解　説

定性推論のパラメータチューニングへの応用†☆

伊藤　潔††　本位田真一†††

1. はじめに

動的なシステムの性能を改善するパラメータチューニングを厳密に行おうとすると，複雑な数式処理や数値計算が必要で計算量も多くなる．また，一般的な解法が存在しないこともあるため，さまざまな近似モデルや近似解法を用いることもある．この近似モデルや近似解法が現実の対象システムから遊離してしまうこともある．このため，専門家は，問題の性質に応じて種々の解法を巧みに組み合わせたり直観的な経験則や知識を用いていることが多い．

調整可能なパラメータがn個ある場合，チューニング可能な増減パターンが2^n通りある．しかし，改善するために必要なパラメータの組み合わせは，n個より少ないパラメータの組み合わせである場合が多い．また，必ずしも，1通りではなく複数の組み合わせが存在し得る．一つの組み合わせを**チューニングプラン** (tuning plan) と呼ぶ．チューニングプランを前もって求めずにむやみにチューニングしようとすると，2^n通り試みる必要がある．

このようなチューニングプランを求めるために，専門家の知識を整理する．この知識はかなり定性的であり定性的な表現の枠組みでうまく定式化できる．パラメータチューニングの第1段階として，定性的な推論で，チューニングに有効であるパラメータの増減を指示した複数のチューニングプランを求める．

この定性推論によるチューニングプランはパラメータの増減の指定のみであるから，具体的な増減量が不明なため，このままでは実用上不十分である．パラメータチューニングの第2段階として，前段の定性的なチューニングプランごとに，定量的な推論によってパラメータの定量的なチューニングプランを求める．

定性推論[BOB85],[APT86],[DEK85],[MIZ89],[NIS88,89]は，経済システムや電気回路などの定性的な挙動を表すモデルとして使われている．本稿で解説する待ち行列ネットワークへの適用は，定性推論の応用としては新規なものである．パラメータチューニングへの定性推論の主な適用例として，Rajagopalan, R. のターボジェットエンジンの対気速度と絞り弁の設定の増減関係を表す定性モデル[RAJ84]がある．本解説の方法では，動的なシステムのパラメータチューニングに，定性推論ばかりでなく相補的に定量推論を用いる．

一般に，定性推論が対象とするシステムは，

① その挙動の詳細が不明であり，かつ挙動を表す微分方程式などの式が不明なシステム，あるいは，

② 挙動がある程度分かっているが，その挙動を表す式が複雑かつ膨大，その解法が明確ではない，あるいは解くのに時間がかかるシステム，である．

定性推論では，このようなシステムの挙動について，数少ないパラメータ間の因果関係でモデル化し，数少ない定性式で説明や推論を行う．**3.2** で述べるとおり，パラメータチューニングを行うために待ち行列ネットワークの挙動をそのまま表すと非線形の連立等式・不等式となる．この連立式は膨大であり解くのに時間がかかる．すなわち，待ち行列ネットワークのパラメータチューニングは②の範疇に属し，定性推論の適用が有効であると考えられる．

---

† Application of Qualitative Reasoning to Parameter Tuning Process by Kiyoshi ITOH (Laboratory of Information Science, Natural Sciences Center, Faculty of Science and Technology, Sophia Univ.) and Shinichi HONIDEN (Systems and Software Engineering Laboratory, Toshiba Corporation).

†† 上智大学理工学部一般科学研究室情報科学部門

††† (株)東芝システム・ソフトウェア技術研究所

☆ 本解説は，待ち行列ネットワークのパラメータチューニングに関する著者らの論文[ITO**ab,**ab],[SAW**ab]の内容を，待ち行列ネットワークの基礎事項も含めて定性推論の応用としての観点からできるだけ分かりやすく解説したものである．

2. 対象の待ち行列ネットワーク

2.1 単一のサーバ

単一の**サーバ** (server) の5つのパラメータを定義する.

λ: 単位時間当りにサーバに到着するエンティティの平均個数 (**到着率**: arrival rate)

μ: 単位時間当りにサーバで処理できるエンティティの平均個数 (**サービス率**: servicing rate)

t: 単位時間当りにサーバから出てくるエンティティの平均個数 (**スループット**: throughput)

ρ: サーバが実際にエンティティを処理する時間の割合 (**稼動率**: utilization rate)

q: サーバの前で処理を待つエンティティの平均個数 (**待ち行列長**: queue length)

図-1 と**図-2** に示すとおり, λ と μ が与えられるとその大小関係によりそのサーバの性能を表す ρ, t が決まる.

図-1 は, 到着率がサービス率より小さい場合, すなわち, 平均到着時間間隔 (到着率の逆数) が平均サービス時間 (サービス率の逆数) より大きい場合, サーバは**よどみなくエンティティを処理できる**. サーバから出るエンティティの量 (スループット) は, サーバへの到着率に等しくなる. 稼動率は,

ρ=(平均サービス時間)/(平均到着時間間隔)

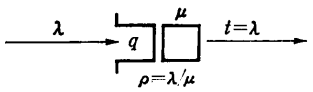


図-1 $\lambda<\mu$ の場合

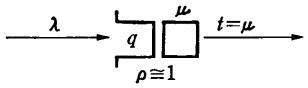


図-2 $\lambda\geqq\mu$ の場合

$=(1/\mu)/(1/\lambda)=\lambda/\mu$ となる.

図-2 は, 到着率がサービス率より大きい場合, すなわち, 平均到着時間間隔が平均サービス時間より小さい場合, サーバは**よどみなくエンティティを処理できない**. すなわち, あるエンティティに対するサービスを行っている間に次のエンティティがそのサーバに到着する. このサーバからのスループットは, サーバのサービス率に等しくなる. 稼動率は, サーバが常に働いていることになるので, $\rho\cong1$ となる.

