定式化集

tokoharu

平成 26 年 12 月 22 日

1　LP と主要問題

1.1　LP, LP の双対問題

LP 標準系

$$
\begin{aligned}
\text{maximize} \quad & c^T x & & (1)\\
\text{subject to} \quad & Ax \leq b & & (2)\\
& x_i \geq 0 & i = 1, ..., n & (3)
\end{aligned}
$$

双対 LP 標準系

$$
\begin{aligned}
\text{minimize} \quad & b^T y & & (4)\\
\text{subject to} \quad & A^T y \geq c & & (5)\\
& y_j \geq 0 & j = 1, ..., m & (6)
\end{aligned}
$$

ただし，$A : m * n$ 行列，$x : n$ 列ベクトル，$y : m$ 列ベクトル

例 :

$$A = \begin{bmatrix} 1 & 1 \\ 5 & 2 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 3 \end{bmatrix}, c = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

のとき最適解 $x = (1/3, 2/3), \gamma = 4/3$，を得る．図を描けばわかりやすい．これの双対 LP 解は $y = (1/3, 1/3), \gamma = 4/3$ である

1.1.1　不等式

最小化問題	最大化問題
変数	制約式
$\leq$	$\geq$
自由	$=$
$\geq$	$\leq$
制約式	変数
$\leq$	$\leq$
$=$	自由
$\geq$	$\geq$

1.2 最短経路問題と LP

概要

1. 自己ループ無し最短経路問題について
2. LP 形から最短経路問題への変換

線形計画問題として記述する

$s-t$ 最短経路問題 (自己ループが存在しないと仮定) とほぼ (?) 同値の問題を解説する．

しかし任意の $s-t$ 最短経路問題を下記の形で書くことで下記 LP の解とは一致しないことに注意が必要．LP から $s-t$ 最短経路問題の変形は可能．

グラフを作り，Bellman-Ford を走らせる場合を考える．負閉路が検出されれば双対問題は解が存在せず，主問題では負へ発散となる．x_t が INF のままなら，双対問題では正へ発散，主問題では解なしとなる．

主問題の式はややわかりにくい形である．この形で出題はなかなかされないように思う

双対問題の式の形式で比較的よく出題される．” 差分制約” とおぼえておくのが良いだろう．競技プログラミング界隈では俗に牛ゲーと呼ばれる．

LP 標準系 ($|E|$ 変数，$|V|$ 制約)

$$\begin{aligned}
&\text{minimize} && \sum_{e=(i,j)\in E} c_e y_e && && (7)\\
&\text{subject to} && \sum_{e=(j,v)\in E} y_e - \sum_{e=(v,j)\in E} y_e \geq 0 && v \neq t && (8)\\
& && \sum_{e=(j,v)\in E} y_e - \sum_{e=(v,j)\in E} y_e \geq 1 && v = t && (9)\\
& && y_e \geq 0 && e \in E && (10)
\end{aligned}$$

双対 LP 標準系 ($|V|$ 変数，$|E|$ 制約)

$$\begin{aligned}
&\text{maximize} && x_t - x_s && && (11)\\
&\text{subject to} && x_j - x_i \leq c_e && e = (i,j) \in E && (12)\\
& && x_v \geq 0 && v \in V && (13)
\end{aligned}$$

1.3 最大流問題と LP

最大流最小カット定理は有名な定理だが，ここではその定式化とその類型を見る．

与えられる変数は c_{ij} である．これは各辺の最大流量を意味する

LP 主問題

x_{ts} を最大化する循環最大流問題を考える．

したがって x_{ts} を辺集合 E に付け加えて E' とする．

この状況下で $|E|+1$ 変数 $|E|+|V|$ 制約となる．

$$
\begin{aligned}
\text{maximize} \quad & x_{ts} && && (14)\\
\text{subject to} \quad & x_e \le c_e && e=(i,j)\in E && (15)\\
& \sum_{e=(j,v)\in E'} x_e - \sum_{e=(v,j)\in E'} x_e \le 0 && (v\in V) && (16)\\
& x_e \ge 0 && e\in E' && (17)
\end{aligned}
$$

E, E' がそれぞれ登場していることに注意

LP 双対問題

v の不等式に対応する変数が p であり，e の不等式に対応する変数が y であることに注意.

