החוג לסטטיסטיקה
הפקולטה למדעי החברה
אוניברסיטת חיפה

הצעת מחקר לעבודת דוקטור:

Stability, Utilization, Fairness and Throughput of Stochastic Processing Networks with Infinite Inputs

יציבות, ניצולת, הוגנות ותפוקה ברשתות עיבוד סטוכסטיות בעלות כניסות אינסופיות

מאת: יוני נצרתי
מנחה: פרופ' גדעון וייס

תאריך הגשה: 1/9/2006
חתימת המנחה: _____

Contents

Ph.d Research Proposal:
Stability, Utilization, Fairness and Throughput
of Stochastic Processing Networks
with Infinite Inputs

Applicant: Yoni Nazarathy
Supervisor: Professor Gideon Weiss

Extended Abstract

This is a Ph.d research proposal. It summarizes the proposed research questions, states the work plan and surveys relevant background. The proposed research questions are labeled for convenience *PRQ0*, *PRQ1*, etc. At certain times we extend our notation and use digits to the right of the decimal point for introducing a hierarchy of the proposed research questions. For example: *PRQ2.3* and *PRQ2.5* are all part of the more general proposed research question *PRQ2*.

The research deals with Stochastic Processing Networks with Infinite Inputs (SPNII). This is a stochastic queueing model that may find a variety of applications in manufacturing plants, complex communication networks and road traffic networks among others. These network models are generalization of Multi Class Queueing Networks (MCQN) and Stochastic Processing Networks that were proposed and studied by Harrison, Dai and many others. The generalization is in the sense of adding the possibility of infinite inputs as opposed to the standard arrival process inputs. Simple yet interesting instances have been studied by Weiss and others.

We first introduce the ultimate goal of our research: given a specific network, does there exist a scheduling policy that maintains stochastic stability, utilizes resources to the fullest and produces outputs from the network at suitable proportions (fairness among output streams). This is the question of existence of a stable, fully utilizing, and fair scheduling policy (*PRQ0.0*).

It is natural to extend this question to the optimization problem of finding the optimal policy in terms of throughput maximization (*PRQ0.1*). Using our notation for proposed research questions, we refer to the combination of both of these proposed research questions as *PRQ0*.

We believe that *PRQ0* is an exceedingly ambitious question to answer at this phase. We thus propose a series of more specific research questions whose study may shed light on the dynamics and mechanics of SPNII models and eventually lead to new results regarding *PRQ0*.

We introduce a set of questions that deal with the re-entrant line model exhibiting infinite inputs (RLINEII). While this model does not capture all of the aspects of a SPNII, its study may still yield the needed experience and insight. We start with *PRQ1* which deals with stability of the Last Buffer First Server (LBFS) policy.

The following sub-questions deal with the case in which the bottleneck resource is used for the first operation: *PRQ1.0* deals with the simpler Markovian case and attempts to show positive recurrence using methods of Lyapunov Functions. *PRQ1.1* extends its former and attempts to find a steady state distribution. *PRQ1.2* extends the Markovian case to the general case and attempts to show stability using fluid theorems. *PRQ1.3* proposes to use Lyapunov Functions on general state spaces to show stability. We then introduce *PRQ1.4* which deals with the case in which the bottleneck resource is not used for the first operation.

Following the specific LBFS policy, we propose to continue with the RLINEII model and attempt to find optimal policies (*PRQ2*). In *PRQ2.0* we propose a dynamic programming formulation of the problem under the Markovian case. In *PRQ2.1* we propose simulation studies regarding optimization and heuristic optimization of RLINEII with respect to several performance measures.

Handling of most of the proposed research questions regarding RLINEII may require investigating a simpler and more trackable model. We thus

handle the simplest RLINEII model possible that still exhibits some sort of re-entrance; this is the 2 resource, 3 buffer infinite input model (2R3BII). This model has been studied extensively by Weiss et.al. We propose several important extensions that have not yet been dealt with.

Most of the analysis of 2R3BII has been performed with respect to the LBFS policy and the Markovian case. In *PRQ3* we propose to extend the research of this tractable case and study additional attributes such as busy period distributions and more. In *PRQ4* we propose to study the LBFS policy on this model when general service times are applied. The plan here is to use Lyapunov Function methods in general state spaces. In *PRQ5* we propose a series of optimization problems regarding this model in the Markovian case. Most of the research questions regarding 2R3BII are separated into sub-cases depending on which of the two resources is the bottleneck.

Another specific case of the RLINEII model is the Flow Shop. This model does not exhibit re-entrancy and thus the interesting features of the infinite input buffers are mostly neutralized. Yet, we propose a series of research questions regarding holding cost optimizations of the fluid analog of this model over a finite time horizon. These questions extend previous work by Weiss and involve Separated Continues Linear Programming (SCLP). We label these as *PRQ6*.

Moving away from the RLINEII model, we meet the 2 resource, 4 buffer infinite input model (2R4BII). This model has been studied by Weiss and Kopzon under the name of the Push-Pull Model. Weiss and Kopzon have found certain policies that achieve full utilization while maintaining stability, these are called generalized threshold policies (GTP). These stability results are currently only available for the Markovian case. We introduce preliminary simulation results of Maximum Pressure Policies (analyzed by Dai and Lin) of these models and show that these scheduling policies are non-stable (*PRQ7*). Comparing these results to the generalized threshold policies we see the strength of the generalized threshold policies. Further more in *PRQ7*,

we propose to obtain simulation results that indicate that stability may exist under the generalized threshold policies under general service time distributions. *PRQ8* deals with proving stability of the generalized threshold policy in the general distribution case. We propose to tackle this proof by means of Lyapunov Functions on general state spaces.

PRQ9 deals with providing additional models that may bridge the gap towards *PRQ0*. It is a big gap and at this point we are not clear with regards to which additional models to investigate and which additional questions to propose on route to tackling *PRQ0*. We believe that after some hard work on most of the questions mentioned above, *PRQ9* may be properly handled and thus additional relevant models may be supplied. At that point of time, further research may be performed on these models towards a better understanding of *PRQ0*.

In addition to these proposed research questions, we also present *PRQ11* which deals with a slightly different subject: near optimal control of a queueing network over a finite time horizon. We are planning to deal with this subject in the upcoming months.

הצעת מחקר לעבודת דוקטור:
יציבות, ניצולת, הוגנות ותפוקה ברשתות עיבוד סטוכסטיות בעלות כניסות אינסופיות

מועמד: יוני נצרתי
מנחה: פרופ' גדעון וייס

<u>תקציר מורחב</u>

זוהי הצעת מחקר לעבודת דוקטור. עבודה זו מסכמת את שאלות המחקר, מתארת את תוכנית העבודה וסוקרת את התחומים הרלוונטיים. לצורך הנוחות, שאלות המחקר מסומנות *PRQ0*, *PRQ1* וכו' (Proposed Research Question). לעיטים אנו מרחיבים סימון זה ומשתמשים בספרות מימין לנקודה העשרונית ובכך יוצרים היררכיה של שאלות מחקר. לדוגמא: *PRQ2.3* ו *PRQ2.5* הינם חלק משאלת מהחקר היותר כללית *PRQ2*.

המחקר עוסק ברשתות עיבוד סטוכסטיות בעלות כניסות אינסופיות - Stochastic Processing Networks with Infinite Inputs (SPNII). זהו מודל תורים סטוכסטי אשר יכול להיות מיושם למספר אפליקציות במפעלי יצור, רשתות תקשורת מורכבות ומערכות כבישים. המודל הינו הכללה של רשתות תורים מרובות מחלקות Multi Class Queuing Networks (MCQN) ורשתות עיבוד סטוכסטיות Stochastic Processing Networks אשר נחקרו ע"י Harrison, Dai ורבים אחרים. ההכללה היא במובן של הוספת האפשרות לכניסות אינסופיות וזאת בניגוד להנחה הסטנדרטית של תהליכי הגאה. דוגמאות פשוטות אך מעניינות נחקרו ע"י Weiss (וייס) ואחרים.

תחילה אנו מציגים את המטרה האולטימטיבית של מחקר זה: בהינתן רשת ספציפית, האם קיימת מדיניות שיבוץ אשר שומרת על יציבות סטוכסטית, מנצלת את כל המשאבים למקסימום האפשרי ומייצרת תוצרים מהרשת על פי הפרופורציות הנדרשות (הוגנות בין זרמי היציאה). זאת השאלה של קיום של מדיניות שיבוץ יציבה, מנצלת במלואה והוגנת (*PRQ0.0*). טבעי להרחיב שאלה זו לבעיית האופטימיזציה ובה אנו מחפשים מדיניות

אופטימאלית מבחינת תעבורה (*PRQ0.1*). את השילוב של שאלות המחקר המוצעות הללו, אנו מכנים *PRQ0*.

אנו מאמינים, כי בשלב זה, *PRQ0* הינה שאלת מחקר אמביציוזית יתר על המידה. אם כך אנו מציעים סדרה של שאלות מחקר יותר ספציפיות אשר קשורות ל *PRQ0*. מחקרם של שאלות אלו יכול לשפוך אור חדש על הדינאמיקה של מודלים מסוג SPNII ולבסוף לתרום לתוצאות חדשות הקשורות ל *PRQ0*.

אנו מציגים סדרת שאלות מחקר הנוגעות למודלי כניסות חוזרות - Reentrant Line ובהם יש כניסה אינסופית של עבודות Reentrant Lines with Infinite Inputs (RLINEII). למרות שמודלים אלו אינם חובקים את כל המרכיבים של מודלי SPNII, מחקרם יוסיף לניסיון וההבנה הדרושה. אנו מתחילים עם *PRQ1* אשר דנה ביציבות של המדיניות אשר נותנת עדיפות לאוגר הקרוב ביותר למוצא המערכת Last Buffer First Serve (LBFS).

שאלות המחקר הבאות דנות במקרה בו משאב צוואר הבקבוק משמש את הפעולה הראשונה: *PRQ1.0* עוסקת במקרה המקרקובי. מקרה זה הוא הפשוט היותר. כאן אנו מנסים להראות התמדה-חיוביות באמצעות שיטות המשתמשות בפונקציות ליאפונוב. *PRQ1.1* מרחיבה את השאלה הקודמת ומציעה חיפוש של התפלגות שווי המשקל. *PRQ1.2* מכלילה את התפלגות זמני השרות ועוסקת ביציבות באמצעות משפטי נוזלים. *PRQ1.3*, מציעה לנסות להוכיח את אותו הדבר באמצעות פונקציות ליאפונוב מעל מרחבי מצבים כללים. לאחר מכן אנו מציגים את *PRQ1.4* אשר דנה במקרה בו משאב צוואר אינו משמש את הפעולה הראשונה.

לאחר הצגת השאלות הקשורות ל LBFS, אנו מציעים להמשיך עם מודלים מסוג RLINEII לטובת מציאת מדיניות אופטימאלית (*PRQ2*). ב*PRQ2.0* אנו מציעים ניסוח של הבעיה כבעיית תכנות דינאמי במקרה המרקובי. ב *PRQ2.1* אנו מציעים מספר מחקרי סימולציה הקשורים לאופטימיזציה ואופטימיזציה הירוסטית של RLINEII ביחס למספר מדדי ביצועים.

ייתכן והתעסקות נכונה עם רוב הבעיות המוצעות בהקשר של RLINEII תצריך מחקר של מודל יותר פשוט. לכן אנו מציעים לחקור את מודל ה RLINEII הפשוט ביותר אשר עדיין מכיל כניסות חוזרות (reentrancy). זהו המודל בעל 2 משאבים ו 3 אוגרים וכניסות אינסופיות 2 Resource 3 Buffer with Infinite Inputs (2R3BII). מודל זה נחקר באופן מקיף ע"י Weiss ושותפים. אנו מציעים מספר הרחבות חשובות אשר עדיין לא טופלו.

רוב המחקר על מודל 2R3BII בוצע ביחס למדיניות LBFS והמקרה המרקובי. ב*PRQ3* אנו מציעים להרחיב את המחקר העוסק במקרה פשוט זה ולבחון מספר תכונות נוספות כגון התפלגות תקופת התעסוקה. ב *PRQ4* אנו מציעים לחקור את מדיניות LBFS כאשר זמני השרות הינם מהתפלגות כללית. התוכנית כאן היא להשתמש בפונקציות ליאפונוב מעל מרחב מצבים כללי. ב *PRQ5* אנו מציעים סדרת בעיות אופטימיזציה הקשורות למודל זה במקרה המרקובי. באופן כללי, רוב שאלות המחקר הקשורות ל 2R3BII מתפצלות לתתי שאלות בהתאם לאיזה מהמשאבים הוא צוואר הבקבוק.

מקרה פרטי נוסף של RLINEII הוא ה Flow Shop. במודל זה אין reentrancy ולכן תכונות מעניינות אשר נובעות ממודלים של רשתות בעלי כניסות אינסופיות מנוטרלות. למרות זאת, אנו מציעים סדרת שאלות מחקר הקשורות לאופטימיזציה של מודל הנוזלים התואם מעל לפרק זמן סופי. שאלות אלו מרחיבות עבודה קודמת של וייס (Weiss) ומשלבות שימוש בתוצאות של Weiss בהקשר לבעיות Separated Continuous Linear Programming (SCLP). אנו מסמנים שאלות אלו ב *PRQ6*.

