分散 AND/OR 探索による Automated Trust Negotiation

中塚康介† 石田亨†

ユビキタスネットワークにおいてサービス利用の可否を決定するための枠組みとして，Automated Trust Negotiation（ATN）が提案されている．ATN では，身元を証明するための証明書と，証明書を開示するための条件を記したポリシを，サービスと利用者が互いに交換しながら交渉を進める．証明書とポリシは機密情報を含むことから，不必要な情報をなるべく開示しないことが求められる．しかしながら，ATN の交渉は統一的な定式化がなされておらず，交渉手続きの評価が困難である．本稿では，(1) ATN を，コストをグラフに付与した AND/OR グラフの分散探索と定式化することにより，AND/OR グラフにおいて最小コストの解グラフを求める問題として定式化する．また，(2) 定式化に従うことで，コスト最小の解グラフを求める AO* アルゴリズムを適用し，証明書とポリシの開示リスクを低減させる．以上のように ATN の交渉手続きを AND/OR グラフの分散探索として理論的に記述し，評価することができるようになる．

Distributed AND/OR Search for Automated Trust Negotiation

Kosuke Nakatsuka† and Toru Ishida†

To control access to services in ubiquitous network environment, Automated Trust Negotiation (ATN) is proposed. In ATN, a service provider and a service user exchange credentials and policies to authorize service use. Credentials are for identification, and policies are for permission of disclosure of credentials. Because credentials and policies involve confidential information, service providers and users should not disclose needless one. In ATN, however, the negotiation protocols are developed individually and there is no common model. Therefore, it is difficult to evaluate their quality. In this paper, (1) we formalize ATN as distributed AND/OR graph search. Disclosure costs of credentials are defined as the cost of the solution graph. Disclosure of policies is mapped as node expansions. And (2) by using the formalization, we can decrease the risk of disclosure of credentials and policies by applying AO* algorithm that ensures minimum-cost solution graph to ATN. Applying our model to ATN, we can evaluate negotiation protocols as search algorithms.

1. はじめに

ユビキタスネットワーク環境上には多くのサービスと多くの利用者が存在する．サービス・利用者の両者にとって，未知の利用者に利用させる，あるいは，未知のサービスを利用することは，セキュリティ上の危険を招く．このため，サービス利用に先立ち，サービスや利用者が，互いに適切な権限を持つことを確認するために認証を行う必要がある．認証に必要な情報には，氏名や住所といった個人情報や，関与している団体などが推定できるような機密情報が含まれている．未知のサービス・利用者に機密情報を漏洩することを防ぐためには，できるだけ少ない情報で認証を行う必要がある．この問題に対して，認証で行われる手続きを自動化することを目的とした Automated Trust Negotiation（ATN）が提案され，ATN 上で認証に必要な情報を少なくする研究が行われている[1)~3)]．

ATN ではサーバとクライアントの 2 主体が存在する．サーバ・クライアントともに，身元を証明する証明書を複数持つ．証明書を通信相手に開示するために必要な条件をポリシと呼ぶ．ポリシは，ある証明書を開示するために相手に要求する証明書を列挙したものである．クライアントとサーバは交互にポリシか証明書を交換し，最終的にサービスの利用要件が満たされることを目標とする．しかし，証明書には個人情報などの機密情報が含まれる．また，“極秘に提携している企業であればサービスを利用できる” といったポリシのように，ポリシ自体にも機密情報が含まれている．したがって，ポリシ・証明書の開示はできる限り少なくする必要がある[2)]．この問題に対して，以下のようにポリシ・証明書の開示を少なくするような手法が提

† 京都大学大学院情報学研究科社会情報学専攻
Department of Social Informatics, Kyoto University

案されている．

- 証明書を開示する前にできるだけ多くのポリシを集め，証明書の開示を少なくする[1),2)]．
- 証明書のみを交換し，ポリシを開示しない[1),2)]．
- 各証明書を節点，開示ポリシを枝に対応させ，この木の上での探索を行う[3)]．

しかし，これらの手法は個別に提案され，交渉手法の比較や，最適性などの交渉手法の性質が明らかでない．

本研究では以下の問題を取り扱い，ATN の統一的な記述と評価を可能とする．

(1) 交渉手法の共通のモデルがなく，交渉手法間の性能比較や，交渉手法の性質が明らかでない．
(2) 各手法について，証明書やポリシの開示をどれだけ少なくするかの評価ができない．

本稿では，コスト付けされた AND/OR グラフ上で，ポリシの開示を節点の展開，証明書の開示に必要なコストを枝のコストと対応付ける．これにより，ATN を “2 エージェントが分散して AND/OR グラフのコスト最小の解グラフを求める探索” と統一的に記述可能にする．また，AND/OR グラフにおいて最小コストの解グラフを求める AO* アルゴリズム[4)] を，定式化に基づいた分散探索として記述する．本研究では以下の 2 点を達成する．

(1) ATN を AND/OR グラフの分散探索として定式化することにより，既存の AND/OR グラフ探索アルゴリズムが適用できる．また，探索アルゴリズムの観点から完全性や最適性を評価することができる．これにより，交渉が解を得て停止するか，証明書の開示コストを最小とするかなどの評価が可能となる．
(2) (1) の定式化に従い AO* を適用する．最適性を持つ AO* により証明書の開示を最小とすることができ，また，従来手法に対してポリシの開示数を減少させる．これにより，証明書の開示による情報漏洩のリスクを最小にし，ポリシの開示によるリスクをできるだけ少なくする．

2. Automated Trust Negotiation

ATN では，クライアントはサーバのサービス R を要求する．サーバは，サービス R を提供するための条件として，クライアントの身元を証明する証明書をクライアントに要求する．クライアントはサーバからの証明書要求に応えて証明書を渡す．しかし，証明書によってはサーバ側の証明書が必要なものがある．この場合，クライアントは証明書をサーバに要求する．サーバ・クライアントともにこのような交渉を繰り返し，サービス R の利用条件が満たされると交渉を終了する．ある証明書の要求に対して，その証明書を相手に提供するための条件をポリシと呼ぶ．証明書やポリシを相手のエージェントに渡すことを，その証明書・ポリシの開示と呼ぶ．

