社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2006－CSEC－34 (37)
2006／7／21

内積比較プロトコルに基づく 分散巡回セールスマン問題のセキュアな最適化

佐久間 淳† 小林 重信†

† 東京工業大学総合理工学研究科 〒226-8502 神奈川県横浜市緑区長津田町 4259
E-mail: †jun@fe.dis.titech.ac.jp, ††kobayasi@dis.titech.ac.jp

あらまし 本稿では分散化された組み合わせ最適化問題のためのセキュアな局所探索プロトコルを提案する．分散組み合わせ最適化問題とは、コスト関数に関する情報が複数のパーティーに分散している最適化問題であり、例えば分散巡回セールスマン問題では都市間移動コストや巡回する都市集合に関する情報が分散化されている。秘匿内積比較プロトコルに基づいた局所探索法では、分散巡回セールスマン問題において分散化された情報を共有することなく最適化を実行することが可能である。提案プロトコルは準備段階において都市数 n に対して $O(n^2)$ の時間計算量を必要とするが，コスト一評価あたりの計算量は $O(1)$ である。計算機実験では都市数が 1000 を越える大規模分散 TSP においても，実用的な時間で局所最適解に到達することが示された．

キーワード マルチパーティープロトコル，プライバシー，最適化，内積比較，TSP，サプライチェーンマネジメント

Secure optimization of Traveling Salesman Problem using Scalar Product Comparison Protocol

Jun SAKUMA† and Shigenobu KOBAYASHI†

† Department of Computational Ingelligence and Systems Science, Nagatsuta-cho, 4259, Midori-ku, Kanagawa, 226-8503, Japan
E-mail: †jun@fe.dis.titech.ac.jp, ††kobayasi@dis.titech.ac.jp

Abstract We propose a secure local search protocol for the distributed combinatorial optimization problem. In the distributed combinatorial optimization problem, information regarding the cost function is distributed among multi parties. In distributed Traveling Salesman Problem (TSP), traveling costs between any two cities and city sets to be visited are distributed and private. Distributed TSP can be securely solved by secure local search based on private scalar product comparision protocol without revealing distributed information. The time complexity of our protocol is $O(n^2)$ in preparation phase where n is the number of cities but the computation time is kept $O(1)$ per one iteration. The waiting time required to complete the optimization is reasonable even when the city-size is more than a thousand and the optimization is processed without sharing the distributed information.

Key words Multi-party protocol, privacy, optimization, scalar product comparison, TSP, supply chain management

1. はじめに

配送ルート，生産スケジューリング，購買計画などの決定問題において解かれる組み合わせ最適化問題は，オペレーションズリサーチの分野でその効率的解法が長年に渡り研究されてきた．複数の企業で構成されるサプライチェーンマネジメント (SCM) は，複数企業の協力関係の下で，顧客対応の高速化やサイクルタイムの短縮を目的としており [1]，同種の問題を，分散化された環境において最適化する必要がある．

分散化された在庫，生産計画，需要予測などの情報をリアルタイムに集約し，意思決定に活用するために，SCM では標準的データ交換方式である EDI (Electronic Data Interchange) などが利用されている [2]．ネットワークによる情報共有は SCM に不可欠であるが，その反面，機密情報の漏洩リスクを増加させる [3]．最適化の観点からは，情報共有の度合いが高まるほど精密な最適化が可能になるが，それにつれて開示すべき情報

量は増えるため，通常の最適化法においては開示される情報量と得られる最適解の精度はトレードオフの関係にあると考えてよい．

そこで本稿では，最適化すべきコスト関数に関する情報がプライベートであり，かつ，複数のパーティーに分散しているような組み合わせ最適化問題を対象として，コスト関数に関連する情報をなるべく共有することなく，最適解を探索する方法について考察する．

Privacy-preserving data mining の分野では，分散した機密データを互いに開示することなくマイニングアルゴリズムを実行するための，さまざまな multiparty computation protocol がビルディングブロックとして提案されている．例えばセキュアな和計算，内積計算 [17] [18]，統計計算 [11]，データベーステーブルの join 演算，intersection 演算 [10] などがこれまでに提案されている．

またプライバシーを保護した最適化に関してもいくつかの研究が知られている．[14] では分散制約最適化問題のための深さ優先探索法が提案された．また [15] では動的計画法に基づく一般化 Vickley オークションのための最適化プロトコルが提案された．

これらのセキュアな最適化プロトコルでは，問題が有する特定の構造を活用することによって計算量や通信量を削減することに成功している．しかしながら，問題構造の活用とプライバシー保護は一般に相反し，両者を両立させることは多くの場合困難である．また，新しい別の問題に適用する際には，その問題に特化したプロトコルを一から構成しなければならないといという難点がある．

一方，メタヒューリスティクスは多項式時間でのアルゴリズムの終了が保障されず，最適解への到達保障もないが，問題をユーザに与えられたブラックボックスとして扱うため，問題構造に対して独立にアルゴリズムを設計することが可能である．従ってメタヒューリスティックに基づくセキュアな最適化のためのプロトコルが設計されれば，解法は問題の構造に依存しないため，漏洩する情報を少なく抑えることができ，また，最適化可能な問題の範囲も広くすることができると考えられる．そこで本稿では，さまざまな分散組み合わせ最適化問題に適用可能なセキュアなメタヒューリスティックを提案する．

分散組み合わせ最適化問題の例として，有名な巡回セールスマン問題 (Traveling Salesman Problem, TSP) を取り上げる．分散 TSP においては，巡回すべき都市と都市間移動コストが複数のパーティーに分散されている．分散 TSP の直感的な理解を助けるために，以下に二つの単純な例を示す．

