3F3-6

分散資源割り当て問題の規模拡大のための異種解法の統合の検討

Study of structured solution methods for large scale distributed resource allocation problem

松井俊浩 松尾啓志
Toshihiro Matsui Hiroshi Matsuo

名古屋工業大学
Nagoya Institute of Technology

Distributed resource allocation is an important application of multiagent systems. Resource allocation problems that are motivated from a power supply network including distributed sources have been studied as typical examples of distributed optimization problems with resource constraints. When the problems are large and complex, exact solution methods are not applicable. Inexact methods that employ approximation are therefore introduced. On the other hand, to satisfy the resource constraints, exact methods are still necessary. As a base of such integrated solution method, we study the effects of search methods on the problems that are approximated using regional division.

1. はじめに

ネットワーク上に存在する共有資源の配分を決定する分散資源割り当ては，マルチエージェントシステムの重要な応用の一つである．センサ網や電力供給網などを含む，複数の資源割り当ての種類があるが，資源割り当ては基本的に制約最適化問題を含む．そのため，資源割り当ての解決の為に，分散最適化手法が必要となる．

マルチエージェントシステムにおける協調問題解決の基本的な枠組みとして，分散制約最適化問題 (DCOP) が研究されている [Mailler 04, Modi 05, Petcu 05]．DCOP の手法では，エージェントの状態とそれらの関係が，制約最適化問題として形式化され，その問題を分散探索アルゴリズムを用いて解く．これらの研究はマルチエージェントシステムの協調プロトコルに含まれる，最適化問題と分散探索アルゴリズムに注目している．より特殊な場合には，DCOP の表現は特定の問題に合致するように拡張され，解法もその問題に合わせて変更される．

ここでは，スマートグリッドの電力供給網資源に動機づけされる資源割り当て問題に注目する．この資源供給網では，初期の段階でソースノードに分散する資源が，全てのノードで共有される．関連研究 [Kumar 09] では，電力網における最適化問題 [Thiebaux 01] に特化した DCOP と解法が提案されている．この研究の目的は，資源制約のもとでフィーダツリー生成することである．また，フィーダツリー上での資源の配分を決定する問題と解法が提案されている [Miller 12, Matsui 12]．これらは，共有資源をエージェント間で分解可能な大域的制約として明示的に表現するように，特化されたクラスの問題である資源制約付き DCOP(RCDCOP) [Matsui 08] と関連する．

このような問題が複雑かつ大規模である場合には厳密解法の適用が困難であるため，非厳密解法の導入が必要となる．しかし，電力網などの資源制約は緩和が困難であることから部分的には厳密解法が望ましく，複数の異なる解法を統合する余地があると考えられる．本研究では，大規模複雑な分散最適化問題において，厳密解法と非厳密解法を統合する方法の基礎検討として，問題を階層的に分割する近似手法および，その枠組みの上に分散探索処理を導入をする方法を示す．

連絡先: 松井俊浩，名古屋工業大学，〒 466-8555 愛知県名古屋市昭和区御器所町，matsui.t@nitech.ac.jp

2. 背景

2.1 動機づけとする問題

分散電源を含むような電力供給網を動機づけとして，ネットワーク上の資源割り当て問題を定義する．このネットワークでは，複数のノード上に分散して存在する資源のある量が，他のノードに割り当てられる．閉路が含まれるネットワークを対象とする．ネットワークは次の要素からなる．

- ノード: 各ノードはある量の資源を供給するか消費する．資源を供給するノードをソースと呼ぶ．また，他のノードの資源を消費するノードをシンクと呼ぶ．供給または消費される資源の量には制限がある．ノードはある資源の量に選好を持つ．ここではシンクの消費の量は定数とする．また，ソースの供給量に比例するコストが生じる．

- リンク: 各リンクは 2 つのノードを接続する．リンクとノードは資源を移送する経路を構成する．あるリンクを経由して移送される資源の量には制限がある．移送の際に失われる資源の量は無視できるものとする．

資源の量の制限に加えて，供給，消費される資源の量の合計における制約がある．すなわち，供給，消費される資源の量の合計は零でなければならない．基本的には，問題の目的は，制約のもとで，全ノードの選好の値を大域的に最適化することである．

