数理解析研究所講究録
833巻 1993年 131-141

中間経由節点をもつKサーバー問題

京都大学工学部　軽野義行 (Yoshiyuki KARUNO)
京都大学工学部　茨木俊秀 (Toshihide IBARAKI)

1 緒論

Kサーバー問題は、n節点完全グラフにおいて生じる一連の要求の処理に対するK台のサーバーの移動計画であり、そのオンラインアルゴリズムが Manasseら[5]などによって研究されている。処理すべき各要求はグラフの1つの節点上に発生し、サーバーの1つがその節点へ移動して処理を実行する。グラフの各枝には距離が定義され、サーバーがある2節点間を移動するとき、その距離に等しいコストが掛かる。なお、要求はその発生順に処理されなければならない。

Kサーバー問題に対するオンラインアルゴリズムでは、各要求に対してどのサーバーを移動させるかという決定を未来の要求の情報なしに行い、K台のサーバーの全移動距離を小さく抑えることが目的になる。

本研究では、上記のようなKサーバー問題のある一般化、すなわち、各要求が、カバーすべき目標節点のほかに、通過すべき中間経由節点を一つ指示できるとした問題を考察する。アルゴリズムがこのような要求を処理するためには、あるサーバーをまず中間経由節点に移動させ、さらにそのサーバーを目標節点へ移動させる必要がある。このタイプの問題は、多数の生産セルを数台の無人搬送車で結合する分散型自動生産システムのマテリアル・ハンドリング等において生じる。また、この問題は Fiatらの文献[3]に見られるKタクシー問題に似ている。しかし、Kタクシー問題においては、目標節点の情報は中間経由節点にサーバー(タクシー)が到着した後にのみ得られる。我々の問題では両節点の情報があらかじめ得られるとしているので、

“無線連絡装置をもつＫタクシー問題”ともいえる。

本研究では、中間経由節点を考慮したＫサーバー問題に対してオンラインアルゴリズムを構築し、“Competitive 解析”の見地からその評価を行う。“Competitive 解析”はSleatorら[7]によって提案されたオンラインアルゴリズムに対する一種の最悪性能比の解析である。すなわち、あるオンラインアルゴリズムＡ、および全要求の情報をあらかじめ保持できる最適オフラインアルゴリズムＯＰＴに対して、要求列σを処理するために必要なＫ台のサーバーの全移動距離を、$cost_A(\sigma)$、$cost_{OPT}(\sigma)$としたとき、任意のσに対して

$$cost_A(\sigma) \leqq c \cdot cost_{OPT}(\sigma) + I \tag{1.1}$$

を満たすような定数 I が存在するならば、オンラインアルゴリズムＡは c-competitive であるという。ただし、定数 I はＫ台のサーバーの初期配置にのみ依存するものであって、要求列σに依存してはならない。アルゴリズムＡに対する c の最小値をＡの competitive ratio という。また、他の全てのオンラインアルゴリズムの competitive ratio がＡのそれ以上であるとき、Ａは最適オンラインアルゴリズムであるという。Manasseら[6]は、中間経由節点のないＫサーバー問題について、Ｋ＝ｎ－１およびＫ＝２に対する competitive なオンラインアルゴリズムを示しているが、ここでは中間経由節点をもつＫサーバー問題について competitive なオンラインアルゴリズムを構築する。さらに、一般のＫに対しても幾つかの知見を述べる。

2　問題の記述

ｎ節点の完全グラフＧを考え、各枝 (u ,v) には 距離 d(u ,v) を定義する。これらの距離 d は三角不等式を満足し、任意の節点 v に対して d(v ,v)＝0、さらに任意の２節点 u ,v に対して d(u ,v)＝d(v ,u) であるとする。また、ＫをグラフＧ内に存在するサーバーの台数、$C=\{C_1, C_2, \ldots, C_M\}$ をそれらの取り得る配置の集合とする。

