数理解析研究所講究録
1526巻 2006年 78-86

非線形最小木問題に対するタブー探索法に基づく近似解法

広島大学大学院・工学研究科　片桐 英樹 (Hideki KATAGIRI)
坂和 正敏 (Masatoshi SAKAWA)
加藤浩介 (Kosuke KATO)
宇野剛史 (Takeshi UNO)
岸本勉 (Tsutomu KISHIMOTO)
Graduate School of Engineering
Hiroshima University

1 はじめに

ノードとアークから構成されるグラフにおいて，全てのノードを含む木を極大木という．アークに重みが付随したグラフにおいて，重みを総和が最小となる極大木を求める問題は最小木問題と呼ばれ，多項式時間アルゴリズムで効率的に解くことができることが知られている．最小木問題は情報ネットワークの構築や施設配置における費用最小化問題など現実の意思決定問題でしばしば見受けられる．

古典的な最小木問題ではアークに付加された重みは確定値であり，これは現実問題においてはアークの建設費用が確定値であるような状況に対応する．しかし，現実の意思決定問題においては，アークの重みが不確実性を含んでおり，確率変数で与えられるような場合が存在する．このような不確実性下において，意思決定者がリスクを考慮して重みの総和の分散を最小化する問題を考える場合，等価な確定問題に変換すると一般に目的関数は非線形となり，古典的な最小木問題の枠組みでは扱うことができない．

Zhou ら [3] は目的関数が 2 次関数である場合の最小木問題に対して遺伝的アルゴリズムに基づく解法を提案した．彼らの解法は目的関数が 2 次関数だけでなく一般の非線形関数でも適用できる．一方，近年，タブー探索 [1] に基づく解法が様々な問題に対して提案され，多くの問題で進化的計算手法などのメタヒューリスティックと比較して優れた結果を示している．Hanafi ら [2] は多次元 0-1 ナップサック問題に対して，戦略的振動を用いたタブー探索に基づく解法を提案した．本研究では，Hanafi らが提案したアルゴリズムに基づいて，非線形最小木問題に対してタブー探索に基づく解法を提案する．さらに数値実験により従来法との比較を行い，提案手法の有効性を検証する．

2 定式化

有限の要素を含むノード集合 $V = \{v_1, v_2, \ldots, v_n\}$ とアーク集合 $E = \{e_1, e_2, \ldots, e_m\}$ からなるグラフ $G = (V, E)$ において，$\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ を次のように定義する．

$$x_i = \begin{cases} 1 & \text{アーク } e_i \text{ が選択されている} \\ 0 & \text{それ以外} \end{cases}$$

本研究で扱う非線形最小木問題は，次のように定式化される．

$$\left.\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) \\ \text{subject to} & A\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in X \end{array}\right\} \quad (1)$$

ここで，$\boldsymbol{x} = (x_1, \ldots, x_n)^t$ は0-1決定変数ベクトル，$f(\boldsymbol{x})$ は $\boldsymbol{x}$ を決定変数とする非線形関数とする．また，A は $m \times n$ 係数行列，$\boldsymbol{b}$ は n 次元列ベクトルであり，制約式が線形であることを表している．X は与えられたグラフにおける極大木（全域木）全体の集合を表す．制約がなく目的関数が線形である場合は通常の最小木問題となり，Prim 法や Kruskal 法をはじめとした種々の改良版の多項式時間アルゴリズムによって，厳密に解けることが示されている．しかし，目的関数が非線形である一般の問題は，NP-hard であることが証明されているため，実用時間内で近似最適解を求める手法を開発することが重要となる．Zhou らは遺伝的アルゴリズムに基づく解法を提案しているが，本研究では，より高精度で高速な解法の開発を目的として，タブー探索に基づく解法を提案する．

3 タブー探索に基づく非線形最小木問題に対する解法

現在の探索解に対応する極大木を T^{cur}，探索中に見つかった最良解を T^{gb}，実行可能領域内に存在する(制約式を満たす) 全ての極大木からなる集合を $\mathcal{T}_{possible}$ とする．T^{cur} からアークを一本削除することで生成される二つの連結成分 T_{part1} と T_{part2} に対して，直近の削除アーク以外のアークの追加で生成される新たな極大木 T^{new} 全体の集合を $\mathcal{T}_{All}^{cur}$ とする．

