

## 総合的ネットワークフローモデルにおける数学的解析とその応用

メタデータ	言語: ja 出版者: 静岡大学 公開日: 2012-01-17 キーワード (Ja): キーワード (En): 作成者: 水野, 信也 メールアドレス: 所属:
URL	<a href="https://doi.org/10.14945/00006382">https://doi.org/10.14945/00006382</a>

電子科学研究科:

GD

K

0008505620 R

538

静岡大学附属図書館

静岡大学 博士論文  
総合的ネットワークフローモデルにおける数学的  
解析とその応用



2007年12月  
大学院 理工学研究科  
システム科学専攻  
水野 信也

# 目次

<b>1</b>	<b>序論</b>	<b>4</b>
1.1	ネットワークモデル	4
1.2	研究の概要と本論文の構成	5
<b>2</b>	<b>待ち行列理論の基礎</b>	<b>6</b>
2.1	待ち行列研究の変遷	6
2.2	ポアソン過程	6
2.3	M/M/1 型待ち行列モデル	7
2.4	M/M/S 型待ち行列モデル	8
2.5	ジャクソンネットワーク	9
2.6	サイクル公式	10
<b>3</b>	<b>ネットワーク型優先権付き待ち行列</b>	<b>13</b>
3.1	はじめに	14
3.2	離散マルコフ連鎖での表現	15
3.3	埋め込み時刻における定常分布の計算	16
3.4	定常分布の計算	23
3.5	数値計算例	25
3.6	まとめ	26
<b>4</b>	<b>サプライチェーン・ネットワークにおける基本情報の算出</b>	<b>27</b>
4.1	サプライチェーン・マネジメントの概要	27
4.2	サプライチェーン・マネジメントの問題点	28
4.3	サプライチェーン・ネットワークの構築	28
4.4	サプライチェーン・ネットワークの定義及び計算	31
4.4.1	変数の定義	31
4.4.2	平衡方程式	31
4.4.3	行列化	32
4.4.4	解の算出	34
4.4.5	基本情報の算出	34
4.5	数値計算例	36
4.6	まとめ	38
<b>5</b>	<b>輸送ネットワークと安全在庫配置の同時改善解の算出とシステム構築</b>	<b>39</b>
5.1	サプライチェーン・ネットワーク改善システム構築の意義	39
5.1.1	サプライチェーン・ネットワーク改善問題	40
5.2	SCN 改善システムの構築	41
5.3	GA を用いた SCN 改善ツールの開発	42
5.3.1	遺伝アルゴリズムの構成と流れ	42
5.3.2	GA を適用した SCN 改善システムの紹介	43

5.3.3	GA での SCN 改善計算例 . . . . .	46
5.3.4	まとめ . . . . .	50
5.4	主双対内点法の適用した SCN 改善システムの開発 . . . . .	50
5.4.1	主双対内点法の概要 . . . . .	50
5.4.2	サプライチェーン・ネットワークへ主双対内点法の適用と SCN 改善システムの構築 . . . . .	52
5.4.3	主双対内点法での SCN 改善計算 . . . . .	54
5.4.4	まとめ . . . . .	55
<b>6</b>	<b>総括と今後の課題</b>	<b>56</b>
6.1	結果と今後の課題 . . . . .	56
6.2	最後に . . . . .	57
	<b>参考文献</b>	<b>58</b>
	<b>付録</b>	<b>62</b>
.1	NUOPT での主双対内点法計算 . . . . .	62
.2	サプライチェーン・ネットワークにおける配送率 . . . . .	62
.3	SCN 改善システムのパラメータ . . . . .	66
	<b>謝辞</b>	<b>74</b>

# 1 序論

## 1.1 ネットワークモデル

世界には、様々なネットワークが存在する。ネットワークモデルとは2つ以上のノードが結びつき様々な影響を受けあいながら形成されている。コンピュータ、交通、物流などいたるところでネットワークが存在する。これまで、実データに基づいてはいるものの、小規模で簡単なネットワークが研究の対象とされるケースが多かった。しかしながら、近年のコンピュータの発展とネットワークの実データへの関心が高まり、大きなネットワークが様々な方法で解析されるようになり、現実にあるネットワークへの理解が急速に高まっている。そのひとつに、待ち行列ネットワークモデルがある。

待ち行列モデルは、20世紀初頭に電話交換機の性能評価モデルとして研究が始まり、その後、生産管理、道路設計、都市計画等様々な分野で応用された。特に最近では、情報通信ネットワークの性能評価モデルを対象に待ち行列モデルの研究が進められている。このような応用分野の拡大とともに、研究分野も単純な単一の待ち行列モデルから、複雑なネットワークモデルへと変わってきている。システム性能評価における待ち行列ネットワークモデルとは、ジョブがCPUや入出力装置などのシステム資源を利用する状況を、客(ジョブ)が窓口(システム資源)の前に行列をつくる状況として考える確率モデルのことである。さらに、待ち行列ネットワークモデル特徴として、複数の窓口や行列がネットワークとして結びついていることがあげられる。実際の現象を待ち行列理論によってモデル化し解析することはモデルを分析することに有用な情報を得ることができる。

また工場、配送センター、顧客が結びついたサプライチェーン・ネットワークがある。現在のサプライチェーン・ネットワークはグローバル化により全体の把握が難しい。ブルウィップ効果のように情報の共有化が不完全なサプライチェーン・ネットワークでは多くのロスが生じる可能性がある。販売機会損失を避けるために、小売店では品切れを防止するため実需を大きく上回って発注してしまい、多段階流通経路中の配送センターではまた販売機会損失を引き起こし、小売店に迷惑をかけてはいけないということで、メーカーには余裕を見て、多く発注しようということになる。工場では納期遅れを起こさないように必要以上に生産してしまう。この原因のひとつに多段階流通経路における情報の共有ができていないことがあげられる。そこで生産進捗情報の共有、メーカー在庫情報の共有、流通在庫情報の共有をしてロスの少ないサプライチェーンを構築する必要がある [47][48]。プロダクトサイクルの短命化や短期間での顧客の購買行動の変化によって市場の不確実性が高まり、変化に従ったサプライチェーン・ネットワークの素早い構築が求められる。

ネットワークモデルを解析するにはまずネットワークの各ノードまたはノード間の情報が不可欠である。ノード間の関係が明らかになれば、ネットワーク全体の設計が可能となる。また、ノード間情報を用いてネットワークの最適要素の情報を算出することも可能になる。現在、通信技術やコンピュータ性能が高まり、あらゆる分野でグローバル化が進みネットワークの改善が不可欠である。本論文では、モデル設計から基本情報を得た後、その情報を用いてネットワークを改善する方法を提案する。

## 1.2 研究の概要と本論文の構成

本研究の目的は現存する様々なネットワークモデルに対して、数学的手法でモデル化し、そのモデルに必要な基本情報を算出する。さらにその基本情報を利用してモデルの改善を行い、またそのツールを開発することである。

本研究では、ネットワークモデルの中で特に物流関連のモデルに注目し2種類のモデルを扱う。ひとつはネットワーク型優先権付き待ち行列のモデル化である。これはマルコフ連鎖またはマルコフ過程を用いて記述されるが、抽象化された一般的なモデルに分類される。このように一般のモデルを考えるのは、応用に柔軟に対応できるためだけでなく、何が普遍的な性質で、何がモデル特有の性質であるかを明確にするためである。本論文では2章で待ち行列理論の基礎を述べる。3章でネットワーク型優先権付き待ち行列のモデル化を行い、定常分布を算出するまでの過程を述べる [37]。またその数値計算を与える。このモデルは1本のメインラインに合流するとき優先度と余裕空間条件を与えている。これは工場でのピッキング処理に応用が期待できる。本論文後半ではサプライチェーン・ネットワークの解析を行う。従来の物流合理化は様々な問題に対して個別に対応してきたが、サプライチェーンの国際化と製品ライフサイクルの短期化などから、サプライチェーン全体の設計と運用が求められている。4章でサプライチェーン・ネットワークの構築方法を紹介し、モデル化を行う [43][44]。それより各拠点での在庫量や発注量などの在庫運営に必要な基本情報を算出する。またその数値計算を与える。5章では4章で求めた基本情報を用いて、サプライチェーン・ネットワークの改善を行う。まず遺伝アルゴリズムを用いたネットワーク改善を行う。次に主双対内点法を適用したネットワーク改善を行う。またそれぞれの計算結果を与える。

一般的にネットワークモデルは複数のノードが存在し、ネットワークは拡大が可能である。柔軟な記述が可能である一方、解析は様々な要素から困難になる傾向がある。本研究で扱うモデルは数学的処理をする上で同じ考えに基づいている。ネットワークモデルを扱う上でノード数や階層などのレベルは可変になり、これが解析を困難にしている原因のひとつである。これに対応できるよう待ち行列モデルでは直前のレベルだけに注目すればよい条件を設け、それを一般化する。サプライチェーンモデルには複数のノードの平衡方程式を行列化し、一括処理を可能にしている。

## 2 待ち行列理論の基礎

この章では一般的に知られている待ち行列理論を述べる。解法等詳しいことは [3] を参照されたい。

### 2.1 待ち行列研究の変遷

待ち行列の研究は、デンマークの電話会社技師 A.K.Erlang による 1907 – 1920 年の研究に始まるとされる。Erlang の研究以来、数学が実際の問題に役立つ面白い研究分野ということで、待ち行列理論は多くの著名な研究者の関心を集め、特に 1950 年代にはその後の待ち行列理論において重要な役割を果たす基本的概念が多数出されている。1975 年前後、計算機の飛躍的な発達に刺激されて、計算機や通信システムに対する新しいモデルの提案やモデルの整備、待ち行列ネットワークへの関心の高まり、アルゴリズム的方法論の台頭など、その成果が発表された。

待ち行列の解析というと従来は数学的な解析がほとんどであり、待ち時間分布のラプラス変換や積分方程式を求めることが目標であった。しかし、待ち行列ネットワークの研究は応用分野からの要請が研究を引っ張った。コンピュータなどへの応用を強く意図した [31] に見られるように、コンピュータシステムの性能評価が待ち行列の問題としてとらえられ、待ち行列ネットワークの利用が試みられ始めた。しかしサービス時間分布が指数分布に限られる Jackson モデル [19] では適用範囲が狭すぎるという不満があった。これをかなりの部分で解決したのが BCMP 型待ち行列ネットワーク [2] である。積形式解を持つという性質を保ち、客のクラスという概念が導入され、クラスによってネットワーク内の異なるルートを指定することが可能となった。さらにサービス分布についても、先着順サービスでは依然として指数分布に限られるものの、割り込み後着順 (LCFS-PR) やプロセッサシェアリング (PS) などのサービス規律のときはクラスごとに異なる相型分布が指定できるようになった。このような理論を組み込んだソフトウェアが作成され、製品導入の際には事前に性能評価を行うようになった。

上記のような発展の中で、待ち行列研究は幅を拡大してきた。a) 新しい応用分野、とりわけ情報通信分野、からの要請による新しいモデル構築と理論体系の整備、b) モデルの性能評価尺度を数値的に求めるアルゴリズム的方法の研究、c) ある範囲のモデルに対して一般的に成立する理論の研究、に分別され、各研究分野はより深化していった。

以下では、待ち行列理論の基礎を述べる。

### 2.2 ポアソン過程

電話がかかってくる”, “メールが届く”, “ファックスが届く”, “お店に客が来る” などの事象は、ランダムに起きる。ランダムに事象が起きる確率過程モデルとしてポアソン過程がある。ポアソン過程は、次の条件に従って事象が発生するモデルである。

1. (独立性) 事象が起きるのはたがいに独立である。
2. (定常性) 事象が起きる確率はどの時間帯でも同じである。

3. (希少性) 微小時間  $\Delta t$  の間にその事象が 2 回以上起きる確率は 0 である.

また, 客の到着人数は率  $\lambda$  のポアソン分布に従い, 客の到着間隔は平均  $1/\lambda$  の指数分布に従う. 時刻 0 から時刻  $t$  までに到着する客の人数が  $n$  である確率  $pi(n)$ , 及び時刻 0 に客が来たものとして時刻  $t$  に次の客が到着する確率  $pe(t)$  は次に与えられる.

$$pi(n) = \frac{(\lambda t)^n e^{-\lambda t}}{n!} \quad (1)$$

$$pe(t) = 1 - e^{-\lambda t} \quad (2)$$

### 2.3 M/M/1 型待ち行列モデル

M/M/1 型の待ち行列とは, 窓口の数が 1 つ, 客の到着がポアソン過程である. つまり到着人数は独立で同一の到着率  $\lambda$  のポアソン分布, 到着間隔は平均  $1/\lambda$  の指数分布に従う. 窓口におけるサービス時間も独立で同一の指数分布に従う待ち行列モデルである. 窓口の単位時間あたりのサービス率を  $\mu$  とすると, 平均のサービス時間は  $1/\mu$  となる.

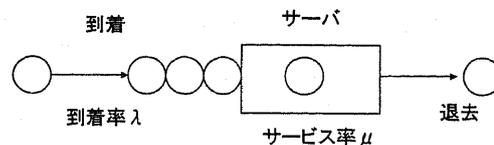


図 1: M/M/1 待ち行列モデル

(3) であらわされる  $\rho$  を占有率という.

$$\rho = \frac{\lambda}{\mu} \quad (3)$$

また, システムが安定するための条件として  $\rho < 1$  である. 時刻  $t$  にシステム内に  $n$  人いる確率を  $p_n(t)$  とすると, M/M/1 モデルの平衡方程式は次のようになる.

$$(\lambda + \mu)p_n = \lambda p_{n-1}(t) + \mu p_{n+1} \quad (n \geq 1) \quad (4)$$

$$\lambda p_0 - \mu p_1 = 0 \quad (5)$$

これより定常分布  $p_n(t)$  が求まる.

$$p_n = (1 - \rho)\rho^n \quad (6)$$

また (6) より平均系内人数  $L$  及び平均待ち時間  $L_q$  が求まる.

$$L = \sum_{n=0}^{\infty} np_n = \frac{\rho}{1-\rho} \quad (7)$$

$$L_q = \sum_{n=1}^{\infty} (n-1)p_n = \frac{\rho^2}{1-\rho} \quad (8)$$

平均系内時間  $W$  はリトルの公式  $L = \lambda W$  で求まる. 同様に平均待ち時間  $W_q$  も  $L_q = \lambda W_q$  より求まる.

## 2.4 M/M/S 型待ち行列モデル

次に窓口が複数になった場合を考える. 待ち行列は 1 列に並んでいて, 各窓口には待ち行列はつくらない. ここでは待ち行列の長さに制限がないとする. 窓口数を  $s$  とする.

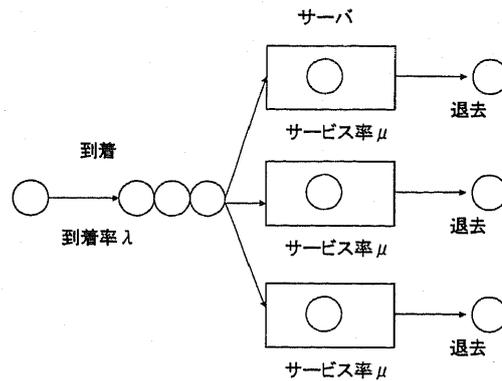


図 2: M/M/S/∞ 待ち行列モデル

次の平衡方程式を得る.

$$\lambda p_0 = \mu p_1 \quad (9)$$

$$(\lambda + n\mu)p_n = \lambda p_{n-1} + (n+1)\mu p_{n+1}, \quad (n = 1, 2, \dots, s-1) \quad (10)$$

$$(\lambda + s\mu)p_n = \lambda p_{n-1} + s\mu p_{n+1}, \quad (n = s, s+1, \dots) \quad (11)$$

ここで,  $a = \lambda/\mu$ ,  $\rho = a/s$  として, 平衡方程式を解くと,

$$p_n = \begin{cases} \frac{a^k}{k!} p_0, & k = 1, 2, \dots, s \\ \rho^{k-s} p_s, & k = s+1, s+2, \dots \end{cases} \quad (12)$$

となる. ここで,

$$p_0 = \left( \sum_{k=0}^{s-1} \frac{a^k}{k!} + \frac{a^s}{s!(1-\rho)} \right)^{-1} \quad (13)$$

である。これから待ち人数  $L_q$ , 系内人数  $L$  の期待値を計算すると,

$$L_q = \frac{s^s \rho^{s+1}}{s!(1-\rho)^2} p_0 \quad (14)$$

$$L = L_q + a \quad (15)$$

また系内時間  $W$ , 待ち時間  $W_q$  はリトルの公式により得られる。

## 2.5 ジャクソンネットワーク

待ち行列が発生する場所が 1ヶ所だけでなく何ヶ所にもある場合に, その待ち行列が起きる場所のいくつかを回らないと 1つのサービスが終了しない場合がある。例えば, 複数の種類の製品をいくつかのラインで生産するとき, 製品の種類によって使用するラインが重複するような場合に, 各製品は各ラインで待ち行列を作る。このように, 待ち行列のシステムがネットワーク状につながっている場合について述べる。ここで考えるネットワークは, ジャクソンネットワークと呼ばれるもので次のようなものである。

- 1 システムを構成する各待ち行列システムのサービスは指数分布に従い, 窓口は 1 個とする。このサービスの窓口をノードと呼ぶ。
- 2 ノードは  $N$  個あり, 1 から  $N$  までの番号が付けられている。
- 3 ノード  $i$  には, 外部から客が到着率  $\lambda_i$  のポアソン到着をする。
- 4 ノード  $i$  でサービスを終了した客は, 確率  $r_{ij}$  でノード  $j$  に行くか, 確率  $r_{i0}$  でシステムから退去する。

ここで,  $r_{i0} = 1 - \sum_{j=1}^N r_{ij}$  とする。  $r_{ij}$  を並べた行列  $R = [r_{ij}]_{i,j=1,\dots,N}$  を経路選択行列と呼ぶ。

ネットワークシステムでは, いくつかのノードから退去する客があわさってあるノードに到着することになるので, その到着率はそれぞれの退去率の和になる。また, あるノードに到着した客は必ず退去するので到着率と退去率は等しいことになる。そこで,

- $l_i(t)$ : 時刻  $t$  におけるノード  $i$  の系内人数
- $a_i$ : ノード  $i$  の客の到着率  $i = 1, \dots, N$

とすると, 到着はシステムの外部からと他のノードからとがあるので,

$$a_j = \lambda_j + \sum_{i=1}^N a_i r_{ij}, \quad j = 1, \dots, N$$

が成立する。この連立方程式を解けば各ノードへの客の到着率を求められる。よって, 系内人数の定常分布は,

$$\begin{aligned}
P(l_1(t) = n_1, \dots, l_N(t) = n_N) &= P(l_1(t) = n_1) \cdots P(l_N(t) = n_N) \\
&= \prod_{i=1}^N (1 - \rho_i) \rho_i^{n_i}
\end{aligned} \tag{16}$$

が成立する. これは, 各ノードにおける系内人数  $l_i(t)$  は, 互いに独立で, 到着率  $a_i$  とサービス率  $\mu_i$  の  $M/M/1$  待ち行列と同じになることを示している.

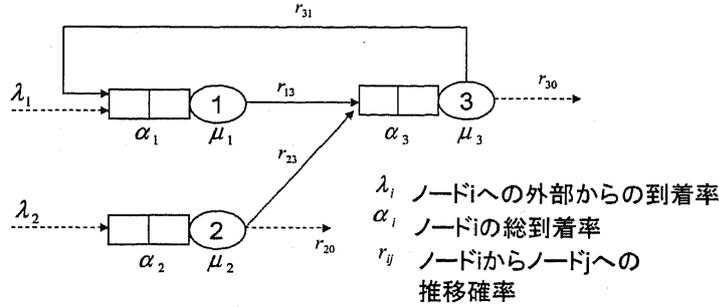


図 3: ジャクソンネットワーク待ち行列モデル

## 2.6 サイクル公式

ここでは, ある期間で特定の時刻で得られている量に対し, 1 サイクルで平均をとり時間的に定常な測度を得る. これは 3 章で使われる公式である. まずずらしの作用素, その定常性及び連動を述べ, サイクル公式を証明する.

定義 [ずらしの作用素]  $\theta_t$  を  $\Omega$  から  $\Omega$  への可測関数で次の条件を満たすとする.

$$\theta_s \circ \theta_t = \theta_{s+t}, \quad s, t \in R.$$

このような  $\theta_t$  をずらしの作用素, その集まりを  $\{\theta_t\}_{t \in R}$  をずらしの作用素群と呼ぶ.  $\square$

定義 [定常なずらし] 任意の  $A \in F$  と任意の  $t \in R$  に対し,

$$P(\theta_t(A)) = P(A),$$

が成り立つとき,  $\{\theta_t\}$  は  $P$  に対し, 定常であるという. ここに  $\theta_t(A) = \{\omega | \theta_t(\omega) \in A\}$  とおく.  $\square$

定義 [連動]  $\{X(t)\}$  を状態空間  $R$  の確率過程とする. 任意の  $s, t \in R$  に対し,

$$X(s) \circ \theta_t = X(s+t),$$

が成り立つとき,  $\{X(t)\}$  は  $\{\theta_t\}$  に連動するという.  $\square$

定義 [点過程, 計数過程] 確率変数列  $\{T_n\}$  が条件

$$\begin{aligned} \lim_{n \rightarrow -\infty} T_n &= -\infty, & \lim_{n \rightarrow \infty} T_n &= \infty, \\ \dots < T_{-1} < T_0 \leq 0 < T_1 < T_2 < \dots, \end{aligned}$$

を満たすとき,  $\{T_n; n \in \mathbb{Z}\}$ ,  $\mathbb{Z}$  は整数, を点過程と呼ぶ.  $N(t) = \sup\{n \geq 0 \mid T_n \leq t\}$  とおく.  $\{N(t)\}$  を計数過程と呼ぶ. 任意の  $t \in \mathbb{R}$  に対し,

$$T_n \circ \theta_t = T_{N(t)+n} - t,$$

が成り立つとき, 点過程  $\{T_n\}$  は  $\{\theta_t\}$  に連動するという. これは  $\{N(t)\}$  が定常計数過程であることと同値である.  $\square$

補題 2.1 単純点過程  $N$  に対して,  $\{\eta_n \in\}$  は可測空間  $(\Omega, F)$  上の離散時間ずらしの作用素群である.