2.1 ポリシと証明書

本稿では，以下のようにポリシを定義する．

定義 1（ポリシ）

- ある証明書 C が，他の証明書の開示を要せずに開示することができる場合，このポリシを $C \leftarrow \mathrm{true}$ と表記する．
- ある証明書 C を開示するために，相手のエージェントの持つ証明書 $C_1, C_2, \ldots, C_k$ が開示されなければならない場合，このポリシを “$C \leftarrow C_1 \wedge C_2 \wedge \cdots \wedge C_k$” と表記する．

同じ証明書について，ポリシが複数ある場合，いずれか 1 つのポリシの条件が満たされていればよい．

たとえば $A \leftarrow B \wedge C$，$A \leftarrow D \wedge E$ というポリシがある場合，A を開示するためには，B と C の 2 つの証明書が開示されるか，あるいは，D と E の 2 つの証明書が開示されれば良い．

以下で定義される証明書を開示可能な証明書とする．

定義 2（開示可能な証明書）

- 無条件に開示することができる証明書は，開示可能な証明書である．
- ある証明書を開示するために必要な証明書がすべて開示可能であれば，この証明書は開示可能な証明書である．

表 1 にポリシの例を示す．表 1 において，R はサービスを表す証明書とする．R を開示可能とすることがサーバ・クライアント両者の目標となる．証明書の開示ポリシには循環が生じないとする．すなわち，ある証明書 C_1 を開示するために C_2 を要し，C_2 を開示するために C_1 を要するということが同時に起こらないとする．

定義 3（証明書の開示コスト）

証明書の開示には開示コストを要するとする．この開示コストは正の実数で表される．開示コストが大きい

表 1 ポリシの例
Table 1 Example of policies.

サーバ	クライアント
$R \leftarrow C_1$，$R \leftarrow C_2 \wedge C_3$	$C_1 \leftarrow S_1$
$S_1 \leftarrow C_2$，$S_1 \leftarrow C_4$	$C_1 \leftarrow S_2$
$S_2 \leftarrow C_2 \wedge C_3$	$C_2 \leftarrow S_3 \wedge S_4$
$S_3 \leftarrow \mathrm{true}$	$C_3 \leftarrow S_3$
$S_4 \leftarrow \mathrm{true}$	$C_4 \leftarrow S_3 \wedge S_4$

証明書ほど，開示することにリスクがあり，開示したくない証明書であることを意味する．

この開示コストの例を**表 2** に示す．R は開示コストによらず，解グラフが得られれば必ずその解グラフに含まれていることから，開示コストは 0 とする．

表 1 による交渉の例を**図 1** に示す．図 1 では上から順に交渉が進む．まず，サーバは，R を開示するためには C_1 が開示可能でなければならないことを表すポリシ $R \leftarrow C_1$ をクライアントに伝える．次に，クライアントが，C_1 を開示するためには S_1 が開示可能でなければならないことを表すポリシ $C_1 \leftarrow S_1$ を伝える．このように交渉が進み，最後に，サーバがポリシ $S_3 \leftarrow \text{true}$，$S_4 \leftarrow \text{true}$ をクライアントに伝える．これにより R を開示可能とすることができ，交渉を終了する．この交渉例において，表 2 に従って開示コストの合計を求めると 11 となる．

2.2 利用シナリオ

信用確立の交渉のシナリオを，チケット予約サービスの例で示す．チケット予約サービス利用には，利用者はサービスに “氏名”・“電話番号”・“振込み証明”，もしくは “氏名”・“クレジットカード番号” を開示する必要があるとする．また，利用者にとって，“カード番号” をサービスに開示するには，“電子商取引サイトの認定証” が必要であり，“電話番号”，“振込み証明” をサービスに開示するには，サービス会社の “住所” が必要であるとする．引用符で囲まれた情報は証明書を表す．また，証明書を開示するための条件はポリシに相当する．利用者は “電話番号” よりも “カード番号” をサービスに開示したくないとする．この選好は証明書の開示コストにあたる．すなわち “電話番号” の開示コストは “カード番号” の開示コストよりも小さい．利用者は “氏名” 開示の条件は設けず．また，サービスも “住所” や “サイト認定証” の開示条件を設けない．このとき，利用者とサービス間の交渉には以下の 2 通りが考えられる．

(1) 利用者がサービスから “サイト認定証” を受け取り，利用者が “氏名” と “カード番号” をサービスに渡す．
(2) 利用者がサービスから “住所” を受け取り，利用者が “氏名”，“電話番号”，“振込み証明” をサービスに渡す．

このような交渉手順は，ポリシにより異なる．また，いずれの方法が選択されるかは，証明書の開示コストによる．たとえば，上記のシナリオでは，利用者は “カード番号” を用いない後者の手順をとると考えられる．

証明書には “電話番号”，“カード番号” などのように，未知の相手には見せたくない機密情報が含まれている．また，たとえば，“極秘に提携している企業であれば資料を見せることができる”，“関連企業であれば割引きが受けられる” といったポリシがある場合，ポリシ自体にも機密情報が含まれている[2]．このため，ポリシを相手に渡すことにも情報漏洩のリスクが生じ，不要なポリシを渡さないことが必要となる．このように，証明書やポリシの開示をできるだけ少なくすることが必要となる．

表 2 開示コストの例
Table 2 Example of disclosure cost.

サーバ		クライアント	
証明書	開示コスト	証明書	開示コスト
S_1	4	C_1	2
S_2	5	C_2	2
S_3	1	C_3	1
S_4	2	C_4	3

メッセージ送信主体	メッセージ
サーバ：	$R \leftarrow C_1$
クライアント：	$C_1 \leftarrow S_1$
サーバ：	$S_1 \leftarrow C_2$
クライアント：	$C_2 \leftarrow S_3 \wedge S_4$
サーバ：	$S_3 \leftarrow \text{true}$, $S_4 \leftarrow \text{true}$

図 1 交渉例
Fig. 1 Example of negotiation.