遷移において追加あるいは削除したアーク全てをタブーリストに保存し，それぞれ一定期間削除あるいは追加を禁止する．ただし，「最良解を更新する」場合には禁止を解除するものとする．本研究で提案する手法のアルゴリズムは次のようになる．

TS for nonlinear minimum spanning tree problem

InitializeParameter($nic_{int}, nic_{osci}, nic_{diver}, nic_{depth}, nic_{all}, tl$)
InitializeFrequency(Freq[ ])
InitializeTabuList($TabuList$)
T^{cur}:=GenerateInitialSolution()
UpdateFrequency($Freq$[ ], T^{cur})
while $T^{cur} \notin \mathcal{T}_{possible}$ **do**
  T^{cur} :=GoToFeasibleArea($T^{cur}, TabuList$)
  UpdateFrequency($Freq$[ ], T^{cur})
end while
$T^{gb} := T^{cur}$
$count_{all} := 0$
while $count_{all} < nic_{all}$ **do**
  $count_{osci} := 0$
  while $count_{osci} < nic_{osci}$ **do**
    $count_{int} := 0$
    while $count_{int} < nic_{int}$ **do**
      T^{cur} :=BasicLocalSearch($T^{gb}, T^{cur}, TabuList$)
      UpdateFrequency($Freq$[ ], T^{cur})
      Update(T^{gb}, T^{cur})
      $count_{int} := count_{int} + 1$

```
        end while
        count_depth := 0
        while count_depth < nic_depth do
            T^cur :=ObjectiveLocalSearch(T^cur, TabuList)
            UpdateFrequency(Freq[ ], T^cur)
            if T^cur ∈ T_possible then
                Update(T^gb, T^cur)
            end if
            count_depth := count_depth + 1
        end while
        while T^cur ∉ T_possible do
            T^cur :=GoToFeasibleArea(T^cur, TabuList)
        end while
        Update(T^gb, T^cur)
        count_osci := count_osci + 1
    end while
    if count_all < nic_all − 1 then
        count_diver := 0
        while count_diver < nic_diver do
            T^cur :=Diversification(Freq[ ], T^cur, TabuList)
            UpdateFrequency(Freq[ ], T^cur)
            count_diver := count_diver + 1
        end while
        while T^cur ∉ T_possible do
            T^cur :=GoToFeasibleArea(T^cur, TabuList)
        end while
        Update(T^gb, T^cur)
    end if
    count_all := count_all + 1
end while
output: T^gb
```

以下では，アルゴリズムで用いられている関数について説明を行う．

3.1 初期設定

InitializeParameter では，局所的探索の繰り返し回数，振動戦略の回数，多様化戦略の回数，振動戦略の深さ，タブー期間をそれぞれ，nic_{int}, nic_{osci}, nic_{diver}, nic_{depth}, nic_{all}, tl で表し，あらかじめ正定数を与えるものとする．また，*InitializeFrequency*, *InitializeTabuList* ではそれぞれアークが探索解に含まれた回数およびタブーリストの初期化を行っている．

3.2 初期解の生成

GenerateInitialSolution() としては，2 つの方法を用いる．1 つ目の方法は，比較的実行可能領域に近い解を生成することを目的として，制約式の係数を重みとした場合において Prim 法を適用する手法である．

具体的には，添字 $i \in \{1,\ldots,m\}$ の制約式をもとに生成された初期解を T_i^{first} とする．ここで，極大木 T^j に対応するベクトル $\boldsymbol{x}^j$ に対して，$I^+ = \{\, i \mid \boldsymbol{a}_i\boldsymbol{x}^j > b_i\ ,\ i \in \{1,\ldots,m\}\}$ としたとき，T_j に対する各制約式の超過分の総和 $\delta(T^j)$ を

$$\begin{array}{ll} \textbf{if } I^+ \not\in \phi \quad \textbf{then} & \delta(T^j) := \displaystyle\sum_{i \in I^+} (\boldsymbol{a}_i\boldsymbol{x}^j - b_i) \\ \textbf{else} & \delta(T^j) := 0 \end{array} \tag{2}$$

と定義すると，Prim 法を用いた初期解の生成では，T_i^{first} の集合の中から，j :=argmin$\{\delta(T_j^{first}) \mid j \in \{1,\ldots,m\}\}$ となる T_j^{first} を選択する．ただし，複数の T_j^{first} を選択可能な場合には，最良の目的関数値をもつ解を初期解 T^{first} とする．

