人　工　知　能　74−10
（１９９１．　１．　１７）

同期型待ち行列ネットワークのボトルネックに対する定性的なパラメータチューニング

志田圭介　（上智大学大学院　理工学研究科　機械工学専攻　博士前期課程）
本位田真一　（(株)東芝　システム・ソフトウェア技術研究所）
伊藤　潔　（上智大学　理工学部　一般科学研究室　情報科学部門）

同期型待ち行列ネットワークの性能パラメータの測定平均値に対して，定性推論でパラメータチューニングする手法を述べる．この手法に基づいて，ボトルネックサーバを同定しその要因を解明し，択一的に選択可能な定性的な改善プランを提示するエキスパートシステム“Bottleneck Diagnosis Expert System for open Synchronized queueing network: BDES-S”（[SHI90]）を開発した．BDES-Sの定性的な改善プランはパラメータ間の増減関係のみを定めるものである．

Parameter tuning on bottleneck of synchronized queueing network by qualitative reasoning

Keisuke Shida*, Shinichi Honiden**, Kiyoshi Itoh***

*: Department of Mechanical Engineering,
Faculty of Science and Technology, Sophia University
Kioi-cho 7-1, Chiyoda-ku, Tokyo 102, Japan
**: Systems and Software Engineering Laboratory, Toshiba Corporation
Yanagi-cho 70, Saiwai-ku, Kawasaki 210, Japan
***: Laboratory of Information Science, Natural Sciences Center,
Faculty of Science and Technology, Sophia University
Kioi-cho 7-1, Chiyoda-ku, Tokyo 102, Japan

This paper describes Bottleneck Diagnosis Expert System for Synchronized queueing network (BDES-S). On the basis of qualitative reasoning, BDES-S can carry out parameter tuning in order to diagnose and improve bottlenecks of synchronized queueing networks. BDES-S can produce several alternative qualitative improvement plans for one bottleneck server.

1 はじめに

並行システムは，多くの構成要素から成り，これらの構成要素が自律的に稼動し，必要に応じてある構成要素から他の構成要素へ同期信号やメッセージ等の情報を送受するシステムである．このような並行システムの性能を分析するためのモデルとして，通常の待ち行列ネットワーク([KLE75])に同期型サーバを導入した同期型待ち行列ネットワークを用いる．

対象の並行システムに過大な負荷がかかっている ― そのモデルである同期型待ち行列ネットワークは非定常(過負荷)状態にある，あるいはその可能性をもつ ― 状況を想定する．この対象システムにはボトルネックがある．通常の待ち行列ネットワークでは，稼動率あるいは待ち行列長が過大なサーバがボトルネックとなる．同期型待ち行列ネットワークでは，このようなサーバに加えて，2節で示す通り，その2つの入力量の間にバランスがとれていない同期型サーバもボトルネックとなる．

ボトルネックサーバの効果的な解消のために，同期型待ち行列ネットワークの性能パラメータの測定平均値に対して，定性推論でパラメータチューニングする手法を述べる．この手法に基づいて，ボトルネックサーバを同定しその要因を解明し，択一的に選択可能な定性的な改善プランを提示するエキスパートシステム“Bottleneck Diagnosis Expert System for open Synchronized queueing network: BDES-S”([SHI90])を開発した．BDES-Sの定性的な改善プランはパラメータ間の増減関係のみを定めるものである．

通常の待ち行列ネットワークのボトルネック改善を行うBDESについて既に著者らの論文([SAW89a], [SAW89b], [ITO89,90])で述べた．BDESは，定性推論([BOB85], [MIZ89], [NIS88,89])を待ち行列ネットワークに適用した最初のものであった．本稿のBDES-Sは，同期型待ち行列ネットワークに適用するものである．

BDES-Sによる診断と改善の流れを述べる．ネットワークの形状は固定パラメータで他は可変パラメータ(診断後に変更可能)とする．たとえば，サービス率(μ)は各サーバの能力として測定前に与えられるが，チューニング後に可変であるとする．

性能パラメータを測定した後，稼動率(ρ)や待ち行列長(q)が過大であるサーバ群，および入力量のバランスの悪い同期型サーバ群をBDES-Sが列挙する．評価者はこのうち1つのサーバを選択する．BDES-Sは，ネットワークの部分形状とその部分形状内の性能パラメータ値に関する定性式で表現できる知識と，この定性式の適用順序の制御に関する知識をもつ．BDES-Sが分析する性能パラメータは，ばらつきなどの値を用いない平均値であり測定上簡単なデータである．BDES-Sは，1つのボトルネックサーバの複数の定性的な改善プランを列挙する．評価者はこのうち1つの改善プランを選択し，その改善プランに示された性能パラメータの修正を行う．他のボトルネックサーバの対しても，評価者は同じ作業を行う．

2 定義

2.1 同期型待ち行列ネットワークの定義

本稿で対象とする同期型待ち行列ネットワークは，処理されるエンティティが全て外部から到着し，それらは処理済みになると外部へ出る，オープン型の待ち行列ネットワークである．また，エンティティは外部から複数のサーバに到着することが可能である．待ち行列ネットワークの内部では，それらは区別して扱わないため，種類に応じて途中の分岐確率やサービス率が異なることはない．すなわち，待ち行列ネットワークの内部では，エンティティは1種類と考える．また，この同期型待ち行列ネットワーク中には，通常のFCFS(First Come First Served)のサーバ(ノーマル型サーバと称する)のほかに，2.3 節で示すような3種類の同期型サーバが混在している．