$|E|+|V|$ 変数 $|E|+1$ 制約である

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e=(i,j)\in E} c_e y_e && && (18)\\
\text{subject to} \quad & y_e + p_j - p_i \ge 0 && e=(i,j)\in E && (19)\\
& p_s - p_t \ge 1 && && (20)\\
& p_v \ge 0, y_e \ge 0 && v\in V, e\in E && (21)
\end{aligned}
$$

$p_s - p_t = 1$ の場合が最適ということはわかる．なので平行移動させて $p_s = 1, p_t = 0$ としてよい．頂点に対応する p_i のうちどれを 1 にしてどれを 0 にするかという問題になり y_e はギリギリのところまで抑えるのがよいので結局最小カット問題であることがわかる

1.4 最小費用流と LP

最小費用流も有名である．

この問題では辺に対して２つの変数が与えられる．

c_{ij}, u_{ij} であり，それぞれ ij 間に 1 流すコスト，ij 間に流せる流量である．

さらに，st 間に流せる量 F も与えられる．

最大流問題のときと同じように $E' = E + \{(t,s)\}$ としておく．

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e=(i,j)\in E} c_e x_e && && (22)\\
\text{subject to} \quad & x_e \le u_e && e=(i,j)\in E && (23)\\
& \sum_{e=(j,v)\in E'} x_e - \sum_{e=(v,j)\in E'} x_e = 0 && v\in V && (24)\\
& x_{ts} = F && && (25)\\
& x_e \ge 0 && e\in E && (26)
\end{aligned}
$$

2 ネットワークフローでの定式化

2.1 maximum closure problem / project selection problem

N 個の頂点がある．最初どの頂点も集合 B に属しているが，これを集合 A に移すことで利益を最大化したい．

i が A に属していた時には利得 p_i を得る．

さらに，x_j が A に属し，かつ y_j が B に属していた時に z_j だけ損失をする．
最大スコアを求めよ

この問題に対する解は

$$\sum_{v \in V} \max(0, p_v) - \mathrm{maxflow}(s, t)$$

で求めることができる．
ただし $maxflow(s,t)$ とは
$p_v > 0$ であれば s から v に容量 p_v の辺を張り
$p_v < 0$ であれば v から t に容量 $-p_v$ の辺を張り
x_j から y_j に容量 z_j の辺を張ったときの st 間の最大流である．

最小カットを用いて解く問題で難しい問題はこの問題を通して考えると見通し良く考えることができる場合があるので有用である．

2.2 有理数フローについて注意

下記 Problem でも紹介するとおり， 有理数の最大流問題の解を求めなければならない場合がある． 単純に Dinic のアルゴリズムで int を double に変えれば良いかというと実際には調整が難しい面がある (過去に泥沼にハマった経験がある)． したがって整数フローに直して解く方がエンバグを少なくできる．

3 Problems

3.1 AOJ2230; How to Create a Good Game

問題概要
DAG が与えられる．DAG の最大長を維持しながら辺に長さを足す．足せる長さの最大値を答えよ
$|E| + |V|$ 変数， $|E| + 1$ 制約．

$$\begin{aligned}
&\text{maximize} && \sum_{e=(i,j) \in E} a_e && && (27)\\
&\text{subject to} && a_e - x_j + x_i \leq -c_e && e = (i,j) \in E && (28)\\
& && x_t \leq D && && (29)\\
& && x_v \geq 0, a_e \geq 0 && v \in V, e \in E && (30)
\end{aligned}$$