2つ目の方法では，解集合をランダムに生成する．集合内の全ての実行不可能解を実行可能領域へ移行し，得られた実行可能解集合の中で目的関数値が最良となる解を初期解とする．

第4節で行う数値実験において，1つ目の初期解生成法を組み込んだタブー探索を TS-P, 2つ目の方法を組み込んだタブー探索を TS-R とする．

3.3 局所的探索

$\mathcal{T}_{All}^{cur}$ に含まれ，かつ $\mathcal{T}_{possible}$ に含まれる (制約式を満たす) 極大木を $T_{possible}^{cur}$ としたとき，全ての $T_{possible}^{cur}$ からなる集合を T^{cur} における近傍 $NH_{Local}(T^{cur})$ とする．局所探索を行う BasicLocalSearch 関数では，現在の解 T^{cur} が制約式を満足した状態で探索を進める．現在の解である極大木 T^{cur} における近傍 $NH_{Local}(T^{cur})$ 内で，タブーでなく，かつ最も目的関数値を改善する（改悪しない）解，あるいは近傍内で最も目的関数値を改善し，かつ願望水準を満たす解を選択し，新たな探索解とする．具体的なアルゴリズムは次のようになる．

BasicLocalSearch($T^{gb}, T^{cur}, TabuList$)
  $NH := NH_{Local}(T^{cur})$, $T^{NonTabu} := NULL$, $T^{Tabu} := NULL$
  while $NH \neq \phi$ **do**
    Choose $T \in NH$
    if T not Tabu **then**
      if $T^{NonTabu} \neq NULL$ **then**
        if $f(T^{NonTabu}) > f(T)$ **then**
          $f(T^{NonTabu}) := f(T)$
        end if
      else
        $f(T^{NonTabu}) := f(T)$
      end if
    else
      if $f(T^{gb}) > f(T)$ **then**
        if $T^{NonTabu} \neq NULL$ **then**
          if $f(T^{NonTabu}) > f(T)$ **then**
            $f(T^{NonTabu}) := f(T)$
          end if
        end if
      else
        if $T^{Tabu} \neq NULL$ **then**
          if $f(T^{Tabu}) > f(T)$ **then**

$f(T^{Tabu}) := f(T)$
end if
else
$f(T^{Tabu}) := f(T)$
end if
end if
$NH := NH/\{T\}$
end while
if $T^{NonTabu} \neq NULL$ then
$T^{new} := T^{NonTabu}$
else
$T^{new} := T^{Tabu}$
end if
Update TabuList
output:T^{new}

3.4 戦略的振動

戦略的振動は実行可能領域と実行不可能領域を行き来することで，より効率的に探索を進めていくことを目的としており，1) 目的関数値のみを考慮した探索，2) 制約違反度のみを考慮して探索の 2 つの手続きから構成されている．

3.4.1 目的関数値のみを考慮した探索

現在の極大木からアークを一本取り除くことにより，二つの連結成分へと分解する操作と，その二つの連結成分を一本のアークで結びつけることにより新たな極大木を構成する操作の二つを行うことで探索を進める．前者では，近傍を現在の探索点である極大木 T^{cur} から 1 つのアーク削除で生成される 2 つの連結成分 (T_{part1}, T_{part2}) 全体の集合を近傍 $NH1_{obj}(T^{cur})$ とする．後者では，連結成分 (T_{part1}, T_{part2}) に対して一つのアーク追加で生成される極大木全体の集合から T^{cur} を取り除いた集合を探索解 $(T^{cur}_{part1}, T^{cur}_{part2})$ における近傍 $NH2_{obj}(T^{cur}_{part1}, T^{cur}_{part2})$ とする．目的関数値のみを考慮した探索の具体的なアルゴリズムは次のようになる．

ObjectiveLocalSearch($T^{cur}, TabuList$)
$NH1 := NH1_{obj}(T^{cur})$, $(T_{part1}, T_{part2}) := (NULL, NULL)$
while $NH1 \neq \phi$ **then**
choose $(T_{part1}, T_{part2}) \in NH1$
if (T_{part1}, T_{part2}) not Tabu **then**
if $(T_{part1}, T_{part2}) \neq (NULL, NULL)$ **then**
if $f(T^{new}_{part1}, T^{new}_{part2}) > f(T_{part1}, T_{part2})$ **then**
$(T^{new}_{part1}, T^{new}_{part2}) := (T_{part1}, T_{part2})$
end if
else
$(T^{new}_{part1}, T^{new}_{part2}) := (T_{part1}, T_{part2})$
end if
end if