シナリオ 1: 二つの運送会社 E_1 と E_2 があるとする．顧客 E_C は二つの運送会社のうちのどちらかに，地点 $F_1, ..., F_n$ における配送を依頼しようとしている．運送会社の選択のためには，顧客 E_C は E_1 と E_2 から提示された最適化された配送ルートのコストを比較する必要があるが，E_C は契約前には配送地点を明かすことができない．一方，運送会社 E_1,E_2 において，それぞれの二地点間の配送コストは機密情報であるため，それを顧客 E_C に明かすことはできない．E_C はどのようにして E_1,E_2 から提示されたルートとコストを，互いに機密情報を開示することなく比較することができるだろうか．

シナリオ 2: 地域 A, B における運送会社をそれぞれ E_A, E_B とする．顧客 E_C は地域 A 内の地点 $F_1^A, ..., F_n^A$ に配送を依頼しようとしている．また，同様に，地域 B 内の地点 $F_1^B, ..., F_n^B$ に配送を依頼しようとしている．今，E_A, E_B と E_C は協力関係を結び，配送コストを減らすために配送ルートを一元化しようとしているとしよう．一元化された最適な配送ルートを決定し，削減可能なコストを見積もるためには，E_A, E_B はそれぞれ二地点間の配送コストを開示しなければならないが，互いにそれは機密情報にあたるため開示することができない．これらの企業はどのようにして最適に一元化された配送ルートを互いに機密情報を明かすことなく決定することができるだろうか．

上記のシナリオから分かるように，配送経路の決定問題では，地点間配送コストと配送地点そのものが機密情報になりえる．そしてこれらの情報が分散している場合，最適経路の探索は困難である．

本稿ではこのような分散 TSP をセキュアに最適化することができる局所探索のためのプロトコルを設計し，その実用性を計算機実験によって評価することを目標とする．提案プロトコルはメタヒューリスティックとしては最も単純な局所探索を取り上げているが，より高度な方法への拡張はごく自然に可能である．また適用する問題も，分散 TSP のみならず，ナップザック問題 (KSP) や配送計画問題 (VRP) など，さまざまな問題にに拡張可能である．

以下，2 章では TSP とその局所探索法について簡単に説明し，TSP において考慮すべきプライバシーとデータ分割モデルについて議論する．3 章ではプロトコル設計のビルディングブロックとなる秘匿内積比較プロトコルを説明し，これを用いたセキュアな局所探索法を提案する．また提案プロトコルの計算量についても議論する．4 章では計算機実験によりネットワーク上の計算機を用いて提案プロトコルの動作速度を測定し，その実用性を評価する．5 章は 3 章で提案したプロトコルが，どのような問題やデータモデルに拡張可能かについて議論する．6 章はまとめと，将来の方向性について触れる．

2. Distributed TSP and its Privacy

2.1 Traveling Salesman Problem

巡回セールスマン問題は，n 個の都市が与えられたとき，これらの都市をちょうど一回ずつ通って最初の都市に戻ってくる最も短い巡回路を求める問題である．都市の対ごとにその 2 都市間の距離テーブルが与えられる．

以下に本稿で用いる TSP の解の表現方法とコストの計算方法を定める．頂点 (都市) $V = \{v_1, ..., v_n\}$ の個数 n に対して，(無向) 枝は $n(n-1)/2$ 本存在し，巡回路はこの枝がハミルトン閉路を生成するように選択された n 本の枝集合で表現される．巡回路に含まれる枝集合を

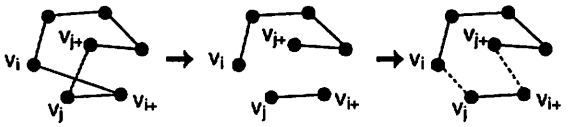

図 1 k-opt 近傍の概念図．左:巡回路 x，中:2 本の枝を取り除いた巡回路，右:新たに 2 本の枝を付け加え生成された 2-opt 近傍．

Fig. 1 An example of 2-opt neighborhood. left: tour x, center: tour removed two edges, right: tour added new two edges

$$x^T = (\sigma_{1,2}, \sigma_{1,3}, ..., \sigma_{2,3}, \sigma_{2,4}, ..., \sigma_{n-1,n}) \quad (1)$$

と表現する．ここで，$\sigma_{i,j} \in \{0,1\}$ であり，頂点 v_i, v_j を結ぶ枝が巡回路中に存在すれば 1，そうでなければ 0 をとるインジケータ変数である．枝に付与された距離を以下のようにベクトルで表現する．

$$y^T = (d(v_1, v_2), ..., d(v_1, v_n), \quad (2)$$
$$d(v_2, v_3), ..., d(v_2, v_n), ..., d(v_{n-1}, v_n)).$$

ここで $d(v_i, v_j)$ は頂点 v_i, v_j 間の距離を定める距離関数である．x, y とも，ベクトル長は $\frac{n(n-1)}{2}$ である．このとき巡回路長 $f(x)$ は下式のように計算される．

$$f(x) = y^T x. \quad (3)$$

TSP 最適化は，x がハミルトン閉路であることを条件とし，ある y について $f(x)$ を最小化する最適化問題として定式化される．

TSP は与えられた距離関数において全ての都市を巡回する最適巡回路を求めることが目的となるが，本研究では，任意に選択された（n 都市を含む）4 以上の都市に関する最適巡回路を求める問題を対象とする．

