平成22年度　　卒業研究論文

電力網の経路割り当てにおける
分散制約最適化問題の探索空間削減の検討

指導教官
松尾 啓志 教授
津邑 公暁 准教授

名古屋工業大学 工学部 情報工学科
平成 19年度入学　19115001番

相木 広識

平成23年3月23日

目 次

第1章

はじめに

停電時、復旧に時間がかかる。時間を短縮する手法が研究されている。ハードウェアリソース、通信コストに制限がある環境で動作するアルゴリズムである必要がある。また、解に必ず到達する必要がある。

電力網において停電時に代替経路を再構築して復旧する問題は，分散環境における資源割り当て応用問題として研究されている．このような問題の解法は，スケーラビリティや耐故障性の観点から分散協調処理による解決の有効性が期待される．早急に解を得る必要があることから，通信を伴う反復処理の回数が少なく，必ず解に必ずを得る手法が必要である。

既存研究では、電力網の経路割り当て問題に対する厳密解法が研究されている。この解法には問題の複雑さに応じて探索空間が指数関数的に増大する問題がある。そのため、実環境に適合するスケーラビリティを得るために，問題の緩和が必要となる。

本研究の目的は、上述の探索空間の大きさを抑制するために、問題の複雑度の緩和する指標と手法を提案することである。

提案手法では、従来解法で用いられる問題に対する木の一部の辺を削除する。また、辺を削除することにより、解に到達しなくなる可能性があるため，実行不可能性を回避する手法についても検討する。これらにより、探索空間を削減しつつ近似解を得ることができる可能性が向上する。

本論文の構成は以下の通りである。第2章では従来研究として，分散制約最適化問題と疑似木，疑似木に基づく動的計画法，さらに電力網の経路割り当て問題への応用

とその解法について説明する。第 3 章では提案手法として，問題に対する疑似木の削除による問題の緩和と，そのためのアルゴリズムの詳細を説明する。第 4 章では提案手法の評価とそれに対する考察をする。第 5 章では本研究についてまとめる。

第2章

分散制約最適化問題

本章では、分散制約最適化問題 (DCOP:Distributed Constraint Optimization Problems) の基本的な形式化と電力網の経路割り当て問題をモデルとした形式化及び、それぞれの既存解法について説明する。

2.1　基本的な分散制約最適化問題

制約最適化問題 (COP) とは、ノードがもつ変数間に与えられた制約を全て満たしまた、制約の評価関数の評価値の総和を最適化する問題である。分散制約最適化問題は $< N, D, R >$ により定義される。

- $N = \{N_1, \ldots, N_n\}$:変数の集合
- $D = \{D_1, \ldots, D_n\}$:各変数が取りうる値域の集合
- $F = \{f_1, \ldots, f_m\}$:制約の集合。$f_i(N_j, N_k) : D_{i_j} \times D_{i_k} \rightarrow \Re$ は変数 N_j, N_k の値のそれぞれの組み合わせのときの評価値を表す。

また、分散制約最適化問題では各変数、制約は各ノードが保持する。各変数はそれを保持するノードの状態を表し、値はそれを保持するノードが設定する。各制約はそれに関連する変数により表される評価関数によりその値が決定され、また、関連する変数を保持するノードに保持される。全ての評価関数の和を最適化することを目標とする。本研究が対象とする電力網の経路割り当て問題では評価関数の和を最小化する。

各ノードは制約で結ばれた隣接ノードとメッセージ交換を行うことにより解を探索する。
説明のため、N_i を変数それ自身のこともしくは、それを保持するノードを表すことにする。

2.1.1 動的計画法に基づく解法 - DPOP アルゴリズム

分散制約最適化問題に対する最適解を求める厳密解法として DPOP[1] が提案されている。DPOP は問題に対する疑似木に基づいた動的計画法による計算を行う。問題点として、問題の複雑さに応じてメッセージサイズ、メモリ量の指数関数的増加が挙げられる。この問題点によりハードウェアに制限のある分散環境への適用は難しいと考えられる。