3. AND/OR グラフと Automated Trust Negotiation

ATN において，証明書・ポリシ両者の開示をできるだけ少なくするため，本章では，ATN を AND/OR グラフの探索問題として記述し，グラフの探索アルゴリズムの適用を可能とする．

3.1 AND/OR グラフ

本稿では文献 5) に従い AND/OR グラフの節点と枝を**図 2** のように記す．ある節点から AND 条件で結ばれた枝をひとまとめにし，コネクタと称す．図 2

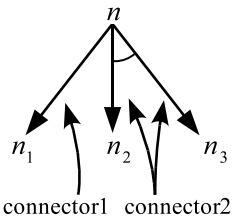

図 2 AND/OR グラフの節点と枝
Fig. 2 Nodes and edges of AND/OR graph.

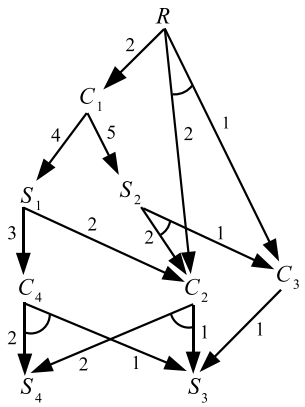

図 3 AND/OR グラフの例
Fig. 3 Example of AND/OR graph.

において，connector1 と記した n から n_1 への枝は，この 1 つの枝でコネクタとなる．また，connector2 と記した n から n_2 への枝と n から n_3 への枝で 1 つのコネクタを構成する．1 つの節点から出るコネクタどうしは OR 条件を表すとする．図 2 においては，n_1，もしくは，n_2 と n_3 が解かれれば，親節点の n が解かれる．

節点 n には K 個のコネクタがあるとする．図 2 では $K = 2$ となる．コネクタ k（$k = 1, \ldots, K$）は枝を L_k 持つとする．節点 n を親節点として，コネクタ k の枝 l（$l = 1, \ldots, L_k$）でつながっている子節点を n_{kl} とする．枝にはコストが付与されているとし，節点 n と n_{kl} を結ぶ枝 l のコストを $c(n, n_{kl})$ で表す．節点 n から終端節点までの最短経路上のコストを $h^*(n)$ と表す．$h^*(n)$ は式 (1) のように表される．

$$h^*(n) = \min_k \sum_{l=1}^{L_k} \{c(n, n_{kl}) + h^*(n_{kl})\} \qquad (1)$$

$h(n)$ を $h^*(n)$ の推定値とする．

図 3 に AND/OR グラフの例を示す．枝に付けられた数字はコストを表している．また，図中の S_3，S_4 は終端節点である．

3.2 AND/OR グラフによる Automated Trust Negotiation の記述

ATN における証明書・ポリシは，以下のように対応付けることにより，枝にコストを持つ AND/OR グラフとして表すことができる．

- 証明書を節点に対応付ける．
- ポリシの左辺の証明書を親節点，右辺の各証明書を子節点とする有効枝を付ける．1 つのポリシによる枝の集合がコネクタとなる．
- 証明書の開示に要する開示コストを，その証明書を子節点とする枝のコストとする．ある節点 n から子節点 n' への枝のコストは $c(n, n')$ となる．
- 無条件に開示できる証明書を終端節点とする．

構成される AND/OR グラフは，サーバのサービスにあたる証明書を根節点をとし，他の証明書の開示を要せずに開示できる証明書を，子節点を持たない終端節点とするグラフとなる．表 1 のようなポリシの場合，R が根節点，S_3，S_4 が終端節点となる．ある節点が複数の開示ポリシを持つ場合，この節点は複数のコネクタを持つことになる．ある節点が開示可能であるためには，この節点を親節点とするコネクタのうち，少なくとも 1 つについて，コネクタの子節点がすべて開示可能である必要がある．また，子節点が同じ枝は親節点によらずコストが等しくなる．節点 n' を子節点とする任意の枝のコストを $c(*, n')$ と記述する．

表 1 に示したポリシ，および，表 2 に示したコストを上記のように AND/OR グラフに対応付けたものは，図 3 のようになる．図 3 では，$c(*, C_2) = 2$，$c(*, C_3) = 1$ というように，子節点が同じ枝は親節点によらずコストが等しい．

AND/OR グラフによってポリシを表し，解グラフが得られれば，解グラフの終端節点から根節点へとたどり証明書を開示することで，認証を行うことができる．得られた解グラフにそって証明書を開示することから，解グラフのコストと開示する証明書のコストは同じとなる．また，ある証明書のポリシの開示は，AND/OR グラフ上ではその証明書の節点を展開し，子節点を生成することにあたる．ポリシの開示数と AND/OR グラフ上で節点を展開した数とは同じとなる．このように，ATN におけるポリシの開示・証明書の開示は以下のように対応する．

(1) ポリシの開示は節点の展開にあたり，ポリシの開示数は節点の展開数となる．
(2) 証明書の開示に必要なコストは解グラフのコストとなる．

定式化により，ATN において証明書の開示を少なくする問題は，AND/OR グラフにおいて解グラフのコストを小さくする探索問題と記述できる．また，ポリシの開示を少なくする問題は，節点の展開数を小さくする探索問題として記述できる．

4. AO* アルゴリズムの ATN への適用

AO* は，AND/OR グラフのヒューリスティック探索アルゴリズムの 1 つである[4]．AO* は，最小コストの推定値を与えるヒューリスティック関数が適格であれば，コスト最小の最適解グラフを得ることが示さ