2.2 待ち行列ネットワーク

待ち行列ネットワーク (queueing network; 以下 QN と略記) は, 複数のサーバが結合してネットワーク構造になったもの (たとえば**図-3** の "QN 4") である. QN 内で二つのサーバが**直列**に結合する場合には, 前段のサーバからのスループット t が, 後段のサーバへの到着率 λ となる. 二つのサーバからのスループット t_1 と t_2 が**合流**し

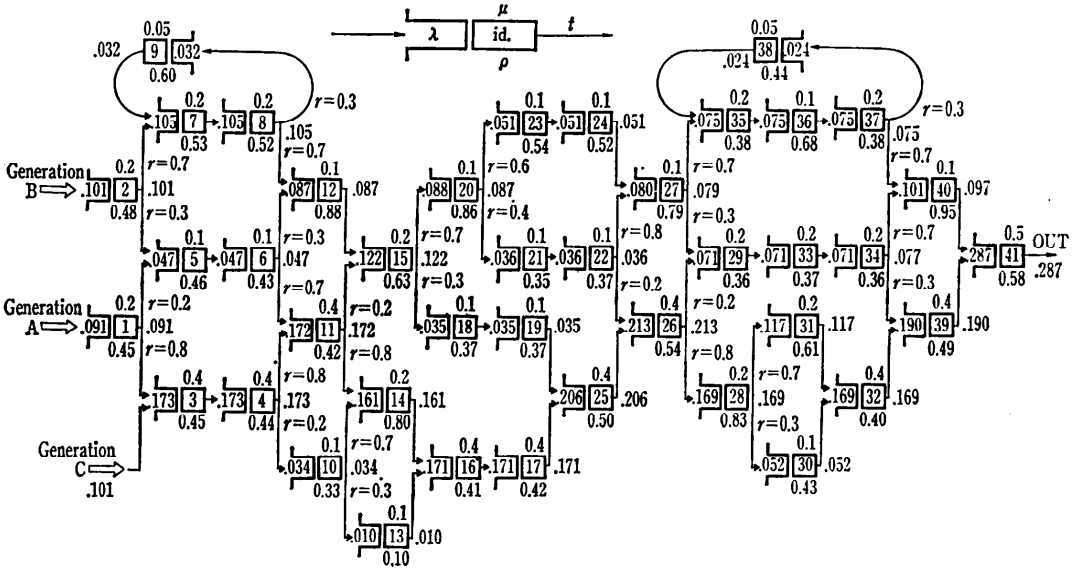


図-3 例題待ち行列ネットワーク "QN 4"

て後段のサーバに入る場合には，t_1+t_2 が後段のサーバへの到着率となる．あるサーバ s_1 のスループット t_1 が二つのサーバ s_2, s_3 に**分岐**する場合には，その**分岐確率** (branching probability) を r, $1-r$ とすると，s_2 と s_3 への到着率は rt_1, $(1-r)t_1$ となる．QN 4 の各サーバの入出力関係を表す**定量式**を図-4 に示す．

対象 QN は，エンティティが全て外部から到着し，処理済みになると外部へ出る**オープン型**の QN である．エンティティは外部から複数のサーバに到着できるが，内部ではそれらを区別しない．すなわち，内部ではエンティティを1種類と考えた**単一フロー型**の QN である．

2.3 稼動率とスループットの順次的計算

QN の構造（サーバ同士の結合の仕方）が固定であり，外部からのエンティティの入力量，サーバからの分岐確率，及び個々のサーバのサービス率が既知であれば，外部からエンティティが入るサーバを起点として，順次，下流に向かって，個個のサーバの λ, ρ, t を，図-1 と図-2 に従って決定できる．すなわち，$\lambda<\mu$ の場合は図-1 によりその到着率 λ をスループット t と決め，$\lambda\geqq\mu$ の場合をサーバのサービス率 μ をスループットと決める．その際，稼動率 ρ も決まる．このスループットは直下流のサーバへの到着率の一部—直列の場合はそのままの量（図-4（b)），合流の場合は他との和（図-4（d)），分岐の場合は分岐確率を乗じた量（図-4（c)(e)）—となり，順次，下流に向かって λ, ρ, t が決定できる．

実際のシステムを QN とみなして**モニタリング** (monitoring) を行い，パラメータを実測する場合，ばらつきの存在や立ち上がり時の測定データの棄却方法次第により，正確には一致しないがほぼ上の議論に従う．

3. QN のパラメータチューニング

3.1 ボトルネック

QN 内で稼動率が1にきわめて近いサーバは**ボトルネック** (bottleneck) である．このボトルネックサーバの**待ち行列** (queue) は無限に成長する恐れがある．ボトルネックサーバが存在する場合，QN は**非定常（過負荷）状態** (unstable or overloaded state) である．

稼動率が1に近くないが過大なサーバは**ボトルネックとなる可能性**がある．サーバへの到着時間間隔やサーバのサービス時間が大きなばらつきをもつ場合，稼動率が1に近くなくても待ち行列長が時には急激に増加する危険性がある．**フェールセーフ** (failure to safety) の考え方を導入して，稼動率が 0.7 以上のサーバにボトルネックの可能性があると経験的に診断していることが多い．

また，稼動率が過大ではないが待ち行列長が過大であるサーバには—測定がそれほど長時間行われていないかもしれない—，測定終了以降にエン

（a） 外部入力
$\lambda_1=ga$, $\lambda_2=gb$
（b） 直列
$\lambda_4=t_3$, $\lambda_6=t_5$, $\lambda_8=t_7$
$\lambda_{17}=t_{16}$, $\lambda_{19}=t_{18}$
$\lambda_{23}=t_{21}$, $\lambda_{24}=t_{23}$
$\lambda_{33}=t_{29}$, $\lambda_{34}=t_{33}$
$\lambda_{36}=t_{35}$, $\lambda_{37}=t_{36}$
（c） 分岐
$\lambda_9=t_{8,9}$, $\lambda_{10}=t_{4,10}$
$\lambda_{13}=t_{10,13}$, $\lambda_{15}=t_{15,15}$
$\lambda_{20}=t_{15,20}$, $\lambda_{21}=t_{20,21}$
$\lambda_{22}=t_{20,22}$, $\lambda_{26}=t_{26,26}$
$\lambda_{30}=t_{28,30}$, $\lambda_{31}=t_{26,31}$
$\lambda_{38}=t_{37,38}$
（d） 合流
$\lambda_3=gc+t_{1,3}$
$\lambda_5=t_{1,5}+t_{2,5}$
$\lambda_7=t_{3,7}+t_9$
$\lambda_{11}=t_{4,11}+t_{6,11}$
$\lambda_{12}=t_{6,12}+t_{8,12}$
$\lambda_{14}=t_{10,14}+t_{11,14}$
$\lambda_{15}=t_{11,15}+t_{13}$
$\lambda_{16}=t_{13}+t_{14}$
$\lambda_{18}=t_{17}+t_{19}$
$\lambda_{25}=t_{22,25}+t_{25}$
$\lambda_{27}=t_{23,27}+t_{24}$
$\lambda_{29}=t_{26,29}+t_{27,29}$
$\lambda_{32}=t_{30}+t_{31}$
$\lambda_{35}=t_{27,35}+t_{32}$
$\lambda_{39}=t_{35}+t_{34,39}$
$\lambda_{40}=t_{34,40}+t_{37,40}$
$\lambda_{41}=t_{39}+t_{40}$
（e） スループットと分岐確率
$t_{1,3}=t_1*r_{1,3}$
$t_{1,5}=t_1*r_{1,5}$
$t_{2,5}=t_2*r_{2,5}$
$t_{3,7}=t_3*r_{3,7}$
$t_{4,10}=t_4*r_{4,10}$
$t_{4,11}=t_4*r_{4,11}$
$t_{6,11}=t_6*r_{6,11}$
$t_{6,12}=t_6*r_{6,12}$
$t_{8,9}=t_8*r_{8,9}$
$t_{8,12}=t_8*r_{8,12}$
$t_{10,13}=t_{10}*r_{10,13}$
$t_{10,14}=t_{10}*r_{10,14}$
$t_{11,14}=t_{11}*r_{11,14}$
$t_{11,15}=t_{11}*r_{11,15}$
$t_{15,15}=t_{15}*r_{15,15}$
$t_{15,20}=t_{15}*r_{15,20}$
$t_{20,21}=t_{20}*r_{20,21}$
$t_{20,22}=t_{20}*r_{20,22}$
$t_{22,26}=t_{22}*r_{22,26}$
$t_{22,27}=t_{22}*r_{22,27}$
$t_{26,26}=t_{26}*r_{26,26}$
$t_{26,29}=t_{26}*r_{26,29}$
$t_{27,29}=t_{27}*r_{27,29}$
$t_{27,35}=t_{27}*r_{27,35}$
$t_{28,30}=t_{28}*r_{28,30}$
$t_{28,31}=t_{28}*r_{28,31}$
$t_{34,39}=t_{34}*r_{34,39}$
$t_{34,40}=t_{34}*r_{34,40}$
$t_{37,40}=t_{37}*r_{37,40}$
$t_{37,38}=t_{37}*r_{37,38}$
（f） 分岐確率
$ri,j+ri,k=1(j\neq k)$

