

<https://doi.org/10.1007/s40747-021-00401-1>

オリジナル記事



# 複数の戦略に基づいたハイブリッド・アントコロニーアルゴリズムによる時間窓付き車両ルーティング問題への対応

呉紅光（ウー・<sup>1,2</sup>ホンガン） - 高裕林（ガオ・ユエリン<sup>1,3</sup>） - 王旺（ワン・ウォンティン<sup>2,3</sup>） - 張子瑜（チャン・ツィーユ<sup>2,3</sup>

受け取りました。2021年1月9日 / Accepted 2021年5月13日  
© The Author(s) 2021

## アブストラクト

本論文では、時間窓を持つ車両ルーティング問題(TWVRP)を提案する。この問題では、車両が特定の時間枠内でしか顧客に対応できないという厳しい時間制約を考慮する。この問題を解決するために、ハイブリッド・アントコロニー(HACO)アルゴリズムを提案する。提案するHACOアルゴリズムには3つの工夫があり、1つ目はフェロモンを新しい方法で更新すること、2つ目は適応パラメータを導入すること、3つ目は突然変異演算を追加することである。提案したアルゴリズムの性能を評価するために、有名なソロモンのインスタンスを用いた。実験結果は、HACOアルゴリズムが時間窓を持つ車両ルーティングの問題を解決するのに有効であることを示している。さらに、提案アルゴリズムは、車両ルーティング問題に対しても実用的な意味を持ち、その結果、実用的な問題に適用でき、効果的であることが示された。

**キーワード** 車両ルーティング問題 - アントコロニーアルゴリズム - Mutation operation - Adaptive parameter - Practical application

## はじめに

21世紀は情報化時代であり、情報技術と科学研究が世界の発展を促進し、グローバル化のプロセスは深化し続けています。

本研究は、中国国家自然科学基金の助成金（11961001, 61561001）、寧夏高等教育一流科目建設プロジェクト（NXYLXK2017B09）、北民衆大学主要専有資金プロジェクト（ZDZX201901）の支援を受けています。

**B** Yuelin Gao  
gaoyuelin@263.net  
Hongguang Wu  
2440576496@qq.com  
王旺（おうおう）  
874812086@qq.com  
Ziyu Zhang  
2541604895@qq.com

<sup>1</sup> 北民衆大学数学・情報科学部、銀川750021、中国

<sup>2</sup> 寧夏省知能情報とデータ処理重点実験室、北民衆大学、銀川750021、中国  
<sup>3</sup> 北民衆大学、銀川750021、中国、寧夏省協力革新センター・オブ・サイエンティフィック・コンピューティング・アンド・インテリジェント・イン・フォーメーション・プロセッシング

経済の急速な発展は、国民の消費と切り離すことができないため、サービス産業の発展はますます加速しており、車両ルーティングの問題も提案されています。車両ルーティング問題(VRP)とは、一群の車両が異なる顧客のニーズを満たすために、経済的で効果的なルートを見つける必要があることを指す。この問題は、物流管理、車両スケジューリング、交通輸送などの分野で、NP困難な組み合わせ問題として有名である[1]。車両ルーティングの問題は、1959年にDantzigとRamserによって初めて言及されました。この問題が提起されたことで、VRPの多くのバリエーションが登場した。これらの拡張VRPもNP困難な組み合わせ問題に属します。

最も単純な経路探索問題は巡回セールスマン問題(TSP)であるが、輸送能力に制限を設けると、この問題は容量付き車両経路探索問題(CVRP)になる[2]。CVRPにおいて車両間の距離を制限すると、距離制約付きVRP、すなわちDVRPとなる[2]。DVRPにおいて、顧客の要求に応えるための時間を考慮すると、この問題は今回議論したいTWVRPになる[3]。

Ombukiら[7]は、TWVRP問題を解決するために、遺伝的アルゴリズムのパレオソート手法を用いて、TWVRPを多目的問題として提示した。

ベースラインデータとの比較により、提案アルゴリズムの有効性を検証した。Ghoseiriら[8]は、遺伝的アルゴリズムに進化的探索とパレオ技術を組み合わせて、TWVRPを解くための適応的な遺伝的アルゴリズムを開発した。彼らはTWVRPをゴールプログラミングに変換し、良い結果を得ました。同様のアルゴリズムは、混合ペナルティ関数に基づく微分進化統合アルゴリズム[9]や、Wangによって提案されたトポロジーに基づく微分進化アルゴリズム[10]のように、このような問題の解決に使用することもできます。

Hoら[11]はTWVRPを解く際にタブー探索アルゴリズムを用いており、実験の結果、タブー探索アルゴリズムによってこの問題を解くことができ、満足のいく結果が得られたとしている。Belhaizaら[12]は、TWVRPを解くために、タブー探索アルゴリズムと可変ネイバーフッド法を組み合わせたアルゴリズムを提案した。このアルゴリズムの利点は、経路生成の過程で顧客の最小待ち時間と最小遅れを記録し、それに応じて到着時刻と出発時刻を調整することができる点である。Wangら[13]は、TWVRPを解くための閾値タブー探索アルゴリズムを提案し、配車担当者が現実的な問題に効果的かつ迅速に対応できるようにした。