TSP において，最も単純かつ実装が容易な近似的最適化解法は，k-opt 近傍による局所探索である．k-opt 近傍とは，現在の巡回路 y に対して任意の k 本の枝を取り去り，ハミルトン閉路が構成されるように新たな k 本の枝を付け加えることで定義される．Fig.1 に 2-opt 近傍の構成例を示す．以下は本研究で用いる局所探索のアルゴリズムを示している．ただし $N(x)$ はユーザーが定義する巡回路 x の近傍である．

[Local search based on neighborhood $N(x)$]

- **Input:** Initial solution x_c
- **Output:** Local optimal solution x^*

（1） $x \in_r N(x_c)$, $N(x_c) \leftarrow N(x_c) \setminus \{x\}$

（2） If $f(x) < f(x_c)$, then $x_c \leftarrow x$

（3） If $N(x_c) = \emptyset$ or satisfies some terminate condition, $x^* \leftarrow x_c$ and output x^*. Else, go to step 1.

局所探索は，用いられた近傍に基づく局所最適解への収束が保障されている．

2.2 Data Partitioning Model and Privacy in Distributed TSP

$n' \leqq n$ ノードを含むような任意の $V' \subseteqq V$ について，巡回路が V' のハミルトンサイクルであり，ツアーベクターが $d' = n'(n'-1)/2$ 個の枝を含むような，n' 都市を巡回する TSP が定義される．このとき巡回路 x のコストは V で構成される TSP と同様に $x \cdot y$ と書かれる．以降，本稿では任意の $V' \subseteqq V$ における TSP を扱う．

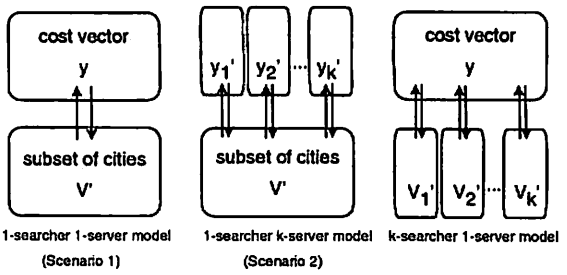

図 2 分散 TSP におけるデータ分割モデル

Fig. 2 Data partitioning model in distributed TSP

1 章に示した二つのシナリオから予期されるように，TSP におけるプライバシーは y，V' および x に相当する．もし y と V' が分散され互いにプライベートな情報であるならば，ツアー候補 x は V' なしでは構築することはできず，また，ツアーのコスト $x \cdot y$ は x および y なしでは評価することはできない．

V' で定義される TSP の最適ツアーを探索するパーティーを *searcher*，コストベクターを保持するパーティーを *server* と呼ぶことにしよう．このとき，最も単純な問題設定は searcher が V' を，server が y を別々に保持しており，V' と y は互いにプライベートである状況である．このデータ分割モデルを *1-server 1-searcher model* あるいは (1,1)-TSP と呼ぶことにする（図 2 左）．1 章に示したシナリオ 1 は，異なるコストベクターにおける二つの (1,1)-TSP の最適解の比較に相当する．二つの運送会社が異なるコストベクターを持ち，クライアントは同一のプライベートな配送地点集合に基づきルート最適化を行い，二つの会社からオファーされた最適ルートを比較する問題である．

コストベクター y が $y = y_1 + y_2 + ... + y_k$ に分割され，i 番目の server が分割された y_i を保持し，searcher が V' を保持しているとき，このデータ分割モデルを *k-server 1-searcher model* あるいは $(k,1)$-TSP と呼ぶ（Fig2 中）．シナリオ 2 は，このモデルに相当している．二つの運送会社がそれぞれ異なるコストベクターをもち，クライアントはプライベートな配送地点集合に関する一元化された最適配送ルートを探索する問題である．．

このモデルと対照して，V' が $V' = V'_1 \cup V'_2 ... \cup V'_k$ のように分割され，i 番目の searcher が分割された V'_i を保持し，server が y を保持するモデルを *1-server k-searcher model* あるいは $(1,k)$-TSP（図 2 右）と呼ぶ．

分散 TSP のプライバシーについて考察するために，4 都市 (1,1)-TSP ($V = \{A, B, C, D\}$) を例にとる．信頼できる第三者 (Trusted Third Party, TTP) を利用してこの (1,1)-TSP を解く場合，searcher は $V' = \{A, B, C, D\}$ を，server は y をそれぞれ TTP に送信する．TTP は何らかの最適化アルゴリズムを

実行し，最適解 $x^* = (A \to B \to C \to D \to A)$ を発見し，それを searcher に送ったとしよう．searcher は最適解 x^* を得ている以上，searcher はその他の可能なツアー $x_1 = (A \to B \to D \to C \to A)$，$x_2 = (A \to C \to B \to D \to A)$ のコストは x^* のコストよりも大きいことを知る．この例から，searcher は明らかに二つの不等式 $x^* \cdot y < x_1 \cdot y$ と $x^* \cdot y < x_2 \cdot y$ を知ることになる．

TTP を用いないいかなるプロトコルも，TTP を用いるプロトコル以上にセキュアにはなりえないため，searcher に 2 以上のこのような不等式を知られずに分散 TSP を解くことは不可能である．これを考慮して，1-server 1 searcher model における分散 TSP のセキュアな最適化は，以下の様に記述される:

Problem Statement 1. ((1,1)-TSP 最適化) *Server* はコストベクター y を，*searcher* は都市の部分集合 $V' \subseteq V$ を保持している．二人の間のプロトコルの実行の結果，*searcher* は V' のハミルトンサイクルであり，$y^T \cdot x$ を最小にする x^* と，ツアーコストに関する二つ以上の不等式を得るが，それ以外は何も得ない．*server* は何も得ない．