לאחר מכן אנו מציגים את המודל בעל 2 משאבים ו 4 אוגרים וכניסות אינסופיות (2R4BII). מודל זה נחקר ע"י וייס וקופזון (Weiss and Kopzon) תחת השם מודל Push-Pull. וייס וקופזון מצאו מדיניות מסוימות אשר משיגות ניצולת מלאה תוך כדי שימוש של יציבות. אלו נקראים מדיניות סף מוכללות Generalize Threshold Policies (GTP). תוצאות יציבות אלו הוכחו עד כה אך ורק למקרה המרקובי. אנו מציגים תוצאות סימולציה ראשוניות של מדינויות אחרות: Maximum Pressure Policies (MPP) (אשר נותחו ע"י Dai ו Lin). התוצאות שלנו מראות כי מדינויות MPP אינן יציבות (*PRQ7*) כאשר העומס הוא כזה אשר מאפשר ניצולת מלאה. ע"י השוואת תוצאות סימולציה אלו ל GTP, אנו רואים

את הכוח של GTP. מעבר לכך, ב *PRQ7* אנו מציעים לבצע מחקר סימולציה אשר יראה כי מדיניות GTP הינה יציבה תחת זמן שרות כללים. *PRQ8* דן בהוכחת יציבות של GTP תחת זמני שרות כללים. אנו מציעים לגשת להוכחה זאת שוב באמצעות פונקציות ליאפונוב מעל מרחבי מצבים כללים.

PRQ9 דן בהצגת מודלים נוספים אשר יכולים לגשר לקראת הבנה של *PRQ0*. זהו פער הבנה גדול. בשלב זה, לא ברור מהם השאלות הנוספות אשר יש לחקר ולבדוק בדרכנו למפגש מעמיק עם *PRQ0*. אנו מאמינים שלאחר עבודה קשה על השאלות אשר הוצגו לעיל, יהיה בידינו הניסיון והידע להגדיר את המודלים הבאים למחקר (*PRQ9*). בשלב זה, יתאפשר להתבצע מחקר נוסף על מודלים אלו בדרכנו להבנה יותר מעמיקה של *PRQ0*.

בנוסף לשאלות מחקר מוצעות אלו, אנו מציגים את *PRQ11* הדנה בנושא קצת שונה: שיבוץ כמעט אופטימאלי של רשת תורים עבור אופק זמן סופי. אנו מתכננים לעסוק בנושא מחקר זה בחודשים הקרובים ולכן החלטנו לצרפו להצעת מחקר זו.

Overview

This is a Ph.d research proposal. It summarizes the proposed research questions, states the work plan and surveys relevant background.

The structure of this research proposal is as follows: Chapter 1 presents all of the proposed research questions and minimal relevant background. Chapter 2 indicates the general work plan regarding the research during our Ph.d studies. References are made to the proposed research questions that are enumerated in chapter 1. Chapter 3 presents a very brief overview. The purpose of the overview is both to summarize results that are directly related to our research (infinite input results by Weiss et. al) and to state which other material is to be summarized and aggregated in the dissertation.

Chapter 1

Proposed Research Questions and Preliminary Results

This chapter describes the proposed research questions (PRQ) that are to be investigated during the course of the research. It also contains preliminary results and conjectures already obtained.

It should be noted that the proposed research questions presented in this chapter are presented in a top-down manner. At first the most general questions are posed. These are then followed by specific more precise questions. While the presentation is such, the course of the research will be in a bottom-up manner, meaning that at first the more specific research questions will be tackled and then the more general ones will be handled. It should also be noted that it is highly possible that the whole of the questions posed in this chapter are more than can be handled within the course of 3-5 research years. Thus possibly the most general questions are presented here for completeness while the more specific and concrete questions that are presented here, are the ones that will be fully contained within the research. More on the work plan regarding the research may be found in Chapter 2.

We begin by introducing the problem domain, Stochastic Processing Networks with Infinite Inputs (SPNII) in its most general from in section 1.1.

We then continue and state the most general questions regarding such models in section 1.2. Following that, specific and more concrete questions regarding special cases of our general model are posed. This is performed in section 1.3 where questions regarding re-entrant lines are posed and also in section 1.4 where questions regarding the 2 resource 4 buffer model are posed. Here we also includes some preliminary results and conjectures. Questions regarding future ideas and models, finally introduced in section 1.5. In section 1.6, we include the extended abstract of other work we are planning to perform: Near Optimal Control of Queueing Networks over a Finite Time Horizon.

We use the following notation for introducing proposed research questions: A proposed research question is labeled as $PRQx.y$ where x indicates the number of the research question and y indicates the number of the sub-questions. This notation introduces a hierarchy of proposed research questions for convenience. For example: *PRQ2.3* and *PRQ2.5* are all part of the more general proposed research question *PRQ2*.

The proposed research question *PRQ0* and its sub-questions are our most general questions and are introduced in section 1.2. The other proposed research questions then follow in the following sections.

1.1 The Problem Domain: SPNII Models

A *stochastic processing network with infinite inputs* (SPNII) is a very general stochastic queueing network model. It is now defined. The term *network* is used synonymously with the SPNII model. We attempt to be consistent in our notation with the model presented in [11], yet some notation has been modified due to our generalization.

The Mechanics of the Model

Our network operates in continuous time $t \in [0, \infty)$. We use the term *an operation of the network* to refer to a specific sample path. A SPNII is

composed of the following entities: *jobs*, *buffers*, *activities*, *resources* and *scheduling policies*. Jobs are not directly identified, they are constantly being brought-in, taken-out, merged and split during the operation of the network. Buffers are labeled $k = 1, \ldots, K$. the set of all buffers is $\mathcal{K}$. Activities are labeled $j = 1, \ldots, J$. The set of activities is $\mathcal{J}$. Resources are labeled $r = 1, \ldots, R$. The set of resources is $\mathcal{R}$. Loosely, a scheduling policy defines which activities to perform during an operation of the network.

Buffers of the network are categorized as either *source buffers*, *intermediate buffers* or *destination buffers*. The source buffers always contain an infinite amount of jobs; these correspond to the infinite input nature of the model. The intermediate buffers may be viewed as finite queues. The destination buffers collect jobs as they are removed from the network. $K = S+I+D$, where S, I and D are the amounts of source, intermediate and destination buffers respectively. These sets of buffers are respectively labeled $\mathcal{S}$, $\mathcal{I}$ and $\mathcal{D}$, thus $\mathcal{K} = \mathcal{S} \cup \mathcal{I} \cup \mathcal{D}$.

Activities are the driving force of the network. For each activity, a set of *input buffers* and *output buffers* is specified. A single operation of an activity removes a given amount of jobs from each of the input buffers of the activity and places a given amount of jobs in each of the output buffers of the activity. The input buffers of an activity may be of the source buffer type or the intermediate buffer type, the output buffers of an activity may be of the intermediate buffer type or the destination buffer type.

The $J \times K$ *operation matrix* $\mathbf{B}$ determines how activities operate on buffers. All values in this matrix are integers. The entry $B_{j,k}$ indicates the positive change that buffer k undergoes upon completion of activity j (thus if it is negative then jobs are removed from the buffer). The signs of the entries are restricted based on the type of buffer (source, intermediate or destination): for $k \in \mathcal{S}$, $B_{\cdot,k} \leq 0$; for $k \in \mathcal{D}$, $B_{\cdot,k} \geq 0$; for $k \in \mathcal{I}$ there are no sign restrictions. Note that rows of the operation matrix do not have to sum to 0, thus job splitting and/or merging is allowed. The input buffers of

activity j are $IN(j) = \{k \in \mathcal{K} : B_{j,k} < 0\}$. The output buffers of activity j are $OUT(j) = \{k \in \mathcal{K} : 0 < B_{j,k}\}$.

Jobs are the basic atoms in the network. They are placed in the buffers of the network and circulate between these buffers as time advances. $Q_k(t)$ indicates the number of jobs in buffer k at time t. This value is infinity for $k \in \mathcal{S}$, non-negative for $k \in \mathcal{I}$ and non-decreasing in t for $k \in \mathcal{D}$. The jobs present in a certain buffer are indistinguishable. At time $t = 0$, there are initial job amounts in the intermediate buffers. These are simply denoted by $Q_k(0)$ for $k \in \mathcal{I}$. The produced number of jobs is initially zero: $Q_k(0) = 0$ for $k \in \mathcal{D}$.

Activities are fueled by resources. Several activities may need a shared resource and several resources may be needed by a single activity. Note that resource splitting is not allowed. The $R \times J$ *resource consumption matrix* $\mathbf{A}$ determines the relationships between resources and activities. The entry $A_{r,j} = 1$ if resource r is required for activity j; it is 0 otherwise.

For each activity j, we indicate by $u_j = \{u_j(l), l \geq 1\}$ the set of durations of operations of the activity; the l'th operation's duration is $u_j(l)$. These are non-negative real numbers. We can now define the counting processes $S_j(t) = \max\{n : \sum_{l=1}^{n} u_j(l) \leq t\}$. This is the amount of operations of activity j completed during the first t time units of operation of activity j.

We indicate by $T_j(t)$ the cumulative activity j processing time in $[0, t]$. Obviously $0 \leq T_j(t) \leq t$. Thus $S_j(T_j(t))$ is the amount of operations of activity j during the interval $[0, t]$.

Given initial job amounts in the intermediate buffers $\{Q_k(0), k \in \mathcal{I}\}$, sequences of activity processing times $\{u_j, j \in \mathcal{J}\}$ and cumulative activity processing times $\{T_j(t), j \in \mathcal{J}\}$ the buffer levels at time t may be uniquely determined as follows:

$$Q_k(t) = Q_k(0) + \sum_{j \in \mathcal{J}} S_j(T_j(t)) B_{j,k} \qquad k \in \mathcal{I} \cup \mathcal{D} \tag{1.1}$$

$$Q_k(t) = \infty \qquad k \in \mathcal{S} \tag{1.2}$$

Resources are limited and thus in general, activities may not always be applied simultaneously; this is specified in the *resource consumption constraints*:

$$\sum_{j \in \mathcal{J}} A_{r,j}(T_j(t) - T_j(s)) \leq t - s \qquad s \in [0,t] \quad r \in \mathcal{R} \tag{1.3}$$

$$T_j(0) = 0 \qquad j \in \mathcal{J} \tag{1.4}$$

$$T_j(t) \leq T_j(t+\epsilon) \qquad j \in \mathcal{J} \quad 0 < \epsilon \tag{1.5}$$

Constraints 1.3 require that the utilization of each resource not exceed 1 within any time period. Constraints 1.4 require that at the start of operation of the network resources still haven't been used. Constraints 1.5 require that the cumulative activity processing time be non-decreasing.

While there are always jobs present in the source buffers, the intermediate buffers act as finite queues. These queues may be empty during certain periods but they may not be negative. These are the *queue size constraints*:

$$0 \leq Q_k(t) \qquad 0 \leq t \quad k \in \mathcal{I} \tag{1.6}$$

Comparing the our model to the model described by Dai and Lin in [11], we find two major differences: (1) Their model allows for probabilistic routing while we have chosen to avoid this feature for simplicity. (2) Their model does not allow activities operating on source buffers to share resources with other activities while our model does not have this restriction. It should be evident that difference (2) makes the infinite input feature of our model interesting because it allows tradeoffs of scheduling activities that pull from source buffers and other activities.

Scheduling Policies

A *network model instance* is given by the following set of parameters: $(\mathcal{S}, \mathcal{I}, \mathcal{D}, \mathbf{B}, \mathbf{A}, \{Q_k(0), k \in \mathcal{I}\})$. We denote by $\mathcal{M}$ the set of all network model instances.

Assume we are given a network model instance $M \in \mathcal{M}$. A *processing times instance* is a sequence of processing times $\{u_j, j \in \mathcal{J}\}$. We denote the set of all processing times instances by $\mathcal{PT}(M)$.

Assume we are given a network model instance $M \in \mathcal{M}$ and a corresponding processing times instance $PT \in \mathcal{PT}(M)$. A *feasible network schedule* is the finite set of functions $\{T_j(t),\ j \in \mathcal{J}\}$ that satisfy constraints 1.3 - 1.6 while the network dynamics follow 1.1 and 1.2. We denote the set of all feasible network schedules by $\mathcal{T}(M, PT)$. This set is trivially non empty because $T_j(t) = 0$, $j \in \mathcal{J}$, is a feasible network schedule.

It should be noted that for a given network model instance and processing times instance the cumulative activity processing times completely describe the operation of the network according to equation 1.1. This leads to the definition of a scheduling policy.

Assume we are given a network model instance $M \in \mathcal{M}$. A *scheduling policy* for this network model is a mapping $P : PT(M) \rightarrow \mathcal{T}(M, PT)$. We denote the set of all scheduling policies by $\mathcal{P}'(M)$.

Our framework is an on-line frame work and not a combinatorial optimization framework. We will thus not be interested in all scheduling policies but only in those policies that cannot take future processing times into consideration. In one sense this concept may be treated as requiring a scheduling policy to be a mapping from the "state" of the network to a decision regarding which activities to perform. Note that we do not deal with probabilistic scheduling policies in this work.

We currently do not know how to provide a useful rigorous definition of on-line scheduling policies. Nevertheless, we denote the set of all such policies

by $\mathcal{P}(M)$. As will be seen in section 1.2, *PRQ0* deals with optimization over this set. We believe that a rigorous definition of $\mathcal{P}(M)$ requires more insight into the nature of *PRQ0* then we have at this time.

It should be noted that our model allows preemption. This is evident since we did not specify any non-preemption restrictions in our definition of scheduling policies.

Probabilistic Assumptions

We assume that the durations of operations of the activities are random variables. These are in turn the primitive processes of our model: given a value for the durations (a processing time instance) the dynamics of the network with respect to a scheduling policy are fully determined.

We assume that all processing times are statistically independent. We assume the processing times of a given activity are identically distributed with distribution F_j. We assume that the processing times have a positive real mean: $m_j = E[u_j(1)]$. In short, each sequence u_j is an i.i.d. sequence with mean m_j. Assume we are given a network model instance $M \in \mathcal{M}$. We denote the set of all possible processing time means by $A(M)$.

While we do not make further assumptions at this point, it should be noted that several (but not all) of the proposed research questions deal with the Markovian case in which $u_j(1) \sim \exp(m_j^{-1})$.