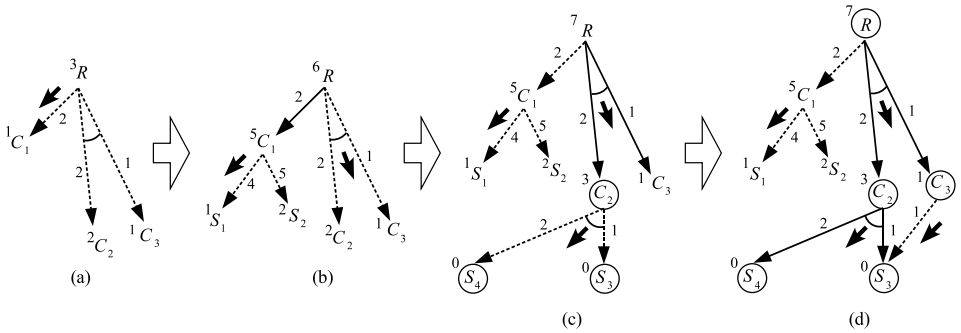

図 4 AO* アルゴリズムによる探索
Fig. 4 Search by AO* algorithm.

れている[4]．本章では，文献 5) に従い AO* アルゴリズムを記述し，定式化に基づいて ATN に適用する．

4.1 AO*

AO* は以下に示すようなアルゴリズムである．

AO* アルゴリズム

(1) 開始節点が SOLVED であれば終了する．

(2) 探索すべきグラフ G において，開始節点からマーク付けされたコネクタをたどり，部分グラフ G' を作る．マークは (5) において付けられる．アルゴリズム開始直後はマーク付けされていない．このとき G および G' は根節点のみを含む．

(3) G' から非終端節点 n を選ぶ．非終端節点が複数ある場合は，推定コストが最も大きいものを選択する．n を展開して，すべての次節点を生成し，G に追加する．すでに G に出現するものを除いた各次節点 n_j についてコスト $h(n_j)$ を求める．これらの次節点のうち，終端節点を SOLVED とする．$S = \{n\}$ とする．

(4) S が空であれば (1) へ戻る．

(5) G 上にあり，子節点が S にない節点 m を S から 1 つ選んで取り除く．m の各コネクタ k について，そのコネクタ k でつながれている m の子節点が L_k 個あるとする．コネクタ k がつないでいる子節点 $m_{k1}, \ldots, m_{kL_k}$ の推定コストとその子節点につながる枝のコストを合計し，この合計の最小値を節点 m の推定コスト $h(m)$ とする．すなわち，$h(m)$ は，式 (1) と同様に

$$h(m) = \min_k \sum_{l=1}^{L_k} \{c(m, m_{kl}) + h(m_{kl})\}$$

と記述される．m から以前に付けられたマークを消し，この最小値を得るコネクタをマークする．もし，コネクタのすべての次節点が SOLVED であれば節点 m を SOLVED にする．

(6) もし，m が SOLVED となった，あるいは，m のコストが更新されれば，マーク付けされたコネクタを通して，m がその次節点であるような m のすべての親節点を S に加える．(4) へ戻る．

AO* アルゴリズムは，最適解グラフが存在し，$h(n)$ が適格，すなわち，任意の節点 n について $h(n) \leq h^*(n)$ であればコスト最小の最適解グラフを得る[4]．解グラフのコストが $+\infty$ の場合，解は存在しない[6]．

AO* を図 3 を対象に実行した例を**図 4** に示す．図 4 において，枝に付けられた数は，その枝のコストを表す．すなわち，親節点を n，子節点を n' とすると，$c(n, n')$ を表している．円で囲まれた節点は SOLVED である節点を表している．実線のグラフはアルゴリズム中の G' を表し，実線・破線をあわせて G を表している．グラフに添えられた矢印は，アルゴリズム中のコネクタに付けるマークを表している．ある節点 n の左上に書かれた数はその節点における推定コスト $h(n)$ を表す．推定コストには，その節点から出る各コネクタに対して，コネクタに含まれる枝の数を合計し，その最小値を用いる．以下に動作の概略を示す．

(a) AO* アルゴリズムの (3) に従い，R を展開して，$h(C_1) = 1$，$h(C_2) = 2$，$h(C_3) = 1$ とする．AO* の (5) に従い，$h(R) = c(R, C_1) + h(C_1) = 3$ とし，R と C_1 のコネクタをマークする．AO* の (1) に戻る．

(b) AO* アルゴリズム (3) に従い節点 C_1 を展開する．$h(S_1) = 1$，$h(S_2) = 2$ とする．AO* の (5) に従い，$h(C_1) = c(C_1, S_1) + h(S_1) = 5$ に更新し，C_1 と S_1 のコネクタをマークする．AO* の (6) に従い，S に C_1 の親節点 R を追加し，AO* の (5) を行う．この結果，$h(R) =$

$c(R, C_2) + h(C_2) + c(R, C_3) + h(C_3) = 6$ に更新され，R から C_2，C_3 のコネクタにマークが付け替えられる．

(c) AO* アルゴリズム (3) に従い，C_2，C_3 のうち，ここでは C_2 を展開する．$h(S_3) = 0$，$h(S_4) = 0$ とする．また，S_3，S_4 を SOLVED とする．AO* の (5) に従い，$h(C_2) = 3$ に更新し，C_2 と S_3，S_4 のコネクタをマークする．C_2 を SOLVED とする．AO* の (6) に従い，S に C_2 の親節点 R を追加し，AO* の (5) を行う．この結果，$h(R) = 7$ に更新される．

(d) AO* アルゴリズム (3) に従い節点 C_3 を展開する．AO* の (5) に従い，S から C_3 を取り除き，$h(C_3) = 1$ とし，C_3 と S_3 のコネクタをマークする．C_3 を SOLVED とする．AO* の (6) に従い，S に C_3 の親節点 R を追加し，AO* の (5) を行う．この結果，R は SOLVED となる．