次に、処理すべき要求の列を $\sigma=\sigma(1), \sigma(2), \ldots, \sigma(N)$ とする。要求はその発生順に処理

される。各要求σ(i)は2節点のペア {p(i) ,q(i)} によって記述される。ここで、p(i) は中間経由節点、q(i)は目標節点を表す。このような要求を処理するためには、あるサーバーが枝 (p(i) ,q(i)) 上を移動しなければならない。勿論、ある要求σ(i)が 中間経由節点を持たない、すなわち、p(i)=q(i)であっても構わない。もし各全ての要求σ(i)に対して p(i)=q(i) ならば、この問題は Manasseらの K サーバー問題[6] と同じになる。アルゴリズムの性能はK台のサーバーの全移動距離（必要な p(i) から q(i) への移動距離も含める)で評価される。

この問題に対するオンラインアルゴリズムAは、既に発生している要求 σ(1) ,..., σ(i) の情報と現在のサーバーの配置状態のみに基づいて、要求σ(i)の処理にどのサーバーを割当てるかを決定する。一方、オフラインアルゴリズムでは全要求 σ(1), σ(2),..., σ(N) の情報をあらかじめ知ることができ、それに基づいたサーバーの移動が可能となる。サーバーの全移動距離を最小にする最適オフラインアルゴリズムをＯＰＴと記す。以下ではK台のサーバーの全移動距離をそのアルゴリズムのコストと呼び、$cost_A(\sigma)$、$cost_{OPT}(\sigma)$等と記す。

3　(n－1)サーバーの場合

本節では、中間経由節点をもつ(n－1)サーバー問題に対して２つのオンラインアルゴリズムＢＡＲＴ１およびＢＡＲＴ２を与え、これらの competitive ratio について検討する。これらのアルゴリズムは以下に示す特殊な要求列に対応する２つのアルゴリズムＢＡＬおよびＲＴを基礎としている。

Manasseら[6]において提案されたように、バランスアルゴリズムＢＡＬは中間経由節点のない問題に対して、次のようにサーバーを移動させる：(n－1)台のサーバーは、互いに重ならないよう常に異なる(n－1)個の節点上に配置される。ＢＡＬは、ある時点において節点 j に位置するサーバーに対して、要求列の処理を開始してからそれまでの累積コスト D_j を保持する。いま、ある要求 σ(i)=(k ,k) が発生したとする。このとき、もし節点 k 上にサーバー

が位置していれば、そのサーバーに処理をさせる。したがって、どのサーバーのコストも増加しない。一方、節点 k が空であれば、$D_j+d(j,k)$ を最小にする節点 j 上のサーバーを節点 k へ移動させる。つまり、ＢＡＬは(ｎ－１)台のサーバーを均等に利用しようとする。アルゴリズムＢＡＬに対しては次が証明されている：

定理１ [6]　*n節点完全グラフにおける中間経由節点のない(n－1)サーバー問題に対して、オンラインアルゴリズムＢＡＬは (n－1)-competitive である。*

ここで、各全ての要求 $\sigma(i)=\{p(i),q(i)\}$ が $p(i)\neq q(i)$ である場合に対して、ＢＡＬを変形した次のような往復アルゴリズムＲＴを考える：(ｎ－１)台のサーバーを互いに重ならないよう常に(ｎ－１)個の節点上に配置する。要求 $\sigma(i)=\{p(i),q(i)\}$ が与えられたとき、もし $q(i)$ にあるサーバーが位置しているならば、ＲＴはそのサーバーに中間経由節点 $p(i)$ との間を往復させる。もし $q(i)$ が空であるならば、$p(i)$上のサーバーを $q(i)$ へ移動させる。$p(i)\neq q(i)$である要求 $\sigma(i)=\{p(i),q(i)\}$ から成る任意の要求列 σ に対して、規定区間の移動距離の総和を

$$S_0=\sum_{i=1}^{N} d(p(i),q(i)) \tag{3.1}$$

としたとき、$cost_{RT}(\sigma)\leqq 2\cdot S_0$、および $cost_{OPT}(\sigma)\geqq S_0$ は明らかである。よって、次が成り立つ：

補題２　*各全ての要求が中間経由節点をもつ(n－1)サーバー問題に対して、オンラインアルゴリズムＲＴは 2-competitive である。*

ただし、アルゴリズムＲＴの 2-competitive 性は、規定区間の距離 $d(p(i),q(i))$ がコストに含まれているために達成できることに注意する。