2.2 性能パラメータの定義

各々のサーバに対して，λ，μ，t，ρ，q，rを以下のように定義する．

- λ：単位時間当りに待ち行列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/組数(サービス率：servicing rate)
- t：単位時間当りにサーバから出て来るエンティティの平均個数(スループット：throughput)
- ρ：1つのサーバが実際にエンティティを処理している時間の割合(稼動率：utilization rate)
- q：待ち行列内に待機しているエンティティの平均個数(待ち行列長：queue length)
- r：エンティティがあるサーバを出た後，下流の複数のサーバのいずれに分岐するかを定める確率(分岐確率：branching probability)

2.3 サーバの種類

同期型待ち行列ネットワークは，次の4種類のサーバが複数個結合したものである(以下の(2)～(4)を同期型サーバと総称)．サーバの定義を例と共に述べる．

(1)ノーマル型サーバ(Table 1 項目1)

あるタスクの実行をこのサーバで表現する．タスク実行時間を，このサーバのサービス時間($1/\mu$)とする．

このサーバは，1つの入力路および1つの出力路を持つサーバである．このサーバの待ち行列に到着したエ

ンティティは，FCFSの規則に従ってサーバに入り，サービス時間を経過した後サーバを出る．

サーバの入力は上流の複数のサーバの出力の合流であってよい．しかし，これらの複数のサーバから合流するエンティティは，ただ一つのFCFSの待ち行列に入る．また，このサーバの出力は下流の複数のサーバに分岐しそれらの入力となってよい．この分岐は確率的に行われる．

(2)スプリット(split)型サーバ(Table 1 項目2)

タスクの生成やメッセージの送信をこのサーバで表現する．タスクの生成やメッセージの送信に要する時間をこのサーバのサービス時間とする．並行的に実行可能なもう1つのタスクや，タスクへのメッセージが生成される．

このサーバは，1つの入力路および，2つの出力路を持つ同期型サーバである．待ち行列に到着した1つのエンティティは，FCFSの規則に従ってサーバに入り，自分の複製である1つの子エンティティを生成する．この複製には，ある有限のサービス時間が必要である．この後，親エンティティと子エンティティは同時に，各々別の出力路から出て行く．すなわち，2つの出力量は常に等しい．

(3)マージ(merge)型サーバ(Table 1 項目3)

タスクの消滅やメッセージの受信をこのサーバで表現する．タスクやメッセージの消滅に要する時間をこのサーバのサービス時間とする．それまで実行していたタスクの1つや，別のタスクからのメッセージが消滅する．

このサーバは，2つの入力路および，1つの出力路を持つ同期型サーバである．別の入力路から到着するエンティティは，それぞれの待ち行列を形成する．一方の待ち行列に到着したエンティティは，もう一方の待ち行列のエンティティを探索する．もし，相手エンティティが複数個存在する場合には，FCFSの規則に従う．相手エンティティが存在すると，2つのエンティティは同時にサーバに入り，1つに融合される(または一方のエンティティが消滅する)．この融合には，ある有限のサービス時間が必要である．この後，融合されたエンティティは，サーバを出て行く．相手エンティティが存在しない場合，相手エンティティが到着するまで，そのまま待機し続ける．

(4)マッチ(match)型サーバ(Table 1 項目4)

タスク間の同期，通信をこのサーバで表現する．タスク同士が同期をとりながらタスクを実行したり，タスク同士が通信するのに要する時間を，このサーバのサービス時間とする．2つのタスクが協同して仕事を行う時，歩調を合わせるために信号のみをやりとりするタスク間同期や，データのやりとりを伴うタスク間通信が行われる．

このサーバは，2つの入力路および，2つの出力路を持つ同期型サーバである．別の入力路から到着するエンティティは，それぞれ別の待ち行列を形成する．一方の待ち行列に到着したエンティティは，もう一方の待ち行列の中のエンティティを探索する．もし，相手エンティティが複数個存在する場合には，FCFSの規則に従う．相手エンティティが存在すると，2つのエンティティは歩調を揃えて同時にサーバに入り，ある有限のサービス時間を経過した後，2つのエンティティは同時に各々別の出力路からサーバを出て行く．すなわち，2つの出力量は常に等しい．相手エンティティが存在しない場合，相手の待ち行列に到着するまで，そのまま待機し続ける．

2.4 各サーバの性能パラメータ

各サーバの性能パラメータ間の関係をTable1に示す．

2.5 ボトルネックの定義

待ち行列ネットワークの中で稼動率が1に極めて近いサーバにボトルネック(bottleneck)がある．そのボトルネックサーバの待ち行列長は無限に成長する恐れがある．このようなボトルネックサーバが存在する場合，待ち行列ネットワークは非定常(過負荷)状態になっている．

稼動率が1に近くなくても過大なサーバはボトルネックとなる可能性がある．サーバへの到着率やサーバのサービス率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．このため，フェールセーフの考え方で，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診断する．

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，測定終了以降にエンティティが過大に到着してその稼動率が更に上昇する危険性がある．

さらに，マージ型・マッチ型サーバにおいて2つの到着率が異なる場合，サーバで実際に処理される量は小さい方の到着率に等しくなる．もし仮に，平均の到着率の差が極僅かであっても，この差が積算され到着率が大きい方のエンティティの形成する待ち行列は無限に成長する危険性がある．

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．しかし，ボトルネックの発生を事前に防止するというフェールセーフの考え方で，測定終了時に平均待ち行列長が1以上のサーバ，また，マージ型・マッチ型サーバで2つの到着率が異なるサーバ全てについて，ボトルネックの可能性があると診断する．