形式的には，問題は $\langle N, L, R, F, \mathcal{L}\rangle$ により定義される．ここで，N, L, R, F および $\mathcal{L}$ は，それぞれ，ノードの集合，リンクの集合，各ノードの資源の量の値域を表す集合からなる集合族，コスト関数の集合，各リンクの資源の量の値域を表す集合からなる集合族，である．

ノード $i \in N$ について，供給および消費される資源の量における選好が，次のように定義される．

- R_i: $R_i \in R$ は，ノード i により供給または消費される，資源の量の値からなる有限集合である．資源の量 $r \in R_i$ は負値であれば，r は供給される資源の量を表す．r が正値であれば，消費される資源の量を表す．ノード i はある資源の量の値を R_i の要素から選択する．

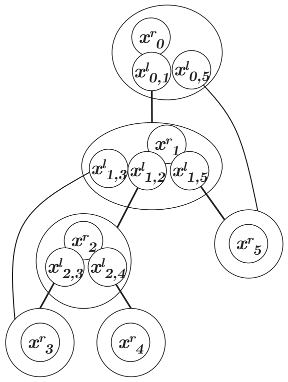

図 1: RCDCOP を表す疑似木

- $f_i(r)$: $f_i(r) \in F$ は，資源の量 $r \in R_i$ から非負の値への写像を表す，コスト関数である．ここでは解法を最小化問題に対して定義するため，ノードの選好を表すためにコスト関数を用いる．

各リンクはノードの組 (i, j) について定義される．リンク $(i, j) \in L$ について，移送される資源の量は次のように表される．

- $\mathcal{L}_{i,j}$: $\mathcal{L}_{i,j} \in \mathcal{L}$ はリンク (i, j) を経由して移送される資源の量の値からなる有限集合である．重の量 $l \in \mathcal{L}_{i,j}$ は，$-l^c_{i,j} \leq l \leq l^c_{i,j}$ なる離散値である．ここで，$l^c_{i,j}$ はリンク (i, j) の容量である．$\mathcal{L}_{i,j}$ に含まれる正負の値は対称である．値 l の符号は，移送の方向を表す．ネットワーク上であらかじめ定められたフローの向きに従って資源を移送するとき，対応するリンクにおける l は正値である．

各ノード $i \in N$ において，r_i および，i に接続する全てのリンク (i, j) についての $l_{i,j}$ の合計は零でなければならない．この制約は，ノード i に流入する向きのリンクの集合 L^{in}_i および流出する向きのリンクの集合 L^{out}_i を用いて $\sum_{(i,j) \in L^{in}_i} l_{i,j} = r_i + \sum_{(i,k) \in L^{out}_i} l_{i,k}$ のように表される．

全てのノードへの資源の割り当て $\mathcal{R}$ について，大域的なコスト $f(\mathcal{R})$ は $f(\mathcal{R}) = \sum_{i \in N} f_i(r_i)$ のように定義される．ここで，r_i は $\mathcal{R}$ における対応する値を取る．問題の目的は，制約のもとで，$f(\mathcal{R})$ を最小化するような，最適な割り当て $\mathcal{R}^*$ を求めることである．

2.2 資源制約付き DCOP としての形式化

DCOP はマルチエージェントシステムにおける協調問題解決の基本的な枠組みとして研究されている．DCOP により，マルチエージェントシステムにおける最適化問題が，制約最適化問題として表現される．問題を構成する，変数，制約および評価関数は，各エージェントに分散して配置される．問題は，メッセージ交換を伴う．分散協調アルゴリズムにより解決される．

RCDCOP [Matsui 08] は拡張された DCOP であり，資源および資源にかかわる制約のための表現を含む．2.1 節で前述した資源割り当て問題は，一種の RCDCOP として，変数，制約，および関数を用いて表すことができる [Miller 12, Matsui 12]．この RCDCOP は $\langle A, X^r, X^l, D^r, D^l, F, C \rangle$ により定義される．ここで，A はエージェントの集合を表す．X^r はノードにおいて供給または消費される資源の量を表す変数の集合である．X^l はリンクにおいて移送される資源の量を表す変数の集合である．D^r と D^l は，それぞれ X^r と X^l に含まれる変数の値域を表す集合からなる，集合族である．F はコスト関数の集合であり，C は資源制約の集合である．

