凹２次型輸送問題の解法

02501950　法政大学　*蜂須賀 博和　HACHISUKA Hirokazu
01900070　法政大学　若山 邦紘　WAKAYAMA Kunihiro

1　はじめに

本研究では、以下のような線形制約のもとで、輸送費が凹２次型の輸送コスト最小化問題を取り上げる。

$$f(x) = c^t x - x^t Dx \rightarrow Min$$

sub.to

$$\sum_{j=1}^{n} = a_i, \quad (i = 1, \ldots, m)$$

$$\sum_{i=1}^{m} = b_j, \quad (j = 1, \ldots, n)$$

$$x_{ij} \geq 0 \quad (i = 1, \ldots, m)(j = 1, \ldots, n)$$

$$D : positive\ diagonal\ matrix$$

実社会おける輸送費は大量輸送のためコスト低減が生じ、非線形型コスト関数によって与えられる。このことから局所解が複数存在し、求められた解が大域的最適解とは限らなくなってしまう。

以下に需要地、供給地が複数存在するヒッチコック型輸送問題を示す。(各流量に対して凹型コスト関数が与えられている。)

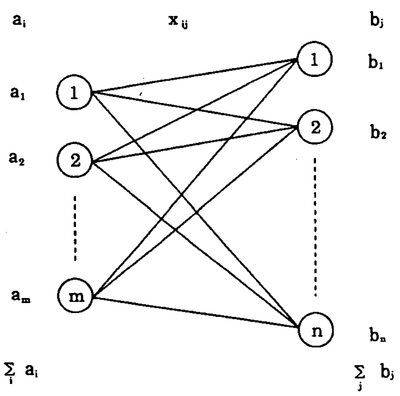

図１　ヒッチコック型輸送問題

*法政大学大学院工学研究科システム工学専攻修士課程

本研究では、降下法、内点罰金関数法及び、切除平面法を用いた凹２次関数の最小化のための解法、及び分枝限定法と *Out-of-Kilter Algorithm* を組み合わせた解法を用いて個々に輸送問題の最小化を図り比較を行なう。

2　解法１

ここでは内点罰金関数法、降下法、切除平面を組み合わせ、最適解を求めるアルゴリズムを提案し、以下に示す。(今回は領域を切りつくした中で最小の値を与える端点を最適解とした。)

Step1 制約領域が空集合でないなら *Step2*、空集合であるなら *Step8* へ

Step2 $F(x)$ が凸となる十分大きな罰金係数 K を与え、制約領域内において初期点を決定する

Step3 降下法を用いて $F(x)$ の最小点を求める

Step4 十分０に近い罰金係数 K をとる

Step5 *Step3* での最小点を初期点とし、$F(x)$ 上で降下法を用いて点列を制約領域の縁に近付ける

Step6 点列が制約領域の縁に近付いた時点で、点列を制約式にのせ、シンプレックス法を用いて端点を求める

Step7 得られた端点において切除平面を生成し、制約領域の一部を取り除く (*Step1* へ)

Step8 終了 :制約領域を取り除いた中で最も最小な値を与える端点が最適解となる

3　解法2

解法2として分枝限定法と *Out-of-Kilter Algorithm*(以下 *OKA*) を組み合わせたアルゴリズムを提案する。(以降に分枝限定法と *OKA* を組み合わせたアルゴリズムを示す。)

3.1　分枝操作及び停止の規則

ここで今回用いた分枝限定法の分枝操作及び停止の条件を以下に記しておく。
分枝操作に関しては各流量ごとに原型関数と近似関数との誤差をとり、最大を与える流量に対して更に近似を行なう。(図2参照)

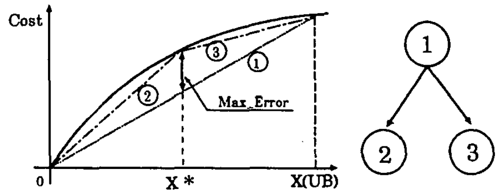

図2　分枝操作の例

次に停止操作について説明する。

$$z^n_{UB} < z^{n+1}_{UB} \rightarrow z^{n+1}_{UB} = z^n_{UB}$$

$$z^n_{LB} > z^{n+1}_{LB} \rightarrow z^{n+1}_{LB} = z^n_{LB}$$

上式によって各ノードでの上限、下限を決定させ、

$$z^n_{LB} > z^n_{UB}$$

となったノードに対してはそれ以降の分枝操作を打ち切る。

Step1 目的関数の線形近似

Step2 *OKA* での最小コストフローの算出

Step3 上限、下限の算出

Step4 上限と下限が一致したら *Step8* へそうでないなら *Step5* へ

Step5 分枝先の決定、下限の中で最小のものを選択

step6 そのノードにおいての最小コストフローにおける目的関数と近似関数の誤差が最大なものを求める

Step7 誤差が最大な目的関数に対し、その流量において線形近似を行なう *Step2* へ

step8 終了 :*Optimal flow & Optimal solution*

4　大規模モデルを適用した解法の考察

今回2つの解法で大規模な輸送問題を解くわけであるが、解法2としては、親ノードの最適流量、ノード価格を子問題に適用し、問題を解くことが可能なので計算の処理時間は親ノードより短いことが期待できる。

解法1においては前年の秋季研究発表において生成した点列が制約式に近付くとステップサイズが微小となり、端点に収束するまでに時間がかかってしまうことが分かった。この原因としては罰金係数 K を十分大きな値で与えているためであり、拡張目的関数を凸にさせる最小の K を求めてやれば、初期点を収束する端点により近く与えてやることができ、処理時間を短縮させることができると考えられる。しかし一般的に凹関数に対して内点罰金関数が凸となる罰金係数を得ることは難しいので、今回も十分大きな罰金係数を与えて問題を解いていく。そこで今回は点列が制約式に近付いたら、点を制約式上にのせ、問題をシンプレックス法を用いて解き、端点を求めるようにする。これにより前回のアルゴリズムより処理時間が短縮されることが期待できるが、次元が増す毎にシンプレックスの処理時間がかさみ、全体的に処理時間が増大すると思われる。

以上のことから推測すると解法2の方が問題を解くステップを重ねていく毎に処理時間が短縮されていくことから、解法1よりも処理時間が速いと考えられる。

結果については研究発表のときに示す。

参考文献

[1] 今野, 山下　「非線形計画法」日科技連 *(1978)*

[2] *D.R.* プレン, 他 「整数計画法入門」 *(1976)*

[3] 蜂須賀 博和, 若山 邦紘　"凹2次型輸送問題の解法" 法政大学経営工学科 卒業論文 *1996*

[4] 蜂須賀 博和, 若山 邦紘　"凹2次型輸送問題の解法" 日本オペレーションズ・リサーチ学会 秋季研究発表会 アブストラクト集 *(1996) 122-123*