DPOP アルゴリズムは、以下の三つのフェーズで構成される。

1. 疑似木 (PseudoTree) の生成
2. UTIL の伝搬
3. 先祖の値の伝搬

以降の説明では、DPOP のかかわる要素を次のように表す。
ノード N_i において

G**:** 問題のグラフ

PT**:** 問題のグラフより生成される疑似木

p_i**:** N_i の親ノード

C_i**:** N_i の子ノードの集合

PP_i**:** N_i の疑似親ノードの集合。疑似親とは後退辺で結ばれるノードの上位側のノード

PC_i**:** N_i の疑似子ノードの集合。疑似子とは後退辺で結ばれるノードの下位側のノード

ST_i**:** 疑似木上で N_i を根とする部分木

$UTIL_i$: N_i がそれを定義する各変数 N_k の値の各組み合わせをキー、その時の利得の合計をバリューとするハイパーキューブ

次に，DPOP のアルゴリズムにおける各フェーズの処理の概要を説明する。

疑似木の生成

後の二つのフェーズで行うメッセージの交換の経路は、生成された疑似木 PT に基づいて行われる。

疑似木 PT は問題のグラフ G を深さ優先探索 DFS により探索することにより得られる生成木に基づいて作成される。グラフ $G = (V, E)$ から生成される疑似木は、G と同じ頂点 V、辺 E により構成され、辺 E は木辺か後退辺に分類される。辺 E のうち DFS による生成木となる辺は木辺、それ以外は後退辺となる。疑似木は部分木間に依存がなため、後の二つのフェーズでは部分木間で並行して動作することができる。

UTIL の伝搬

UTIL の伝搬は葉ノードから始め、上位側のノードへ木辺を介して伝搬される。ノード N_i が P_i へ送信する $UTIL_i$ を定義する変数の集合 $dimension(UTIL_i)$ は、疑似木上で N_i を根とする部分木を ST_i とすると、

$$dimension(UTIL_i) = \{x \in N \cap \overline{ST_i} \mid P_i \cup PP_{ST_i}\}$$

すなわち、ノード N_i の親及び部分木 ST_i の各ノードが持つ疑似親ノードのうち ST_i に含まれない変数の集合を表す。

先祖の値の伝搬

根ノード N_r が全ての C_r から $UTIL_{C_i}$ を受け取った後、最適値の伝搬を開始し、全ての葉ノードまで到達したとき、アルゴリズムを終了する。ノード N_i はそれらから最適値を選択し、自身が保持する変数の値を選択した最適値を取る組み合わせにおける自身の値に設定する。親から伝搬された先祖の値に自身の値を追加し、C_i へそれらの値を伝搬する。

2.2 電力網の経路割り当て問題

従来研究では，分散制約最適化問題を電力網の経路割り当て問題に適用する手法が提案されている [2]。ここでは電力網の経路割り当て問題の形式化を示す。電力網は電力網に電力を供給する変電所、電力を消費する需要、電力を送信する配電線からなる。全ての需要に対して電力を供給され、送電経路は木構造をなし、配電線に流れる電力は配電線がもつ最大容量を違反しないような送電経路を求めることが目的である。このような問題の解は以下の条件を満たす。

非循環性 解により表される送電経路は木構造

フロー保存則 それぞれのノードはキルヒホッフの第一法則を守る

容量制約 それぞれの配電線に流れる電力量は、自身の容量を越えない

以降では、電力網の経路割り当て問題を形式化した分散制約最適化問題の各要素を示す。

2.2.1 変数

各ノードは供給元変数、負荷変数の二つを持つ。

供給元変数 自身へ電力を供給するノードを表す。値域は自身へ電力を供給する可能性のあるノードの集合

負荷変数 自身へ供給される電力量

2.2.2 制約

2.2 で述べた問題の解が満たすべき条件を満たすために以下の制約を導入する。

木制約 作成される電力経路は木構造である必要がある。そのための制約である。また、最適化の対象である送電損失についての評価を行う。

KCL 制約 キルヒホッフの第一法則すなわち、自身に供給される電力量は自身で消費する電力量と他のノードへ供給する電力量の総和に等しくなる。そのための制約である。

制約の評価値

2.2.3 電力網の経路割り当て問題への DPOP の適用

ここでは、DPOP アルゴリズムを電力網の経路割り当て問題における分散制約最適化問題に対する厳密解法に拡張したアルゴリズムについて説明する。特に，アルゴリズムにかかわる主要な変更点を示す。

グラフの構成要素

電力網におけるグラフは $G = (V, E)$ により表される。頂点 V は電力を供給する電力源と、電力を消費する需要の二つに分類される。辺 E は電力を伝達する配電線を表す。頂点と辺は，前述の各制約にかかわる情報が関連付けられるものとする。

容量制約の評価

新たに追加した配電線に流すことのできる最大の電力量に関する容量制約の評価は、生成された疑似木上に存在するそれぞれの閉路で最も上位側のノード N が評価を行う。各ノードは自身を含む閉路で消費する電力量がわからなければ配電線の容量制約を違反しているかどうかを評価することはできない。したがって、自身を含む閉路で消費する電力量が決定する N が容量制約の評価を行う。