図-4 各サーバの入出力の定量式

ティティが過大に到着して稼動率が大きくなる危険性がある．この過大な待ち行列長は後に徐々に解消されるかもしれないが，ボトルネックの発生の事前防止というフェールセーフの考え方で，測定時に平均待ち行列長が1以上のサーバにボトルネックの可能性があると診断する．

3.2 パラメータチューニングの難しさ

ボトルネックを改善するチューニングプラン(通常一つ以上存在)は，QNの構造のみを固定としほかの全てのパラメータを可変とした場合に，稼動率ρが0.7を下回るためにどのパラメータを変更すればよいか，という問題の，択一的に選択可能な解である．

図-3のQN4で，その構造は固定であり，外部からのエンティティの入力量，サーバからの分岐確率，及び各サーバのサービス率が既知であるとする．実測，シミュレーション，上流からの順次的な計算(**2.3**)などの方法があるが，この例ではシミュレーションにより各サーバの稼動率，スループット，到着率が求められた．この後，ボトルネックを改善するために，構造のみを固定としてパラメータチューニングする．

QN4ではボトルネックの可能性のあるサーバとして，$\rho \geqq 0.7$ の $s12$, $s14$, $s20$, $s27$, $s28$, $s40$ がある．改善の例として $s20$ をあげる．この $s20$ の稼動率 $\rho 20$ が0.7を下回るようにするためには，$s20$ にのみ着目すると，

(1) サービス率 $\mu 20$ を増加する，あるいは

(2) 到着率 $\lambda 20$ を減少する　という方法をとり，

(新しい $\rho 20$) = (新しい $\lambda 20$)/(新しい $\mu 20$) < 0.7

とする．

単一サーバではなく複数のサーバが結合しているQNでは，この(1)，(2)の適用のみでは不十分である．

(1)の方法を実施すると，これまで稼動率が高いため $s20$ の待ち行列に滞留する傾向のあったエンティティが，$s20$ のサービスを以前より多く受けることが可能となり，$s20$ のスループットが増加する．$s20$ の直下流の $s21$, $s23$ への到着率を増加させ，さらにその下流のサーバへの到着率も増加させる．$s20$ の下流にありすでにボトルネックの可能性のあったサーバ($s27$, $s28$, $s40$)の稼動率をさらに増加させるとともに，ボトルネックではなかったサーバもその可能性をもつものにしてしまう危険性がある．

(2)の方法を実施するためには，$s20$ から上流に遡行し外部からの入口までの経路上の

(2-1) 分岐確率を減少する，あるいは，

(2-2) 外部からの入力量を減少する，方法をとる．

(2-1)の方法を実施するためには，上流への経路上に複数個の分岐点が存在する場合，どの分岐点の分岐確率を減少するかを決定しなければならない．$s20$ から上流への経路には，3個の分岐点($s15$, $s11$, $s4$)が存在する．また，その分岐点の他方は分岐確率が増加するので下流の到着率が増加することになり，(1)と同様に下流にあるボトルネックに留意しなければならない．

チューニングプランを得るために**2.3**の順次的計算式を使うと，上記の(1)，(2)，(2-1)，(2-2)に現れたパラメータを変数とした式が立式できる．この式には，(2-1)の複数の分岐確率を変数とした積項が含まれる．また，改善後の稼動率が0.7を下回るから全てのサーバについて $\rho = \lambda/\mu < 0.7$ という式を与える必要がある．立式は非線形の連立等式・不等式となる．

この連立式は膨大であり解くのに時間がかかる．すなわち，待ち行列ネットワークのパラメータチューニングは，**1.** の②の範疇に属し，定性推論の適用が有効であると考えられる問題である．

待ち行列理論 (queueing theory) (たとえば文献[KLE 75], [GEL 80])では，平均値だけではなく，到着時間間隔やサービス時間にさまざまな分布を仮定し，稼動率，待ち時間などの性能パラメータの厳密解や近似解を求めるものが多い．性能を改善するためにパラメータ群をチューニングする問題を対象とすることはあまり多くない．

非線形の連立式を解かないで，択一的に選択可能なチューニングプランを可能なかぎり効率的に得たい．このために，QNの構造と性能パラメータに関する経験則を用いて定性推論の枠組みでチューニングプランを列挙する方法を **4.** で述べる．

3.3 チューニングプランの意味

一つのチューニングプランは，①外部からの入力量を減少，②分岐確率を減少，③サーバのサービス率を増加，などのパラメータ変更の組合わせ

となる．これらのパラメータを全て可変としており，そのため，どのチューニングプランも定性的には選択可能である．

①はシステムに対する負荷の減少を意味する．②は，エンティティがこれまで流れていたパスの部分的な変更を意味する．分岐後が並列なパスであれば，分岐確率の変更は各パスへの負荷を変更することになり，②は実現可能である．並列なパスではない場合，分岐確率を変更してよいかどうかはシステムによる．

③の説明として，**図-5**（a）のボトルネックサーバ α を改善するためにサーバの稼動率を 0.7 に減少させる例を述べる．**図-5**（b）では，サービス率が 2 倍（サービス時間が 1/2）のサーバに置き換えて同じ仕事をさせる．このようなサーバの置き換えが現実的ではない場合，**図-5**（c）のサーバのパイプライン化や**図-5**（d）のサーバの並列化が，実際的な方法である．図-5（c）は，14h かかる仕事を 7h と 7h に直列に分割できる場合，二つのサーバをパイプライン化して並べる方法である．図-5（d）は，14h で行うサーバをもう一つ用意しておのおのへ同じ分岐確率で分岐させる方法である．パイプライン化と並列化についての一般的な例は本解説では省略する．