証明.

$$\begin{aligned} \eta_n \circ \eta_m &= \theta_{T_n} \circ \eta_m \\ &= \theta_{T_{m+n}-T_m} \circ \theta_{T_m} \\ &= \theta_{T_{m+n}} = \eta_{m+n}. \end{aligned}$$

$\square$

上記で述べたことを用いてサイクル公式を導く.

**サイクル公式** 単純点過程  $N$  は  $\theta_t$  に連動し,  $0 < E[N((0, 1])] < \infty$  を満たすとする.  $P$  が  $\theta_t$ -定常ならば,  $P_0$  は  $\eta_n$  定常であり,  $P_0$  から  $P$  が次式により得られる.

$$P(A) = \lambda E_0 \left( \int_0^{T_1} \mathbf{1}_{\theta_u^{-1}(A)} du \right), \quad A \in F. \quad (17)$$

証明.  $X(u)$  が実数  $u$  をパラメータにもつ非負値確率変数であるとき,

$$E \left[ \int_{-\infty}^{+\infty} X(u) \circ \theta_u N(du) \right] = \lambda E_0 \left[ \int_{-\infty}^{+\infty} X(u) du \right]. \quad (18)$$

が成り立つ [41]. (18) に対して,  $X(u) = \mathbf{1}(N((-u, 0)) = 0, u > 0) \mathbf{1}_{\theta_{-u}^{-1}(A)}$  とおく. このとき,

$$X(u) \circ \theta_u = \mathbf{1}(N((0, u)) = 0, u > 0) \mathbf{1}_A$$

であるから,

$$E \left[ \int_{-\infty}^{+\infty} X(u) \circ \theta_u N(du) \right] = E[\mathbf{1}(N((0, T_1)) = 0) \mathbf{1}_A] = P(A)$$

である. 一方,

$$\begin{aligned} E_0\left[\int_{-\infty}^{+\infty} X(u)du\right] &= E_0\left[\int_0^{-T_{-1}} \mathbf{1}_{\theta_{-u}(A)}du\right] \\ &= E_0\left[\int_{T_{-1}}^0 \mathbf{1}_{\theta_u^{-1}(A)}du\right] \\ &= E_0\left[\int_{T_{-1}\circ\eta_1}^0 \mathbf{1}_{(\theta_u\circ\eta_1)^{-1}(A)}du\right] \end{aligned}$$

である. ここで最後の等号は  $P_0$  が  $\eta_m$ -定常であることから得られる.  $T_{-1}\circ\eta_1 = T_0 - T_1$  であり,  $\theta_u\circ\eta_1 = \theta_{u+T_1}$  であるから, 最後の式の積分を  $u$  から  $u+T_1$  へ変数変換することにより, (17) が得られる.  $\square$

### 3 ネットワーク型優先権付き待ち行列

2章で述べたように、待ち行列理論が各分野で発展する一方で数学的記述も高度になったことや、システムが安定するための条件から、現実モデルの適用が難しい面がある。2章で述べたことを言い換えると、i) 数学モデル、ii) 応用モデルに分けられる。数学モデルは現実モデルへの適用が難しく、応用モデルは数学的記述が不完全になる可能性がある。

ネットワーク型待ち行列は [19] での基本モデルから始まり、[7] では状態に独立な経路選択を持つ待ち行列ネットワークに対して、積形式の必要十分条件を与え、準可逆性や局所平衡性との関係を導いている。また [38] では状態に依存した経路選択を持つ待ち行列ネットワークに対して、同様に積形式の必要十分条件を与えている。また [39] では線形トラフィックネットワークに対する定常分布を与え、[12] では Jackson ネットワークに負の客を導入した。今あげた先行研究は数学モデルとして書かれている。本研究では対象となる現実モデルを先に考え、待ち行列理論を適用し定常分布を算出する方法をとる。これは数学モデルと応用モデル両方を考えた方法であり、従来のモデル化とは異なっている。後で詳細に述べるが、物流モデル及び交通モデルを対象としたものを扱い、定常分布の算出過程を紹介する。

交通流の解析にも古くから待ち行列は用いられている。交通流の数学的解析では、流体モデルにおいて、偏微分方程式を用い交通密度波の伝播を解析する研究がよく知られている。流体モデルは一般に非積形式であることが証明されている [25]。流体モデルは近似解ではあるが、積形式や線形トラフィックネットワークとは異なる定常分布形を持つことから応用上重要である。交通モデルの特徴として、車の到着がランダムではないことがあげられる。速度の遅い車の後には車が連なっている可能性があり、赤信号によって車の群ができることも考えられる。この結果、交差点に到着する車は、ある程度の群をなしていると考えられ、これは ATM 交換機等の通信システムの到着過程として 1990 年代に盛んに研究されてきたバースト入力と同じ性質である。

交差点のモデルに対する 1960 年代の研究では、解析の容易さからランダムな到着であるポアソン分布が仮定されていた [10]。1970 年代になると、より現実的なモデル化を行うためいろいろなバースト到着が考えられた。単位時間当たりの到着台数が離散マルコフ連鎖をなすモデルの解析 [33] や、車の群を直接モデル化するために、群のサイズを任意の分布、群の中の車の到着間隔を 1、群の最後尾と次の群の先頭の車の到着間隔は幾何分布に従うと仮定した研究 [9] がある。しかしこの後交通流におけるバースト到着の研究は大きく進展しなかった。これは、交通流のバースト性が通信網のそれと比べてあまり高くなかったと想像される。最近では、通信システムのバースト入力モデルとして考えだされた MAP (Markovian Arrival Process) を交差点モデルの到着分布に用いた研究 [1] がある。

これらの先行研究では基本的にひとつの交差点に対し考えられたものである。本研究では任意個の交差点を設定し、そこに到着して合流を待っている車の待ち行列をバースト到着に考え、解析することを考える。流体近似ではなく、待ち行列をそのまま適応し、さらに交差点を任意個に拡張し定常分布を算出したものではなく、モデルとしての評価を得られると考えられる。

また通信システムの進歩に従い、注目されたもののひとつに CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) があり、これは無線 LAN に用いられるアクセ

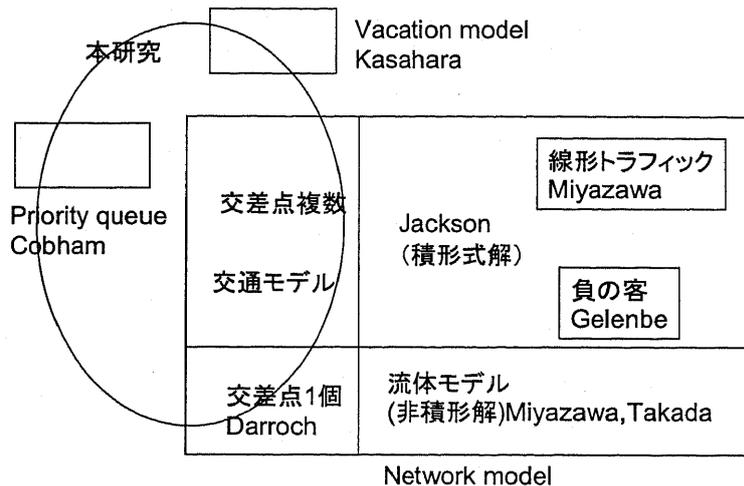


図 4: 先行研究と本研究の位置付け

ス制御方式で、搬送波感知多重アクセス/衝突回避方式と訳される。無線 LAN ではコリジョンを検出できないため、各端末は通信路が一定時間以上継続して空いていることを確認してからデータを送信する。このように、交通や通信システムでは処理をするために余裕空間または余裕時間が必要な場合が多い。待ち行列理論ではこのような余裕時間を遊休期間 (vacation) と呼び、古典的モデルとして従来から多方面に使われてきた。従来の研究は数学モデルが主体であり、サーバ数も 1 個で行われている [20][21][22][23][24]。本研究ではサーバ数は任意個であり、その各サーバにできる待ち行列に対して余裕空間を設けて解析を行う。

### 3.1 はじめに

私たちが自動車を運転するときなど、しばしば混雑に対面する。その中でも典型的なものは幹線道路に支線から合流するとき起こるものである。これは通信分野でも発生し、例えば DQDB (Distributed Queue Dual Bus) や SONET (Synchronous Optical Network)[53] が挙げられる。このタイプの混雑は境界のある領域内で自動車、データや客がランダムに発生するとき起こる。本論文ではこのようなランダムに発生するものをトラフィックと呼ぶことにする。すなわち、トラフィックはポアソン過程に従って発生し、領域内においてランダムで独立に発生場所が決まる。これらのトラフィックは支線上を動き、幹線の合流地点まで移動しある条件の下で幹線に合流する。合流後、トラフィックは幹線の終点にある同じ目的地に一定速度で移動する。本モデルの目的は合流地点の混雑度を評価することである。

一般的に支線から幹線に合流するときには合流スペースに余裕が必要な場合がある。例えば車を運転するとき、合流時安全のため十分なスペースが必要となる。データ転送時にはこのようなスペースはデータ転送準備時間と考えられる。よって幹線への支線から流入

時にスペースに余裕を持つことは自然な考えである。そのために次のような余裕空間条件を提案する。支線で発生したトラフィックは合流点で固定された空き空間が通過後、幹線に合流できる。空き空間が続けてくれば連続して合流できる。これを連続余裕空間条件と呼ぶ。定常性を仮定して各合流点でのトラフィックの長さの定常方程式を考える。このモデルは [34] によって連続余裕空間条件を含めない形で提案された。本論文ではアプリケーションに有用な連続余裕空間条件を含めて、違う方法で解析する。

このモデルは本来連続型モデルである。しかしながら、適切な時間単位で離散モデルとして提案する。トラフィックが合流点で幹線に合流し次の合流点に移動する時間を1単位時間とし、幹線上ではトラフィックは一定速度で移動する。よって幹線を分割することができる。トラフィックの発生方法からトラフィックの各合流点での長さは各合流点で独立であるが、先の合流点の影響を受けることから分布は依存している。また幹線に対して合流点の場所から合流する順番が付けられることから連続余裕空間がある優先権付き待ち行列と見ることができる。

本稿は、まず離散型マルコフ連鎖として定義し、退去直前の時点の各合流点のトラフィックの長さを考える。その後、定常分布を計算しその結果から数値計算を与える。

### 3.2 離散マルコフ連鎖での表現

先に述べた合流地点における離散時間待ち行列を定義する。 $N$  は合流地点の合計であり、幹線の開始地点から順に  $1, 2, \dots, N$  と付けられている。合流地点  $k$  のことをレベル  $k$  と呼ぶ。また  $L = \{1, 2, \dots, N\}$ 。離散時間モデルの仮定は以下のようである。

- 1)  $c$  を正の整数とする。レベル1を除いて、各レベルでは合流地点で待っているトラフィックは幹線上に  $c+1$  個の連続な空きスペースが上位レベルから通過するときのみ、 $c+1$  番目の空きスペースに合流できる。そして  $c$  個の連続な空きスペースが通過後、続けてくる  $c+1$  の空きスペースに合流すれば合流地点で待っているトラフィックはそれに続く空きスペースに連続して合流できる。これを連続余裕空間条件と呼ぶ。レベル1では、 $c$  個の連続空きスペースがなくても合流できる。
- 2) トラフィックがレベル  $(k-1)$  からレベル  $k$  に幹線上を移動するには1単位時間必要かかる。
- 3)  $A_k(m, n)$  を時間間隔  $(m, n)$  にレベル  $k$  に到着する個数とする。ここで  $m < n$  である。 $A_k(n, n+1)$   $n = 0, 1, \dots$  は有限な平均と分散を持ち独立に同一に分布すると仮定する。特に  $\{A_k(n, n+1); n \geq 0\}$  はベルヌーイ過程であり、 $A_k(m, n) = \sum_{\ell=m}^{n-1} A(\ell, \ell+1)$  である。

図5はこのモデルを表現したものである。 $A_k(n, n+1)$  は各  $k$  に対するポアソン分布に従う。しかしながら、このモデルの記述は  $A_k(n, n+1)$  と独立していることから、 $A_k(n, n+1)$  がポアソン分布に従うという仮定をはずし、一般分布に従うと仮定する。

このモデルを確率過程モデルとして述べる。 $k$  に対して  $\in L$ ,  $X_k^-(n)$  をレベル  $k$  での時刻  $n \geq 0$  直前の待っているトラフィック数とする。簡単のために、 $X_k^-(n) = 0, n \leq 0$  とす

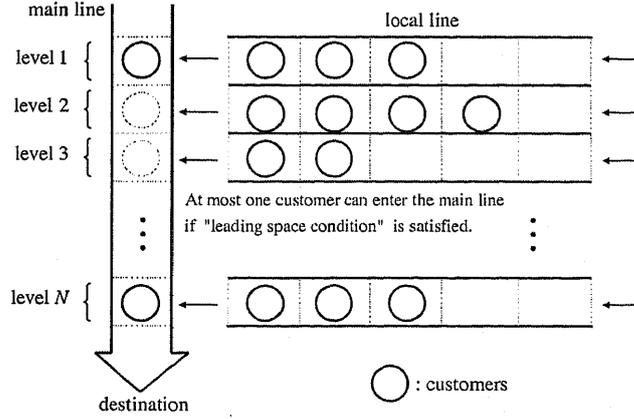


図 5: 合流地点における待ち行列モデル

る. 各  $k \in L \cup \{0\}$ ,  $n \geq 0$  に対して次の事象を定義する.

$$G_k(n) = \{X_k^-(n-l) = 0, \quad 0 \leq l \leq c\},$$

$$F_k(n) = G_k(n) \cap F_{k-1}(n-1),$$

ここで  $F_0(n)$  は全標本空間  $\Omega$  である.  $F_k(n)$  は時刻  $n$  にレベル  $k$  において  $c+1$  個の空きスペースが連続的に通過したことが観測されたことを意味する事象である. よって  $X_k^-(n)$  は再生過程である. さらに連続空き空間によってレベル  $k$  の待ちトラフィックは時刻  $n$  に幹線に合流できる. 言い換えれば, 事象  $F_{k-1}(n-1)$  が起こったとき, 時刻  $n$  でレベル  $k$  において即時サービスが可能である. よって,  $n \geq 1$  に対して

$$X_k^-(n+1) = \max\{X_k^-(n) - \mathbf{1}_{F_{k-1}(n-1)}, 0\} + A_k(n, n+1), \quad k \in L,$$

ここで  $\mathbf{1}_A$  は事象  $A$  の指示関数である. これらの方程式は  $\{X_k^-(n); n \geq 0\}$  が  $\{A_\ell(n, n+1); n \geq 0\}$ ,  $1 \leq \ell \leq k$  から再帰的に決定されることを示している.

$(X_1^-(n-(N-1)), X_2^-(n-(N-2)), \dots, X_N^-(n))$  は離散時間型優先権付き待ち行列と考えられる. しかしながら, 連続余裕空間条件があるので単純優先権付き待ち行列ではない [52]. さらに,  $c \geq 2$  ではこの結合過程はマルコフ連鎖にならない.

### 3.3 埋め込み時刻における定常分布の計算

ここでの第一の目標は  $k \in L$  に対して,  $\{X_k^-(n); n \geq 0\}$  の定常分布を算出することである.  $k=1$  を除いて, マルコフ連鎖になっていないので, 待ち行列が合流できる直前の時刻に注目した過程を考える. この合流可能時刻を埋め込み時刻と呼ぶ. その結果, この埋め込み過程はマルコフ連鎖と考えられる. この推移確率は埋め込み時刻間の到着人数分布が必要であり, これはひとつ上のレベルのマルコフ連鎖から得ることができる. その結果, レベル 1 から再帰的にレベル  $k$  の埋め込み時刻における定常分布を計算できる. この

章で最初に埋め込み過程を考え、その結果から 3.4 節で  $\{X_k^-(n); n \geq 0\}$  の定常分布を計算する。

最初に、レベル 1 を考える。この場合、次を得る。

$$X_1^-(n+1) = \max\{X_1^-(n) - 1, 0\} + A_1(n, n+1). \quad (19)$$

この方程式は  $M/G/1$  待ち行列で退去直前の人数と同じことをあらわしている。  $0 < z \leq 1$  に対して次の母関数を定義する。

$$\tilde{a}_k(z) = E[z^{A_k}], \quad k \in L,$$

ここで  $A_k$  は  $A_k(n, n+1)$  と同じ分布に従う確率変数である。  $k \in L$  に対して、  $X_k^-$  をレベル  $k$  における待ち人数をあらわす確率変数とする。そして  $\tilde{x}_k(z)$  は次にあらわされる母関数である。  $\tilde{x}_k(z) = E[z^{X_k^-}]$ 。 よって、ポラチック-ヒンチンの公式 [3] から次の結果が得られる。

**補題 3.1**  $E[A_1] < 1$  のとき、  $\{X_1^-(n); n \geq 0\}$  は定常分布を持ち、その母関数  $\tilde{x}_1(z)$  は次で与えられる。

$$\tilde{x}_1(z) = (1 - E[A_1]) \frac{(z-1)\tilde{a}_1(z)}{z - \tilde{a}_1(z)}. \quad (20)$$

各  $k \in L \setminus \{1\}$  において、  $\{X_k^-(n); n \geq 0\}$  は上位の合流地点の影響を受けるのでマルコフ性を持たない。マルコフ性を持つために次の確率変数を紹介する。  $k \in L$  と  $n \geq 1$  に対して、

$$\begin{aligned} \sigma_k(0) &= 0, \quad \sigma_k(n) = \inf_{\ell \geq 0} \{\ell > \sigma_k(n-1); F_k(\ell) \text{ holds}\}, \\ T_k(n) &= \sigma_k(n) - \sigma_k(n-1), \\ Y_k^-(n) &= X_k^-(\sigma_{k-1}(n) + 1). \end{aligned}$$

とおく。また、  $n = 1, 2, \dots$  に対して  $T_0(n) = 1$  とおく。  $F_k(0)$  は常に成立している。  $k = 0, 1, \dots, N$  に対して、  $\{T_k(n); n \geq 1\}$  は i.i.d 再生確率変数であり、そして  $\{\sigma_k(n); n \geq 0\}$  はその再生過程となる。  $\sigma_k(n)$  は  $\{X_k^-(n)\}$  の再生時刻になる。さらに、  $n = 0, 1, \dots$  に対して、  $\sigma_{k-1}(n) + 1$  はレベル  $k$  で連続余裕空間条件を満たす時刻となる。よって、  $\{Y_k^-(n)\}$  はレベル  $k$  でサービスが行われる直前の時刻での待ち行列の長さとなる。また次を定義する。

$$U_k(n) = A_k(\sigma_{k-1}(n-1) + 1, \sigma_{k-1}(n) + 1),$$

これはレベル  $k$  での合流可能時刻間隔の総到着人数である。

図 6 にレベル  $k-1$  と  $k$  で余裕空間条件  $c=3$  でのサンプルパスをかく。レベル  $(k-1)$  で " $c+1=4$ " の空き空間が時刻 0, 5, 6 そして 12 に観測されたとき、  $\{X_{k-1}^-(n); n \geq 0\}$  となる。その結果、  $\sigma_{k-1}(0) = 0$ ,  $\sigma_{k-1}(1) = 5$ ,  $\sigma_{k-1}(2) = 6$  そして  $\sigma_{k-1}(3) = 12$  となり、  $T_{k-1}(1) = 5$ ,  $T_{k-1}(2) = 1$  そして  $T_{k-1}(3) = 6$  を示している。同様に、レベル  $k$  での再生

時刻は  $\sigma_k(0) = 0, \sigma_k(1) = 1$  として  $\sigma_k(2) = 6$ , また再生間隔は  $T_k(1) = 1$  と  $T_k(2) = 5$  より計算する. レベル  $k$  での客は時刻  $n-1$  にレベル  $(k-1)$  で再生時刻のとき, レベル  $k$  で時刻  $n$  においてサービス可能となる.  $U_k(1) = 0, U_k(2) = 3$  として  $U_k(3) = 1$  はレベル  $k$  での再生間隔での到着人数であるので, レベル  $k$  でのサービス可能時刻の直前の人数は  $Y_k^-(0) = Y_k^-(1) = 0$  として  $Y_k^-(2) = Y_k^-(3) = 3$  となる.

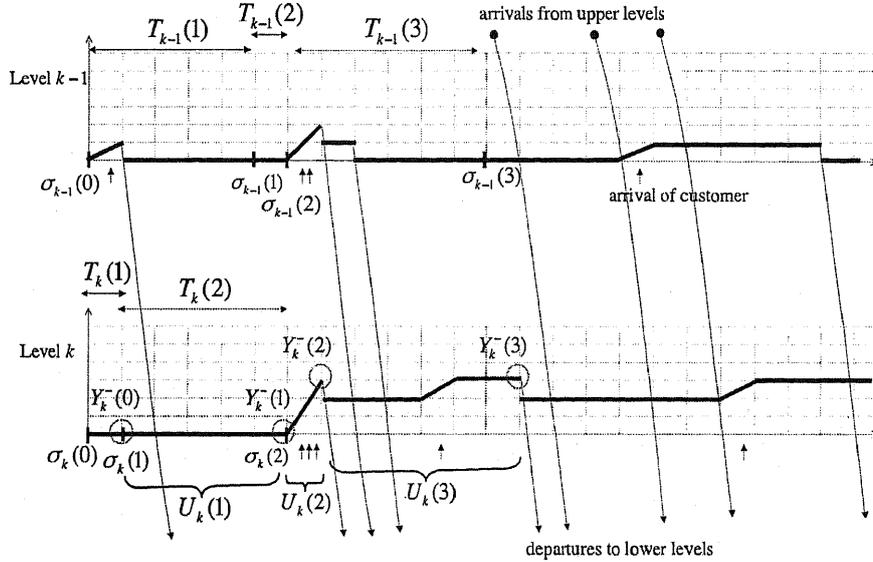


図 6: レベル  $k-1$  と  $k, c=3$  でのサンプルパス

$\{U_k(n); n \geq 1\}$  は i.i.d の確率変数である.  $\tilde{u}_k(z)$  と  $\tilde{g}_k(z)$  は  $n$  に対して独立なそれぞれ  $U_k(n)$  と  $T_k(n)$  の母関数である.  $U_k(1) = \sum_{m=1}^{T_{k-1}(1)} A_k(m, m+1)$  なので, 次を得る.

$$\begin{aligned} \tilde{u}_k(z) &= E \left( z^{\sum_{m=1}^{T_{k-1}(1)} A_k(m, m+1)} \right) \\ &= \sum_{\ell=1}^{\infty} \tilde{a}_k^\ell(z) P(T_{k-1}(1) = \ell) = \tilde{g}_{k-1}(\tilde{a}_k(z)). \end{aligned} \quad (21)$$

(21) から, 次の式が導かれる.

$$E[U_k] = E[A_k]E[T_{k-1}], \quad (22)$$

ここで  $U_k$  と  $T_{k-1}$  はそれぞれ  $U_k(n)$  と  $T_{k-1}(n)$  の同じ分布を持つ確率変数である.