$NH1 := NH1/\{(T_{part1}, T_{part2})\}$
end while
Update TabuList
$NH2 := NH2_{obj}(T^{new}_{part1}, T^{new}_{part2})$, $T^{new} := NULL$
while $NH2 \neq \phi$ **then**
choose $T \in NH2$
if T not Tabu **then**
if $T^{new} \neq NULL$ **then**
if $f(T^{new}) > f(T^{now})$ **then**
$T^{new} := T$
end if
else
$T^{new} := T$
end if
end if
$NH2 := NH2/\{T\}$
end while
Update TabuList
output:T^{new}

3.4.2 制約違反度のみを考慮した探索

制約違反度のみを考慮した探索においては，T^{cur} における近傍を $NH_{con}(T^{cur}) = \mathcal{T}^{cur}_{All}$ とする．具体的なアルゴリズムは次のようになる．

GoToFeasibleArea(T^{cur},*TabuList*)
$NH := NH_{con}(T^{cur})$, $T^{new} := NULL$
while $NH \neq \phi$ **then**
choose $T \in NH$
if T not Tabu **then**
if $T^{new} \neq NULL$ **then**
if $\delta(T^{new}) > \delta(T)$ **then**
$T^{new} := T$
end if
else
$T^{new} := T$
end if
end if
$NH := NH/\{T\}$
end while
Update TabuList
output:T^{new}

3.5 多様化戦略

多様化戦略では，現在までに探索を進めた解領域から，未探索領域に移行して探索を進めることで，最良解を更新することを目的とする．現在の探索点 T^{cur} の近傍 $NH1_{diver}(T^{cur})$ の中で，探索解に含まれた頻度が最大のアークを削除することで生成される 2 つの連結成分を次の探索点 $(T^{new}_{part1}, T^{new}_{part2})$ とする．その後，近傍 $NH2_{diver}(T^{new}_{part1}, T^{new}_{part2})$ の中で頻度が最小のアークを加えることで生成される解を次の探索点とする．具体的なアルゴリズムは次のようになる．

Diversification$(Freq[\,], T^{cur}, TabuList)$
$E_{del} := E(T^{cur})$
e_{del} :=argmax$\{Freq[e] \mid e \in E_{del}, e \notin TabuList\}$
$(T^{new}_{part1}, T^{new}_{part2}) := T^{cur} - e_{del}$
$E_{add} := \{e = (v, v') \in E(G) \mid v \in V(T^{new}_{part1}), v' \in V(T^{new}_{part2})\}/\{e_{del}\}$
e_{add} :=argmin$\{Freq[e] \mid e \in E_{add}, e \notin TabuList\}$
$T^{new} := (T^{new}_{part1}, T^{new}_{part2}) + e_{add}$
Update TabuList
output:T^{new}

3.6 その他の関数

UpdateFrequency$(Freq[\,], T^{cur})$ では，任意の $e \in T^{cur}$ に対して，$Freq[e] := Freq[e] + 1$ とする．また，**Update**(T^{gb}, T^{cur}) では，$f(T^{gb}) > f(T^{cur})$ ならば，$T^{gb} := T^{cur}$ とする．

4 数値実験

非線形最小木問題に対するタブー探索法を用いた提案手法の有用性を検証するため，GA との比較を行った．ここでは，ノード数 30 の完全グラフに対して，各手法ともに 30 回ずつ試行し，得られた最良解が与える目的関数値の中での最良値，平均値，最悪値，また，最良解が得られるまでにかかった平均計算時間を求めた．

1. **問題例 1**
アークに付随する重みがファジィランダム変数である場合の最小木問題において，次のような分散最小化モデルに基づいて得られた確定問題に対して実験を行う．

$$\left.\begin{array}{ll} \text{minimize} & \dfrac{1}{\left(\displaystyle\sum_{j=1}^{n} \beta_j x_j - g^1 + g^0\right)^2} x^t V x \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} \left\{ \sum_{s=1}^{S} p_s c_{js} + \delta\beta_j \right\} x_j \le (1-\delta)\, g^0 + \delta g^1, \quad x \in X \end{array}\right\} \tag{3}$$