Chiangら[14]は、TWVRPを解くためにハイブリッドシミュレーテッドアニーリングアルゴリズムを提案し、短期記憶機能のアニーリングプロセスを増やすためにタブーテーブルと組み合わせました。数値実験の結果、この手法は従来の研究結果よりも優れていることがわかった。Dengら[15]は、シミュレーテッドアニーリングアルゴリズムに記憶機能を加えて改良し、二重終端原理を採用してTWVRPを解いた。実験結果からも、新アルゴリズムがより良い解を得られることが確認されています。Wangら[16]は、VRPSPDTW (Pick up and Delivery at the same time within a specific time window) を解くために、残余容量と半径方向の追加挿入に基づいて改良された並列シミュレーテッド・アニーリング・アルゴリズムを使用し、混合整数計画モデルを用いて目標を小型化した。

Yuら[17]は、時間窓を持つ期間車両ルーティング問題 (PTWVRP) を解く際に、フェロモンマトリクス法を用いてヒューリスティックな情報を蓄積してACOを求め、アルゴリズムにダブルクロスオーバー演算を導入してアルゴリズムの性能を向上させている。Dingら[18]は、災害演算子を導入し、フェロモンを調整することで、ハイブリッド・アントコロニー最適化アルゴリズムを提案した。Wangら[19]は、TWVR

Pを多目的問題として扱い、アントコロニーアルゴリズムとシミュレーテッドアニーリングを組み合わせたハイブリッドアントコロニー最適化アルゴリズムを提案した。改良されたアルゴリズムの性能は大幅に向上しました。また、粒子群最適化アルゴリズム [20-23]、ビーコロニーアルゴリズム [24-26]、クジラアルゴリズム [24-26]、アリのコロニーアルゴリズム [24-26]、アリコロニーアルゴリズム [24-26]などのアルゴリズムも、ベヒクルルーティング問題の解決に使用することができます。

最適化アルゴリズム[27]、ダイクストラ・アルゴリズム[28]などがあります。

アントコロニー最適化(ACO)は、組合せ最適化問題を解くための有効なヒューリスティックアルゴリズムであり、VRPを解くための新たなアイデアを提供する。Clolmiら[29]は、実在のアリに基づいて提案されたアントコロニーアルゴリズムを提案し、巡回セールスマン問題への適用に成功した。また、Bulleneimerら[30]は、最初にアリシステムを提案し、そのアルゴリズムを改良してVRPに適用している。実世界の複雑さと広さのため、基本的なアリ・コロニー・アルゴリズムでは実用的な問題を解決するには限界がある。そのため、多くの研究者がアントコロニーアルゴリズムを改良し、適用範囲を拡大するための新しい手法や新しいアイデアを提案している。Yuら[31]は、フェロモンの更新モードを変更し、IACOを得るために突然変異操作を導入した。この改良型アルゴリズムは、局所最適化を避けることができるが、アルゴリズムの複雑さが大幅に増している。Mavrovouniotisら[32]は、動的巡回セールスマン問題を解くために、アントコロニーアルゴリズムとローカルサーチアルゴリズムを組み合わせた。このアルゴリズムは解答の質を向上させるが、動的なインスタンスへの適用効果は低い。Dingら[18]は、ハイブリッド・アントコロニーアルゴリズムを提案し、災害演算子を導入した。彼らは提案したアルゴリズムの収束速度を研究したが、アルゴリズムの複雑さについては考慮していない。Wangら[33]は、災害時の緊急輸送モデルにハイブリッド・アントコロニーアルゴリズムを適用した。Jabirら[34]は、アントコロニーアルゴリズムと可変近傍探索アルゴリズムを組み合わせ、多段グリーンベヒクルルーティング問題(MDGVPR)に適用した。Liら[35]は、改良型のアントコロニーアルゴリズムをマルチターゲットベヒクルルーティング問題(MVRP)に適用した。Liら[35]は、比較的単純なモデルを提案しており、性能評価の際には提案アルゴリズムと基本アルゴリズムの比較のみを行っている。

そこで本稿では、TWVRPを解くための複数の戦略に基づいたHACOアルゴリズム(HACO)を提案する。その主な内容は、フェロモンの更新方法の改善、フェロモンの揮発性因子の適応的な調整、スワップ演算子とインサート演算子の導入により、アルゴリズムが局所最適に陥ることを回避し、アルゴリズムの収束率を向上させることである。本論文の枠組みは、6つの部分に分かれている。第1部分は導入部、第2部分は問題のモデル化、第3部分は問題の解決、

第4部分は数値実験と結果の分析、第5部分は応用例、第6部分は結論である。

## 問題の説明と策定

TWVRPでは、車両グループがデポを出発して、顧客サービスの要求を満たすために、以下のことが求められます。



の時間間隔である。これをもとに、最短距離の効率的なルートを探します。TWVRPは物流の現場で広く使われています。

agementと、輸送に関する最も一般的な問題システムを使用しています。

グラフ理論を用いて、TWVRPを有向完全グラフ  $G = (V, E)$

[37]としてモデル化することができ、その頂点セット

は  $V = \{0, 1, 2, \dots, n\}$  と、エッジセット  $E = \{(i, j) \mid i, j \in V\}$  である。

TWVRPでは、各頂点がノード（またはお客様）を表しており、各エッジは2つのノード間の距離を表しています。この

問題、 $M = \{1, 2, \dots, m\}$ は車

$$x_{i0k} = 0, \forall k \in \{1, 2, \dots, m\} \quad (6)$$

ここでは、デポを頂点  $0$

とし、顧客をノードとして紹介する。車両  $k \in M$

は、デポから出発し、ある数のノードにサービスを

提供した後、デポに戻る必要がある。このとき、要

求されたノードに対しては、1台の車両のみがサービ

スを提供することができ、車両の最大容量を超える

ことはできない。

の車両です。また、この車両は以下の条件を満たす必

要があります。

時間間隔  $a, b$ ; それは左の時間窓より早くない

$a$  または適切なタイムウィンドウ  $b$  よりも遅くなります

。

この論文では、車両のキャパシティに対する制約を考

えます。

車の使用量と移動距離

$2, \dots, n\}$  (9)

and minimize the costs associated with it.

