数理モデル化と問題解決　21－3
（1998．10．7）

整列問題のネットワークフローモデルによる解釈とその拡張

萩原　斉[†]　中森　眞理雄[††]

線形計画問題を，情報科学の基本的問題の記述に適用した事例報告である．具体的には，整列問題を線形計画問題として記述した例を示し，変数や双対問題の意味のさまざまな角度からの解釈を試みる．さらに，線形計画問題として記述された整列問題を割り当て問題として記述し，それを実現する回路を示す．この回路が $O(n\log_2 n)$ 個の素子で実現できること，また，このようなモデルでは最低でも $O(n\log_2 n)$ 個の素子を要することを示す．
次に，この手法を一般に困難とされている問題に拡張する方法を試みる．具体的には，グラフにおいてクリークが存在するか否かを判定する問題を取り上げ，双線形計画問題として記述した例を示す．

Interpretation of Sorting with Network-flow Models and Extensions

Hitoshi Hagiwara [†] and Mario Nakamori[††]

In this present paper, we show instances to apply linear programming problem to basicaly problems of computer sience. For instance, it is shown that the problem of sorting data is described as linear progamming problems, and we try to interpret one with variables and a dual problem. Futher, a sort problem is described as an assignment problem and an electric circuit. This circuit is made up of $O(n\log_2 n)$ electrical elements at least.
We extend this method to a problem that is generally recognized as hard problem. The instance that a clique problem is described as bilinear programming problem is shown.

1. まえがき

計算機科学においては，比較的能率の良いアルゴリズムが存在するか否かによる問題の複雑さの分類が知られている[1),2)]．一般に，多項式（問題の規模に対して）オーダの計算時間の決定性アルゴリズムが存在する問題はクラスPと呼ばれ，多項式オーダの計算時間の非決定性アルゴリズムが存在する問題はクラスNPと呼ばれる．クラスPには，整列（ソート）や最短路問題，最大流問題，最小費用流問題などがあり，現実的な高速アルゴリズムが存在する．線形計画問題にも多項式時間のアルゴリズム[3)]が知られているが，その時間のオーダは必ずしも低いものではない．

最短路問題，最大流問題，最小費用流問題などは線形計画問題と解釈することができる[4),5)]．経験によれば，高速アルゴリズムが存在する問題の多くは線形計画問題として記述することができる．これは，線形計画問題と双対問題[6)~8)]が，可能解の改善方向を示唆する強力な尺度を提供するためである．

筆者らは，整列問題[9)]も線形計画問題として解釈することが可能と予想し，記述をいろいろ試みた．その結果，整列やそれに関連するいくつかの問題を線形計画問題やネットワークフロー問題として記述したり変数の意味をさまざまな観点から解釈したりすることができたので，それらを事例として報告する．

以下に，多数の数値の中から最大値や最小値を求める問題，数値の大きい方（小さい方）から k 個の和を求める問題，数値の大きい方（小さい方）から k 番目のものを求める問題，数値を整列する問題を，線形計画問題として記述した例を示す．特に，整列問題に対しては，数値を大小半分ずつに分割する問題を最小費用流問題として記述する例を示し，それを再帰的に適用することにより，$O(n\log_2 n)$ 個の変数を用いて n 個の数値を整列する線形計画問題を示す．この結果は，整列問題に対する最良のアルゴリズムの手間が $O(n\log_2 n)$ であることと比較すると，興味深い．

クラスNPの問題を線形計画問題として記述する方法は知られていない（仮にそれが可能であるとするとP=NPとなる）．本論文では，クラスNPの問題の例として，グラフにおけるクリークを求める問題を，双線形計画問題として記述した例を示す．

† 株式会社 構造計画研究所
KOZO KEIKAKU ENGINEERING Inc.
†† 東京農工大学
Tokyo A&T University

2. 整列と線形計画モデル

本節では，与えられた正の数値 $A = \{a_1, a_2, \cdots, a_n\}$ $(a_i \neq a_j;\ i,j = 1,2,\cdots,n;\ i \neq j)$ に対して，最大値や最小値を求める問題および整列する問題などを線形計画問題として記述し，各種変数の意味を論ずる．