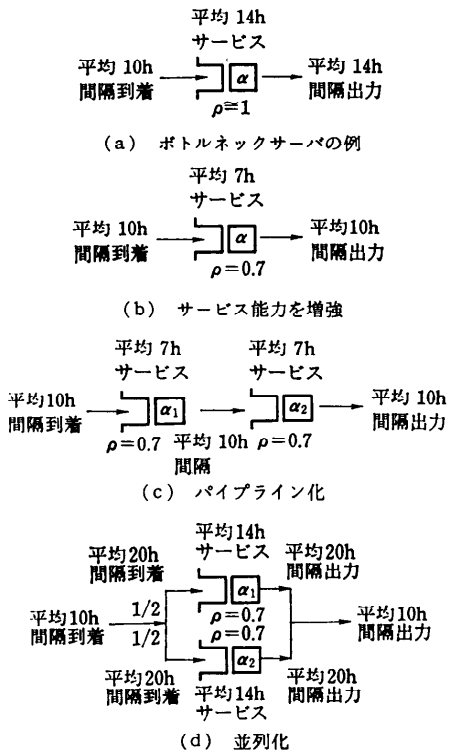


図-5 ボトルネックサーバのサービス能力を増強する方法
到着時間間隔，出力時間間隔，サービス時間の逆数は，到着率，スループット，サービス率

4. 定性的なパラメータチューニング

4.1 ボトルネックランドマーク

専門家の経験則として，稼動率 0.7 を**ボトルネックランドマーク** (bottleneck landmark : BL) とし，また，行列長 1 を待ち行列長に関するボトルネックランドマーク (QBL) とし，$\rho \geqq$BL，あるいは，$q \geqq$QBL ならばボトルネックの可能性があるとする．

定性推論記法で表現する．$[\rho]$ と $[q]$ は，おのおの，BL と QBL を原点として評価した定性値 (+, 0, −) である．

$[\rho]=+$：ボトルネックの可能性がある．この場合フェールセーフの考え方で $d\rho=+$ と考える．

$[q]=+$：ボトルネックの可能性がある．この場合 $dq=+$ と考える．

4.2 単一サーバの定性式

サーバの挙動を定性式で表す．ボトルネックの可能性の有無で用いる式の種類が異なることに特長がある．以下の式をサーバの**定性挙動式** (qualitative behavior expression : QLBE) と呼ぶ．ボトルネックの可能性がある場合（下記の b））を特に**定性改善式** (qualitative bottleneck improvement expression : QL-BIE) と呼ぶ．

a） $\rho <$BL の場合（$\lambda < \mu$ の場合：図-1 参照）すなわち

$[\rho]=-$（ρ が BL より小さい）の場合

a-1) λ について

$d\rho=\pm \leftarrow d\lambda=\pm$（$\lambda$ と ρ は同じ方向に増減）

$dt=\pm \leftarrow d\lambda=\pm$（$\lambda$ と t は同じ方向に増減）

a-2) μ について

$d\rho=\mp \leftarrow d\mu=\pm$（$\mu$ と ρ は反対方向に増減）

$dt=0 \leftarrow d\mu=\pm$（μ の増減に t は追随せず）

b） $\rho \geqq$BL の場合（$\lambda \geqq \mu$ の場合：図-2 参照）すなわち

$[\rho]=+$or 0（ρ が BL 以上，すなわちボトルネックの可能性がある）

または，$q \geqq$QBL，すなわち $[q]=+$or 0 の場合

b-1) λ について

$d\rho=- \leftarrow d\lambda=-$（$\lambda$ の減少に ρ は追随）

$dt=0 \leftarrow d\lambda=-$（$\lambda$ の減少に t は追随せず）

b-2) μ について

$d\rho=-\leftarrow d\mu=+$ (μ が増加すると ρ は減少)

$dt=+\leftarrow d\mu=+$ (μ が増加すると t は増加)

b-3) q について

$dq=-\leftarrow d\rho=-$ (ρ の減少に q は追随)

$d\rho=-\leftarrow dq=-$ (q の減少に ρ は追随)

4.3 経験則を用いない定性挙動推論

3.2 で述べたとおり，サーバの λ の変化のためには，上流のサーバからの t の変化が必要である．あるサーバの t の増加は下流のサーバへの λ の増加につながるので，下流のボトルネックへの留意が必要である．チューニングの一つの方法は，全てのサーバに対して定性式で立式して，それらを組み合わせて解く方法である．

図-3 の QN 4 で，ボトルネックの可能性のあるサーバの定性改善式 (QL-BIE) 及びそのほかのサーバの定性挙動式 (QLBE) を**図-6** に示す．QN 4 に対する式の個数は，図-6 に示すとおり

for $i=\{s12, s14, s20, s27, s28, s40\}$
(which may be bottleneck)
・qualitative state : $[\rho i]=+$
$d\rho i=+$
・QL-BIE ;
$d\rho i=-\leftarrow d\lambda i=-$
$dti=0\leftarrow d\lambda i=-$
$d\rho i=-\leftarrow d\mu i=+$
$dti=+\leftarrow d\mu i=+$
subtotal : 24 expressions
=4 expressions×6 (servers)

for $j=\{s1, s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s13, s15, s16, s17, s18, s19, s21, s22, s23, s24, s25, s26, s29, s30, s31, s32, s33, s34, s35, s36, s37, s38, s39, s41\}$
・qualitative state : $[\rho j]=-$
・QLBE : $d\rho j=\pm\leftarrow d\lambda j=\pm$
$dtj=\pm\leftarrow d\lambda j=\pm$
$d\rho j=\mp\leftarrow d\mu j=\pm$
$dtj=0\leftarrow d\mu j=\pm$
subtotal : 280 expressions
=8 expressions×35 (servers)

total : 304 expressions

図-6 QN 4 の各サーバの定性挙動式と定性改善式

(a) 外部入力

$d\lambda_1=\pm\leftarrow dga=\pm$

$d\lambda_2=\pm\leftarrow dgb=\pm$

subtotal : 4 expressions
=2 expressions×2 (servers)