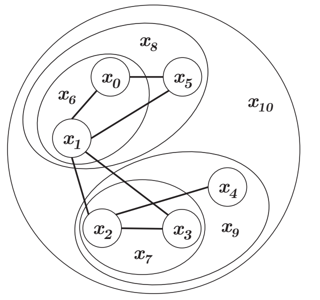

図 2: 階層的な分割

RCDCOP における各エージェント $i \in A$ は，資源割り当て問題における 1 つのノードに対応する．

以下では簡単のため，必要に応じてエージェントとノードを区別せずに用いる．また，あるノードを根とする疑似木 [Petcu 05] に基づいて，エージェントの集合における半順序関係，すなわち変数の順序が定義される．疑似木と対応する変数の順序に基づいて，各エージェント i について親エージェント p_i，子エージェントの集合 Ch_i，上位の近傍ノードの集合 (疑似親)PP_i，子孫の近傍ノードの集合 CC_i がそれぞれ定義される．

エージェント i は，供給または消費される資源の量を表す，変数 $x^r_i \in X^r$ を持つ．また，i は，i を経由して移送される資源の量を表す，変数の集合 $X^l_i \subset X^l$ を持つ．$x^l_{i,j} \in X^l_i$ は，エージェント i から子または子孫の近傍エージェント j に移送される資源の量を表す．同様に，$x^l_{k,i} \in X^l_k$ は，i の親または疑似親エージェント k からエージェント i に移送される資源の量を表す．エージェント i は，X^l_i に含まれる各変数の値を決定する．

$D^r_i \in D^r$ は，エージェント i の変数 x^r_i の値域を定義する．D^r_i は r_i が取りうる値からなる．$D^l_{i,j} \in D^l$ はリンク (i, j) の変数 $x^l_{i,j}$ の値域を定義する．$D^l_{i,j}$ は $\mathcal{L}_{i,j}$ の取りうる値からなる．

$f_i(x^r_i) \in F$ は，資源割り当て問題のノード i におけるコスト関数 $f_i(r_i)$ に対応する．同様に，$c_i \in C$ はノード i における資源制約に対応する．ここで，資源の流出入の向きは疑似木の上位から下位への向きに従う．全ての変数 $X^r \cup X^l$ についての割り当て $\mathcal{X}$ に関する大域的コスト関数 $f(\mathcal{X}) = \sum_{i \in N} f_i(x^r_i)$ を制約のもとで最小化する，割り当て $\mathcal{X}^*$ が最適解である．

RCDCOP を表す疑似木の例を図 1 に示す．疑似木はネットワーク上の生成木の辺に対応する木辺と，木辺以外のネットワーク上の辺である後退辺からなる．各ノード i は，子および下位の近傍ノード $j \in CC_i$ に資源の量を割り当てる．

この問題には，木探索と部分的な動的計画法を用いる DCOP の厳密解法を適用できる．例えば，文献 [Matsui 08] で示された解法を改変して適用できる．しかし，一般に，疑似木に基づく解法は，後退辺の数が多い場合には効率的ではない．さらに，この RCDCOP の形式化では，各ノードが持つ部分問題の規模が次数に応じて指数関数的に増加するため，ノードの次数とリンクの容量が大きい場合には厳密解法の適用は困難である．

3. 階層化を用いる非厳密解法と木探索の統合

3.1 階層的な問題の分割

本研究では，上記のような資源割り当て問題を近似する方法として，問題を階層的に領域に分割する手法を適用する．ネットワークを複数の領域に分割し，それらの間にあるリンクを単一のリンクに近似して扱う．このような手法は実行可能性を減少させるが，分割された領域内に十分な経路があれば問題の規模を削減する効果を活用できると考えられる．問題の分割を容

易にするために，領域を 2 分割して階層化する．分割された領域の構造は二分木により表現される．図 2 に階層的な領域の分割の例を示す．階層構造は，互いに近傍である 2 ノードをクラスタリングする操作の反復により，ボトムアップに構築される．ここでは，ノード数が少ないクラスタを優先して統合する．

2 ノード i, j を統合して新たなノード k を生成する際に，i, j 間の辺は削除される．i, j に接続するその他のリンクに対応して，k に接続するリンクが生成される．2 ノードから同一の他のノード o へのリンク (i, o),(j, o) ある場合は，これらを統合して辺 (k, o) が生成される．この時，辺 (k, o) の容量 $l^c_{k,o}$ は $l^c_{k,o} = l^c_{i,o} + l^c_{j,o}$ とする．各ノードから異なる他ノード p への辺 (i, p) がある場合には，この辺に直接対応する辺 (k, p) が生成される．この時，辺 (k, p) の容量は $l^c_{i,p}$ と等しいものとする．