UTIL メッセージ

2.2.3 で説明した理由により、容量制約を評価する生成された疑似木上に存在するそれぞれの閉路で最も上位のノードは、所属する閉路のノードがとりうる値のすべての組み合わせを知る必要がある。したがって、各ノード Ni が親へ送信する UTIL メッセージを定義する変数は Ni を含んでいる。

また、閉路で最も上位側のノードが送信するUTILメッセージは、自身を根とする部分木の最適値Opt_iおよび、消費する電力量$Power_i$を送信する。それを受信する親ノード P_iは自身と隣接するノード間の制約にOpt_iを加算し、また自身が消費する電力にP_iを加算する。これによりP_iより上位のノードはN_i以下の割り当てについて知る必要がなくなる。

最適値の伝搬

生成された疑似木上に存在するそれぞれの閉路で最も上位のノードは、自身が含まれる閉路の経路を決定する。すなわち、その閉路中のすべてのノードの値を決定する。その後、決定した値をもとに自身の値を変更する。また、決定した値をメッセージとして各ノードに伝搬する。メッセージを受け取ったノードはメッセージをもとに自身の値変更する。

問題点

上述の形式化にもとづくDPOPでは、グラフに存在する閉路に含まれるノード数の最大値により，動的計画法の表とメッセージのサイズが指数関数的に増加する。このような極端な記憶，メッセージのサイズは実際の環境においては現実的ではない。したがって，必要となる記憶の量を削減するための緩和手法が必要である。

第3章

提案手法

本研究では，メッセージサイズを抑制する手法、および解に到達しなくなることを抑制する手法を提案する。

3.1 辺の削除

2章で示した形式化に基づくDPOPでは，問題に対するグラフの複雑さに応じて動的計画法の表とメッセージのサイズが指数関数的に増大することが問題であった。

そこで動的計画法の表とメッセージのサイズを抑えるために、後退辺を削除して問題を緩和する手法を提案する。DPOPは疑似木にもとづくため，辺の削除においては，疑似木を維持することを考慮する必要がある。しかし，いずれの後退辺を削除しても，得られるグラフは疑似木となる。

さらに、後退辺の削除に加えて，木辺の削除を行う手法を提案する。後退辺の場合とは異なり，疑似木の構造を維持しつつ削除することができる木辺は，全体の一部である。また，木辺を削除すると疑似木の深さが小さくなる傾向がある。

これらの提案手法により、動的計画法の表とメッセージのサイズが減少する。そのトレードオフとして，得られる解の精度や、実行可能性の低下が起こりうる。

提案手法では、対象の木辺の削除を行った後の生成木が疑似木の構造を持っている必要がある。すなわち、削除した木辺の下位側のノード Ni を根とする部分木に含まれるノード Njが自身を含んでいない他の部分木に接続が存在してはならない。そのような接続は以下を満たす辺を削除した場合に存在する。削除する木辺の下位側のノード

Ni を根とする部分木に含まれるノード Nj が後退辺を介して接続しているノード集合 Np について考える。ノード Npn が後退辺を介して接続するノードの深さが、ノード Ni の親ノードの深さ以上かつ、ノード Ni の次の親ノード候補すなわち、もっとも下位側の疑似親ノードの深さ以下であることを満たす場合である。したがって、以上を満たす木辺は削除してはならない。

以上により、接続先の各疑似親の深さの最大値、すなわちもっとも下位側の疑似親の深さがノード N がもつ後退辺により接続する疑似親の深さの最大値に一致した場合にノード N の親側の木辺は削除可能である。

3.2 後退辺及び木辺を削除する手法

説明の為、以下のように定義する。

ノード Ni において、

$MaxDms$: 疑似木 PT に存在する閉路のうち、含まれるノード数がもっとも多い閉路のノード数

$MaxNodes$: アルゴリズムにあたえるパラメータ
値域は $\{MaxNodes|1,...,$グラフのノード数$\}$
アルゴリズム完了時の疑似木 PT^* の $MaxDms$ の上限
$MaxNodes$ はユーザが設定する値でアルゴリズム実行中は不変

D_i : 疑似木の根から N_i までの木辺を介したホップ数。すなわち、根からの深さ

MSD_i : N_i が直接接続するノードのうち、疑似木の最も上位のノードの深さの値 D_{P_i} 及び D_{PP_i} の最小値。

CSD_i : ST_i から N_i を除いたノード集合のそれぞれのノードが直接接続するノードのうち、疑似木のもっとも上位のノードの深さの値
ST_i から N_i を除いたノード集合を $N = \{n|n \subset ST_i$かつ $n \subset \overline{N_i}\}$ としたとき、D_{N_k} の最小値

$candidate$: $candidate[i][0], candidate[i][1]$ には i 番目の削除候補辺の両端に接続されるノード id を格納している。