次に、$p(i)=q(i)$ および $p(i)\neq q(i)$ の ２つの要求のタイプが混在する一般の場合に対して、アルゴリズムＢＡＲＴ１とＢＡＲＴ２を以下に与える。ＢＡＲＴ２は、上のＢＡＬとＲＴまとめたものであり、ＢＡＲＴ１は解析の容易さのため、ＲＴの一部を少し変形したアルゴリ

ズムとなっている。

ALGORITHM BART1

1（初期設定）．（n－1）台のサーバーを互いに重ならないように配置する。
2（σ(i)の処理）．発生した要求σ(i)（i=1,2,...）を次のように処理する。
ルール1　p(i)=q(i)ならば、BALを適用する。
ルール2　p(i)≠q(i)ならば、節点 q(i) 上のサーバーに規定区間を往復させる。もし節点 q(i)が空ならば、節点 p(i) 上のサーバーに規定区間を往復させる。□

ただし、BALにおいて必要な各サーバーの累積コスト D_j はルール1適用時にのみ更新する。

定理3　*中間経由節点をもつ(n－1)サーバー問題に対して、オンラインアルゴリズム BART1は(n＋1)-competitive である。*

（証明）　要求列σを各要求に適用されるルールが1であるか2であるかにしたがって、2つの部分列σ_1、σ_2に分割する。σ_1(σ_2)に属する要求の相対的な順序はσにおける順序と同じである。

ルール2を適用してもサーバーの配置は変化しないので、BART1のコストは部分列σ_1に対するコストとσ_2に対するコストの和となる。さらに、定理1および補題2を適用すると、

$$\begin{aligned} cost_{BART1}(\sigma) &= cost_{BART1}(\sigma_1)+cost_{BART1}(\sigma_2) \\ &= cost_{BAL}(\sigma_1)+2\cdot S_0 \\ &\leqq (n-1)\cdot cost_{OPT}(\sigma)+I+2\cdot cost_{OPT}(\sigma) \\ &= (n+1)\cdot cost_{OPT}(\sigma)+I \end{aligned} \tag{3.2}$$

となる。ただし、S_0は規定区間の総移動距離である(式(3.1)参照)。▮

アルゴリズムBART1のルール2では、q(i)が空の場合もサーバーに規定区間を往復させるため、アルゴリズムの効率が悪くなる。その点を改良したものが次のBART2である。

ALGORITHM BART2

1（初期設定）．（n－1）台のサーバーを互いに重ならないように配置する。また、空のスタックを1つ用意する。
2（σ(i)の処理）．発生した要求σ(i)（i=1,2,...）を次のように処理する。
ルール1　p(i)=q(i)のとき、スタックが空ならばBALを適用する。スタックが空でなければ、1つ前の配置に戻すことによってその要求を処理する。
ルール2　p(i)≠q(i) ならば、RTを適用する。そのとき、もしサーバーの配置が変化するならば、元の配置をスタックに記憶する。□

ただし、各サーバーの累積コスト D_j は、ここでも ＢＡＬ を適用した際にのみ更新される。

ＢＡＲＴ２のルール２では q(i) が空の場合、p(i)のサーバーを q(i) へ移動させる。そのため、ルール１においてＢＡＬを適用した直後のサーバーの配置が、次にそれを適用する際に変化してしまっているかもしれないので、アルゴリズムのコストの評価が困難になる。そこで、スタックを用いて、次にＢＡＬが適用されるまでに前回のＢＡＬ適用直後のサーバーの配置を復元できるようにしている。このことより、ＢＡＲＴ２のルール１においてＢＡＬが適用されるのは、σ_Iの部分列σ_I'である（σ_Iの定義は、定理３の証明参照）。$cost_{BAL}(\sigma_I') \leqq cost_{BAL}(\sigma_I)$、したがって $cost_{BART2}(\sigma) \leqq cost_{BART1}(\sigma)$ であるから、次の定理が成立する：

