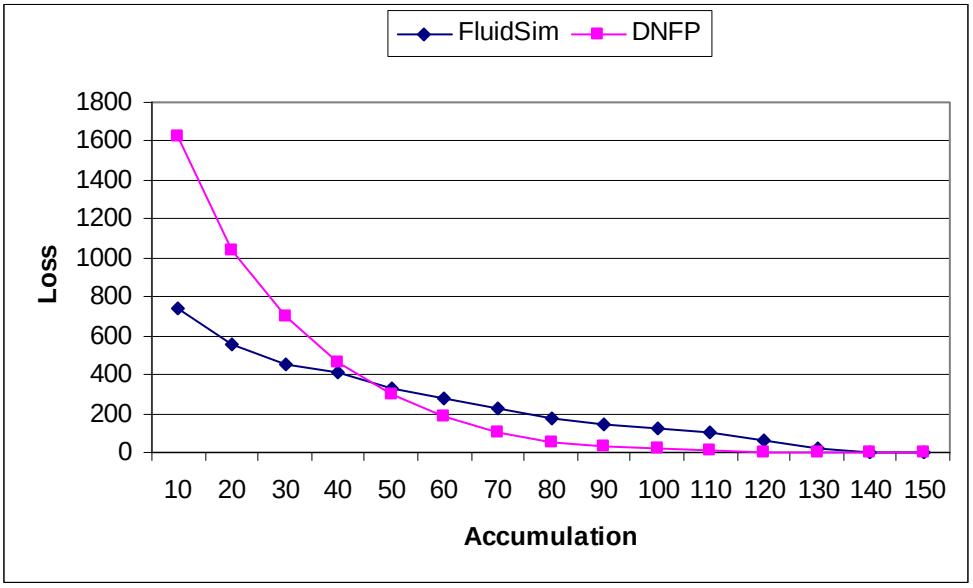


Figure 4.5 Loss Volume Comparison of FluidSim and DNFP

From the figure, we see that the dynamic network flow approach gives much higher loss volume when we have severe blocking effects. We believe this is because of error source 2. Dynamic network flow approach gives lower loss volume when we have relatively large accumulation due to error source 1.

4.4.2 Ignoring the Merge Control Logic Effect

In the dynamic network flow model, we did not model the merge control logic explicitly when the network is more complex than tandem system. The model seeks the optimum control logic according to the minimum loss criterion. The optimum control

logic is normally impossible due to lack of information or complexity in implementation. Also because of this optimal allocation of the capacity, the model lacks the ability to determine where to put the accumulation. We illustrate this through a 10 segment example shown in Fig 4.6.

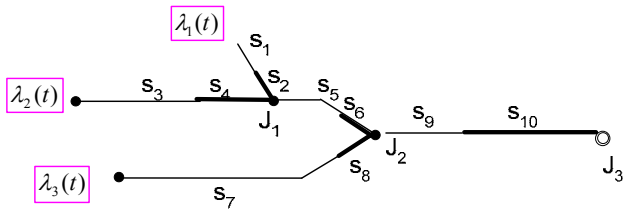


Figure 4.6 Example 10 segment A/M system

We consider two designs with same total accumulation conveyor length but different accumulation distribution. The other design parameters are exact same for two designs. The accumulation configuration and loss quantities for two designs are listed in table 4-1.

Table 4.1 Comparison of Two Designs with Same Total Accumulation Length

Accumulation Conveyor Length	Design1	Design2
s2	30	5
s4	30	5
s6	20	5
s8	30	5
s10	20	110
Total Accumulation Length	130	130
DNFP Loss	349	349
FluidSim Loss	149	182

Dynamic network flow approach returns the same loss for two designs. This is because the system can best utilize the accumulation anywhere in the system by manipulating the flow allocation in each junction at any time. In fluid simulation approach, we can preset the merge control logic as the one we are going to implement. Once this control logic is set, the two designs have different performance measures.

4.4.3 Computational Speed

The uniform discretization algorithm to solve linear problem DP is a pseudo-polynomial algorithm since the running time depends polynomially on the planning horizon T. Problem DP has $p(n+m)$ non-negative variables and $p(2n+m)$ constraints, where $n:=|V|$, $m:=|A|$ and $p = T / \Delta$. Figure 4.7 shows the increase of computational time to solve DP using CPLEX as T increasing for a 10-segment model.

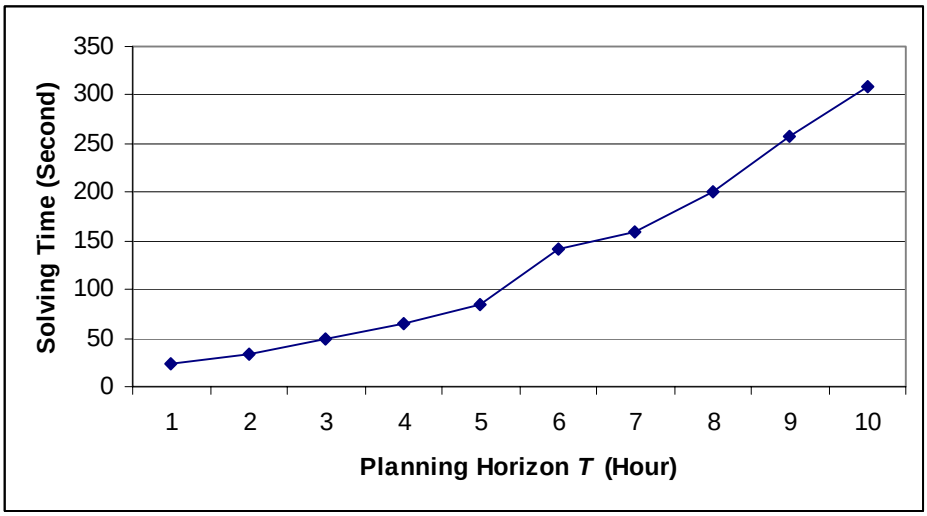


Figure 4.7 DNFP Solving Time for a 10 Segment Network

From running time magnitude on the graph, we can expect current computers can handle the problem of networks with medium complexity over a time horizon less than a few hours. Large computational time required by this method will be a concern when using it to solve more complex network design problem.

4.5 Summary

In this chapter, we presented an approximate conveyor continuous model - Delay and Stock Model. Based on this model we formulate the performance evaluation problem with a deterministic input stream as a dynamic network flow problem. Uniform discretization method is proposed to solve the model to obtain the loss volume. The gradient of loss volume with respect to each parameter can be read out from the dual

variables of the corresponding linear programming problem. We gave numerical example of incorporating this model into the sample path based A/M parametric optimization problem solution framework introduced in chapter 3. The limitations of this approach are also discussed which motivate us to investigate the possibility of using more accurate and fast performance evaluation tool in the stochastic optimization framework.

CHAPTER 5

OPTIMIZATION VIA FLUID SIMULATION

In this chapter, we apply the sample path based optimization solution procedure in conjunction with fluid simulation on A/M network optimization problem. We study the parametric optimization problem of a basic building block of a conveyor system – a two-segment tandem conveyor network consisting of a non-accumulation segment connecting to an accumulation segment. We derive sample derivative of loss volume with respect to the length of accumulation segment using IPA and use the finite difference gradient with respect to sorter speed in the algorithm. Numerical convergence is studied. We discuss the extension of this approach to more complex networks.

5.1 Two-Segment Tandem System

We consider the same two-segment tandem system shown in Figure 4.1. For simplicity, we assume $v_1 = v_2 \equiv v$.

The key calculation in the solution procedure is to evaluate the $L(\lambda, \theta)$ and its gradient with respect to the decision variables for each replication in the sample and at each iteration.

In chapter 2 we present the Batch on Conveyor Model for non-accumulation and accumulation segment. We first see how the characteristic quantities in Batch on Conveyor Model interact in this two segment system. Notation convention is consistent with chapter 2.

First the input flow to segment 1 is just λ function and the maximum discharge rate of segment 2 is just the sorter speed. That is

$$\mathcal{I}_1(t) = \lambda(t) \tag{5-1}$$

$$\mathcal{O}_2(t) = r_3 . \tag{5-2}$$

Segment 2 receives output from segment 1, so the input flow rate of segment 2 is

$$\mathcal{I}_2(t) = vd_{o,1}(t)\,. \tag{5-3}$$

In Batch on Conveyor Model, we assume $\mathcal{I}_1(t) \le vd_1$ and $\mathcal{I}_2(t) < vd_2$. This is equivalent to

$$\lambda(t) \le vd_1 \tag{5-4}$$

and $d_1 < d_2$. (5-5)

We also assume $r_3 < vd_2$, otherwise loss will always be zero. In summary, we have the following assumption about the system parameters.

Assumption 5-1: We assume $v_1 = v_2 \equiv v$, $\lambda(t) \le v \cdot d_1, \forall t \in [0,T]$ and $d_1 < d_2, r_3 < vd_2$.

We can write the maximum discharge rate of segment 1 as

$$\mathcal{O}_1(t) = \begin{cases} vd_2 & \text{if } a_2(t) < l_2 \\ \text{Min}(r_3, vd_2) & \text{if } a_2(t) = l_2 \end{cases} = \begin{cases} vd_2 & \text{if } a_2(t) < l_2 \\ r_3 & \text{if } a_2(t) = l_2 \end{cases}. \tag{5-6}$$

According to (2-4), the overflow function is

$$F_1(t) = \text{Max}\left(0, \mathcal{I}_1(t) - \frac{\mathcal{O}_1(t)}{d_{o,1}(t)} d_1\right) = \text{Max}\left(0, \lambda(t) - \frac{\mathcal{O}_1(t)}{d_{o,1}(t)} d_1\right)$$

The above $F_1(t)$ and $\mathcal{O}_1(t)$ expressions lead to the following lemma.

Lemma 5-1: $F_1(t) > 0$ implies $a_2(t) = l_2$.

Proof: if $a_2(t) < l_2$, then $\mathcal{O}_1(t) = vd_2$. We have $F_1(t) = \text{Max}\left(0, \lambda(t) - \frac{vd_2}{d_{o,1}(t)} d_1\right)$.

We also know $\frac{vd_2}{d_{o,1}(t)} d_1 \ge vd_2 > vd_1 \ge \lambda(t)$, thus, we arrive at $F_1(t) = 0$. □

Lemma 5-1 indicates overflow only happens when accumulation segment is full. Using Lemma 5-1, we can write $F_1(t)$ more explicitly as

$$F_1(t) = \begin{cases} 0 & \text{if } a_2(t) < l_2 \\ \text{Max}\left(0, \lambda(t) - \dfrac{r_3}{d_{o,1}(t)} d_1\right) & \text{if } a_2(t) = l_2 \end{cases} \quad (5\text{-}7)$$

Loss volume L can be directly computed by integrating $F_1(t)$,

$$L = \int_0^T \left(\mathcal{I}_1(t) - I_1(t)\right)dt = \int_0^T F_1(t)dt \,. \quad (5\text{-}8)$$

5.2 IPA Sample Derivative Estimation

In this section, we will use infinitesimal perturbation analysis (IPA) to estimate the first order partial derivatives of the loss L with respect to l_2. This technique allows us to estimate gradients with one sample path. Comprehensive discussions of IPA and its application can be found in Glasserman (1991) and Ho and Cao (1991).

IPA derives $\dfrac{\partial L}{\partial l_2}$ by comparing a *nominal* sample path under some accumulation length l_2 and a *perturbed* sample path under $l_2 + \Delta l$. For simplicity, we only consider the case where $\Delta l > 0$, leading to an estimate of the right sample derivative of L; the case $\Delta l < 0$ is similar, leading to an estimate of the left derivative of L. In representation, we add a prime to the quantity name describing the perturbed sample path.

Lemma 5-1 implies overflow only occurs when the accumulation is full. We call a period (α, β) a *full period* if accumulation becomes full at time α and maintain full until β. We call a period (ξ, η) an *overflow period* if $F_1(t)$ becomes positive at time ξ and maintain as a constant until η. An overflow period will always be within a full period. If there exists at least one overflow period in a full period, we call this full period the *overflow full period*. Suppose there are K overflow full periods denoted by $(\alpha_1, \beta_1),..,(\alpha_K, \beta_K)$ in increasing order over [0,T] and for each $k = 1,...,K$, there are M_k overflow periods denoted by $(\xi_{k,1}, \eta_{k,1}),..,(\xi_{k,M_k}, \eta_{k,M_k})$. Then we can write the loss volume L as

$$L = \sum_{k=1}^{K} \sum_{m=1}^{M_k} F_1(\xi_{k,m})(\eta_{k,m} - \xi_{k,m}) \tag{5-9}$$

Before introducing assumption 5-2, we need first to introduce a concept called *regeneration point* and a lemma regarding how a regeneration point is created. We define a r*egeneration point* as a time point *t* at which the batch condition in the perturbed sample path $\{B^1(t)', B^2(t)', .., B^{K'}(t)'\}$ is exactly the same as the batch sequence at the same time in the nominal sample path $\{B^1(t), B^2(t), .., B^K(t)\}$. Mathematically, this means $K = K'$ and $l^n(t) = l^n(t)', d^n(t) = d^n(t)', x^n(t) = x^n(t)'$ for all *n*=1,..,*K*.