さらに，2 ノード i, j の評価関数が統合され，新たなノード k の評価関数が生成される．生成される評価関数の値は，元の評価関数の合計値である．同一の合計値についての評価値は最小値を用いる．ただし，統合される 2 ノード間のリンクおよび，それ以外の近傍へのリンクの容量の制約を満たさなければならない．制約を満足しない場合の評価値は上限値 ∞ とする．統合前の i に接続する辺のうち，(i, j) を除く辺の容量の合計を l^c_i，j についての同様の合計を l^c_j，(i, j) の容量を $l^c_{i,j}$ とするとき，新たな評価関数 $f_k(r_k)$ は次のように計算される．

$$f_k(r_k) = \min_{r_i + r_j = r_k} f_{i,j} \tag{1}$$

$$f_{i,j} = \begin{cases} f_i(r_i) + f_j(r_j) & c_{i,j} \\ \infty & \text{otherwise} \end{cases} \tag{2}$$

$$\begin{aligned} c_{i,j}: \quad & \exists l, (-l^c_{i,j} \le l \le l^c_{i,j}) \land \\ & (-l^c_i \le r_i - l \le -l^c_i) \land (-l^c_j \le r_j + l \le -l^c_j) \end{aligned} \tag{3}$$

上記のような統合を，未統合の近傍ノードが無くなるまで反復する．

3.2 階層構造に基づく近似解法

階層構造に基づいて，資源の割り当てをトップダウンに計算する．この過程では，上位の階層のノード k は，自身に統合された下位の階層の 2 ノード i, j への資源を割り当て $\tilde{r}_i, \tilde{r}_j$ を決定する．

ノード k が最上位のノードのとき，$\tilde{r}_k$=0 である．$\tilde{r}_k$ による制約条件 $\tilde{r}_k$=$\tilde{r}_i + \tilde{r}_j$ が満足されなければならない．$\tilde{r}_i, \tilde{r}_j$ とともに，リンク (i, j) を経由して i が授受する資源の量 l_i と，j が授受する資源の量 l_j が決定される．ここで $l_i = -l_j$ である．

ノード k が最上位以外の場合，自身の近傍に接続する辺がある場合がある．このような辺は 2 種類に分類される．一方は，(a) ノード k とともに上位ノードで統合された近傍ノードへの辺である．他法は，(b) 自身が含まれる上位階層のノードが統合された時，その上位ノードの近傍ノードへの辺と対応していた辺である．(a), (b) ともに，その辺を経由して授受される資源の量は上位ノードにより決定される．特に (b) の場合は，上位では統合されていた辺を経由するの資源の量が，トップダウンな処理の過程で複数の辺に分解された場合がある．

ノード k は自身の近傍への辺 (k, p) を，統合された下位の階層の 2 ノード i, j それぞれへの辺に対応付ける．自身の近傍への辺を経由する資源の量が上位ノードによって既に割り当てられているから，i, j が授受する資源の量を決定できる．ここで，(k, p) が 2 つの辺 $(i, p), (j, p)$ に対応するとき，資源の量を分配する必要がある．ここでは，楽観的な方法として，容量 $l^c_{i,p}$ と $l^c_{j,p}$ の比に比例して分配を決定するものとする．決定された配分はノード i, j に通知され，さらに下位の分配が決定される．この方法は解空間を減少させるため，実行可能解が失われる場合がある．

ノード k は，自身の近傍への辺を i, j の辺に分解したのち，これらの辺と 2 ノード i, j 間の辺を考慮し，評価関数 $f_k(r_k)$ の値を最大化するような $\tilde{r}_i, \tilde{r}_j, e_i, e_j$ を決定する．ただし，制約条件 $\tilde{r}_k$=$\tilde{r}_i + \tilde{r}_j$ とともに，式 (3) と同様の制約条件を満足しなければならない．このとき，同式の l^c_i, l^c_j による境界の代わりに，既に上位で配分された資源の量を用いる．

3.3 探索の導入

3.2 節で述べた解法は単なる貪欲的手法であるため，実行不可能な局所解に陥る場合がある．そこで，木探索を導入する．あるノードが制約条件を満足できない場合，そのノードは充足不能であることを上位ノードに通知する．これによりバックトラックが発生し，他の割り当てが決定される．