定理４　*中間経由節点をもつ$(n-1)$サーバー問題に対して、オンラインアルゴリズム BART2も $(n+1)$-competitive である。*

もしアルゴリズムＢＡＲＴ２において、$(n-1)\cdot cost_{OPT}(\sigma)+I \geqq cost_{BAL}(\sigma_I')+(n-1)\cdot S_0$、または、$cost_{OPT}(\sigma)+I \geqq cost_{OPT}(\sigma_I')+S_0$ が成立するならば、ＢＡＲＴ２は（ｎ－１）-competitive であるといえる。しかし、これらは未解決である。ただし、σ_I'に属する要求がｎ節点全てから発生していないときは $cost_{OPT}(\sigma_I')=0$, $cost_{OPT}(\sigma) \geqq S_0$ であるから、上式が成立する。

4　２サーバーの場合

４．１　残余関数　オンラインアルゴリズムの c-competitive 性を決定するために、Manasseら[6]によって、以下のような“残余関数 Rc ”が導入された：

$$Rc(i, C_j) = c\cdot cost_{OPT}(i, C_j) - cost_A(i). \tag{4.1}$$

ここで、$cost_{OPT}(i, C_j)$ は要求列σの最初の i 個を処理し、配置 C_j で終了するある最適オフラインアルゴリズムのコスト、$cost_A(i)$ はオンラインアルゴリズムＡがσの最初の i 個の要求を処理したときに生じたコストを表している。ある初期配置が与えられると、動的計画法

によって $cost_{OPT}(i, C_j)$、したがって $Rc(i, C_j)$ が容易に計算できる。最適オフラインアルゴリズムのコストは $\min_j\{cost_{OPT}(N, C_j)\}$ である。よって、式(1.1)より、任意の N と C_j に対して、

$$Rc(N, C_j) \geqq -I \tag{4.2}$$

となるような定数 I が存在すれば、オンラインアルゴリズムAは c-competitive である。考慮すべき配置の集合の大きさ M が小さいとき、この方法は有効である。

4．2　アルゴリズムの記述　中間経由節点をもつ２サーバー問題に対して、前述の残余関数を用いて 2-competitive なオンラインアルゴリズムＩＲＥＳを構築することが出来る。これは、Manasseら[6]のアルゴリズムＲＥＳに若干の修正を加えたものとなっている。

アルゴリズムＩＲＥＳは、２台のサーバーを常に異なる２つの節点に配置する。いま、要求 $\sigma(i-1)=\{p(i-1), q(i-1)\}$を処理した直後とする。 $q(i-1)=a$ とするとＩＲＥＳの１台のサーバーは節点 a 上に位置しており、もう１台は節点 b ($\neq a$) にある。このとき、最適オフラインアルゴリズムにおいても１台のサーバーが節点 a に位置しているが、もう1台のサーバーはどこに位置するか分からない。したがって、ＩＲＥＳが考慮しなければならい残余関数の数は、もう１台のサーバーの可能な位置 u に対応するｎ個である。簡単のため、サーバーが節点 a 及び u に位置する最適オフラインアルゴリズムとＩＲＥＳを比較した残余関数を $R(a, u)$ $(= R_2(i, \{a, u\}))$と記す。

アルゴリズムＩＲＥＳにおいてサーバーの初期位置が a と b であるとすると、残余関数の初期値は各 u に対して $R(a, u)= d(a, b)+2\cdot d(b, u)$ とする。一般に、要求$\sigma(i)=\{s', a'\}$を処理するために、ＩＲＥＳはまず（s'が空節点であれば）s'へあるサーバーを移動させる必要がある。ここで、次のようなルールを採用する：もし

$$\min_k \{R(a, k)+2\cdot d(k, s')\} \geqq 2\cdot d(a, s')+d(a, b) \tag{4.3}$$

ならば、ＩＲＥＳは節点 a 上のサーバーを s'へ移動させ、そうでないならば 節点 b 上のサ

ーバーを移動させる。その後、そのサーバーを a'へ移動させる。ただし、s'が空であっても、既に節点 a'上にもう1台のサーバーが位置しているときは、それに規定区間を往復させる。

