2002年日本オペレーションズ・リサーチ学会
春季研究発表会

ナップサック制約付き最大全域木問題の一解法*

01700900 防衛大学校情報工学科 山田 武夫† YAMADA Takeo
加入申請中 防衛大学校情報工学科 渡辺宏太郎 WATANABE Kohtaro
01107880 防衛大学校情報工学科 片岡 靖詞 KATAOKA Seiji

1 はじめに

無向グラフ $G = (V, E)$ において, 各枝 $e \in E$ に重量 $w(e)$ と利得 $p(e)$ が付されているとする. G の全域木 T について, その重量 $w(T)$ と利得 $p(T)$ を, それぞれ, T に含まれる枝の重量および利得の和と定義する. 本稿では次のナップサック制約付き最大全域木問題（**KCMST**: knapsack constrained maximum spanning tree problem）を考察する.

$$
\begin{array}{lll}
\text{Maximize} & z := p(T) & (1) \\
\text{subject to} & w(T) \le C, & (2) \\
 & T \text{ は全域木} & (3)
\end{array}
$$

以下では重量最小の全域木を $T^w_{\min}$, 利得最大の全域木を $T^p_{\max}$ と記し, さほど一般性を失うことなく次を仮定する.

$\mathbf{A}_1$. C と枝重量, 利得はすべて正整数.

$\mathbf{A}_2$. $w(T^w_{\min}) \le C$.

$\mathbf{A}_3$. $w(T^p_{\max}) > C$

このとき, 次がいえる.

定理 1 **KCMST** は $\mathcal{NP}$-困難 [1] である.

2 ラグランジュ緩和

KCMST のラグランジュ緩和問題を

$$
\begin{array}{lll}
\text{Maximize} & p(T) + \lambda(C - w(T)) & (4) \\
\text{subject to} & T \text{ は全域木} & (5)
\end{array}
$$

とする. この問題は枝 $e \in E$ の重量が $p(e) - \lambda w(e)$ のときの最大全域木問題で容易に解けるが, その最適値を $L(\lambda)$, 最適解（の一つ）を T_λ とすると, 以下が成立する.

命題 1

(i) 任意の $\lambda \ge 0$ に対し, $L(\lambda)$ は **KCMST** の上界値を与える.

(ii) $L(\lambda)$ は $\lambda \ge 0$ で下に凸な区分的線形関数.

(iii) $L(\lambda)$ が λ で微分可能ならば,

$$dL(\lambda)/d\lambda = C - w(T_\lambda). \tag{6}$$

$\lambda \ge 0$ で $L(\lambda)$ が最小となる点を λ^* ($\min\{L(\lambda) \mid \lambda \ge 0\} = -\infty$ のときは, $\lambda^* = \infty$) とすると, $\bar{z} := L(\lambda^*)$ は最良の上界値を与える. また, $T^* := T_{\lambda^*}$, $T^+ := T_{\lambda^*+\epsilon}$ とする. ここに, ϵ は十分に小さい正数である. このとき, 次が証明出来る.

命題 2

(i) $\lambda^* = \infty$ ならば, **KCMST** は実行不可能である.

(ii) $T_0 (= T^p_{\max})$ が実行可能 ($w(T_0) \le C$) ならば, $\lambda^* = 0$ で, T_0 は **KCMST** の最適解である.

(iii) T^* において, $w(T^*) = C$ ならば, T^* は **KCMST** の最適解である.

(iv) T^+ は実行可能 ($w(T^+) \le C$) で, $\underline{z} := p(T^+)$ は **KCMST** の下界値を与える.

仮定 $\mathbf{A}_2$, $\mathbf{A}_3$ のもとでは, 上の (i), (ii) は生起しない. そこで, $\lambda_0 = 0$ と $C - w(T_{\lambda_1}) \ge 0$ であるような λ_1 を取り, $[\lambda_0, \lambda_1]$ を初期区間として次の２分探索法によって λ^* を求める.

Step 1: $\lambda := (\lambda_0 + \lambda_1)/2$.
Step 2: T_λ を求める.
Step 3: $w(T_\lambda) \le C$ なら $\lambda_1 := \lambda$, そうでなければ $\lambda_0 := \lambda$ とする.
Step 4: 区間幅が十分小さくなれば終了, そうでなければ Step 1 へ戻る.

3 近似解

命題２より, 近似解 T^+ を得たが, これを局所探索法によりさらに改善する事が出来る. 一般に, 全域木 T に対して T に含まれない枝 e を付加するとサイクルが生じる. そこで, そのサイクルから e 以外の枝 e' を除くと, 再び全域木 $T \cup \{e\} \setminus \{e'\}$ を得るが, このようにし

*富山国際会議場 大手町フォーラム, H14.3.27-28

†E-mail: yamada@nda.ac.jp

て T から一組の枝の交換によって得られる全域木全体の集合を T の近傍 $N(T)$ という.

本稿の局所探索法は, 近似解 $T := T^+$ から出発して, $N(T)$ を探索し, 実行可能で T よりも 利得の大きい全域木が見つかり次第, 解を更新するという操作を可能な限り反復する, というものである.