(b) 直列

$d\lambda_4=\pm\leftarrow dt_3=\pm$
$d\lambda_6=\pm\leftarrow dt_5=\pm$
$d\lambda_8=\pm\leftarrow dt_7=\pm$
$d\lambda_{17}=\pm\leftarrow dt_{16}=\pm$
$d\lambda_{19}=\pm\leftarrow dt_{18}=\pm$
$d\lambda_{22}=\pm\leftarrow dt_{21}=\pm$
$d\lambda_{24}=\pm\leftarrow dt_{23}=\pm$
$d\lambda_{33}=\pm\leftarrow dt_{29}=\pm$
$d\lambda_{34}=\pm\leftarrow dt_{33}=\pm$
$d\lambda_{36}=\pm\leftarrow dt_{35}=\pm$
$d\lambda_{37}=\pm\leftarrow dt_{36}=\pm$

subtotal : 22 expressions
=2 expressions×11 (servers)

(c) 分岐

$d\lambda_9=\pm\leftarrow dt_{8,9}=\pm$
$d\lambda_{10}=\pm\leftarrow dt_{4,10}=\pm$
$d\lambda_{13}=\pm\leftarrow dt_{10,13}=\pm$
$d\lambda_{18}=\pm\leftarrow dt_{15,18}=\pm$
$d\lambda_{20}=\pm\leftarrow dt_{15,20}=\pm$
$d\lambda_{21}=\pm\leftarrow dt_{20,21}=\pm$
$d\lambda_{23}=\pm\leftarrow dt_{20,23}=\pm$
$d\lambda_{28}=\pm\leftarrow dt_{26,28}=\pm$
$d\lambda_{30}=\pm\leftarrow dt_{28,30}=\pm$
$d\lambda_{31}=\pm\leftarrow dt_{28,31}=\pm$
$d\lambda_{38}=\pm\leftarrow dt_{37,38}=\pm$

subtotal : 22 expressions
=2 expressions×11 (servers)

(d) 合流

$d\lambda_3=\pm\leftarrow dgc=\pm$; $dt_{1,3}=\pm$
$d\lambda_5=\pm\leftarrow dt_{1,5}=\pm$; $dt_{2,5}=\pm$
$d\lambda_7=\pm\leftarrow dt_{2,7}=\pm$; $dt_9=\pm$
$d\lambda_{11}=\pm\leftarrow dt_{4,11}=\pm$; $dt_{6,11}=\pm$
$d\lambda_{12}=\pm\leftarrow dt_{6,12}=\pm$; $dt_{8,12}=\pm$
$d\lambda_{14}=\pm\leftarrow dt_{10,14}=\pm$; $dt_{11,14}=\pm$
$d\lambda_{15}=\pm\leftarrow dt_{11,15}=\pm$; $dt_{12}=\pm$
$d\lambda_{16}=\pm\leftarrow dt_{13}=\pm$; $dt_{14}=\pm$
$d\lambda_{25}=\pm\leftarrow dt_{17}=\pm$; $dt_{19}=\pm$
$d\lambda_{26}=\pm\leftarrow dt_{22,26}=\pm$; $dt_{25}=\pm$
$d\lambda_{27}=\pm\leftarrow dt_{12,27}=\pm$; $dt_{24}=\pm$
$d\lambda_{29}=\pm\leftarrow dt_{26,29}=\pm$; $dt_{27,29}=\pm$
$d\lambda_{32}=\pm\leftarrow dt_{30}=\pm$; $dt_{31}=\pm$
$d\lambda_{35}=\pm\leftarrow dt_{27,35}=\pm$; $dt_{28}=\pm$
$d\lambda_{39}=\pm\leftarrow dt_{32}=\pm$; $dt_{34,39}=\pm$
$d\lambda_{40}=\pm\leftarrow dt_{34,40}=\pm$; $dt_{37,40}=\pm$
$d\lambda_{41}=\pm\leftarrow dt_{39}=\pm$; $dt_{40}=\pm$

subtotal : 68 expressions
=4 expressions×17 (servers)

(e) スループットと分岐確率

$dt_{1,3}=\pm\leftarrow dt_1=\pm$; $dr_{1,3}=\pm$
$dt_{1,5}=\pm\leftarrow dt_1=\pm$; $dr_{1,5}=\pm$
$dt_{2,5}=\pm\leftarrow dt_2=\pm$; $dr_{2,5}=\pm$
$dt_{2,7}=\pm\leftarrow dt_2=\pm$; $dr_{2,7}=\pm$
$dt_{4,10}=\pm\leftarrow dt_4=\pm$; $dr_{4,10}=\pm$
$dt_{4,11}=\pm\leftarrow dt_4=\pm$; $dr_{4,11}=\pm$
$dt_{6,11}=\pm\leftarrow dt_6=\pm$; $dr_{6,11}=\pm$
$dt_{6,12}=\pm\leftarrow dt_6=\pm$; $dr_{6,12}=\pm$
$dt_{8,9}=\pm\leftarrow dt_8=\pm$; $dr_{8,9}=\pm$
$dt_{8,12}=\pm\leftarrow dt_8=\pm$; $dr_{8,12}=\pm$
$dt_{10,13}=\pm\leftarrow dt_{10}=\pm$; $dr_{10,13}=\pm$
$dt_{10,14}=\pm\leftarrow dt_{10}=\pm$; $dr_{10,14}=\pm$
$dt_{11,14}=\pm\leftarrow dt_{11}=\pm$; $dr_{11,14}=\pm$
$dt_{11,15}=\pm\leftarrow dt_{11}=\pm$; $dr_{11,15}=\pm$
$dt_{15,18}=\pm\leftarrow dt_{15}=\pm$; $dr_{15,18}=\pm$
$dt_{15,20}=\pm\leftarrow dt_{15}=\pm$; $dr_{15,20}=\pm$
$dt_{20,21}=\pm\leftarrow dt_{20}=\pm$; $dr_{20,21}=\pm$
$dt_{20,23}=\pm\leftarrow dt_{20}=\pm$; $dr_{20,23}=\pm$
$dt_{22,26}=\pm\leftarrow dt_{22}=\pm$; $dr_{22,26}=\pm$
$dt_{22,27}=\pm\leftarrow dt_{27}=\pm$; $dr_{22,27}=\pm$
$dt_{26,28}=\pm\leftarrow dt_{26}=\pm$; $dr_{26,28}=\pm$
$dt_{26,29}=\pm\leftarrow dt_{26}=\pm$; $dr_{26,29}=\pm$
$dt_{27,29}=\pm\leftarrow dt_{27}=\pm$; $dr_{27,29}=\pm$
$dt_{27,35}=\pm\leftarrow dt_{27}=\pm$; $dr_{27,35}=\pm$
$dt_{28,30}=\pm\leftarrow dt_{28}=\pm$; $dr_{28,30}=\pm$
$dt_{28,31}=\pm\leftarrow dt_{28}=\pm$; $dr_{28,31}=\pm$
$dt_{34,39}=\pm\leftarrow dt_{34}=\pm$; $dr_{34,39}=\pm$
$dt_{34,40}=\pm\leftarrow dt_{34}=\pm$; $dr_{34,40}=\pm$
$dt_{37,40}=\pm\leftarrow dt_{37}=\pm$; $dr_{37,40}=\pm$
$dt_{37,38}=\pm\leftarrow dt_{37}=\pm$; $dr_{37,38}=\pm$

subtotal : 120 expressions
=4 expressions×30 (servers)