定理5 *中間経由節点をもつ2サーバー問題に対して、ＩＲＥＳは 2-competitive である。*

(略証) アルゴリズムＩＲＥＳの保持する残余関数の値が、任意の u と v に対して常に以下の下界値をもつことを、場合分けと帰納法によって示すことが出来る：

$$R(a,u)+R(a,v) \geqq Q(a,b,u,v), \tag{4.4}$$

ここで、

$$Q(v,w,x,y) = 2\cdot\max\{d(v,w)+d(x,y), d(v,x)+d(w,y), d(v,y)+d(w,x)\}. \tag{4.5}$$

式(4.4)においては、u=v も可である。もし式(4.4)が成立すれば、各残余関数の値は常に非負である。よって、式(4.2)より、この下界値はアルゴリズムの 2-competitive 性を保証する。

詳細は割愛するが、いま、要求σ(i-1)を処理した直後の残余関数が式(4.4)を満足していると仮定し、例えば、次の要求σ(i)={s',a'}において s'≠a'かつ s',a'≠a,b、さらに、移動決定ルール式(4.3)が成立しているならば、ＩＲＥＳは節点 a のサーバーを移動させる。もう1台のサーバーは、節点 b(⇒ b')に留まる。ここで $\min_k\{R(a,k)+2\cdot d(k,s')\}$を最小にするインデックスを k とすると、残余関数は次のように更新される：

$$R'(a',a) = R(a,k)+2\cdot\{d(k,s')+d(s',a')\}-\{d(a,s')+d(s',a')\},$$

$$R'(a',u) = R(a,u)+2\cdot\{d(a,s')+d(s',a')\}-\{d(a,s')+d(s',a')\}, \ (u\neq a).$$

これらの更新された残余関数の値も式(4.4)を満たすことが帰納的に示される。∎

上記の定理は Manasseら[6] の結果を含んでおり、次を系としておく：

系 6 [6] *中間経由節点のない2サーバー問題に対して、オンラインアルゴリズムＩＲＥＳは 2-competitive である。*

系 7 *全ての要求が中間経由節点を有する(つまり、各σ(i)={p(i),q(i)}において p(i)≠q(i)) 2サーバー問題に対して、アルゴリズムＩＲＥＳは2-competitiveである。*

5　一般のKに対する結果

任意のK≧1に対して、Manasseら[6]は「中間経由節点のないKサーバー問題に対して、ある K-competitive なオンラインアルゴリズムが存在する」と予想している。この予想は、3節および4節でみたように、K＝n－1、K＝2に対しては証明されている。しかし、一般のKに対しては未解決である。また、彼らは任意のオンラインアルゴリズムに対して、その competitive ratio の下界を示している：

定理8 [6] *中間経由節点のないKサーバー問題に対する任意のオンラインアルゴリズムの competitive ratio は少なくともKである。*

この下界は我々の問題に対しても成立する。さらに、次の定理を得る：

定理9　*もし中間経由節点のないKサーバー問題に対して K-competitive なアルゴリズムが存在するならば、中間経由節点をもつKサーバー問題に対する最適オンラインアルゴリズムの competitive ratio は、KとK＋2の間である。*

(証明)　下界は定理8によって与えられているので、ここでは上界を示す。要求列σに対し、σ_qを中間経由節点を省略した要求の列とする。すなわち、$\sigma(i)=\{p(i),q(i)\}$のとき、$\sigma_q(i)=\{q(i),q(i)\}$である。中間経由節点のないKサーバー問題に対する K-competitive なアルゴリズムをBとする。中間経由節点をもつKサーバー問題に対するオンラインアルゴリズムAは、要求列σ_qに対してその K-competitive なアルゴリズムBを模倣し、目標節点 q(i) に移動させるサーバーを決定する。アルゴリズムAは、そのサーバーをいったん中間経由節点 p(i) に移動させて、さらにそれを目標節点 q(i) に移動させる。節点間の距離は三角不等式を満足しているので、

$$
\begin{aligned}
\mathrm{cost}_A(\sigma) &\leqq \mathrm{cost}_B(\sigma_q)+2\cdot S_0 \\
&\leqq K\cdot\mathrm{cost}_{OPT}(\sigma_q)+2\cdot S_0+I \\
&\leqq K\cdot\mathrm{cost}_{OPT}(\sigma)+2\cdot\mathrm{cost}_{OPT}(\sigma)+I \\
&\leqq (K+2)\cdot\mathrm{cost}_{OPT}(\sigma)+I \qquad (5.1)
\end{aligned}
$$