1.1.1 Attributes and Measures of Performance

We will now discuss the four attributes and measures of performance that are stated in the title of this research proposal: stability, utilization, fairness and throughput. Our description assumes that we are given a network model instance $M \in \mathcal{M}$, a corresponding scheduling policy $P \in \mathcal{P}(M)$ and a corresponding processing time means vector $A(M)$. For this section we use the term "network" to refer to the combination of these items.

We will assume that $Q_k(t)$ are random variables arising in an operation of this network. We assume that all expectations and limits exist and that the network is ergodic. We do not make these assumptions explicit in this proposal but rather assume that the definitions may be made rigorous. Attempting to handle *PRQ0* will require more rigorous definitions.

Stability

We say our network is *stable* if $lim_{t\to\infty} Q_k(t) < \infty$ a.s. for $k \in \mathcal{I}$.

Utilization

We define the utilization of resource r as $\rho_r = lim_{t\to\infty} \sum_{j\in\mathcal{J}} A_{r,j} T_j(t)/t$. It is evident that the utilization is at most 1. We define the *minimum target utilizations* as $\{\widehat{\rho}_r,\ r \in \mathcal{R}\}$. We say the network *meets targets utilization* if $\widehat{\rho}_r \leq \rho_r$ for $r \in \mathcal{R}$ a.s.

Ideally we will be interested in having target utilizations of 1, but this is not always possible if we require stability. We thus see the property of meeting target utilization implies that we are utilizing our resources by at least a predetermined amount (the target utilization).

Fairness

We define the *total jobs arriving to destinations* by time t as $Q_{\mathcal{D}}(t) = \sum_{k\in\mathcal{D}} Q_k(t)$. We define the *proportion of production of* k as $DP_k = lim_{t\to\infty} Q_k(t)/Q_{\mathcal{D}}(t)$. For a given network we are interested in *target proportions*: $\{\widehat{DP}_k,\ k \in \mathcal{D}\}$ such that $\sum_{k\in\mathcal{D}} \widehat{DP}_k = 1$. We will say that our network is *fair* if $DP_k = \widehat{DP}_k$ for $k \in \mathcal{D}$ a.s.

Thus the fairness of the network is a property of being able to produce jobs to destination buffers at the target proportion. We will see that there are some networks that are fair with respect to some target proportions but not others (such an example is the 2R4BII model).

Throughput

The throughput of the network is $lim_{t\to\infty} Q_{\mathcal{D}}(t)/t$. Again we assume that this limit exists and that it is the same for every realization.

1.1.2 An Example

We now present an example of an SPNII model. This example tries to incorporate all of the interesting features that are available in SPNII models: job splitting, job merging, routing and tradeoffs between pushing new jobs into the system or pulling jobs towards destination buffers.

These are the buffers: $K = 6$, $S = 1$, $I = 3$, $D = 2$, $\mathcal{S} = \{1\}$, $\mathcal{I} = \{2, 3, 4\}$, $\mathcal{D} = \{5, 6\}$. These are the activities: $J = 7$, $\mathcal{J} = \{1, \ldots, 7\}$. This is the 7×6 operation matrix.

$$\mathbf{B} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & -1 & -3 & 0 & 1 & 2 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$$

These are the resources: $R = 5$, $\mathcal{R} = \{1, \ldots, 5\}$. This is the 5×7 resource consumption matrix:

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

We set $Q_k(0) = 0$, $k \in \{2, 3, 4\}$. The processing times means of the activities are as follows: $m_1 = 10.0$ and $m_j = 1.0$, $j \in \{2, \ldots, 7\}$. The following page contains an illustration of this example. The squares in the

illustration denote buffers, the circles denote activities and the groupings of activities denote resources. The values of the operation matrix are labeled by the arrows coming in to an activity and the arrows going out. These are the input buffers and output buffers respectively.

There are several points to notice regarding this example and SPNII models in general:

- From a mathematical point of view, it is meaningless to have more than one source buffer. Nevertheless, it may sometimes be easier to model applications in this manner.

- SPNII models allow to model stochastic arrival streams. This is evident in the example with activity 1 and resource 1. Assuming that our policy is work conserving with respect to resource 1, this resource and activity model an arrival process of jobs into buffer 2.

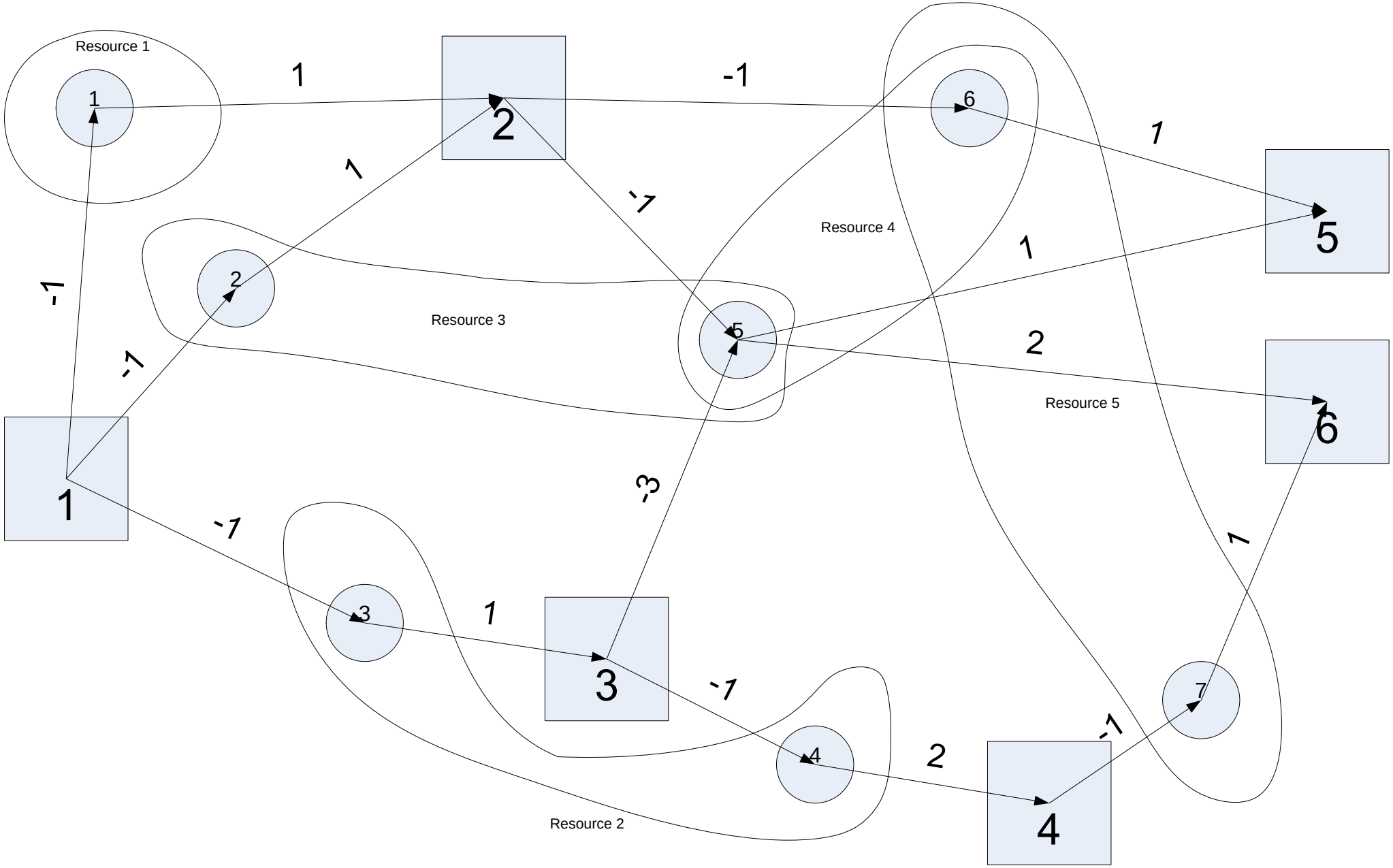
- The allocation of resources 3 and 4 is directly linked to the stability of the system. Examining resource 3 for example, it may be used to perform activity 2 or activity 5. It is evident that performing activity 2 increases the number of jobs in the system while performing of activity 5 decreases this number. We will be interested in finding policies that allow full (or at least maximal) utilization of these resources while maintaining the system stable. We thus expect these policies to balance between work that pulls jobs from the input buffers, and work that pushes jobs out towards draining of the system.

- Notice that this system produces two products, these are set to the destination buffers 5 and 6. We will be interested in finding policies that produce these products according to some target proportions (fairness).

- Activity 5 is an example of an activity that performs both job splitting and job merging: each application of this activity takes 4 jobs from its

input buffers and outputs 3 jobs to its output buffers.

- Notice that our model supports routing, this may be seen for example by looking at buffer 3. Jobs at this buffer may either be processed by activity 5 (which processes 3 jobs at a time) or by activity 4. This may be seen as a routing decision that has to be made.


Resource 1
1
1
2
-1
6
1
5
1
-1
-1
Resource 4
2
Resource 3
1
5
-1
2
Resource 5
6
1
-3
-1
1
3
3
1
-1
7
-1
4
2
4
Resource 2

1.1.3 Applications

We now discuss how SPNII models can be used in manufacturing and communications applications. We also believe that these models can be used to model rush hour road traffic and overloaded call centers without abandonments but we do not go into the details here.

Manufacturing

The manufacturing setting is a classic setting for SPNII models. Infinite inputs are in many situations the natural assumption to make (as opposed to stochastic inputs). This is because in many instances factories are required to produce in the short term at a maximal rate and there is no scarcity of raw materials (inputs)

A reasonable way to apply our model to manufacturing is as follows: The network models a *factory*. The factory has *machines* which are modeled as resources in our model. There are *tasks* that are performed by these machines (sometimes several machines may collaborate to perform a single task), these tasks are modeled as activities in our model. Manufactured *parts* traverse through the factory and are processed by the machines performing the tasks. The parts which are modeled as *jobs* reside in *bins* which are modeled as buffers in our model. All parts residing in a bin are homogenous. The bins of parts are either bins of *raw materials* which are modeled as source buffers, bins of *intermediate parts* that have not yet been released from the factory which are modeled as intermediate buffers and bins of *finished parts (products)* which are modeled as destination buffers.

When a task is performed it moves parts from its input bins to its output bins. Routing may be incorporated in the model by defining several tasks that are similar in the machines that they use but differ in some of their output bins. Merging and splitting of parts is also possible. This is because there is no requirement that there be a one to one correspondence between

the input parts and the output parts of each task.

Our four attributes and measures of performance are highly important in the manufacturing setting. Stability is natural to require because without it, the bins of intermediate parts will eventually overflow. In addition, without stability, cycle times will continuously increase. Even if a finite horizon situation is considered, stability will usually yield low inventory costs. High utilization of resources is a natural measure of efficiency: it is highly inefficient to invest millions in an expensive machine and end up not using it close to a 100% of the time. Fairness is important when the factory produces more then one finished product. In an attempt to maximize throughput, situations may arise where a big quantity of one product is manufactured while a much smaller quantity of another product is manufactured and this contradicts the desires of the factory management. We say that this situation isn't fair. Finally throughput is the overall measure of performance of a factory: "how fast can it produce". As we will see in section 1.2, we believe that a natural question is that of attempting to maximize throughput subject to some constraints regarding stability, utilization and fairness

Communication Networks

Packet switched communication networks are usually viewed as a distributed mechanism for transferring messages. This mechanism is composed of nodes and links. Decisions regarding link allocation and route selection must be taken. Messages entering the network usually have a source and a sink and the network's role is to transfer the messages between the source and the sink. Under this view, communication networks are usually modeled has having stochastic inputs rather then infinite inputs. This is reasonable because it is assumed that messages arrive to the network according to some arrival process. The goal is usually to find media access control (MAC) and routing policies that maintain stability and fairness while maximizing throughput.

There is less focus on maximizing utilization. This is evident since in practice network links are usually over-provisioned heavily.

We believe that there are certain types of communication networks that can be modeled more appropriately with infinite inputs. In these types of networks, full utilization of the network is one of the goals of the scheduling mechanisms. While it does not make sense to have an infinite supply of messages because the network will not be able to service all messages, it does make sense to have a infinite supply of information to transport. The assumption here is that the application using the network will be able to make good use of as much bandwidth as it is given.

As a simple illustrative example, consider a network where a source node is equipped with a video camera and a sink node is equipped with a monitor that can display video. Assume that both the video camera improve their quality indefinitely as they are offered a higher bit rate. It will then make sense to utilize the network to the fullest between the source and the sink and thus achieve the highest quality video stream possible.

When applying SPNII models to communication networks we treat the source buffers as sources of information and the destination buffers as consumers of information. Jobs signify chunks of information. These are not necessarily packets because jobs may merge or split while traversing the network. The resources are the communication links. Activities correspond to transmission and receiving of information.

1.2 *PRQ0*: Optimization of Throughput Subject to Stability, Utilization and Fairness

Given an SPNII model we would like to be able to perform scheduling on it in a stable, properly utilizing and fair manner. In addition we would like to be able to maximize throughput as we perform this scheduling. We convert

this idea to the following research question:

> *PRQ0.0*: Find algorithms, characterizations and theorems that may answer this: Assume we are given an SPNII model instance M, a corresponding processing time means vector $A(M)$, corresponding target utilizations $\{\widehat{\rho}_r,\ r \in \mathcal{R}\}$ and corresponding target proportions $\{\widehat{DP}_k,\ k \in \mathcal{D}\}$. Does there exist a scheduling policy $P \in \mathcal{P}(M)$ such that the network is stable, meets target utilization, is fair with respect to the target proportions and has positive throughput? Should such a policy exist, what is it?

Our framework has made it clear that we are interested in demanding stability, pinning down utilization and fairness and attempting to maximize throughput. We thus enhance the previous research question to this optimization problem:

> *PRQ0.1*: Find algorithms, characterizations and theorems that may answer this: Assume that we are given an SPNII model as in *PRQ0.0*. What is the maximal achievable throughput?