2.1 最大値および最小値 – 事例 1 –

A の中の最大値を求める問題を線形計画問題として記述した例を次に示す．

問題 $\pi 1$

maximize

$$z = \sum_{i=1}^{n} a_i x_i,$$

subject to

$$\sum_{i=1}^{n} x_i = 1,$$

$$x_i \geq 0 \quad (i = 1, 2, \cdots, n).$$

ここで，仮に a_{i_0} を最大値とすると，解 $\hat{x}$

$$\hat{x}_{i_0} = 1,\ \hat{x}_i = 0 \quad (i \neq i_0)$$

は可能解であり，別の解 $\tilde{x}_i$ に対して

$$\begin{aligned}
&\sum_i a_i \hat{x}_i - \sum_i a_i \tilde{x}_i \\
&= a_{i_0}(1 - \tilde{x}_{i_0}) - \sum_{i \neq i_0} a_i \tilde{x}_i \\
&\geq a_{i_0}(1 - \tilde{x}_{i_0}) - \max_{i \neq i_0} a_i \sum_{i \neq i_0} \tilde{x}_i \\
&\quad = a_{i_0}(1 - \tilde{x}_{i_0}) - (\max_{i \neq i_0} a_i)(1 - \tilde{x}_{i_0}) \\
&\quad = (a_{i_0} - \max_{i \neq i_0} a_i)(1 - \tilde{x}_{i_0}) \geq 0
\end{aligned}$$

となることから，$\hat{x}$ は最適解であることが証明される．

次に，問題 $\pi 1$ の双対問題を示す．

問題 $\delta 1$

minimize

$$w = y,$$

subject to

$$y \geq a_i \quad (i = 1, 2, \cdots, n).$$

この問題 $\delta 1$ は，「A のいずれに対してもそれ以上である値（つまり A の上界）の中で最小のもの」という意味であり，A が有限集合であることから，A の上界の最小値と A の最大値は一致する．

問題 $\pi 1$ と問題 $\delta 1$ は，互いに双対であるから，両者の最適解に対する目的関数の値は一致する．また，最適解 $\hat{x}$ に対しての問題 $\pi 1$ の制約条件は $\sum_{i=1}^{n} \hat{x}_i = 1$ が成り立っている．このことと，問題 $\delta 1$ の最適解 $\hat{y} = a_{i_0}$ が 0 でないこととは，線形計画問題の相補性条件 (complementary slackness condition) からも裏付けられる．

同様に，A の中の最小値を求める問題とその双対問題を線形計画問題として記述した例を次に示す．

問題 $\pi 2$

minimize

$$t = \sum_{i=1}^{n} a_i u_i,$$

subject to

$$\sum_{i=1}^{n} u_i = 1,$$

$$u_i \geq 0 \quad (i = 1, 2, \cdots, n).$$

問題 $\delta 2$

maximize

$$s = v,$$

subject to

$$v \leq a_i \quad (i = 1, 2, \cdots, n).$$

2.2 大きい方および小さい方から k 個の和 – 事例 2 –

A の中の大きい方から k 個の和を求める問題を，線形計画問題として記述した例を次に示す．

問題 $\pi 3$

maximize

$$z = \sum_{i=1}^{n} a_i x_i,$$

subject to

$$\sum_{i=1}^{n} x_i = k,$$

$$x_i \leq 1 \quad (i = 1, 2, \cdots, n),$$

$$x_i \geq 0 \quad (i = 1, 2, \cdots, n).$$

仮に大きい方から k 個を $a_{i_1}, a_{i_2}, \cdots, a_{i_k}$ とすると，解 $\hat{x}$

$$\hat{x}_i = 1,\ \hat{x}_{\bar{i}} = 0\ (i \in I, \bar{i} \notin I; I = \{i_1, i_2, \cdots, i_k\})$$