The following lemma gives the condition that we can observe a regeneration point in the perturbed sample path.

Lemma 5-2: $t_0 + \frac{l_1 + l_2}{v}$ is a regeneration point in the perturbed sample path if the interval $\left[t_0, t_0 + \frac{l_1 + l_2}{v}\right]$ satisfies the condition that $a_2(t) < l_2 \ \ \forall t \in \left[t_0, t_0 + \frac{l_1 + l_2}{v}\right]$ and $a_2\left(t_0 + \frac{l_1 + l_2}{v}\right) = 0$.

Proof: Figure 5.1 illustrates the batch condition at time t_0 and $t_0 + \frac{l_1 + l_2}{v}$. A block with different shade represents different batch.

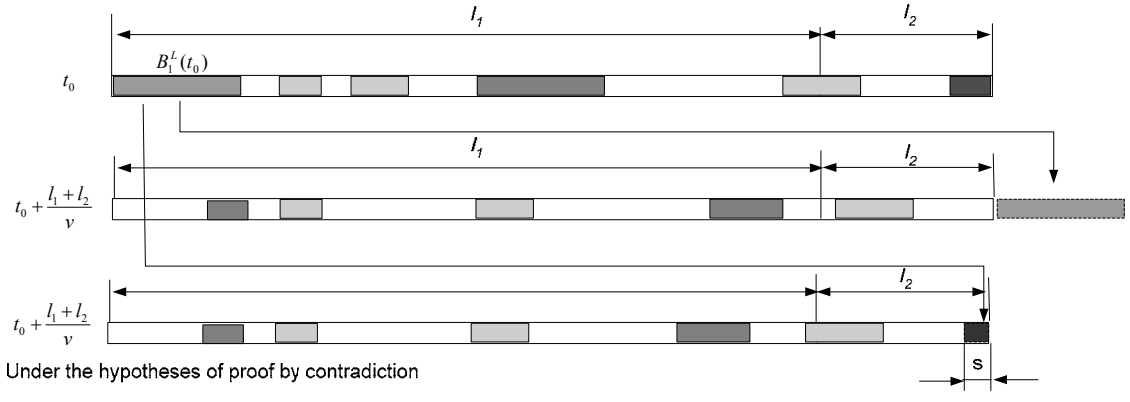


Figure 5.1 Illustration of Lemma 5-2 Proof

Let $B_1^L(t_0)$ be the last batch at the conveyor 1 at time t_0, first we show $B_1^L(t_0)$ will completely leave the conveyor 2 by time $t_0 + \frac{l_1 + l_2}{v}$. Suppose not; then there is a positive length *s* between the back end of $B_1^L(t_0)$ and the EXIT end of conveyor 2. Since the back end of $B_1^L(t_0)$ always moves forward at speed v unless it meets the accumulation interface. So we have $a_2\left(t_0 + \frac{l_1 + l_2}{v}\right) = s$ which contradicts the condition $a_2\left(t_0 + \frac{l_1 + l_2}{v}\right) = 0$. Since the $B_1^L(t_0)$ is the last batch at conveyor 1 at time t_0, the above statement implies all batches generated before t_0 will leave the conveyor by the time $t_0 + \frac{l_1 + l_2}{v}$. So next we only need to show the new generated batches during the interval $\left[t_0, t_0 + \frac{l_1 + l_2}{v}\right]$ in the perturbed case are the same as generated in the nominal case.

Any batch generated during an interval $[\tau + \Delta\tau] \subseteq \left[t_0, t_0 + \frac{l_1 + l_2}{v}\right]$ can be written as $B_{i,1}(\tau) : [\Delta t v_{c,1}(\tau), d_{i,1}(\tau), \Delta t v_{c,1}(\tau)]$. According to equation (2-2), $v_{c,1}(\tau) = \text{Min}\left(v, \frac{\mathcal{O}_1(\tau)}{d_{o,1}(\tau)}\right)$. Since $a_2(\tau) < l_2$, maximum discharge rate of conveyor 1 is just the maximum admitting capacity conveyor 2, i.e., $\mathcal{O}_1(\tau) = v d_2$. Then we have $\frac{\mathcal{O}_1(\tau)}{d_{o,1}(\tau)} = \frac{v d_2}{d_{o,1}(\tau)} > v$ which leads to $v_{c,1}(\tau) = v$. Also according to (2-6) $d_{i,1}(\tau) = \frac{\mathcal{I}_1(t)}{v} = \frac{\lambda(t)}{v}$. Since $d_{i,1}(\tau)$ and $v_{c,1}(\tau)$ only depend on $\lambda(t)$ and *v*. The input batch of the nominal path should be the same as the input batch of the perturbed path, i.e.

$B_{i,1}(\tau) = B_{i,1}(\tau)'$. Since this applies to any interval $[\tau + \Delta\tau] \subseteq \left[t_0, t_0 + \frac{l_1 + l_2}{v} \right]$, this completes the proof. □

In order to convert the problem to a manageable scale, we make the following assumption so that we are able to isolate each overflow full period from others and conduct IPA analysis on them one by one.

Assumption 5-2: A regeneration point exists between every two consecutive overflow full periods.

Now we focus on a particular overflow full period (α_k, β_k). Suppose during this period, output batch density $d_{0,1}(t)$ changes S_k times at time points $u_{k,1},..,u_{k,S_K}$ in increasing order and $\lambda(t)$ change H_k times at time points $w_{k,1},..,w_{k,H_K}$ in increasing order. Denote the corresponding output batch densities as $d_{0,1}(\alpha_k), d_{0,1}(u_{k,1}),..., d_{0,1}(u_{k,S_K-1})$ and input rates as $\lambda(\alpha_k), \lambda(w_{k,1}),..., \lambda(w_{k,H_K})$. We assume no two events may occur at the same time. So $\alpha_k, u_{k,1}..,u_{k,S_k}, w_{k,1},..,w_{k,H_k}$ are all distinct time points. Figure 5-2 shows such an overflow full period with $S_k = 3$ and $H_k = 2$.

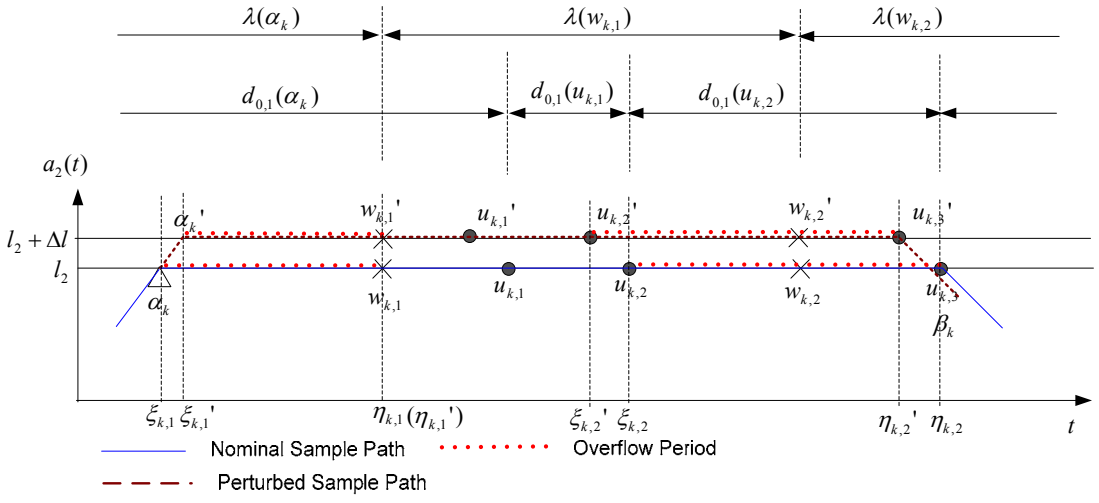


Figure 5.2 Illustration of an Overflow Full Period

$a_2(t) = l_2 \ \forall t \in [\alpha_k, \beta_k]$ means the derivative of $a_2(t)$ will keep at zero during this time period, i.e., $\dot{a}_2(t^+) = 0 \ \forall t \in [\alpha_k, \beta_k)$. This requires $\mathcal{I}_2(t) \geq O_2(t) \forall t \in [\alpha_k, \beta_k)$ from equation (2-11). Substitute $\mathcal{I}_2(t)$ and $O_2(t)$ with (5-3) and r_3, we have $vd_{o,1}(t) \geq r_3 \ \forall t \in [\alpha_k, \beta_k)$. So we have

$$d_{0,1}(\alpha_k) \geq \frac{r_3}{v} \text{ and } d_{0,1}(u_{k,s}) \geq \frac{r_3}{v} \text{ for } s = 1,..,s_k - 1. \tag{5-10}$$

Since an overflow full period ends because of a drop of $d_{0,1}(t)$, so β_k must be the last $d_{0,1}(t)$ switch time, i.e.,

$\beta_k = u_{k,s_K}$.

Next we see how these α, u, w points change in the perturbed sample path. This was summarized into following three lemmas.

Lemma 5-3 : $\lim_{\Delta l \to 0} \frac{\alpha_k' - \alpha_k}{\Delta l} = \frac{(d_2 - d_{o,1}(\alpha_k))}{d_{o,1}(\alpha_k)v - r_3}$

Proof: Because of assumption 5-2. We can find a time $t_k^0 < \alpha_k$ at which the batch conditions on two paths are the same. Particularly, we would like this t_k^0 to satisfy followings conditions.

$$d_{o,1}(\alpha_k) = d_{o,1}(t_k^0) = d_2^2(t_k^0) \tag{5-11}$$

$$\alpha_k - t_k^0 = \frac{l_2 - a_2(t_k^0)}{\dot{a}_2(t_k^{0+})} \tag{5-12}$$

These conditions imply that after time t_k^0, the accumulation interface moves towards the junction of segment 1 and 2 at a constant speed $\dot{a}_2(t_k^{0+})$ until it finally reaches the junction. In the perturbed path, when Δl is small enough $a_2(t)$ will reach $l_2 + \Delta l$ at time α_k' before the output batch density of segment 1 changes. We have

$$\alpha_k' - t_k^0 = \frac{l_2 + \Delta l - a_2(t_k^0)'}{\dot{a}_2(t_k^{0+})'}.$$

Because the batch conditions are the same at time t_k^0, we have

$$\alpha_k' - t_k^0 = \frac{l_2 + \Delta l - a_2(t_k^0)}{\dot{a}_2(t_k^{0+})} \tag{5-13}$$

Subtract (5-12) from (5-13), we have

$$\alpha_k' - \alpha_k = \frac{\Delta l}{\dot{a}_2(t_k^{0+})} = \frac{\Delta l(d_2 - d_2^2(t_k^0))}{d_2^2(t_k^0)v - O_2(t)} = \frac{\Delta l(d_2 - d_{o,1}(t_k^0))}{d_{o,1}(t_k^0)v - r_3} = \frac{\Delta l(d_2 - d_{o,1}(\alpha_k))}{d_{o,1}(\alpha_k)v - r_3}.$$

The equivalence is established through equation (2-11) and (5-11).

This leads to $\lim_{\Delta l \to 0} \frac{\alpha_k' - \alpha_k}{\Delta l} = \frac{(d_2 - d_{o,1}(\alpha_k))}{d_{o,1}(\alpha_k)v - r_3}$ □

Lemma 5-4 $\lim_{\Delta l \to 0} \frac{w_{k,h}' - w_{k,h}}{\Delta l} = 0$ for $h = 1,..,H_k$ (5-14)

This is because the change of $\lambda(t)$ is exogenous event to the system. So the event time is independent of system parameters.

Lemma 5-5 $\lim_{\Delta l \to 0} \frac{u_{k,s}' - u_{k,s}}{\Delta l} = -\frac{d_2}{r_3}$ for $s = 1,..,S_k$ (5-15)

Proof: Figure 5.3 illustrates the proof.