最後に R が SOLVED となりアルゴリズムは終了する．

4.2 AO* を用いた分散探索による交渉

AO* を ATN に適用することにより，証明書の開示コストを最小にすることが可能となる．すなわち，不要な証明書を開示せずに信用を確立できるようになる．

ATN ではサーバとクライアントの 2 エージェントで交渉が行われる．サーバ・クライアントでポリシが異なるため，エージェント間で協調しながら探索を行う必要がある．このような問題は，マルチエージェントによる探索問題として記述される[7]．AND/OR グラフによる ATN の定式化に従い，AO* を用いて 2 エージェントによる分散探索手続きを構築する．

2 エージェントで AO* を実行するために，AO* にエージェント間のメッセージ送受信を追加する．エージェント間で交換されるメッセージを以下に記す．

- AO* アルゴリズムの (2) に対応して，マーク付けされたコネクタをたどるために，コネクタに対応するポリシを送る．
- AO* アルゴリズム (3) において，推定コストを送る．AO* アルゴリズム (3) で計算される推定コストは相手のエージェントにより計算されるため，推定コストの計算要求・計算結果が送受信される．
- AO* アルゴリズム (5) において，更新された推定コストや SOLVED となったかどうかを親節点へと伝播させるため，相手のエージェントに送る．
- AO* アルゴリズム (6) において，一方のエージェントで集合 S に追加される節点は，他方のエージェントが持つ節点であるため，S を相手のエージェントに送る．

以上のメッセージをエージェント間で送受信することにより，AO* を用いた交渉が可能となる．

4.3 交 渉 例

図 5 に表 1，表 2 のポリシ・開示コストを用いた AO* アルゴリズムによる交渉例を示す．グラフとコストは図 3 の AND/OR グラフと同一である．

図 5 において，(a)，(c)，(e)，(g) はサーバが持つ部分グラフを表しており，(b)，(d)，(f) はクライアントが持つ部分グラフを表している．枝に付けられた数は，その枝のコストを表す．すなわち，親節点を n，子節点を n' とすると，$c(n, n')$ を表している．円で囲まれた節点は，SOLVED である節点を表している．実線で描かれたグラフはアルゴリズム中の G' を表し，実線・破線の両者をあわせて G を表している．グラフに添えられた矢印は，アルゴリズム中のコネクタに付けるマークを表している．ある節点 n の左上に書かれた数はその節点における推定コスト $h(n)$ を表す．推定コストを求めるヒューリスティック関数には，その節点から出る各コネクタに対して，コネクタに含まれる枝の数を合計し，その最小値を用いる．図 5 において，各グラフをつないでいる矢印は，サーバ・クライアント間で，矢印に添えられたメッセージがやりとりされていることを表す．たとえば，(a) から (b) へはポリシ $R \leftarrow C_1$ がサーバからクライアントへ送られている．以下に動作の概略を示す．

(a) サーバは節点 R を展開し，AO* アルゴリズム (3) における推定コストの問合せをクライアントに行う．推定コストは節点から出る各コネクタに対して，コネクタに含まれる枝の数を合計し，その最小値を用いて，$h(C_1) = 1$，$h(C_2) = 2$，$h(C_3) = 1$ となる．また，クライアントは証明書の開示コスト $c(*, C_1) = 2$，$c(*, C_2) = 2$，$c(*, C_3) = 1$ を返す．サーバは $h(R)$ を $h(R) = c(R, C_1) + h(C_1) = 3$ に更新し，C_1 のコネクタをマークする．サーバ・クライアントは AO* の (1) に戻る．

(b) サーバは G' を構成するコネクタを示すため，$R \leftarrow C_1$ をクライアントに開示する．クライアントは AO* アルゴリズム (3) に従い C_1 を展開する．クライアントはサーバに問合せを行い，サーバは $h(S_1) = 1$，$h(S_2) = 2$，$c(*, S_1) = 4$，$c(*, S_2) = 5$ を返す．クライアントは AO* の (5) に従い，$h(C_1) = c(C_1, S_1) + h(S_1) = 5$

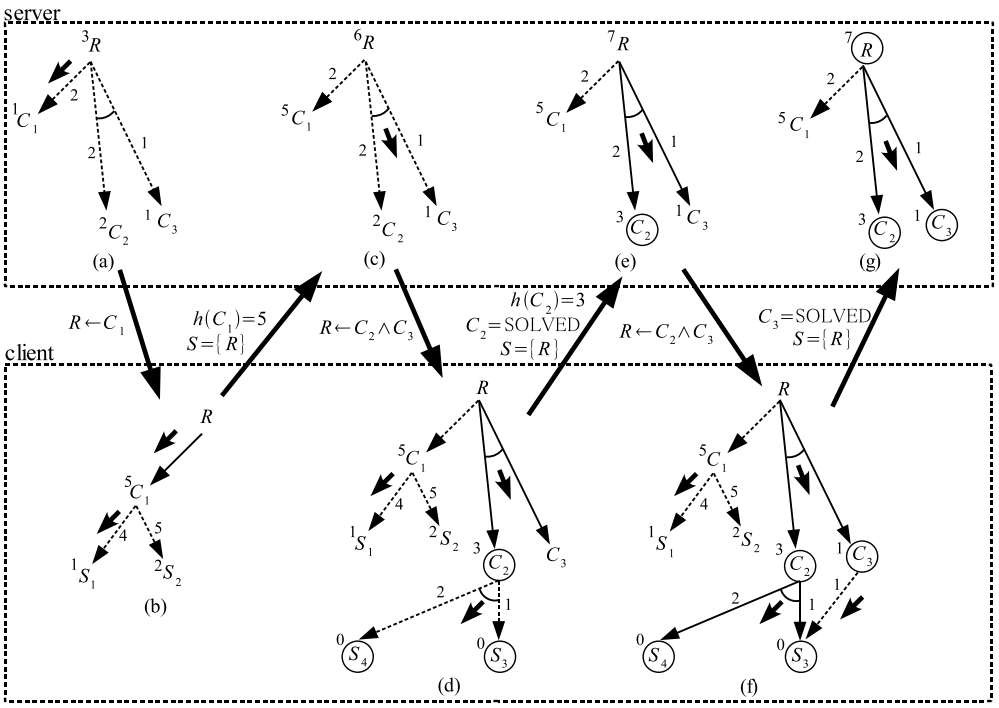

図 5 2 エージェントによる探索
Fig. 5 Search performed by two agents.