レベル  $k$  の一人の客は  $\sigma_{k-1}(n) + 1$  の直後サービスが行われるので, 到着した客は次の可能サービス時間まで待ち行列に入る. つまり  $\sigma_{k-1}(n+1) + 1$  である. よって, 次を得る.

$$Y_k^-(n+1) = \max\{Y_k^-(n) - 1, 0\} + U_k(n+1).$$

よって,  $\{Y_k^-(n); n \geq 1\}$  は離散時間型  $M/G/1$  型待ち行列の待ち行列の長さの過程と考えることができる. よく知られている結果で  $\{Y_k^-(n); n \geq 1\}$  は安定条件と言われる  $E[U_k] < 1$  の時, 定常分布を持つ.  $Y_k^-$  を  $Y_k^-(n)$  の定常分布に従う確率変数とおき, その母関数を  $\tilde{y}_k(z)$  と書く. 補題 3.1 と同様に, 次の結果を得る.

補題 3.2  $E[U_k] < 1$  のとき, 母関数  $Y_k^-$  は次で与えられる.

$$\tilde{y}_k(z) = (1 - E[U_k]) \frac{(z-1)\tilde{u}_k(z)}{z - \tilde{u}_k(z)}, \quad k \in L. \quad (23)$$

よって,  $\tilde{u}_k(z)$  を得るために必要な情報は, (21) で与えられる  $\tilde{g}_{k-1}(z)$  によって得られる. この章の最後に,  $k \in L$  に対して  $\tilde{g}_k(z)$  を計算する.  $k \in L$  に対して  $\eta_k(n)$  と  $\tau_k$  を次のように定義する.

$$\begin{aligned} \eta_k(n) &= \sigma_{k-1}(n) + 1, \quad n \geq 0, \\ \tau_k &= \min \{ \eta_k(\ell) - 1; X_k^-(\eta_k(\ell)) = \max \{ X_k^-(\eta_k(0)) - 1, 0 \}, \ell \geq 1 \}. \end{aligned}$$

言い換えれば,  $\eta_k(n)$  はレベル  $k$  の  $n$  番目のサービス可能時刻である. またこの時間間隔は独立であり, これを  $\tau_k$  とする. 特に,  $m \geq 1$  に対して,  $\tau_k$  はレベル  $k$  で待ち行列の長さがひとつ減る時間間隔を表している.

$\{\tau_k(\ell); \ell \geq 1\}$  は  $\tau_k$  と同じ分布を持つ確率変数である. レベル  $k$  でひとつのサービス期間をみると, それはレベル  $k-1$  の再生間隔に相当し, それは  $T_{k-1}$  で分布する. もし  $j$  人の客がこの期間に来れば,  $\tau_k(1) + \dots + \tau_k(j)$  の時刻が待ち行列がひとつ減るために必要となる. よって, 次を得る.

$$\tau_k \stackrel{d}{\sim} T_{k-1} + \sum_{\ell=1}^{A_k(1, T_{k-1}+1)} \tau_k(\ell), \quad (24)$$

ここで  $\stackrel{d}{\sim}$  は分布について等しいことを意味している. 図 7 は (24) のサンプルパスをかいたものである.

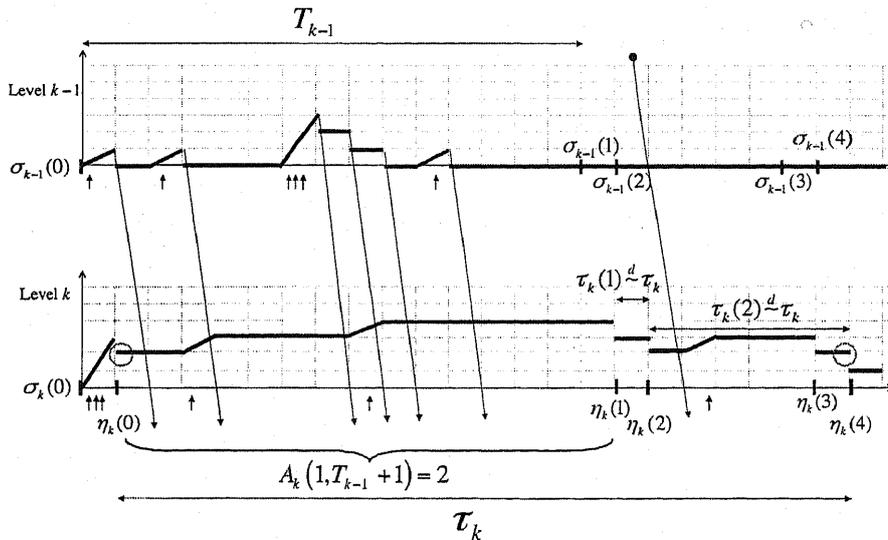


図 7:  $\tau_k$  の  $c=3$  におけるサンプルパス

この図で,  $\sigma_{k-1}(0) = 0, \sigma_{k-1}(1) = 15, \sigma_{k-1}(2) = 16, \sigma_{k-1}(3) = 21$  と  $\sigma_{k-1}(4) = 22$  はレベル  $(k-1)$  での再生時刻である. よって, レベル  $k$  でのサービス可能時刻は  $\eta_k(0) = 1, \eta_k(1) = 16, \eta_k(2) = 17, \eta_k(3) = 22$  と  $\eta_k(4) = 23$  になる. 従って  $X_k^-(\eta_k(0)) = 3, X_k^-(\eta_k(1)) = 2, X_k^-(\eta_k(2)) = 3$  と  $X_k^-(\eta_k(3)) = 2$ , よって  $\tau_k = \eta_k(4) - \eta_k(0) = 22$  となる.

$\tilde{\tau}_k(z)$  を  $\tau_k$  の母関数とすると, 次の結果を得られる.

**補題 3.3**  $0 < z < 1$  に対して,  $\tilde{\tau}_k(z)$  は次によって一意に決定される.

$$\tilde{\tau}_k(z) = \tilde{g}_{k-1}(z\tilde{a}_k(\tilde{\tau}_k(z))), \quad k \in L, \quad (25)$$

ここで  $\tilde{g}_0(z) = z$ .

証明. 母関数 (24) を用いると, 次を得る.

$$\begin{aligned} \tilde{\tau}_k(z) &= E[z^{T_{k-1} + \sum_{\ell=1}^{A_k(1, T_{k-1}+1)} \tau_k(\ell)}] \\ &= \sum_{u=0}^{\infty} \sum_{j=1}^{\infty} E[z^{j + \sum_{\ell=1}^u \tau_k(\ell)}] P(T_{k-1} = j) P(A_k(1, j+1) = u) \\ &= \sum_{j=1}^{\infty} \{z\tilde{a}_k(\tilde{\tau}_k(z))\}^j P(T_{k-1} = j). \end{aligned}$$

よって, (25) を得る. 次に  $\tilde{\tau}_k(z)$  の一意性を示す.  $z \in (0, 1)$  に対して,  $\xi = \tilde{\tau}_k(z)$  とおく, そのとき (25) は次のように書ける

$$\xi = \tilde{g}_{k-1}(z\tilde{a}_k(\xi)). \quad (26)$$

この方程式が  $\xi \in (0, 1)$  で一意に決まることを示している.  $\tilde{g}_{k-1}(z\tilde{a}_k(\xi))$  は凸関数であり  $z$  に対して  $\xi$  の増加関数となる.  $\tilde{a}_k(0) = P(A_k = 0) > 0$  及び  $\tilde{a}_k(\xi)$  は凸関数であり  $\xi$  で増加関数であるので, 次を得る.

$$0 < \tilde{g}_{k-1}(z\tilde{a}_k(0)) < \tilde{g}_{k-1}(z\tilde{a}_k(1)) = \tilde{g}_{k-1}(z) < 1, \quad 0 < z < 1.$$

(26) は  $(0, 1)$  で唯ひとつの解を持つことを示している. □

補題 3.3 から, 各  $z$  に対し次のアルゴリズムで数値解析的に  $\tilde{\tau}_k(z)$  が計算できる.

**( $\tilde{\tau}_k(z)$  のアルゴリズム)**

*Step 1*  $\xi_0 = 0$  とおく

*Step 2*  $\xi_n = \tilde{g}_{k-1}(z\tilde{a}_k(\xi_{n-1})), n \geq 1$  を  $|\xi_n - \xi_{n-1}| < \epsilon$  まで繰り返す, ここで  $\epsilon$  は十分小さい正数.

*Step 3*  $\xi_n$  を返す

$k \in L$  に対して  $\tilde{g}_k(z)$  を考える．そのために、いくつかの表現が必要である．各  $k \in L$  に対して、

$$q_k = \prod_{i=1}^k P(A_i = 0) = \prod_{i=1}^k \tilde{a}_i(0),$$

及び  $D_k$  はパラメータ  $q_k$  の幾何分布に従う確率変数であり、次のようになる．

$$P(D_k = j) = (1 - q_k)q_k^j, \quad j \geq 0.$$

$\hat{T}_k$  は  $A_k(0, 0) \geq 1$  または  $T_{k-1} \geq 2$  の条件で与えられる  $T_k$  の確率変数である．

$$P(\hat{T}_k = \ell) = P(T_k = \ell | A_k(0, 1) \geq 1 \text{ or } T_{k-1} \geq 2) \quad \text{for } \ell \geq 1,$$

これを用いて、次のサンプルパスが得られる．

**補題 3.4**  $k \in L$  に対して、

$$T_k \stackrel{d}{\sim} T_{k-1} + \begin{cases} 0, & \text{if } A_k(0, T_{k-1}) = 0, \\ \sum_{m=1}^{A_k(0, T_{k-1})-1} \tau_k(m) + c + 1, & \text{if } A_k(0, T_{k-1}) \geq 1, D_k \geq c + 1, \\ \sum_{m=1}^{A_k(0, T_{k-1})-1} \tau_k(m) + D_k + \hat{T}_k, & \text{if } A_k(0, T_{k-1}) \geq 1, D_k \leq c. \end{cases}$$

証明．時刻 0 をレベル  $k$  の再生時刻とする．そのとき、時刻  $-1$  はレベル  $k-1$  の再生時刻である．よって、 $(-1, T_{k-1} - 1]$  はレベル  $k-1$  の再生間隔であり、 $T_{k-1}$  はレベル  $k$  での時刻 0 以降の最初のサービス可能時刻となる．最初に  $A_k(0, T_{k-1}) = 0$  の場合を考える．この場合、レベル  $k$  には  $(0, T_{k-1})$  の間到着がない．よって、 $T_k$  は  $T_{k-1}$  となる．次に、 $A_k(0, T_{k-1}) \geq 1$  の場合を考える．この場合、レベル  $k$  にはこの時間間隔に到着がある．その到着数は  $A_k(0, T_{k-1})$  であり客がこの期間の最後にサービスを受ける、このレベル  $k$  の待ち行列は  $\sum_{m=1}^{A_k(0, T_{k-1})-1} \tau_k(m)$  時間経過後再び 0 となる．その後、連続的な  $c+1$  個の空きスペースを観測すれば、レベル  $k$  の次の再生時刻になる．これが 2 番目のケースである．その他では、連続的な  $c+1$  個の空き空間を観測するまで同様の期間を繰り返す．その期間は条件  $A_k(0, 1) \geq 1$  or  $T_{k-1} \geq 2$  で与えられる  $T_k$  の分布に従う、これが 3 番目のケースである．  $\square$

$\tilde{g}_k(z)$  を計算するために次の補題が必要である．

**補題 3.5**  $\hat{g}_k(z)$  を  $\hat{T}_k$  の母関数とする、よって

$$\hat{g}_k(z) = \frac{\tilde{g}_k(z) - zq_k}{1 - q_k}. \quad (27)$$

証明． $\hat{T}_k$  の定義から、次を得る．

$$\begin{aligned} \hat{g}_k(z) &= E[z^{\hat{T}_k} | A_k(0, 1) \geq 1 \text{ or } T_{k-1} \geq 2] \\ &= \frac{1}{1 - P(A_k(0, 1) = 0, T_{k-1} = 1)} (E[z^{\hat{T}_k}] - E[z^{\hat{T}_k} \mathbf{1}_{(A_k(0, 1) = 0, T_{k-1} = 1)}]). \end{aligned}$$

$A_k(0, 1) = 0$  及び  $T_{k-1} = 1$  のとき、 $P(A_k(0, 1) = 0, T_{k-1} = 1) = q_k$  及び  $T_k = 1$  なので、(27) を得る．  $\square$

$\tilde{g}_k(x)$  を補題 3.3 を用いて  $k = 1$  から  $N$  に再帰的に求め、次の結果を得る。

補題 3.6  $T_k$  の母関数  $\tilde{g}_k$  は

$$\tilde{g}_k(z) = 1 + \frac{(\tilde{\tau}_k(z) - 1)\tilde{g}_{k-1}(z\tilde{a}_k(0))}{\tilde{\tau}_k(z) - \gamma_k(z)f_k(z)}, \quad (28)$$

ここで

$$f_k(z) = \tilde{\tau}_k(z) - \tilde{g}_{k-1}(z\tilde{a}_k(0)), \quad \gamma_k(z) = \frac{1 - (zq_k)^{c+1}}{1 - zq_k}.$$

証明. 補題 3.4 から,  $T_k$  のように計算する.

$$\begin{aligned} E[z^{T_k}] &= E[z^{T_{k-1}} \mathbf{1}_{\{A_k(0, T_{k-1})=0\}}] \\ &\quad + E\left[z^{T_{k-1} + \sum_{m=1}^{A_k(0, T_{k-1})-1} \tau_k(m) + c + 1} \mathbf{1}_{\{A_k(0, T_{k-1}) \geq 1, D_k \geq c+1\}}\right] \\ &\quad + E\left[z^{T_{k-1} + \sum_{m=1}^{A_k(0, T_{k-1})-1} \tau_k(m) + D_k + \hat{T}_k} \mathbf{1}_{\{A_k(0, T_{k-1}) \geq 1, D_k \leq c\}}\right]. \end{aligned}$$

第 1 項の右辺は次のように計算を行う.

$$\begin{aligned} E[z^{T_{k-1}} \mathbf{1}_{\{A_k(0, T_{k-1})=0\}}] &= \sum_{n=1}^{\infty} E[z^n \mathbf{1}_{\{A_k(0, n)=0\}}] P(T_{k-1} = n) \\ &= \sum_{n=1}^{\infty} z^n \tilde{a}(0)^n P(T_{k-1} = n) \\ &= \tilde{g}_{k-1}(z\tilde{a}_k(0)). \end{aligned}$$

第 2 項は

$$\begin{aligned} &E\left[z^{T_{k-1} + \sum_{m=1}^{A_k(0, T_{k-1})-1} \tau_k(m) + c + 1} \mathbf{1}_{\{A_k(0, T_{k-1}) \geq 1, D_k \geq c+1\}}\right] \\ &= z^{c+1} P(D_k \geq c+1) \sum_{n=1}^{\infty} z^n \sum_{\ell=1}^{\infty} E\left[z^{\sum_{m=1}^{\ell-1} \tau_k(m)}\right] P(A_k(0, n) = \ell) P(T_{k-1} = n) \\ &= (zq_k)^{c+1} \tilde{\tau}_k(z)^{-1} \{\tilde{g}_{k-1}(z\tilde{a}_k(\tilde{\tau}_k(z))) - \tilde{g}_{k-1}(z\tilde{a}_k(0))\} \\ &= (zq_k)^{c+1} \tilde{\tau}_k(z)^{-1} f_k(z), \end{aligned}$$

ここで最後の等式は (25) を用いることで得られる. 同様に, 第 3 項は

$$\begin{aligned} &E\left[z^{T_{k-1} + \sum_{m=1}^{A_k(0, T_{k-1})-1} \tau_k(m) + D_k + \hat{T}_k} \mathbf{1}_{\{A_k(0, T_{k-1}) \geq 1, D_k \leq c\}}\right] \\ &= \sum_{n=1}^{\infty} \left\{ \sum_{\ell=1}^{\infty} \left\{ \sum_{d=0}^c z^n z^d E\left[z^{\sum_{m=1}^{\ell-1} \tau_k(m)}\right] \hat{g}_k(z) P(D_k = d) \right\} P(A_k(0, n) = \ell) \right\} P(T_{k-1} = n) \\ &= \hat{g}_k(z) \tilde{\tau}_k(z)^{-1} \{\tilde{g}_{k-1}(z\tilde{a}_k(\tilde{\tau}_k(z))) - \tilde{g}_{k-1}(z\tilde{a}_k(0))\} \sum_{d=0}^c z^d P(D_k = d) \\ &= (\tilde{g}_k(z) - zq_k) \tilde{\tau}_k(z)^{-1} f_k(z) \gamma_k(z) \end{aligned}$$

ここで最後の等式は補題 3.5 を用いることで得られる。3 項の和を求めると、次を得る。

$$\begin{aligned}\tilde{g}_k(z) &= \tilde{g}_{k-1}(z\tilde{a}_k(0)) + (zq_k)^{c+1}\tilde{\tau}_k(z)^{-1}f_k(z) + (\tilde{g}_k(z) - zq_k)\tilde{\tau}_k(z)^{-1}f_k(z)\gamma_k(z) \\ &= \tilde{\tau}_k(z)^{-1}(\tilde{g}_{k-1}(z\tilde{a}_k(0)) + (1 - (1 - zq_k)\gamma_k(z) + (\tilde{g}_k(z) - zq_k)\gamma_k(z))f_k(z)) \\ &= \tilde{\tau}_k(z)^{-1}(\tilde{g}_{k-1}(z\tilde{a}_k(0)) + (\tilde{g}_k(z) - 1)\gamma_k(z)f_k(z)).\end{aligned}$$

よって (28) が得られる。 □

補題 3.3 及び 補題 3.6 から、 $E[T_k]$  を計算するための再帰公式を与える。

補題 3.7  $k \in L$  に対して、

$$E[T_k] = \frac{E[T_{k-1}]\tilde{g}_{k-1}(\tilde{a}_k(0))}{(1 - E[A_k]E[T_{k-1}])(1 - f_k(1)\gamma_k(1))}, \quad (29)$$

ここで  $E[T_0] = 1$ ,  $f_k(1) = 1 - \tilde{g}_{k-1}(\tilde{a}_k(0))$  及び  $\gamma(1) = \frac{1-q_k^{c+1}}{1-q_k}$ .  
証明. (25) の両辺を微分し  $z = 1$  を代入すると、

$$E[\tau_k] = (1 + E[\tau_k]E[A_k])E[T_{k-1}],$$

これは次を導く。

$$E[\tau_k] = \frac{E[T_{k-1}]}{1 - E[A_k]E[T_{k-1}]}. \quad (30)$$

同様に、(28) の両辺を微分し  $z = 1$  を代入すると、

$$E[T_k] = \frac{E[\tau_k]\tilde{g}_{k-1}(\tilde{a}_k(0))}{1 - f_k(1)\gamma_k(1)}.$$

よって、(29) を得る。 □

### 3.4 定常分布の計算

定常性  $E[U_k] < 1$  の下で  $X_k^-$  の定常分布を計算する準備ができた。ここまで補題 3.1 においてレベル 1 での待ち人数の定常分布を求め、補題 3.2 では上位レベルの再生点を利用して時刻  $\sigma_{k-1} + 1$  における待ち人数の定常分布を算出した。両補題はともに合流可能時刻に注目したときの待ち人数分布である。また、補題 3.3 から補題 3.7 の各補題で再生間隔の定常分布及びそのモーメントまでの計算過程を示した。これらからサイクル公式 [3] を適用して任意時刻での待ち人数の定常分布を算出する。また  $\check{a}_k(\theta)$  及び  $\check{x}_k(\theta)$  をそれぞれ  $A_k$  と  $X_k^-$  の積率母関数とする。  $\check{a}_k = \check{a}_k(e^\theta)$  及び  $\check{x}_k(\theta) = \check{x}_k(e^\theta)$ .

定理 3.1  $k \in L$  に対して,  $E[U_k] < 1$  の下で,

$$\check{x}_k(\theta) = \frac{1 - E[A_k]E[T_{k-1}]}{E[T_{k-1}]} \frac{(e^\theta - 1)\check{a}_k(\theta)}{e^\theta - \check{g}_{k-1}(\check{a}_k(\theta))} \frac{1 - \check{g}_{k-1}(\check{a}_k(\theta))}{1 - \check{a}_k(\theta)}, \quad \theta \leq 0. \quad (31)$$

証明. (22) より  $E[A_k]E[T_{k-1}] = E[U_k] < 1$  であり, また仮定より,  $E[T_{k-1}]$  は有限である. よって  $X_k^-(n)$  の定常分布を求めるために  $\{T_{k-1}(n)\}$  各々にサイクル公式を適用する. よって次を得る.

$$\begin{aligned} E[z^{X_k^-}] &= \frac{1}{E[T_{k-1}]} E \left[ \sum_{n=1}^{T_{k-1}} z^{X_k^-(n)} \right] \\ &= \frac{1}{E[T_{k-1}]} E \left[ \sum_{n=1}^{T_{k-1}} z^{\max(Y_k^- - 1, 0) + A_k(0, n)} \right] \\ &= \frac{1}{E[T_{k-1}]} E \left[ z^{\max(Y_k^- - 1, 0)} \right] E \left[ \sum_{n=1}^{T_{k-1}} \tilde{a}_k^n(z) \right]. \end{aligned}$$

(21), (23) 及び  $\tilde{y}_k(0) = 1 - E[U_k]$  より,

$$\begin{aligned} E \left[ z^{\max(Y_k^- - 1, 0)} \right] &= z^{-1} E \left[ z^{Y_k^-} \mathbf{1}(Y_k^- \geq 1) \right] + P(Y_k^- = 0) \\ &= z^{-1} (\tilde{y}_k(z) + (z - 1)\tilde{y}_k(0)) \\ &= (1 - E[U_k]) \frac{z - 1}{z - \check{g}_{k-1}(\tilde{a}(z))}. \end{aligned}$$

一方,

$$E \left[ \sum_{n=1}^{T_{k-1}} \tilde{a}_k^n(z) \right] = \tilde{a}_k(z) E \left[ \frac{1 - \tilde{a}_k^{T_{k-1}}(z)}{1 - \tilde{a}_k(z)} \right] = \frac{\tilde{a}_k(z)(1 - \check{g}_{k-1}(\tilde{a}_k(z)))}{1 - \tilde{a}_k(z)}.$$

$z = e^\theta$  とおくと, (31) が得られる. □

$\theta = 0$  のまわりで (31) の両辺を微分すると, 次の結果が得られる.