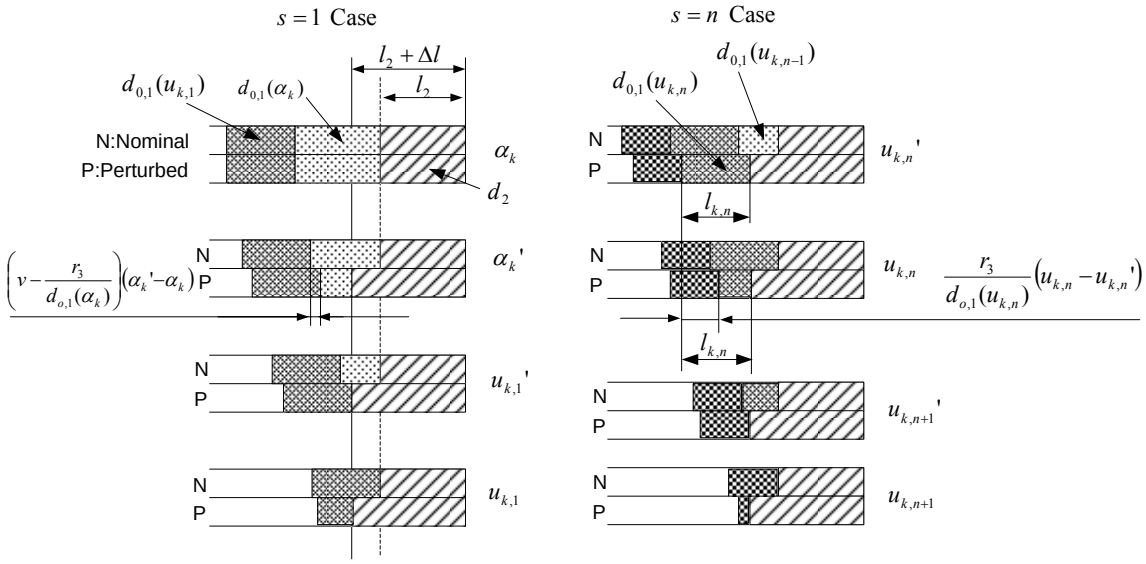


Figure 5.3 Proof of Lemma 5-5

First look at the case when $s = 1$. $u_{k,1}$ is the time that the back end of the batch that has density $d_{o,1}(\alpha_k)$ reaches the two segment junction. Because of assumption 5-2, by the time α_k, this back end has the same position in the perturbed case as in the nominal case. From α_k to α_k', this end moves forward at a reduced speed $v_{c,1}(\alpha_k) = \dfrac{r_3}{d_{o,1}(\alpha_k)}$ in the nominal case, while it still moves at speed *v* in the perturbed case. So by the time α_k', the positions of this end in two cases already has a gap of value $\left(v - \dfrac{r_3}{d_{o,1}(\alpha_k)}\right)(\alpha_k' - \alpha_k)$. After time α_k', this end will move forward in speed $\dfrac{r_3}{d_{o,1}(\alpha_k)}$ in both cases. Since the junction is moved by Δl to the non-accumulation segment side in perturbed case, we derive the total time difference as

$$u_{k,1}'-u_{k,1} = -\frac{\Delta l + \left(v - \frac{r_3}{d_{o,1}(\alpha_k)}\right)(\alpha_k'-\alpha_k)}{\frac{r_3}{d_{o,1}(\alpha_k)}}$$

$$= -\frac{\Delta l\left(1 + \left(v - \frac{r_3}{d_{o,1}(\alpha_k)}\right)\left(\frac{(d_2 - d_{o,1}(\alpha_k))}{d_{o,1}(\alpha_k)v - r_3}\right)\right)}{\frac{r_3}{d_{o,1}(\alpha_k)}}$$

$$= -\frac{\Delta l\left(1 + \frac{d_2 - d_{o,1}(\alpha_k)}{d_{o,1}(\alpha_k)}\right)}{\frac{r_3}{d_{o,1}(\alpha_k)}}$$

$$= -\Delta l\left(\frac{d_2}{r_3}\right)$$

So $\lim_{\Delta l \to 0} \frac{u_{k,1}'-u_{k,1}}{\Delta l} = -\frac{d_2}{r_3}$.

Suppose the statement is true for $s = n$, that means we have $u_{k,n}'-u_{k,n} = -\frac{d_2}{r_3}\Delta l$.

We want to prove the statement is also true for $s = n+1$.

$u_{k,n+1}'$ is the time that the back end of the batch that has density $d_{o,1}(u_{k,n})$ reaches the two segment junction in the perturbed case. During time period $[u_{k,n}', u_{k,n}]$, the back end of this batch moves forward of a distance $\frac{r_3}{d_{o,1}(u_{k,n})}\left(u_{k,n} - u_{k,n}'\right)$. At time $u_{k,n+1}$, this back end arrived at the junction in nominal case. From this time on, this end will move at the same speed of $\frac{r_3}{d_{o,1}(u_{k,n})}$ until reaching the junction. In this nominal case, the distance needs to travel is the length of this batch denoted as $l_{k,n}$, while in perturbed case the distance needs to travel is $l_{k,n} - \frac{r_3}{d_{o,1}(u_{k,n})}\left(u_{k,n} - u_{k,n}'\right)$. So

$$u_{k,n+1}'-u_{k,n+1}=-\frac{\frac{r_3}{d_{0,1}(u_{k,n})}\left(u_{k,n}-u_{k,n}'\right)}{\frac{r_3}{d_{0,1}(u_{k,n})}}=u_{k,n}-u_{k,n}'=-\frac{d_2}{r_3}\Delta l$$

That is $\lim_{\Delta l\to 0}\frac{u_{k,n+1}'-u_{k,n+1}}{\Delta l}=-\frac{d_2}{r_3}$. This completes the proof. □

The above three lemmas give the derivatives of three types of switch points with respect to accumulation segment length. We derive them because the starting point or ending point of an overflow period must be a switch point of three types as we show in the following.

Let $(\xi_{k,m},\eta_{k,m})$ be any overflow period in full period (α_k,β_k). From (5-7), we see $F_1(t)$ is piece-wise constant function and its value only depends on $a_2(t)$, $\lambda(t)$ and $d_{0,1}(t)$. So $\xi_{k,m}$ and $\eta_{k,m}$ must be some α,u,w points. Formally we have

$$\xi_{k,m}\in\{\alpha_k,u_{k,1}..,u_{k,S_k-1},w_{k,1},..,w_{k,H_k}\}$$

$$\eta_{k,m}\in\{u_{k,1}..,u_{k,S_k},w_{k,1},..,w_{k,H_k}\}$$

Furthermore we have the following lemma.

Lemma 5-6 $F_1(\xi_{k,m})=F_1(\xi_{k,m})'$ for $m=1,...,M_k$

Proof: By definition, we have

$d_{o,1}(\alpha_k)=d_{o,1}(\alpha_k');d_{o,1}(u_{k,s})=d_{o,1}(u_{k,s}')$ for $s=1,..,S_k$

$\lambda(w_{k,h})=\lambda(w_{k,h}')$ for $h=1,..,H_k$. (5-16)

Because $\alpha_k,u_{k,1}..,u_{k,S_k},w_{k,1},..,w_{k,H_k}$ are all distinct time points, these is positive distance between each consecutive pair. From lemma 5-3, 5-4 and 5-5, we can have a small enough Δl that guarantees that points $\alpha_k',u_{k,1}',..,u_{k,S_k}',w_{k,1}',..,w_{k,H_k}'$ appear in the same time sequence in perturbed path as $\alpha_k,u_{k,1}..,u_{k,S_k},w_{k,1},..,w_{k,H_k}$ in nominal path. This implies

$$\lambda(\alpha_k) = \lambda(\alpha_k'); \lambda(u_{k,s}) = \lambda(u_{k,s}') \text{ for } s = 1,..,S_k$$

$$d_{o,1}(w_{k,h}) = d_{o,1}(w_{k,h}') \text{ for } h = 1,..,H_k . \quad (5\text{-}17)$$

(5-16) and (5-17) imply $F_1(\xi_{k,m}) = F_1(\xi_{k,m})'$ because of $\xi_{k,m} \in \{\alpha_k, u_{k,1}..,u_{k,S_k-1}, w_{k,1},..,w_{k,H_k}\}$. □

Applying lemma 5-6, we can write the right sample derivative of L as

$$\begin{aligned}
\frac{\partial L}{\partial l_2} = \lim_{\Delta l \to 0} \frac{L'-L}{\Delta l} &= \lim_{\Delta l \to 0} \frac{\sum_{k=1}^{K}\sum_{m=1}^{M_k} F_1(\xi_{k,m})'(\eta_{k,m}'-\xi_{k,m}') - \sum_{k=1}^{K}\sum_{m=1}^{M_k} F_1(\xi_{k,m})(\eta_{k,m}-\xi_{k,m})}{\Delta l} \\
&= \lim_{\Delta l \to 0} \frac{\sum_{k=1}^{K}\sum_{m=1}^{M_k} F_1(\xi_{k,m})\left((\eta_{k,m}'-\eta_{k,m}) - (\xi_{k,m}'-\xi_{k,m})\right)}{\Delta l} \\
&= \sum_{k=1}^{K}\sum_{m=1}^{M_k} F_1(\xi_{k,m}) \lim_{\Delta l \to 0} \frac{(\eta_{k,m}'-\eta_{k,m})}{\Delta l} - \sum_{k=1}^{K}\sum_{m=1}^{M_k} F_1(\xi_{k,m}) \lim_{\Delta l \to 0} \frac{(\xi_{k,m}'-\xi_{k,m})}{\Delta l} \\
&= \sum_{k=1}^{K}\sum_{m=1}^{M_k} F_1(\xi_{k,m}) \left(\frac{\partial \eta_{k,m}}{\partial l_2} - \frac{\partial \xi_{k,m}}{\partial l_2} \right)
\end{aligned} \quad (5\text{-}18)$$

$\frac{\partial \eta_{k,m}}{\partial l_2}$ and $\frac{\partial \xi_{k,m}}{\partial l_2}$ can be computed from one of equations (5-13),(5-14) and (5-15) based on whether it is α point, u point or w point. The algorithm to compute them is summarized in next section.

The above analysis is based on assumption 5-2. Lastly, we show through an example that between two consecutive regeneration points, it is possible for the perturbation effect on one overflow full period to propagate and affect the successive overflow full periods.

Fig 5.4 gives such an example.

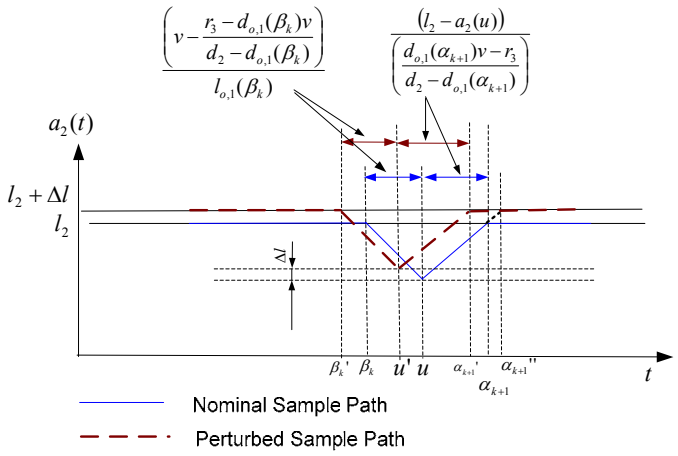


Figure 5.4 Illustration of Perturbation Propagation

The figure shows the transition from one overflow full period (α_k, β_k) to another overflow full period $(\alpha_{k+1}, \beta_{k+1})$ of two sample paths. Overflow full period (α_k, β_k) ends because of $vd_{o,1}(\beta_k) < r_3$, so we can think the two sides of batch $B_{o,1}(\beta_k)$ move forward at different speeds. The front end is just the accumulation interface which moves at speed $\dot{a}_2(\beta_k) = \frac{r_3 - d_{o,1}(\beta_k)v}{d_2 - d_{o,1}(\beta_k)}$. The back end still moves at speed *v*. Suppose the two ends meet at time u and the next batch arriving at accumulation interface has high density and eventually triggers the next overflow full period $(\alpha_{k+1}, \beta_{k+1})$. As the figure shows, it is easy to derive that

$$u - \beta_k = \mu' - \beta_k = \frac{\left(v - \frac{r_3 - d_{o,1}(\beta_k)v}{d_2 - d_{o,1}(\beta_k)}\right)}{l_{o,1}(\beta_k)} \text{ and}$$

$$\alpha_{k+1} - u = \alpha_{k+1}' - u' = \frac{(l_2 - a_2(u))}{\frac{d_{o,1}(\alpha_{k+1})v - r_3}{d_2 - d_{o,1}(\alpha_{k+1})}}.$$

So we have $\alpha_{k+1} - \alpha_{k+1}' = \beta_k - \beta_k' = \Delta l \frac{d_2}{r_3}$. Our IPA algorithm will estimate α_{k+1}' as $\alpha_{k+1}'' = \alpha_k + \frac{(d_2 - d_{o,1}(\alpha_k))}{d_{o,1}(\alpha_k)v - r_3}\Delta l$. This example shows when assumption 5-2 is unsatisfied our algorithm can introduce error and gives an approximate sample derivative. We will study the effect of this assumption in numerical analysis toward the end of the chapter.