以下の説明の便宜上、以下のように定義します。

$a_i$ : ノード  $i$  の左タイムウィンドウ  $b_i$ :

ノード  $i$  の右タイムウィンドウ  $c_{ij}$ :

ノード  $i$  とノード  $j$  の距離  $Cap_k$ :

車両  $k$  の容量

$d_i$ : ノード  $i$  の需要

$g_k$ : 車両  $k$  を使用する際に発生する固定費

$h_k$ : 車両  $k$  の単位距離当たりの輸送コスト

$t_{ij}$ : ノード  $i$  からノード  $j$  までの移動に要する時間

$s_i$ : ノード  $i$  のサービスタイム

$r_i$ : 車両がノード  $i$  に到着した時刻

決定変数

$x_{ijk}$ : 車両  $k$  がノード  $i$  からノード  $j$  まで移動する場合

$0$ , それ以外は

$1$

$x_{ijk}$

$0$ , それ以外は

$1$

$x_{ijk}$

$0$ , それ以外は

$1$

$x_{ijk}$

$0$ , それ以外は

$1$

$x_{ijk}$

$$\min \quad \sum_{k=1}^m \sum_{i=0}^n \sum_{j=1}^n h_k c_{ij} x_{ijk} + \sum_{k=1}^m g_k x_{0k} \quad (3)$$

$$s.t. \quad \sum_{k=1}^m d_i y_{ik} \leq Cap_k, \forall k \in \{1, 2, \dots, m\} \quad (4)$$

$$y_{ik} \leq 1, \forall i \in \{1, 2, \dots, n\} \quad (5)$$

$$x_{ijk} = y_{ik}, \forall i \in \{1, 2, \dots, n\} \forall k \in \{1, 2, \dots, m\} \quad (7)$$

$$x_{ijk} = y_{ik}, \forall i \in \{1, 2, \dots, n\} \forall k \in \{1, 2, \dots, m\} \quad (8)$$

$$x_{ijk} = y_{ik}, \forall i \in \{1, 2, \dots, n\} \forall k \in \{1, 2, \dots, m\} \quad (8)$$

$$x_{ijk} (t_{ij} + s_i + r_i) = r_j, \forall k \in \{1, 2, \dots, m\} \quad (10)$$

上記のモデルでは、目的(3)は車両のコストを最小化することです。制約条件(4)は、ノードの総需要が車両の能力を超えないことを保証するものである。制約条件

(5)は、各ノードが1台の車両しか使用できないことを保証します。制約(6)は、ノードにサービスを提供する際に、車両がデポを開始し、終了することを保証する。制約条件(7)と(8)は、各ノードが常に1台の車両を持っていることを保証する。制約条件

制約条件(9)は、サービス車両が時間内に到着することを保証する。制約条件(10)は、サービス時間と移動時間を経て、ノード  $i$  からノード  $j$  に最終的に車両が到着することを保証する。

$r$  1、  
 $y$  ノード  $i$  が車両  $k$  でサービスを受けている  
 場合  
 0、それ以外は

目的：最小コスト  
 制約条件

総需要が車両の能力を超えることはありません。  
 各ノードには、サービスを提供する車両があり、1  
 台の車両でしかサービスを提供できません。  
 車両は、時間間隔内にノードにサービスを提供し  
 なければならない。  
 車両はデポを出発し、最後にデポに戻ります。  
 以上のように定義された目的、パラメータ、およ  
 び決定変数によれば、問題は次のように表されます。

## (1) 解決方法

(2)

アリ・コロニー・アルゴリズムは、自然界における  
 アリの採餌行動に基づいて進化したものです。蟻は  
 餌を探す過程で、通過した道に沿って分泌物を残し  
 ます。アリのコロニーは、自分の巣からの最短経路  
 を常に見つけることができる  
 は、これらの分泌物を介して食物源にアクセスしま  
 す。Dorigo[38-  
 40]は、アリのコロニーの採餌行動にヒントを得て、  
 ACOの概念を提案しました。ACOのアイデアは、ア  
 リが残した分泌物を使って情報を交換し、問題に必  
 要な最適経路を見つけるというものである。この分  
 泌物はフェロモンと呼ばれています。アリは、フェ  
 ロモンの分布をもとに次の行動を決定します。ある  
 経路にフェロモンが多く存在する場合、アリがその  
 経路を選択する確率は



が高くなります。アリがフェロモンを残す量が少なければ、アリがこの道を選ぶ確率は低くなります。フェロモンの量が増えれば増えるほど、より多くのアリが選択することになり、アリは餌を探すための最短の道を見つけることができるのです。ACOはヒューリスティックなアルゴリズムとして、様々な最適化問題の解決によく使われており、特にVRPやTSPなどの拡張問題で実用化されています。