は可能解であり，別の解 $\tilde{x}$ に対して

$$\begin{aligned}
&\sum_i a_i \hat{x}_i - \sum_i a_i \tilde{x}_i \\
&= a_{i_1}(1 - \tilde{x}_{i_1}) + a_{i_2}(1 - \tilde{x}_{i_2}) + \cdots \\
&\quad + a_{i_k}(1 - \tilde{x}_{i_k}) - \sum_{i \notin I} a_i \tilde{x}_i \\
&\geq a_{i_1}(1 - \tilde{x}_{i_1}) + a_{i_2}(1 - \tilde{x}_{i_2}) + \cdots \\
&\quad + a_{i_k}(1 - \tilde{x}_{i_k}) - \max_{i \notin I} a_i \sum_{i \notin I} \tilde{x}_i \\
&\geq \min(a_{i_1}, a_{i_2}, \cdots, a_{i_k}) \cdot (k - \tilde{x}_{i_1} - \tilde{x}_{i_2} - \cdots - \tilde{x}_{i_k}) \\
&\quad - \max_{i \notin I} a_i (k - \tilde{x}_{i_1} - \tilde{x}_{i_2} - \cdots - \tilde{x}_{i_k}) \geq 0
\end{aligned}$$

となることから，$\hat{x}$ が最適解であることが証明される．

問題 $\pi 3$ の双対問題は次の通りである．

問題 $\delta 3$

minimize

$$w = ky_0 + \sum_{i=1}^{n} y_i,$$

subject to

$$y_0 + y_i \geq a_i \quad (i = 1, 2, \cdots, n),$$
$$y_i \geq 0 \quad (i = 1, 2, \cdots, n).$$

問題 $\delta 3$ の最適解において，y_0 は大きい方から k 番目の値 a_k に対して $y_0 \leq a_k$ となる．また，a_k より大きい a_i に対応する y_i は $y_i = 0$ となり，a_k より小さい a_i に対応する y_i は $y_i = y_0 - a_i$ となる．

$k = 1$ の場合は問題 $\pi 1$ や問題 $\delta 1$ と同じである筈だが，上記の記述は少し異なっている（問題 $\pi 1$ では条件 "$x_i \leq 1\ (i = 1, 2, \cdots, n)$" がない）．これは，問題 $\pi 1$ では

$$\sum_i x_i = 1, \quad x_i \geq 0\ (i = 1, 2, \cdots, n)$$

であることから，

$$x_i \leq 1\ (i = 1, 2, \cdots, n)$$

が自明に成り立つためである．$x_i \leq 1\ (i = 1, 2, \cdots, n)$ を問題 $\pi 1$ に加えて形式的に双対問題を作ると

問題 $\delta' 1$

minimize

$$w = y + \sum_i u_i,$$

subject to

$$y + u_i \geq a_i \quad (i = 1, 2, \cdots, n),$$
$$u_i \geq 0 \quad (i = 1, 2, \cdots, n).$$

となる．この問題 $\delta' 1$ において，解 $\hat{y}$

$$\hat{y} = a_{i_0}, \quad \hat{u}_i = 0\ (i = 1, 2, \cdots, n)$$

は最適解である．各 $\hat{u}_i\ (i = 1, 2, \cdots, n)$ の値が 0 であることは，問題 $\pi 1$ において $x_i \leq 1\ (i = 1, 2, \cdots, n)$ が冗長であることを意味する．

同様に，A の中の小さい方から k 個の和を求める問題とその双対問題を線形計画問題として記述した例を次に示す．

問題 $\pi 4$

minimize

$$t = \sum_{i=1}^{n} a_i u_i,$$

subject to

$$\sum_{i=1}^{n} u_i \geq k,$$
$$u_i \leq 1 \quad (i = 1, 2, \cdots, n),$$
$$u_i \geq 0 \quad (i = 1, 2, \cdots, n).$$

問題 $\delta 4$

maximize

$$s = kv_0 - \sum_{i=1}^{n} v_i,$$

subject to

$$v_0 - v_i \leq a_i \quad (i = 1, 2, \cdots, n),$$
$$v_i \geq 0 \quad (i = 1, 2, \cdots, n).$$