5.2.3 Algorithm

To implement the above derivative estimation, during each simulation run, we need to keep certain records so that we can have a full knowledge of $\lambda(t), I_1(t), a_2(t)$ and $d_{o,1}(t)$ four functions by the end of simulation. Then we can identify all overflow periods by comparing $\lambda(t)$ and $I_1(t)$. After that, we scan each overflow period and decide the case of starting point and ending point of each overflow period, whether it is α point, u point or w point. Then we can calculate the sample derivative with respect to accumulation length according to (5-18). Detail algorithm is illustrated through Fig 5.5.

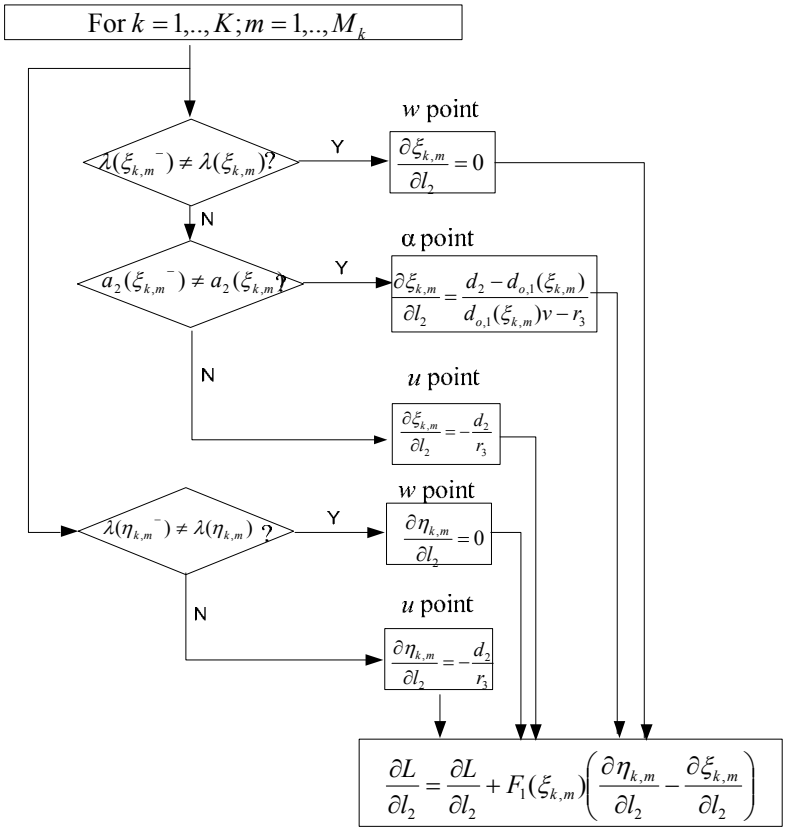


Figure 5.5 Two-Segment Tandem Network Sample Derivative Algorithm

5.3 Numerical Optimization Experiment

We apply the above methodology to the same numerical example used in the chapter 4.

$L^j(\lambda^j, \theta_k)$ is evaluated through fluid simulation. $\frac{\partial L^j(\lambda^j, \theta^k)}{\partial l_2}$ is obtained through the sample derivative estimation algorithm illustrated in Fig 5.5. The derivative with respect to sorter speed $\frac{\partial L^j(\lambda^j, \theta^k)}{\partial r_3}$ is estimated through finite difference method.

Convexity of Loss Volume Function

Gradient based solution method requires that average loss function $g_N(\theta) = N^{-1}\sum_{j=1}^{N} L(\lambda^j, \theta)$ is a convex function of design parameter θ. In chapter 4 solution procedure, the convexity of loss volume of each replication is guaranteed through linear programming theorem. For fluid simulation, we observe from the experiments that the loss volume function of each replication could be non-convex with small bumps in the curve. However, these small bumps tend to even out in average loss volume curves and result in convex average loss functions. Figure 5.6 shows example loss volume functions with respect to accumulation length l_2 and sorter speed r_3 in replication level and average level. We leave further investigation and formal proof to the future study. Here we simply assume the average loss function is convex.

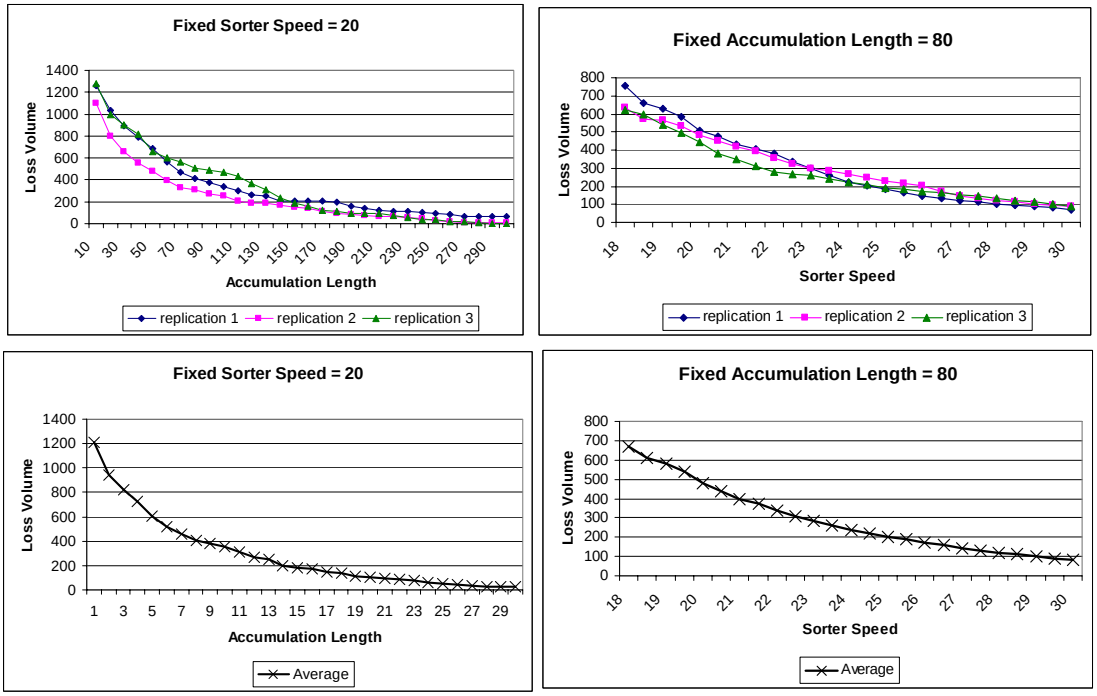


Figure 5.6 Convexity of Sample Average Loss Volume

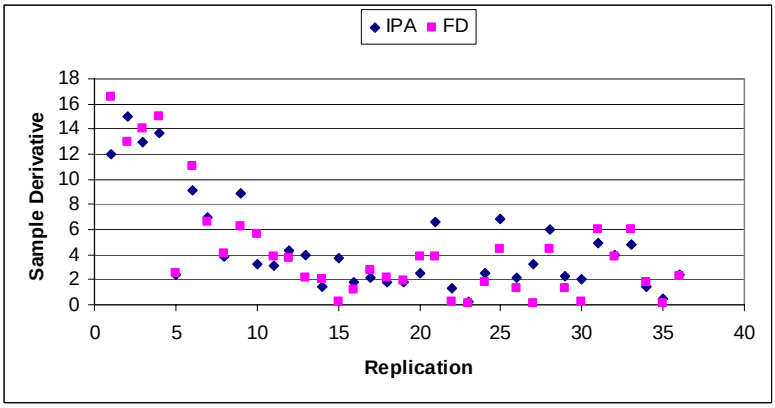
Accuracy of IPA Sample Derivative Estimation

We conduct an analysis on the accuracy of our approximate IPA sample derivative estimation approach. We selected 12 cases (each case representing a set of

design parameters). For each case, we calculate the IPA sample derivative estimator and Finite Difference (FD) sample derivative estimator for three replications and also the average over these three replications. In total we have 36(12*3) replication data (see Appendix D for the data). We compare the difference between these two estimators from replication level as well as sample average level. Data are shown graphically in Figure 5.7 and statistical summary data is listed on Table 5.1.

Table 5.1 Comparison of IPA and FD Sample Derivative Estimators

	Mean Error	Error Std	95%CI	Correlation
Replication Level (36)	0.26	1.66	0.54	0.93
Average Level (12)	0.3	1.1	0.6	0.97



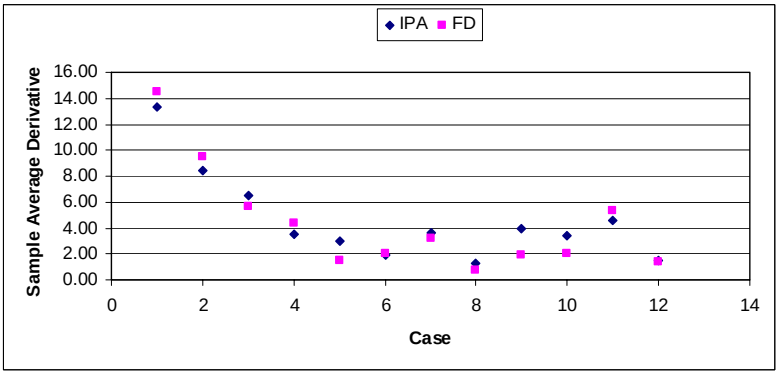


Figure 5.7 Replication Level and Average Level Sample Derivative Comparison

In both levels, large correlation coefficients of two groups of data illustrate the close association of IPA and FD sample derivative estimators. The mean difference is

close to 0. This indicates the IPA estimator is not seriously biased. 95% confidence interval of difference is no bigger than 1 on both levels. We consider the performance of this IPA estimator is satisfactory.

Convergence of a Sample Optimization

Fig 5.8 shows the penalized objective value function changes over the iterations for a sample with *N*=5. Because of the way we choose a_k, we have geometric convergence when optimizing the current sample. Since subgradient method is not a strict decent method, we keep track of the best solution found so far along the iterations.

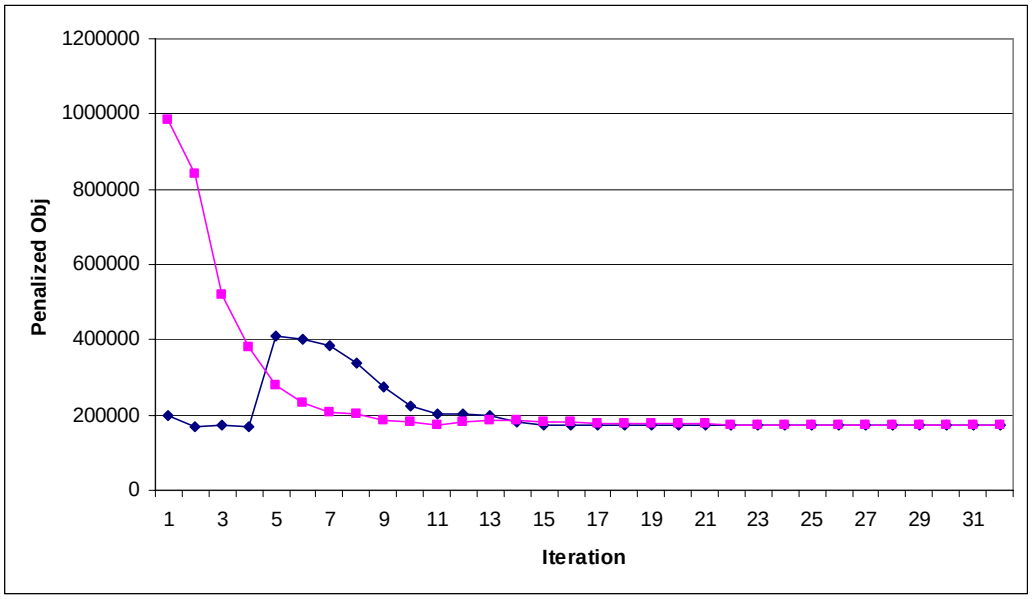


Figure 5.8 Penalized Objective Function When N=5

Two curves represent the iterations starting from two different initial points, one is a feasible solution and another is infeasible in the beginning. Table 5.2 compares the optimal solution we obtained from this chapter fluid simulation approach and the one we obtained from chapter 4 dynamic network flow approach. For this small size example we can also find near optimal solution from total enumeration. Specifically, we enumerate the total solution space discretely by using a step size for each decision variable and choose the minimum feasible design as near optimal solution. In experiments, we choose accumulation length step size as 10 and sorter speed step size as 0.5.