We believe that the above is a deep and intriguing question for which the answer is not near. The continuation of this chapter will handle very special cases and propose related research questions.

Note that the 2R4BII model (defined in section 1.4), was the motivating example for this broad research question. This is because it is an SPNII model in for which there exist stable, and fully utilizing policies. Note though that in achieving the full utilization, the target proportions are very heavily constrained.

1.3 RLINEII Models

A *reentrant line with infinite inputs* (RLINEII) is a SPNII model in which $S = 1$, $D = 1$, $J = I + 1$ and each row of the $I + 1 \times (I + 2)$ operation matrix

$\mathbf{B}$ is all zeros except for a single -1 entry and a single 1 entry. In addition each row of the resource requirement matrix is all zeros except for a single 1. This definition implies the existence of a single route that is followed by each of the processed jobs. The route must reenter some resources when $R < I$. There is also no need for a distinction between activities and buffers because each activity processes a unique buffer. Note that we set $\mathcal{S} = \{1\}$ (the single source buffer is labeled 1). In addition we assume that the activity operating on this buffer 1 and the rest of the activities/buffers are labeled in the order of the route. This implies that the operation matrix's center diagonal is filled with -1 values and the diagonal to the right of it is filled with 1 values.

We define the *constituency* of resource r to be $C_r = \{j \ \in \mathcal{J} : A_{r,j} = 1\}$. The *workload* of resource r is $w_r = \sum_{j \in C_r} m_j$, this is the amount of work that is required by resource r that a single job requires. The bottleneck is defined as $r^* = \arg\max_r w_r$. This can either be a single resource (single bottleneck) or a set of resources (multiple bottlenecks). We will be primarily interested in the case where there is a single bottleneck.

This is a generalization of the reentrant line model that has been introduced by Kumar in [23] and investigated heavily in recent years. We believe that the generalization of allowing infinite inputs may be more applicable for modeling complex manufacturing situations than the original models that exhibit stochastic arrivals.

The scheduling policy indicates which buffer a resource should serve. When preemption is allowed this question arises at all time instances during the operation of the reentrant line. When preemption is not allowed, this question is only posed at the completion time of each operation. We will treat the last buffer first serve (LBFS) policy extensively. This scheduling policy gives priority of resource r to the highest numbered activity whose corresponding buffer is non-empty. This policy has been investigated with regards to models with infinite inputs in [24], [30] and [1]. It has also been addressed widely with regards to reentrant lines with stochastic inputs (see

[37]). In addition to LBFS, we will also address optimization problems with regards to finding optimal policies.

We begin by addressing questions regarding general RLINEII models in section 1.3.1. Continuing to section 1.3.2, we define a simple yet interesting RLINEII having only 2 resources and 3 buffers (the 2R3BII model) and introduce specific research questions relating to it in section. Finally in section 1.3.3, flow shops and corresponding research questions are introduced.

1.3.1 *PRQ1*, *PRQ2*: The General RLINEII Model

The proposed research questions contained in *PRQ1* deal with stability under the LBFS policy and those contained in *PRQ2* deal with finding optimal policies.

Stability with a Single Bottleneck at the First Buffer

Assume that r^* contains a single resource (single bottleneck) and that the bottleneck resource is the one operating on the source buffer ($1 \in C_{r^*}$). This is the *single bottleneck at first buffer* case. Our simulation results in [24] have led us to believe that LBFS is stable in this case under any service time distribution (having a finite expectation). This has not yet been proven. Hence we propose the following research questions.

> *PRQ1.0*: Prove that an RLINEII model with a single bottleneck at the first buffer is stable under the LBFS policy when the processing times are exponential. Use the Foster-Lyapunov criterion for this.

> *PRQ1.1*: Find the steady state distribution of an RLINEII model with a single bottleneck at the first buffer when the LBFS policy is used and the processing times are exponential. Attempt

to perform this separately for the case where preemptions are allowed or not.

PRQ1.2: Prove that an RLINEII model with a single bottleneck at the first buffer is stable under the LBFS policy when the processing times are from a general distribution. Do this using adaptations of fluid model methods introduced by Dai in [9].

PRQ1.3: Prove that an RLINEII model with a single bottleneck at the first buffer is stable under the LBFS policy when the processing times are from a general distribution. Do this using Lyapunov function methods on general state spaces as describe by Foss and Konstantopoulos in [13].

Behavior with Single Bottleneck Anywhere Case

When the bottleneck resource is not the resource used for the source buffer ($1 \notin C_{r^*}$) we do not believe that all buffers are stable under LBFS. In [24], we have analyzed the fluid model for LBFS in this case and have found the following: As time progresses, the fluid amount of some buffers remains 0 while the fluid amount of other buffers increases at a constant rate. We have stated an algorithm for finding the set of *constant buffers* and the set of *increasing buffers* and their rate of increase. We have conjectured in [24] that the corresponding stochastic process converges to a steady state distribution for the constant buffers and is continuously increasing for the increasing buffers. This leads us to the following research question:

PRQ1.4: Prove that an RLINEII model with a single bottleneck not used for the source buffer is stable for the constant buffers and increasing at the appropriate rate for the increasing buffers. This proof should be based on our algorithm from [24]

and should make use of adaptations of the fluid model methods from [9].

Optimization

Up to now we have discussed the LBFS policy. We are now interested in finding optimal policies:

> *PRQ2.0*: For the case of exponential processing times, formulate a dynamic programming problem for optimal scheduling of an RLINEII model in terms of minimizing the expectation of $\lim_{t\to\infty} \sum_{k\in\mathcal{I}} a_k Q_k(t)$.

We do not believe that for general RLINEII models, much more can be done than *PRQ2.0*. Nevertheless, it is a pressing practical problem to find optimal or near optimal policies. One approach is simply to apply LBFS (since we believe that it is stable), another is to use approximation techniques that involve iterations of simulation and model adjustment to find near optimal policies or heuristics:

> *PRQ2.1*: Devise a mechanism (a novel approach) for finding near optimal heuristics for RLINEII models with general service times.

1.3.2 *PRQ3*, *PRQ4*, *PRQ5*: The 2R3BII Model

We now introduce the *2 resource 3 buffer infinite input model* (2R3BII). This is the simplest interesting RLINEII model since it involves reentrancy with the minimal number of buffers and demonstrates the critical scheduling decision that must be taken. It has been investigated previously by Weiss and Adan (see section 3.3.3).

This is the model: $\mathcal{S} = \{1\}$, $\mathcal{I} = \{2, 3\}$, $\mathcal{D} = \{4\}$, $\mathcal{J} = \{1, 2, 3\}$, $\mathcal{R} = \{1, 2\}$. This is the 3×4 operation matrix and the 2×3 resource consumption matrix:

$$\mathbf{B} = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

We treat two cases: (1) Bottleneck is resource 1 ($m_1 + m_3 > m_2$). (2) Bottleneck is resource 2 ($m_1 + m_3 < m_2$). In case 1, we strive for full utilization of resource 1, the bottleneck. This can be achieved by any non-idling policy of resource 1, that maintains buffers 2 and 3 stable. It has been shown that LBFS is such as policy . In case 2, we are not able to fully utilize resource 2, the bottleneck. We are thus looking for a policy that will utilize it at a given rate $\rho < 1$. One such policy sets a threshold B_2 for buffer 2. When the number of jobs at buffer 2 drops below B_2, resource 1 is allocated for buffer 1, otherwise it is allocated for buffer 3. We can now set the target ρ arbitrarily close to 1 by increasing this threshold.

We propose the following types of research questions regarding this model: Standard queueing theory analysis (*PRQ3*), stability under general service time distributions with LBFS (*PRQ4*), optimizing scheduling policies (*PRQ5*).

Standard Queueing Theory Analysis

We may treat the 2R3BII model as a standard queueing system. It may be possible to use traditional techniques to gain some insight:

> *PRQ3.0*: Analyze the 2R3BII model when resource 1 is the bottleneck operating under LBFS. Attempt to find the distribution of the sojourn time of a job in the system.

PRQ3.1: Assume that the processing time of one of the buffers is taken from a general distribution and the processing time of the other 2 buffers are exponential. Attempt to adapt results regarding busy period analysis of M/G/1 queues with vacations for finding busy period distributions in the system and possibly the steady state distribution of buffer 2 and buffer 3.

Stability with LBFS

We propose the following research questions:

PRQ4.0: When the bottleneck is resource 1, prove that the corresponding Markov chain is positive Harris recurrent under general service time distributions. Use Lyapunov function methods.

PRQ4.1: When the bottleneck is resource 1, prove that the corresponding Markov chain is positive Harris recurrent under general service time distributions. Adapt fluid model methods.

Optimization

We will assume that the processing times are exponential. In this case the resulting state space is a grid on the positive quadrant. Assume we are trying to find an optimal policy for minimizing the expectation of $\lim_{t\to\infty} a_2 Q_2(t) + a_3 Q_3(t)$. Some research questions arise:

PRQ5.0: Formulate the dynamic programming optimization problem for the case in which resource 1 is the bottleneck, assuming that full utilization is achieved.

PRQ5.1: Given at target $\rho < 1$ for resource 2, formulate the dynamic programming optimization problem for the case in which resource 2 is the bottleneck.

PRQ5.1: For the case in which resource 1 is the bottleneck, prove or disprove the existence of a switching curve.

PRQ5.2: Find the optimal policy in the case that resource 1 is the bottleneck.

PRQ5.3: Find the optimal policy in the case that resource 2 is the bottleneck.

1.3.3 *PRQ6*: The Flow Shop Model

The flow shop with infinite inputs is a reentrant line where $R = J$. Each resource performs a single activity and thus there is no renterancy. It is basically a tandem queueing systems with $R-1$ queues in tandem and an arrival rate of $1/m_1$. Thus, it does not have the interesting attributes that models with infinite inputs exhibit (the scheduling dilemma regarding allocation of a resource for pushing a job into the system or pulling one out).

Nevertheless, during the preparation of this research proposal we have examined some interesting results relating to this model. These are results from the preprint of Weiss [32]. In this work, Weiss treats the flow shop as a finite time horizon fluid model with given initial amounts $Q_k(0),\ k = 1, \ldots, K$. The goal is to calculate optimal fluid flows in the time interval $[0, T]$ for minimizing $\int_0^T \sum_{k=1}^K c_k Q_k(t)dt$, where $c_k,\ k = 1, \ldots, K$ are holding costs constants. There are still some open issues regarding Weiss's algorithm:

PRQ6.0: Is the flow shop algorithm a polynomial time algorithm in the number of buffers? The proof for this in [32] is still open.

PRQ6.1: The current description of the algorithm does not yield an immediate simple implementation. We thus propose to implement the algorithm and in the process refine it's description.

1.4 2R4BII Models

Moving onward from reentrant line models, we introduce the *2 resource 4 buffer infinite input model* (2R4BII). This model has been termed the push-pull model by Weiss and Kopzon (see section 3.3.4 for a brief review).

Using our SPNII framework, this is the model: $\mathcal{S} = \{11, 21\}$, $\mathcal{I} = \{12, 22\}$, $\mathcal{D} = \{13, 23\}$, $\mathcal{J} = \{11, 12, 21, 22\}$, $\mathcal{R} = \{1, 2\}$. We have named the buffers and activities using numbers with two decimal digits in this manner: The model contains two routes (1 and 2). Each route contains a source buffer, an intermediate buffer and a destination buffer. Thus for example for route 2, the source buffer is 21, the intermediate buffer is 22 and the destination buffer is 23. Each activity corresponds to exactly one source or intermediate buffer. As a result of this description This is the 4×6 operation matrix and the 2×4 resource consumption matrix (ordering activities the order $(11, 12, 21, 22)$ and buffers in the order $(11, 12, 13, 21, 22, 23)$):

$$\mathbf{B} = \begin{pmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

We label the processing rates (the reciprocal of the means) for the activities $(11, 12, 21, 22)$ by $(\lambda_1, \mu_1, \lambda_2, \mu_2)$.

This model is the motivating model for our entire study. The results of Weiss and Kopzon have shown that fully utilizing, stable policies are achievable. It should be noted though, that they are only achievable under specific growth rates of the destination buffers, hence there is little flexibility in fairness. See section 3.3.4 for a calculation of these rates.

We mainly intend to augment the work of Weiss and Kopzon by handling general processing times (as opposed to exponential). This is described with

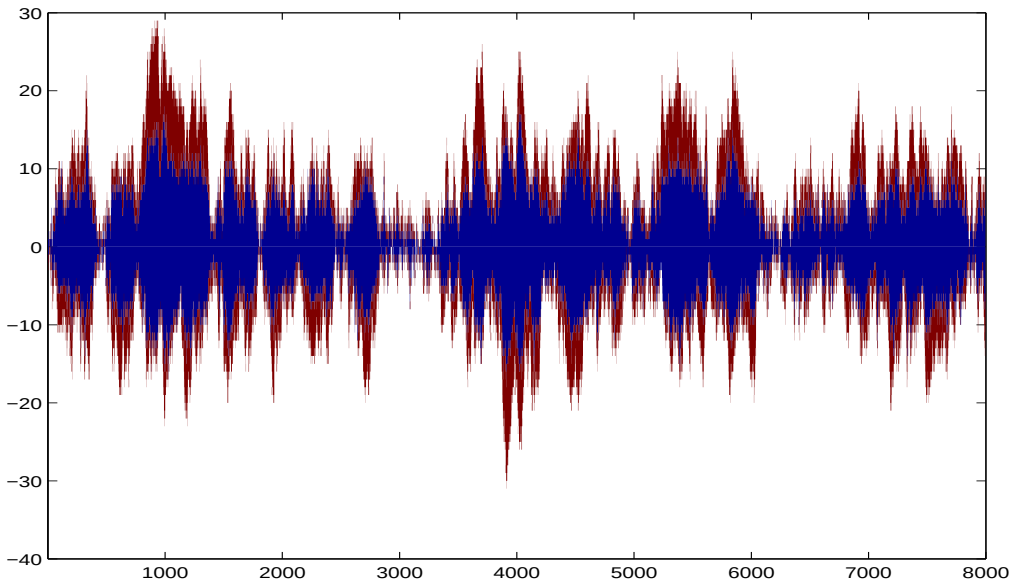
regards to *PRQ8* in section 1.4.2. In addition, we have recently contributed some simulation results to an upcoming publication that is based on [22]. We describe our contribution and future work (*PRQ7*) in the following section.