系 3.1  $k \in L$  に対して,

$$E[X_k^-] = \frac{V[A_k]E[T_{k-1}] + E[A_k]^2V[T_{k-1}]}{2E[A_k]E[T_{k-1}](1 - E[A_k]E[T_{k-1}])} - \frac{V[A_k]}{2E[A_k]} + \frac{E[A_k]}{2}, \quad (32)$$

また, リトルの公式より, レベル  $k$  の平均待ち時間  $E[W_k]$  は

$$E[W_k] = \frac{1}{E[A_k]} \times E[X_k^-].$$

これらの結果は  $E[X_k^-]$  及び  $E[W_k]$  が上位レベルの  $T_{k-1}$  に影響されることを示している。これらを算出するために、 $A_1, A_2, \dots, A_{k-1}$  の与えられた分布から  $E[T_{k-1}]$  及び  $V[T_{k-1}]$  を計算する必要がある。補題 3.7 から、 $\tilde{g}_0(\tilde{a}_1(0)), \tilde{g}_1(\tilde{a}_2(0)), \dots, \tilde{g}_{k-2}(\tilde{a}_{k-1}(0))$  を用いて再帰的に  $E[T_{k-1}]$  を計算することができる。補題 3.7 と同様に、補題 3.6 を用いて  $V[T_{k-1}]$  を計算できる。

$$V[T_{k-1}] = \frac{\tilde{g}_{k-2}(\tilde{a}_{k-1}(0))(V[\tau_{k-1}] + E[\tau_{k-1}]^2) + 2\tilde{a}_{k-1}(0)\tilde{g}'_{k-2}(\tilde{a}_{k-1}(0))E[\tau_{k-1}]}{1 - \gamma_{k-1}(1)f_{k-1}(1)} - \frac{2\tilde{g}_{k-2}(\tilde{a}_{k-1}(0))E[\tau_{k-1}]B_{k-1} + E[\tau_{k-1}]^2\tilde{g}_{k-2}(\tilde{a}_{k-1}(0))^2}{(1 - \gamma_{k-1}(1)f_{k-1}(1))^2}, \quad (33)$$

ここで  $B_{k-1} = E[\tau_{k-1}] - \gamma'_{k-1}(1)f_{k-1}(1) - \gamma_{k-1}(1)f'_{k-1}(1)$  である。また、 $E[\tau_{k-1}]$  は (30) より計算される。(25) を微分すると、 $V[\tau_{k-1}]$  は同様に次のように計算される。

$$V[\tau_{k-1}] = \frac{V[T_{k-2}] + V[A_{k-1}]E[T_{k-2}]^3}{(1 - E[A_{k-1}]E[T_{k-2}])^3}. \quad (34)$$

その結果、 $V[T_{k-1}]f'_{k-1}(1) = E[\tau_{k-1}] - \tilde{a}_{k-1}(0)\tilde{g}'_{k-2}(\tilde{a}_{k-1}(0))$  となるので  $E[T_{k-2}]$ ,  $V[T_{k-2}]$ ,  $\tilde{g}_{k-2}(\tilde{a}_{k-1}(0))$  及び  $\tilde{g}'_{k-2}(\tilde{a}_{k-1}(0))$  より算出される。よって、 $\tilde{g}_\ell(\tilde{a}_{\ell+1}(0))$  と  $\tilde{g}'_\ell(\tilde{a}_{\ell+1}(0))$   $\ell = 0, 1, \dots, k-2$  がわかっているれば  $V[T_{k-1}]$  を計算できる。これらを得るために、下にある再帰的計算アルゴリズムを用いる。

$k \geq 3$  を仮定する。それ以外するとき、計算は明らかである。 $\ell = k-2$  とおき、 $z = \tilde{a}_{\ell+1}(0)$  に対して  $\tilde{g}_\ell(z)$  及び  $\tilde{g}'_\ell(z)$  を考える。それらには  $\tilde{\tau}_\ell(z)$ ,  $\tilde{\tau}'_\ell(z)$ ,  $\tilde{g}_{\ell-1}(z)$  及び  $\tilde{g}'_{\ell-1}(z)$  が必要となる。 $\tilde{\tau}_\ell(z)$  はアルゴリズム 1 によって  $\tilde{g}_{\ell-1}(z)$  から計算され、 $\tilde{g}_{\ell-1}(z)$  が必要である。一方、(25) を微分すると、

$$\tilde{\tau}'_\ell(z) = \frac{\tilde{g}'_{\ell-1}(z\tilde{a}_\ell(\tilde{\tau}_\ell(z))) \cdot \tilde{a}_\ell(\tilde{\tau}_\ell(z))}{1 - z\tilde{g}'_{\ell-1}(z\tilde{a}_\ell(\tilde{\tau}_\ell(z))) \cdot \tilde{a}'_\ell(\tilde{\tau}_\ell(z))}.$$

よって、 $\tilde{\tau}'_\ell(z)$  は  $\tilde{\tau}_\ell(z)$  及び  $\tilde{g}'_{\ell-1}(z\tilde{a}_\ell(\tilde{\tau}_\ell(z)))$  から算出される。よって、 $\tilde{g}_\ell(z)$  及び  $\tilde{g}'_\ell(z)$  が  $\tilde{g}_{\ell-1}(z)$  と  $\tilde{g}'_{\ell-1}(z)$  から計算される。この方法で、 $\ell$  が  $\tilde{g}_0(z) = z$  を式に持つ 1 になるまで再帰的に計算過程が呼び出される。

### 3.5 数値計算例

系 3.1 とその後与えたアルゴリズムによって  $E[X_k^-]$  の平均を計算する。最初にシミュレーション結果との一致性をみる。もちろん、そのような比較は数学的には重要でないが数学的記述の誤りの発見等に役に立つ。表 1 は  $c = 3$  の場合でレベル  $k$  には率  $\lambda_k$  のポアソン過程に従って到着する。 $A_k(n, n+1)$  は平均  $\lambda_k$  のポアソン分布を持つ。ここでシミュレーション実行時間は  $10^8$  である。ここで、"the 95% interval" は 95% 信頼区間を意味している。これらの結果から十分な正確性が言えると考えられる。

図 8 で示す数値計算例は各レベルでの平均待ち人数がどのようにになるかを示したものである。 $c = 3, 4, 5, 6$ ,  $N = 6$  そしてポアソン到着とする。結果より、平均到着率が十分に小さいと待ち行列の長さも小さくできることがわかる。再帰計算を行うに要する時間は  $k = 5$  までは結果はすぐに得られる。しかし、 $c = 6$  及び  $k = 6$  の場合は 40 秒かかる。これらの計算は C 言語を用いてコンピュータは 2.8 GHz Pentium D CPU のもので行った。

表 1:  $c = 3, N = 6$  ポアソン過程でのシミュレーションとの比較

Level $k$	1	2	3	4	5	6
$\lambda_k$	0.05	0.06	0.04	0.08	0.07	0.1
Numerical	0.051316	0.097791	0.110680	0.367563	0.792541	6.105358
Simulation	0.051332	0.097876	0.110809	0.367630	0.793398	6.133683
95% interval	$4.4 \cdot 10^{-5}$	$6.2 \cdot 10^{-5}$	$7.6 \cdot 10^{-5}$	$1.5 \cdot 10^{-4}$	$2.5 \cdot 10^{-4}$	$7.2 \cdot 10^{-4}$

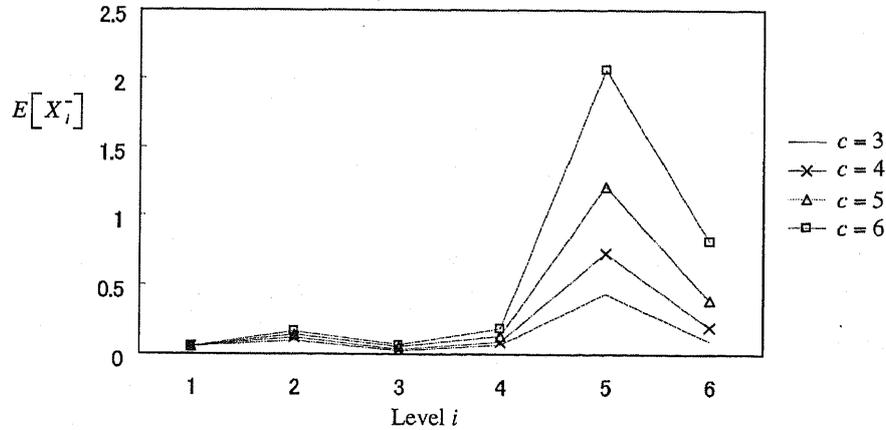


図 8:  $\lambda_k = 0.05, 0.06, 0.01, 0.02, 0.1, 0.01$  for  $k = 1, 2, \dots, 6$

### 3.6 まとめ

ここでは待ち行列の長さの平均を求める計算方法を提案した. これは第一の目標が解析的結果を求めることで, もちろん同様の方法で定理 3.1 を利用して, さらに高次のモーメントを計算することができる. レベル  $k$  の待ち行列の長さの  $m$  次モーメントは  $l = 1, 2, \dots, k-1$  及び  $n = 1, 2, \dots, m+1$  に対して  $\tilde{\tau}_l^{(n)}(z)$  の  $n$  回微分が必要となる. よって, さらに多くの再帰計算が必要になる.

得られた結果が複雑なため, 近似解での解析方法も考えられる. 例えば, 離散時間で行った時間間隔や待ち行列の長さの単位を変更することで流体モデルでの解析も考えられる. そのとき, 連続時間型及び連続領域での優先権付き待ち行列となる. 先行研究として [50] があるが, 標準的な優先権ルールで行われている. もし述べたような連続余裕空間条件のような複雑な優先権ルールをいれると, 定常的解析は困難に考えられる. 領域での流体近似は [34] で述べられている. また流体モデルにたくさんの有効な手段が見つけられる [57]. 本論文のモデル解析がそのような研究に有用な情報を与えることを望む.

## 4 サプライチェーン・ネットワークにおける基本情報の算出

日本の成長をになってきた要素のひとつに生産システムがある。一方で、それを支えてきた物流については未開拓の部分が多く、今後の一層の合理化と生産性の向上が求められている。現在物流産業はGDPの8%、40兆円の規模に達していて、この付加価値が寄与する部分は大きい。

従来の物流合理化は様々な問題に対して個別に対応してきたが、サプライチェーンの国際化と製品ライフサイクルの短期化などから、サプライチェーン全体の設計と運用が求められている。

前章で述べた待ち行列理論のネットワーク解析モデルとして有名なものにジャクソンネットワークがある。待ち行列理論とサプライチェーンを結びつけると客の到着が発注したものの到着であり、サービス時間がある種のリードタイムに相当する。しかしサプライチェーンを待ち行列でモデル化するときのひとつの問題は待ち行列理論では発注が受動的になることである。つまり、実際のサプライチェーンでは配送センターまたは顧客が必要量を納期を指定して発注するが、待ち行列理論では発注が客の到着でランダムになり、納期や発注量の指定ができない。ここでは、平衡方程式から定常性を仮定して待ち行列理論とは別のアプローチをしてサプライチェーン・ネットワークをモデル化する。

### 4.1 サプライチェーン・マネジメントの概要

資材調達・生産・物流・販売までの全ビジネスプロセスは、企業内外の組織を超えてグローバルに連鎖している。連鎖する業務間の連携のムダを最小化することが、全体最適に必須である。サプライチェーン（供給連鎖）マネジメント（以下SCM）とは、分散した全ビジネスプロセスで情報技術を取り込んで情報を共有し、連鎖業務の連携をマネジメントし、顧客満足度につながる価値を生まない物理的・時間的ムダを省き、企業利益を最大化するための考え方と方法論である。

一般的な流通経路はメーカー、工場から卸業者や配送センターで在庫を置き、小売業者に配送するという三段階のサプライチェーンである。独自のシステムとして米国のデル・コンピュータ社が考案したBTO(Build to Order)というデルモデルとして知られる注文方式やトヨタ自動車のかんばん方式に代表されるように在庫を持たないようにするJust In Time方式がある。

SCM手法は変遷は1960年代のMPR(Material Requirements Planning)に始まり1970年代のサプライヤーの生産と在庫を計画するために必要な機能を有しフィードバックするClosed-loop MRP、1990年代のERP(Enterprise Resource Planning)というものがあつた。これらのSCMの発展の源泉となっているものがSCP(Supply Chain Planning)で使われているアルゴリズムである。これらの中には、予測手法、制御手法、生産計画、在庫計画等が含まれており、さらに、知識ベースを用いた状態選択法、遺伝操作を適正変化させながら最適解を探索する遺伝アルゴリズム(GA)など、幾つかの計画手法が提案されるようになっている。本論文でも5.3でGAを用いたツールの作成、5.4で主双対内点法を用いたツールの開発を紹介する。

## 4.2 サプライチェーン・マネジメントの問題点

消費者の位置から離れるにつれて、消費者の需要変化の影響は大きくなる。消費者需要の変化に起因する影響は、消費者から初期生産者へ及ぶ。この影響の流れは、消費者から離れる程に大きくなる。これはブルウィップ効果と呼ばれている。販売機会損失を避けるために、小売店では品切れを防止するため実需を大きく上回って発注してしまい、多段階流通経路中の配送センターではまた販売機会損失を引き起こし、小売店に迷惑をかけてはいけないということで、メーカーには余裕を見て、多く発注しようということになる。工場では納期遅れを起こさないように必要以上に生産してしまう。この原因のひとつに多段階流通経路における情報の共有ができていないことがあげられる。そこで生産進捗情報の共有、メーカー在庫情報の共有、流通在庫情報の共有をシロスの少ないサプライチェーンを構築する必要がある [47][48]。プロダクトサイクルの短命化や短期間での顧客の購買行動の変化によって市場の不確実性が高まり、変化に従ったサプライチェーン・ネットワークの素早い構築が求められる。サプライチェーンを改善するソフトウェアのひとつにネットワーク最適化計算ツール (iLAND)、株式会社フレームワークス、がある。これは輸送コストを拠点から詳細に計算し最適なサプライチェーン・ネットワークを提供する。しかし拠点の在庫コストについては一定数を加算していることから明確な最適化とはなっていない。これは各拠点の安全在庫量や在庫量分散といった情報をネットワーク設計時に算出する方法がないからである。配送に関する基本情報と在庫に関する基本情報をネットワーク情報から同時に算出できることでより精密なサプライチェーン・ネットワークを作ることが可能になる。

本論文では従来からあるメーカーの工場－配送センター（中間倉庫）－小売店という3段階の物流モデル ([13][47][49]) を拡張したモデルを考える。現在のサプライチェーンはグローバル化しておりリードタイムや中間在庫の経由数も変化に富んでいる。また同じ配送経路でも航空便やトラック配送など輸送方法に違いがあり、リードタイムや輸送にかかるコストも異なっている。現実モデルに対応するには柔軟なネットワーク設計が必要であり、ダミーノードを置いたネットワーク設計法を示す。次にネットワーク情報から配送と在庫の基本情報を同時に算出する方法を提案する。現在の各ノードの情報から定常性を仮定して基本情報を算出することで全ノードにおける関係が明らかになる。この情報から基本設計をし、市場の変化が起きる場合はノード間の関係が明らかになっているので即座に変化に対応した情報を提供できる。

## 4.3 サプライチェーン・ネットワークの構築

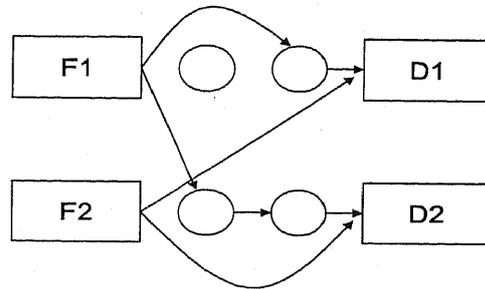
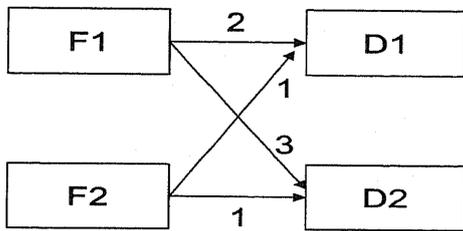
現在のサプライチェーンはグローバル化しておりリードタイムや中間在庫の経由数も変化に富んでいる。また同じ配送経路でも航空便やトラック配送など輸送方法に違いがありリードタイムや輸送にかかるコストも異なっている。現実モデルに対応するには柔軟なネットワーク設計が必要であり、ダミーノードを用いたネットワーク設計法を以下に提案する。今回は次の例でネットワーク構築を行う。工場3拠点 ( $F1, F2, F3$ )、配送センター3拠点 ( $D1, D2, D3$ )、小売店4拠点 ( $R1, R2, R3, R4$ ) の計10拠点がある。各拠点間のリードタイムは表のようになっている。

表 2: 各拠点間のリードタイム

	D1	D2	D3	R1	R2	R3	R4
F1	2	3	1	3	3	1	1
F2	1	1	3	1	1	3	3
F3	1	2	1	2	2	2	1
D1				1	1	3	2
D2				1	1	3	3
D3				3	3	1	1

空欄はパスがないことを示す. ここで工場-小売店では直送を許可する. またこの例のノード間の配送方法は1区間1方法とする.

各ノード間でリードタイムが異なる場合はダミーノードを置きリードタイムを分割する. [13]で行っている解析法は複数の工場や配送センターを持つ場合, モデル解析が困難になる. その原因の1つにリードタイムが複数存在する場合があげられる. ダミーノードを置き, リードタイムが単一になるようにする.



リードタイムをすべて1に分割

図 9:  $F1, F2, D1, D2$  間リードタイム 図 10: ダミーノードを用いたトポロジー

図 9 ではリードタイムが 3, 2 と 1 で混在している. ノード間にダミーノードを置き今回はすべてリードタイムを 1 にしている (図 10). また直送を行う場合, 直送用ダミーノードを置く (図 11,12).

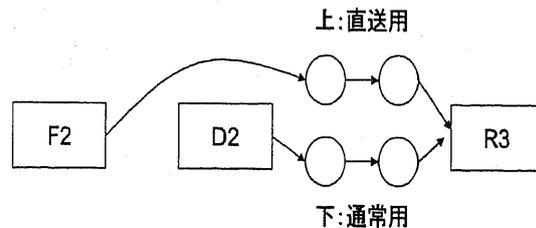
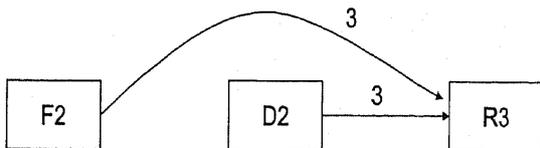


図 11:  $F2, D2, R3$  間のリードタイム 図 12: 直送ダミーノードを用いたトポロジー

各拠点間の配送方法は複数あってもよい。F1-D1間で自動車配送と船での配送がある場合にはダミーノードを種類ごと用意する(図13)。ノード間で配送方法が複数ある場合は種類分けしたダミーノードをおくことで対応できる。

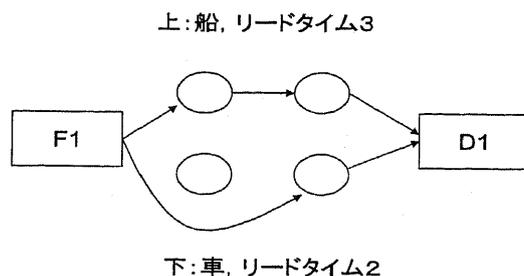


図 13: 複数配送手段がある場合のモデル作成方法

ダミーノードを利用し例のネットワークを設計すると以下のようになる(図14)。各ノード間はリードタイム分だけダミーノードを経由することになる。

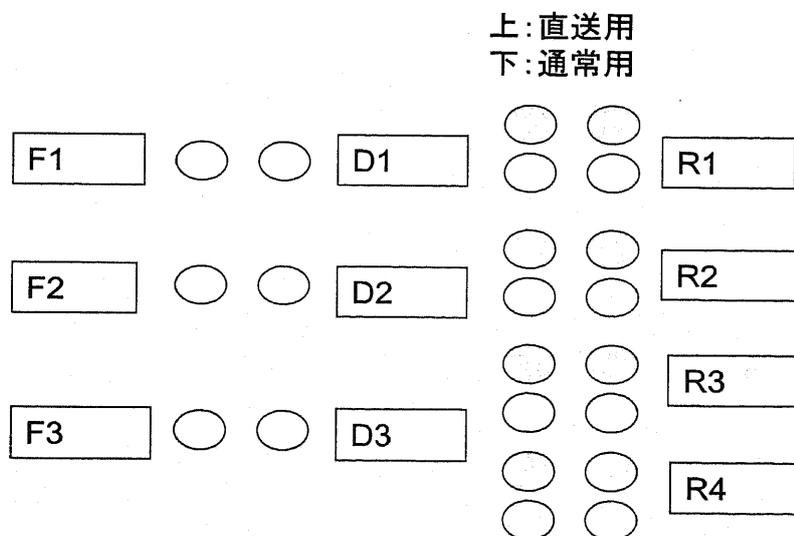


図 14: ダミーノードを利用したトポロジー

## 4.4 サプライチェーン・ネットワークの定義及び計算

### 4.4.1 変数の定義

以下に多段階ネットワーク型サプライチェーンのモデル化に必要な変数, 定数を定義する. ここでモデルはリードタイムの1時間をモデルの1単位時間とする離散モデルとして記述する. レベルとは各ノードの相対的位置を示したもので, 1から始まり  $W$  までとする. 各ノードはいずれかのレベルに所属し, 各レベルに属するノードは複数あってもよい. レベル  $k$  のノード数を  $M^{(k)}$  であらわす. 例えば各レベル間でリードタイムが全て一定な3段階モデルでは1レベルが工場, 2レベルが配送センター, 3レベルは小売店に相当する. ノードは自分が所属するレベルで一意的番号を持ちレベル番号とあわせて  $k$  レベルの  $i$  番目ならば  $(k, i)$  と表記する.