(f) 分岐確率

for $i=\{s1, s2, s4, s6, s8, s10, s11, s15, s20, s22, s26, s27, s28, s34, s37\}$
$dri, j=\mp\leftrightarrow dri, k=\pm\ (j\neq k)$

subtotal : 30 expressions
=2 expressions×15 (servers)

total : 266 expressions

図-7 各サーバの入出力の定性式
図4 (a)～(f)の定量式に対応，(d)の+，(e)の*は；(or) になる

304 本である．ボトルネックの改善を行う場合には，これらの式に加えてサーバ間の入出力情報(**図-7 の入出力の定性式**)：266 本 を与えなければならない．しかし，これらを全て組み合わせて定性挙動推論により解くと，状態数の爆発が起きて効率的ではない．

4.4 経験則に基づく定性挙動推論

全ての立式（計 570 本）を組み合わせて解くのではなく，専門家の経験則（浅い知識）を導入する．それは，サーバの結合による部分形状ごとの性能パラメータ間の増減関係を表した定性改善式である．

著者らは，この定性改善式を扱い定性推論によりチューニングプランを列挙するエキスパートシステム BDES (Bottleneck Diagnosis Expert System) を開発した．

BDES には，ボトルネック改善のための 9 つの知識（下流への影響，直列，合流，分岐，ループなど）がある．この知識により，使用する性能パラメータの個数が，パラメータ全てに着目する場合よりかなり減少する．

図-8 は，ボトルネックサーバの改善のためそのμを増加するとスループットが増加するため，下流の過大なρを減少させる知識である．

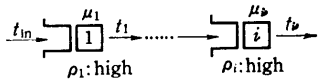
図-9 の直列型 (tandem) のサーバ $s2$ の ρ_2 を改善するために，t_1 と μ_2 のバランスを変える必要がある．そのために接続する他のサーバのパラメータの大きさを変える．図-9 は，一つ上流のサーバ $s1$ がボトルネックでない場合，t_{in} を減少して ρ_2 を改善する知識である．

図-10 は，合流の場合，合流入力の大きさにより改善すべきパラメータを決める知識である．

図-11 の分岐の知識は，分岐確率を修正する場合と，直上流のスループットを修正する場合に分かれる．

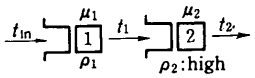
図-12 の知識はまず，ループの機能を変えないために，ループからの分岐確率の修正を禁止する．また，ループからのスループットを受け取るボトルネックサーバ $s3$ を改善するためには，ループ内部のパラメータではなくループへの到着量のみを減少する．すなわち，$d\rho_3 = - \leftarrow dt_{in} = -$ という定性改善式のみを使用する知識である．

図-13 は，ループ内のボトルネックを改善するために，ループへの到着率を減少する知識である．ボトルネックサーバのサービス率を増加するとスループットが増加し，これがループして自身



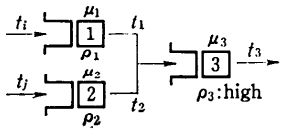
ボトルネックサーバのサービス率を増加する場合，
下流のサーバの稼動率を減少する必要あり
$d\rho_1 = - \leftarrow d\mu_1 = +$
$d\rho_1 = - \leftarrow d\rho_i = -$

図-8 下流への影響



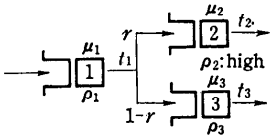
$d\rho_2 = - \leftarrow dt_{in} = -$

図-9 直列型の知識



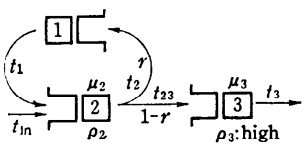
$t_1 > t_2$ の場合
$d\rho_3 = - \leftarrow dt_1 = -$
$t_1 = t_2$ の場合
$d\rho_3 = - \leftarrow dt_1 = -$
$d\rho_3 = - \leftarrow dt_2 = -$

図-10 合流型の知識



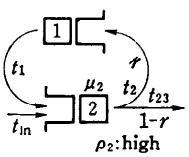
分岐確率が修正可能な場合
$d\rho_3 = - \leftarrow d\gamma = -$
s_3 とその下流のサーバの稼動率にも留意
修正不可能な場合
$d\rho_3 = - \leftarrow dt_1 = -$

図-11 分岐型の知識



ループからの分岐確率は修正禁止
$d\rho_3 = - \leftarrow dt_{in} = -$

図-12 ループ直後のボトルネック



サービス率の増加は禁止
$d\rho_3 = - \leftarrow dt_{in} = -$

図-13 ループ内のボトルネック

への入力の一部となり，ボトルネックの改善にはつながらない．

以上の知識と定性改善式は，専門家のもついわゆる**浅い知識**である．**4.2** で述べたサーバごとの定性挙動式，定性改善式及び入出力の定性式はいわゆる**深い知識**である．一般に浅い知識の妥当性は簡単には証明ができないといわれているが，文献 [ITO 90 a] に示すとおり，本節の浅い知識は深い知識により証明可能である．

図-3 の QN 4 のボトルネックサーバ $s20$ に対する，BDES の定性挙動推論プロセスを図-14 に示す．図-6 と図-7 に比べ式の本数はかなり減少し，状態数の爆発は抑制される．

図-14 の最上部のブロック 1 は，ボトルネック $s20$ を改善したいというチューニングのゴール ($d\rho_{20}=-$) である．ブロック 1 からブロック 2・3 への分岐は $s20$ に対する定性改善式 (**4.2**) の適用である．ブロック 2 は，図-8 の下流への影響に関する知識である．ブロック 3 は，図-11 の分岐型の知識によりブロック 4 と 5 に分岐する．以下同様に知識が適用される．

以上の推論の結果，ブロック 2, 4, 9, 15, 20, 22, 24 の下に，改善のゴール ($d\rho_{20}=-$) を実現するための択一的に選択可能な定性的なチューニングプラン (plan 1～7) が示されている．たとえば，プラン 1 は，$s20$ の改善のために，$s20$ 自体という局所的なサービス率の増加のほかに，広域的に $s27$, $s28$, $s40$ の稼動率の減少を指示している．