2.3 大きい方から k 番目のもの －事例 3－

A の中の大きい方から k 番目の値は，大きい方から k 個と小さい方から $n-k+1$ 個の両方に含まれるので，両者の和から全体の和 $a_1 + a_2 + \cdots + a_n$ を減じたものに等しい．そこで，最大化問題である問題 $\pi 3$ と問題 $\delta 4$ を組み合わせることにより，大きい方から k 番目の値を求める問題を線形計画問題として次のように記述することができる．

問題 $\pi 5$

maximize

$$h = \sum_{i=1}^{n} a_i x_i + (n-k+1)v_0 - \sum_{i=1}^{n} v_i - \sum_{i=1}^{n} a_i,$$

subject to

$$\sum_{i=1}^{n} x_i \leq k,$$
$$x_i \leq 1 \quad (i = 1, 2, \cdots, n),$$
$$x_i \geq 0 \quad (i = 1, 2, \cdots, n),$$
$$v_0 - v_i \leq a_i \quad (i = 1, 2, \cdots, n),$$
$$v_i \geq 0 \quad (i = 0, 1, 2, \cdots, n).$$

問題 $\pi 5$ の双対問題，小さい方から k 番目のものを求める主問題およびその双対問題は，この問題 $\pi 5$ と同様に，前述の問題 $\pi 3$, $\pi 4$, $\delta 3$, $\delta 4$ を組み合わせた問題として記述することができる．

2.4 整列 －事例 4－

A を大きい方から順に並べる問題（整列あるいはソート[2)]）およびその双対問題を特殊な線形計画問題である割り当て問題として記述してみる．

割り当て問題とは，n 人の従業員と n 個の仕事があり，従業員 i に仕事 j を割り当てるときのメリットが c_{ij} であるとき，メリットの総和が最大になるように従業員と仕事を 1 対 1 に対応付ける問題である[4)]．

この問題は，線形計画問題として次のように記述できることが知られている．

問題 P0

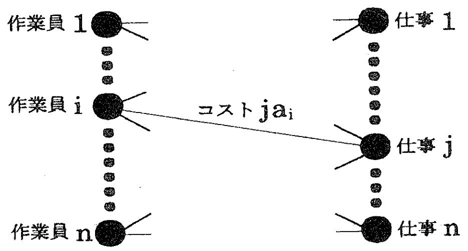

図 1　整列問題の割り当て問題としての解釈
Fig. 1 The interpretation sort problem as assign problem

maximize

$$z = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij},$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1, 2, \cdots, n),$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1, 2, \cdots, n),$$

$$x_{ij} \geq 0 \quad (i, j = 1, 2, \cdots, n).$$

問題 P0 の制約条件では，変数 x_{ij} $(i, j = 1, 2, \cdots, n)$ は特に "離散値をとる" と断っていないが，問題 P0 の最適解において変数 x_{ij} の値は 0 または 1 となることが知られている．

問題 P0 の最適解 x_{ij} の意味は，次のとおりである．

$$x_{ij} = \begin{cases} 1 & \text{従業員 } i \text{ に仕事 } j \text{ を割り当てる．} \\ 0 & \text{従業員 } i \text{ に仕事 } j \text{ を割り当てない．} \end{cases}$$

この考え方を整列に応用する（図 1）．

いま，数値 a_i を従業員 i，順位 j を仕事 j と考え，従業員 i に仕事 j を割り当てる（数値 a_i が大きい方から j 番目の値である）ときの "メリット" を ja_i と考えると，