$estimate$: i 番目の削除候補辺を削除した場合に生成される疑似木 PT' の $MaxDms'$ を格納する。

3.2.1 アルゴリズムの概要

ここでは提案手法を実現するアルゴリズムについて説明する。

先ず始めに、現在の疑似木 PT において各ノード N_i が疑似木中での位置を特定するために D, MSD, CSD を更新する。この更新については後述する。次に、PT に存在するそれぞれの閉路のノード数のうち、最大値を現在の $MaxDms'$ する。$MaxDms'$ が入力パラメータである $MaxNodes$ 以下であれば、アルゴリズムを終了する。

1. 現在の疑似木 PT において、各ノードの D, MSD, CSD を更新する。(詳細は後述)
2. PT における $MaxDms$ を計算する。(詳細は後述)
3. $MaxDms \leq MaxNodes$ であるならアルゴリズムを終了する。そうでないなら、３．へ進む。
4. i 番目の削除候補辺の両端のノード id を $candidate[i][0], candidate[i][1]$ に登録する。
5. i 番目の削除候補辺について以下を行う
   (a) ノード id が $candidate[i][0], candidate[i][1]$ である二つのノード間の後退辺を削除する。
   (b) 各ノードの D, MSD, CSD を更新する。
   (c) 現在の疑似木 PTi' の $MaxDms'$ を計算し、$estimate[i]$ にその値を登録する。
   (d) 削除した辺を復元する。
   (e) 各ノードの D, MSD, CSD を更新する。
6. $estimate[i]$ が最小となる i^* を選択する。
7. 後退辺で接続される二つのノード $candidate[i][0], candidate[i][1]$ での下位側のノードの親への木辺が削除可能である場合、いずれかを削除する。
   (木辺が削除可能であるかどうかの判定は後述)

8. １．から繰り返す。

3.2.2 各ノードにおける辺の削除のための指標の計算

各木辺を削除可能かどうか判定する前処理としてこの計算を行う。
疑似木の根から木辺に沿って探索を行う。

1. 探索されたノード N_i は、D_i および CSD_i を親から受け取った引数 D で、MSD_i を $D-1$ で初期化する。
2. D_{PP_i} の最小値を自身の MSD_i とする。
3. C_oi を探索する。
4. C_i からの返り値のうち最少のものを CSD_i とする。
5. MSD_i と CSD_i で最小値を P_i に返り値として渡す。

3.2.3 疑似木における辺の削除のための指標の計算

この指標は問題のグラフに対する探索空間の規模を表す。直感的には、問題のグラフに存在する最大の閉路に含まれるノード数である。動的計画法の表とメッセージサイズの最大値はこの指標に応じて変化する。提案手法ではこのノード数を抑制することで動的計画法の表とメッセージサイズを抑制する。

ノード SN_i: $MSD_i == D_i - 1 \quad \&\& \quad CSD_i == D_i$ を満たすノード N_i

1. 疑似木上でノード SN_i から SN_i を根とする部分木で探索を開始する。$check[i] = 1$ とする
2. 探索されたノード N_i は、$nodes = 0$ で初期化する。
3. $check[i] == -1$ もしくは、ノード SN_i で $check[i] == 0$ を満たす場合、５．へ飛ぶ。
4. $check[i] = 1, nodes+ = 1$
   $nodes+ = C_i$ の返り値
5. 返値として $nodes$ を親に渡す。

ノード SN_i : 自身を根とする部分木がそこに含まれていないノードへ後退辺で接続していない、すなわち、$D_i = CSD_i$ かつ $D_i - 1 = MSD_i$ であるノード

各ノード SN_i は疑似木のうち自身を根とする部分木に含まれるノードの数をカウントする。

この時、他のノード SN_i が部分木に含まれる場合、その部分木のノードをカウントしない。

各ノード SN_i がカウントしたノード数の最大値が現在の疑似木の $MaxDms_i$ とする。

3.2.4 木辺が削除可能であるかどうかの判定

各ノードにおける辺の削除のための指標の計算により設定された各値をもとに以下の計算を行う。

1. 木辺の下位側のノード N_r の D_{PP_r} のうち最大値を $myppd$ とする。
2. 木辺の下位側のノード N_r を根とする部分木を木辺に沿って探索する。そのとき、その根ノードの深さ D_r を引数として渡す。
   (a) 探索されたノード N_i は、C_i を探索する。その時、引数として受け取った D_r を引数として渡す。
   (b) PP_i で D_r 以下であるもののうち最大値を $young$ とする。
   (c) C_i からの返り値及び、$young$ のうち最大値を P_i に返り値として渡す。
3. N_r は２．の返り値 $cppd == myppd$ であれば、N_r の親側への木辺は削除可能である。