1-searcher k-server model($(1,k)$-TSP) と k-searcher 1-server model($(k,1)$-TSP) はこの定義の自然な拡張によって定義される．次章では，秘匿内積比較プロトコルに基づく分散 TSP のセキュアな局所探索法について説明する．

3. 分散 TSP のためのセキュアな最適化プロトコル

3.1 秘匿内積比較プロトコル

分散 TSP の局所探索において，step 1 および 3 は searcher が単独で実行可能である．一方 step2 は searcher 単独では実行できない．step 2 では解 x_c とその近傍 x のコスト $y^T \cdot x$ と $y^T \cdot x_c$ の比較結果が必要であることから，searcher と server の間の局所探索は，以下の秘匿内積比較問題に帰着される．秘匿内積問題の定義を以下に示す．

Problem Statement 2. (秘匿内積比較問題) *Alice* がベクター $x_1, x_2 \in \{1,...,K\}^d$，*Bob* がベクター $y \in \{1,...,K\}^d$ を保持している．ただし $x_1 \cdot y > 0$, $x_2 \cdot y > 0$ である．二人の間のプロトコルの実行の結果，二人は内積の大小比較の結果を表す不等式 $I \in \{I_+, I_-\}$ を共有するが，それ以外は何も得ない．

上記の問題定義に基づく秘匿内積プロトコル [21] を示す:

[Protocol : scalar product comparison]

- **Input of Alice:** Private vectors $x_1, x_2 \in \{1,...,K\}^d$
- **Input of Bob:** Private vector $y \in \{1,...,K\}^d$
- **Output of Alice and Bob:** Inequation $I \in \{I_+, I_-\}$

(1) Bob: Generate a private and public key pair (s_k, p_k) for homomorphic encryption and send p_k to Alice

(2) Bob: Encrypt Bob's vector $c_i = Enc_{p_k}(y_i)$ and send it to Alice$(i = 1,...,d)$

(3) Alice: Generate numbers $r, r', r'' \in Z_m$ randomly, such that $d^6K^{12} < r$, $r < r' < (1 + \frac{1}{dK^2})r$, $0 < r'' < dK^2$

(4) Alice: $x_m \leftarrow rx_2 - r'x_1$

(5) Alice: Compute $w_m = \left(\prod_{i=1}^{d} c_i^{x_{m,i}}\right) \cdot Enc_{p_k}(r'')$ and send w_m to Bob

(6) Bob: Compute $S = Dec(w_m)$

(7) Bob: If $S < 0$, send I_- to Alice. Else, send I_+ to Alice.

(8) Alice: Receive an inequation I

ただし $x_{m,i}$ および c_i はそれぞれ x_m, c の i 番目の要素をあらわす．プロトコルの正しさとセキュリティに関する証明は [21] を参照されたい．このプロトコルにおいて，Alice は自由に x_1 と x_2 を事前に交換することができる．つまり，比較結果は両者が共有するが，ベクターの並べ方は Alice が自由に決定できるため，結果として大小は Alice にしか知られない．

3.2 秘匿内積比較に基づく局所探索

本節では，秘匿内積比較プロトコルを用いて，(1,1)-TSP のための局所探索法を提案する．

入力ベクターの定義域 Z_K^d，都市集合 V および公開鍵 p_k が server と searcher の間で合意されているとき，秘匿内積比較に基づくセキュアな局所探索プロトコルは以下のように記述される:

[Secure local search for distributed TSP]

- **Input of Server:** Private cost vector $y \in Z_K^d$
- **Input of Searcher:** Private city set $V' \subseteq V$, $|V'| = n' < n$
- **Output of Searcher:** Local optimal tour $x^* \in Z_K^d$

(1) Server: Generate a private and public key pair (s_k, p_k) for homomorphic encryption and send p_k to Searcher

(2) Server: Encrypt cost vector $c_i = Enc_{p_k}(y_i)$ and send it to searcher $(i = 1,...,d)$

(3) Searcher: Generate a random initial tour x_c from V'

(4) Searcher: Generate a set of neighborhood $N(x_c)$

(5) Searcher: $x \in_r N(x_c)$, $N(x_c) \setminus \{x\}$. $b \in_r \{0,1\}$. If $b = 0$, $x_1 \leftarrow x_c$, $x_2 \leftarrow x$. Else, $x_1 \leftarrow x_c$, $x_2 \leftarrow x$

(6) Searcher: Generate numbers $r, r', r'' \in Z_m$ randomly, such that $d^6K^{12} < r$, $r < r' < (1 + \frac{1}{dK^2})r$, $0 < r'' < dK^2$, $x_m \leftarrow rx_2 - r'x_1$

(7) Searcher: Compute $w_m = \left(\prod_{i=1}^{d} c_i^{x_{m,i}}\right) \cdot Enc_{p_k}(r'')$ and send w_m to server

(8) Server: Compute $S = D(w_m)$. If $S < 0$, send I_- to searcher. Else, send I_+ to searcher.

(9) Searcher: Receive an inequation I from server. If $b = 1$ and $I = I_+$, $I \leftarrow I_-$. Else if $b = 1$ and $I = I_-$, $I \leftarrow I_+$.

(10) Searcher: If $I = I_-$, $x_c \leftarrow x$ and go to step 4.