問題 $\pi 6$

minimize

$$z = \sum_{i=1}^{n}\sum_{j=1}^{n} ja_i x_{ij},$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \quad (i = 1, 2, \cdots, n),$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad (j = 1, 2, \cdots, n),$$

$$x_{ij} \geq 0 \quad (i, j = 1, 2, \cdots, n).$$

問題 $\delta 6$

maximize

$$w = \sum_{i=1}^{n} y_i + \sum_{j=1}^{n} q_i,$$

subject to

$$q_j + y_i \leq ja_i \quad (i, j = 1, 2, \cdots, n).$$

問題 $\pi 6$ の最適解における x_{ij} が離散値 0 または 1 をとることは，一般の割り当て問題から自然に導かれる．

これに対して，A を小さい方から順に並べる問題およびその双対問題を線形計画問題として記述した例は次のとおりになる．

問題 $\pi 7$

maximize

$$t = \sum_{i=1}^{n}\sum_{j=1}^{n} ja_i u_{ij},$$

subject to

$$\sum_{j=1}^{n} u_{ij} = 1 \quad (i = 1, 2, \cdots, n),$$

$$\sum_{i=1}^{n} u_{ij} = 1 \quad (j = 1, 2, \cdots, n),$$

$$u_{ij} \geq 0 \quad (i, j = 1, 2, \cdots, n).$$

問題 $\delta 7$

minimize

$$s = \sum_{i=1}^{n} v_i + \sum_{j=1}^{n} p_j,$$

subject to

$$p_j + v_i \leq ja_i \quad (i, j = 1, 2, \cdots, n).$$

問題 $\pi 6$（問題 $\pi 7$）において変数 x_{ij}（変数 u_{ij}）は，a_i が A の中で大きい（小さい）方から j 番目のときに 1 でその他のときに 0 になる．問題 $\pi 6$（問題 $\pi 7$）が整列問題であることは，一般に $g_1 \leq g_2 \leq \cdots \leq g_n$ および $h_1 \leq h_2 \leq \cdots \leq h_n$ が成り立つとき，$1, 2, \cdots, n$ のいかなる順列 $p_1, p_2, \cdots, p_n$ に対しても，

$$\begin{aligned} & g_1 h_n + g_2 h_{n-1} + \cdots + g_n h_1 \\ \leq\ & g_1 h_{p_1} + g_2 h_{p_2} + \cdots + g_n h_{p_n} \\ \leq\ & g_1 h_1 + g_2 h_2 + \cdots + g_n h_n \end{aligned}$$

であることから分かる．

ネットワークフローの理論によれば，割り当て問題は図 2 のような定電圧源，定電流源，ダイオードからなる電気回路の解として解くことができる．

3．整列問題のネットワークフローモデル − 事例 5 −

問題 $\pi 6$（問題 $\pi 7$）では，整列問題を n^2 個の変数

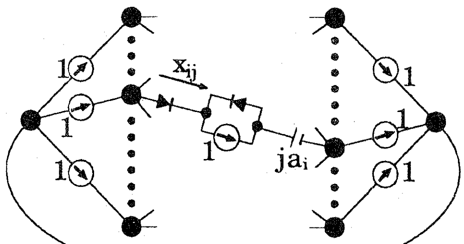

図 2 割り当て（整列）問題を解く回路
Fig. 2 The circuit to solve assign(sort) problems

x_{ij} $(i, j = 1, 2, \cdots, n)$ を用いて割り当て問題として定式化した．本節では，整列問題を問題 $\pi 6$ よりも少ない個数の変数を用いた線形計画問題として記述してみる．

最小費用流問題とは，ある品種を m 箇所の生産地から n 箇所の消費地へ最小費用で運ぶための輸送量を決定する問題である．生産地 i では s_i だけ生産され，消費地 j では d_j だけ消費される．ここで，

$$\sum_{i=1}^{m} s_i = \sum_{j=1}^{n} d_j$$

を仮定しておく．

生産地 i から消費地 j へ品物を 1 単位運ぶのに要する費用を e_{ij} とする．以上，$s_i(i = 1, 2, \cdots, m)$, $d_j(j = 1, 2, \cdots, n)$, $e_{ij}(i = 1, 2, \cdots, m;\ j = 1, 2, \cdots, n)$ はデータとして与えられているとする．