1.4.1 *PRQ7*: Preliminary Simulation Results

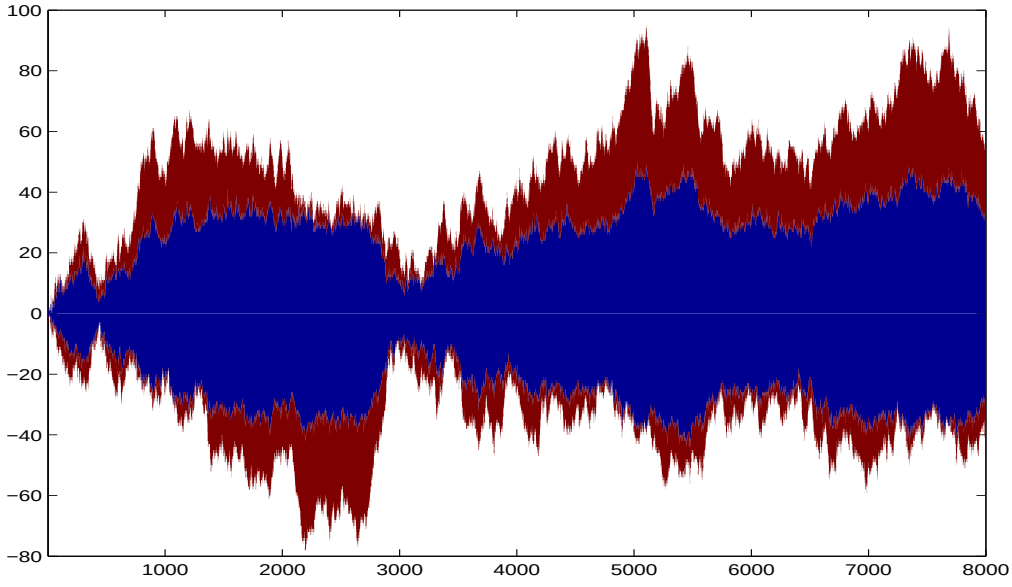
The 2R4BII model is based on a the Rybko Stolyar (RS) queueing network. The difference is that in the RS network there aren't infinite inputs but rather stochastic inputs. See [8] for a description of the RS network and related results. Technically, the RS network may be modeled as an SPNII by taking the 2R4BII model, converting buffers 11 and 21 to intermediate (finite) buffers and adding source buffers 10 and 20 each with their own "private" resource and activities which move jobs from them into 11 and 21 respectively. The rates of the RS source buffers are α_1 and α_2 respectively (these are the arrival rates). The arrival rates along with the service rates determined the offered load on each of the resources. For resource 1: $\rho_1 = \alpha_1\lambda_1 + \alpha_2\mu_2$. For resource 2: $\rho_2 = \alpha_2\lambda_2 + \alpha_1\mu_1$.

We have implemented a simulation of the 2R4BII model and the RS network that uses the maximum pressure policy ([11]). For the RS network we have used exponential processing times with the following parameters $\lambda_1 = \lambda_2 = 1.25$, $\mu_1 = \mu_2 = 1.0$ and $\alpha_1 = \alpha_2$ are set to four different values such that $\rho_1 = \rho_2$ takes one of these four values: $(0.9, 0.99, 1.0, 1.2)$. Following are four sample paths that were obtained for these increasing offered loads:

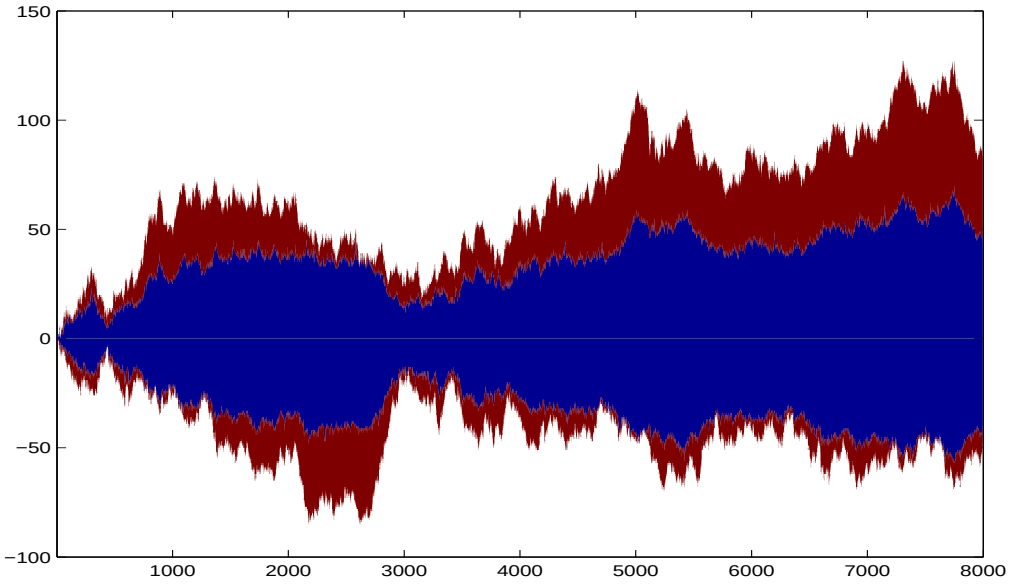
This is for $\rho = .9$:



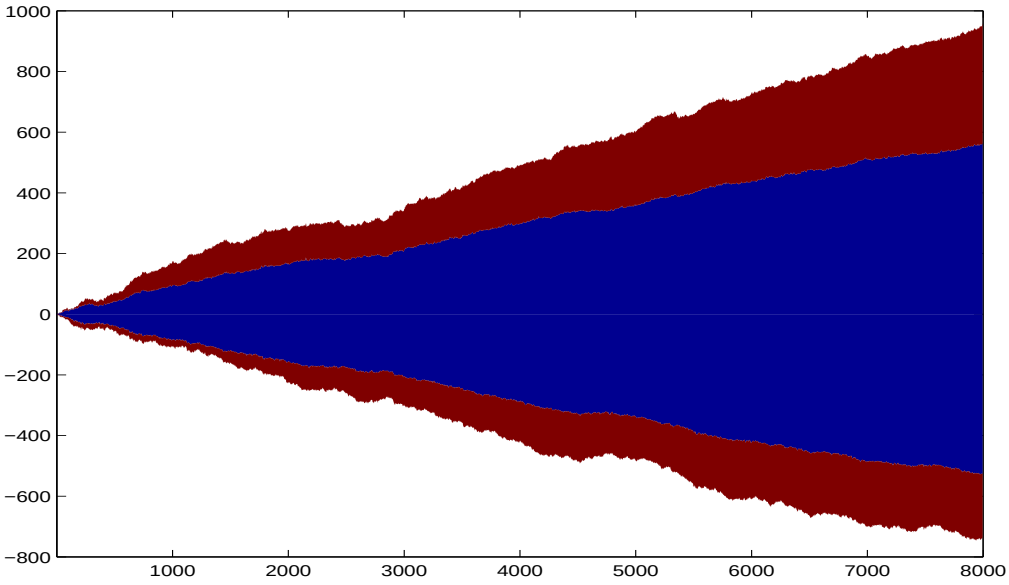
This is for $\rho = .99$:



This is for $\rho = 1.0$:



This is for $\rho = 1.2$:

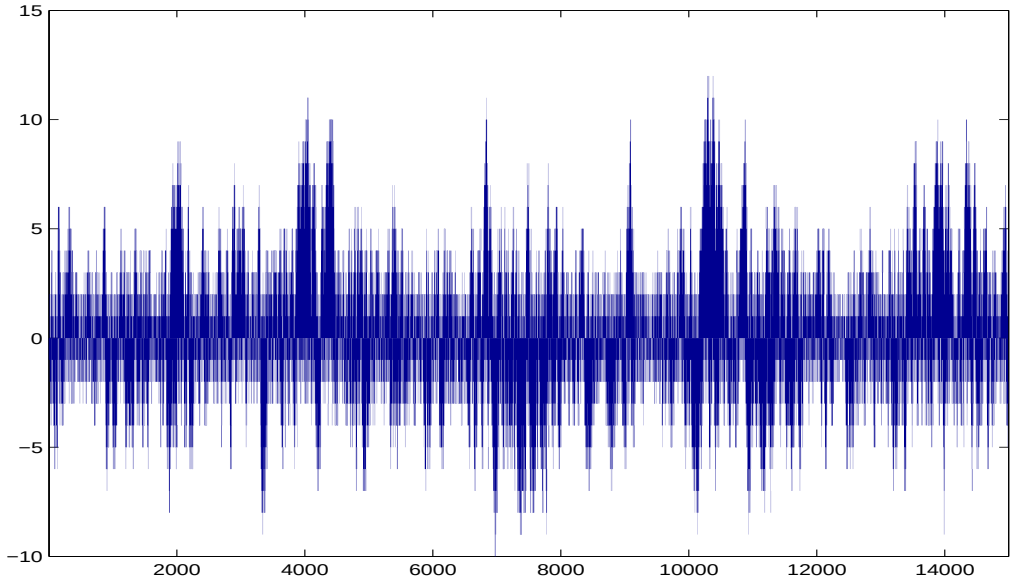


Each illustration plots the evolution of four queue sizes. Above the x-axis the queues of route 1 are plotted (first 11 and above it 12). Below the x-axis, the queues of route 2 are plotted similarly.

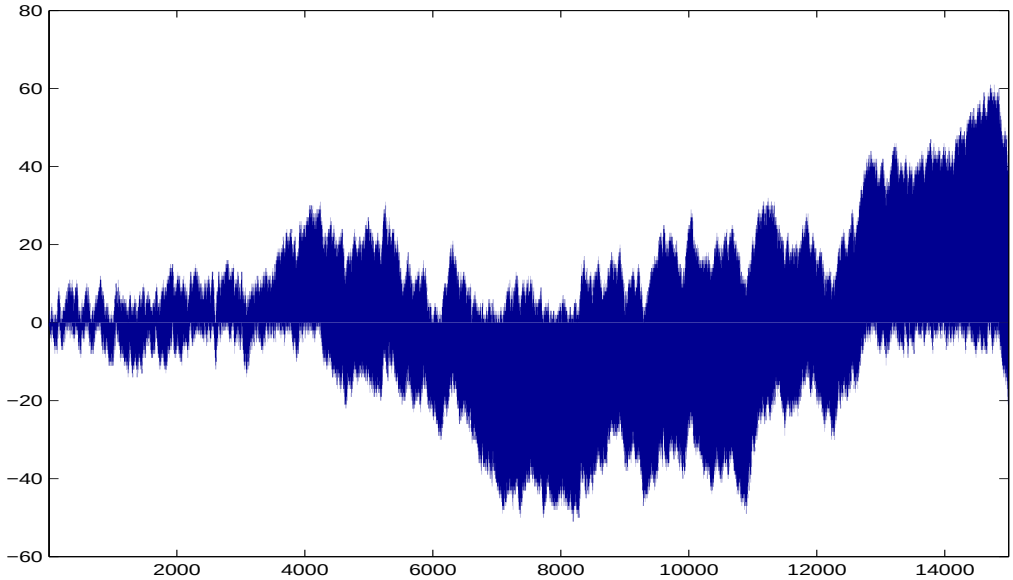
Following the simulation of the RS network (using maximum pressure) we simulated the 2R4BII network also using maximum pressure. In this case, the maximum pressure was applied as follows: Maximum pressure requires a value for the grading the state of both the source queues and the intermediate queues. For the intermediate queues we used the normal queue size and for the source queue we use this value: $\alpha t - D_k(t)$ where $D_k(t)$ is the number of jobs which have been taken out of the source buffer by time t.

Following are plots of four sample paths using the same parameters as before. The plots now indicate buffer 12 above the x-axis and buffer 22 below the x-axis.