(11) Searcher: If $N(x_c) = \emptyset$ or satisfies some terminate

condition, $x^* \leftarrow x_c$ and output x^*. Else, go to step 5.

step 5 において，x と x_c はランダムに交換されるが，searcher は step 9 における符号反転操作によって，$x \cdot y$ と $x_c \cdot y$ に関する正しい不等式を得ることができる．ゆえに，step 5 から step 9 にかけての計算は，局所探索における step 2 と全く同じ操作を実現している．このとき，もし同一の乱数の種が利用されているならば，このプロトコルにより得た解は，通常の局所探索において得た解と一致する．step 5 および 9 における swap 操作は，S の符号に関する server の統計的攻撃 (statistical attack) を防ぐために行われている(注1)．

server と searcher の両方は $x_1 \cdot y$ と $x_2 \cdot y$ の比較結果を得るが，steps4 から 8 にかけては秘匿内積比較に相当するため，それ以外の情報を得ないことが保障される．この局所探索プロトコルのセキュリティは，二つのプライベートなサブプロトコルの合成からなる計算のプライバシーを保障する"composition theorem"によって証明することができる．詳細は [5] を参照されたい．

3.3 時間計算量と通信計算量に関する考察

本節では，提案プロトコルの時間計算量および通信計算量について考察する．各ステップの暗号学的計算と通常の計算の時間計算量および通信計算量は表 1 にまとめられている．

通信計算量: server と searcher の間の通信は step 2，7 および 8 において発生する．分散 TSP においては，server のコストベクター y は固定されているが，searcher のツアーベクター x，x_c は頻繁に変更される．それゆえ，サーバサイドのベクター y が server によって暗号化され，step 2 において searcher に送信されている．この通信は server-sice のベクターが変更されない限り，一度のみ実行されればよい．

server-side の暗号化ベクターが送信された後は，通信は，ツアーベクトルを一つ生成するたびに w_m を searcher から server へ送信する step 7 と，不等式を server から searcher に送信する step 8 の二ステップである．両ステップとも通信コストは定数オーダーであり，たとえプライバシーが考慮されないとしてもこの通信は省略できないため，このステップに関してはこれ以上通信コストを減らすことは不可能である．

時間計算量: step 4 において，近傍は searcher によって生成される．もし近傍が網羅的に数え上げるとしたら，時間計算量は $O(n^2)$ が必要になるが，定数個の近傍を生成し，使い果たした場合のみ近傍を再生成することで，おおむね $O(n)$ にまで削減することが可能である．

時間計算量の観点からも最もコストかかるステップは，step 7 における累乗計算である．このステップをナイーブに実装すると時間計算量は $O(d)(= O(n^2))$ であり，大規模問題にはおいてはこのステップがボトルネックになる．このステップにおける計算量を削減するために，近傍操作によって変更される枝数は，全枝数に比べて非常に少ないことを利用する．

例えば，2-opt は 2 つの枝を除き，新しい二つの枝を加える．このとき，変更枝数は $\frac{n(n-1)}{2}$ 個のうちわずか 4 である．x を x_c から 2-opt によって生成されたツアーとしよう．このとき，ベクター $rx - r'x_c$ の二つの要素は r，二つの要素は $-r'$ であり，残りの全ての要素は全て 0 である．step 7 におけるべき乗計算では，この大部分の 0 をスキップすることが可能であるため，時間計算量は 2-opt 近傍では定数オーダーである．より一般的には，近傍操作によって変更される枝数が u である場合，step 7 における時間計算量は $O(u)$ に節約され，都市数 n に依存しない．分散化されていない TSP においてもコスト評価には $O(u)$ を要することから，この計算はこれ以上節約することは困難であると考えられる．

step 6 では，x と x_c の差異を調べるために $O(n)$ が必要である．用いる近傍操作の種類によっては，この操作を定数オーダーの計算量で実行することが可能であるが，ここで一般性を考えこのようにした．

重要なのは，step 2 以外の step では，通信計算量と暗号学的計算に要する時間計算量が，一つの近傍解生成あたり定数オーダーに抑えられ，都市数には依存しない点である．

表 1 提案プロトコルの時間計算量および通信計算量

Table 1 The time and communication complexity of the proposed protocol.

step	time (non-cryptographic)	time (cryptographic)	communication
2	—	$O(n^2)$	$O(n^2)$
4	$O(n)$	—	—
6	$O(n)$	—	—
7	—	$O(1)$	$O(1)$
8	—	$O(1)$	—
9	—	—	$O(1)$

3.4 計算機実験

提案プロトコルのスケーラビリティを調べるために，分散 TSP における計算機実験を行った．この実験で用いられた問題は burma14, berlin52, st70, rat195, att532, rat575, pcb1173 である．問題名にある数字は，その問題の都市数を表している．全ての問題は TSPLIB [20] に掲載されているものである．プロトコルにおいて，searcher は任意の訪問する都市集合 V' を選択することができるが，簡単のため，ここでは $V' = V$ とした．

512-bit および 1024-bit keyPaillier 暗号系 [19] を準同型性暗号として用いた．server および searcher プログラムは J2SE ver. 1.4.1 で実装された．server および searcher は 100Mbps Ethernet 上に設置され，プログラムは Xeon2.8GHz(CPU), 1GB(RAM) Windows PC において実行された． 実験においては，以下の 4 種類の計算時間を測定した．