ただし，$\beta_j, g^1, g^0, p_s, c_{js}, \delta$ は正定数であり，V は正定値行列である．この問題の目的関数は準凸関数，制約式は線形であり，問題としては比較的易しい問題であると言える．なお，プリム法により初期解を生成するタブー探索法を TS-P，ランダムに生成した解を初期解とするタブー探索法を TS-R，遺伝的アルゴリズムを用いた解法を GA とする．
TS-P，TS-R は GA に比べ，非常に短い計算時間で最良解を導出している．TS-R は解の精度において TS-P よりも良い結果を得ることができたが，計算時間の面において TS-P よりやや劣る結果となった．

表 1: 問題例 1 に対する実験結果

	TS-P	TS-R	GA
最良目的関数値	4.04×10^{-17}	2.98×10^{-17}	1.89×10^{-11}
平均目的関数値	7.30×10^{-12}	7.01×10^{-12}	5.11×10^{-10}
最悪目的関数値	4.64×10^{-11}	4.44×10^{-11}	1.96×10^{-9}
平均計算時間 (秒)	3.790	4.107	716.579

2. 問題例 2

次に，ネットワークの信頼性を最大化する問題 (4) に対して数値実験を行う．

$$\left.\begin{array}{ll} \text{maximize} & \displaystyle\prod_{i=1}^{n}(r_i + \sum_{j=1}^{n} c_{ij}x_j)^{x_i} \\ \text{subject to} & Ax \leq b,\ \ x \in X \end{array}\right\} \tag{4}$$

ただし r_i, c_{ij} は定数であり，n は全アーク数を表す．

表 2: 問題例 2 に対する実験結果

	TS-P	TS-R	GA
最良目的関数値	5.59×10^{-1}	5.61×10^{-1}	4.55×10^{-1}
平均目的関数値	5.59×10^{-1}	5.59×10^{-1}	3.95×10^{-1}
最悪目的関数値	5.59×10^{-1}	5.59×10^{-1}	3.20×10^{-1}
平均計算時間 (秒)	6.273	6.311	96.9325

すべてのノードにおいて TS-P，TS-R は GA に比べ，非常に短い計算時間で最良解を導出している．TS-P，TS-R は非常に安定した解を得ることができたが，TS-R のほうが最良値で優れた結果を示した．

3. 問題例 3

多峰性を持つ問題として，次の問題 (5) を解く．

$$\left.\begin{array}{ll} \text{minimize} & \displaystyle 10n_1 + \sum_{i=1}^{n_1}\{y_i^2 - 10\cos(2\pi y_i)\} \\ \text{subject to} & \displaystyle y_i = 5.14 \times \frac{\sum_{j=1}^{i}(-1)^j x_j}{n_2 - 1},\ \ i = 1 \ldots n_1 \\ & Ax \leq b \\ & x \in X \end{array}\right\} \tag{5}$$

ただし，n_1 は全アーク数，n_2 は全ノード数を表す．

TS-P で得られた最良解は GA で得られた最良解を上回るが，TS-P で得られた最悪解は GA で得られた最悪解に劣る結果となった．しかし，TS-R に関しては，TS-P，GA と比較して，良い解を安定した解を得ることができた．

5 おわりに

本研究では，目的関数が非線形関数である最小木問題に対して，戦略的振動を組み込んだタブー探索に基づく近似解法を提案した．数値実験の結果は TS-R が最も優れていることを示唆しているが，今回は適用し

表 3: 問題例 3 に対する実験結果

	TS-P	TS-R	GA
最良目的関数値	83.68	83.68	412.80
平均目的関数値	617.42	97.43.68	528.14
最悪目的関数値	697.64	145.04	601.89
平均計算時間 (秒)	55.672	50.624	110.5253

た問題数が少ないため，さらなる実験が必要である．特に 3 つ目の多峰性をある問題に対しては，TS-R の計算時間が GA の 2 倍程度であり，他の 2 つの問題に比べて優位性は落ちている．したがって，今後は非線形性が強い他の問題やノード数がさらに増えた大規模問題に対して数値実験を行い，提案手法の有効性について検証する予定である．

参考文献

[1] F. Glover, M. Laguna, Tabu Search, Kluwer Academic Publishers, 1997.

[2] S. Hanafi and A. Freville, An efficient tabu search approach for the 0-1 multidimensional knapsack problem, European Journal of Operational Research, Vol. 106, pp. 659-675, 1998.

[3] G. Zhou and M. Gen, An efficient genetic algorithm approach to the quadratic minimum spanning tree problem, Computers & Operations Research, Vol. 25, pp. 229-237, 1998.