に更新し，AO* の (6) に従い，S に C_1 の親節点 R を追加し，$S = \{R\}$ とする．更新された $h(C_1) = 5$ と $S = \{R\}$ をサーバに送る．

(c) サーバは AO* アルゴリズム (5) を行う．この結果，$h(R) = c(R, C_2) + h(C_2) + c(R, C_3) + h(C_3) = 6$ に更新され，R と C_1 を結ぶコネクタから R と C_2，C_3 を結ぶコネクタにマークが付け替えられる．サーバは $R \leftarrow C_2 \wedge C_3$ をクライアントに開示する．

(d) クライアントは AO* アルゴリズム (3) に従い，C_2，C_3 の 1 つを選び展開する．ここでは C_2 を展開する．クライアントはサーバに問合せを行い，サーバは $h(S_3) = 0$，$h(S_4) = 0$，$S_3 =$ SOLVED，$S_4 =$ SOLVED，$c(*, S_3) = 1$，$c(*, S_4) = 2$ を返す．クライアントは AO* の (5) に従い，$h(C_2) = 3$ に更新し，C_2 と S_3，S_4 のコネクタをマーク，C_2 を SOLVED とする．クライアントは AO* の (6) に従い，S に C_2 の親節点 R を追加し，更新された $h(C_2) = 3$，$C_2 =$ SOLVED と $S = \{R\}$ をサーバに送る．

(e) サーバは $h(R) = 7$ に更新し，ポリシ $R \leftarrow C_2 \wedge C_3$ をクライアントに開示する．

(f) クライアントは AO* アルゴリズム (3) に従い C_3 を展開する．クライアントは AO* の (3) に従いサーバに問合せ，サーバは $h(S_3) = 0$，$S_3 =$ SOLVED，$c(*, S_3) = 1$ を返す．クライアントは AO* の (5) に従い S から C_3 を取り除き，$h(C_3) = 1$ とし，C_3 と S_3 のコネクタをマーク，C_3 を SOLVED とする．また，AO* の (6) に従い，S に C_3 の親節点 R を追加する．クライアントは $C_3 =$ SOLVED と $S = \{R\}$ をサーバに送る．

(g) サーバは，S から R を取り出し，AO* アルゴリズム (5) を行い，R を SOLVED に更新する．

最後に，R が SOLVED となることから，アルゴリズムは終了する．解グラフのコストは 7，展開した節点は R，C_1，C_2，C_3 であり，展開数は 4 となる．

5. 評　　価

本章では，従来手法が本稿の AND/OR グラフの探索によるモデルに対応付けられることを示し，シミュレーションによる AO* アルゴリズムの評価を示す．また，サーバとクライアントで個別の評価関数を用いた場合について考察する．

5.1 従来手法との対応

従来，ATN で用いられてきたアルゴリズムには，

AND/OR 木における分枝限定法によるもの[3] や，文献 1) の Parsimonious Negotiation などがある．分枝限定法によるものは，深さ優先で AND/OR グラフを探索し，開示される見込みのない枝については探索しない．Parsimonious Negotiation は，AND/OR グラフの探索においては根節点から探索を開始する Top-down 探索に相当するが，具体的にどのような探索を行うかについては定められていない．Parsimonious Negotiation は，最良優先探索における節点 n の評価関数を $f(n)$ とすると，任意の節点に対して $f(n) = 0$ となる特殊な場合であると考えられる．ATN を AND/OR グラフの探索問題として記述することにより，従来の交渉手法に対応して，解グラフのコスト・展開数といったアルゴリズムの評価が可能となる．また，一般的な探索アルゴリズムを適用することが可能となる．

AO* や分岐限定法を用いた交渉手法のようにポリシを交換し交渉を行う手法のほかに，開示可能な証明書から開示する Eager Strategy がある[1]．Eager Strategy は，終端節点から開始節点へと探索を行う Bottom-up 探索[4] として，本稿の AND/OR グラフに対応付けられる．Bottom-up 探索によってコスト最小の解グラフを求めるアルゴリズムが示されている[4] が，ATN に適用した場合，必要でない証明書を開示してしまうことが問題となる．

5.2 シミュレーション

AO* では分枝限定法のように解グラフのコストを最小化するのに加え，良いヒューリスティック関数を用いることにより，展開数を分岐限定法より少なくすることができる．解グラフのコストを最小にする手法のうち，従来用いられてきた分岐限定法と，本稿で ATN に適用した AO* の展開数を比較する．AND/OR グラフの特徴を与え，解グラフのコストや展開数に影響するものには以下の 3 つがある．

- 1 節点のコネクタ数：1 節点からの各コネクタは OR 条件を表している．どのコネクタを選択するかによって，解グラフのコストや展開数に影響する．
- 1 コネクタの枝数：1 コネクタからの各枝は AND 条件を表している．枝の数が増加すると，その節点において満たすべき条件が増加し，解グラフのコストが増加する．
- グラフの深さ：グラフの深さは ATN における交渉の長さに対応する．交渉が長ければ解グラフのコストが増加し，証明書の開示数が増える．

シミュレーションでは，分岐限定法と AO* について，深さとコネクタ数に対する展開数の変化を調べる．1 コネクタの枝数は AND 条件を表すため，分岐限定法と AO* の両者において類似した振舞いを示す．一方，分岐限定法と AO* ではコネクタの選択に違いが生じ，また，解グラフの深さも変わる．したがって，深さとコネクタ数が展開数の変化に影響すると考えられる．