(1) Encryption : step 2 において，サーバサイドのベクターを暗号化するために要した計算時間

(2) Downloading : step 2 において，暗号化されたサーバサイドのベクターを searcher がダウンロードするために要した

(注1)：符号を一様ランダムに反転させることによって，$S < 0$ の確率と $S \geq 0$ の確率はともに 1/2 になる．

時間

(3) Comparison : step 5 から step 10 において，searcher が二つの異なる解候補のコストを比較するために要した時間

(4) Optimization : step 2 からプロトコル終了時まで，searcher が初期解を生成した後に，最終的な局所最適解を得るために要した時間

図 3 は，上記各項目の，都市数に対する計算時間を示している．横軸は都市数，縦軸が計算時間である．

図 3 上段は，サーバサイドのベクターの暗号化および暗号化ベクトルの searcher のダウンロードに要する時間が，都市数に対して 2 乗のオーダーで増加することを示している．暗号化に要する時間は非常に長いが，暗号化はサーバが単独で実行可能であるため，searcher の待ち時間には影響しないことに注意されたい．ダウンロードもまた比較的時間を要する step であるが，この step は searcher が server と初めて通信する初回のみ実行される．512-bit key の結果では，ダウンロード時間にオーバーヘッドが生じている．

図 3 左下はコストの比較一回あたりの計算時間を示している．局所探索の実行中には，非常に多くの回数の比較操作が行われるため，このステップの計算時間が少ないことが求められる．前節で触れたように，比較に要する計算時間は都市数にかかわらず一定に保たれている．ただし，512-bit key における rat575 および pcb1173 の結果ではわずか上昇が見られる．この結果は非暗号化計算 (step 2(e)) における線形の計算によるものである．

図 3 右下は，searcher が局所最適解を得るために要した待ち時間を表している．512-bit key において，rat575 および pcb1173 はそれぞれ 8.2 分および 38 分で最適化された．1024-bit key においては，それぞれ 50 分および 166 分で最適化された．3.3 に示した最適化の工夫を行わない場合，512-bit key において rat195 を最適化した場合 4 日以上の時間がかかるため，高速化の効果が非常に大きいことがわかる．

4. 提案プロトコルの拡張

本章では，秘匿内積比較のマルチパーティーへの拡張と，それを用いたセキュアな局所探索法を示す．また，分散 TSP 以外の組み合わせ最適化問題への適用についても議論する．

4.1 $(1,k)$-TSP への拡張

秘匿内積比較は，Alice が x_1, x_2 を，Bob が y を保持しているときに，$x_1 \cdot y$ および $x_2 \cdot y$ の大小を比較する．

Bob のベクターが $y = \sum_{i=1}^{k} y^i$ のように分割され，Bob-1,...,Bob-k がそれぞれ秘密のベクター y^i を保持しているとしよう．このとき，秘匿内積比較プロトコルは，$\left(\sum_{i=1}^{k} y^i\right) \cdot x_1$ と $\left(\sum_{i=1}^{k} y_k\right) \cdot x_2$ を比較するためのプロトコルに拡張することができる．これを $(1,k)$-内積比較と呼ぶことにする．

簡単のために，3-party モデル ($(1,2)$-内積比較) を示すが，k-party モデルへの拡張は容易である．秘匿 $(1,2)$-内積比較問題の問題定義を以下に示す．

Problem Statement 3. (秘匿 $(1,2)$-内積比較問題) *Alice* がベクター x_1, x_2 を，*Bob* がベクター y_1 を，*Carol* が y_2 を保持している．ただし $y = y_1 + y_2$，$x_1, x_2, y \in Z_K^d$ である．また不等式 I_- を $x_2 \cdot (y_1 + y_2) - x_1 \cdot (y_1 + y_2) < 0$，$I_+$ を $x_2 \cdot (y_1 + y_2) - x_1 \cdot (y_1 + y^r) \geqq 0$ とする．三人の間のプロトコルの実行の結果，*Alice* は不等式 $I \in \{I_+, I_-\}$ を得るがそれ以外は何も得ない．*Bob* と *Carol* は何も得ない．

秘匿 $(1,2)$-内積比較問題のためのプロトコルを以下に示す．

[Private (1,2)-Scalar Product Comparison]

- **Input of Alice:** Private vectors $x_1, x_2 \in \{1,...,K\}^d$
- **Input of Bob:** Private vector $y_1 \in \{1,...,K\}^d$
- **Input of Carol:** Private vector $y_2 \in \{1,...,K\}^d$
- **Output of Alice, Bob and Carol:** Inequation $I \in \{I_+, I_-\}$

(1) Bob: Generate a private and public key pair (s_k^1, p_k^1) for homomorphic encryption and send p_k^1 to Alice

(2) Bob: Encrypt vector $c_{1,i} = Enc_{p_k^1}(y_{1,i})$ and send it to Alice$(i = 1,...,d)$

(3) Carol: Generate a private and public key pair (s_k^2, p_k^2) for homomorphic encryption and send p_k^2 to Alice

(4) Carol: Encrypt vector $c_{2,i} = Enc_{p_k^2}(y_{2,i})$ and send it to Alice$(i = 1,...,d)$

(5) Alice: Generate random numbers r, r', r'', such that $d^6K^{12} < r$, $r < r' < (1 + \frac{1}{dK^2})r$, $0 < r_1'' < dK^2$, $0 < r_2'' < dK^2$, $x_m \leftarrow rx_2 - r'x_1$

(6) Alice: Compute $w_m^1 = \left(\prod_{i=1}^{d} c_{1,i}^{x_{m,i}}\right) \cdot Enc_{p_k^1}(r_1'')$ and send w_m to Bob

(7) Alice: Compute $w_m^2 = \left(\prod_{i=1}^{d} c_{2,i}^{x_{m,i}}\right) \cdot Enc_{p_k^2}(r_2'')$ and send w_m to Carol

(8) Bob: Compute $S^1 = D(w_m^1)$ and send S^1 to Carol

(9) Carol: Compute $S^2 = D(w_m^2)$. If $S^1 + S^2 < 0$, send I_- to Alice. Else, send I_+ to Alice.