Table 5.2 Optimal Solution Comparison

	Initial Point 1		Initial Point 2		Total Enumeration
	FluidSim	DNFP	FluidSim	DNFP	Solution
Objective Function	172770	169100	173000	169440	172950
Sorter Speed	25.5	25.4	26.6	26.4	25.5
Accumulation Length	139	120	125	108	140
Non-Accumulation Length	561	580	575	592	560

We can see from the data, the solutions from fluid simulation method are very close the solution obtained from total enumeration. DNFP approach also returned with similar designs in term of sorter speed, they only tend to use less accumulation lengths. This is because of the approximate representation of Delay and Stock Model.

Resampling

Sample path based optimization is often implemented as applying the deterministic convex programming algorithm for increasing value of N; terminate the process when convergence of optimal objective function value is conjectured. Many methods can be used for testing convergence ranging from simple graphic methods to formal statistical tests such as Geweke test (Geweke, 1992). In the experiment, we did not apply the algorithm with N larger than 5. Figure 5.9 shows the quick convergence of discrete total enumeration solution for this example.

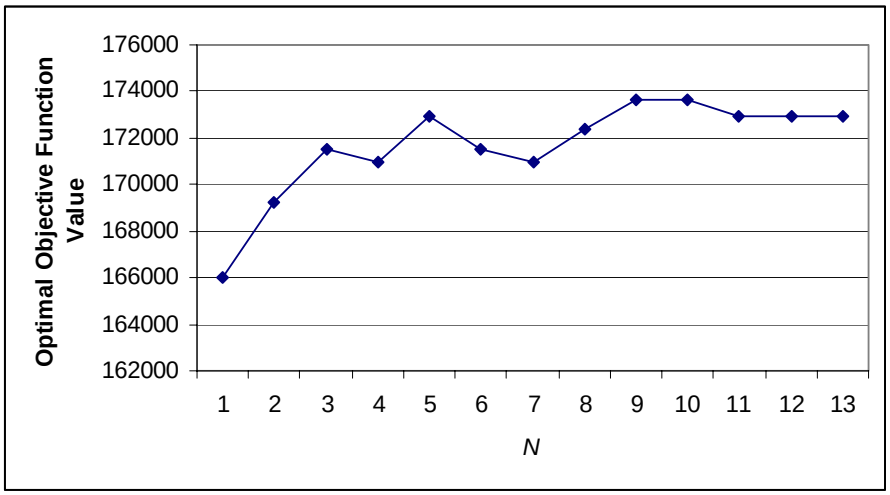


Figure 5.9 Convergence of Optimal Objective Function Value

5.4 Extension to More Complex Network

In this section, we briefly discuss the idea of how to extend the above optimization approach to more complex conveyor networks. Specifically, we focus on how the IPA derivative estimation is conducted on more complex network since the sample path based optimization solution procedure does not change.

We illustrate the idea by an example. A six-segment conveyor network and its Petri Net abstraction are shown on Fig 5.10. It has two inputs and we can think the network consisting of three two-segment building blocks. As in the segment Batch on Conveyor Model, we assume $v_1 = v_2, v_3 = v_4, v_5 = v_6$, $d_1 < d_2, d_3 < d_4, d_5 < d_6$ and $\lambda_1(t) < v_1 d_1, \lambda_1(t) < v_3 d_3$. The design parameters are three accumulation conveyor lengths and the sorter speed $\theta = \{l_2, l_4, l_6, r_7\}$.

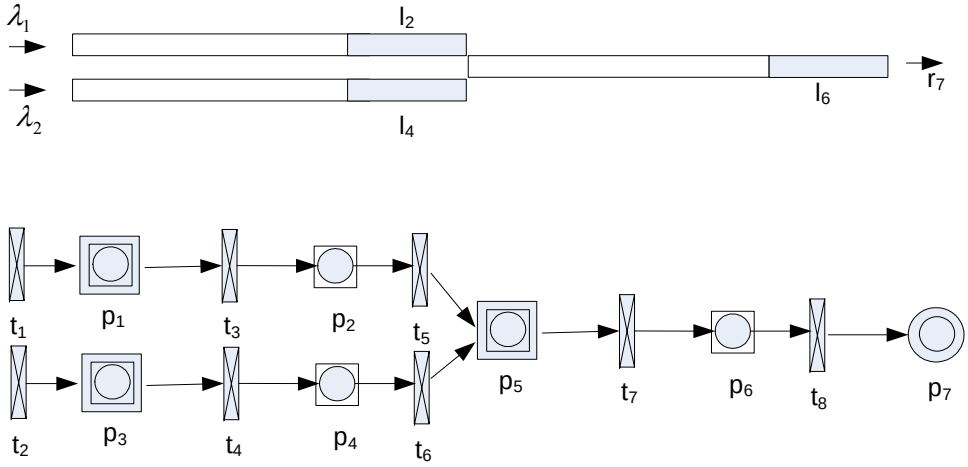


Figure 5.10 Six-Segment Conveyor Network

We can still identify full periods and overflow periods for each input. Now these periods should be indexed by input numbers. The loss volume can be expressed as

$$L = \sum_{i=1}^{2} \sum_{k=1}^{K_i} \sum_{m=1}^{M_K} F_i(\xi_{i,k,m})(\eta_{i,k,m} - \xi_{i,k,m}) . \qquad (5\text{-}19)$$

According to Batch on Conveyor Model, we still have

$$F_1(t) = \text{Max}\left(0, \lambda_1(t) - \frac{\mathcal{O}_1(t)}{d_{o,1}(t)} d_1\right)$$

$$\mathcal{O}_1(t) = \begin{cases} v_1 d_2 & \text{if } a_2(\text{t}) < l_2 \\ \text{Min}\,(\mathcal{O}_2(t), v_1 d_2) & \text{if } a_2(\text{t}) = l_2 \end{cases}$$

Suppose conveyor 2 and 4 merge to conveyor 5 according to processor share merge logic with parameter (1/2, 1/2) (see Appendix A for logic description). Mathematically, the merge logic can be stated as

$$\begin{cases} \mathcal{O}_2(t) = \mathcal{O}_4(t) = \frac{1}{2} v_{c,5}(t) d_5 & \text{if } d_{\text{o},2}(t) v_1 \geq \frac{1}{2} v_{c,5}(t) d_5 \text{ and } d_{\text{o},4}(t) v_3 \geq \frac{1}{2} v_{c,5}(t) d_5 \\ \mathcal{O}_2(t) = v_{c,5}(t) - d_{\text{o},4}(t) v_3 ; \mathcal{O}_4(t) = d_{\text{o},4}(t) v_3 & \text{if } d_{\text{o},2}(t) v_1 \geq \frac{1}{2} v_{c,5}(t) d_5 \text{ and } d_{\text{o},4}(t) v_3 < \frac{1}{2} v_{c,5}(t) d_5 \\ \mathcal{O}_2(t) = d_{\text{o},2}(t) v_1 ; \mathcal{O}_4(t) = v_{c,5}(t) d_5 - d_{\text{o},2}(t) v_1 & \text{if } d_{\text{o},2}(t) v_1 < \frac{1}{2} v_{c,5}(t) d_5 \text{ and } d_{\text{o},4}(t) v_3 \geq \frac{1}{2} v_{c,5}(t) d_5 \\ \mathcal{O}_2(t) = \mathcal{O}_4(t) = \frac{1}{2} v_{c,5}(t) d_5 & \text{if } d_{\text{o},2}(t) v_1 \leq \frac{1}{2} v_{c,5}(t) d_5 \text{ and } d_{\text{o},4}(t) v_3 \leq \frac{1}{2} v_{c,5}(t) d_5 \end{cases}$$

. $v_{c,5}(t)$ is computed as

$$v_{c,5}(t) = \begin{cases} v_5 & \text{if } a_6(\text{t}) < l_6 \\ \dfrac{r_7}{d_{o,5}(t)} & \text{if } a_6(\text{t}) = l_6 \end{cases}.$$

Within an overflow full period, besides the previously identified α, u, w points, ξ or η could also be another type of switch points where $\mathcal{O}_2(t)$ change; we call them z points. Follow the same reasoning as used in section 5.2, we can arrive at

$$\frac{\partial L}{\partial l_j} = \sum_{i=1}^{2} \sum_{k=1}^{K_i} \sum_{m=1}^{M_K} F_i(\xi_{i,k,m}) \left(\frac{\partial \eta_{i,k,m}}{\partial l_j} - \frac{\partial \xi_{i,k,m}}{\partial l_j} \right) \text{ for } j = 2,4,6. \qquad (5\text{-}20)$$

$\frac{\partial \eta}{\partial l_j}$ and $\frac{\partial \xi}{\partial l_j}$ are just $\frac{\partial \alpha}{\partial l_j}, \frac{\partial u}{\partial l_j}, \frac{\partial w}{\partial l_j}$ or $\frac{\partial z}{\partial l_j}$ depending on its type. $\frac{\partial \alpha}{\partial l_j}, \frac{\partial u}{\partial l_j}, \frac{\partial w}{\partial l_j}$ can still computed using lemma 5-3, 5-4 and 5-5 results. If it is a z point, it could be a point where $d_{o,4}$ changes, $d_{o,5}$ changes or $a_6(t)$ reaches l_6. We derive $\frac{\partial z}{\partial l_j}$ for each

case. We skip detail case analysis which is similar to the two-segment case analysis. Instead we summarize the results to calculate $\frac{\partial \xi_{1,k,m}}{\partial l_2}$ or $\frac{\partial \eta_{1,k,m}}{\partial l_2}$ in Table 5.3.

Table 5.3 Six-Segment Sample Derivative Analysis Results

	α point	w point	u point	z point		
				$a_6(t) = l_6$	$d_{o,5}$ change	$d_{o,4}$ change
$\frac{\partial \xi_{1,k,m}}{\partial l_2}$	$\frac{d_2 - d_{0,1}(\xi_{1,k,m})}{v_2 * d_{0,1}(\xi_{1,k,m}) - O_2(\xi_{1,k,m})}$	0	$-\frac{d_2}{O_2(\xi_{1,k,m})}$	0	0	0
$\frac{\partial \xi_{1,k,m}}{\partial l_6}$	0	0	0	$\frac{d_6 - d_{0,5}(\xi_{1,k,m})}{v_6 * d_{o,5}(\xi_{1,k,m}) - r_7}$	$-\frac{d_6}{r_7}$	0
$\frac{\partial \xi_{1,k,m}}{\partial l_4}$	0	0	0	0	0	0

Although the algorithm to get the sample derivative becomes more complex, we also save much more simulation run time comparing against finite difference (FD) gradient estimation. We compare the number of simulation runs needed for two methods in the Table 5.4.

Table 5.4 Simulation Run Times Comparison

Network Structure	Number of variables	Single Replication		*N*=5		*N*=10	
		IPA	FD	IPA	FD	IPA	FD
2-segment	1+1	3	4	15	20	30	40
6-segment	3+1	3	8	15	40	30	80
10-segment	5+1	3	12	15	60	30	120

As N increases, the computational requirements of two methods differ tremendously. This is because that two simulation runs are required for each variable in FD estimation method.

5.5 Summary

In this chapter, we utilize fluid simulation as a performance evaluation tool in the A/M network parametric optimization solution procedure. Based on the Batch on Conveyor Model, we derived approximate estimator for sample derivative of loss volume with respect to accumulation length using IPA in a two-segment tandem conveyor system. Numerical experiments show that this approach returns fairly accurate sample derivative estimation. We study convergence results on the same numerical example used in chapter 4. Solution returned from this approach is closer to the optimal solution than solution obtained from chapter 4 method. This is because "Batch on Conveyor Model" represents the conveyor transportation more accurately than "Delay and Stock Model". The idea of extending this approach to more complex network is briefly discussed.

CHAPTER 6

CONCLUSIONS

In this thesis, we developed a set of continuous modeling approach in simulation and optimization for the design and analysis of conveyor sortation system. We make the following contributions.