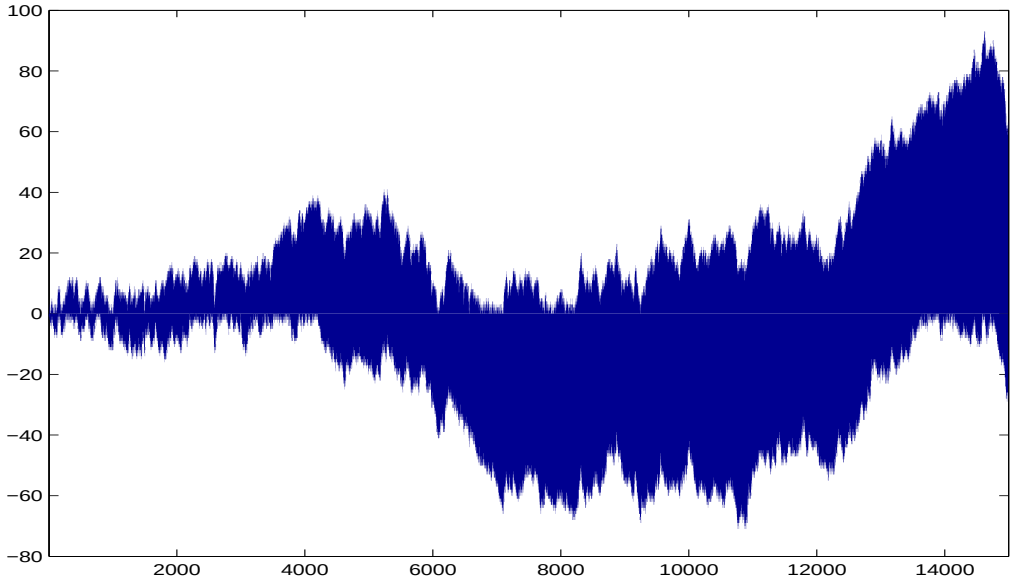
This is for $\rho = .9$:



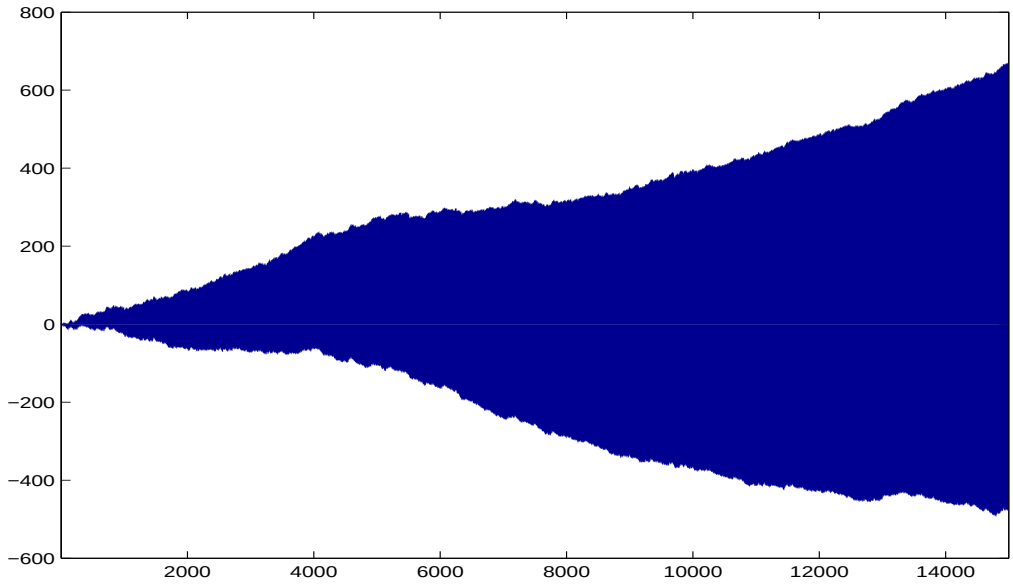
This is for $\rho = .99$:



This is for $\rho = 1.0$:



This is for $\rho = 1.2$:



What can be seen from these traces is that in steady state systems $\rho < 1$, a stable level of queue length is reached after a period, where both the time to stabilize and the level at which the queues stabilize grows with ρ. We believe that the cases of $\rho = 1$, namely full utilization, are transient for both systems, so that while they look quite similar to the case of $\rho = .99$ over most of the time horizon of the simulation, they would not stabilize but will continue to have longer and longer excursion with very high queue lengths. The overloaded cases clearly show linear growth of the queues.

Weiss and Kopzon have used these results to compare maximum pressure to their threshold policies. They have shown that the threshold policies are stable even when $\rho = 1$. As we continue to research the threshold policies with regards to this model, we will be interested in continued use of our simulation software. Hence we propose the following research question:

PRQ7: Simulate the 2R4BII model using both maximum pressure and several variants of the generalized threshold policies. Use

processing time distributions from several families. Use the simulation results to conjecture with regards to stability, instability of the policies and to compare the expected queue length achieved in several stable policies.

1.4.2 *PRQ8*: Stability Under a GTP with General Service Time Distributions.

The stability results of Weiss and Kopzon regarding generalized threshold policies (GTP) has only been obtained when the processing distributions are exponential. Under these exponential (memoryless) distributions the state space is the countable grid in the positive quadrant. We conjecture that this result continues to be true for general processing times. Perhaps there are some technical conditions on the distributions such as being spread out (see [3] for a description on this subject or [9] for an example of using it)? We thus introduce the following research questions:

PRQ8.0: Find technical conditions needed for stability of the generalized threshold policies when the service time distributions are general.

PRQ8.1: Prove that the fixed threshold policy is stable when the service time distribution is general.

PRQ8.1: Prove that the generalized threshold policy is stable when the service time distribution is general.

1.5 *PRQ9*: Further Models

Up to this point we have described a variety of specific research questions. Some were precisely defined and some were a bit more vague, but in general they all dealt with concrete questions. As we have stated, our long term

goal is to analyze the tradeoffs, techniques and possibilities regarding SPNII models with regards to stability, utilization, fairness and throughout. Each of our proposed research questions attempts to make a step with regards to this issue.

It should be noted though, that almost all of the questions do not relate to issues of, routing, job merging and job splitting. In fact, they almost all relate to multi-class queueing networks with infinite inputs and do not use the more general modeling possibilities of stochastic processing networks. We would thus like to define some more interesting, simple and hopefully tractable examples that will yield further insight:

PRQ9.0: Define a tractable model with infinite inputs that embeds with in it some routing decision. Pose feasible questions regarding this model.

PRQ9.1: Define a tractable model with infinite inputs that embeds with in it some tradeoffs regarding fairness. Pose feasible questions regarding this model.

PRQ9.2: Define a tractable model with infinite inputs that embeds with in it possibilities of job splitting and job merging and displays how this affects the fairness. Pose feasible questions regarding this model.

1.6 *PRQ11*: Extended Abstract Regarding Near Optimal Control of Queueing Networks Over a Finite Time Horizon

We have recently prepared this extended abstract regarding Near Optimal Control of Queueing Networks Over a Finite Time Horizon. We have submitted this abstract to a conference that deals with communication networks

hence it is written in a manner consistent with communications applications. We will label work on this subject by *PRQ11*:

Extend Abstract: Communication networks are often modeled by multi-class queueing networks: Messages are classified into several classes (e.g. according to source and destination) and messages of each class queue up in a buffer. Processing nodes with finite processing capacity are used to process the queues of messages, where each node has a constituency of classes which it will process. Decisions involve the processing capacities allocated at each time by the nodes to the processing of the buffers, and routing of messages between the buffers. In many situations it is desired to determine the optimal control of a communications network for a given time window: In that case one starts from the state of the system at the beginning of the time window, and one looks for a control which will react to the input of the system over the time window, and will optimize both the cost of processing over time and the terminal state at the end of the time window. Such a problem is called control of a transient system over a finite time horizon.

We suggest a novel approach to such transient finite horizon control problems: (1) Approximate the system by a fluid system which is continuous and deterministic, in contrast to the original discrete and stochastic communications network. (2) Calculate the optimal control of the fluid system, and obtain the fluid buffer levels and processing rates. (3) Track the fluid solution with the real system, using the states of the fluid and real systems to determine the actions. We base our approach on the algorithm of Weiss [29] for separated continuous linear programs to solve the fluid problem, and on an adaptation of the maximum pressure policy of Dai and Lin [11] to track the fluid solution. In theory this method can be shown to be asymptotically optimal under appropriate fluid scaling assumptions.

Our main purpose in this paper is to illustrate this approach through a simple example: We consider a multi-class queueing network with two processing nodes and three buffers. Messages move from buffer 1 to buffer 2 (processing by node A), then from buffer 2 to buffer 3 (processing by node B), and finally from buffer 3 and out (processing again by node A). In the communications context, node A can be thought of as a half-duplex communication link in which each message travels first in one direction and then in the opposite direction, and each message is also processed by node B in between its transmission in both directions.

We assume that initial queues of messages in the three buffers are given, $Q_i(0)$, and denote by $Q_i(t) \geq 0,\ i = 1, 2, 3$ the queue at time t with time horizon $t \in [0, T]$. Messages are processed singly, with no preemptions, where node A processes messages from buffers 1 and 3, and node B processes messages from buffer 2. Processing times are all random and independent, with average processing times $1/\mu_i$ for

messages in buffer i. Assuming work conserving processing, the following scheduling decision is to be made: Whenever node A is available, and the queues in buffers 1 and 3 are not empty, decide whether to start processing an item from buffer 1 or start processing an item from buffer 3. The finite horizon control problem is to choose these decisions so as to minimize the expected value of $\int_0^T \sum_{k=1}^3 Q_i(t)dt$. This objective corresponds to minimizing inventory costs, equivalently minimizing the sum of all the waiting times over the time horizon, equivalently maximizing the total sum of times from completion of messages to the time horizon. For simplicity we assume no external inputs in $[0, T]$.

The corresponding fluid system for this problem consists of the fluid buffers with $q_i(t)$ being the amount of fluid in buffer i at time t, initially $q_i(0) = Q_i(0)$. Processing is continuous and allocation of a fraction $a_i(t)$ of the processor to buffer i at time t results in outflow of rate $u_i(t) = \mu_i a_i(t)$. Minimization of $\int_0^T \sum_{k=1}^3 q_i(t)dt$ is a separated continuous linear programming problem [29]. Its solution partitions the time horizon into $0 = t_0 < t_1 < \cdots < t_N = T$, with constant flow rates u_i^n in the nth interval, and with continuous piecewise linear buffer levels $q_i(t)$.

For each time interval, we partition the buffers into two sets: during time interval n, K_0^n is the set of buffers that are empty of fluid, and K_∞^n are buffers which have positive amount of fluid. These sets are well defined. Note that a fluid buffer can be empty and still have a positive outflow (equal to the inflow).

For the purpose of control we compare the fluid solution $q_i(t), u_i(t)$ with the actual system, $Q_i(t), D_i(t)$ where $D_i(t)$ is the departure counting process. For $i \in K_0^n$ we use $Q_i(t)$, the non-negative queue length of the real system at time $t_{n-1} < t < t_n$. For $i \in K_\infty^n$ we use $R_i(t) = u_i^n(t - t_{n-1}) - (D_i(t) - D_i(t_{n-1}))$, the backlog of actual departures compared to the nominal optimal fluid output rate. We let $Z_i(t),\ t_{n-1} < t < t_n$ be equal to $Q_i(t)$ for $i \in K_0^n$, and to $R_i(t)$ for $i \in K_\infty^n$. The process $Z_i(t)$ is the state used for the control of the system, where our purpose is to keep $Z_i(t)$ close to 0 so that the actual system will track the fluid solution.

Dai and Lin [11] have recently introduced the max pressure policy, for the control of stochastic processing networks, and in particular for the control of multi-class queueing networks. A MCQN with traffic intensity ≤ 1 will remain pathwise stable under the maximum pressure policy. Our approach adapts the maximum pressure policy so that we control the process $Z(t)$. Using our approach, the actual maximum pressure control in each interval depends both on the values of $Z(t)$ and on the sets of buffers K_0^n, K_∞^n. Under appropriate scaling $\int_0^T \sum_{k=1}^3 |Z_i(t)|dt \Big/ \int_0^T \sum_{k=1}^3 Q_i(t)dt$ approaches 0, and the control is asymptotically optimal.

Our approach adapts the maximum pressure policy (which keeps all queues stationary and minimal under long term homogeneous conditions) to a transient

situation, in which conditions are not homogeneous over the finite time horizon. We note that in those intervals in which the fluid is positive, the actual queue length will be strictly positive (at least in the interior of the interval, excluding short periods at the beginning and end of the interval) and thus R_i is the quantity that is tracked. As opposed to that, when the fluid is 0 it is expected that the queue will follow stable busy period - idle period cycles and thus Q_i is the quantity that is tracked. Our fluid tracking approach thus allows us to handle both very big queues (positive fluid amounts) and small stable queues (zero fluid amounts) using a single policy while maintaining the proper flow rates and without incorrect bias towards the big queues (as might result from more naive fluid tracking approaches).

For our two node and three buffer example we obtain simple rules for scheduling node A to the messages in queues 1 and 3. The asymptotic optimality indicates that as the initial number of messages becomes large and the processing rate is increased in proportion, the system performance should approach the optimal fluid solution. We perform extensive simulation studies to assess how well this works in practice.

Chapter 2

Planned Course of Action for Research

The aim of this short chapter is to lay down the intended work plan of the research. Section 2.1 states the general disciplines that will be used during the research and lays down the order and manner in which proposed research questions will be tackled. Section 2.2 states the planned layout of the dissertation. Section 2.3 outlines the several publications that may be submitted based on work that will be performed.

2.1 Work Plan

Our proposed work plan is comprised of the following nine tracks of activities: *general queueing background track*, *infinite inputs background track*, *stochastic optimization track*, *Lyapunov stability track*, *fluid stability track*, *diffusion approximations track*, *simulation track*, *problem solving track* and *publication track*. In the general queueing background track, we will continuously gain more broad knowledge with regards to known results and theory of queueing networks. In the infinite inputs background material track, we will master all previous results (mostly by Weiss et.al) with regards to infinite inputs systems. In the stochastic optimization, Lyapunov stability, fluid

stability and diffusion approximations tracks we will attempt to master these mathematical techniques. In the simulation track, we will develop and refine the simulation tools used in our previous work [24] and in section 1.4.1. In the problem solving track we will tackle the proposed research questions. Here we will make use of techniques acquired and background material that has been studied. Finally in the publication track we will prepare papers, presentations and dissertation chapters.

As our research progresses, we will work on all tracks in parallel. Within each track, our course of action will be mostly sequential. We will attempt to synchronize our advance in each of the tracks so that the products of each of the tracks collaborate. We now describe our intended course of action in each of these tracks separately.

General Queueing Background Track

In this track we will study general known results of queueing theory and queueing networks. Our goal is simply to gain general background knowledge in these fields. As we acquire more knowledge, certain specific bits of it will be summarized according to our point of view in the background material dissertation chapters. We would like to touch a variety of classical queueing systems results. These include some of the rudimentary results that appear in [36] and some of the more advanced results that appear in [3].