4 厳密解法

4.1 分枝限定法

F ($\subseteq E$) をサイクルを含まない枝の集合とし, R ($\subseteq E$) を F と素な枝集合とする. このとき, Fの枝をすべて含み, Rの枝を全く含まないような全域木を (F, R)-許容な全域木という. **KCMST** で, 条件 (3) を

$$T \text{ は } (F, R)\text{-許容な全域木} \tag{7}$$

に置き換えた問題を部分問題 $P(F, R)$ と呼ぶと, 元の問題は, $P(\emptyset, \emptyset)$ となる.

第2節の議論を部分問題 $P(F, R)$ の場合に修正することは容易で, それにより上界値 $\bar{z}(F, R)$, 下界値 $\underline{z}(F, R)$, 近似解 $T(F, R)$ などが得られる. $P(F, R)$ を処理している時点での暫定値を $\hat{z}$ とすると, ここでは以下の処理を行う.

1. $P(F, R)$ が実行不可能, または $\bar{z}(F, R) \leq \hat{z}$ の場合, $P(F, R)$ は最適解を含まないので終端する.
2. (命題2の(ii), (iii) により) $P(F, R)$ の厳密解が得られた場合, 必要なら暫定解 (値) を更新して終端.

部分問題 $P(F, R)$ が終端されない場合には, (F, R)-許容な近似解 $T(F, R) = F \cup \{e^1, e^2, \cdots, e^k\}$ を用いて問題を以下のような子問題群に分割する. すなわち, $i = 1, \cdots, k$ に対して $F_i = F \cup \{e^1, e^2, \cdots, e^{i-1}\}$, $R_i = R \cup \{e^i\}$ としてk個の部分問題 $P(F_i, R_i)$ を考えると, 各子問題の実行可能領域は互いに素で, その和集合は $P(F, R)$ のそれに一致するので, 各 $P(F_i, R_i)$ $(i = 1, \cdots, k)$ が最適に解ければ, $P(F, R)$ も解けたことになる.

4.2 区間縮小法

分枝限定法では, 通常最初の暫定値を $-\infty$ とする. これにより, 厳密解が必ず得られるが, 暫定値が十分に大きくない初期の段階で (本来なら終端すべき) 部分問題が多数生き残ってしまい, 膨大な計算時間を要することが多い.

ところで, 元問題 $P(\emptyset, \emptyset)$ の上下界値$\underline{z} := \underline{z}(\emptyset, \emptyset)$, $\bar{z} := \bar{z}(\emptyset, \emptyset)$ が分かっているので, これらの中間の適当な値を仮想的な暫定値として, 分枝限定法を実行することが考えられる. この値は大きいほど生成される部分問題数が少なくなり, 計算時間も短くてすむが, 最適値 $z^\star$ より大きい値を指定すると, すべての部分問題を見切ってしまい, 最適解を見出すことなく終了してしまう. そこで, 次のような区間縮小法を提案する.

1. 適当な下界値 $\underline{z}$ と上界値 $\bar{z}$ をとる.
2. $\hat{z} := \alpha\underline{z} + (1 - \alpha)\bar{z}$.
3. $\hat{z}$ を仮想暫定値として分枝限定法を実行する. 最適解が得られればそれを表示して終了.
4. $\bar{z} := \hat{z}$として, ステップ2へ戻る.

5 数値実験

前節の解法について, 数値実験によりその性能評価を行っているが, 表1に完全グラフ K_n で, 枝重量と利得が $[1, 100]$ 間で一様かつ独立な場合の結果を示す. ナップサック容量は $C = 20(n - 1)$ で, $\alpha = 0.1$ とした. 各行はそれぞれの例題について, 100 回の独立な試行の平均値で, 最適値 ($z^\star$), 区間縮小法の反復回数 (#rep.), 生成された部分問題の総数 (#sub), IBM RS/6000 Model 270 での CPU 時間 (秒) を表す. 計算時間は必ずしも n に比例しないが, 約 $n = 200$ 程度までが数百秒で解けた.

表 1. 実験結果

例題	$z^\star$	#rep.	#sub	CPU sec.
K_{20}	1698.6	2.0	69.7	0.0540
K_{40}	3673.3	1.3	187.9	0.8720
K_{60}	5686.3	1.0	179.1	1.8930
K_{80}	7682.7	1.0	352.6	6.5370
K_{100}	9686.5	1.0	376.7	12.4810
K_{120}	11701.9	1.0	468.5	23.6990
K_{140}	13717.3	1.0	832.4	60.9540
K_{160}	15714.3	1.0	9800.1	476.2643
K_{180}	17724.2	1.0	5152.7	636.5428
K_{200}	19733.1	1.0	2356.8	375.2570

6 むすび

今後, 上と異なる分枝ルールとの比較などを含め, さらに本格的な数値実験を行う予定である.

参考文献

[1] M.R. Garey and D.S. Johnson, *Computers and Intractability*, Freeman, 1979.