1. This dissertation advances the understanding of conveyor systems, their simulation and modeling. In the understanding of the system, we derive many dynamic relationships between input and output and other parameters in the system. In the simulation, we identify that the current cell-based simulation has an inherit error source whenever the speed of one of the connected segment is not multiple integer of the other. Such situations are common in merge and in space generating segments. In modeling, we present two continuous conveyor models ("Delay and Stock Model" and "Batch on Conveyor Model") in a unified mathematical framework. We study the accuracy difference of these two representations and give rationales of generating this difference. We also conduct rigorous sensitivity analysis on these two models which lays the foundation to utilize them in optimization applications.

2. Based on the Batch on Conveyor Model, we develop a full functional fluid simulation methodology applying to high volume complex conveyor network simulation in chapter 2. We address the feasibility of implementing fluid simulation from modeling capabilities, simulation algorithm, performance measure collection and simulation performance in terms of accuracy and simulation time. We also identify major factors that contribute to the computational effort of fluid simulator by conducting analysis on number of generated batches. Experimental results show that fluid simulator can achieve the similar accuracy as cell based simulation in much shorter simulation time. The computational savings is especially significant when simulating a network that comprises of many accumulation conveyor segments. This research demonstrates that fluid

simulation is a promising fast simulation methodology applying to the high volume transportation conveyor network simulation.

3. In chapters 3, 4 and 5, we model the A/M network parametric optimization problem under stochastic condition and propose a simulation optimization solution framework to solve the problem. It provides a systematic way for rates determination and accumulation design of accumulation and merge subsystem where simulation is the only method in the current design toolkit of such system. From solution methodology perspective, to the best of our knowledge, this is the first attempt to integrate the continuous modeling, sensitivity analysis and simulation optimization methodology into a solution procedure in conveyor network design research. Promising results are shown through theoretical analyses and some numerical experiments.

Future research can be pursued in the following directions.

1. Statistical traffic model analysis. This includes how to conduct traffic measurement to get characteristic trace data, how to construct stochastic fluid input model to capture the important statistical properties of measured trace data and developing efficient parameter estimation procedure for the stochastic fluid process. Actually this research relies on a good fluid traffic model that can capture important characteristics of real input source.

2. Further improvement of fluid simulation speed. Fluid simulation is faster than cell-based simulation because it has fewer discrete events, but each event in the fluid simulation can take much more computation time than a single event in cell-based simulation. Most computational requirement lies in computation of instantaneous firing flows (IFF) at each step. In the current computation procedure, we examine all batch transitions once the IFF calculation is required. An intuitive approach to reduce the computational effort is to only examine a subset of batch transitions that are possibly affected by the current trigger event while keeping the IFFs of other transitions

unchanged. Research needs to be conducted to see whether there exists an efficient method to identify this subset of transitions for a particular kind of trigger event.

3. Extension of IPA sample gradient estimation method to complex network structures. In the end of chapter 5 we only outline the idea of extending the IPA gradient estimate method to the more complex conveyor network through an example. Real implementation is needed to obtain such estimator. More importantly, accuracy comparison needs to be done to see whether the IPA accuracy will deteriorate as the network complexity increasing.

APPENDIX A

COMMON MERGE AND DIVERGE RELEASE LOGICS

We describe commonly used merge release logics and its implementation in S-BPN model in this section.

Priority Merge (PM):

Priority Merge refers to the following release control logic: the items from the lower priority class input conveyor can only merge when they do not impede the passing of items from the high priority class conveyors. An example application is the merge from a branch to a main conveyor. The priority merge is characterized by a set of integers to indicate the priority number of each input lines. The lowest number designates the highest priority. The PM capacity allocation algorithm in IFF calculation works as follows. When the sum of the input rates is less than the merge capacity, flow from each input line will pass the merge point without any delay. Otherwise the merge capacity is allocated equally among input lines in the highest priority class; any left-over capacity is allocated equally among lines in the next-highest priority class, and so on, until capacity runs out or all lines are satisfied. In the priority merge case, the $\text{Post}(p_i, t_j)$ with $t_j \in {}^{\circ}p_i \cap T_b$ indicates the priority number of each input batch transition. Lowest numbers have highest priority.

Processor Share Merge (SM):

Processor Share Merge can be used to model the merge conveyor when each induction conveyor merges on an alternating or round robin basis, as long as there are objects available; an induction conveyor with no package just simply skipped. Suppose a merge device is shared by *J* input sources and operates at a fixed rate r, the information needed to implement the SM merge policy is contained in the weight vector $\alpha = (\alpha_1, ..., \alpha_J)$ with $\alpha_j > 0 \forall j$ and $\sum_{j=1}^{J} \alpha_j = 1$. We assign the $\text{Post}(p_i, t_j)$ as the weight

α_j. The SM capacity allocation algorithm in IFF calculation works as follows. When all sources have accumulation, source *j* is allotted a fraction α_j of the merge capacity. When some sources achieve no accumulation using less than their allotted capacity, the remaining merge capacity is split among the other sources in proportion to their α_j's.

Time-Sliced Merge

Time Sliced Merge models the situation where each input line is released for a pre-defined time period before switching to another line as long as it has positive flow. An induction conveyor with no package is just simply skipped. A high-speed saw-tooth merge often adopts this logic to increase the merge speed. Since at any time point, only one pre-transition of the merge place is enabled, it is not treated as a merging unit in S-BPN. Instead, the control logic is enforced through discrete part of the net.

Similar diverge release logics are defined analogous to the above merge logics:

Priority Diverge (PD)

Priority Diverge refers to the following release control logic: the upstream items will be released as much as possible to the downstream channels with higher priority provided that the capacity is allowed. The priority merge is characterized by a set of integers to indicate the priority number of each input lines. The lowest number designates the highest priority. Priority number may be set according to distance or transportation cost. In the priority diverge case, the $\Pr e(p_i, t_j)$ with $t_j \in p_i^{\circ} \cap T_b$ indicates the priority number of each output batch transition.

Blocking Diverge (BD) & Reroute Diverge (RD)

They are diverge logics analogous to processor share merge. Both logics imply that, when unblocked, the out flow of the in-feed conveyor will be routed to downstream conveyors according to pre-defined proportions. Blocking diverge means all transferring will be blocked if there is at least one downstream with insufficient receiving capacity while reroute diverge indicates that in the presence of insufficient capacity downstream

conveyor, the leftovers is split among the other downstream channels in proportion to their predefined routing proportions. In both cases, the $\Pr e(p_i, t_j)$ with $t_j \in p_i^{\circ} \cap T_b$ represents the routing percentage indicating $\Pr e(p_i, t_j)$ percent of total outflow of conveyance system p_i will be conveyed to a downstream conveyor through batch transition t_j provided that the capacity allowed.

Time-Sliced Diverge (TD)

In the Time-Sliced diverge release, the in-feed items will be released to an output line for a pre-defined time-period before switching to another line as long as it has positive receiving capacity. A current blocked downstream conveyor is just simply skipped. Since at any time point, only one post-transition of the merge place is enabled, it is not considered as a diverging unit in IFF calculation. The control logic is enforced through discrete part of the net.

APPENDIX B

S-BPN SPECIFICATION

Structural Condition Relaxation of S-BPN

BPN requires that if an arc joins a discrete place at a batch transition, there must exist a reciprocal arc linking this continuous or batch transition at this discrete place. This is relaxed in S-BPN. Figure B-1 illustrates this simplification.

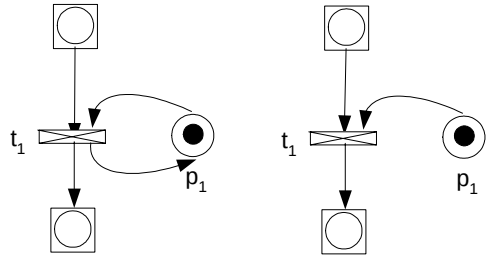


Figure B.1 Structural simplification of S-BPN

In Fig B-1, the original condition means $p_1 \in {}^0 t_1$, $p_1 \in t_1{}^0$ and $\Pr e(p_1, t_1) = \text{Post}(p_1, t_1)$. In BPN, batch transition uses the following firing rule to ensure the correct markings of its pre and post discrete places:

1) The number of tokens equal to the firing flow multiplied by the weight of arc joining the discrete places to the batch transition is removed from pre-discrete places of the batch transition.

2) The number of tokens equal to the firing flow multiplied by the weight of arc joining the batch transition to the discrete places is added to post-discrete places of the batch transition.

In the example in Fig B.1, if t_1 is fired at time t during a delay dt, the marking of p_1 will change as follows:

$$M_1(t+dt) = M_1(t) - \phi_1(t) * \Pr e(p_1, t_1) * dt \text{ since } p_1 \in {}^0 t_1$$

$$M_1(t+dt) = M_1(t) + \phi_1(t) * \text{Post}(p_1, t_1) * dt \text{ since } p_1 \in t_1{}^0$$

By this firing rule, we can see M_1 is unchanged during dt which is equivalent to say that the firing of a batch transition will not affect the markings of discrete places. In S-BPN we just achieve the same effect by changing the firing rule instead of enforcing this structural constraint. The above reciprocal arcs are not required. We change the firing rule of batch transitions as follows: The firing of a batch transition does not change the number of markings in its pre and post discrete places.

By this condition, any arc that links the batch transition to a discrete place has no effect at all. The arc linking the discrete place to batch transition only affects its enabled condition.

We can eliminate many unnecessary arcs by using this approach, but we lose uniformity of firing conditions since we need to distinguish among pre-discrete places, pre-batch places and pre-continuous places. In general Hybrid Petri Net, this uniformity is important since it can save firing processing time by treating the discrete places and continuous places in the same way. In BPN, it is less important since the firing effect to batch place is always different from other places. So we take this approach to keep the model neat.

S-BPN Enabling and Firing Conditions

Table B.1 S-BPN Enabling Conditions

	$t_j \in T_d$		$t_j \in T_b$	
	Pre Condition	Post Condition	Pre Condition	Post Condition
$p_i \in P_d$	Regular arc: $m_i(t) \geq \text{Pre}(p_i, t_j)$ Inhibit arc: $m_i(t) < \text{Pre}(p_i, t_j)$		$m_i(t) \geq \text{Pre}(p_i, t_j)$	N/A
$p_i \in P_c$	Regular arc: $m_i(t) \geq \text{Pre}(p_i, t_j)$ Inhibit arc: $m_i(t) < \text{Pre}(p_i, t_j)$	$m_i(\text{t}) + \text{Post}(p_i, t_j) \leq b_i$	$m_i(t) > 0$ or $m_i(t) = 0$ and *fed**	$m_i(t) < b_i$ or $m_i(t) = b_i$ but *drain**
$p_i \in P_b$	Regular arc: $V(Cap_j(i)) = 1$ inhibit arc : $V(Cap_j(i)) = 0$	N/A	$B_{o,i}(t) \neq$ NULL	

Table B.2 S-BPN Firing Rules

	$t_j \in T_d$		$t_j \in T_b$	
	Pre Places	Post Places	Pre Places	Post Places
$p_i \in P_d$	Regular arc: $m_i(t) = m_i(t) - \text{Pre}(p_i, t_j)$	$m_i(t) = m_i(t) + \text{Post}(p_i, t_j)$		N/A
$p_i \in P_c$	Regular arc: $m_i(t) = m_i(t) - \text{Pre}(p_i, t_j)$	$m_i(t) = m_i(t) + \text{Post}(p_i, t_j)$	$m_i(t + \Delta t) = m_i(t) - \Delta t \phi_j(\text{t}) \text{Pre}(p_i, t_j)$	$m_i(t + \Delta t) = m_i(t) + \Delta t \phi_j(\text{t}) \text{Post}(p_i, t_j)$
$p_i \in P_b$		N/A	Batch Condition Updating	Batch Condition Updating

*: Fed: We call a batch place or a continuous place *fed* if there is at least one pre-batch transition with positive instantaneous firing rate. The sum of the IFFs of all pre-batch transitions are called *fedrate*.

Drain: we call a batch place or a continuous place *drain* if there is at least one post-batch transition with positive instantaneous firing rate. The sum of the IFFs of all post-batch transitions are called *drainrate*.