第4章

評価

ここでは、二つの実験で従来手法および従来手法に提案手法を適用した手法を実験し評価する。一つ目の実験では、緩和後に解が必ず存在する問題を用いて、メッセージサイズ及び得られる解の精度について評価する。二つ目の実験では、緩和後に解が存在しなくなる可能性のある問題を用いて、メッセージサイズ及び得られる解の精度、解への到達率について評価する。

4.1 緩和の自由度が高い問題における評価

この実験では、提案手法による問題の緩和後に解が必ず存在する問題すなわち、各配電線の容量制約が十分に大きくこの制約に関して違反が発生しない問題グラフを用いる。また、厳密解法である従来手法、非厳密解法である提案手法でパラメータ MaxNodes をふった場合それぞれにおいて問題を解く。評価はそれぞれの場合における、メッセージサイズ及び得られる解の精度について評価する。

問題グラフは以下の設定で行う。

- ノード数：１５
- 配電線数：２２
- 次数３以下の連結グラフ
- 配電線の送電損失：１～９の１刻み

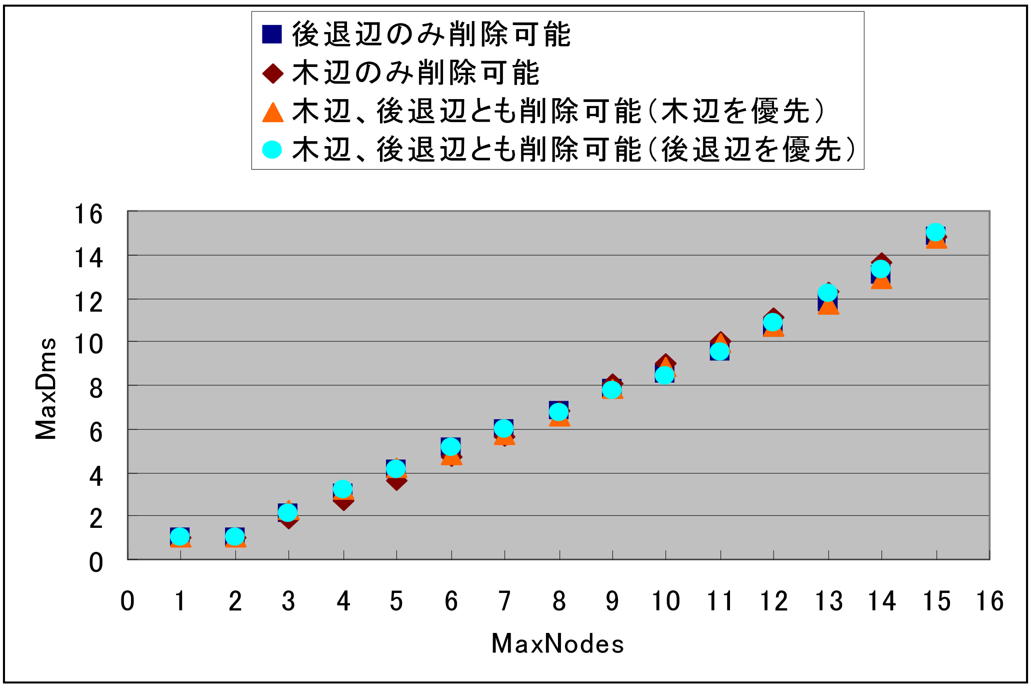

図 4.1: メッセージサイズの変化

- 配電線の容量制約：十分に大きい
- グラフの生成方法:辺が全く無い状態から、次数が 3 未満である任意の二つのノード間に制約をランダムに追加する。ランダムは一様分布乱数を用いる。
- グラフの数:グラフを 100 個用意した。
- それぞれのグラフに対して、パラメータ MaxNodes を 1 から 14 の間で 1 刻みでふり、プログラムを実行した。

4.1.1 解の精度とメッセージサイズ

パラメータ MaxNodes の値が大きいとき、メッセージサイズは減少し、解の精度は悪化している。小さいときはその逆となっている。つまり、パラメータ MaxNodes の値に応じて、メッセージサイズと解の精度はトレードオフとなっている。各辺の送電