(10) Alice: Receive an inequation I.

$S_1 + S_2 = rx_2 \cdot (y^1 + y^2) - r'x_1 \cdot (y^1 + y^2) + r_1'' + r_2''$ より，証明はここでは省略するが，プロトコルの正しさとセキュリティは [21] と同様の方法で示すことができる．$(1,1)$-秘匿内積比較との相違点は，Bob から Carol へ，$S^1 = rx_2 \cdot y_1 - rx_1 \cdot y_1 + r_1''$ が送信される点である．Carol は x_1 および x_2 について，S_2 と同様の理由で，S^1 からは何の情報も得ることができない．また異なる乱数 r_1'' および r_2'' がそれぞれ S^1 および S^2 において用いられているため，Carol は S^1 と S^2 の差から y_1 について情報を得ることができない．

この秘匿 $(1,k)$-内積比較に基づき，$(1,k)$-TSP のためのセキュアな局所探索プロトコルはただちに導かれる．

4.2 分散 $(k,1)$-TSP への拡張

Alice のベクターが $x_1 = \sum_{i=1}^{k} x_1^i, x_2 = \sum_{i=1}^{k} x_2^i$ のように分割され，Alice-1,...,Alice-k がそれぞれベクター x_1^i, x_2^i を保持している時，$\left(\sum_{i=1}^{k} x_1^i\right) \cdot y$ と $\left(\sum_{i=1}^{k} x_2^i\right) \cdot y$ を大小比較するプロトコルが考えられる．これを秘匿 $(k,1)$-内積比較と呼ぶ．

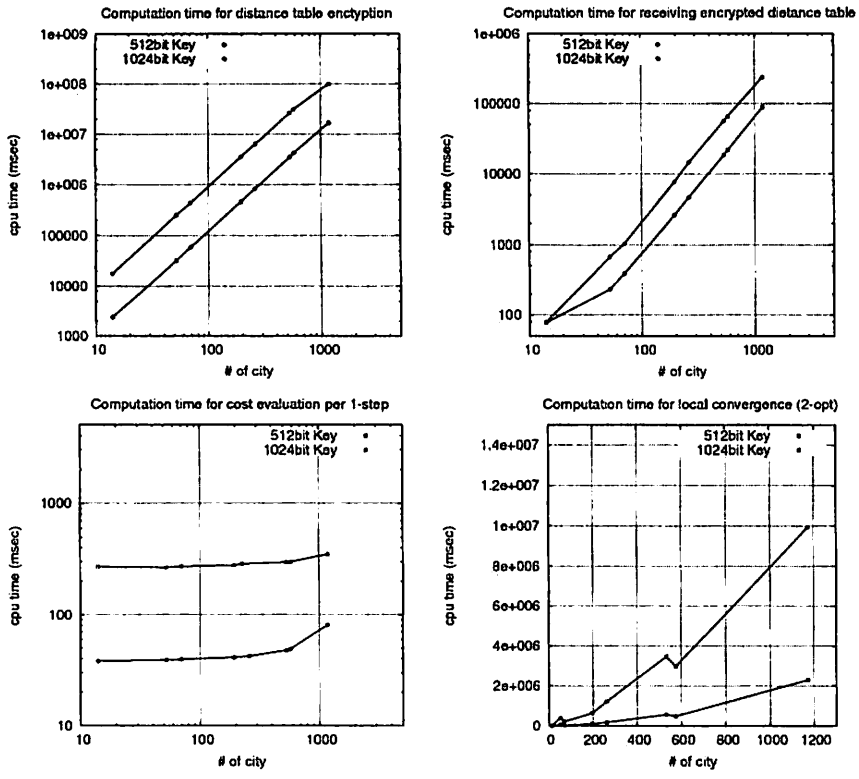

図 3　左上：都市数 vs Encryption (step 2)，右上：都市数 vs Downloading (step 2)，左下：都市数 vs 比較一回あたりに要する計算時間 (step 5-10)，右下：都市数 vs　最適化終了までに要する計算時間 (step 2-プロトコル終了)

Fig. 3　Top left : num. of city vs Encryption (step 2), Top right : num. of city vs downloading (step 2), Bottom left : num. of city vs comparison per one iteration (step 5-10), Bottom right : num. of city vs optimization (step 2-termination of the protocol)

秘匿 $(k,1)$-内積比較のためのプロトコルは，秘匿 $(1,k)$-内積比較と同様，$(1,1)$-内積比較の自然な拡張で得ることができる．

分散 $(k,1)$-TSP は，部分都市集合が $V' = V'_1 \cup V'_2 \cup ... \cup V'_k$ のように分割され，k party に保持されている分散 TSP である．分散 $(1,k)$-TSP のためのセキュアな局所探索は秘匿 $(1,k)$-内積比較からただちに導くことができたが，分散 $(k,1)$-TSP のためのセキュアな局所探索を秘匿 $(k,1)$-内積比較から導くことは困難である．なぜならば分散した部分都市集合 V'_i を互いに明かさずにランダム初期解や近傍解を構成することが困難だからである．分散 $(k,1)$-TSP のためのセキュアな最適化法の構築は，今後の課題である．

4.3　そのほかの組み合わせ最適化問題への拡張

本稿では，TSP のコスト関数を内積計算に帰着させることによって，セキュアな局所探索を秘匿内積比較プロトコルを用いて構成した．従って，提案プロトコルはコスト関数が内積計算によって計算されるならば，そのほかの組み合わせ最適化問題に適用可能である．