Infinite Inputs Background Track

In this track we will follow closely the known results regarding systems with infinite inputs. Many of our proposed research questions are extensions of these known results. Specifically we will study the following: [1], [2], [34], [30], [31], [24], [10] and [22].

Stochastic Optimization Track

Several of the proposed research questions deal with finding optimal scheduling policies. For this we require use of both classical and advanced optimization techniques. We plan to start off by gaining basic knowledge in this exciting field by studying [28]. We will then follow up on more recent advances with regards to bandit problems and issues regarding the existence of switching curves of certain problems.

Lyapunov Stability Track

Lyapunov function methods are a very popular and convenient way for showing that a Markov chain is positive recurrent. We will initially study several applications of this method according to [7] and other resources. We will then continue to the generalization of the method to general state spaces that is presented in [13].

Fluid Stability Track

In this track we plan to study recent advances with regards to applications of fluid models for proving stochastic stability. We will closely study [9] (a 1990's paper) and [11] (a 2005 paper). A proper understanding of this subject will require following some 15 to 20 related papers that have appeared during the last decade and a half. For general reference on this subject we will use [8].

Diffusion Approximations Track

Diffusion approximation methods have in many ways become the mainstream methods that are used in the study of queueing networks. We are thus almost obligated to relate to this subject in our work plan. In this track we will study the theory and mechanism used in diffusion approximations. Our primary sources for the basic theory will be [8] and [35]. We will test and sharpen

our understanding of this subject by attempting to perform an adaptation of the results of [12] to SPNII models. Proper understanding of this subject may also require following the theory in [5] and the ideas presented in [15]. We may also need to strengthen our knowledge of analysis and probability, for this will mostly use [14] and [27].

Simulation Track

Currently we have in hand the following simulation tools that we have designed and programmed: the *Job Shop Simulator*, the *3 Buffer Maximum Pressure Simulator* and the *4 Buffer Maximum Pressure Simulator*. The Job Shop Simulator was developed during our previous work [24]. It is possible that we will update this tool to handle the context of general SPNII models. The 3 Buffer Maximum Pressure Simulator is currently being used for work related to [33] and *PRQ11*. The 4 Buffer Maximum Pressure Simulator has been used for obtaining the results that are presented in section 1.4.1 (*PRQ7*). Both of these simulation tools are very specific to their problem instance and the maximum pressure policy. Given additional simulation related problems we will probably either write a new tools or use our modification of the Job Shop Simulator.

Another tool that we have been working on previously is the *Fluid Flow Shop Solver*. This tool essentially implements the algorithm presented in [32]. We may continue working on this tool when handling *PRQ6*.

Problem Solving Track and Publication Track

Most of our advance in the previously mentioned tracks will be driven and motivated by the problem solving track. As we handle the research questions presented in chapter 1, we will advance in parallel in one or more of the previously mentioned tracks.

We are currently working on *PRQ11*. Following the submission of the

proposal our plan is to initially handle *PRQ4* and *PRQ8*. These proposed research questions deal with stability of the 2R3BII and 2R4BII models respectively. Handling these research questions requires advances in the Lyapunov stability track and fluid stability track.

We next intend to handle *PRQ5*. This research question deals with stochastic optimization. Thus in parallel we will advance in the stochastic optimization track.

In parallel, we intend to work on *PRQ3* and *PRQ6* independently. *PRQ3* deals with more traditional queueing analysis of the 2R3BII model. This type of analysis will require some advances in the general background and infinite input background tracks. But this may be done out of order of any other activity. *PRQ6* deals with deterministic optimization and it too may be handled at any time.

PRQ1 and *PRQ2* deal with general RLINEII models. At the moment, we do not plan to explicitly handle these research questions but rather keep them in mind during our work.

PRQ0 is our ultimate question and we do not believe that it can be properly handled at this time. Related to it, is *PRQ9*, this is an abstract question asking to define further models. We will give thought to it while preparing the summary of our dissertation

As a consequence of this plan, our dissertation will include results regarding the following research question: *PRQ4*, *PRQ8*, *PRQ5 PRQ3* and *PRQ6*. As we conquer each of the research questions we will document our results as part of the publication track.

2.2 Planned Dissertation Chapters

These are the planned chapters:

1. **Overview and Main Results** - This chapter will summarize the main results along with a summary of known results regarding infinite input systems. This is the only chapter in the dissertation that is to contain background material except for the appendix. Our main results will be proofs of stability of systems (*PRQ4* and *PRQ8*), optimizing scheduling policies (*PRQ5*) and additional queueing analysis results regarding our simplest models (*PRQ3*). We will not post results regarding *PRQ6*, the fluid flow shop, in the dissertation. Our results will be presented in a manner that is meshed with previous results regarding infinite input systems.

2. **Stability of Certain Networks** - This chapter will contain the details of the work performed with respect to *PRQ4* and *PRQ8*.

3. **Optimization of Certain Networks** - This chapter will contain the details of the work performed with respect to *PRQ5*.

4. **Future Directions** - This chapter will introduce the unsolved *PRQ0* and state all that is known regarding this research question.

5. **Appendix: Queueing Networks Background** - This appendix will contain a unique compilation of most of the background material that we will study during the course of the research.

2.3 Planned Publications

This is a list of the publications that we may submit during the duration of our research. Note that this list also contains some additional items that are not planned to be included in the dissertation and haven't been mentioned anywhere in this research proposal:

- **Stability of a Simple Reentrant line under LBFS with General Distributions** - This publication will include the results of *PRQ4*.

- **Stability of a Push Pull Queueing System with General Distributions** - This publication will include the results of *PRQ8*.
- **An Optimizing Policy of a Simple Reentrant line** - This publication will include the results of *PRQ5*.
- **A Survey of Infinite Input Systems** - This publication will be based on the first chapter of the dissertation. It aggregates all of the known results regarding infinite input systems.
- **Optimizing a Fluid Flow Shop** - This publication was prepared by Weiss but is not ready for publication at this time. We will complement it with implementations of the algorithms and numerical results as explained in *PRQ6*.
- **Fluid Tracking of SCLP Solutions Using Maximum Pressure** - This publication was prepared by Weiss but is not ready for publication at this time. We have performed simulation studies regarding this publication and will continue this work. A preliminary publication on this subject might be released due to current work on *PRQ11*.
- **A Push-Pull Queueing System** - This publications was prepared entirely by Weiss and Kopzon and we are only supporting it with simulations of the 2R4BII problem.
- **Asymptomatically optimal Job-Shop Scheduling Heuristics** - This publication will be based on our previous work [24].
- **Optimizing the Calender of the Supreme Court** - This publication will be based on near future field work.
- **All Sink All Drain Information Passing in Sensor Networks** - We have spent some time planning ideas for this publication and we may pursue it in the future.

- **Optimizing a Mobile Wireless Network with Known Time Varying Rates Over a Finite Time Horizon** - We have spent some time planning ideas for this publication and we may pursue it in the future.

Chapter 3

Background Material to be Studied and Summarized

This chapter surveys some of the background material that is to be studied during the course of our research. In some sections, background material is summarized. In others, we merely list the references and general principles that are to be studied. This chapter is also designed to serve as a skeleton for the background material dissertation chapters and some of our proposed publications.

We begin with section 3.1 where we overview the field of queueing networks. Starting of with a summary of the interesting rudimentary results and applications of queueing theory. Then continuing to an evolutionary summary of queueing networks from the most basic networks to the frontier of research today. We then devote our attention to heavy traffic and fluid models. We finish the section with a summary of known results regarding reentrant line models.

In section 3.2 we briefly touch the probalistic and mathematical tools that we will study, summarize and utilize during our research.

In section 3.3 we summarize all known results regarding models with infinite inputs. This section is directly relevant to the subject of our research and

may be read in conjunction with our introduction of the research questions in chapter 1.

3.1 Overview of Queueing Networks

This section is written as a skeleton for background chapter (or appendix) that will be presented in the dissertation. Each of the following sub-sections is to be expanded to a large section.

3.1.1 Elements of Queueing Theory

There are many elements of queueing theory that are to be covered. We will obviously not be able to cover them all but there are several principal issues which we find interesting. In this regard, we intend to study and summarize the following subjects: (1) Reversibility of Markovian systems and their applications (from [19]). (2) Results regarding the M/G/1 and G/M/1 queues and their relations (from [36] and [20]). (3) Priority queues (from [36] and [21]).

3.1.2 Queueing Networks: From Jackson Networks to Multi-Class Queueing Networks

We intend to introduce the subject of queueing networks within the Markovian setting: open and closed Jackson networks. We will mainly follow the lines of [8] for this purpose but in addition we will look for examples in [19] and study some more rigorous Markovian theory in [3].

We then intend to expand to the description of Multi-Class Queueing Networks and summarize recent results regarding these models. We intend to start to summarize this subject by following [4]. This is because [4] deals with Lyapunov functions (which are of interest to us) and does not focus on diffusion approximations (which we plan to deal with else where).

3.1.3 Heavy Traffic and Diffusion Approximations

A single server queue with arrival rate λ and service rate μ is roughly said to operate under a heavy traffic regime if $\lambda \approx \mu$. For a more general queueing network, heavy traffic has been defined in several ways. In [16], Harrison proposes to analyze the "bottleneck subnetwork" of the queueing network under the assumption that all resources of the subnetwork are heavily loaded. Harrison's framework in [16] makes use of the mathematical foundations previously exemplified by Reiman in [26]. This framework was later expanded by many others, the notable papers being [18] by Kelly and Laws, [17] by Harrison and Van Mieghem, yet there are many others.

The motivation for analysis of queueing networks under heavy traffic stems both from the tractability of these models using variants of the functional central limit theorem (FCLT) and from the fact that the heavy traffic assumption is argued to be a valid one in modeling.

We intend to study the backbone of this research (the notable papers) and summarize the evolution of this subject. For reference we will use [35] and mostly [8]. Understanding this line of results is fundamental to our research because in many situations we also propose models that operate under heavy traffic due to their infinite input nature.

3.1.4 Fluid Models

The fluid view of a stochastic system makes usually uses some form of the functional strong law of large numbers (FSLLN). Regarding this subject, we intend to study and summarize [9] and [12]. We will use [8] for reference.

3.1.5 Reentrant Line Models

Reentrant line models have been analyzed frequently due to their apparat applicability to silicon wafer manufacturing plants. These are some notable

papers regarding this subject that are interesting for our research: [23], [10] and [37]. We will summarize these and several others.

3.2 Probabilistic Tools of Stability Analysis

We intend to closely follow [13] by Foss and Konstantopoulos. This paper is an exposition of stochastic stability methods. It is also the only known reference to us that expands the Lyapunov function method to general polish spaces. In the description of the content of [13] we will initially fully summarize and give original examples of applications of the Foster-Lyapunov theorem on countable Markov chains. For this we intend to primarily follow [7]. We will later explore the relation between the Lyapunov function methods and fluid stability methods.

3.3 Known Results Regarding Models with Infinite Input

This section summarizes the references to all the known results regarding models with infinite inputs. All of the known results regarding models of this nature have been attained by Weiss et. al. Surprisingly this class of models has not been investigated previously.

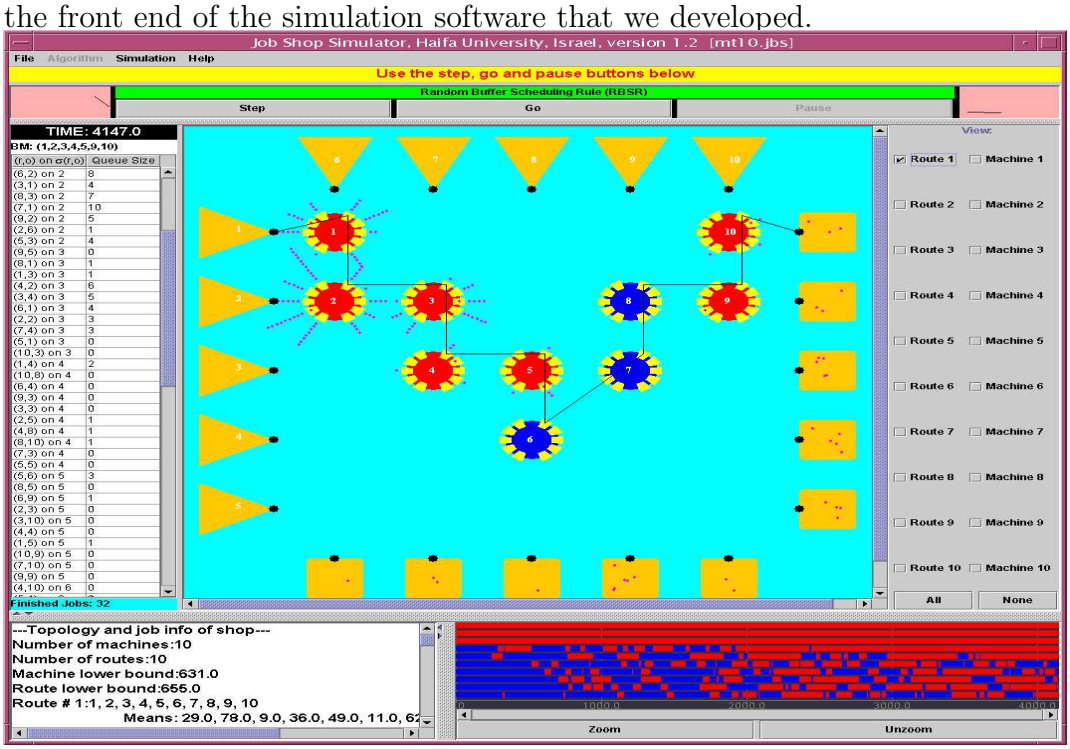
We briefly describe the nature of each of the results and will summarize them in an expansive fashion in our dissertation along with our planned contributions.