以上のような基準でボトルネックの可能性があると

診断されたサーバが存在する場合，待ち行列ネットワークは，非定常(過負荷)状態となる危険性がある．

3 待ち行列理論，およびペトリネット理論との関係

3.1 待ち行列理論との関係

一般のオープン型待ち行列ネットワークでエンティティが1種類の場合，定常状態ではトラフィック方程式が成り立つ(例えば文献[GEL85])．しかし，BDESの論文([SAW89a]，[SAW89b]，[ITO89,90])で示したように，ノーマル型サーバだけで構成されている場合でも，ボトルネックが存在する時には，ネットワークは定常状態ではなくこの方程式は成立しない．さらに，同期型サーバを導入した場合には，ボトルネックが存在していない時でさえも，簡単な方程式で表すことは容易ではない．すなわち，このような場合エキスパートシステムの導入が有効である．

3.2 ペトリネット理論との関係

一般に，並行システムのモデル化には，ペトリネット([PET84])が適している．ペトリネットモデルには，2つの型のノードがある．システムの中で起こる動作(事象)を表すトランジションと，その動作が起こるための条件を表すプレースである．プレースの中にトークンを置くことが，その条件の成立に対応する．

本稿で導入している4種類のサーバは，全てサーバをトランジション，プレースをバッファ，エンティティをトークンに置き換えることによってペトリネットグラフとして表現可能である．

しかし，より実際の対象をより正確にモデル化するためには，時間の概念が必要である．初期のペトリネット理論には，時間の概念は存在しない．1つのトランジションは，その入力プレースにトークンが存在していれば発火が可能であるが，その発火に時間はかからない．すなわち，発火後直ちに(入力プレースからトークンを取り去り)出力プレースにトークンを投入する．このため，ペトリネット理論に時間や確率変数の概念を導入した時間ペトリネットや確率ペトリネット([RAZ85])が考案された．そこでは，トランジションの発火時間間隔，トランジションの発火に要する時間やプレースの滞在時間などにある確率分布に従う有限時間の間隔を持たせるなど様々な形で時間の概念が導入されている．

本稿で導入した同期型待ち行列ネットワークは，一般の待ち行列理論で用いられる待ち行列ネットワークに対してペトリネット理論による同期の概念を導入したものであり，時間ペトリネットに変換できる．同期型待ち行列ネットワークに言及したものとして Florin, G.らの論文がある([FLO89])．彼の論文では，確率ペトリネットを同期型待ち行列ネットワークとみなしている．本稿では，一般の待ち行列ネットワークに2.3節の同期型サーバを導入したものを同期型待ち行列ネットワークと呼ぶ．

4 単一サーバの定性挙動式と定性改善式

エキスパートのヒューリスティクスとして稼動率0.7をボトルネックランドマーク(BL)とし，$\rho \geqq BL$ ならば，ボトルネックの可能性があるとする．また，待ち行列長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，$q \geqq QBL$ ならば，ボトルネックの可能性があるとする．

これを定性推論記法で表現する．以下，$[\rho]$と$[q]$は，各々BLとQBLを原点として評価した定性値である．

- $[\rho] = +$ ： ボトルネックの可能性がある．
  この時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．
  この時同様に $dq = +$ と考える．

定性推論の考え方に基づいて，個々のサーバの挙動を表す定性式を構成する．ボトルネックの可能性がない場合とある場合で用いる式の種類を変えることに特長がある．これらの式を個々のサーバの定性挙動式(qualitative behaviorexpression:QLBE)と呼ぶ．ボトルネックの可能性がある場合を特に定性改善式(qualitative bottleneck improvement expression: QL-BIE)と呼ぶ．

各々のサーバにおけるQLBEとQL-BIEをTable 2に示す．$[\rho]=-$の項目はQLBE，$[\rho]=+$の項目はQL-BIEである．例えば，Table 2-1の1段目の項目は，ノーマル型サーバにおいて$[\rho]=-$の時には，λの増減に対してρ，tが共に追随することを示す．式で表すと，

$$d\rho = \pm,\ dt = \pm \quad \leftarrow \quad d\lambda = \pm$$

となる．同様に，Table 2-3の3段目の項目は，マージ型サーバにおいて，$[\rho]=-$，$[q_A]=-$，$[q_B]=+$である時，例えば，λ_Bのみを減少するとq_Bが減少し，ρ，t，q_Aには変化がないことを示す．また，λ_A，λ_Bを共に減少するとq_Bが減少し，ρ，tが減少，q_Aには変化がないことを示す．

あるサーバのλを変化させるためには，それより上流のサーバのtを変化させなければならない．あるサーバのtを増加すると，下流のサーバへのλの増加につながるので，下流のサーバがボトルネックとなる可能性があり，そうならないようにしなければならない．すなわち，ボトルネック改善のためには，全てのサーバに対して立式して，それらを組み合わせて解かなければならない．

Fig.1に本稿で扱う例題待ち行列ネットワーク("SQN1"とよぶ)を示す．この性能パラメータはシミュレーションパッケージによって測定されたものである．SQN1に対する定性挙動式・定性改善式の個数は，約350本である．ボトルネックの改善を行なう時には，これらの式以外にサーバ間の結合情報や分岐確率などの情報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．

5 部分形状とその性能パラメータに基づくボトルネック改善のための定性改善式

5.1 ボトルネック改善のための浅い知識

全てのサーバの立式を組み合わせて解く方法ではなく，サーバ間の結合情報を取り込んだ部分形状を踏まえて，部分形状毎に内部の性能パラメータ間の増減関係を表した式で立式して全体の式の個数を減らす．

5.1.1 ボトルネックサーバの初期診断のための知識

ボトルネックサーバを初期診断するためのヒューリスティクな知識を挙げる．

1) ρが一番高いサーバにボトルネックの可能性がある．
2) ρが0.9以上のサーバにボトルネックがある．
3) ρが0.7以上のサーバにボトルネックの可能性がある．
4) ρが0.7より小さいが，qが1.0以上のサーバにボトルネックの可能性がある．
5)マージ型・マッチ型サーバでは，2つの入力路へのλが異なるサーバには，その ρやqの値にかかわらずボトルネックの可能性がある．