S-BPN IFF Calculation

In general, IFF calculation is an iterative procedure since several conditions require the total upstream IFF of a place should be equal to the total downstream IFF. This can cause inter-dependence. These conditions include an empty or full continuous place, a full accumulation batch place packed in maximum density and a non-accumulation batch place having an output batch with maximum density. At each iteration, the IFF of a batch transition is the minimum of three values: its MFF value, the minimum speed constraints posted by pre-places and the minimum of the speed constraints posted by its post-places. Merging and diverging unit add extra complexity in IFF calculation. Since specific merge and diverge release control logic is reflected as different way to allocate the merge/diverge capacity, the IFFs of transitions belonging to the same merging or diverging unit should be calculated simultaneously with specific algorithm determined by logic used. This was implemented as 2-pass algorithm. In the first pass, we calculate a tentative IFF for each transition in the merging unit individually, ignoring the merge capacity constraint. We call them sub-IFFs. In the second pass, we check whether these sub-IFFs satisfied the merging capacity constraint. If the constraint is satisfied, sub-IFFs become our final IFFs. If not, the merging capacity will be allocated according to merge logics. Diverging unit is treated similarly. In the following, we first introduce the notation and then give the pseudo code for IFF calculation.

Notation:

TE : set of enabled transition

MS: set of merging units

DS: set of diverging units

NS: set of normal transitions

K: iteration index

UT_i: upstream theoretical limit imposed by a place p_i to its post-transitions

DT_i: downstream theoretical limit imposed by a place p_i to its pre-transitions

ϕ_{uthj}: upstream theoretical flow of batch transition t_j

ϕ_{dthj}: downstream theoretical flow of batch transition t_j.

ϕ: IFF vector of at iteration k with component ϕ_j denoting the IFF of batch transition t_j.

ϕ_j': sub-IFF of transition t_j in a merging unit or diverging unit.

$d_{o,i}$: density of output batch in place p_i.

A merging unit: {

Merge batch place p_m where $p_m \in P_{MB} \cap P_b$

Transitions $\{t_1, t_2, ..., t_n\}$ with $t_j \in {}^{\circ}p_m \cap T_b \ \forall j = 1...n$ and t_j enabled

}

A diverging unit: {

Diverge batch place p_d where $p_d \in P_{DB} \cap P_b$

Transitions $\{t_1, t_2, ..., t_n\}$ with $t_j \in p_d{}^0 \cap T_b \ \forall j = 1, ..., n$

}

Algorithm 1: Calculating IFFs

Step1: Identifying merging and diverging units, so that TE can be partitioned into TE=NS∪MS∪DS

Step2: k=0, ϕ^0=0

Step3: k=k+1

Step4: Using algorithm 2,3,4 to calculate ϕ_j^k 's according to t_j belonging to NS,MS or DS.

Step5: if $\phi^k = \phi^{k+1}$. go to step 6;else go to step 3.

Step6: $\phi_j(t) = \phi_j^k$

Algorithm 2: Normal batch transition IFF calculation

$$\phi_j{}^k = \min\left(\phi_{uthj}{}^k, \Phi_j(t), \phi_{dthj}{}^k\right)$$

$$\phi_{uthj}{}^k = \min_{\{p_i \in {}^0 t_j\}}\left(UT_i{}^k\right)$$

$\phi_{dthj}{}^{k} = \min_{\{p_i \in t_j{}^{0}\}} \left(DT_i{}^{k} \right)$

The formulations to determine *UT* and *DT* are summarized in table B.3.

Table B.3 Formulations to Determine *UT* and *DT*

		$t_j \in T_b$	
		UT_i	DT_i
$p_i \in P_d$		∞	∞
$p_i \in P_c$		∞ if $m_i(t) > 0$; fedrate if $m_i(t) = 0$	∞ if $m_i(t) < b_i$; drainrate if $m_i(t) = b_i$
$p_i \in P_b$	'A'	$v_i d_{o,i}$	$v_i d_i$ if not full; drainrate if full;
	'N'		$v_i d_i$ if $B_{o,i}(t) = \text{NULL}$; min($v_i d_i$, (drainrate/ $d_{o,i}$) d_i), if $B_{o,i}(t) \neq \text{NULL}$

Algorithm 3: Merging unit IFF calculation

Step0: Normalize the arc weights. $\text{Post}'(p_m, t_j) = \dfrac{\text{Post}(p_m, t_j)}{\sum_{i=1,..,n} \text{Post}(p_m, t_i)}$

Step1: Using algorithm 2 to calculate SubIFFs $\phi_j{}', j = 1...n$ with $t_j{}^{o} = t_j{}^{o} \setminus p_m$

Step2: Set merge capacity $F = DT_m$.

Step3: If $\sum_{j \in \{1,..,n\}} \phi_j{}' \leq F$ then $\phi_j = \phi_j{}'$ for all j;

Else calculate ϕ_j 's using algorithm 5 or 6 according to $mc(p_m)$.

Algorithm 4: Diverging unit IFF calculation

Step0: Normalize the arc weights. $\text{Pre}'(p_d, t_j) = \dfrac{\text{Pre}(p_d, t_j)}{\sum_{i=1,..,n} \text{Pre}(p_d, t_i)}$

Step1: Using algorithm 2 to calculate SubIFFs $\phi_j{}', j = 1...n$ with $t_j{}^{o} = t_j{}^{o} \setminus p_m$

Step2: Set Diverging capacity $F = UT_d$.

Step3: Calculate ϕ_j 's using algorithm 7 or 8 according to $dc(p_d)$.

Algorithm 5: IFF allocation $mc(p_m) = \text{PM}$

Let $h_{\max} = \max\{\text{Post}(p_m, t_1), ..., \text{Post}(p_m, t_n)\}$

Let $C_p = \{j : \text{Post}(p_m, t_j) = p\}, p = 1...h_{\max}$

$F = DT_m$; $h = 1$; $\phi_j = 0 \forall j = 1...n$

Do while ($F > 0$ and $h < h_{\max}$)

{

$C = C_h; N = \text{Cardinality}(C);$

For each $j \in C$

$\phi_j = \min(\phi_j', F/N);$

$F = F - \sum_{j \in C} \phi_j;$

$h = h + 1;$

}

Algorithm 6: IFF allocation $mc(p_m) = \text{SM}$

Let $C = \{1 \ldots n\}$; $F = DT_m$; $\phi_j = 0 \forall j = 1 \ldots n$

Do while (True)

{

$S = \left\{ j \mid j \in C; \phi_j' < \frac{\text{Post}'(p_m, t_j)}{\sum_{h \in C} \text{Post}'(p_m, t_h)} \cdot F \right\}$ if ($S \neq \varnothing$){

$\phi_i = \phi_i' \quad \forall i \in S$

$F = F - \sum_{j \in S} \phi_j$

$C = C \setminus S;$

exit;

}else{

$\phi_j = \frac{\text{Post}'(p_m, t_j)}{\sum_{h \in C} \text{Post}'(p_m, t_h)} \cdot F \quad \forall j \in C$ exit;

}

}

Algorithm 7: IFF allocation $dc(p_d) = \text{BD}$

Set $\phi_j = 0 \forall j = 1 \ldots n$; flag=1

For j=1 to n{

If $\phi_j' < UT_d * \text{Pre}(p_d, t_j)$ {

flag = 0;

break;

}

}

If (flag = =1){

$\phi_j = UT_d * \text{Pre}(p_d, t_j) \quad \forall j = 1,..n$

}else{ $\phi_j = 0 \; \forall j = 1 \ldots n$ }

Algorithm 8: IFF allocation $dc(p_d) = \text{RD}$

Set $\phi_j = 0 \forall j = 1..n$

$L = 0$;
For j =1 to n {
    If $(\phi_j^{'} < UT_d * \mathrm{Pre}^{'}(p_d, t_j) + L)$ {
        $\phi_j = \phi_j^{'}$
        $L = UT_d * \mathrm{Pre}^{'}(p_d, t_j) + L - \phi_j^{'}$
    }else{
        $\phi_j = UT_d * \mathrm{Pre}^{'}(p_d, t_j) + L$
        L=0
    }
}

APPENDIX C

NETWORK CONFIGURATION USED IN THE COMPUTATIONAL COMPARISON EXPERIMENTS

2-Segment Network

Refer to Figure 4.1 for network layout and its S-BPN graph.

15-Segment Network

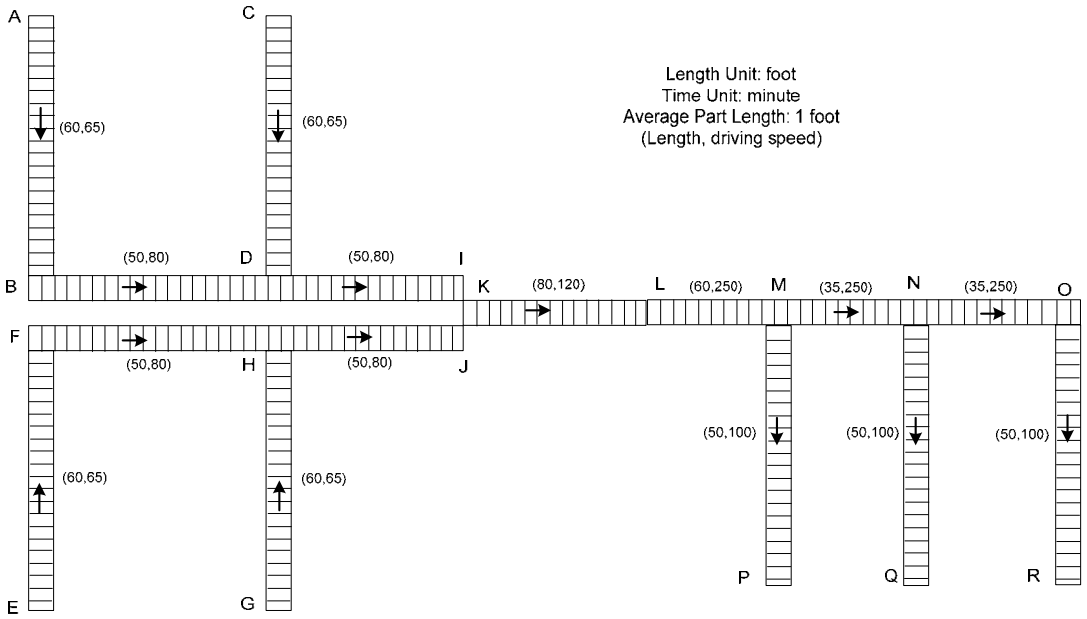


Figure C.1 15-Segment Conveyor Transportation Network

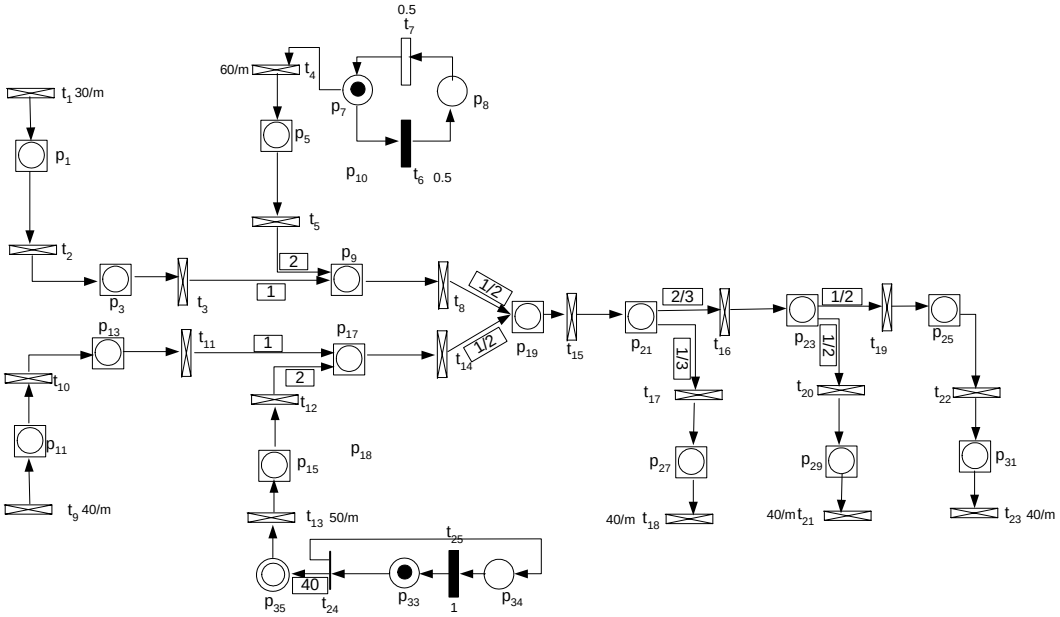


Figure C.2 S-BPN Model of 15-Segment Conveyor Network

76-Segment Network

Refer to Figure 2.8 and Figure 2.9 for network layout and its S-BPN graph.