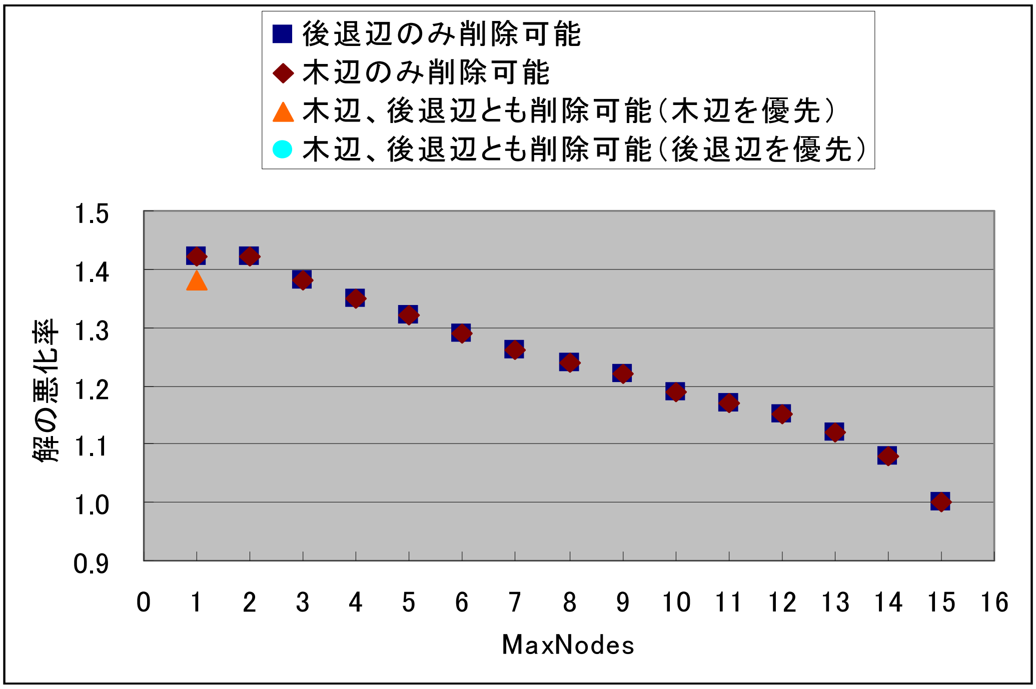

図 4.2: 解の悪化率の変化

損失は一様分布乱数により決定しているため、解の精度に大きな違いが表れなかったと考えられる。

4.1.2 辺を削除する戦略の比較

図 4.1、図 4.2 で示されるようにメッセージサイズ及び解の悪化率に関して、辺を削除する戦略間で大きな差はなかった。

4.2 緩和の自由度が低い問題における評価

この実験では、提案手法による問題の緩和後に解が存在しなくなる可能性のある問題すなわち、各配電線の容量制約の値は制限され、この制約に関して違反が発生する問題グラフを用いる。また、厳密解法である従来手法、非厳密解法である提案手法でパラメータ MaxNodes をふった場合それぞれにおいて問題を解く。評価はそれぞれの

場合における、メッセージサイズ及び得られる解の精度、解への到達率について評価する。

問題グラフは以下の設定で行う。

- ノード数：１５
- 配電線数：２２
- 次数 3 以下の連結グラフ
- 配電線の送電損失：１～９の１刻み
- 配電線の容量制約：１０
- グラフの生成方法:辺が全く無い状態から、次数が 3 未満である任意の二つのノード間に制約をランダムに追加する。ランダムは一様分布乱数を用いる。
- グラフの数:厳密解法で解が存在するグラフを 100 個用意した。
- それぞれのグラフに対して、パラメータ MaxNodes を 1 から 14 の間で 1 刻みでふり、プログラムを実行した。

4.2.1 実行可能性とメッセージサイズ

図 4.5、図 4.3 に示されるように、メッセージサイズの減少と共に解への到達率が低下している。また、その逆も成り立つ。したがって、メッセージサイズと解への到達率はトレードオフとなっている。

4.2.2 解の精度とメッセージサイズ

図 4.4、図 4.3 に示されるように、メッセージサイズの現象と共に解の精度が低下している。また、その逆も成り立つ。したがって、解の精度とメッセージサイズはトレードオフとなっている。