5.1.2 ネットワーク変形のための知識

本稿で導入している同期型待ち行列ネットワークには，前述の通り4種類のサーバが混在している．4節末尾で述べたように全てのサーバの立式を組み合わせて解く方法を採用すると，式の数が多くなり解くのに時間がかかってしまう．これに対して，サーバ間の結合情報を取り込んだ部分形状を踏まえて，部分形状毎の主要な性能パラメータ間の増減関係についての知識を表した式で立式すると式の個数は減少する．しかし，この時，4種類のサーバを組み合わせた構造全てを考えると式の種類は膨大となる．実際のシステムにおいては，ノーマル型サーバの占める割合が他のサーバに比べて大きいと考えられ，数多くの種類の式を羅列しても効率的でない．

従って，同期型待ち行列ネットワークを一般の待ち行列ネットワークに変形し，その変形された待ち行列ネットワークに対してボトルネックの診断を行う．この変形は，同期型待ち行列ネットワークのボトルネック診断のために一時的に行う変形である．さらに，変形は診断前に行うのではなく，診断の過程で行う．診断の過程で同期型サーバを訪れた時点で4節に示した定性式の情報を保持したまま，このサーバをノーマル型サーバに変形する．この変形を行いながら，ノーマル型サーバのみから成る一般の待ち行列ネットワークとして診断を行う．

ネットワーク変形のための知識をTable 3に示す．例えば，上流への診断の過程でスプリット型サーバを訪れ，下流診断が必要となった場合(そのtの変化によってその下流の窓口群に何らかの悪影響を与えるかを調査する場合)に適用される知識がTable 3-1に示されている．スプリット型サーバの2つの下流窓口群をそれぞれ下流窓口群とする，2つのノーマル型サーバに変形し，それぞれ別のネットワークと見なす．また，上流への診断の過程でマージ型サーバを訪れた場合に適用される知識をTable 3-2に示す．ここで，このマージ型サーバの2つのλに差がある場合(Table 3-2上段に対応する)には，そのλのどちらか一方が原因であると診断する．(ここではλ_Aが原因になっている．)次に，このマージ型サーバs3を，原因とされたλ_Aをスループットとしているサーバs1のみを上流に持つノーマル型サーバに変形し，上流への診断を継続する．λ_Bをスループットとしているサーバs2は，このtが外部に出力されているものと見なす．

5.1.3 ボトルネックの要因診断のための知識

Table 4は，部分形状とその内部のパラメータ値によるボトルネック改善のための14個の知識を示す．これらは対象とする同期型待ち行列ネットワークがノーマル型サーバのみからなる待ち行列ネットワークに変形された後に適用される．Table 4の第3項の場合の直接的な改善法は，直列型(tandem)のサーバs2のボトルネックを改善するために，t1とμ2のバランスを変えることである．しかし，その根本的な改善法は，接続する他のサーバのパラメータの大きさを変えることである．たとえば第3項の知識3に示すことは，1つ上流のサーバのt_{1n}を減少させることである．また，第2項の知識2に示す通り，μを増加する時には，下流の過大なρを減少させる必要がある．

Table 4の知識と定性改善式の示すことは，待ち行列ネットワークの部分形状に着目すれば，ボトルネックを改善するために着目する性能パラメータの個数が，サーバのパラメータに全て着目する場合よりかなり少なくなることである．例えば，第7項の知識7に示す通り，ループに関係する形状で，ループからのスループットを受け取るサーバs3がボトルネックとなる可能性

がある場合，このボトルネックを改善するためには，ループ内の性能パラメータではなくループへの到着量のみを減少すればよい．すなわち，このボトルネックを改善するためには，

$$d\rho 3=- \leftarrow dt_{in}=-$$

という定性改善式のみを使用すればよい．

以上の知識1～9は，ノーマル型サーバのみからなる待ち行列ネットワークに対して適用したものと同じ知識である．一方，変形前にマージ型・マッチ型であるサーバについては，過大な待ち行列を解消するために，変形後のノーマル型サーバのλを増加しなければならないこともある．あるサーバのλを増加するためには，それより上流のサーバのｔを増加させなければならない．すなわち，あるサーバのλの増加は，その上流のサーバのρの増加につながり，これらのサーバがボトルネックとなる可能性を生じる．従ってこのようなボトルネックの改善のためには，知識1～9とは異なる増加用のTable 4の第10項以降の知識10～14を適用する．

マージ型・マッチ型サーバにおいて，一方のλAの形成するｑAが過大な場合，そのλAを減少することによって，ｑAの減少が可能である．しかし，λAともう一方のλBを同時に，異なった割合（dλA>dλB）で減少しても，ｑAの減少が可能である(この際，ｑBには変化がない)．小さい方のλBを増加する場合についても全く同様である．すなわち，一方のｑが過大である場合には，４種類の改善法(大きい方のλを減少，小さい方のλを増加，両方のλを増加／減少)が存在する．

5，2　あるサーバにボトルネックの可能性があると診断された場合の知識の適用順序

あるサーバにボトルネックの可能性があると診断された場合に，Table 4の第2項の知識2でその直接的な要因と下流を調べる．(スプリット型・マッチ型サーバについては，これらが二つの下流の窓口群を持つため，同じ知識を二度適用する．)この知識2の適用後，そのサーバとその直上流のサーバとの接続関係によりTable 4内の第3～14項の知識3～14を適用する．上述のネットワーク変形のための知識(Table 3)は，上流への遡行の過程で直上流のサーバが同期型サーバとなった時点で適用する．