となる。ただし、S_0 は規定区間の総移動距離(式(3.1)参照)である。∎

これらの結果の延長として、ここで「もし各全ての要求$\sigma(i)$が中間経由節点を有する(つまり、$p(i)\neq q(i)$)ならば、中間経由節点をもつKサーバー問題に対しては、ある2-competitiveなアルゴリズムが存在する」という予想を与える。この予想は、中間経由節点へサーバーを移動させるための必要コストの和は、規定区間の移動に必要なコストの和に比べて極端に大きくはならないであろうということから来ている。事実、補題2および系7は各々 $K=n-1$、$K=2$に対する 2-competitive なアルゴリズムの存在を示しており、さらに次も証明される：

定理10　*各全ての要求が中間経由節点を有する$(n-2)$サーバー問題に対して、ある 2-competitive なアルゴリズムが存在する。*

(証明)　次のようなオンラインアルゴリズムARTを考える。$(n-2)$台のサーバーを、常に互いに重ならないように配置する。要求$\sigma(i)=\{p(i),q(i)\}$ に対して、もし節点 $p(i)$ もしくは節点 $q(i)$ 上にサーバーが位置しているならば、そのサーバーに規定区間を往復させる（両方の節点にサーバーが位置しているときは、どちらを起用しても良い）。もし節点 $p(i)$，$q(i)$のどちらにもサーバーが存在しないときは、任意のサーバーを選んで $p(i)$ に向かわせ、その後、そのサーバーを $q(i)$ へ移動させる。ARTのコストは次を満たす：

$$\begin{aligned} cost_{ART}(\sigma) &\leqq 2\cdot S_0+D-d_0 \\ &\leqq 2\cdot cost_{OPT}(\sigma)+(D-d_0), \end{aligned} \tag{5.2}$$

ここで、$D=\max\{d(u,v)\}$, d_0 は初期配置において空であった2節点間の距離、また S_0 は規定区間の総移動距離(式(3.1)参照)である。この結果、ARTは 2-competitive である。∎

6　結　論

本研究では、中間経由節点をもつKサーバー問題を考察した。第1に、$K=n-1$、$K=2$に対するオンラインアルゴリズムの構築および“Competitive 解析”によるそれらの最悪性能比の検討を行い、中間経由節点をもたない Manasseら[6] の結果を一般化した。第2に、一般の

Kにおいて、元の問題に対する K-competitive なオンラインアルゴリズムが存在するならば、我々の問題に対して（K＋2）-competitive なオンラインアルゴリズムを構築できることを示した。しかし、幾つかの事項が未解決のままである。今後はそれらの解決を目指すとともに、分散型アルゴリズムや確率アルゴリズム（例えば、[1][4]）の適用も検討していきたい。

参考文献

[1] Y.Batral and A.Rosen. The Distributed k-Server Problem. *Proceedings 33rd Annual Symposium on Foundations of Computer Science*, 344-353, 1992.

[2] A.Borodin, N.Linial and M.Saks. An Optimal Online Algorithm for Metrical Task System. *Proc.19th Annual ACM Symposium on Theory of Computing*, 373-382, 1987.

[3] A.Fiat, Y.Rabani and Y.Ravid. Competitive k-Server Algorithms. *Proceedings 31st Annual Symposium on Foundations of Computer Science* VolⅡ,454-463,1990.

[4] E.Grove. The Harmonic Online K-Server Algorithm Is Competitive. *Proc. 23rd Annual ACM Symposium on Theory of Computing*, 260-266, 1991.

[5] M.S.Manasse, L.A.McGeoch and D.D.Sleator. Competitive Algorithms for On-Line Problems. *Proc. 20th Annual ACM Symposium on Theory of Computing*, 322-333,1988.

[6] M.S.Manasse, L.A.McGeoch and D.D.Sleator. Competitive Algorithms for Server Problems. *Journal of Algorithms*, 11:208-230, 1990.

[7] D.D.Sleator and R.E.Tarjan. Amortized Efficiency of List Update and Paging Rules. *Communications of the ACM*, 28(2):202-208, 1985.