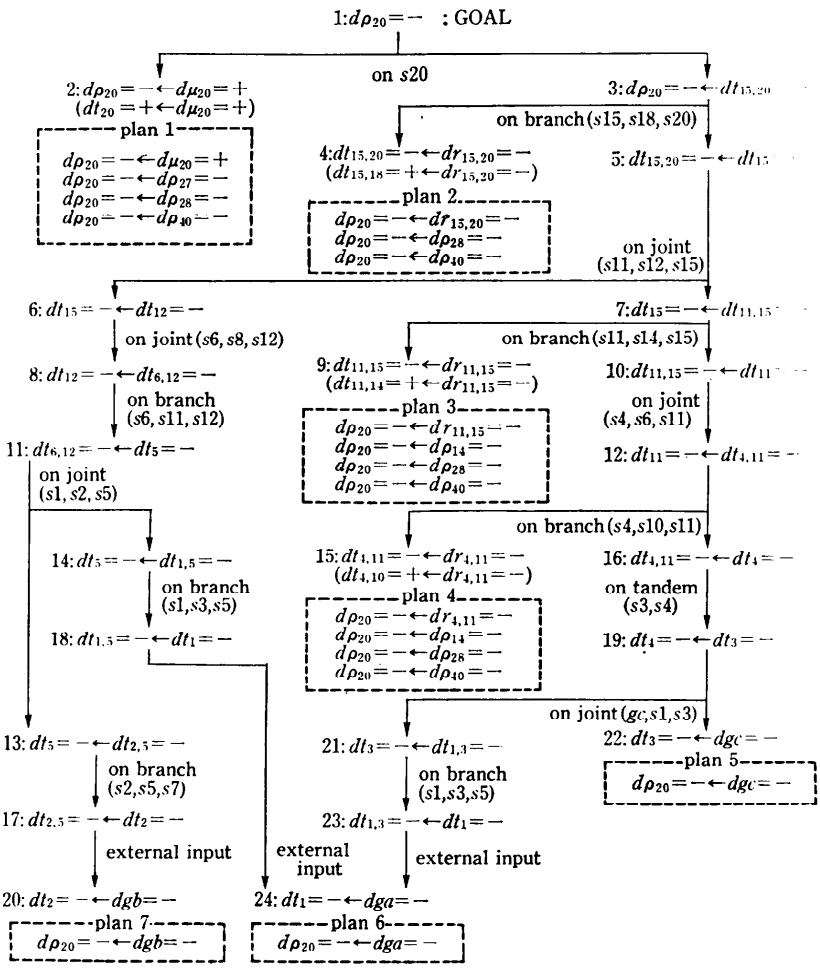


図-14 QN 4 の改善のための定性推論

5. 定量的なパラメータチューニング

著者らは，ボトルネック改善のための性能パラメータの定量推論を行うエキスパートシステム BIES (Bottleneck Improvement Expert System) を開発した．図-14 に示したとおり，一つのボトルネックに対して定性的なチューニングプランが複数得られたが，これらは択一的に選択可能である．ユーザが選んだ一つの定性チューニングプランに対して，BIES は改善後のパラメータの定量値を推論する．すなわち低いサービス率を定量的に増加させたり，高い到着率や高い分岐確率を定量的に減少させる．さらに，このような一つのサーバのパラメータチューニングの後，ほかのサーバの新しい到着率や新しいスループットを自動的に算出する．

図-15 は，図-14 の BDES の 7 つの定性的なチューニングプラン全てに対して定量的なチューニングを行ったプロセスである．定性チューニングプラン 1 に対して，BIES がきめ細かな定量推論を行うと定量チューニングプランが 8 個提示される．また，定性チューニングプラン 5 は外部入力 gc を減少することを示しているが，BIES の定量チューニングプランでは，きめ細かな定量推論を行うと，この gc の改善のみでは $s20$ のボトルネックを解消できないことが示されている．定性推論を行わないで直接，複数の定量チューニングプランを列挙することは必ずしも容易ではない．このように定性推論と定量推論を相補的に使用することが有効である．

図-14 の全ての定性チューニングプランや，このおのおのに対する図-15 の全ての定量チューニングプランを，BDES と BIES は順次全て出力する．たとえばあるパラメータが変更禁止の固定パ

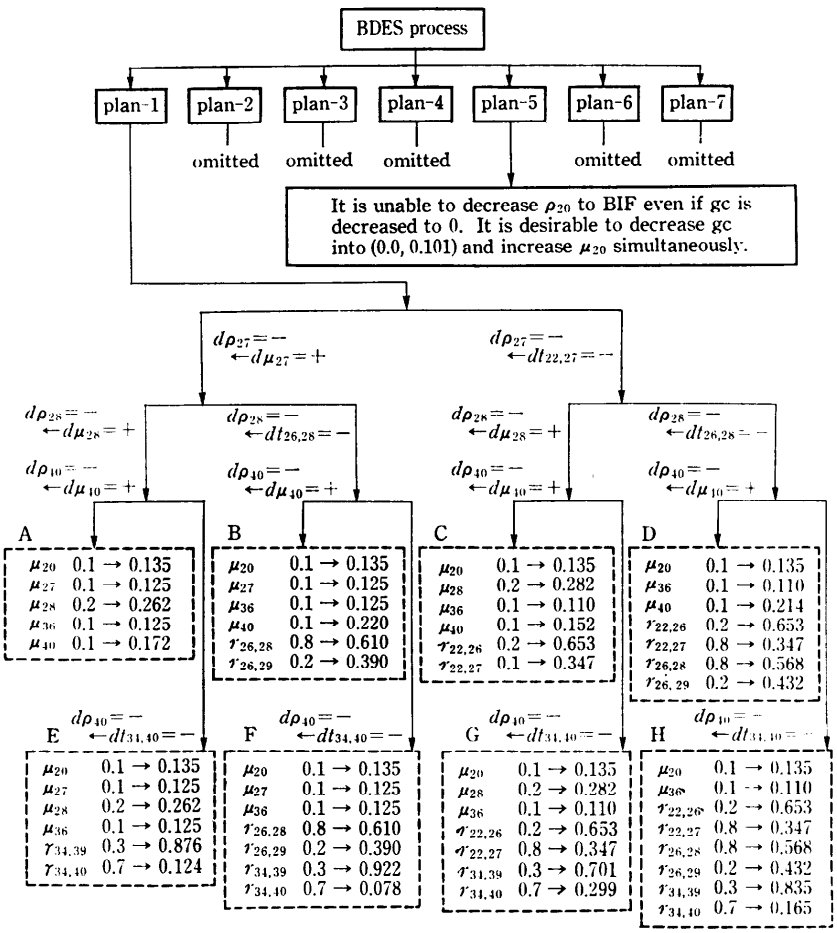


図-15　QN 4 の改善のための定量推論

ラメータであれば，そのような改善プランをユーザは選択せず，次のプランの提示を促す．また選択したプランが不適切であるとユーザが判断したら，BDES と BIES はパラメータを元の値に自動的に復帰させるバックトラックを行う．