- $S_I^{(k,i)}(t)$  :  $k$  レベル  $i$  番目の時刻  $t$  直前のノードの在庫量
- $S_d^{(k,i)}(t)$  :  $k$  レベル  $i$  番目の時刻  $t$  直前のノードの発注量
- $T_S^{(k,i)}(t)$  :  $k$  レベル  $i$  番目の時刻  $t$  直前のノードの総配送量
- $L^{(k)}$  :  $k$  レベルへ配送する時のリードタイム
- $S^{(k,i)}$  :  $k$  レベル  $i$  番目のノードの安全在庫
- $p_{(k,i),(l,j)}$  :  $l$  レベル  $j$  番目のノードから  $k$  レベル  $i$  番目ノードへの配送率 ( $l \geq k$ )

ここで

$$\sum_{(k,i) \in I_{(l,j)}} p_{(k,i),(l,j)} = 1$$

$I_{(l,j)}$  は  $(l, j)$  への配送元集合である. 配送元とは  $p_{(k,i),(l,j)} > 0$  となるレベル  $k$  の  $i$  番目のノードである. また本文中で  $W$  レベル方向を下位レベル, 1 レベル方向を上位レベルと呼ぶ.

### 4.4.2 平衡方程式

**仮定 1** 多段階ネットワーク型サプライチェーンの関係式は以下のとおりである. 以下は  $1 \leq k \leq W, 1 \leq i \leq M^{(k)}$  とする.

$$S_I^{(k,i)}(t) = S_I^{(k,i)}(t-1) + S_d^{(k,i)}(t - L^{(k)} - 1) - T_S^{(k,i)}(t), \quad (35)$$

$$S_d^{(k,i)}(t) = (L^{(k)} + 1)E[T_S^{(k,i)}(t)] + S^{(k,i)} - \sum_{l=1}^{L^{(k)}} S_d^{(k,i)}(t-l) - S_I^{(k,i)}(t). \quad (36)$$

$$T_S^{(k,i)}(t) = \sum_{v=k}^W \sum_{l=1}^{M^{(v)}} p_{(k,i),(v,l)} S_d^{(v,l)}(t) \quad (1 \leq k \leq W-1), \quad (37)$$

$$T_S^{(W,i)}(t) = \sum_{l=1}^{M^{(W)}} p_{(W,i),(W,l)} S_d^{(W,l)}(t) + d^{(i)}(t) \quad (k = W). \quad (38)$$

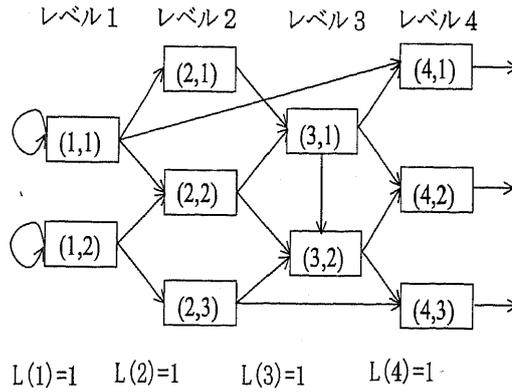


図 15: 多段階ネットワーク型サプライチェーン

ここで  $d^{(i)}$  は 0 以上の実数値をとる  $W$  レベル  $i$  番目のノードの時刻  $t$  における外部への配送量とする.  $d^{(i)}(t)$  は平均  $\mu_R^{(i)}$ , 標準偏差  $\sigma_R^{(i)}$  を持つ分布に従う変数とする. またレベル 1 への配送は, 工場での生産とみなせる. ここでは生産はリードタイムに従い生産量が十分あるとする.

(36) で  $(L^{(k)} + 1)E[T_S^{(k,i)}(t)] + S^{(k,i)}$  は定数であるので, 計算を簡単にするために変換をしておく.

$$S_d^{(k,i)}(t) = \hat{S}_d^{(k,i)}(t) - \gamma^{(k)}. \quad (39)$$

(36) に代入して計算し定数部分を 0 とおくと

$$\hat{S}_d^{(k,i)}(t) = - \sum_{l=1}^{L^{(k)}} \hat{S}_d^{(k,i)}(t-l) - S_I^{(k,i)}(t), \quad (40)$$

$$\gamma^{(k)} = - \frac{S^{(k)}}{1 + L^{(k)}} - E[T_S^{(k,i)}]. \quad (41)$$

#### 4.4.3 行列化

各ノードの関係式を行列化する. 各ノードの方程式を種類別, レベルごとに行列化することで各レベルのノード数を意識せず計算が可能になる. 表現を簡略化するために以下を利用する.

$$X^{(k)} = \begin{bmatrix} X^{(k,1)} \\ \vdots \\ X^{(k,M^{(k)})} \end{bmatrix}.$$

$$1 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}.$$

$k$  レベルでは  $M^{(k)}$  個の縦ベクトルをあらわす.

また (37) を行列化するために次のベクトル  $F_d^{(k)}$  を定義する. つまり  $F_d^{(k)}$  は  $k$  レベル以下の各ノードの発注量から成るベクトルである.

$$\begin{aligned} F_d^{(W)} &= \{S_d^{(W,1)}, \dots, S_d^{(W,M^{(W)})}\}, \\ F_d^{(W-1)} &= (F_d^{(W)} | S_d^{(W-1,1)}, \dots, S_d^{(W-1,M^{(W-1)})}), \\ &\vdots \\ F_d^{(1)} &= (F_d^{(2)} | S_d^{(1,1)}, \dots, S_d^{(1,M^{(1)})}). \end{aligned}$$

分配確率行列  $P$  を定義する.

$$P^{(k)} = \begin{bmatrix} P_{(k,1),(k,1)} & \cdots & P_{(k,1),(W,M^{(W)})} \\ \vdots & \ddots & \vdots \\ P_{(k,M^{(k)}),(k,1)} & \cdots & P_{(k,M^{(k)}),(W,M^{(W)})} \end{bmatrix}.$$

これを用いると式 (37) と (38) は次のようにかける.

$$T_S^{(k)}(t) = P^{(k)} F_d^{(k)}(t), \quad (42)$$

補題 4.1 (35), (40) と (37) を上記の表現を利用して行列化すると次を得る.

$$S_I^{(k)}(t) = S_I^{(k)}(t-1) + S_d^{(k)}(t-L^{(k)}-1) - T_S^{(k)}(t), \quad (43)$$

$$\hat{S}_d^{(k)}(t) = -\tilde{\hat{S}}_d^{(k)}(t-1) - S_I^{(k)}(t), \quad (44)$$

$$S_d^{(k)}(t) = \hat{S}_d^{(k)}(t) - \gamma^{(k)} 1. \quad (45)$$

ここで

$$\tilde{\hat{S}}_d^{(k)}(t-1) = \begin{bmatrix} \hat{S}_d^{(k,1)}(t-L^{(k)}) & \cdots & \hat{S}_d^{(k,1)}(t-1) \\ \vdots & \ddots & \vdots \\ \hat{S}_d^{(k,M^{(k)})}(t-L^{(k)}) & \cdots & \hat{S}_d^{(k,M^{(k)})}(t-1) \end{bmatrix}.$$

証明. 式 (43) と式 (45) はそれぞれ式 (35) と (39) から明らか. また式 (40) は次のようにかける.

$$\begin{bmatrix} \hat{S}_d^{(k,1)}(t) \\ \vdots \\ \hat{S}_d^{(k,M^{(k)})}(t) \end{bmatrix} = - \begin{bmatrix} \hat{S}_d^{(k,1)}(t-L^{(k)}) \\ \vdots \\ \hat{S}_d^{(k,M^{(k)})}(t-L^{(k)}) \end{bmatrix} - \cdots - \begin{bmatrix} \hat{S}_d^{(k,1)}(t-1) \\ \vdots \\ \hat{S}_d^{(k,M^{(k)})}(t-1) \end{bmatrix} - \begin{bmatrix} S_I^{(k,1)}(t) \\ \vdots \\ S_I^{(k,M^{(k)})}(t) \end{bmatrix}$$

よってこれをまとめると式 (44) が得られる.  $\square$

#### 4.4.4 解の算出

前章で得られた方程式から、定常性を仮定してZ変換を利用し解を求めていく。ここで、このシステムが定常性を持つことは各ノードの総配送量  $T_S^{(k,i)}$  が存在しなければならない。つまり  $T_S^{(k,i)}$  は配送率  $p^{(k,i),(l,j)}$  と需要をあらわす  $d^{(i)}(t)$  から構成されるものであるので、 $d^{(i)}(t)$  が有限の平均及び分散をもつことが定常性の条件になる。Z変換から次の補題を得る。

#### 補題 4.2

$$S_I^{(k)}(z) = \frac{z^{-(L^{(k)+1)}S_d^{(k)}(z) - T_S^{(k)}(z)}{1 - z^{-1}}, \quad (46)$$

$$\hat{S}_d^{(k)}(z) = -\frac{1 - z^{-1}}{1 - z^{-(L^{(k)+1)}}S_I^{(k)}(z), \quad (47)$$

$$T_S^{(k)}(z) = P^{(k)}F_d^{(k)}(z), \quad (48)$$

$$S_d^{(k)}(z) = \hat{S}_d^{(k)}(z) - \frac{1}{1 - z^{-1}}\gamma 1. \quad (49)$$

証明. (43),(44),(42) と (45) を Z 変換で計算すると

$$S_I^{(k)}(z) = z^{-1}S_I^{(k)}(z) + z^{-(L^{(k)+1)}S_d^{(k)}(z) - T_S^{(k)}(z),$$

$$\hat{S}_d^{(k)}(z) = -Z(\tilde{\hat{S}}_d^{(k)}(t-1)) - S_I^{(k)}(z),$$

$$T_S^{(k)}(z) = P^{(k)}F_d^{(k)}(z).$$

(48) は明らか. ここで  $Z(\tilde{\hat{S}}_d^{(k)}(t-1))$  は

$$\begin{aligned} Z(\tilde{\hat{S}}_d^{(k)}(t-1)) &= \sum_{u=1}^{L^{(k)}} z^{-u}\hat{S}_d^{(k)}(z) \\ &= \frac{z^{-1}(1 - z^{-L^{(k)}})}{1 - z^{-1}}\hat{S}_d^{(k)}(z). \end{aligned}$$

これより (46),(47),(49) が得られる.

□

#### 4.4.5 基本情報の算出

Z変換で得られた方程式から逆変換から解を求め、基本情報として平均と分散を算出する。

定理 4.1 定常性の仮定の下で  $S_I^{(k)}(t), S_d^{(k)}(t)$  が得られる。

$$S_I^{(k)}(t) = -\sum_{l=0}^{L^{(k)}} T_S^{(k)}(t-l) + S^{(k)} + (1 + L^{(k)})E[T_S^{(k)}], \quad (50)$$

$$S_d^{(k)}(t) = T_S^{(k)}(t) + \left\{ \frac{S^{(k)}}{1 + L^{(k)}} + E[T_S^{(k)}] \right\} 1_{(t \leq L^{(k)})}. \quad (51)$$

ただし

$$\begin{aligned} S_I^{(k)}(0) &= S^{(k)} + (1 + L^{(k)})E[T_S^{(k)}], \\ S_d^{(k)}(u) &= 0 \quad (-L^{(k)} - 1 \leq u \leq 0). \end{aligned}$$

とする.

証明.

(46),(47),(49) より  $S_I^{(k)}(z)$  と  $S_d^{(k)}(z)$  が得られる.

$$S_I^{(k)}(z) = -\frac{1 - z^{-(L^{(k)}+1)}}{1 - z^{-1}} \times (T_S^{(k)}(z) + \frac{z^{-(L^{(k)}+1)}}{1 - z^{-1}} \gamma I), \quad (52)$$

$$S_d^{(k)}(z) = T_S^{(k)}(z) - \frac{1 - z^{-(L^{(k)}+1)}}{1 - z^{-1}} \gamma I. \quad (53)$$

(52) の第 2 項目を留数定理を用いて計算すると

$$\begin{aligned} & \frac{1}{(2-1)!} \frac{d}{dz} \left[ \frac{(z^{L^{(k)}+1} - 1)z^{t-2L^{(k)}-1}}{(z-1)^2} (z-1)^2 \right] \Big|_{z=1} \\ &= \frac{d}{dz} (z^{t-L^{(k)}} - z^{t-2L^{(k)}-1}) \Big|_{z=1} \\ &= L^{(k)} + 1. \end{aligned}$$

$\gamma$  を代入すると

$$\begin{aligned} S_I^{(k)}(t) &= -\sum_{l=0}^{L^{(k)}} T_S^{(k)}(l) - (L^{(k)} + 1)\gamma I \\ &= -\sum_{l=0}^{L^{(k)}} T_S^{(k)}(l) + S^{(k)} + (1 + L^{(k)})E[T_S^{(k)}]. \end{aligned}$$

また (51) も (53) から同様に求まる. □

系 4.1 上の式より基本情報を求める. 定常性を仮定し  $t \rightarrow \infty$  の条件で求めると

$$E[S_I^{(k)}] = S^{(k)}, \quad (54)$$

$$E[S_d^{(k)}] = E[T_S^{(k)}], \quad (55)$$

$$V[S_I^{(k)}] = (L^{(k)} + 1)V[T_S^{(k)}], \quad (56)$$

$$V[S_d^{(k)}] = V[T_S^{(k)}]. \quad (57)$$

つまり基本情報は配送率,  $W$  レベルの平均需要量と需要量標準偏差で決定される. ここで系 1 の各式を計算するとき  $W$  レベルの情報で表す必要がある. つまり全ての配送パスを考慮したときの  $(k, i)$  ノードから  $(W, j)$  ノードへの配送率  $p^{W(k,i),(W,j)}$  を算出する.  $W$  レベル  $\rightarrow$  1 レベルの順で計算する.

$$pw_{(k,i),(W,j)} = \sum_{u=k}^W \sum_{v=1}^{M(u)} p_{(k,i),(u,v)} pw_{(u,v),(W,j)} + p_{(k,i),(W,j)} \quad (58)$$

式(58)を用いると、式(37)は次のようにかける。

$$T_S^{(k,i)}(t) = \sum_{l=1}^{M(W)} pw_{(k,i),(W,l)} d^{(l)}(t) \quad (1 \leq k \leq W) \quad (59)$$

式(59)は  $W$  レベルの外部への需要変数  $d^{(i)}$  とノード  $(k, i)$  から  $W$  レベルへの配送率によって書き表すことができる。各ノードの総配送量の平均と分散は次のようになる。

$$E[T_S^{(k,i)}] = \sum_{l=1}^{M(W)} pw_{(k,i),(W,l)} \mu_R^{(l)} \quad (60)$$

$$V[T_S^{(k,i)}] = \sum_{l=1}^{M(W)} pw_{(k,i),(W,l)}^2 (\sigma_R^{(l)})^2 \quad (61)$$

$S^{(k,i)}$  は  $S^{(k,i)} = D \times \sqrt{L^{(k)} + 1} \times \sqrt{V[T_S^{(k,i)}]}$  の式で与えられる。  $D$  は定数である。つまり  $S^{(k,i)}$  は  $V[T_S^{(k,i)}]$  によって決まる。系4.1の各ノードの基本情報は次のようにかける。ここで、 $\mu_R^{(i)}$ 、 $\sigma_R^{(i)}$  はそれぞれ  $W$  レベル  $i$  番目のノードにおける外部への需要量平均と需要量標準偏差である。

$$\begin{aligned} E[S_I^{(k,i)}] &= D \times \sqrt{L^{(k)} + 1} \times \sqrt{V[T_S^{(k,i)}]} \\ &= D \times \sqrt{L^{(k)} + 1} \times \sqrt{\sum_{l=1}^{M(W)} pw_{(k,i),(W,l)}^2 (\sigma_R^{(l)})^2}, \end{aligned} \quad (62)$$

$$E[S_d^{(k,i)}] = \sum_{l=1}^{M(W)} pw_{(k,i),(W,l)} \mu_R^{(l)}, \quad (63)$$

$$V[S_I^{(k,i)}] = (L^{(k)} + 1) \left( \sum_{l=1}^{M(W)} pw_{(k,i),(W,l)}^2 (\sigma_R^{(l)})^2 \right), \quad (64)$$

$$V[S_d^{(k)}] = \sum_{l=1}^{M(W)} pw_{(k,i),(W,l)}^2 (\sigma_R^{(l)})^2. \quad (65)$$

#### 4.5 数値計算例

ここでは4.3節で紹介したサプライチェーンモデルで以下の仮定で数値計算を行う。

- 拠点は工場 3 拠点 ( $F1, F2, F3$ ), 配送センター 3 拠点 ( $D1, D2, D3$ ), 顧客 4 拠点 ( $R1, R2, R3, R4$ ) である。
- 工場-顧客間の直送は可能。
- 工場-配送センタ間, 配送センタ-顧客間, 工場-顧客間の配送方法は 1 方法とする。
- 各ノードの安全在庫は  $S^{(k,i)} = D \times \sqrt{L^{(k)} + 1} \times \sqrt{V[T_s^{(k,i)}]}$  で算出する。

各拠点間のリードタイムは表 2 で与えたものとする。このリードタイムを考慮し、4.3 節で紹介した方法でネットワークの構築を行うと図 16 のようになる。顧客の需要平均と需要標準偏差は表 3 で与えられる。また、各ノード間の配送率は付録の表 5, 6, 7 で与える。

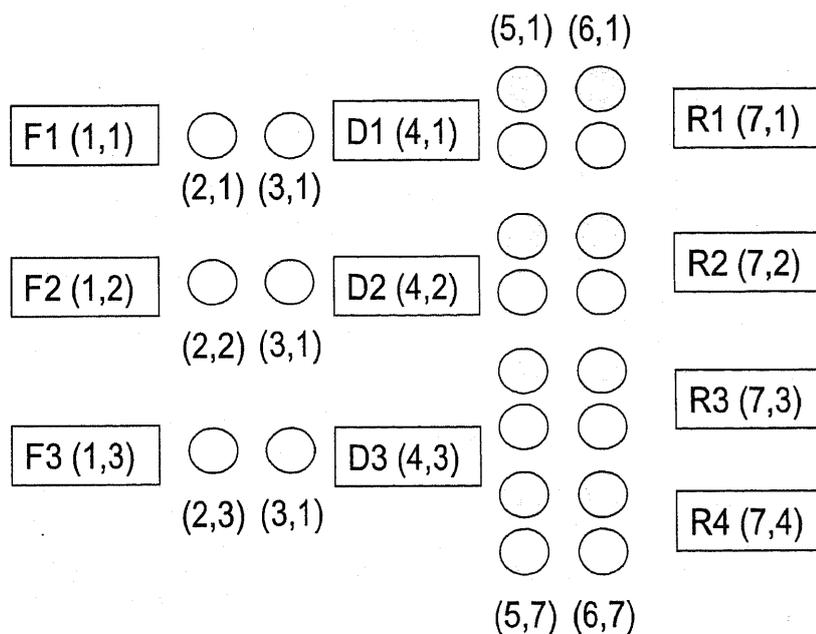


図 16: ダミーノードを利用したネットワークとノード番号

表 3: 顧客の需要平均, 需要偏差

	$R1$	$R2$	$R3$	$R4$
需要平均	150	90	120	90
需要偏差	10	10	10	10

以上のパラメータから  $D = 1.65$  とおき計算すると表 4 となる。

表 4: 計算結果

	<i>F1</i>	<i>F2</i>	<i>F3</i>	<i>D1</i>	<i>D2</i>	<i>D3</i>	<i>R1</i>	<i>R2</i>	<i>R3</i>	<i>R4</i>
配送量平均	123.22	241.67	85.11	55.72	94.82	66.39	150.00	90.00	120.00	90.00
配送量分散	44.11	127.46	17.29	7.79	20.24	14.46	100.00	100.00	100.00	100.00
安全在庫	15.50	26.34	9.70	7.30	14.24	12.58	29.60	30.71	36.37	35.13

#### 4.6 まとめ

現実のサプライチェーンをダミーノードを用いて柔軟にモデル化する方法を提案し、平衡方程式から定常性を仮定しサプライチェーン・ネットワークの各ノードの基本情報を算出した。この情報はサプライチェーンの初期設計時に有効な指標となる。また市場の動向が変化した場合や災害時にある区間の輸送手段が停止した場合、配送率を変化させて計算することでネットワークの変化をみてとれる。また方程式の解として求めているので必要な情報を与えればシミュレーションを行うことなく即座に情報を得られる。4.3節でも複数の配送方法にモデルが対応していると述べたが、この手法を用いれば、各配送手段にコストをかけることで全体のコスト算出ができその値から最適なサプライチェーン・ネットワークの構築が期待できる。本論文で述べたモデル構築の柔軟性や算出した基本情報の適用は今後さらにグローバル化していっくだろうサプライチェーン・ネットワークに有用な指標を与えると考えられる。

## 5 輸送ネットワークと安全在庫配置の同時改善解の算出とシステム構築

政治や経済など社会の様々な場面や、あるいは単純に個人での判断において、『どれが一番自分にとって都合が良いだろうか?』といった最も都合のよい意思決定を必要とする場面に直面することがある。このような問題解決手法のことを一般的に「最適化(Optimization)」と呼び、「最適化を行う(Optimize)」と言う。専門的に説明すると、最適化とは『与えられた規準(criterion)の下で、対象とする問題の最適な解決策(solution)を求めること』と考えられ、そのための理論(定式化)と手法(アルゴリズム)を研究し、その応用を与えることが目標とされている。従って、その理論的側面においては、なぜ最適解なのか、どのような性質を持つ最適解なのかなどの最適性の条件に代表される解の定性的性質の総合的かつ数学的考察が行われる必要がある。また、アルゴリズム的側面においては、最適解をコンピュータを用いて数値的に計算するための実用的なアルゴリズムが開発され、その有効性の理論的考察が行われている。このような研究分野を数理計画(Mathematical Programming)と呼ぶ。さらに、応用的側面においては、現実問題を数学的モデルとして定式化してそれを解くことにより意思決定を行うための指針を与えることを目指して研究が行われている。このようなアプローチをオペレーションズ・リサーチ(Operations Research, 通称OR)とも呼び、今日においては、より現実的な問題を解決するために、他の問題解決手法(人工知能, エキスパートシステム, システム理論, ファジィ集合, ニューロ, 遺伝アルゴリズム, シミュレーテッド・アニーリング法などの様々なパラダイム)と結びつき、より複雑な最適化問題とその解法が提案されている。