6　BDES-Sを用いたパラメータチューニングの実行例

BDES-Sは，Table 3のネットワーク変形のための知識と，Table 4の定性改善式を用いて定性推論により改善プランを列挙するようにProlog言語を用いてインプリメントされた．ユーザはこのBDES-Sを用いて，対象システムのボトルネック診断を対話的に行うことができる．

Fig.1のSQN1のボトルネック窓口s6(マージ型サーバ)に対して，BDES-Sを用いて対話的に診断を実行した例をFig.2に示す．(1)では，初期診断のための知識を適用し，ボトルネックサーバを列挙し，(2)で，診断を行いたいサーバの初期診断結果を表示，(3)で，このサーバのボトルネック改善の方法を表示している．(4a～4d)以下では，(3)で提示された方法に従って順次診断を行っている．

(5)では，変形の知識を適用し，マージ型サーバ(s6)をノーマル型サーバに変形している．(6)で，下流の同期型サーバへの影響を調査し，(7)で知識3を適用して上流サーバへ診断を継続している．(8)では，ｔの減少による，スプリット型サーバ(s5)のもう一方の下流サーバ群に対する影響を調査し，(9)で下流の同期型サーバに悪影響があると診断し，同時に，s35からs38へのｔを減少するように指示している．(10)では，知識9を適用し，(11)でｒの変更による下流サーバ群に対する影響を調査，(12)で上流のサーバが2つある場合，ユーザがどちらかのサーバを選択し，診断を継続し，(13)では，(12)で選択されなかったサーバについてバックトラック診断を行っている．

(14)では，診断<1>,<2>の結果を組み合わせて，両方のλを減少するためのプランを表示している．(15)の選択方法は，(1 2 3 4)から1つ，(5 6 7 8 9)から1つ選択することを示す．同様に，(16)で診断<1>の結果を表示し，大きい方のλを減少するためのプランを表示している．

Table 5では，Fig.1のSQN1のボトルネック窓口s6に対する定性改善プランのうち[③の3]を選択し，ｇaを0.069から0.066に変更し，同時にｒ16,21を0.40から0.38に変更した時の改善前と改善後の性能パラメータを使って測定を行った際のρとｑの比較を行った．ρ16とρ27の値が減少し，ボトルネックが解消された．s6,s8の各々の２つのｑは，必ずしも小さくなっているわけではないが，その差はかなり軽減された．測定平均値のみを用いた改善が有効であることがわかる．

7　おわりに

本稿では，定性推論を用いた同期型待ち行列ネットワークのボトルネック改善方法を考察し，改善のための知識や定性改善式，BDES-Sによる定性推論と実行例を示した．多くの定性的な改善プランから１つのプランを選択する基準について今後考察したい．また，定性的な改善プランの列挙の後，通常の待ち行列ネットワークでは定量的な改善が半自動的に行われたが，同期型待ち行列ネットワークに対しても同様に行いたい．

8 参考文献

[APT86]Apte, C. et al.: Using qualitative reasoning to nderstand financial arithmetic, Proc. AAAI'86, (1986).

[BOB85]Bobrow, D.G., et al. ed.: Qualitative reasoning about physical systems, MIT Press, (1985).

[DEK85]De Kleer, J.: How circuits work, in [BOB85], (1985).

[FLO89]Gerard Florin and Stephane Natkin: Necessary and Sufficient Ergodi city Condition for Open Synchronized Queueing Networks, Proc. IEEE'89.

[GEL85]Gelenbe,E. et al.:Analysis and Synthesis of Computer Systems, Academic Press, (1989).

[ITO89]Itoh,K. et al.:Knowledge-Based Parameter Tuning for Queueing Network Type System — A New Application of Qualitative Reasoning, CAPE'89, (October, 1989).

[ITO90]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).

[KLE75]Kleinrock,L.: Queueing Systems, John Wiley & Sons, Inc., (1975).

[MIZ89]溝口他編： 定性推論，共立出版， (1989).

[NIS88]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, No.11, (1988).

[NIS89]西田：定性推論の基礎，人工知能学会誌，Vol.4, NO.5, 522-527 (1988).

[PET84]J.L.Peterson, 市川，小林(訳):ペトリネット入門，共立出版(1984).

[RAJ84]Rajagopalan, R.: Qualitative modeling in the turbojet engine domain, Proc. AAAI'84, (1984).

[RAZ85]R.R.Razouk and C.V. Phelps:Performance Analysis Using Timed Petri Nets, in Protocol Specification, Testing and Verification IV, (1985).

[SAW89a]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，IPSJ WGAI，(January, 1989).

[SAW89b]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，30.8,990-1002(August,1989).

[SHI90]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，IPSJ WGAI，(January, 1990).