表 3 に AO* と分岐限定法のそれぞれの節点の展開数，また，**図 6** に分岐限定法に対する AO* の展開数の改善率，すなわち 1 − (AO* の展開数)/(分岐限定法の展開数) を示す．図 6 は横軸にグラフの深さと 1 節点の最大コネクタ数をとり，同時に変化させた結果を示している．たとえば横軸 3 の点では，グラフの深さは 3，1 節点の最大コネクタ数は 3 としている．結果は各設定において 500 回ずつの試行の平均をとった．サーバ・クライアントの証明書とポリシは，解グラフが存在するようにランダムに生成した．また，証明書の開示コストは一様分布な乱数を用いて設定した．

図 6 より，グラフの深さとコネクタ数が増加すると，AO* の展開数が少なくなることが示される．コネクタ数が増加し，展開できる節点の選択肢が増えても，AO* は，より見込みのある節点を展開していく．このため，グラフの深さに加えてコネクタの数が増えても，分岐限定法より展開数が少なくなる．

このように，ATN を AND/OR グラフに対応付け，統一的に記述することにより，図 6 に示すような交渉手法の比較が可能となる．また本稿の AO* による手

表 3 AO* と分岐限定法のポリシ開示数
Table 3 Number of disclosed policies of branch-and-bound and AO*.

#	AO*	分岐限定法	#	AO*	分岐限定法
3	6.06	6.17	9	9.74	10.36
4	7.13	7.28	10	10.23	10.94
5	7.54	7.76	11	10.87	11.77
6	8.01	8.35	12	11.25	12.31
7	8.87	9.32	13	11.21	12.14
8	9.63	10.14	14	11.64	12.70

#は深さ・コネクタ数

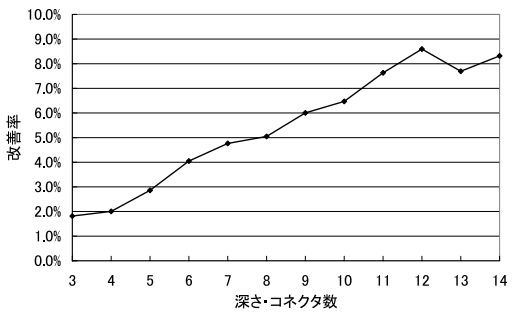

図 6 分岐限定法に対する AO* の改善率：深さ・コネクタ数
Fig. 6 Comparison of improvement between AO* and branch and bound: depth and connectors.

法は，交渉の選択肢が多い場合や交渉が長びく場合に，従来用いられてきた分岐限定法のようなアルゴリズムより展開数を少なくし，ポリシの漏洩を減少させる．

5.3 評価関数の分割

本稿のモデルではクライアントとサーバが協調的に動作するとし，サーバとクライアント両者の証明書の開示コストを合計した．一方で，証明書の開示コストを，サーバ・クライアントが個別に最小化する場合が考えられる．このとき，サーバはサーバの証明書に関する開示コストのみを合計し，クライアントはクライアントの証明書に関する開示コストのみを合計する．節点 n においてサーバ，クライアントそれぞれの証明書に関する開示コストを $h_s(n)$，$h_c(n)$ とする．図 4 を例とし $h_s(n)$ と $h_c(n)$ を示す．$h_s(n)$ は，n より下にあるクライアントの証明書にあたる未展開節点の推定コストと，n より下にあるコネクタの中で子節点がサーバの証明書にあたるものの枝のコストの合計となる．同様に $h_c(n)$ も定める．(a) では $h_s(R) = h(C_1) = 1$，$h_c(R) = c(R, C_1) = 2$ となる．(b) ではマークが $R \leftarrow C_2 \wedge C_3$ のコネクタに付けかえられて $h_s(R) = h(C_2) + h(C_3) = 3$，$h_c(R) = c(R, C_2) + c(R, C_3) = 3$ となる．なぜならば，$R \leftarrow C_1$ のコネクタでは $h_s = c(C_1, S_1) = 4$，$h_c = c(R, C_1) + h(S_1) = 3$ となるからである．(b) においてサーバは $R \leftarrow C_2 \wedge C_3$ をマークすることで証明書の開示コストを減少させる．一方，クライアントは証明書の開示コストが変わらない．図 4 の例では (b) 以降もクライアントの証明書の開示コストは変わらないが，証明書・ポリシによっては，サーバとクライアントがそれぞれ異なるコネクタを選ぶ方が各主体の開示コストを減少させる場合もある．このように，大域的な最適解がサーバ・クライアントの個別の主体にとって最適とならない場合がある．

また，交渉中の部分グラフが**図 7** のようになり，サーバが AO* アルゴリズムの (5) に従い枝をマークすると仮定する．S_1，S_2，S_3 はサーバの証明書にあたる節点，C_1，C_2 はクライアントの証明書にあたる節点とする．S_2，S_3 は未展開の節点とする．このとき，開示コストが式 (2)，(3) を満たしているとする．

$$c(S_1, C_1) + c(C_1, S_2) + h(S_2) < c(S_1, C_2) + c(C_2, S_3) + h(S_3) \quad (2)$$

$$c(C_1, S_2) > c(C_2, S_3) \quad (3)$$

式 (2) は，サーバ・クライアント両者の開示コストを合計して考えたとき，すなわち，図 7 中の (A)，(B)，(C) のすべての領域の証明書の開示コストを (i)，(ii) の枝それぞれで合計したとき，(i) の枝の方が (ii) の枝よりもコストが小さくなることを表している．

式 (3) は，サーバ・クライアント両者の開示コストを分割して考えたとき，すなわち，サーバ側のコストにあたる (B) の領域の証明書の開示コストのみを (i)，(ii) の枝それぞれで計算すると，(i) の枝の方が (ii) の枝よりもコストが大きくなることを表している．