一般的に最適化問題は数理的に与えて、数理的な方法で解く。具体的には、 $n$ 次元の実数ベクトル  $x$  ( $n$  個の数字の組: 変数) で、与えられた不等式・論理式など条件を全て満たすものの中で、目的関数を最大または最小にするようなものを求める。実社会に存在する現象をモデル化し数値として表現することで数理計画法を適用でき、現在では様々なアルゴリズムによって最適解を算出できる。数理計画法の中でも、線形計画法(LP: Linear Programming)は早くから実用化され、広く社会に寄与していることはよく知られている。現在では数学モデルが非線形の場合も研究が進みニュートン法を基礎として、準ニュートン法や主双対内点法が広く使われるようになった。数学的アルゴリズムが提供されるにつれ、それを適用したソフトウェアの開発が重要である。本論文でも最適化アルゴリズムを適用したアプリケーション開発を試みる。

### 5.1 サプライチェーン・ネットワーク改善システム構築の意義

線形計画法は様々な問題に使われ、現在までにその技術は大きく進歩した。かつては数千の変数、数千の制約の線形計画法の問題を解くのは容易でなく、その計算費用も相当なものだった。現在ではごく普通のパソコンと優れたソフトウェアの組み合わせによりかなり大きな問題を解くこともできる。はじめて線形計画法が考えられてから実に190万倍もの速さで解けるようになったという[18]。この章では4章で求めたサプライチェーン・ネットワークの安全在庫量や発注量の基本情報を用いて輸送ネットワークと安全在庫配置

の同時改善解を算出するシステム (以下 SCN 改善システム) を開発する。

今回の SCN 改善システムを構築する意義を述べる。この研究は実務に対応できるシステムとツール作成が目標である。サプライチェーン・ネットワークを改善するような問題では次のことを前提としている。(i) タイミングにあった決定を行う, (ii) その決定が, より最適に近いと考えられること, (iii) 実際の結果は計算結果に一致しないが, 近づける努力を行う, という方法論である。つまり, 早く, 実用に耐えるレベルの解が算出でき, 意思決定を行える指標が必要である。実務者から見た場合, 数学的な完全な裏付けが得られていれば理想的だが, 多くの制約が存在し様々な環境変化がめまぐるしく発生するなかで, 迅速にその向かう方向性を判断できる材料が必要である。それは100%の精度ではなく70%や80%の精度でも有効な指標になる場合が多い。

SCN 改善システムは大きく分けると次の2つが考えられる。ひとつひとつ動くあらゆるイベントを把握し瞬時に計画と実績を監視して, 即座に反応する動的環境最適化システムと多くのイベントから中長期的にプロセスの見直しをかけていく静的環境最適化システムである。4章の結果は定常性を仮定しているので, 静的なシステムの情報である。これを改善システムに取り入れると静的条件下での改善となる。これを [17] 等の動的な最適化システムとリンクさせることで, 効果的なシステムが期待できる。

### 5.1.1 サプライチェーン・ネットワーク改善問題

まず, 準備として  $IC[k, i], TC[k, i]$  を定義する。それぞれノード  $(k, i)$  の在庫コスト, 配送コストとする。

$$IC[k, i] = S^{(k,i)} \times CIC[k, i] \quad (66)$$

$$TC[k, i] = \sum_{l=k}^W \sum_{j=1}^k p_{(k,i),(l,j)} E[T_S^{(l,j)}] \times CTC[(k, i), (l, j)] \quad (67)$$

ここで  $CIC[k, i]$  はノード  $(k, i)$  の在庫コスト単価,  $CTC[(k, i), (l, j)]$  はノード  $(k, i), (l, j)$  間の配送コスト単価である。次の問題を考える。

問題 1 次の目的関数

$$cost = \sum_{k=1}^W \sum_{i=1}^{M^{(k)}} (IC[k, i] + TC[k, i]) \quad (68)$$

を最小化せよ。

制約条件

$$\sum_{(k,i) \in I_{(l,j)}} p_{(k,i),(l,j)} = 1, \quad (69)$$

$$p_{(k,i),(l,j)} \geq 0, \quad (70)$$

$$p_{(k,i),(l,j)} = 0, \quad (k, i) \notin I_{(l,j)} \quad (71)$$

$$\sum_{(k,i) \in Id_{(l,j)}} pd_{(k,i),(l,j)} \leq D_{(l,j)} \quad (72)$$

とする。ネットワークにより、直送の制限が加わる場合がある。直送制限の式は (58) に指定されたレベル間 ( $k'$  レベルから  $W$  レベル) で計算する条件を加え、その配送率が指定された数値以下になるようにする。ここで  $Id_{(l,j)}$  はノード  $(l,j)$  への直送配送元集合であり  $pd_{(k,i),(l,j)}$  は式 (73) で与えられるノード  $(k,i)$  からノード  $(l,j)$  への直送配送率である。

$$pd_{(k,i),(W,j)} = \sum_{u=k'}^W \sum_{v=1}^{M^{(u)}} p_{(k,i),(u,v)} pd_{(u,v),(W,j)} \quad (73)$$

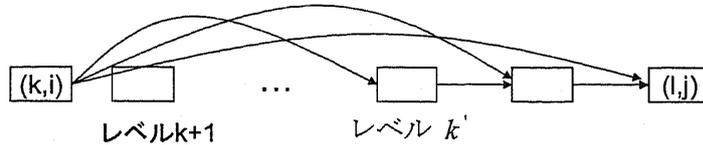


図 17: 直送計算

目的関数 (68) は全ノードの在庫コストと配送コストの和で構成されている。これまでは (i) 拠点選択と輸送量の配分を最適化するアプローチ, (ii) 拠点選択と在庫量の配分を最適化するアプローチ, の2つの方法で解析を行ってきた。この相反するようなトレードオフを一度に改善できる。

## 5.2 SCN 改善システムの構築

今回考えるネットワーク改善は実務的な要素を多く含み、多くのモデルに対応できる必要がある。ここではパラメータ取り込みから出力までをシステム化する。また最適化アルゴリズムが変更されても対応できるようにする。

**処理手順** 提案する SCN 改善システムのフローチャートを以下に示す。

- (i) ダミーノードを利用したネットワーク構築：4.3節で紹介した方法でリードタイムを分割したネットワークを構築する。
- (ii) パラメータの取り込み：配送コスト (表 14,15,16), 在庫コスト (表 8), 分割済みリードタイム (表 9), 小売需要平均 (表 10), 小売需要標準偏差 (表 11), 配送先トポロジー (表 17,18,19), 及び直送制限 (表 13) の各情報ファイルを取り込む。
- (iii) 目的関数を決定。
- (iv) 最適化アルゴリズムを用いてコスト計算を行う
- (v) 情報出力：最適な遺伝子で求めた情報を出力する (配送率, 配送量平均, 配送量分散, 安全在庫, 配送コスト, 在庫コスト, 総コスト)。
- (vi) 実モデルへの変換：ダミーノードを利用して求めた結果をレベル指定 (表 12) を使い元のモデルに対応させる。

今回問題で示した目的関数は例示的なものである。輸送ネットワーク安全在庫同時改善を行うには在庫コスト (66) と配送コスト (67) で目的関数を構成する必要があるが、必ず

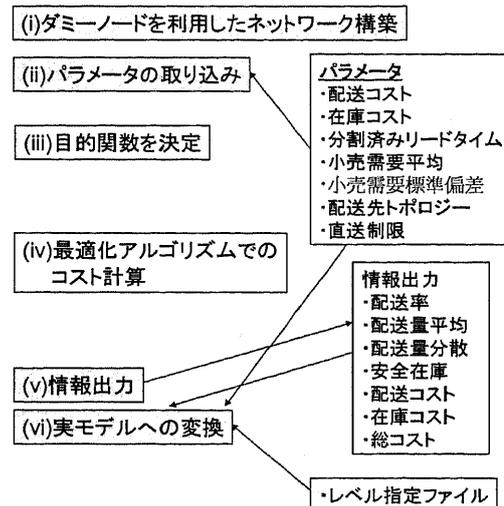


図 18: サプライチェーン・ネットワーク改善システム

しも (68) でないといけない訳ではない。目的関数は初期値あるいは現状値からの相対的な改善の度合いを示す尺度である。このシステムは目的関数の厳密性を問うのではなく、実用的に実務者にとって有効なものを目指す。実務者が改善したいシステムにあわせた目的関数を用意する必要がある。このシステムに対し、5.3 で GA を用いた方法、5.4 で主双対内点法を適用した方法を提案する。

### 5.3 GA を用いた SCN 改善ツールの開発

遺伝アルゴリズムとは、ダーウィンの理論にヒントを得た、適用分野を限定しない探索技術であり、最適解を目指し、効率的に、系統的に解空間を探索するために自然界の進化過程をシミュレートするものである。ある有利な特徴を持つ個体は、より長く生き延びることが可能で、その子孫にもその有利な特徴を伝えられるという、自然淘汰の理論にもとづいている。組合せ最適化問題を解くために遺伝アルゴリズムを用いるとき、解を染色体の形で、効率良く表現する方法をみつけなければならない。各染色体にはそれぞれ適応度が与えられ、これが最も高いものを持つ染色体が生き残るという自然界をシミュレートする。すると、世代が進むにつれて高い適応度を持つより良い個体が生まれてくることが期待できる [46]。

#### 5.3.1 遺伝アルゴリズムの構成と流れ

1. 初期染色体集団の生成  
乱数を利用し、様々な遺伝子を持った染色体の初期集団の生成を行う。
2. 適応度の計算  
各染色体に対して、その染色体が持っている遺伝子情報から得られた環境に対する度合いを表した値を計算したものである。

3. 両親の選択  
適応度に応じた染色体を染色体集団から選択する。次世代に良質な染色体を残すために適応度の高い染色体が選択されやすく、適応度の低い染色体は選択されにくい。
4. 交叉  
2つの親となる染色体の遺伝子を一部交換させ、新しく2つの子となる染色体を作ることである。
5. 突然変異  
染色体中のランダムに選ばれた遺伝子について、他の対立遺伝子に変更する。突然変異を行うと、親の世代では持ち合わせていない新しい特徴を手に入れることができ、その結果、解集団に多様性がでてくる。
6. 終了条件  
条件を満たすと世代の繰り返しが終了する。

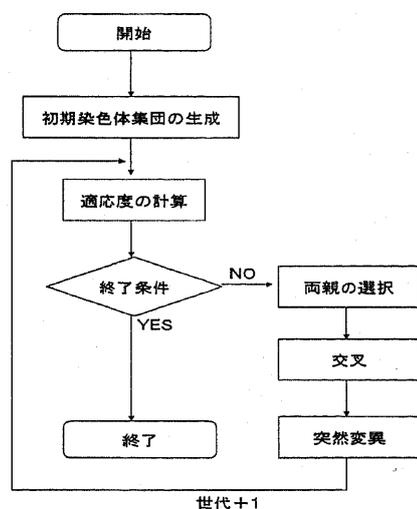


図 19: 遺伝アルゴリズムの流れ

### 5.3.2 GA を適用した SCN 改善システムの紹介

- 処理手順 提案する GA を適用した SCN 改善システムのフローチャートを以下に示す。
- (A) ダミーノードを利用したネットワーク構築：4.3 節で紹介した方法でリードタイムを分割したネットワークを構築する。
  - (B) パラメータの取り込み：配送コスト (表 14,15,16), 在庫コスト (表 8), 分割済みリードタイム (表 9), 小売需要平均 (表 10), 小売需要標準偏差 (表 11), 配送先トポロジー (表 17,18,19), 及び直送制限 (表 13) の各情報ファイルを取り込む。
  - (C) 目的関数決定。

- (D) 遺伝子初期解生成処理：配送率 (図 21) の初期値をランダムに生成する。直送制限を制約とする。
- (E) 各遺伝子のコスト評価：4.4.5 節で算出した基本情報を求め、コストを計算する。
- (F) 終了判定：あらかじめ決まっている終了条件を満たせば終了する。終了すれば (I) へ移動する。
- (G) 世代更新：(F) で終了しない場合、世代を更新する。
- (H) 交叉，突然変異：コストによって順位付けされた遺伝子を多点交叉法により交叉する。また決められた確率で突然変異が起こる。(E) に移り計算をする。
- (I) 情報出力：最適な遺伝子で求めた情報を出力する (配送率，配送量平均，配送量分散，安全在庫，配送コスト，在庫コスト，総コスト)。
- (J) 実モデルへの変換：ダミーノードを利用して求めた結果をレベル指定 (表 12) を使い元のモデルに対応させる。

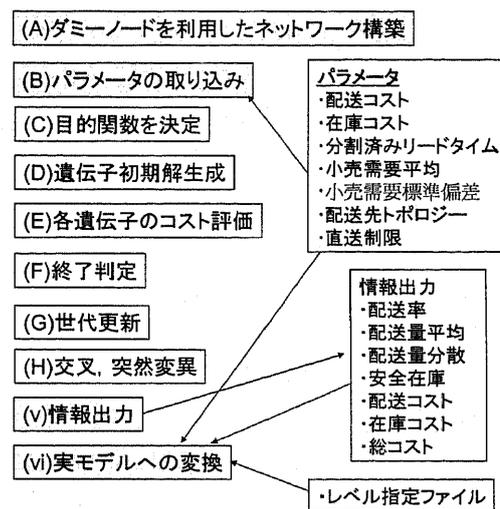


図 20: GA-SCN 改善システム

**遺伝子構成例** この問題の遺伝子構成を図 21 に示す。遺伝子は各ノード間の配送率を各レベルのノード番号の順に並べたものである。

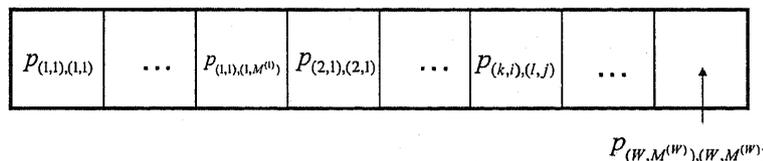


図 21: 遺伝子構成

**選択** 選択は生物の自然淘汰をモデル化したもので、適応度にもとづいて個体を増やしたり削除したりする操作である。選択のアルゴリズムには次のようなものがある。

- 1 ルーレット選択
- 2 ランキング選択
- 3 トーナメント選択

などがある。ルーレット選択は個体  $i$  を選ぶ確率を  $p_i$  と置いたとき、 $p_i = f_i / \sum_{k=1}^N f_k$  とする選択方式である。上記の式の  $f_i$  は個体  $i$  の適応度を表す。この方式はホランドが最初に提案したときに使われた選択方式であり、最も有名な選択方式であるが適応度が負の値を取らないことが前提になっている。また適応度が高いことが前提になっているため最小値を求める問題では使いづらい。さらに、もし個体間の適応度の格差が激しい場合は適応度の高い個体の選ばれる確率が非常に高くなり、初期収束の原因にもなる。このため、実際には適応度をスケールした値を使用することが多い。初期収束とは、最初の方の世代で偶然他の個体より適応度が圧倒的に高い個体が生まれたとき、その個体の遺伝子が集団中に爆発的に増えて探索がかなり早い段階で収束してしまう現象である。ルーレット選択の設定が甘い場合や、突然変異の効果が上手く表れないときに起こりやすい。

ランキング選択は各個体を適応度によってランク付けして、1位なら確率  $p_1$ 、2位なら確率  $p_2$ 、3位なら…というふうにランクごとにあらかじめ確率を決めておく方式である。この方法は、ルーレット選択と違い選択確率が適応度の格差に影響されない。しかし、これは逆に適応度にあまり差がない個体間でも選択確率に大きな差が生じる可能性がある。また、個体にランク付けをするため次世代が揃うたびにソートを行う必要がある。今回採用した選択法もこのランキング選択である。各遺伝子で式 (68) を計算後、並び替えを行う。

トーナメント選択はあらかじめ決めた数（トーナメントサイズという）だけ集団の中からランダムで個体を取り出し、その中で最も適応度の高い個体を選択する方式である。

**突然変異** 突然変異は生物に見られる遺伝子の突然変異をモデル化したもので、個体の遺伝子の一部を変化させる操作である。今回用いた突然変異の発生方法は次のとおりである。ここで遺伝子の正規化とは式 (74) を満たすようにすることである。

- 1 全遺伝子から定められた率で遺伝子を選び出す。
- 2 選び出された遺伝子の各要素を定められた率で突然変異を起こすか決定し、ランダムな値に変更する。
- 3 遺伝子を (74) によって正規化し新たな遺伝子  $p'_{(k,i),(l,j)}$  を決定する。

$$p'_{(k,i),(l,j)} = \frac{p_{(k,i),(l,j)}}{\sum_{(k,i) \in I_{(l,j)}} p_{(k,i),(l,j)}} \quad (74)$$

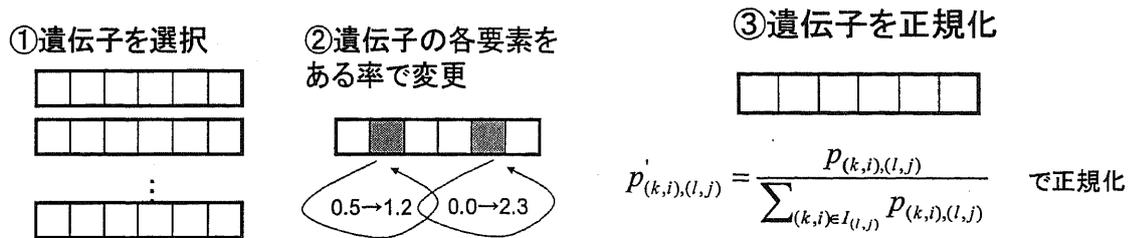


図 22: 突然変異

**交叉** 交叉は生物が交配によって子孫を残すことをモデル化したもので、個体の遺伝子の一部を入れ換える操作である。交叉はその性質上、最も重要な遺伝的操作とすることができる。今回行った交叉の方法は以下である。

- 1 全遺伝子からランダムに交叉を行うペアを選び出す。交叉するペアの数は定められた発生率に依る。
- 2 選出されたペアでランダムに要素を選び出し、交換する。交換する個数は定められた発生率に依る。
- 3 遺伝子を (74) によって正規化し新たな遺伝子  $p'_{(k,i),(l,j)}$  を決定する。

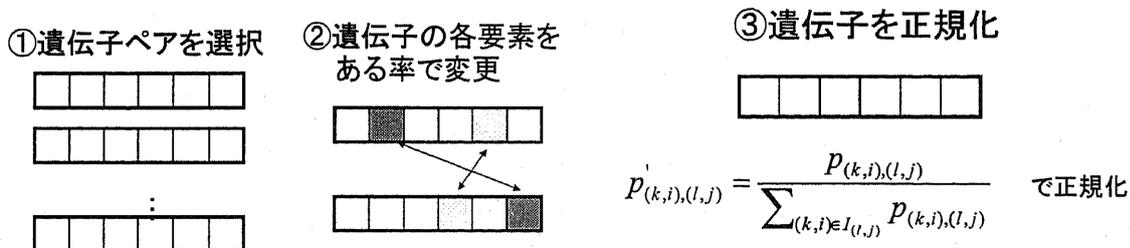


図 23: 交叉

### 5.3.3 GA での SCN 改善計算例

ここでは前述したモデルに対して計算した結果を述べる。最初は終了条件をあるコスト以下になった場合の結果である。以下では全て突然変異率は全遺伝子に対して1%及び遺伝子の各要素に対して3%、交叉はランキングの下位70%で組み合わせを作成し、行う。

[計算例1] GAは収束が不安定になる傾向がある。ここではまず比較実験として、エクセルのソルバーで行った数値計算との比較を行う。ここでコスト22000を基準にしソルバーでは約3時間を要している。

遺伝子数	終了条件
100	コスト 22000 以下

各ノード間の配送率

	D1	D2	D3	R1	R2	R3	R4
F1	0.23	0.10	0.54	0.00	0.02	0.22	0.13
F2	0.39	0.75	0.22	0.30	0.58	0.07	0.09
F3	0.39	0.14	0.23	0.01	0.04	0.16	0.37
D1				0.17	0.11	0.12	0.00
D2				0.42	0.22	0.11	0.02
D3				0.01	0.02	0.32	0.37

各ノード間の配送量

	D1	D2	D3	R1	R2	R3	R4
F1	11.8	10.5	47.1	0.33	1.87	26.7	12.0
F2	19.8	72.7	19.5	45.7	52.8	8.65	8.32
F3	19.8	14.1	20.2	1.37	4.03	18.9	33.7
D1				25.8	9.98	14.9	0.70
D2				63.1	19.8	13.0	1.50
D3				13.8	1.51	37.9	33.7

各ノードの総配送量

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
総配送量	110.4	227.5	112.2	51.37	97.39	86.89	150	90	120	90

各ノードの配送量分散

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
配送量分散	32.92	124.4	34.52	5.733	23.73	24.88	100	100	100	100

各ノードの安全在庫

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
安全在庫	13.4	26.0	13.7	6.87	15.5	16.9	27.9	28.7	38.3	28.6

コスト値

配送コスト	在庫コスト	総コスト	計算時間
19444.96	2551.39	21996.35	3分

このGAアルゴリズムでは約3分で行うことができた。この計算で用いているトポロジーはダミーノードを含め32ノードである。ソルバーでは一度に計算することは出来ずに5個の計算過程に分割し計算を行わなければならない。これより有効な計算方法だということになる。

[計算例 2] 次に遺伝子数を 1000, 終了条件をループ数 100 にした場合の計算を行う.