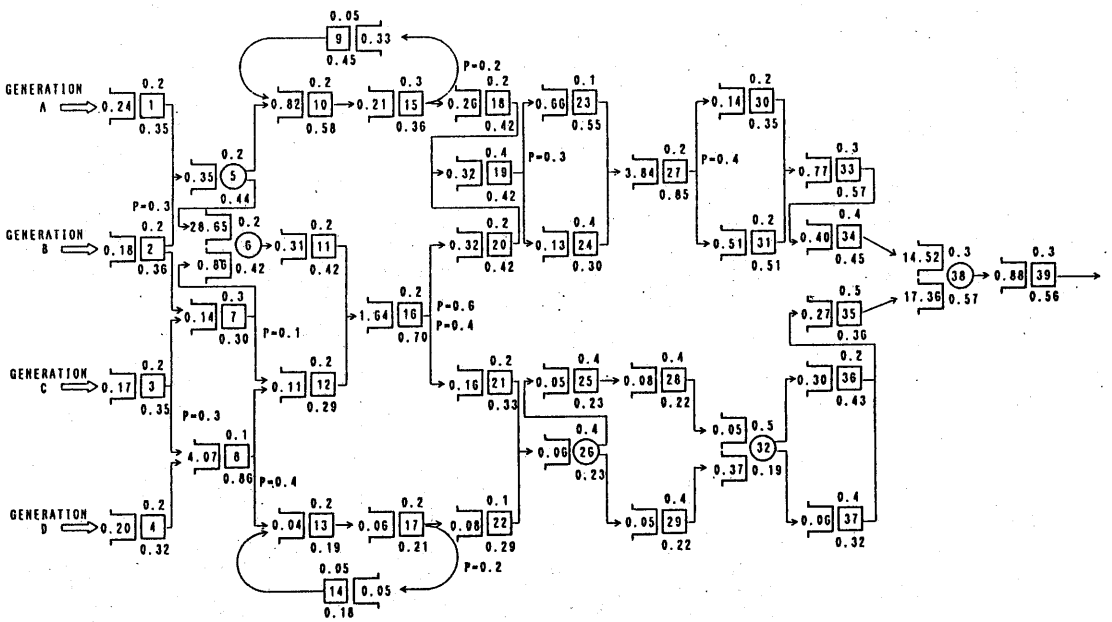
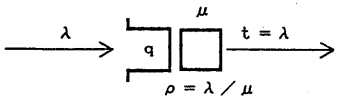


Fig.1 An example of Synchronized queueing network (SQN1)

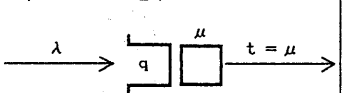
Table 1 Four types of servers and their performance parameters

1)normal server

1-a)case of $\lambda < \mu$

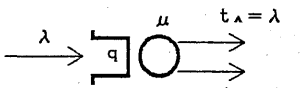


1-b)case of $\lambda \geqq \mu$

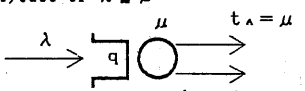


2)split server

2-a)case of $\lambda < \mu$

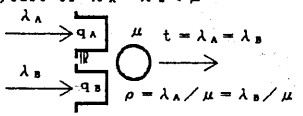


2-b)case of $\lambda \geqq \mu$



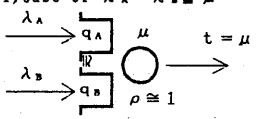
3)merge server

3-a)case of $\lambda_A = \lambda_B < \mu$

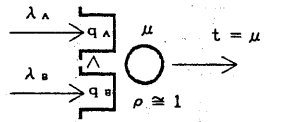


3-b)case of $\lambda_A \geqq \mu$ and $\lambda_B \geqq \mu$

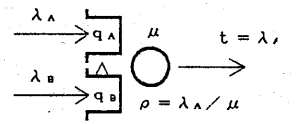
3-b-1)case of $\lambda_A = \lambda_B \geqq \mu$



3-b-2)case of $\mu \leqq \lambda_A < \lambda_B$

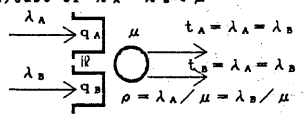


3-c)case of $\lambda_A < \lambda_B$ and $\lambda_A < \mu$



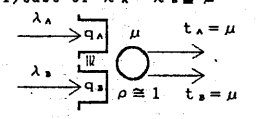
4)match server

4-a)case of $\lambda_A = \lambda_B < \mu$

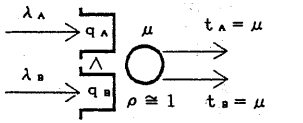


4-b)case of $\lambda_A \geqq \mu$ and $\lambda_B \geqq \mu$

4-b-1)case of $\lambda_A = \lambda_B \geqq \mu$



4-b-2)case of $\mu \leqq \lambda_A < \lambda_B$



4-c)case of $\lambda_A < \lambda_B$ and $\lambda_A < \mu$

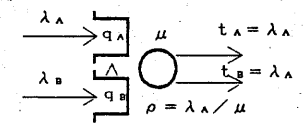


Table 2 QLBE and QL-BIE for each type of servers

Table 2-1 normal server

	dρ	dt	dλ	dμ
[ρ] = −	±	±	±	
	∓	0		±
[ρ] = +	−	0	−	
	−	+		+

	dρ	dq	dρ	dq
[ρ] = +		−	−	
	−			−

Table 2-2 split server

	dρ	dt_A	dt_B	dλ	dμ
[ρ] = −	±	±	±	±	
	∓	0	0		±
[ρ] = +	−	0	0	−	
	−	+	+		+

	dρ	dq	dρ	dq
[ρ] = +		−	−	
	−			−

Table 2-3 merge server

	dρ	dt	dq_A	dq_B	$d\lambda_A$	$d\lambda_B$	dμ
[ρ] = −	±	±	0	0	±	±	
[q_A] = −	0	0	+	0	+	0	
[q_B] = −	−	−	0	+	−	0	
	∓	0	0	0			±
[ρ] = +	−	0	−	−	−	−	
[q_A] = +	−	+	−	−			+
[q_B] = + ※	−	+	−	−		−	+
[ρ] = −	0	0	0	−		−	
[q_A] = −	−	−	0	−	−	−	
	+	+	0	−	+		
[q_B] = +	+	+	0	−	+	+	

Table 2-4 match server ***omitted***

※ denotes the expression to be used in case of $q_A < q_B$

Table 3 Knowledge for network transformation

Table 3-1 Knowledge for downstreams (Knowledge m1)

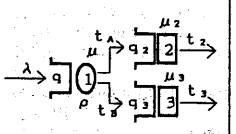
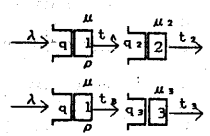
before transformation	after transformation	Knowledge m1
		Apply to match or split servers. Duplicate match or aplit server. Consider it as normal server.