式 (2)，(3) は同時に起こりうる．前節のように，サーバ・クライアントの証明書の開示コストを合計した場合は式 (2) より，(i) の枝がマークされ，S_2 が展開される．しかし，サーバ・クライアントの証明書の開示コストを個別に計算した場合，サーバは式 (3) により (ii) の枝をマークし S_3 を展開する．なぜならば，(ii) の枝によりサーバの証明書の開示コストを減少できるからである．図 7 において S_3 が SOLVED である場合には，最適解ではない (ii) の枝を解グラフとしてしまう．この例ではサーバがマークする場合を示したが，クライアントがマークするときにも同様の問題が起こる．

以上の問題は，サーバ，クライアントが証明書開示コストを個別に最小としようとすると，展開する順番が変わるだけでなく，得られた結果が最適とならないことを示している．本稿では AO* アルゴリズムの場合を示したが，分岐限定法など他のアルゴリズムを適用した場合にも同様に最適性が保証されなくなる．サーバ，クライアントが個別に開示コストを最小とした場合にも，大域的に最適な証明書の開示手順が得られるプロトコルの開発が今後の課題である．

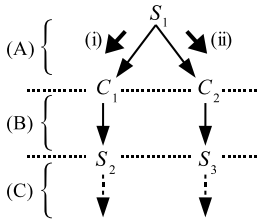

図 7 開示コストの分割
Fig. 7 Separation of disclosure cost.

6. おわりに

本稿では以下の 2 点について述べた．

(1) ATN を 2 エージェントによる AND/OR グラフの分散探索として定式化した．
(2) (1) の定式化に従い，AND/OR グラフにおいて最小コストの解グラフを求める AO* アルゴリズムを適用した．

ATN はポリシ・証明書の開示を少なくすることを目標とするが，交渉の統一的な記述がなく，ポリシ・証

明書の開示が最小である基準が明確でない．本研究では ATN を AND/OR グラフ上の探索として定義し，

- 節点の展開をポリシの開示
- 解グラフのコストを証明書の開示コスト

として対応付けた．また，AND/OR グラフで最小コストの解グラフを与える AO* を ATN に適用した．

本研究の定式化と探索アルゴリズムの適用は，ATN に対して以下の点で有用であると考えられる．

- **定式化のメリット**：ATN を AND/OR グラフの探索と定式化することにより，既存の AND/OR グラフ探索のアルゴリズムを適用できる．完全性や最適性などの評価を可能とすることで，各アルゴリズムをどのような交渉に適用できるのかを明らかにすることができる．また，どれだけ証明書やポリシの開示を少なくするかを解グラフのコストや節点の展開数によって示すことで，交渉手法間の性能比較を可能とする．
- **アルゴリズムのメリット**：定式化に従い AO* アルゴリズムを適用し，解グラフのコストを最小とした．解グラフのコストが最小であることは，証明書の開示コストが最小であることを意味する．これにより，不要な証明書の開示による情報漏洩を防ぐことができる．また，AO* アルゴリズムは従来の分岐限定法と比較してポリシの開示数を減少させる．これにより，不要なポリシの開示による情報漏洩を従来手法に比べて少なくし，情報漏洩のリスクを減少させる．

本稿の定式化に従った AO* の適用により，証明書の開示コストについては最適となるアルゴリズムを示した．今後はポリシの開示を少なくするため，節点の展開数の少ない探索手法の開発やヒューリスティック関数の改善が必要となる．本稿のモデルではポリシの開示のコストは定めておらず，“このポリシを開示するよりも，他のポリシを開示した方が良い” といった選好は考えていない．これに対して，ポリシの開示のコストを導入するモデル拡張も必要となる．

また，本稿のモデルでは，サーバ・クライアントの両エージェントがサービスの提供・利用を目標として交渉手続きを行うが，実際の環境においては，両エージェントは，各エージェントの効用を最大化するなど，利己的に振る舞う．そのような場合に対応できるプロトコルを考えることが必要となる．

謝辞 本研究を進めるにあたり，有益なご助言をくださいました京都大学大学院情報学研究科社会情報学専攻の八槇博史講師に感謝申し上げます．本研究は，日本学術振興会科学研究費 基盤研究（A）(15200012, 2003-2005）の補助を受けた．

参 考 文 献

1) Winsborough, W.H., Seamons, K.E. and Jones, V.E.: Automated Trust Negotiation, *DARPA Information Survivability Conference & Exposition*, pp.88–102 (2000).
2) Seamons, K., Winslett, M. and Yu, T.: Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation, *Proc. Network and Distributed System Security Symposium* (*NDSS 2001*) (2001).
3) Yu, T., Ma, X. and Winslett, M.: An Efficient and Complete Strategy for Automated Trust Negotiation over the Internet, *ACM Conference on Computer and Communications Security*, pp.210–219 (2000).
4) Martelli, A. and Montanari, U.: Additive AND/OR Graphs, *Proc. IJCAI*, pp.1–11 (1973).
5) Nilsson, N.J.: *Principles of Artifical Intelligence*, chapter3.2 AO*: A Heuristic Search Procedure for AND/OR Graphs, pp.103–109, Morgan Kaufmann (1980).
6) Pearl, J.: *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, chapter 2.4 Specialized Best-First Algorithms: Z*, A*, AO, and AO*, pp.56–65, Addison-Wesley (1984).
7) Yokoo, M. and Ishida, T.: Search Algorithm for Agents, *Multiagent Systems*, pp.165–199, MIT Press (1999).

(平成 17 年 11 月 25 日受付)
(平成 18 年 6 月 1 日採録)

中塚 康介（正会員）

平成 12 年京都大学工学部情報学科卒業，平成 14 年同大学大学院情報学研究科社会情報学専攻修士課程修了．現在，同大学院博士課程在学中．

石田 亨（正会員）

昭和 51 年京都大学工学部情報学科卒業，昭和 53 年同大学大学院修士課程修了．同年日本電信電話公社電気通信研究所入所．現在，京都大学大学院情報学研究科教授．工学博士．自律エージェントとマルチエージェントシステム，セマンティック Web 技術に取り組む．