APPENDIX D

SAMPLE DERIVATIVE ESTIMATION DATA

Table D.1 Sample Derivative Estimation Data

Case	l2	r3	Average Loss		Replication1	Replication 2	Replication 3	Average
1	11.1	24.6	716	IPA	12	15	13	13.33
				FD	16.60	13.00	14.00	14.53
				DIFF	-4.60	2.00	-1.00	-1.20
				DIFF%	-27.7%	15.4%	-7.1%	-8.3%
2	11.8	28.3	393	IPA	13.7	2.4	9.1	8.40
				FD	15.0	2.5	11.1	9.53
				DIFF	-1.3	-0.1	-2.0	-1.13
				DIFF%	-8.7%	-4.0%	-18.0%	-11.9%
3	12.6	31.9	253	IPA	6.9	3.8	8.9	6.53
				FD	6.6	4.1	6.3	5.67
				DIFF	0.3	-0.3	2.6	0.87
				DIFF%	4.5%	-7.3%	41.3%	15.3%
4	14.1	36.8	121	IPA	3.2	3.1	4.3	3.53
				FD	5.6	3.8	3.8	4.39
				DIFF	-2.4	-0.7	0.5	-0.86
				DIFF%	-42.9%	-18.4%	13.8%	-19.6%
5	106	23.6	146	IPA	4	1.4	3.7	3.03
				FD	2.2	2.0	0.2	1.45
				DIFF	1.8	-0.6	3.5	1.58
				DIFF%	85.3%	-30.0%	1750.0%	108.8%
6	85.5	27.4	87	IPA	1.8	2.2	1.8	1.93
				FD	1.3	2.7	2.1	2.02
				DIFF	0.6	-0.5	-0.3	-0.08
				DIFF%	44.0%	-18.5%	-14.3%	-4.1%
7	105	20.1	345	IPA	1.8	2.5	6.6	3.63
				FD	1.9	3.9	3.8	3.20
				DIFF	-0.1	-1.4	2.8	0.43
				DIFF%	-5.3%	-35.9%	73.7%	13.5%
8	162	22.9	65	IPA	1.3	0.2	2.5	1.33
				FD	0.2	0.1	1.8	0.70
				DIFF	1.1	0.1	0.7	0.63
				DIFF%	550.0%	100.0%	38.9%	90.5%
9	122	23.8		IPA	6.8	2.1	3.3	4.07
				FD	4.5	1.3	0.1	1.97
				DIFF	2.3	0.8	3.2	2.10
				DIFF%	51.1%	61.5%	3200.0%	106.8%
10	125	24		IPA	6	2.3	2	3.43
				FD	4.5	1.3	0.3	2.03
				DIFF	1.5	1.0	1.7	1.40
				DIFF%	33.3%	76.9%	566.7%	68.9%
11	50	29		IPA	4.9	4.00	4.80	4.57

				FD	6.0	3.87	6.00	5.29
				DIFF	-1.1	0.1	-1.2	-0.72
				DIFF%	-18.3%	3.4%	-20.0%	-13.7%
12	80	30		IPA	1.50	0.50	2.40	1.47
				FD	1.75	0.10	2.25	1.37
				DIFF	-0.3	0.4	0.2	0.10
				DIFF%	-14.3%	400.0%	6.7%	7.3%

REFERENCES

Apple, J. M., "Batch picking and sortation, the design of high-volume case picking operations", The Progress Group white paper, http://www.theprogressgroup.com/publications/wp5_pick.html (Accessed April 7, 2006)

Arandtes, J.C. and Deng, S., "Modeling and solution methods for the design and control of conveyor systems with merge configuration", *Progress in Material Handling Research*, pp. 35-50, Material Handling Institute, Charlotte, North Carolina, 1996.

Atmaca, T., "Approximation analysis of a conveyor system", *International Journal of Production Research*, vol. 32, pp. 2645-2655, 1994.

Audry, N. and Prunet, F., "Controlled Batches Petri Nets", *IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp.1849-1854, 1995.

Balduzzi, F. and Giua, A., "Modeling and simulation of manufacturing systems with first-order hybrid Petri Nets", *International Journal of Production Research*, vol. 39(2), pp. 255-282, 2001.

Bastani, A., "Analytical solution of a closed-loop conveyor system with discrete and deterministic material flow", *European Journal of Operation Research*, vol. 35, pp. 187-192, 1988.

Bastani, A., "Closed-loop conveyor systems with breakdown and repair of unloading stations", *IIE Transactions*, vol. 22 (4), pp. 351-360, 1990.

Bastani, A.S., "Blocking in closed-loop conveyor systems connected in series with discrete and deterministic material flow", *Computers & Industrial Engineering*, vol. 11, n. 1-4, pp. 40-45, 1986.

Bozer, Y. A., and Hsieh, Y. J., "Expected waiting times at loading stations in discrete-space closed-loop conveyors", *European Journal of Operational Research*, vol. 155(2), pp. 516-532, 2004.

Bozer, Y.A., and Sharp, G. P., "An empirical evaluation of a general purpose automated order accumulation and sortation system used in batch picking", *Material Flow*, vol. 2, pp. 111-131, 1985.

Bozer, Y.A., and Quiroz, M. and Sharp, G.P., "An evaluation of alternate control strategies and design issues for automated order accumulation and sortation system", *Material Flow*, vol. 4, pp. 265-282, 1988.

Caradec, M.; Prunet, F, "A new modeling tool for hybrid flexible systems", *IEEE International Conference on Systems, Man, and Cybernetics*, vol. 3, pp12-15, 1997.

Chang, J. X., "Dynamic scheduling of open multiclass queueing networks in a slowly changing environment", Doctoral dissertation, Department of Industrial Engineering, Georgia Institute of Technology, 2004.

Choe, K. IL., "Aisle-based order pick system with batching zoning and sorting", Ph.D. dissertation, School of Industrial and System Engineering, Georgia Institute of Technology, 1990.

Choudhury, G. L. and Mandelbaum, A. and Reiman, M. I., and Whitt, W., "Fluid and diffusion limits for queues in slowly changing environments", *Stochastic Models*, vol. 13, pp. 121–146, 1997.

Coffman, E.G. and Gelenbe, E., "Analysis of a conveyor queue in a flexible manufacturing system", *European Journal of Operational Research*, vol. 35, pp. 382-392, 1988.

Demongodin, I. And Prunet, F., Simulation modeling for accumulation conveyors in transient behavior. *COMPEURO 93*, Paris. France, pp. 29-37, 1993.

Demongodin, I. and Prunet, F., "Batches Petri Nets", *IEEE international Conference on Systems, Man and Cybernetics*, Le Touquet, France, pp. 607-617, October 1993

Demongodin, I., "Extended structures of Batches Petri Nets", *IEEE International Conference on Systems, Man and Cybernetics*, pp. 182-187, 1999.

Disney, R., "Some results of multichannel queueing problems with ordered entry- An application to conveyor theory", *The Journal of Industrial Engineering*, 105-108, 1963.

Fleischer, L. and Tardos, E., "Efficient continuous-time dynamic network flow algorithms", *Operations research letters*, 23, 71-80, 1998.

Fleischer, L., "Universally maximum flow with piecewise-constant capacities", *Networks*, vol. 38(3), 115-125, 2001.

Ford, L.R. and Fulkerson, D.R, Flows in Networks, Princeton University Press, Princeton, NJ, 1962.

Geinzer, C. M. and Meszaros, J. P., "Modeling high volume conveyor sorting system", *Proceedings of the Winter Simulation Conference*, 1990.

Gewekw, J., "Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments", *Bayesian Statistics* 4, pp. 169-193, Oxford University Press, 1992.

Glasserman, P., Perturbation Analysis for Gradient Estimation, Kulwer, 1991.

Gurkan, G., and Ozge, A.Y. and Robinson, S.M., "Sample-path optimization in simulation" *Proceeding of the Winter Simulation Conference*, 247-254, 1994.

Ho, Y. C. and Cao, X. R., Perturbation Analysis of Discrete Event Dynamic Systems, Kluwer, 1991.

Hoppe, B. and Tardos, E., "The quickest transshipment problem", *Mathematics of Operation Research*, vol 25(1), 2000.

Jing, G. and Kelton, W. D. and Arantes, J.C., "Modeling a controlled conveyor network with merging configuration", *Proceeding of the Winter Simulation Conference*, pp. 1041-1048, 1998.

Johnson, M. E. and Lofgren, T., "Model Decomposition speeds distribution center design", *Interfaces*, vol. 24(5), pp. 95-106, 1994.

Johnson, M. E., "The impact of sorting strategies on automated sortation system performance", *IIE Transactions*, vol. 30(1), pp. 67-77, 1998.

Johnson, M. E. and Russell, M., "Performance analysis of split-case sorting systems", *Manufacturing & Service Operations Management*, vol. 4(4), pp 258-274, 2002.

Kesidis, G. and Singh, A., "Feasibility of fluid-driven simulation for ATM network" *IEEE GLOBECOM*, pp. 2013-2017, 1996.

Kwo, T.T., "A theory of conveyors", *Management science*, vol. 6, pp. 51-55, 1958.

Lin, T., "Operation analysis and design of large complex conveyor networks", Technical Report, Material Handling Research Center, Georgia Institute of Technology, 1994

Liu, B. and Guo, Y., "Fluid simulation of large scale networks: issues and tradeoff", *PDPTA 99*, pp. 2136-2142, 1999.

Liu, B. and Figueiredo, D., "A study of networks simulation efficiency: Fluid simulation vs. packet-level simulation", *INFOCOM'01*, 2001.

Maxwell, W. and Wilson, R., "Dynamic network flow modeling of fixed path material handling systems", *IIE Transactions*, vol. 13(1), pp. 12-21, 1981.

Maxwell, W. and Wilson, R., "Deterministic models of accumulation conveyor dynamics," *International Journal of Production Research*, vol. 19(6), pp. 645-655, 1981.

Mayer, H., An introduction to conveyor theory, Western Electric Engineer, 1960

Morris, W. T., Analysis for material handling management, Richard D. Irwin. Inc, 1962.

Muth, E. J., "Analysis of closed-loop conveyor systems", *AIIE Transactions*, vol. 4(2), pp. 134-143, 1972.

Ogier, R.G., "Minimum-delay routing in continuous-time dynamic networks with piece-wise constant capacities", *Networks*, 18, 303-318, 1988.

Olafsson, S. and Kim, J. "Simultion Optimization", *Proceedings of the Winter Simulation Conference*, 79-84, 2002.

Plambeck, E.L. and Fu, S.R. and Robinson, S.M., "Sample-path optimization of convex stochastic performance functions", *Mathematical Programming*, vol. 75, pp. 137-176, 1996.

Russell, M. L. and Meller, R. D., "Cost and throughput modeling of manual and automated order fulfillment systems", *IIE Transactions*, 35, 589-603, 2003.

Schmidt, L. C. and Jackman, J., "Modeling recirculating conveyors with blocking", *European Journal of Operational Research*, vol. 124, pp. 422-436, 2000.

Shapiro, A., "Stochastic programming by Monte Carlo simulation methods", *Stochastic Programming*, vol. 10 of *Handbooks in Operations Research and Management Science*. Elsevier North Holland, 2003.

Sonderman, D., "An analytic model for recirculating conveyors with stochastic inputs and outputs", *Internal Journal of Production Research*, vol. 20(5), pp.591-605, 1982

Stubbs, K., "Light & medium load conveyor transportation systems", *Automated Material Handling & Storage Systems Conference*, 24, 1980.

Veldsma, T., "System design and installation considerations for conveyor sortation systems", Perspective on material handling practice, http://www.mhiastore.org/moreinfo.cfm?Product_ID=242 (Accessed April 7, 2006).

Wagner, G., "Basic conveyor system considerations", Perspective on material handling practice, http://www.mhiastore.org/moreinfo.cfm?Product_ID=309(Accessed April 7, 2006).

Wang, S. J., "An integrated construction/simulated annealing approach for designing sortation conveyor system in automated distribution center", Ph.D Dissertation. Department of Industrial Engineering, University of Houston, 1997.

Wang, Y. and Zhou, C., "Fluid-based simulation approach for high volume conveyor transportation Systems", *Journal of System Science and System Engineering*, vol. 13(2), 2004

Wang, Y. and Zhou, C., "Fluid-based simulation model for high volume DC conveyor systems", *Proceedings of winter simulation conference,* 2005

Xue, J. and Proth, "Closed loop conveyor system", *INFOR*, vol. 25, pp. 84-92, 1987.

Zrnic, DJ. N., and Cupric, N. LJ., "A study of material flow systems (input/output) in high-bay warehouses", *International Journal of Production Research*, vol. 30(9), pp. 2137-2149, 1992.