生産地 i から消費地 j へ運ぶ量を x_{ij} とすると，最小費用流問題は次のとおりに記述される．

<u>問題 $\pi 8$</u>

minimize
$$z = \sum_{i=1}^{m} \sum_{j=1}^{n} e_{ij} a_i x_{ij},$$
subject to
$$\sum_{j=1}^{n} x_{ij} = s_i \quad (i = 1, 2, \cdots, m),$$
$$\sum_{i=1}^{m} x_{ij} = d_j \quad (j = 1, 2, \cdots, n),$$
$$x_{ij} \geq 0 \quad (i = 1, 2, \cdots, m;\ j = 1, 2, \cdots, n).$$

この最小費用流問題において，$m = n$, $s_i = 1(i = 1, 2, \cdots, n)$, $d_j = 1(j = 1, 2, \cdots, n)$ とすれば，割り当て問題になる．ただし，従業員 i を仕事 j に割り当てるときのメリット c_{ij} に対して $M - c_{ij}$ を e_{ij} とする．ただし，M は十分大きな数である．

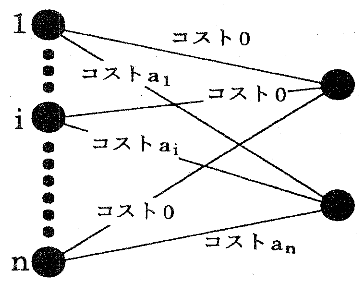

図 3 数値を大きい方と小さい方に分ける回路
Fig. 3 The circuit to divide numbers to upper and lower

ここで，n 箇所の生産地 $1, 2, \cdots, n$ と 2 箇所の消費地 1, 2 の最小費用流問題を考える．各生産地における生産量は 1，各消費地における消費量は $\frac{n}{2}$ とする（n が奇数の場合は，一方の消費量を $\frac{n+1}{2}$，他方の消費量を $\frac{n-1}{2}$ とする）．

また，単位輸送量あたりの費用は生産地 i から消費地 1 へは 0，消費地 2 へは a_i とする．いま，

$c_1 = 0, \quad c_2 = 1$

と定めるなら，生産地 i から消費地 j への単位輸送量あたりの費用は $c_j a_i$ と書ける．以上の最小費用流問題は，次のとおりに記述できる．

<u>問題 $\pi 9$</u>

minimize
$$z = \sum_{i=1}^{n} \sum_{j=0}^{1} c_j a_i x_{ij},$$
subject to
$$\sum_{j=0}^{1} x_{ij} = 1 \quad (i = 1, 2, \cdots, n),$$
$$\sum_{i=1}^{n} x_{ij} \leq n/2 \quad (j = 0, 1),$$
$$x_{ij} \geq 0 \quad (i = 1, 2, \cdots, n;\ j = 0, 1).$$

問題 $\pi 9$ の最適解 $\hat{x}_{ij}$ は，$a_1, a_2, \cdots, a_n$ を大きい方半分と小さい方半分に分ける．すなわち，a_i が大きい方から $\frac{n}{2}$ 番目（n が奇数の場合は $\frac{n+1}{2}$ 番目）以内にあるならば，

$\hat{x}_{i1} = 1, \quad \hat{x}_{i2} = 0$

であり，さもなければ

$\hat{x}_{i1} = 0, \quad \hat{x}_{i2} = 1$

となる．

以上により，$a_1, a_2, \cdots, a_n$ を大小半分に分ける問題を変数の個数が $O(n)$ の最小費用流問題として記述できることが示された．この最小費用流問題も，電気回路によって解くことができる（図 3）．

ここで，得られた大きい方（小さい方）半分に対し

て同様の問題を考える．それぞれの問題の変数の個数は前の問題の変数の個数の半分であるから，両者を合わせて $O(n)$ となる．この考え方を再帰的に適用すれば，最終的に $a_1, a_2, \cdots, a_n$ を整列する一連の問題が得られる．再帰の段数は $O(\log_2 n)$ であるから，変数の個数の合計は $O(n \log_2 n)$ となる．このことを，n 個の数値を整列する最良のアルゴリズムの計算複雑度が $O(n \log_2 n)$ であることと比較することは興味深い．