ACOの核心は、アリが次に選択するノードを決定し、より早く最短経路を見つけるためにアリをどのように導くかというフェロモンの更新式を設計することである。本論文におけるアントコロニーアルゴリズムの改良は、Dorigo (1992) [36]やDing et al. (2012) [18]を参考にしており、主にフェロモンの更新式の改良と適応パラメータの追加を行っています。フェロモン更新式のパラメータ $\rho$ を定数から動的数値に変更することで、バッドパスの持続時間を短くしている。

そして、最適化プロセスにおけるグッドパスの役割を高めていきます。

## ACOの改善について

HACOを使ってTWVRPを解く場合、各アリはデポから制約条件を満たすパスを構築する必要があります。もし、現在の経路で次に選択されたノードが制約条件を満たさないことが判明した場合、アリはデポに戻り、残りのノードで新しい経路を再構築する必要があります。このプロセスはすべてのノードが選択されるまで続く。各アリは一定の確率で次のノードを選択します。したがって、ノード $i$ が次のノード $j$ を選択するための式は

$$P_{ij}^k = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{j \in N_k^i} \tau_{ij}^\alpha \eta_{ij}^\beta} \quad (11)$$

式 (11) において、 $N_k^i$ はノード $i$ を離れた後にアクセスできるノードの集合 $k$ を表し、 $\tau_{ij}$ は現在のノード $i$ と可能性のあるノード $j$ との間の経路上のフェロモンの数を表し、 $\eta_{ij}$ は強度を表し、その式は $\eta_{ij} = \frac{1}{width_{ib} - width_{ia}}$ であり、ノード $i$ と可能性のあるノード $j$ との間の距離の逆数であり、 $width_{ib}$ は時間窓の幅またはノードの要求の緊急度を表し、以下のように表される。 $width_{ib} = a - b$ 、 $a$ はアリの軌道決定するフェロモン因子、 $b$ は期待値因子であり、外部情報の軌跡に対する重要性を反映している。 $\gamma$ はサービスノードの軌跡選択に対する緊急性の重要性を表している。選択されたノード $j$ は、緊

$q_0$ は、探索進行中のアリの探索に対する搾取を制御するパラメータである。

アリが次に行きたいノードを決定するとき、そして方向を決定することに相当する。仲間の情報を伝えるためにフェロモンを道に残します。前者が残したフェロモンは、後のアリが選ぶ道に影響を与える、つまりフェロモンの濃度が道の選択を決めるのです。ACOの原理によれば、経路が短いほどフェロモン濃度は高く、経路が長いほどフェロモン濃度は低くなります。アリの探索プロセスを高速化するために、フェロモンの更新式を改良し、以下のようにしました。

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho \tau_{ij}(t) \quad (12)$$

$$\tau_{ij}(t) = \tau_{ij}(t) \quad (13)$$

$$\tau_{ij}^u = \begin{cases} \frac{Q}{L^u} & \text{蟻} u \text{ が経路}(i, j) \text{ を通過する場合} \\ 0 & \text{それ以外の場合} \end{cases} \quad (14)$$

上式において、 $\rho$ はフェロモンの揮発性係数、 $\tau^u$ は現在のサイクルにおけるアリ $u$ の経路上のフェロモンの数、 $Q$ はアリが堆積したフェロモンの量に関連する定数、 $L^u$ は現在のサイクルにおけるアリ $u$ の経路の長さを表す。

The pheromone volatility coefficient  $\rho$  is a dynamic value.  $\rho$ の値が小さければ、経路中のフェロモンの蒸発が遅くなり、経路上に残るフェロモン濃度が大きくなるため、後続のアリの経路選択に大きな影響を与え、その結果として $\rho$ の値が大きい場合、アリのランダム探索能力。

のフェロモンはすぐに蒸発してしまいますが、道に残っているフェロモンの濃度を小さくすることで、探索プロセスにおけるアリのグローバルな探索能力を高め、他の経路を探索しやすくする。アリの探索過程において、貧弱な経路によるアリの新経路探索への干渉を軽減するために、本稿ではパラメータ $\rho$ を適応的に調整しており、サブサイクルで最適解が大きく改善されない場合には、以下の式に従って値を調整している。

急性が高いほど、フェロモン濃度が高い

$$\rho = 0.90\rho_{\text{max}}$$

(15)

また、 $q_0$   $\leq$   $>$   
 $q$ のときに $0$ ノード $j$ を選択するのは、確率式 $P^k$ とルーレット法によるもので、 $q \in \text{rand}(0, 1)$ が乱数となる。

式(15)では、より小さな状況でノード選択の機会が得られるように、 $\rho_{min}$ はアリ探索プロセスで定義された最小値である。



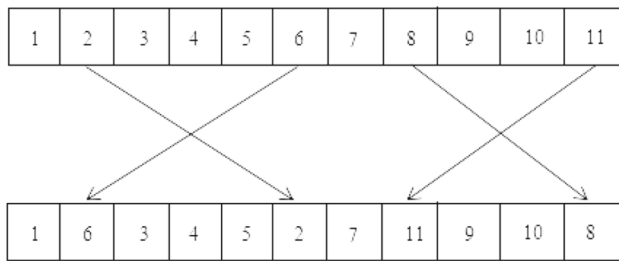


図1 スワップ演算子

ステップ1: パラメータの初期化 (反復NCの最大数、アリの数 $V$ 、制限容量 $Cap_k$ 、すべての $i \in I$ のノード $d$ の需要)。  
 $\{1, 2, \dots, n\}$ .