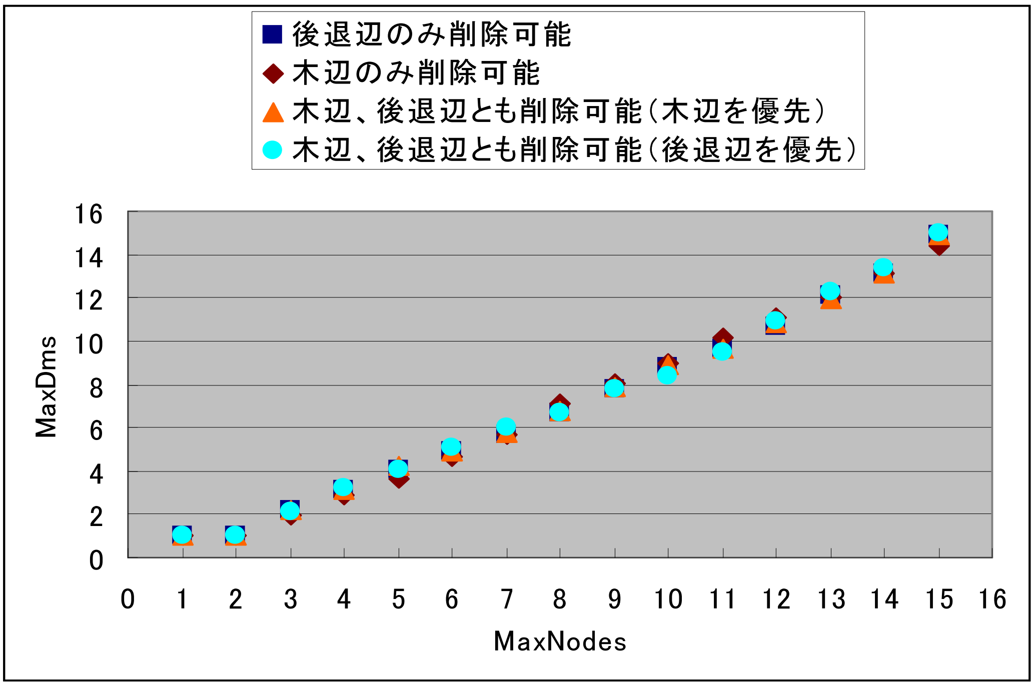

図 4.3: メッセージサイズの変化

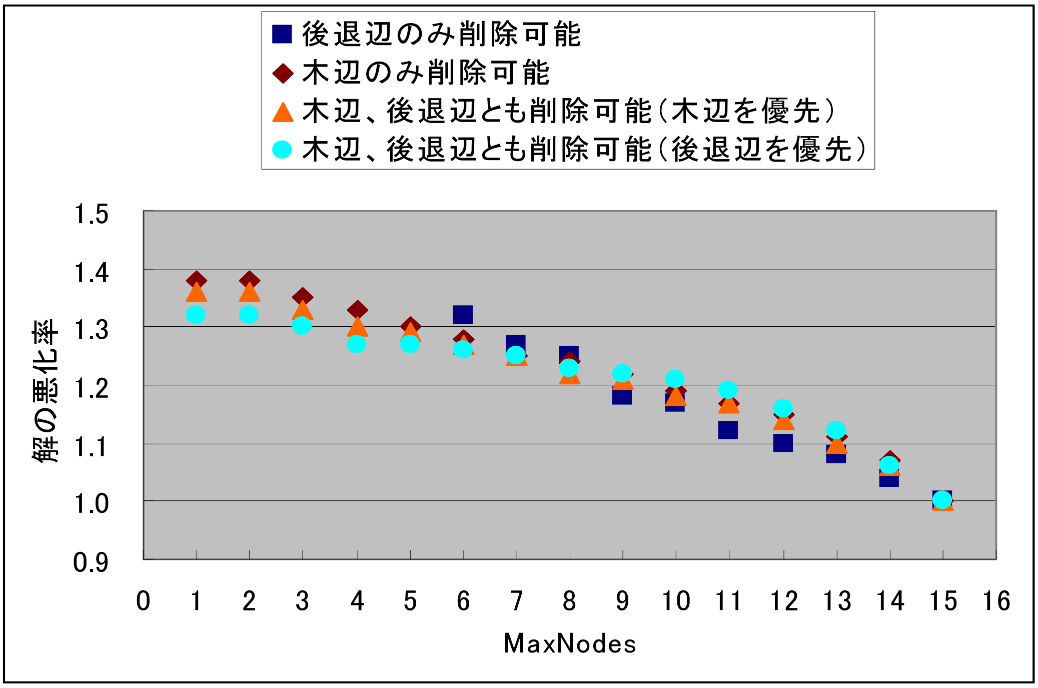

図 4.4: 解の悪化率の変化

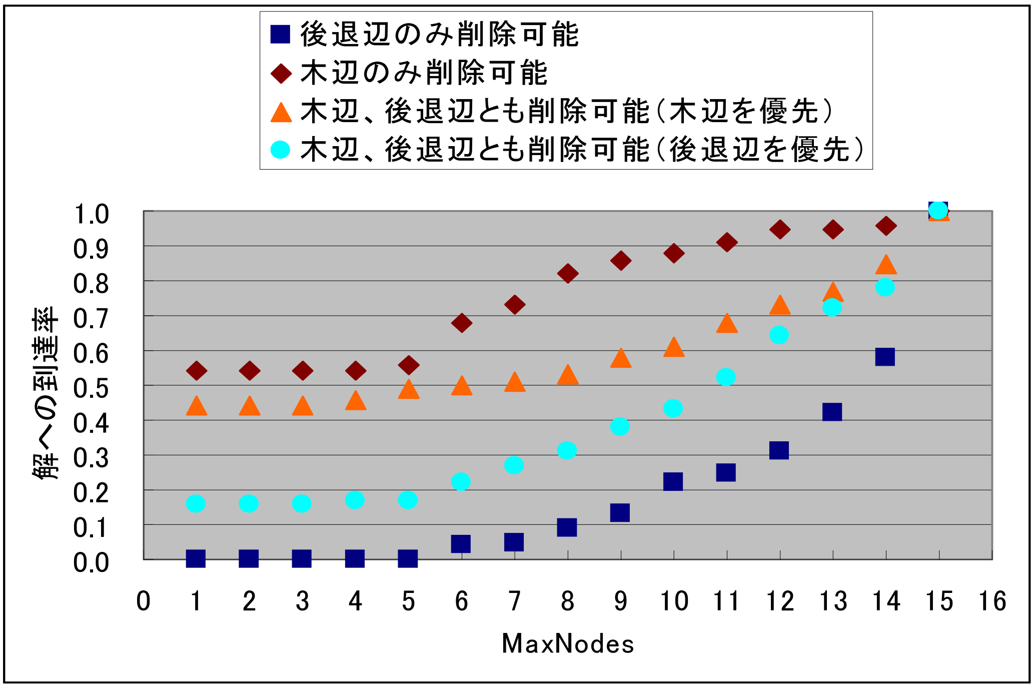

図 4.5: 解への到達率の変化

4.2.3 枝を削除する戦略比較

解に到達する割合について後退辺のみを削除した場合、他と比較して極端に悪化している。後退辺、木辺を削除した場合、後退辺のみ削除した場合よりは悪化しない。木辺のみを削除した場合、他と比較してもっとも解に到達する割合は高かった。

後退辺のみを削除する場合、はじめに生成される疑似木 PT の木の高さは削除後の疑似木 PT*と比較して変化しない。一方で木辺のみを削除する場合、例えばアルゴリズムの途中に生成される疑似木 PT' において深さ D が最大値をとる葉ノードが唯一であるとする。この葉ノードの親ノードへの木辺を削除した場合の疑似木 PT" は高さが減少する。高さが変化しない場合は減少した場合と比較して、疑似木の上位側の配電線にかかる電力負荷が増大すると考えられる。以上により、配電線の容量制約を違反することが多くなることで解に到達する割合が悪化したと考えられる。

MaxNodes を小さくした場合、解への到達率が低下している。後退辺のみを削除する場合は、木辺のみを削除する場合と比較して探索する経路を表す木は高くなる。す

なわち、上位の経路は相対的に大きな電力が流れることになる。したがって、後退辺みのを削除する場合は解への到達率が他と比較して大幅に低下していると考えられる。

第5章

まとめ

本研究では、電力網の経路割り当てのための分散制約最適化問題において，従来手法において問題となる動的計画法の表とメッセージのサイズを削減するための，問題の緩和手法を提案した。