Table 3-2 Knowkledge for upstreams (Knowledge m2, m3)

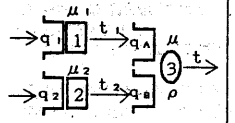
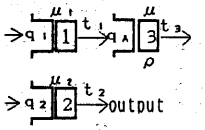
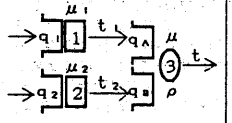
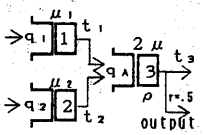
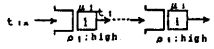
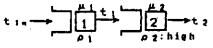
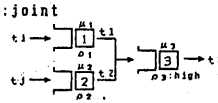
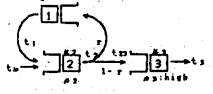
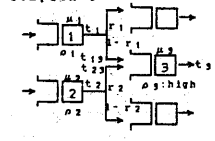
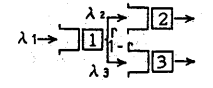
before transformation	after transformation	Knowledge m2
 q A > q B		Apply to merge or match server in case that either λ_A or λ_B is source of bottleneck.
QL-BIE $dq_A = - \leftarrow dt_1 = -$	QL-BIE $dq_A = - \leftarrow dt_1 = -$	
before transformation	**after transformation**	**Knowledge m3**
 q A = q B		Apply to merge or match server in case that both λ_A and λ_B are sources of bottlenecks.
QL-BIE $dq_A = - \leftarrow dt_1 = -$, $dt_2 = -$	QL-BIE $dq_A = - \leftarrow dt_1 = -$ and $dq_A = - \leftarrow dt_2 = -$ <need for matching>	

Table 4 Substructure-based knowledge and QL-BIE adopted by BDES-S

substructure	knowledge	QL-BIE
1 The servers whose "ρ"s $\geq$ 0.7 or whose "q"s $>$1.0 are may be bottleneck.		
2 	On improving a bottleneck server by increasing it's "μ_1", decrease "ρ_1"s of it's downstream servers whose "ρ_1"s $\geq$0.7 or whose "q_1"s $>$ 1.0.	$d\rho_1$=- $\leftarrow d\mu_1$=+ $d\rho_1$=- $\leftarrow d\rho_1$=-
3:tandem 	Decrease t_{1n} for decreasing ρ_2.	$d\rho_2$=- $\leftarrow dt_{1n}$=-
4:joint 	(1) In case of $t_1 > t_2$, decrease t_1 for decreasing ρ_3.	$d\rho_3$=- $\leftarrow dt_1$=-
	(2) In case of $t_1 \cong t_2$, decrease t_1 or t_2 for decreasing ρ_3.	$d\rho_3$=- $\leftarrow dt_1$=- $d\rho_3$=- $\leftarrow dt_2$=-
omitted		
7:loop 	Decrease t_{1n} for decreasing $t_{2,3}$ (output of a loop).	$d\rho_3$=- $\leftarrow dt_{in}$=-
omitted		
9:complex-5 	(1)Case of $t_2 * r_2 \gg t_1 * (1-r_1)$, decrease $t_{2,3}$ for decreasing ρ_3.	$d\rho_3$=+ $\leftarrow dt_{2,3}$=-
	(2)Case of $t_2 * r_2 \ll t_1 * (1-r_1)$, decrease $t_{1,3}$ for decreasing ρ_3.	$d\rho_3$=+ $\leftarrow dt_{1,3}$=-
	(3)Case of $t_2 * r_2 \cong t_1 * (1-r_1)$, decrease $t_{1,3}$ for decreasing ρ_3.	$d\rho_3$=+ $\leftarrow dt_{1,3}$=- $d\rho_3$=+ $\leftarrow dt_{2,3}$=-
omitted		
12:branch 	a)When r is modifiable, increase r for increasing λ_2.	$d\lambda_2$=+ $\leftarrow dr$=+
	b)When r isn't modifiable, increase λ_1 for increasing λ_2. (max: $\lambda_1 = \mu_1 * BL$)	$d\lambda_2$=+ $\leftarrow \lambda_1$=+
omitted		

同期型待ち行列ﾈｯﾄﾜｰｸ*sqn1*のﾎﾞﾄﾙﾈｯｸ診断行います．

ρが最大の serverの（server名・ρ・q）　(1)
　(s8　0.86　4.07)
ρが0.9以上の serverはありません．
ρが0.7以上の serverの（server名・ρ・q）
　(s8　0.86 4.07)(s16　0.7 1.64)(s27　0.85 3.84)
ρが0.7以下でqが1以上の serverはありません．
ふたつのλが異なるserverの（server名・ρ・maxq）
　(s6　0.42 28.65)(s38　0.57 17.36)

診断する server名を入力して下さい．　s6.
[serverの初期診断結果]　(2)
　診断する窓口 s6 は merge-type-server です．
　直下流には1個 server があり，さらに下流には
　21個server があります．
　この server は ρ(q)が過大ではありません．
　また，２つのλが異なっていると初期診断されて
　います．
　２つのλを比較すると，server s5からのλの方が，
　server s7 からのλより大きく，server s5 のλ
　の形成するqが過大となっています．
　この server の下流には ρ(q)が過大であると初
　期診断された serverが，2個 [s16,s27] あります．
　この server の下流には２つのλが異なっている
　と初期診断された serverが，1個 [s38] あります．
[この server のﾎﾞﾄﾙﾈｯｸ改善の方法]　(3)
　①両方のλを減少　②両方のλを増加
　③大きい方のλを減少　④小さい方のλを増加