従来解法では，疑似木に基づいて，木探索が実行される．この探索ではボトムアップに評価値が計算される．これに対して，本手法では，領域の分割のための階層構造を表す二分木において探索が実行される．また，評価値は既に集計されているため，充足可能性のみが探索される．

4. 解法の正しさと複雑度

本手法は，統合されたリンクそれぞれに資源の量を分配する方法が楽観的であり，非厳密解法である．したがって解の精度と実行可能性は元の問題よりも低下する場合がある．しかし，辺の密度が比較的高く，容量に余裕があれば，実行可能性は比較的高いと考えられる．この点は，辺の密度が高く木幅が大きい場合に反復回数が著しく増加する傾向がある，疑似木に基づく従来解法よりも優れる点と考えられる．

また，本手法では，ノードがクラスタリングされる過程で評価関数が統合され，その変域は単調に増大する．各ノードの評価関数の変域が同一であれば，変域の増加はノード数に関して線形であるが，この影響はノード数が多い場合に顕著である．しかし，解の精度および実行可能性とのトレードオフを許容すれば，評価関数値を適切な基準により間引くことができる．

従来手法では，ノード内の計算量にリンクの容量の次数乗が含まれ，次数について指数関数的である．従って，ノードの次数とリンクの容量が大きい場合の計算量は莫大となる．これに対し本手法は，リンクの容量は 2 つのリンクが統合された箇所で合計される．前処理と探索における評価値の計算量は，この容量の合計値について線形である．

5. 評価

提案手法を実験により評価した．問題に含まれるノードはソースまたはシンクに区別される．ここでは，問題をパラメータ $\langle n_p, n_l, n_e, l_c, r^l, r^u, p^l, p^u, c^l, c^u \rangle$ により設計した．n_p, n_l, n_e はそれぞれ，ソース，シンク，リンクの数である．l_c はリンクの容量である．全てのリンクは同一の容量とした．r^l と r^u はシンクが要求する資源の量の最小および最大値である．p^l と p^u はソースが供給する資源の量の上限の最小および最大値である．なお，下限値は 0 である．c^l と c^u はソースが供給する資源のコスト値の係数の最小および最大値である．各ノードのパラメータはそれぞれの区間から一様分布に従ってランダムに選択した．問題のパラメータは，

表 1: 解法が終了した問題の数と実行不可能な問題の数 ($< n_p, n_l, n_e >$)

problem	<10, 10, 19>		<10, 10, 21>		<10, 90, 99>		<50, 50, 250>	
algorithm	#termination	#infeasible	#termination	#infeasible	#termination	#infeasible	#termination	#infeasible
binclust	15	10	8	17	2	23	0	25
binclustsch50	18	7	20	5	9	16	24	1
binclustsch100	18	7	20	5	10	12	24	0
binclustsch	18	7	20	5	12	12	24	0
psdtreesch	18	7	21	4				
psdtreeschfirst	18	7	21	4				

表 2: 得られた解のコスト値 ($< n_p, n_l, n_e >$)

algorithm	<10, 10, 19>	<10, 10, 21>	<10, 90, 99>	<50, 50, 250>
binclustsch50	1.65	1.69	200	2.01
binclustsch100	1.65	1.69	182	1.92
binclustsch	1.65	1.69	182	1.95
psdtreesch	1.59	1.51		
psdtreeschfirst	2.12	2.3		

$\langle 10, 10, 19, 2, 1, 1, 1, 10, 1, 10\rangle$，$\langle 10, 10, 21, 2, 1, 1, 1, 10, 1, 10\rangle$，$\langle 10, 90, 99, 20, 1, 2, 20, 40, 1, 10\rangle$，$\langle 50, 50, 250, 5, 1, 2, 5, 10, 1, 10\rangle$ の 4 種類を用いた．以降では，紙幅の節約のために，$\langle n_p, n_l, n_e\rangle$ または，$\langle n_p, n_l\rangle$ により略記する．ネットワークは単一連結成分のランダムなグラフとした．問題には実行可能な問題と不可能な問題が含まれる．それぞれのパラメータの問題について 25 例の結果を平均した．

比較手法は次の通りである．

- binclust: 階層的な領域分割に基づく貪欲的解法．
- binclustsch, binclustsch50, binclustsch100: binclust の木構造を用いる探索を適用した解法．binclustsch50 および 100 では評価関数の変域を 50 および 100 に制限した．
- ptreeesch, ptreeeschfirst: 疑似木に基づく RCDCOP の解法．ptreeeschfirsth は最初に発見した最良解を用いる．