3.3.1 Simulation Results of High Volume Job Shop Problems

In [24] we conducted a simulation study of job shop scheduling problems (see [25] for an introduction to this subject) in which there are many jobs on each

route and the processing times of all steps on each route are i.i.d. random variables. This study, was motivated by the results of Dai and Weiss in [10] and Boudoukh, Penn and Weiss in [6]. The previous results had shown that under several assumptions regarding the processing time distributions the job shop can be scheduled such that the idle time of the bottleneck machine is constant in the number of jobs, thus meaning that the schedule is near optimal when the number of jobs is large. In our study we had relaxed some of these assumption and we were still able to perform the efficient scheduling.

Results from our simulation study for finite job shops are also applicable for infinite horizon models with infinite inputs (as was described in [24]). This is because during the initial operation of a high volume jobs shop (with N jobs) waiting to be processed, the job shop acts like a model with infinite inputs. Thus the results in [24] have led to some of the research questions presented in this proposal. The following picture is a screen shot taken of the front end of the simulation software that we developed.



3.3.2 Infinite Input Jackson Networks

In [31], Weiss introduced Jackson networks with an unlimited supply of work. These are the standard Jackson networks with the following modification: Some of the nodes have an infinite supply of work. These nodes give priority to customers that are queued, but when the queue of the node is empty, it processes jobs from the infinite supply and routes them in accordance with the normal probabilistic routing matrix $\mathbf{P}$. It is assumed that when an internal node that is processing a job from the "infinite supply queue" gets a "real job", it preempts the infinite supply job.

Let E to be set the of nodes with an infinite supply of work. Let λ_i be the rate at which items arrive into node i (counting both exogenous input and routing from other nodes). Let μ_i be the processing rate of jobs at node i. For $i \notin E$, let α_i be the rate at which external jobs arrive to node i (for $i \in E$ this rate is infinite). Then at equilibrium, for $i \in E$ we get:

$$\lambda_i = \alpha_i + \sum_{\substack{j \notin E \\ j \neq i}} \lambda_j P_{ji} + \sum_{\substack{j \in E \\ j \neq i}} \mu_j P_{ji}$$

Based on these equations it is shown that the joint steady state distribution of the queues of the nodes $i \notin E$ is the standard Jackson style product form distribution. It is also shown that the marginal steady state distribution of the queues of the nodes $i \in E$ is geometric.

In [2], Adan and Weiss investigate a special case where there are two nodes, both having an infinite supply of work. Here they find the joint steady state distribution of the system.

3.3.3 The 2R3BII Model

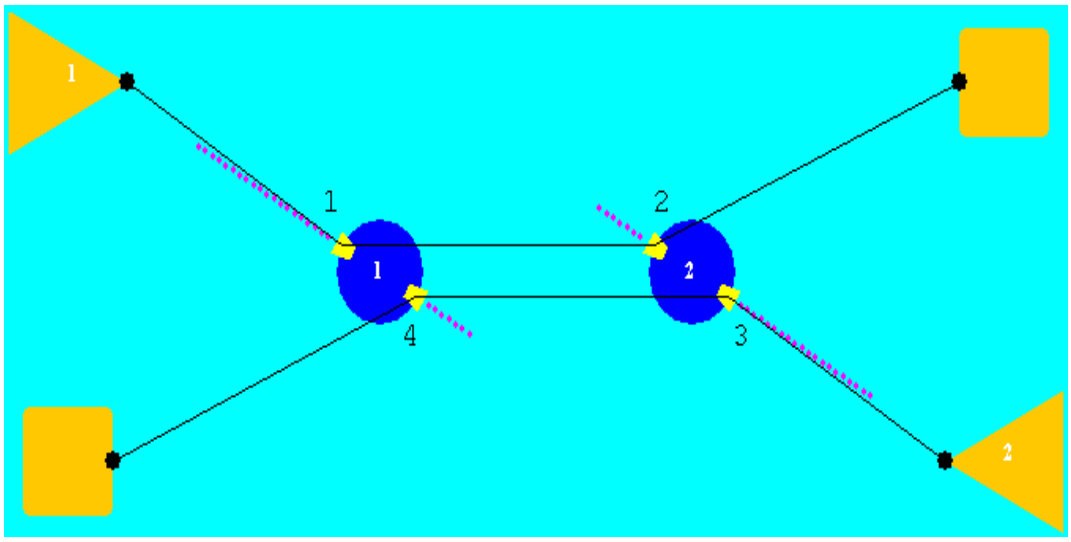
In [30] Weiss investigates stability of the 2R3BII model. This is a reentrant line with infinite inputs having 3 steps on two resources such that the first and third steps are on resource 1 and the second step is on resource 2. Exponential

processing times are assumed so the state space of the model is the grid on the positive quadrant. Weiss proves that the resulting Markov chain is positive - recurrent when the LBFS policy is used (priority to buffer 3) and when resource 1 is the bottleneck. The proof uses both Lyapunov function methods (on countable state spaces) and coupling arguments. This work is continued in [1] where the steady state distribution is calculated along with other interesting sample path properties. The case in which resource 2 is the bottleneck is also interesting and is promptly described in [30].

3.3.4 The 2R4BII Model

In [34] and [22] Weiss and Kopzon analyze the 2R4BII model. They call this model "push-pull". This model has two routes, each with two buffers (we use the term route in the context of SPNII models when there is no job splitting or merging). Route 1 is composed of buffers (activities) 11 and 12 which are powered by resource 1 and resource 2 respectively. Route 2 is composed of buffers/activities 21 and 22 which are powered by resource 2 and resource 1 respectively. Buffers 11 and 21 are source buffers. Buffers 12 and 22 are intermediate buffers. Weiss and Kopzon do not explicitly define the destination buffers but rather indicate that jobs exit the system after being processed at buffer 12 and 22. The processing rates of the source buffers are labeled λ_1 and λ_2 respectively. The processing rates of the destination buffers are labeled respectively with μ's.

The picture below is a screen shot of this model, from the front end of the simulation software developed in [24]. In the picture the buffers of route 1 are labeled 1 and 2 and the buffers of route 2 are labeled 3 and 4. Also note that the picture shows a finite amount of jobs on the source buffers, while in the 2R4BII model this should be an infinite amount.



Note that the notation for the rate parameters is such that if the resources are fully allocated to route i then buffer $i2$ operates like a GI/GI/1 queue with input rate λ_i and service rate μ_i. In [34], the authors handle the "inherently stable case" in which $\lambda_i < \mu_i$ and in [22], the authors handle the "inherently unstable case" in which $\mu_i < \lambda_i$. In both cases, the authors find a scheduling policy that is fully utilizing (resources are always being used) and stable. The stability is in the sense of positive recurrence of the corresponding Markov chain when the processing times are independent exponentials. These results constitute a surprising positive result that exemplifies a model in which stability is maintained when there is full utilization.

The Pull Priority Policy in the Inherently Stable Case

In the in the inherently stable case, the authors analyze a buffer priority policy that gives priority to the intermediate buffers (12 and 22) over the source buffers (11 and 21). If preemption is allowed the analysis of the system is quite simple: looking at the recurrent states in the system yields that the

system acts like two M/M/1 queues such that when one of the queues is non-empty, the other is empty and when both are empty the system can start a busy period at one of the two M/M/1 queues. Appropriate probabilities for this are determined by exponential races.

Continuing in the inherently stable case, things are a bit more complicated when pre-emption is not allowed. For this case, balance equations are formulated and the steady state distribution is solved by using generating functions. The authors then expand this case to the situation in which the service times of the intermediate buffers are from a general distribution. Here the M/G/1 model with vacations is employed (see [36]) to solve for the steady state distribution.

Generalized Threshold Policies in the Inherently Unstable Case

The pull priority policy is not stable for the inherently unstable case. The question arises if there exist other policies that can stabilize the system while maintaining full utilization of the resources.

The authors have offered a class of policies named *generalized threshold policies* that have this attribute. Vaguely, these policies define thresholds for the buffers 12 and 22. The pushing activity (intermediate buffer) is activated only when the number of jobs in the buffers is above a given threshold. The policy is called "generalized" because the threshold for each buffer is allowed to be an increasing function of the number of jobs in the buffer.

The authors show that the resulting Markov chain (under generalized threshold policies) is positive recurrent. They use Lyapunov function methods for this. In addition, steady state probabilities for fixed thresholds and increasing thresholds at a constant rate are calculated.

Production Rates

At the heart of the analysis (in both the inherently stable case and inherently unstable case) is a calculation of the rates of production on each route. If destination buffers were assumed, this would be the rate of growth of each destination buffers. For this, Weiss and Kopzon do the following: assume that resource i allocates a fraction α_i of the time for working on it's intermediate buffer (pulling jobs out of the system) and a fraction $1 - \alpha_i$ for working on it's source buffer (pushing jobs into the system). Now denote the rate of production on route i by ν_i. Stability will now require:

$$\nu_1 = (1 - \alpha_1)\lambda_1 = \alpha_2\mu_2$$
$$\nu_2 = (1 - \alpha_2)\lambda_2 = \alpha_1\mu_1$$

Now solving for α_1 and α_2:

$$\alpha_1 = \frac{\lambda_2(\mu_2 - \lambda_1)}{\mu_1\mu 2 - \lambda_1\lambda_2}$$
$$\alpha_2 = \frac{\lambda_1(\mu_1 - \lambda_2)}{\mu_1\mu 2 - \lambda_1\lambda_2}$$

Thus:

$$\nu_1 = \frac{\mu_2\lambda_2(\mu_2 - \lambda_1)}{\mu_1\mu 2 - \lambda_1\lambda_2}$$
$$\nu_2 = \frac{\mu_1\lambda_1(\mu_1 - \lambda_2)}{\mu_1\mu 2 - \lambda_1\lambda_2}$$

It is thus evident that in the 2R4BII model, for given processing rates, there is a single attainable rate of production (ν_1, ν_2) when full utilization and stability is assumed.

Bibliography

[1] Ivo Adan and Gideon Weiss. Analysis of a simple markovian re-entrant line with infinite supply of work under the lbfs policy. 2004.

[2] Ivo Adan and Gideon Weiss. A two node jackson network with infinite supply of work. 2004.

[3] Soren Asmussen. *Applied Probability and Queues*. 2003.

[4] Dimitris Bertsimas, David Gamarnik, and Tsitsiklis John N. Performance of multiclass markovian queueing networks via piecewise linear lyapunov functions. 2000.

[5] Patrick Billingsely. *Convergence of Probability Measures*. 1999.

[6] Tami Boudoukh, Michal Penn, and Gideon Weiss. Scheduling jobshops with some identical or similar jobs. 2001.

[7] Pierre Brémaud. *Markov Chains Gibbs Fields, Monte Carlo Simulation and Queues*. 2003.

[8] Hong Chen and David D. Yao. *Fundamentals of Queueing Networks, Performance, Asymptotics and Optimization*. 2003.

[9] J. G. Dai. On positive harris recurrence of multiclass queueing networks: A unified approach via fluid limit models. 1995.

[10] J. G. Dai and Weiss Gideon. A fluid heuristic for minimizing makespan in job-shops. 2001.

[11] J. G. Dai and Wuqin Lin. Maximum pressure policies in stochastic processing networks. *Operations Research*, 53(2), 2005.

[12] J. G. Dai and Wuqin Lin. Asymptotic optimality of maximum pressure policies in stochastic processing networks. 2006.

[13] Serguei Foss and Takis Konstantopoulos. An overview of some stochastic stability methods. 2004.

[14] Avner Friedman. *Foundations of Modern Analysis*. 1970.

[15] Michael J. Harrison. *Brownian Motion and Stochastic Flow Systems*. 1985.

[16] Michael J. Harrison. Brownian models of queueing networks with heterogenous customer populations. 1988.

[17] Michael J. Harrison and J.A Van Mieghem. Dynamic control of brownian networks: state space collapse and equivelent workload formulations. 1997.

[18] F.P. Kelly and C.N. Laws. Dynamic routing in open queueing networks. 1993.

[19] Frank Kelly. *Reversibility and Stochastic Networks*. 1979.

[20] Leonard Kleinrock. *Queueing Systems, Volume I: Theory*. 1975.

[21] Leonard Kleinrock. *Queueing Systems, Volume II: Computer Applications*. 1976.

[22] Anat Kopzon. *The Push-Pull system: A queueing network with two machines that feed each other*. PhD thesis, 2006.

[23] P. R. Kumar. Re-entrant lines. 1993.

[24] Yoni Nazarathy. Evaluation of on-line scheduling rules for hgih volume job shop problems, a simulation study. Master's thesis, 2001.

[25] Michael Pinedo. *Scheduling: Theory, Algorithms and Systems*. 1995.

[26] M.I. Reiman. Open queueing networks in heavy traffic. 1984.

[27] Sidney I. Resnick. *A Probability Path*. 1999.

[28] Sheldon M. Ross. *Introduction to Stochastic Dynamic Programming*. 1983.

[29] Gideon Weiss. A simplex based algorithm to solve separated continuous linear programs. *Preprint*, 2001.

[30] Gideon Weiss. Stability of a simple re-entrant line with infinite supply of work the case of exponential processing times. 2004.

[31] Gideon Weiss. Jackson networks with unlimited supply of work. 2005.

[32] Gideon Weiss. Optimal control of a fluid flowshop. *Preprint*, 2005.

[33] Gideon Weiss. Finite horizon control of processing networks via fluid approach: Separated continuous linear programs, infinite virtual buffers and maximum pressure policies. *Preprint*, 2006.

[34] Gideon Weiss and Anat Kopzon. A push pull queueing system. 2001.

[35] Ward Whitt. *Stochastic-Process Limits, An Introduction to Stochastic-Process Limits ans Their Applications to Queues*. 2001.

[36] Ronald W. Wolff. *Stochastic Modeling and the Theory of Queues*. 1989.

[37] Jiankui Yang, J. G. Dai, Jian-Gong You, and Hanqin Zhang. A simple proof of diffusion approximations for lbfs re-entrant lines. 2005.