1> server s6の大きい方のλを減少すると，下流への
　tの変化はありません．　(4a)
　s7からs6への接続関係を抹消し，s7の出力先を外
　部とします．　(5)
　match-server s32の２つのλは，ともに変化があ
　りません．よって，　(6)
　server s32 への影響はありません．
　merge-server s38の２つのλは，ともに変化があ
　りません．
　よって，server s38 への影響はありません．
[上流との接続]
　server s6 の直上流には 1 個の server [s5] が
　あります．
　直上流の server s5 の直下流には，他に 1 個の
　server [s10] があります．
　直上流の server s5は，split-type-server なの
　で，server s6 のλを減するためには，server
　s5 のtを減少する必要があります．　(7)
　server s5 のtを減少すると，server s5 から
　server s10へのtも減少するので，server s10 の
　下流を調査する必要があります．　(8)
[s10の下流の状況]
　server s10 のtが減少すると，merge-server
　s38の２つのλのうち s34からのλは減少します．
　小さい方のλが減少するので，同時に s35から
　s38へのλも減少が必要です．　(9)
[上流へ週行]
　診断する窓口 s5 は split-type-server です．
　直下流には2個 server があり，さらに下流には
　25個 server があります．
[上流との接続]
　server s5 の直上流には 2 個の server [s1,s2]
　があります．
　直上流の server s1 の直下流には，他にserverは
　ありません．
　直上流の server s2 の直下流には，他に 1 個の
　server [s7] があります．
　server s1 および server s2 とserver s5 の接
　続は合流型です．　(10)
　したがって，server s2 からserver s7 へのrを
　増加してserver s2 から server s5 へのrを減少
　するためには，server s7 の下流を調査する必要
　があります．　(11)
[s7 の下流の状況]
　直下流には1個 server があり，さらに下流には
　21個 server があります．
　server s2 からserver s7 へのrを増加して
　server s2 からserver s5 へのrを減少することが
　できます．
　または，server s5 のλを減少するためには，
　server s2 のtを減少する必要があります．
　または，server s5 のλを減少するためには，
　server s1 のtを減少する必要があります．
server s1 とserver s2 のどちらを選択しますか？
→ s1.　(12)
[上流へ週行]
　診断する窓口 s1 は nomal-type-server です．
　直下流には1個 server があり，さらに下流には
　27個 server があります．
[上流との接続]
　server s1 の直上流には　server はありません．
　外部入力 [ga] があります．したがって，server
　s1 のλを減少するためには，外部入力 [ga] を減
　する必要があります．
<ﾊﾞｯﾄﾗｯｸ診断を行います>　(13)
[上流へ週行]
　診断する窓口 s2 は nomal-type-server です．
……<省　略>……
2> server s6の小さい方のλを減少すると，下流への
　tの変化は減少です．　(4b)
　s5からs6への接続関係を抹消し，s5の出力先を外
　部とします．
　match-server s32の２つのλは，ともに減少しま
　す．よって，　server s32への影響はありません．
　merge-server s38の２つのλは，ともに減少しま
　す．よって，　server s38への影響はありません．
[上流との接続]
　server s6 の直上流には 1 個の server [s7] が
　あります．
……<省　略>……
3> server s6の小さい方のλを増加すると，下流への
　tの変化は増加です．　(4c)
　よって server [s16,s27] のρを減少する必要が
　あります．
　s5からs6への接続関係を抹消し，s5の出力先を外
　部とします．
……<省　略>……
4> server s6の大きい方のλを増加すると，下流への
　tの変化はありません．　(4d)
　s7からs6への接続関係を抹消し，s7の出力先を外
　部とします．
……<省　略>……
<ﾊﾞｯﾄﾗｯｸ診断を行います>
[上流へ週行]
　診断する窓口 s2 は nomal-type-server です．直
　下流には2個 server があ
　り，さらに下流には28個 server があります．
[上流との接続]
　server s2 の直上流には　server はありません．
　外部入力 [gb] があります．したがって，server
　s2 のλを増加するためには，外部入力 [gb] を増
　する必要があります．

/**BDES-Sによるﾎﾞﾄﾙﾈｯｸ診断結果を出力します**/
①両方のλを減少　(14)
　1　rs2s5　減
　2　g[ga]　減
　　g[gd]　減
　3　g[ga]　減
　　rs16s21　減
　4　g[gb]　減
……<省　略>……
　9　g[gc]　減
　選択方法　-1 -2 -3 -4　(15)
　　+5 -6 -7 -8 -9

②両方のλを増加
……<省　略>……
③大きい方のλを減少　(16)
　1　rs2s5　減
　2　g[ga]　減
　　g[gd]　減
　3　g[ga]　減
　　rs16s21　減
　4　g[gb]　減
　選択方法　-1 -2 -3 -4

④小さい方のλを増加　(17)
　1　rs7s6　増
　　ρ[s16,s27]減
……<省　略>……
/**BDES-Sによるﾎﾞﾄﾙﾈｯｸ診断を終了します**/

Fig. 2 Parameter tuning on bottleneck server s6 by BDES-S

Table 5 Comparison of ρs and qs between two measurements

parameter	measurement-1	improved parameter	measurement-2
q 6A q 6B	28.65 0.86	g a　0.069→0.066	0.45 8.04
q 38A q 38B	14.52 17.36	$r_{16,21}$ 0.40→0.38 $r_{16,28}$ 0.60→0.62	9.81 3.74
ρ_{16}	0.70	μ_{16} 0.20→0.265	0.67
ρ_{27}	0.85	μ_{27} 0.20→0.220	0.65