ナップザック問題 (KSP) は一つのナップザックの中になるべく多くの品物が入るように選択する問題である．各品物のコストを $c_1, ..., c_d$ とし，それぞれの品物の価値を $v_1, ..., v_d$ とする．KSP は $\sum_{j \in S} w_j \leqq W$ を満たし，価値の総和 $\sum_{j \in S} c_j$ が最大となるような部分集合 $S \subseteqq \{1, ..., d\}$ を発見する問題である．解 S は明らかに TSP と同様 $\{0,1\}^d$ で表現され，コスト関数および制約は内積によって評価されるため，提案プロトコルが利用可能である．

また配車計画問題 (VRP) は以下の様に記述される．ノード集合 V とコスト行列 C は TSP と同様に定義される．ノード v_0 は発着場であり，そのほかのノードは客の停留所である．VRP はコストが最小となるようなツアーの集合を発見する問題である．各ツアーは発着場から出発し，各ノードを正確に一度づつ訪問し，再び発着所に戻ってくる必要がある．配送中の荷物の積載量は車両の積載量 Q を超えてはならない．

ツアーの集合は TSP 同様ベクターで表現されるため，VRP においても提案プロトコルが利用可能である．

5. おわりに

本稿では，秘匿内積比較プロトコルに基づいて，セキュアな局所探索のためのプロトコルを提案した．また，分散 TSP の

データ分割モデルとプライバシーを定義し，分散 TSP が提案プロトコルによって最適化されることを示した．提案プロトコルは，準備段階において $O(n^2)$ の時間計算量を必要とするが，コスト一評価あたりの計算量は定数オーダーに抑えられているため，都市数が 1000 を越えても必要な待ち時間は比較的短いことを実験的に示した．

提案プロトコルでは最も単純な局所探索における実装例を示したが，二つの異なる解のコスト比較のみを利用して構成されるタブ探索や遺伝的アルゴリズムなど，より高性能のメタヒューリスティックへの拡張は容易である．より高度なメタヒューリスティックや分散 TSP 以外への拡張は，今後の課題である．

文　献

[1] Philip W. Blasmeier and Wendell J. Voisin, Supply Chain Management : A Time-Based Strategy, Industrial Management, September-October 1996, pp. 24-27, (1996).

[2] United Nations Economic Commission for Europe, http://www.unece.org/trade/untdid/welcome.htm

[3] Robert B. Handfield and Ernest L. Nichols, Introduction to Supply Chain Management, Prentice Hall; 1st edition (1998).

[4] Gregory Gutin and Abraham P. Punnen, The Traveling Salesman Problem and Its Variations, Kluwer Academic Publishers, (2002).

[5] O. Goldreich. Secure multi-party computation, working draft, (2001).

[6] Moni Naor and Kobbi Nissim, Communication preserving protocols for secure function evaluation, Proceedings of the thirty-third annual ACM symposium on Theory of computing, pp. 590-599, (2001).

[7] R. Agrawal and S. Ramakrishnan. Privacy-preserving data mining. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, pp. 439–450, 2000.

[8] Y. Lindell and B. Pinkas, Privacy preserving data mining. In Advances in Cryptology - CRYPTO '00, volume 1880 of Lecture Notes in Computer Science, pages 36–54. Springer-Verlag, 2000.

[9] Jaideep Vaidya and Chris Clifton. Privacy Preserving Association Rule Mining in Vertically Partitioned Data. In Proceedings of The 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages pp. 639-644, Edmonton, Alberta, Canada, (2002).

[10] Rakesh Agrawal, Alexandre Evfimievski and Ramakrishnan Srikant, Information sharing across private databases, Proceedings of the 2003 ACM SIGMOD, pp. 86-97, (2003).

[11] Hiranmayee Subramaniam, Rebecca N. Wright, Zhiqiang Yang: Experimental Analysis of Privacy-Preserving Statistics Computation. Secure Data Management, pp. 55-66, (2004).

[12] Jaideep Vaidya, Yu Michael Zhu, Christopher W. Clifton, Privacy Preserving Data Mining, Advances in Information Security, Vol. 19, Springer (2005).

[13] Pascal Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, EUROCRYPT 1999, pp223-238 (1999).

[14] M.-C. Silaghi and D. Mitra. Distributed constraint satisfaction and optimization with privacy enforcement. Intelligent Agent Technology, pp, 531-535, (2004).

[15] Koutarou Suzuki and Makoto Yokoo, Secure Generalized Vickrey Auction using Homomorphic Encryption, Seventh International Financial Cryptography Conference (FC-03) (2003).

[16] Michael J. Freedman, Kobbi Nissim, and Benny Pinkas. Efficient Private Matching and Set Intersection. In Christian Cachin and Jan Camenisch, editors, Advances in Cryptology , EUROCRYPT (2004).

[17] Wenliang Du and Mikhail J. Atallah. Privacy-preserving statistical analysis. In Proceeding of the Seventeenth Annual Computer Security Applications Conference, (2001).

[18] Bart Goethals, Sven Laur, Helger Lipmaa and Taneli Mielika"inen. On Private Scalar Product Computation for Privacy-Preserving Data Mining. The 7th Annual International Conference in Information Security and Cryptology (ICISC 2004), volume 3506 of Lecture Notes in Computer Science, pages 104–120 (2004).

[19] Pascal Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, EUROCRYPT 1999, pp223-238 (1999).

[20] http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

[21] 佐久間，小林，プライバシを保護した内積比較プロトコルの提案，電子情報通信学会 コンピュータセキュリティ研究会 (IPSJ-CSEC) (2006).