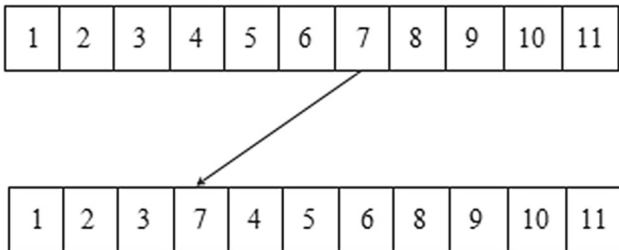


図2 インサート操作部

## 変異操作の紹介

本論文では、ACOにおける2種類の変異演算子 (swap演算子とinsert演算子) を紹介する。変異演算子を導入することで、最適解の多様性を高めるだけでなく、アルゴリズムの局所最適への収束を回避することができる。まず、本稿で使用する変異演算子について簡単に説明する。

### Swap

演算子。遺伝子配列内の遺伝子の位置をランダムに交換すること。図1では、遺伝子2と6、8と11の位置が入れ替わっている。

挿入オペレーター。遺伝子配列の中で、ある遺伝子が他の遺伝子にランダムに挿入される位置のこと。図2では、遺伝子7が削除され、遺伝子4の位置に挿入されている。

TWVRPの解答例を図3に示しますが、これを遺伝子配列に変換すると図4aようになります。遺伝子配列4(a)を入れ替えて、Fig.4bを得た。図4bで表される遺伝子配列を挿入して図4cを得た。ここで、0を遺伝子配列に挿入することで、制約条件を満たす解形成のパスを表す。

## ソリューションのステップ

TWVRPモデルをHACOで解く手順は以下の通りです

。

図3 TWVRP用ソリューションの一例

図4 シーケンスコード

ステップ2: アリ  $l$   $=$   $1, 2, \dots, v$  をデポに入れ、すべての未訪問ノードを記録するためのオプションテーブル  $Cand_l$   $1, 2, \dots, n$  を作成する。アリがノードを通過した後、対応するパスレコードテーブル  $tabu$   $l =$   $\emptyset$  にノードを記録する必要がある。

Step3: テーブル  $Cand$  から未訪問のノードを探す  $1, 2, \dots, n$  で、式(11)に従って、次にアクセスするノード  $j$  を選択する。

ステップ4: アリ  $l$  がノード  $j$  の車両容量、時間、ノード需要の制約を満たしているかどうか、制約を満たしている場合はステップ5へ、そうでない場合はステップ3へ。

ステップ5: 選択されたノード  $j$  を table  $tabu$  に入れ、現在の結果を保存し、table  $tabu$  を更新します。

Step6: アリ  $l$  が全てのノードを横断したかどうか、そうであればStep7へ、そうでなければStep2へ。

ステップ7: アリが移動した経路の長さを計算し、今回見つかった経路に変異操作を行う。変異操作後の経路が制約条件を満たしていれば、最適な経路を記録し、経路記録表を更新する。

ステップ8: 式 (12) に従って、フェロモンを更新する。

0	1	2	3	0	4	5	6	0	7	8	9	0	10	11	0
---	---	---	---	---	---	---	---	---	---	---	---	---	----	----	---

(a) The gene sequence transformed by Figure 3

0	1	6	3	0	4	5	2	0	7	11	9	0	10	8	0
---	---	---	---	---	---	---	---	---	---	----	---	---	----	---	---

(b) The swap of gene sequences

0	1	6	3	0	7	4	5	0	2	11	9	0	10	8	0
---	---	---	---	---	---	---	---	---	---	----	---	---	----	---	---

(c) The insert of gene sequence

$=$   
 $\{$

ステップ9：解答空間を包括的に探索するために、パラメータ $p$ を式(15)に従って適応的に調整する必要があります。

ステップ10現在のサイクル番号が最大反復番号より大きい。そうであれば、サイクルを停止して結果を出力し、そうでなければステップ2に進みます。

## 数値解析

提案するHACOの性能を検証するために、SolomonのTWVRP標準インスタンスに適用しました。このインスタンスには56個のデータセットがあり、各データセットには100個のデータが含まれ、データはSolomonのウェブサイトから入手できます。<sup>1</sup>

ソロモン標準インスタンスは、6つのカテゴリーに分けられます。C1、C2、R1、R2、RC1、RC2です。ソロモン標準インスタンスは、顧客の地理的分布に基づいて3つのタイプに分けることができ、第1のタイプは、この2つのセットのインスタンス顧客が地理的に集中しているため、セットC1とC2であり、第2のタイプは、この2つのセットのインスタンス顧客が地理的にランダムに分散しているため、セットR1とR2であり、比較的離散的であり、第3のタイプは、集中と分散の2つのクラスの間にあるセットRC1とRC2である。ソロモン標準インスタンスは、車両容量とフリートサイズ、および時間窓によって、さらに2つのカテゴリーに分けられる。第1のカテゴリーはC1、R1、RC1で、これらの3つのセットは、車両容量が小さく、フリートサイズが小さく、時間窓が狭いため、車両で提供される顧客数が少ない。第2のカテゴリーはC2、R2、RC2で、これらの3つのセットは、第1のカテゴリーの3つのセットとは反対であるため、車両で提供される顧客数が多い。数値実験では、標準的なインスタンスを上記の第2の方法で分類し、2つのカテゴリーに分けてアルゴリズムのテストと評価を行った。

## パラメータ設定

ACOがTWVRPの解を求める際、アルゴリズムのパラメータが解の質に影響を与えます。パラメータ値の選択には通常2つの方法があります[41]。一つ目の方法は、アルゴリズムの実行前にパラメータを設定し、実行中にパラメータが変化しないようにする方法である。第2の方法は、アルゴリズムの実行中にパラメータを設定する方法であるが、実行前に

初期値を設定する必要がある、そのようなパラメータは実行中に常に変化する。ここでは、C101、R101、RC01の3組のインスタンスを選択してパラメータを解析します。一つのパラメータを解析すると、他のパラメータは変化しませんでした。以下、HACOを用いた数値解析と、それに基づくパラメータの検討を行います。

<sup>1</sup> <http://w.cba.neu.edu/solomon/problems.htm>.