解法はいずれも局所的な最良優先の戦略を用いる．疑似木の生成および，階層的な分割は前処理とし，その後の割り当て処理をメッセージ交換に基づく分散アルゴリズムとしてシミュレーションした．実行の処理単位はメッセージサイクルとし，10000 回をカットオフサイクル数とした．

表 1 にカットオフサイクル数までに解を得て終了した問題の数と実行不可能となって終了した問題の数を示す．$\langle n_p, n_l\rangle = \langle 10, 10\rangle$ の問題では，binclustsch の結果が厳密解法である ptreeesch の結果とほぼ同様である．その一方で，探索を行わない binclust は他の解法よりも得られた解の数が明らかに少ない．より大きな規模の $\langle 10, 90\rangle, \langle 50, 50\rangle$ の問題では，ptreeesch の各ノードの計算量が大きく，現実的な時間では評価できなかった．これらの結果では，評価関数の変域をより制限した binclustsch50 では得られる解の数が少なかった．これらの問題で binclustsch の評価関数の変域の最大サイズはそれぞれ，298, 371 であった．特に辺の密度と容量が比較的大きい $\langle 50, 50\rangle$ のような場合はある程度までは実行可能性を大きく損なわずに評価関数の変域サイズを削減する余裕があると言える．

表 2 に得られた解のコスト値を示す．$\langle 10, 10\rangle$ の問題では，階層的な領域分割に基づく手法のコスト値は，厳密解法である ptreeesch の結果よりもやや大きい．その一方で，ptreeesch の最初の最良解を採用した場合よりはやや小さい評価値が得られた．$\langle 10, 90\rangle, \langle 50, 50\rangle$ の問題では，評価関数の変域サイズを制限した場合に，解空間が減少した影響によりコスト値が増加したが，その程度は比較的小さいと考えられる．

6. おわりに

本研究では，大規模複雑な分散最適化問題において，厳密解法と非厳密解法を統合する方法の基礎検討として，問題を階層的に分割する近似手法および，その枠組みの上に分散探索処理を導入をする方法を示した．本手法は，資源の配分における制約を緩和し，解の精度および実行可能性とのトレードオフとして計算量を抑制できる場合がある．実験により小規模な問題では従来解法と比較して比較的同程度の精度が得られることおよび，従来解法が適用困難な問題で解を得ることが示された．実行可能性についての解析および探索処理の効率化手法が今後の課題である．

参考文献

[Kumar 09] Kumar, A., Faltings, B., and Petcu, A.: Distributed constraint optimization with structured resource constraints, in *8th International Conference on Autonomous Agents and Multiagent Systems*, pp. 923–930 (2009)

[Mailler 04] Mailler, R. and Lesser, V.: Solving distributed constraint optimization problems using cooperative mediation, in *3rd International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 438–445 (2004)

[Matsui 08] Matsui, T., Silaghi, M., Hirayama, K., Yokoo, M., and Matsuo, H.: Resource Constrained Distributed Constraint Optimization with Virtual Variables, in *23rd AAAI Conference on Artificial Intelligence*, pp. 120–125 (2008)

[Matsui 12] Matsui, T. and Matsuo, H.: Considering Equality on Distributed Constraint Optimization Problem for Resource Supply Network, in *2012 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pp. 25–32 (2012)

[Miller 12] Miller, S., Ramchurn, S. D., and Rogers, A.: Optimal decentralised dispatch of embedded generation in the smart grid, in *11th International Conference on Autonomous Agents and Multiagent Systems*, Vol. 1, pp. 281–288 (2012)

[Modi 05] Modi, P. J., Shen, W., Tambe, M., and Yokoo, M.: Adopt: Asynchronous distributed constraint optimization with quality guarantees, *Artificial Intelligence*, Vol. 161, No. 1-2, pp. 149–180 (2005)

[Petcu 05] Petcu, A. and Faltings, B.: A Scalable Method for Multiagent Constraint Optimization, in *19th International Joint Conference on Artificial Intelligence*, pp. 266–271 (2005)

[Thiebaux 01] Thiebaux, S. and Cordier, odile M.: Supply restoration in power distribution systems - a benchmark for planning under uncertainty, in *6th European Conference on Planning (ECP-01*, pp. 525–532 (2001)