双対を取る．

$$\begin{aligned}
&\text{minimize} && Dp_t - \sum_{e=(i,j) \in E} c_e y_e && && (31)\\
&\text{subject to} && -\sum_{e=(j,v) \in E} y_e + \sum_{e=(v,j) \in E} y_e \geq 0 && v \in V, v \neq t && (32)\\
& && p_t - \sum_{e=(j,v) \in E} y_e \geq 0 && v = t && (33)\\
& && y_e \geq 1 && e \in E && (34)\\
& && y_e \geq 0, p_t \geq 0 && e \in E && (35)
\end{aligned}$$

これは p_t を元々のグラフの頂点 t から t' への辺の容量と解釈することで，最低流量 1 の $s-t'$ 最小費用流であると解釈できる．

3.2 ジョブ割当問題

n 個のジョブと m 人の作業員がいる．S_i はジョブ i を実行できる作業員の集合である．以下の様な定式化のもとで x_{ij} としてありうる最適解をひとつ求めよ．

$$\begin{aligned}
&\text{minimize} && \max_{j=1,\ldots,m} \sum_{i:j\in S_i} x_{ij} && && (36)\\
&\text{subject to} && \sum_{j\in S_i} x_{ij} = t_i && i = 1,\ldots,n && (37)\\
& && && && (38)
\end{aligned}$$

max が出現した時の常套手段はそれに変数をつけて max の構成要素を制約式に持ってくることである

$$\begin{aligned}
&\text{minimize} && T && && (39)\\
&\text{subject to} && \sum_{j\in S_i} x_{ij} = t_i && i = 1,\ldots,n && (40)\\
& && \sum_{i:j\in S_i} x_{ij} \le T && j = 1,\ldots,m && (41)
\end{aligned}$$

この式で，ある値 T を決め打ちした時に解が存在するかどうかは判定が可能である． 頂点 s と頂点 i の間に容量 t_i の辺を張り， j と t の間に容量 T の辺を張って $\sum_{i=1}^{n} t_i$ だけ最大流が流れるかどうかをチェックすればよい．

あとは T を二分探索すれば最適解を見つけることができる．

3.3 POJ3155; Hard Life

無向グラフ G が与えられる．この部分グラフの中で (辺数) / (頂点数) を最大化されるような部分グラフを求めよ (頂点数 ≤100， 辺数 ≤1000)

二分探索で解を求めることを考える．すなわち，(辺数)/(頂点数) > x なるような部分グラフが存在するかどうかを判定する．

式変形をすると

$$(\text{辺数}) - x(\text{頂点数}) > 0 \tag{42}$$

なる部分グラフを持つかを判定する問題になる．

(42) 式の最大値が 0 より大きいことを判定することは ProjectSelection 問題を適用すれば判別できる． (注 : ProjectSelection 問題において最適値は 0 以上の値になる．(42) 式の最大値は，部分グラフとして空のグラフを選べると考えれば 0 以上になる．しかし空グラフは元の問題の定義では定義ができない．したがって，最適値が 0 になるような x では元の問題の実行可能でない解である可能性がある．そのため，(辺数) − x(頂点数) ≤ 0 という式で考えようとしたり，他の問題でぴったり 0 になるときのグラフを復元しようと思った時には苦労する可能性がある．)

辺と頂点の集合を利得最大になるように選ぶ問題という風に考えて， 辺を選ぶと 利得が+1 される， 頂点を選ぶと 利得が $-x$ されるという風に解釈すれば良い． さらに， ある辺を使用すればその両端点は使用しなければならない． この制約も ProjectSelection で表現可能である．

したがって (パラメータ x によって作られた ProjectSelection 問題の解) > 0 を判定する二分探索を実装すれば良い．

謝辞

この文章を書く際，Mi_Sawa さんに文章を添削をして頂きました．この場を借りて感謝いたします．

編集履歴

2014/10/17	アジア地区予選用のライブラリの一部として書く
	LP，最短経路，最大流，最小費用流, Maximum Closure Problem,
	How to create a good game，ジョブ割当問題を書く
2014/12/14(?)	有理数フロー, Hard Life を追記する。
2014/12/22	記事の修正（修正箇所は多い）