$g$   $k500$   $h_k$   
 1000。最適解を分析することで、パラメータが解の質に与える影響を説明し、次にパラメータがアルゴリズムの全体的なパフォーマンスに与える影響を説明します。

#### 1. パラメータ $\alpha$

図5から、異なるパラメータ $\alpha$ を用いて得られた近似最適目的値を見ることができます。パラメータ $\alpha$ の値が1前後のとき、良い値が得られている。したがって、実験では、パラメータ $\alpha$ の値として得られる最良の値は1に等しい。

#### 2. パラメータ $\beta$

図6からわかるように、パラメータ $\beta$ の値が区間[3,5]の間にあるとき、比較的良好な値が得られる。したがって、パラメータ $\beta$ の値の範囲は[3,5]となります。

#### 3. パラメータ $\gamma$

図7から分かるように、パラメータ $\gamma$ の値が区間[2,4]の間にある場合、得られる近似最適目的値は比較的良好である。したがって、パラメータ $\gamma$ の値の範囲は[2,4]とする。

#### 4. パラメータ $q_0$

図8からわかるように、パラメータ $q_0$ の値が区間[0.3, 0.6]の間にあるとき、より良い近似最適目的値を得ることができる。したがって、パラメータ $q_0$ の値の範囲は[0.3, 0.6]とする。

#### 5. パラメータ $\rho$

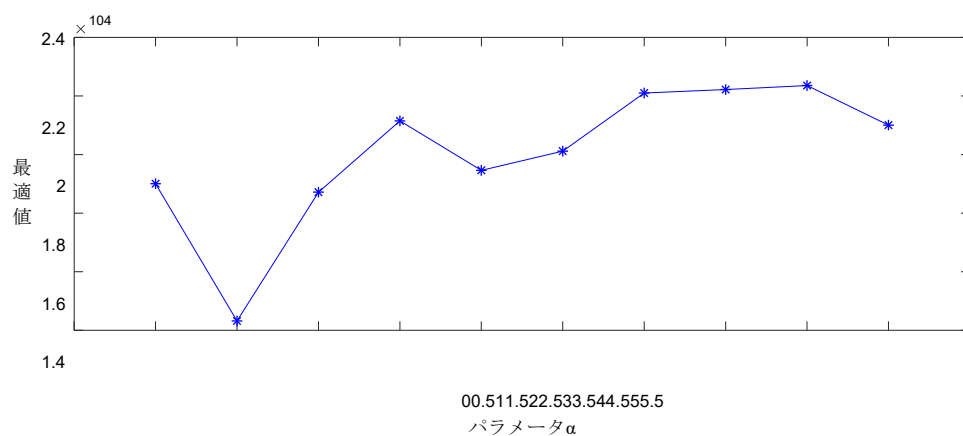
図9からわかるように、パラメータ $\rho$ の値が区間[0.3, 0.7]の間にあるとき、比較的良好な近似最適目的値を得ることができます。 $\rho$ の値が小さくなると、経路上にフェロモンが多く残ってしまうため、より良い経路を見分けることができず、最適解を得ることが難しくなります。しかし、 $\rho$ を大きくすると、悪い経路にフェロモンが少なくなるため、アルゴリズムが局所最適に収束する可能性があります。そのため、パラメータ $\rho$ の値の範囲は[0.3, 0.7]としています。

ソロモン標準インスタンスは、HACOの性能をテストし、最適解の品質を評価するためのテスト関数として使用されます。表1と表2にACOとHACOの結果を示す。

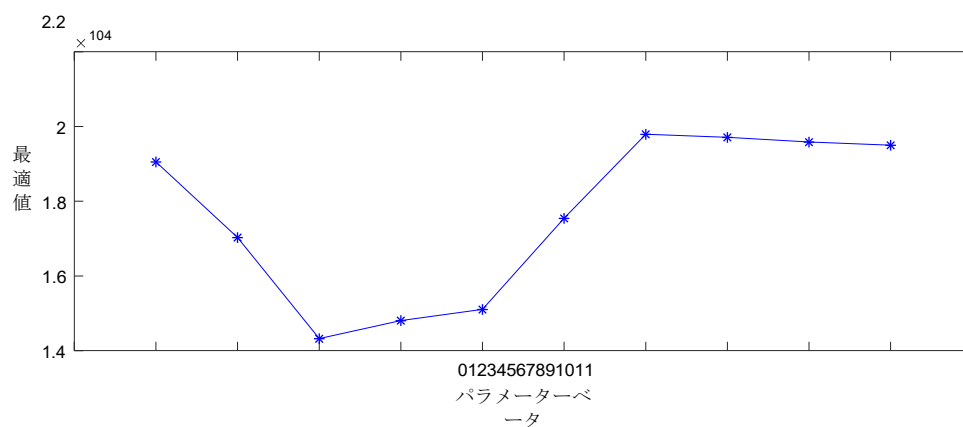
## 実験結果

HACOの長所と短所を説明するために、実験結果を2つの側面から分析します。1つ目は、HACOによって得られた結果をACOと比較し、2つ目は、HACOによって得られた結果を他の知的アルゴリズムと比較します。