遺伝子数	終了条件
1000	ループ数 100

各ノード間の配送率

	D1	D2	D3	R1	R2	R3	R4
F1	0.12	0.14	0.60	0.03	0.00	0.36	0.41
F2	0.34	0.83	0.16	0.20	0.55	0.01	0.08
F3	0.54	0.02	0.23	0.16	0.00	0.10	0.09
D1				0.31	0.05	0.09	0.09
D2				0.21	0.38	0.07	0.19
D3				0.08	0.02	0.36	0.14

各ノード間の配送量

	D1	D2	D3	R1	R2	R3	R4
F1	8.18	13.0	42.6	4.34	0.08	43.2	37.4
F2	24.1	75.7	11.5	30.4	49.2	1.72	7.46
F3	37.9	2.05	16.1	24.6	0.04	12.6	7.76
D1				47.0	4.07	11.2	7.83
D2				31.4	34.5	8.00	16.8
D3				12.2	2.00	43.2	12.8

各ノードの総配送量

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
総配送量	148.8	200.2	101.0	70.17	90.73	70.26	150	90	120	90

各ノードの配送量分散

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
配送量分散	67.52	114.7	21.49	11.67	23.03	15.72	100	100	100	100

各ノードの安全在庫

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
安全在庫	19.17	24.99	10.82	8.90	14.65	12.28	32.31	25.47	31.74	37.94

コスト値

配送コスト	在庫コスト	総コスト	計算時間
19038.91	2540.94	21579.85	13分 25秒

[計算例 3] 次に遺伝子数 1000, 終了ループ数 1000 で行う.

遺伝子数	終了条件
1000	ループ数 1000

各ノード間の配送率

	D1	D2	D3	R1	R2	R3	R4
F1	0.34	0.06	0.86	0.15	0.01	0.33	0.53
F2	0.38	0.93	0.00	0.17	0.52	0.07	0.00
F3	0.29	0.01	0.14	0.05	0.02	0.09	0.13
D1				0.27	0.01	0.10	0.01
D2				0.33	0.44	0.05	0.14
D3				0.02	0.00	0.35	0.18

各ノード間の配送量

	D1	D2	D3	R1	R2	R3	R4
F1	18.4	6.57	53.8	22.2	1.32	40.1	47.8
F2	20.4	101.3	0.07	25.9	46.7	8.66	0.27
F3	15.6	0.88	8.55	7.55	1.60	10.7	11.6
D1				40.7	0.95	12.0	0.69
D2				49.9	39.3	6.54	13.0
D3				3.79	0.10	42.9	16.6

各ノードの総配送量

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
総配送量	190.2	203.3	56.47	54.36	108.78	62.42	150	90	120	90

各ノードの配送量分散

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
配送量分散	102.6	125.2	7.107	8.38	32.56	15.68	100	100	100	100

各ノードの安全在庫

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
安全在庫	23.64	26.11	6.221	9.036	15.03	9.259	32.60	24.74	33.67	30.42

コスト値

配送コスト	在庫コスト	総コスト	計算時間
18003.63	2440.35	20443.98	1時間46分42秒

計算例2に比べ、時間は7倍ほどかかったが一番最小のコストが得られた。

次に、GAアルゴリズムを用いた場合、収束が不確実なものになりやすく結果もばらけやすい。図24では、遺伝子数を100で固定しループ数でのコストの収束を考える。点線のグラフは個々の計算で得られたコストの推移を、実線は10回の計算の平均を表している。表より遺伝子100のときはループ数が約600回で有効な値が計算できると推測でき

る。そのときの時間は約 10 分であり、アプリケーションとして使うときに問題ない範囲と考えられる。また計算例 3 では 100 分以上時間がかかっているが、さらに詳細な改善計算が行われる。

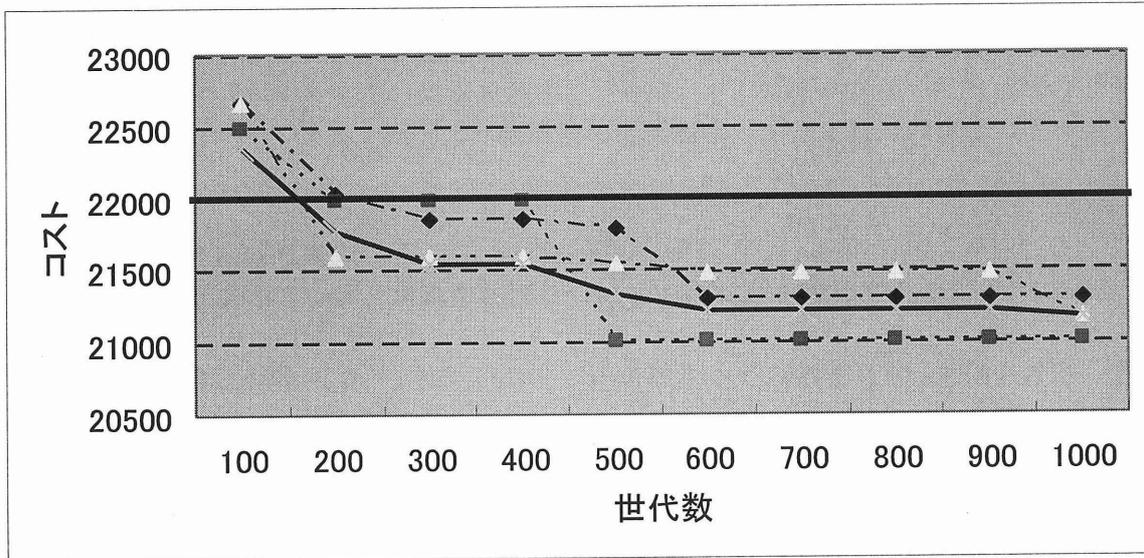


図 24: ループ数に対するコスト値

#### 5.3.4 まとめ

GAは幅広いモデルに適用できるが、収束に問題点が多い。今回のシステムは入出力に必要な項目を確立し、エクセルファイルから一貫して情報を取り込み、書き出しをすることで、システムの拡張をよりやりやすいものとすることができた。また入出力形式をそろえたことで既存のソフトウェア i-LAND との連携が容易となり、出力方法がより豊かなものとなる。また数値計算例からも一般的に業務として考えられる時間内で有効な結果が算出できることが示された。

### 5.4 主双対内点法の適用した SCN 改善システムの開発

前節で述べた GA を用いた最適化法は多くの問題に適用でき有効な手段である。しかし数学的手段が反映されないため収束までに多くの時間がかかり、収束にも不確実な点が多い。ここでは最適化アルゴリズムの 1 つである主双対内点法を適用した SCN 改善システムを紹介する。

#### 5.4.1 主双対内点法の概要

カーマーカー (Karmarkar) 法以来、LP 問題や QP 問題に対する内点法が理論的にも実用的にも非常に活発に研究されてきた。LP 問題に対する内点法の中でも、近年、特に主

双対内点法 (primal-dual interior point method) が有望視されている。主双対内点法に関する初期の研究として Kojima, Mizuno and Yoshise, Tanabe などがあり、特に前者は多項式オーダ性を扱ったものである。その後、多くの研究者によって主双対内点法の大域的収束性 (特に多項式オーダ性) や局所的収束性が研究されているばかりでなく、ソフトウェアパッケージにも組み込まれてその実用性が高く評価されている。

こうした状況の中で、非線形最適化問題に対する内点法の研究が現在進行中である。これは古典的な内点ペナルティ法の欠点を解消し、かつ、LP 問題で成功した内点法のアイデアを盛り込んだ研究である。以下で非線形最適化問題に対する主双対内点法のアルゴリズムを紹介する。

### 主双対内点法のアルゴリズム

以下に主双対内点法のアルゴリズムをあげる。詳しくは [58] を参照されたい。

step[0]  $\mu > 0$  と  $0 < \tau < 1$  を与える。

step[1] 以下の反復法によって、 $\|r(\omega; \mu)\|$  が十分に小さくなるような  $\omega$  を見つける。ここで、 $\|r(\omega; \mu)\|$  は式 (75) に与える。

step[1.0] 初期点  $\omega_0$  と初期行列  $B_0$  を与える。  $k = 0$  とおく。

step[1.1] 連立方程式 (76) を解いて探索方向  $\delta\omega_k$  を求める。

step[1.2]  $x_k + \alpha_{xk}\delta x_k > 0, z_k + \alpha_{zk}\delta z_k > 0$  となり、かつ、Armijo の基準を満足するステップ幅  $\alpha_{xk}, \alpha_{zk}$  を求める。

step[1.3]  $x_{k+1} = x_k + \alpha_{xk}\delta x_k, y_{k+1} = y_k + \alpha_{yk}\delta y_k, z_{k+1} = z_k + \alpha_{zk}\delta z_k$  とおく。

step[1.4] 行列  $B_{k+1}$  を求める。

step[1.5]  $k := k + 1$  とおいて内部反復の Step1.1 へいく。

step[2] もし  $\mu$  の値が十分に小さければ、求めた  $\omega$  を解とみなして終了する。さもなければ、 $\mu := \tau\mu/2$  とおいて外部反復の Step[1] へいく。

$$r(\omega; \mu) = \begin{pmatrix} \nabla_x L(\omega) \\ h(x) \\ XZe - \mu e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (75)$$

また、 $y \in R^l, z \in R^n$  をそれぞれ等式条件、非負条件に対するラグランジュ乗数、そして  $w = (x, y, z)^T \in R^n \times R^l \times R^n, e = (1, 1, \dots, 1)^T \in R^n, X = \text{diag}(\chi_1, \chi_2, \dots, \chi_n) \in R_{n \times n}, Z = \text{diag}(z_1, z_2, \dots, z_n) \in R^{n \times n}$  とする。上記の問題のラグランジュ関数は

$$L(\omega) = f(x) - y^T h(x) - z^T x$$

よって、KKT 条件は次の公式で与えられる。

$$\nabla_x L(\omega) = 0, h(x) = 0, XZe = 0,$$

ただし、 $\text{diag}(v_1, v_2, \dots, v_n)$  は  $v_i$  を対角成分に並べた対角行列である。またニュートン方程式に対して近似を考えた方程式は、

$$J_k \delta\omega_k = -r(\omega; \mu), \quad (86)$$

ここで,

$$\mathbf{J}_k = \begin{pmatrix} B_k & -\nabla h(\mathbf{x}_k) & -I \\ \nabla h(\mathbf{x}_k)^T & 0 & 0 \\ Z_k & 0 & X_k \end{pmatrix}.$$

$\mu$  の値を正の値から 0 に近づけたときに生成される  $\mu$ -センターの軌跡をセンターパスという。以上のように、主双対内点法の基本的な考え方は、整数  $\mu$  が与えられたときに、 $\mu$ -センターの近似解を求め、 $\mu \rightarrow 0$  として最終的に  $KKT$  点を求めていくものである。

#### 5.4.2 サプライチェーン・ネットワークへ主双対内点法の適用と SCN 改善システムの構築

4.3 節で記述したモデルに主双対内点法を適用する過程を述べる。ここで対象となる変数は  $p_{(k,i),(l,j)}$ , ( $k \leq l$ ) であるが、表現を簡単にするために

$$I = \sum_{n=1}^{k-1} M^{(n)} + i \quad (77)$$

$$N = \sum_{n=1}^W M^{(n)} \quad (78)$$

とする。上記の表現を使うと  $p_{(I,J)}$  とかける。また (37) 及び (38) は次のようにかける。

$$T_S^{(I)}(t) = \sum_{j=I+1}^N p_{(I,j)} S_d^{(j)}(t) \quad (1 \leq I \leq N - M^{(W)}) \quad (79)$$

$$T_S^{(I)}(t) = \sum_{j=I+1}^N p_{(I,j)} S_d^{(j)}(t) + d^{(j)}(t) \quad (N - M^{(W)} + 1 \leq I \leq N). \quad (80)$$

(79) 及び (80) を  $p_{(I,J)}$  で微分すると

$$\frac{\delta T_S^{(I)}(t)}{\delta p_{(I,J)}} = S_d^{(j)}(t) \quad (81)$$

となる。また (57) を計算すると、

$$\begin{aligned} V[S_d^{(I)}] &= V[T_S^{(I)}] \\ &= \sum_{j=I+1}^N p_{(I,j)}^2 V[T_S^{(j)}] + 2 \sum_{k < j} p_{(I,k)} p_{(I,j)} Cov[T_S^{(k)}, T_S^{(j)}] \end{aligned} \quad (82)$$

とかける。ここで  $Cov[X, Y]$  は変数  $X, Y$  の共分散をあらわす。以上より、 $T_S^{(I)}$  は微分可能であり、主双対内点法に適用できることがわかる。

## KKT 条件

問題 1 の KKT 条件を述べる．式 (77) を用いて目的関数 (68) を次のようにかく．

$$f(p_{(1,1)}, p_{(1,2)}, \dots, p_{(N,N)}) = \sum_{n=1}^N (IC[n] + TC[n]). \quad (83)$$

ここで  $(p_{(1,1)}, p_{(1,2)}, \dots, p_{(N,N)})$  は全てのノード間における配送率で構成され， $N \times N$  である．また制約条件をそれぞれ

$$\sum_{n \in I_J} p_{(n,J)} - 1 = 0, \quad (84)$$

$$-p_{(I,J)} \leq 0, \quad (85)$$

$$p_{(n,J)} = 0, \quad n \notin I_{(J)} \quad (86)$$

$$\sum_{n \in Id_J} pd_{(n,J)} - D_J \leq 0, \quad (87)$$

として式 (84) は  $N$  個，式 (85) は  $N \times N$ ，式 (86) は  $N \times N$  及び式 (87) は  $N$  個の制約式がある．よってその制約式の総数を  $M$  として，それぞれを  $g_i(p_{(1,1)}, p_{(1,2)}, \dots, p_{(N,N)}) \leq 0$ ， $i = 1, \dots, M$  とおく．また最適点  $p^* = (p_{(1,1)}^*, p_{(1,2)}^*, \dots, p_{(N,N)}^*)$  における目的関数の勾配ベクトルを  $\nabla f(p^*)$ ，第  $i$  番目の制約条件の勾配ベクトルを  $\nabla g_i(p^*)$  としたとき，ある  $\alpha_i \geq 0$ ， $i = 1, \dots, M$  が存在して KKT 条件が成立する．

$$\nabla f(p^*) + \sum_{i=1}^M \alpha_i \nabla g_i(p^*) = 0 \quad (88)$$

$$\alpha_i g_i(p_{(1,1)}^*, p_{(1,2)}^*, \dots, p_{(N,N)}^*) = 0, \quad i = 1, 2, \dots, M \quad (89)$$

$$g_i(p_{(1,1)}^*, p_{(1,2)}^*, \dots, p_{(N,N)}^*) \leq 0, \quad i = 1, 2, \dots, M \quad (90)$$

**処理手順** 基本的な手順は GA で提案した方法と同様である．つまり取り込むパラメータと出力する情報は同じ形式である．

(A) ダミーノードを利用したネットワーク構築：4.3 節で紹介した方法でリードタイムを分割したネットワークを構築する．

(B) パラメータの取り込み：配送コスト (表 14,15,16)，在庫コスト (表 8)，分割済みリードタイム (表 9)，小売需要平均 (表 10)，小売需要標準偏差 (表 11)，配送先トポロジー (表 17,18,19)，及び直送制限 (表 13) の各情報ファイルを取り込む．

(C) 目的関数決定．

(D) 主双対内点法でコスト計算．

(E) 情報出力：最適な遺伝子で求めた情報を出力する (配送率，配送量平均，配送量分散，安全在庫，配送コスト，在庫コスト，総コスト)．

(F) 実モデルへの変換：ダミーノードを利用して求めた結果をレベル指定 (表 12) を使い元のモデルに対応させる．

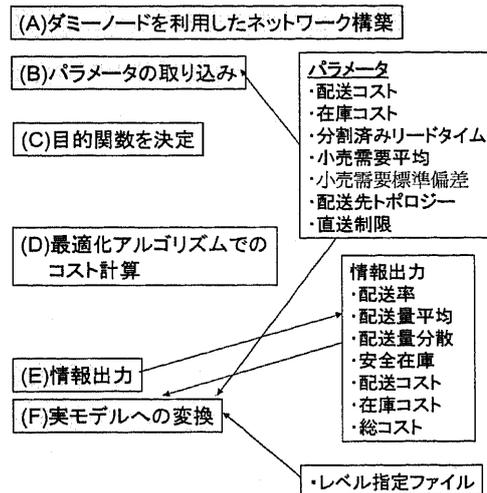


図 25: 主双対内点法 SCN 改善システム

### 5.4.3 主双対内点法での SCN 改善計算

ここでは 4.5 節で紹介したサプライチェーン・ネットワークを改善する。取り込むパラメータ等は GA-SCN システムと同様である。

#### [計算例]

各ノード間の配送率

	D1	D2	D3	R1	R2	R3	R4
F1	0.41	0.41	0.26	0.05	0.09	0.39	0.27
F2	0.25	0.31	0.42	0.32	0.52	0.07	0.11
F3	0.34	0.28	0.33	0.03	0.06	0.04	0.28
D1				0.39	0.09	0.10	0.08
D2				0.13	0.15	0.09	0.09
D3				0.08	0.10	0.32	0.16

各ノード間の配送量

	D1	D2	D3	R1	R2	R3	R4
F1	35.4	21.07	18.88	7.59	7.68	46.84	24.66
F2	21.35	16.14	30.69	47.34	46.59	7.89	9.57
F3	28.98	14.41	24.05	4.85	5.44	4.97	25.57
D1				59.03	7.7	11.43	7.6
D2				18.98	13.22	11.04	8.38
D3				12.21	9.37	37.83	14.22

各ノードの総配送量

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
総配送量	162.13	179.57	108.26	85.76	51.62	73.62	150	90	120	90

各ノードの配送量分散

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
配送量分散	57.50	74.38	27.18	17.84	5.47	14.18	100	100	100	100

各ノードの安全在庫

	F1	F2	F3	D1	D2	D3	R1	R2	R3	R4
安全在庫	17.69	20.12	12.17	13.94	11.44	16.11	30.25	35.58	33.89	34.61

コスト値

配送コスト	在庫コスト	総コスト	計算時間
19602.14	2672.87	22275.00277	2時間 52分 19秒

#### 5.4.4 まとめ

主双対内点法のシステム開発もGAと同様なエクセルでの入出力形式で作ることができた。GAでのツールと比較するとある一定コストまでの収束は主双対内点法でのツールのほうが早い。しかし、あるコスト以降極端に動きが遅くなる。これはモデルの記述の問題の可能性があり、特に目的関数がモデルの記述に有効か調べる必要がある。また今回は主双対内点法を用いることができるNUOPT(株式会社、数理システム)を用いている。ソフトウェアとしての制約が存在する可能性もある。将来的には自作のエンジンが求められる。しかし今回GAツールと並んで同じモデルの解析を他方面から行えたことはモデルの記述の正確性を確かめるという点において非常に有効であった。また既存ソフトを用いることで開発期間もGAに比べ短い期間で開発を行えた。

## 6 総括と今後の課題

本章では本論文の総括と今後の課題述べて本論文を結ぶ。本論文は大きく分けて、ネットワーク型優先権付き待ち行列、サプライチェーン・ネットワークの基本情報の算出、そしてそれを利用したシステム構築と改善について論じた。次にまとめる。

### 6.1 結果と今後の課題

**ネットワーク型優先権付き待ち行列** 本論文の目的は様々なネットワークモデルを数学的に記述し、その基本情報を算出する。その取り組みのひとつとしてネットワーク型優先権付き待ち行列のモデル化を行い、各合流点での待ち人数の定常分布を算出した。マルコフ性を有効に使うことで、複雑なモデルを簡略化できる。交通モデルに対して複数のサービス点と余裕空間を取り入れて定常分布を算出できたことは新しい事実である。

また現実モデルに適応するため余裕空間数  $c$  を各レベルで決定できることが望ましい。今回述べた解析方法は  $k-1$  レベルが再生時刻になったとき、 $k$  レベルが合流可能になる。このため  $c_1 \leq c_2 \leq \dots \leq c_N$  のときは必ず上位レベルが再生時刻になった後、下位レベルが合流可能になる。しかし  $c_{k-1} \geq c_k$  のときは  $k-1$  レベルが再生時刻になっていなくても  $k$  レベルが合流可能になる可能性がある。この場合は違う解析方法を用いる必要がある。

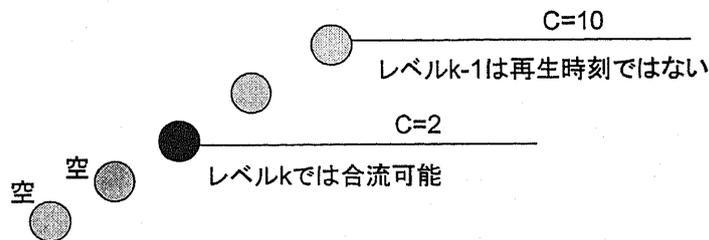


図 26: 余裕空間個数を可変としたときの問題点

本モデルは応用面では工場のピッキング処理の解析に期待ができる。各合流点での情報を元に5章で行った方法によりシステム改善の指標が容易に算出できる。

**サプライチェーン・ネットワークにおける基本情報の算出** サプライチェーン・ネットワークに対しては待ち行列ではなく、各平衡方程式をレベルごと行列化して、 $Z$ 変換を行うことで基本情報を算出した。ネットワークモデルに対してモデルにあった記述をすることで、必要な情報を算出できる。サプライチェーンは3段階モデルでは解析されていることはすでに述べたが、任意個のノードに対応するネットワーク構築法そしてその基本情報を数学的に算出することが出来た。配送と在庫の情報を同時に求めたことで、よりシステムを精密に解析することができる。

輸送ネットワークと安全在庫配置の同時改善解の算出とシステム構築 4章で算出した基本情報を用いて、SCNシステムを構築した。GAと主双対内点法の2方法で行い、比較実験することで基準が明確になり改善解を提案できる。またインターフェースを明確にすることでシステムが確立し、その他の最適化アルゴリズムにも対応可能にした。今後の課題として今回の実験では主双対内点法での収束精度が悪く、GAでの計算の基準点としか使われていない点があげられる。考えられる原因として式(68)中で使われている在庫コスト(66)は $S^{(k,i)}$ を用いている。 $S^{(k,i)} = D \times \sqrt{L^{(k)} + 1} \times \sqrt{V[T_S^{(k,i)}}$ より在庫コストが非線形になっている。これが正定値が保障できない原因と考えられ、収束に影響がでている可能性がある。改善方法として、(66)を2乗しルートをとったときの動きを観察することがあげられる。また式(68)はあくまでも例示的なものであるので、

$$cost = \sum_{k=1}^W \sum_{i=1}^{M^{(k)}} (\alpha(IC[k, i])^2 + (1 - \alpha)TC[k, i])$$