4. クリーク存在判定問題と双線形計画モデル – 事例6 –

本節では，与えられたグラフ G に対して，その節点数が K ($K \leq |V|$) である完全部分グラフ（クリーク）が存在するか否かを判定する問題を双線形計画問題として記述する例を示す．

$v = |V|$ および $e = |E|$ とし，次に示す隣接行列 $\mathcal{G} = (g_{ij})$ $(i, j = 1, 2, \cdots, v)$ によりグラフ G を表現する．ただし，ここでは単純無向グラフを対象とするため $g_{ij} = g_{ji}$ が成立する．

$$g_{ij} = \begin{cases} 1 & 2\text{ 節点 } i, j \text{ 間に枝が存在するとき} \\ 0 & \text{上記以外のとき} \end{cases}$$

大きさ K のクリーク存在判定問題を双線形計画問題として記述した例を次に示す．

問題 10

minimize

$$\varphi = \sum_{j=1}^{v} \{x_j(1 - u_j) + (1 - x_j)u_j\},$$

subject to

$$\sum_{j=1}^{v} x_j = K, \qquad (1)$$

$$\sum_{j=1}^{v} (1 - g_{ij})x_j \leq 1 \quad (i = 1, 2, \cdots, v), \qquad (2)$$

$$0 \leq x_j \leq 1 \quad (j = 1, 2, \cdots, v),$$

$$0 \leq u_j \leq 1 \quad (j = 1, 2, \cdots, v).$$

問題 10 の目的関数 φ は，すべての (x_j, u_j) の組合せに対して

$$(x_j, u_j) = (0, 0) \text{ or } (1, 1)$$

が成立するとき，かつこのときに限り最大値 v をとる．

また，(1) は「クリークに含まれる節点数は K でなければならない」という条件を表し，(2) は「間に枝が存在しない 2 節点は同時にクリークに含まれることはできない」という条件を表す．

したがって，問題 10 の最適解において $\varphi = v$ が成立するとき，連続変数 x_j $(j = 1, 2, \cdots, v)$ は次のような値をとり，与えられたグラフ G に大きさ K のクリークが存在する．

$$x_j = \begin{cases} 1 & \text{節点 } j \text{ がクリークに含まれるとき} \\ 0 & \text{上記以外のとき} \end{cases}$$

5. まとめ

数値の集合が与えられたときに、最大値や最小値を求める問題，大きい方（小さい方）から k 個の和を求める問題，大きい方（小さい方）から k 番目のものを求める問題，数値を整列する問題を，線形計画問題として記述した例を示した．特に，整列問題に対しては，数値を大小半分ずつに分割する問題を最小費用流問題として記述する例を示し，それを再帰的に適用することにより，$O(n \log_2 n)$ 個の変数を用いて n 個の数値を整列する線形計画問題を示した．整列問題に対する線形計画問題は最小費用流問題として記述されているので，それを実現する電気回路を示した．このような回路を入力されたデータから作り出す方法は今後の研究課題である．

整列問題を線形計画問題として記述する方法の拡張としてグラフにおけるクリークを求める問題を，双線形計画問題として記述してみた．本論文の方法で記述された双線形計画問題を解くアルゴリズムも今後の課題である．

参考文献

1) E. Börger, *Computability, Complexity, Logic*, North-Holland, 1989.
2) A. V. Aho, J. E. Hopcroft and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.
3) N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, 4, 374-395 (1984).
4) M. Iri, *Network Flow, Transportation, and Scheduling – Theory and Algorithms*, Academic Press, 1969.
5) L. R. Ford and D. R. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.
6) 茨木俊秀, 福島雅夫, "最適化の手法," 共立出版, 1996.
7) V. Chvátal, *Linear Programming*, W. H. Freeman and Company, 1983.
8) G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, 1963.
9) D. E. Knuth, *The Art of Computer Programming*, Vol.3 (Sorting and Searching), Addison-Wesley, 19.