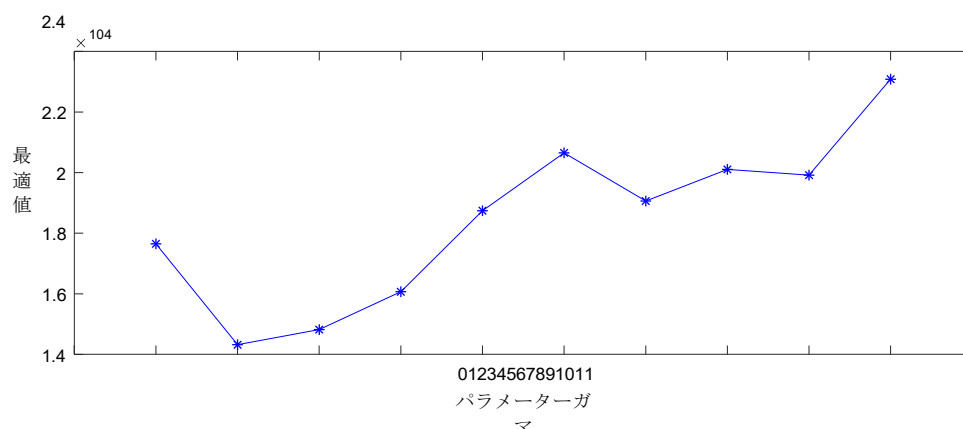
**図5**  
 $\alpha$ を変えた場合の近似最適値



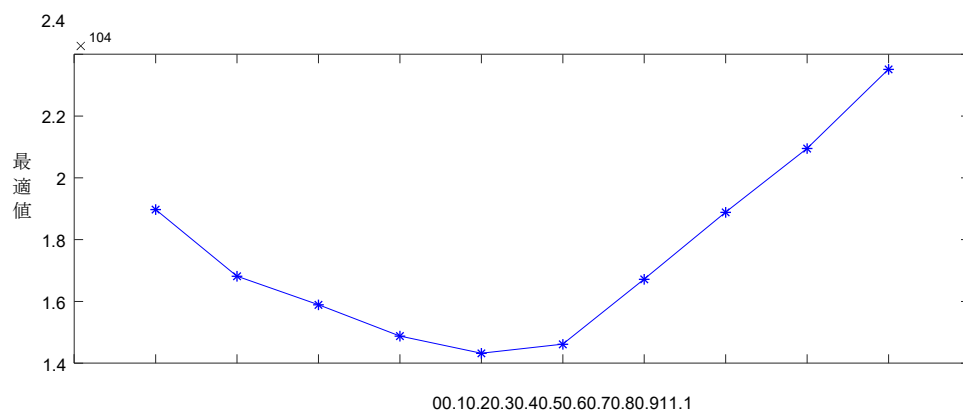
**図6**  
 $\beta$ を変えた場合の近似最適値



**図7**  
 $\gamma$ を変えた場合の近似的な最適値



**図8**  
異なる  $q_0$  での近似的な最適値



ラ  
メ  
ー  
タ  
q  
0



図9  
 $\rho$ を変えた場合の近似的な  
最適値

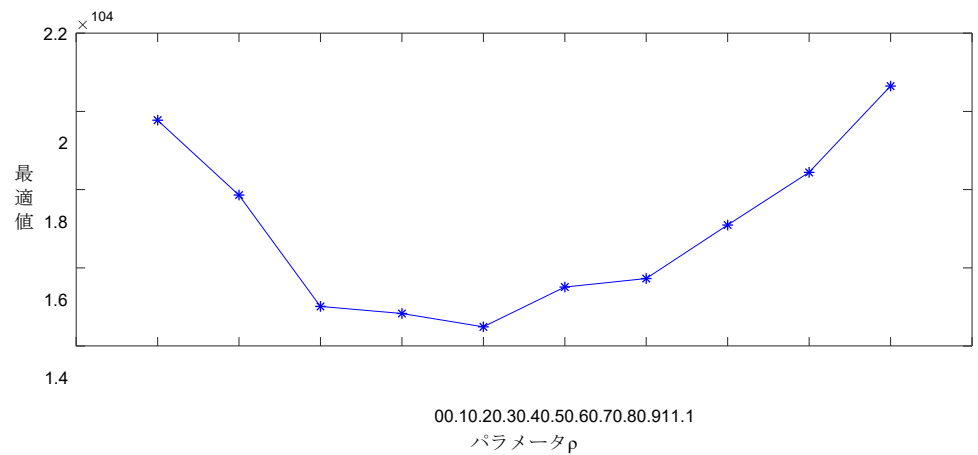


表1  
タイプ1 (C1、R1、RC1)  
の 카테고리におけるACO  
との比較

インスタンス	ACO		HACO			
	10回目の出場で最高の成績		10回目の出場で最高の成績			
	<i>V ehDis</i>	コスト	<i>V ehDis</i>	ギャップ(%)	コスト	ギャップコスト(%)
C101	C102131538.7319.673131693.119.1214627.	131299.02 19.798	131262.53	-2.89	14320	-38.25
	-9.0812549	-20.27	C103111669.	3815093111530.39		
	C104101278.6012280101307.092.1811382			-7.89		
	-2.1312287	-35.52	C105121271.	4816651111244.97		
	-4.2414496	-22.79	C106131522.	9417800131460.96		
	C107131367.3117781131377.250.7213521			-31.51		
	-6.8613447	-8.34	C108121399.	5114568121309.69		
	-6.3013671	-14.06	C109111275.	4515593111199.90		
	-1.4934008	-2.33	R101262588.	9434802262550.81		
	-1.1228536	-3.62	R102232370.	2329570232343.94		
	-6.7621064	-0.74	R103171973.	6121220161848.63		
			R104131404.6715610131398.59			
	-0.43165605.74		R105191966.7824318191870.37			
	-5.15253163.94		R106161859.	2022026161843.72		
	-0.8421038	-4.70	R107131647.	1816676131630.95		
	-1.0015791	-5.60	R108121317.	3014459121285.82		
	-2.4514447	-0.08				

RC109151760.	2118879151696.26	105192571.	-37924890392362.44	-0.93
R1108.8623536	4316700341566.01		-0.6015615	-6.99
R111141512.	7316716141510.99		-0.1216656	-0.36
RC106151917.75	20196151935.82	0.93202210.12		
R112121324.	9014433121272.96		-4.0813390	-7.79
RC101212444.	2125654212350.94	RG107141752.	-8317928141733.07	-8.02
RCT02162142.	0521450162132.71		-0.4421379	-0.33
RC103131802.	4818049131791.28	RG108131589.	-0.6315764131567.73	-5.71
RCT041216357	4015794121643.07		-0.7514833	-6.48

表1と表2では、ACOとHACOが10回実行し、最も良い結果を選択しています。ここで、Vehは車両数、Disは車両の移動距離、Costは

は車両のコスト、 $Gap_{Dis}$  ( % ) はHACOで得られた距離とACOで得られた距離の差を表し、その式は

**表2**  
タイプ2 (C2、R2、RC2)  
の 카테고리におけるACO  