として各 $\alpha$ に対してコストを求め、システム改善の尺度をよりよくできる可能性がある。またシステム構築にあたり、改善解が算出されたとき、それを基準にした解の算出法が明確になっていない。つまり、ある改善解がでているときその1部を変更した場合、制約に全てを取り入れない限り、また全体計算を行ってしまう。部分計算を行えるようにシステムを構築する必要がある。

## 6.2 最後に

本論文で述べたことは現在行っている研究の基礎になっている。前半で待ち行列理論とその応用を述べたが、本論文ではモデル化し必要な情報を算出したことが主である。待ち行列理論は数学的には大変美しい理論であるが、現状のモデルへの適用が難しいと言われている。算出した結果を最適化理論に取り込み、システム全体の最適化に応用できれば、より待ち行列理論の適用範囲が広がり、さらに有効な手段になるであろう。また最適化は多岐にわたり行われている。モデル解析し必要な情報を数学的に得て、それを基に静的条件化での最適化システムを構築したが、動的システムとリンクさせ、精密な最適化を行えるシステムの確立を課題とする。

## 参考文献

- [1] Alfa, A.S., and Neuts, M.F.: Modelling Vehicular Traffic using the Discrete Time Markovian Process, *Transportation Sci.*, **29**, (1995) 109-117.
- [2] F.Baskett, K.M.Chandy, R.R.Muntz and F.G.Palacios: Open, closed and mixed networks of queues with different classes of customers, *J.ACM*,**22**,(1975) 248-260.
- [3] P. Bremaud: *Markov chains gibbs fields, monte carlo simulation, and queues* , (Springer, 1998).
- [4] P.J.Burke: The Output of a queueing system, *Operations Research*, **4**,(1956) 699-704.
- [5] P.J.Burke: The output of a stationary M/M/s queueing systems, *Annals of Mathematical Statics*, **39**, (1968) 1144-1152.
- [6] K.M. Chandy, J.H.Jr.Howard and D.F.Towsley: Product form and local balance in queueing networks, *J.ACM*, **24**,(1977) 250-263.
- [7] X.Chao, M.Miyazawa, R.Serfozo and H.Takada: Markov Network Processes with Product Form Stationary Distributions, *Queueing Systems*, **28**, (1998) 377-403.
- [8] A.Cobham: Priority assignment in waiting line problems. *Operations Research*, **2** (1953), 70-76.
- [9] R.Cowan : An Improved Model for Signalised Intersections with Vehicle - Actuated Control, *J.Appl.Prob.*, **15**, (1978) 384-396.
- [10] J.N.Darroch , et al.:Queues for a Vehicle - Actuated Traffic Light, *Operations Reserch*, **12**, (1964) 882-895.
- [11] 藤野直明: サプライチェーン経営入門, 日本経済新聞社, (1999).
- [12] E.Gelenbe: Product-form queueing networks with negative and positive customers, *Journal of Applied Probability*, **28**, (1991) 656-663.
- [13] 後藤正幸: ミルクラン方式のアナロジーによる直送と一括配送の混合配送方式について, 日本経営工学学会平成17年度春季大会予稿集,(2005), 48-49.
- [14] 後藤正幸, 増井忠幸, 俵 信彦: 制御理論に基づくミルクラン型物流システムの特性解析, 日本経営工学学会平成17年度秋季大会予稿集, (2005), 80-81.
- [15] 伊佐田文彦: 戦略的アーキテクチャの視点からみたサプライ・チェーン・マネジメント. *Nucb Journal Economics and Information Science*,**49-2**, 57-70.
- [16] 伊庭斉志: 遺伝的アルゴリズムの基礎, オーム社, (1994).

- [17] 井上春樹, 佐々木淳, 米田多江, 船生豊:セル GA 法と 2-opt 法を適用した大規模実時間配送計画システムの開発, 計測自動制御学会, **3-11**, (2004), 80-90.
- [18] 岩間一雄:新世代の計算限界—その解明と打破—, 科学研究費補助金「特定領域研究」平成 16 年度発足特定領域申請書.
- [19] J.R.Jacson: Networks of waiting lines, *Operations Research*, **5**,(1957) 518-521.
- [20] S.Kasahara, T.Takine, Y.Takahashi, and T.Hasegawa : Analysis of an SPP/G/1 System with Multiple Vacations and E-limited Service Discipline, *Queueing Systems*, **14**, (1993) 349-367.
- [21] S.Kasahara, Y.Takahashi and T.Hasegawa : Analysis of Waiting Time of M/G/1/K System with Vacations under Random Scheduling and LCFS, *Performance Evaluation*, **21**, (1995) 239-259.
- [22] S.Kasahara, T.Takine, Y.Takahashi and T.Hasegawa : MAP/G/1 Queues under N-policy with and without Vacations, *Journal of the Operations Research Society of Japan*, **39**, (1996) 188-212.
- [23] S.Kasahara, H.Takagi, Y.Takahashi and T.Hasegawa : M/G/1/K System with Push-out Scheme under Vacation Policy, *Journal of Applied Mathematics and Stochastic Analysis*, **9**, (1996) 143-157.
- [24] S.Kasahara, Y.Takahashi and T.Hasegawa : Analysis of Waiting Time of M/G/1/K System under Random Scheduling and LCFS, *Engineering Simulation*, **17**, (2000) 819-834.
- [25] O.Kella : Stability and nonproduct form of stochastic fluid networks with Levy inputs, *Annals of Applied Probability*, **6**, No 1, (1996) 186-199.
- [26] 加藤 豊, 小沢 正典: OR の基礎, 実教出版株式会社, (1998).
- [27] 菊池康也: SCM の理論と戦略, 税務経理協会, (2006).
- [28] 北野宏明: 遺伝的アルゴリズム , 産業図書, (1993).
- [29] 北野宏明: 遺伝的アルゴリズム 2 , 産業図書, (1995).
- [30] 北野宏明: 遺伝的アルゴリズム 3 , 産業図書, (1997).
- [31] L.Kleinrock: Queueing Theory, *Wiley*, (1975).
- [32] 小島 紀男, 篠崎 寿夫: Z 変換入門, 東海大学出版会, (1981).
- [33] Lehoczky, J.P. :Traffic Intersection Control and Zero - Switch Queues under Conditions of Markov Chain Dependence Input, *J.Appl.Prob.*, **9**, (1972) 382-395.

- [34] 三浦英俊: 交通機関の利用待ち時間モデル, 日本オペレーションズ・リサーチ学会平成9年度秋季研究発表会予稿集 (1997), 64-65.
- [35] M.Miyazawa: Rate conservation laws: a survey, *Queueing Systems*, **15**, (1994) 1-58.
- [36] M.Miyazawa and R.W.Wolff: Symmetric queues with batch departures and their networks, *Advance in Applied Probability*, **28**, (1996) 308-326.
- [37] S.Mizuno, Y.Sakuma and M.Miyazawa: A queueing model for local traffics to join a main stream under a leading space condition, *Journal of the Operations Research Society of Japan*, **49**, (2006) 319-331.
- [38] M.Miyazawa and H.Takada: Traffic flows and product form solutions in stochastic transfer networks, *Queueing Systems*, **37**, (2001) 199-232.
- [39] M.Miyazawa : Structure-reversibility and departure functions of queueing networks with batch movements and state dependent routing, *Queueing Systems*, **25**, (1997), 45-75.
- [40] 宮沢政清: 待ち行列ネットワークと積形式解, 日本オペレーションズ・リサーチ学会, **43**, No 8, (1998) 442-448.
- [41] 宮沢政清: 待ち行列の数理とその応用, 牧野書店, (2006).
- [42] 水野信也, 宮沢政清: 電子情報通信学会技術研究報告. IN, 情報ネットワーク, **102**, (2002) 61-65.
- [43] 水野信也, 高井英造, 八巻直一: ネットワーク型サプライチェーンの解析, 日本経営工学学会, 平成18年5月27日 春季大会予稿集 112-113.
- [44] 水野信也, 高井英造, 八巻直一: ネットワーク型サプライチェーンの解析法の提案, スケジューリング学会, 平成18年10月21日 講演論文集 17-21.
- [45] E.Reich: Note on queues in tandem, *Annals of Mathematical Statistics*, **34**, (1963) 338-341.
- [46] 白石洋一: 組合せ最適化アルゴリズムの最新手法, 丸善株式会社, (2002).
- [47] スミチ・レビ, E. スミチ・レビ, P. カミンスキ, 久保幹雄監修訳: サプライチェーンの設計と管理—コンセプト・戦略・事例, 朝倉書店, (2002).
- [48] 圓川隆夫, 曹徳弼: ロジスティクスシステムにおける情報共有化とリードタイム短縮: 小ロット化の効果と対策に関する研究, 日本経営工学会誌, **44-6**, (1994), 501-509.
- [49] 社団法人日本ロジスティクスシステム協会: 輸送ラベルに関わるビジネスモデル検討報告書 (ALIS), (2001).
- [50] L. Takaács: Priority Queues. *Operations Research*, **12**, (1964), 63-74.

- [51] H. Takagi: Priority queues with batch Poisson arrivals. *Operations Research Letters*, **10**, (1991) 225-232.
- [52] H. Takagi: *Queueing analysis, Vol 3: Discrete-time system*, (North-Holland, Amsterdam, 1993).
- [53] J. Walrand, P. Varaiya: *High-performance Communication Network, Second edition* (Morgan Kaufmann, San Francisco, California, 2000, 211-223).
- [54] 高橋 幸雄: 待ち行列研究の新しい潮流 (1) 待ち行列研究の変遷, 日本オペレーションズ・リサーチ学会, **43**, No 9, (1998) 495-499.
- [55] 特許庁技術調査課: サプライチェーン・マネジメントに関する技術動向調査, 平成13年5月31日.
- [56] 渡邊真也, 廣安知之, 三木光範: 遺伝的アルゴリズムを用いた多目的最適化へのアプローチ, 日本機械学会誌, **106**, (2003) 49.
- [57] W. Whitt: *Stochastic-Process Limits, An Introduction to Stochastic-Process Limits and Their Application to Queues* (Springer, 2001).
- [58] 八巻 直一, 矢部 博: 非線形計画法, 朝倉書店, (1999).
- [59] 山下 英明 : 交差点での交通流モデル, 日本オペレーションズ・リサーチ学会, **43**, No 5, (1998) 286-287.

## 付録

### .1 NUOPT での主双対内点法計算

主双対内点法を適用できる NUOPT (ニューオプト) というソフトを使い最適化を行う。NUOPT は線形計画, 整数計画, 2 次計画, 更に一般の非線形計画問題に対応した数理計画法パッケージで, 株式会社 数理システムが開発, 販売を行っている。NUOPT の主な特徴を以下にあげる。

- 大規模問題の安定かつ高速な求解を意図として開発され, 行列算法等の基礎的な数値演算技術からアルゴリズムにわたるまで, 現在得られる最新の技法に基いている。また, 問題規模の制限はない。
- 内点法と単体法のクロスオーバーにも対応している。
- NUOPT はモデリング言語 SIMPLE を標準添付しており, 通常のプログラミング言語を使うよりもはるかに少ないステップ数で, ほぼ数式を書き下すように記述し, 解を得ることができる。
- NUOPT は C++ のライブラリとして, C++ コードや VB, VBA の中から利用することも可能。

#### モデリング言語 SIMPLE

モデリング言語 SIMPLE はシステムの記述をなるべく人間に馴染みのある数学的な記述方法で簡単に行い, 実際のシミュレータやソルバなどが認識できるような表現に翻訳して所要の解析を行うことを目的としている。モデリング言語として, SIMPLE は次のような特徴を備えている。

- SIMPLE を用いると数式に近い自然な形で問題記述ができるほか, 自動微分機能により, 大規模で複雑な非線形計画問題に対しても高階の微係数の情報を活かした精度の高いアルゴリズムを適用することが可能。
- 概念上モデルとデータが分離されており, 大規模問題を簡潔に記述することができる。またシステムの解析法に関してユーザが意識する必要がない。
- SIMPLE は C++ のクラスライブラリとして実現されており, 制御ループや出力などについては C++ 固有の機能を生かすことができるほか, 他の C / C++ プログラムとのリンクも可能。
- モデリング言語 AMPL と接続するインタフェースもあり, AMPL 使用していた場合は既存の .nl ファイルをそのまま使って問題を入力可能。

### .2 サプライチェーン・ネットワークにおける配送率

表 5: 配送率の設定 (1)

	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
(1,1)	0.00	0.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.32
(1,2)	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.67	0.82	0.00
(1,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.21	0.00	0.47
(2,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(2,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00
(2,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
(3,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.00
(3,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.00
(3,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.21
(4,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(4,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(4,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,4)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,5)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,6)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,7)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,8)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,4)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,5)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,6)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,7)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,8)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,4)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

表 6: 配送率の設定 (2)

	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(6,1)	(6,2)
(1,1)	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(1,2)	0.00	0.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00	0.00
(1,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.55	0.00
(2,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(2,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(2,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(3,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(3,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(3,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(4,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(4,2)	0.00	0.00	0.00	0.00	0.00	1.00	0.00	1.00	0.00	0.00
(4,3)	0.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.45	0.00
(5,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
(5,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,4)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,5)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,6)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,7)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,8)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,4)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,5)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,6)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,7)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(6,8)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,4)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

表 7: 配送率の設定 (3)

	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(7,1)	(7,2)	(7,3)	(7,4)
(1,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.32	0.40
(1,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.37	0.43	0.00	0.00
(1,3)	0.84	0.00	0.28	0.00	0.00	0.00	0.00	0.00	0.00	0.22
(2,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(2,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(2,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(3,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(3,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(3,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(4,1)	0.00	0.69	0.00	0.24	0.00	0.60	0.21	0.00	0.00	0.00
(4,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.28	0.31	0.00	0.00
(4,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.36	0.04
(5,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,3)	0.16	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,4)	0.00	0.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,5)	0.00	0.00	0.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(5,6)	0.00	0.00	0.00	0.76	0.00	0.00	0.00	0.00	0.00	0.00
(5,7)	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
(5,8)	0.00	0.00	0.00	0.00	0.00	0.40	0.00	0.00	0.00	0.00
(6,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00
(6,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.00	0.00
(6,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.19	0.00	0.00
(6,4)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00
(6,5)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.00
(6,6)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.00
(6,7)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05
(6,8)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.30
(7,1)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,2)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,3)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
(7,4)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

### .3 SCN改善システムのパラメータ

表 8: 在庫コスト

(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
10	10	10	20	20	20	20	20	20	20	20	20
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(6,1)	(6,2)		
10	10	10	10	10	10	10	10	10	10		
(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(7,1)	(7,2)	(7,3)	(7,4)		
10	10	10	10	10	10	10	10	10	10		

表 9: 分割済みリードタイム

(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
1	1	1	1	1	1	1	1	1	1	1	1
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(6,1)	(6,2)		
1	1	1	1	1	1	1	1	1	1		
(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(7,1)	(7,2)	(7,3)	(7,4)		
1	1	1	1	1	1	1	1	1	1		

表 10: 小売需要平均

(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
0	0	0	0	0	0	0	0	0	0	0	0
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(6,1)	(6,2)		
0	0	0	0	0	0	0	0	0	0		
(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(7,1)	(7,2)	(7,3)	(7,4)		
0	0	0	0	0	0	150	90	120	90		

表 11: 小売需要標準偏差

(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
0	0	0	0	0	0	0	0	0	0	0	0
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(6,1)	(6,2)		
0	0	0	0	0	0	0	0	0	0		
(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(7,1)	(7,2)	(7,3)	(7,4)		
0	0	0	0	0	0	150	90	120	90		

表 12: レベル指定

(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
1	1	1	2	2	2	3	3	3	4	4	4
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(6,1)	(6,2)		
5	5	5	5	5	5	5	5	6	6		
(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(7,1)	(7,2)	(7,3)	(7,4)		
6	6	6	6	6	6	7	7	7	7		

表 13: 直送制限

0.4	1-13-21-29	1-21-29	1-29	2-13-21-29	2-21-29	2-29	3-13-21-29	3-21-29	3-29
0.667	1-15-23-30	1-23-30	1-30	2-15-23-30	2-23-30	2-30	3-15-23-30	3-23-30	3-30
0.5	1-17-25-31	1-25-31	1-31	2-17-25-31	2-25-31	2-31	3-17-25-31	3-25-31	3-31
0.667	1-19-27-32	1-27-32	1-32	2-19-27-32	2-27-32	2-32	3-19-27-32	3-27-32	3-32

表 14: 配送コスト (1) 空欄は 0 をあらわす

	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
(1,1)					15		15					29
(1,2)						15				29	29	
(1,3)								15		29		29
(2,1)												
(2,2)								15				
(2,3)									15			
(3,1)										29		
(3,2)											29	
(3,3)												29
(4,1)												
(4,2)												
(4,3)												
(5,1)												
(5,2)												
(5,3)												
(5,4)												
(5,5)												
(5,6)												
(5,7)												
(5,8)												
(6,1)												
(6,2)												
(6,3)												
(6,4)												
(6,5)												
(6,6)												
(6,7)												
(6,8)												
(7,1)												
(7,2)												
(7,3)												
(7,4)												

表 15: 配送コスト (2) 空欄は 0 をあらわす

	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(6,1)	(6,2)
(1,1)	15		15							
(1,2)					15		15			
(1,3)									15	
(2,1)										
(2,2)										
(2,3)										
(3,1)										
(3,2)										
(3,3)										
(4,1)										
(4,2)						15		15		
(4,3)		15		15						
(5,1)									15	
(5,2)										15
(5,3)										
(5,4)										
(5,5)										
(5,6)										
(5,7)										
(5,8)										
(6,1)										
(6,2)										
(6,3)										
(6,4)										
(6,5)										
(6,6)										
(6,7)										
(6,8)										
(7,1)										
(7,2)										
(7,3)										
(7,4)										

表 16: 配コスト (3) 空欄は0をあらわす

	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(7,1)	(7,2)	(7,3)	(7,4)
(1,1)									4	4
(1,2)							4	4		
(1,3)	15		15							4
(2,1)										
(2,2)										
(2,3)										
(3,1)										
(3,2)										
(3,3)										
(4,1)		15		15		15	35			
(4,2)							35	35		
(4,3)									35	35
(5,1)										
(5,2)										
(5,3)	15									
(5,4)		15								
(5,5)			15							
(5,6)				15						
(5,7)					15					
(5,8)						15				
(6,1)							4			
(6,2)							35			
(6,3)								4		
(6,4)								35		
(6,5)									4	
(6,6)									35	
(6,7)										4
(6,8)										35
(7,1)										
(7,2)										
(7,3)										
(7,4)										

表 17: 配送先トポロジー (1) 配送ルートあり:1

	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)	(4,1)	(4,2)	(4,3)
(1,1)					1		1					1
(1,2)						1				1	1	
(1,3)								1		1		1
(2,1)												
(2,2)								1				
(2,3)									1			
(3,1)										1		
(3,2)											1	
(3,3)												1
(4,1)												
(4,2)												
(4,3)												
(5,1)												
(5,2)												
(5,3)												
(5,4)												
(5,5)												
(5,6)												
(5,7)												
(5,8)												
(6,1)												
(6,2)												
(6,3)												
(6,4)												
(6,5)												
(6,6)												
(6,7)												
(6,8)												
(7,1)												
(7,2)												
(7,3)												
(7,4)												

表 18: 配送先トポロジー (2) 配送ルートあり:1

	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)	(6,1)	(6,2)
(1,1)	1		1							
(1,2)					1		1			
(1,3)									1	
(2,1)										
(2,2)										
(2,3)										
(3,1)										
(3,2)										
(3,3)										
(4,1)										
(4,2)						1		1		
(4,3)		1		1						
(5,1)									1	
(5,2)										1
(5,3)										
(5,4)										
(5,5)										
(5,6)										
(5,7)										
(5,8)										
(6,1)										
(6,2)										
(6,3)										
(6,4)										
(6,5)										
(6,6)										
(6,7)										
(6,8)										
(7,1)										
(7,2)										
(7,3)										
(7,4)										

表 19: 配送先トポロジー (3) 配送ルートあり:1

	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)	(7,1)	(7,2)	(7,3)	(7,4)
(1,1)									1	1
(1,2)							1	1		
(1,3)	1		1							1
(2,1)										
(2,2)										
(2,3)										
(3,1)										
(3,2)										
(3,3)										
(4,1)		1		1		1	1			
(4,2)							1	1		
(4,3)									1	1
(5,1)										
(5,2)										
(5,3)	1									
(5,4)		1								
(5,5)			1							
(5,6)				1						
(5,7)					1					
(5,8)						1				
(6,1)							1			
(6,2)							1			
(6,3)								1		
(6,4)								1		
(6,5)									1	
(6,6)									1	
(6,7)										1
(6,8)										1
(7,1)										
(7,2)										
(7,3)										
(7,4)										

## 謝辞

本論文の研究をするにあたり、指導教授である八巻直一先生には研究に対する考え方、特に実務的な捉え方を教えていただき新しい分野で研究を進めることができました。深く感謝致します。また、(株)フレームワークスの高井英造先生、後藤一孝氏には実践的な考え方からシステムの構築まで幅広くアドバイスを頂きました。ありがとうございました。

また、東京理科大学工学部情報科学科の宮沢政清先生には本論文の前半に書いた待ち行列理論での解析で多大なご指導を頂きました。ありがとうございました。またお忙しい中、審査会での的確なご指摘をしていただいた野飼享先生、吉村仁先生、関谷和之先生、上田芳伸先生、井上春樹先生に感謝申し上げます。特に関谷先生は細部に至るまで丁寧にご指導していただきました。ありがとうございました。

研究を進める上で、論文前半の待ち行列理論では東京理科大学工学部情報科学科の佐久間さん、浜松情報専門学校の花村さんには協力してもらいました。後半部分の最適化では浜松情報専門学校の青田さん、飯田さんにはソフトウェア開発の面で多大な協力を頂きました。また同校の小粥さん、岡野さん、村松さん、山下さんにも協力を頂きました。

この論文を完成させるために、たくさんの方々にご指導及びご協力を頂き深く感謝致します。最後に、常に私を支えてくれた妻をはじめ、家族に感謝します。