6. おわりに

本解説では，定性推論と定量推論を組み合わせて QN を例にあげてそのボトルネックを改善するためのパラメータチューニングの方法を解説した．この方法は，複数種類のエンティティが流れる多重フロー型 QN，エンティティ同士が同期を取る同期型 QN，時間／確率ペトリネットなどのボトルネック改善にも適用した[MAT90],[SHI90]．定性推論に基づいて並列オブジェクトプログラムの性能デバガにも適用した[HON90ab]．

待ち行列ネットワーク理論では，外部からの到着率，サーバのサービス率，分岐確率を与えて，各サーバの稼動率や待ち行列長を求めるという**順問題** (direct problem：初期条件を与えても対象を支配する方程式を解く問題）を解くことが多い．パラメータチューニングとは，本解説のようにたとえば稼動率を 0.7 より小さくするためには，到着率，サービス率，分岐確率などをどのように設定すればよいかという**逆問題** (inverse problem)[SUU86],[SIM90] を解くことである．待ち行列ネットワークの逆問題を解析的な最適化手法で解く方法も数少ない(たとえば [WAD90])．本解説の手法は，この逆問題を解析的ではない定性推論で解くヒューリスティックな解法である．パラメータチューニングを逆問題としてとらえ定性推論により解く方法は，ほかの分野でもあまり報告されていない．しかし，システムの構造がネットワークやグラフで表現されその上を何かが伝播するシステム（力学／電気回路，経済システムなど数多い）のパラメータチューニングにきわめて有効であると考えている．

謝辞 BDES と BIES の共同研究者である上智大学院生沢村淳氏(現，ソニー(株))，志田圭介氏に感謝します．

参考文献

[APT 86] Apte, C. et al.: Using Qualitative Reasoning to Understand Financial Arithmetic Proc. AAAI (1986).

[BOB 85] Bobrow, D. G. et al. ed.: Qualitative Reasoning about Physical Systems, MIT Press (1985).

[DEK 85] De Kleer, J.: How Circuits Work, in [BOB 85].

[GEL 80] Gelenbe, E. et al.: Analysis and Synthesis of Computer Systems, Academic Press (1980).

[HON 90a] 本位田，内平，伊藤：定性推論と定量推論による待ち行列ネットワークの性能改善エキスパートシステム BDES & BIES—その5 並列プログラムへの適用—，4th NC JSAI (July 1990).

[HON 90b] Honiden, S. and Itoh, K. et al.: Rapid Prototyping Method for Performance Design in Realtime Systems, InfoJapan '90, Vol. 1, pp. 103-110 (Oct. 1990).

[ITO 89a] Itoh, K. et al.: Knowledge-Based Parameter Tuning for Queueing Network Type System—A New Application of Qualitative Reasoning, Proc. CAPE '89, pp. 209-216 (Oct. 1989).

[ITO 89b] Itoh, K. et al.: Knowledge-Based Diagnosis on Simulation Result for Queueing Network, Proc. Beijing I. C. on System Simulation and Scientific Computing (Oct. 1989).

[ITO 90a] 伊藤，本位田，沢村，志田：定性推論と定量推論を導入した待ち行列ネットワークのボトルネック診断と改善法，人工知能学会誌，Vol. 5, No. 1 (1990).

[ITO 90b] Itoh, K., Honiden, S. et al.: Role of Qualitative and Quantitative Reasoning in Diagnosis and Improvement for Queueing Network Bottleneck, InfoJapan '90, Vol. 2, pp. 171-178 (Oct. 1990).

[KLE 75] Kleinrock, L.: Queueing Systems, John Wiley & Sons, Inc. (1975).

[MAT 90] 松永，志田，本位田，伊藤：定性推論と定量推論による待ち行列ネットワークの性能改善エキスパートシステム BDES & BIES—その4 多重フロー型待ち行列ネットワークのエキスパートシステム—4th NC JSAI (July 1990).

[MIZ 89] 溝口，古川，安西編：定性推論，共立 (1989).

[NIS 88] 西田：定性推論に関する最近の研究動向，情報処理，Vol. 29, No. 9 (Sep. 1988).

[NIS 89] 西田：定性推論の基礎，人工知能学会誌，Vol. 4, No. 5, pp. 522-527 (1988).

[RAJ 84] Rajagopalan, R.: Qualitative Modeling in the Turbojet Engine Domain, Proc. AAAI (1984).

[SAW 89a] 沢村，本位田，伊藤：定性推論を導入した待ち行列ネットワークのボトルネック診断，情報処理学会知識工学と人工知能研究会 (Jan. 1989).

[SAW 89b] 沢村，本位田，志田，伊藤：知識工学的手法を用いた待ち行列ネットワークのボトルネック診断，情報処理学会論文誌，Vol. 30, No. 8, pp. 990-1002 (Aug. 1989).

[SHI 90] 志田，伊藤，本位田，早瀬：定性／定量推論による同期型待ち行列ネットワークのボトルネック診断と改善，情報処理学会知識工学と人工知能研究会 (Jan. 1990).

[SIM 90] コンピュータ利用による逆問題解析，シミュレーション，9.1 (Mar. 1990).

[SUU 86] 逆問題，数理科学，274 (Apr. 1986).

[WAD 90] 和田，米田：待ち行列網の逆問題とその設備計画への応用，SSOR 90 (1990).

(平成2年9月10日受付)

伊藤　潔（正会員）

1951年生．1974年京都大学工学部情報工学科卒業．1979年同大学院情報工学専攻博士課程修了．京都大学工学博士．1979年より上智大学に勤務．1985年より助教授．現在，同理工学部一般科学研究室情報科学部門所属．ソフトウェア工学，シミュレーション手法，定性理論の待ち行列ネットワークへの応用，三面図からのソリッドモデルの自動合成法の研究に従事．情報処理学会学会誌編集委員．「仕様記述の効率的適用と評価」研究グループ幹事．ISO/TC 184/SC 2 WG 3（産業用ロボットの安全性）議長団．著書「ソフトウェア開発のためのプロトタイピングツール」（共著），「システムプログラム」，訳書「並行処理と Unix」（共訳）．人工知能学会，IEEE，ACM など各会員．

本位田真一（正会員）

1953年生．1976年早稲田大学理工学部電気工学科卒業．1978年同大学院理工学研究科電気工学専攻修士課程修了．工学博士．同年（株）東芝入社．現在，同社システム・ソフトウェア技術研究所研究主務．早稲田大学非常勤講師．主として，ソフトウェア工学，人工知能の研究に従事．ソフトウェアの基礎理論に興味をもつ．1986年情報処理学会論文賞受賞．著書「ソフトウェア開発のためのプロトタイピング・ツール」（共著），「KE 養成講座② エキスパートシステム基礎技術」（共著），「オブジェクト指向システム分析」（共訳），人工知能学会，日本ソフトウェア科学会，電気学会，AAAI 各会員．