探索空間の増大の原因である，問題に対する疑似木に含まれる閉路の大きさを，後退辺および一部の木辺の削除により制限する手法を提案した。さらに、削除する辺を選択する戦略による解への到達率の向上を提案した。また、それらを実験により評価し、探索空間の大きさと近似解の精度がトレードオフの関係を示した。削除する辺を選択する千尺により解への到達率に差が生じることを示した。以上により、探索空間を抑えつつ近似解を得ることに対する、提案手法の一定の有効性が示された。今後の課題としては、複数の辺を同時に削除することにより，効率的に探索空間を削減し、解品質と実行可能性の向上を検討することがあげられる。

謝辞

本研究のために多大なご尽力を頂き、日頃から熱心な御指導を賜った名古屋工業大学の松尾啓志教授、津邑公暁准教授、松井俊浩助教に深く感謝致します。

また、本研究の際に多くの助言をして頂いた松尾・津邑研究室ならびに齋藤研究室の皆様に深く感謝致します。

参考文献

[1] Petcu, A. and Faltings, B.: DPOP: A scalable Method for Multiagent Constraint Optimization, in *IJCAI 05*, pp. 266–271, Edinburgh, Scotland (2005).

[2] Kumar, A., Faltings, B. and Petcu, A.: Distributed constraint optimization with structured resource constraints, in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pp. 923–930, Richland, SC (2009), International Foundation for Autonomous Agents and Multiagent Systems.