との比較

インスタンス		ACO		HACO			
		10回目の出場で最高の成績		10回目の出場で最高の成績			
		<i>V ehDis</i>	コスト	<i>V ehDis</i>	ギャップ(%)	コスト	ギャップコスト(%)
C201		3591.56	3603.92	3591.56	0	3603.92	0
C202		4982.80	5270.53	<b>4905.92</b>	<b>-8.49</b>	<b>5202.31</b>	<b>-1.31</b>
				C2033978.605152.844876.94			
		-11.595302.752.83					
C204	3	994.06		5171.	<b>553979.53</b>	<b>-1.485155.46</b>	<b>-0.31</b>
C205	4	674.47		4800.	<b>374669.25</b>	<b>-0.784688.43</b>	<b>-2.39</b>
C206	4	776.75		4851.	<b>464753.34</b>	<b>-3.113996.40</b>	
C207	4	756.17		<b>-21.40</b>			
C208	4	773.63		5065.	<b>524715.13</b>	<b>-5.744985.96</b>	<b>-1.60</b>
R201	4	2063.83		4896.	<b>774720.31</b>	<b>-7.404785.73</b>	<b>-2.32</b>
R202	4	2097.53		8151.	<b>8042018.20</b>	<b>-2.268100.00</b>	<b>-0.64</b>
R203	3	1665.79		8376.	<b>0041977.10</b>	<b>-6.097862.00</b>	<b>-6.54</b>
R204	3	1238.69					
R205	4	1495.27		5827.7031761.755.455846.800.33			
R206	3	1470.71		4444.	<b>2031195.32</b>	<b>-3.634371.20</b>	<b>-1.67</b>
R207	3	1339.77		5746.	<b>4041470.54</b>	<b>-1.685717.00</b>	<b>-0.51</b>
R208	3	1088.24		5487.	<b>5031455.13</b>	<b>-1.074693.30</b>	
R209	3	1501.85		<b>-16.92</b>			
R210	3	1508.14		4491.	<b>9031333.29</b>	<b>-0.494489.50</b>	<b>-0.05</b>
R211	3	1084.07		4226.	<b>1031044.05</b>	<b>-4.234222.50</b>	<b>-0.09</b>
RC201	5	2364.20		5607.	<b>4031440.30</b>	<b>-4.275602.70</b>	<b>-0.08</b>
RC202	4	2029.48					
RC203	4	1691.10		5627.8031561.413.415670.900.76			
RC204	3	1267.54					
RC205	5	2138.76		4346.3031189.928.904195.10		-3.60	
RC206	4	1729.86		8444.	<b>5652298.07</b>	<b>-2.888360.24</b>	<b>-1.01</b>
RC207	4	1656.32					
				7220.9342046.150.817467.313.30			
				5987.	<b>0241684.92</b>	<b>-0.375764.29</b>	<b>-3.86</b>
				4396.	<b>7831245.31</b>	<b>-1.794378.33</b>	<b>-0.42</b>
				7304.	<b>0452096.55</b>	<b>-2.017233.18</b>	<b>-0.98</b>
				5729.8641748.731.085814.341.45			
				5694.4741618.84			
				-2.325740.280.80			
				RC20831357.874597.		<b>8631306.25</b>	
		<b>-3.954431.72</b>	<b>-3.75</b>				

ギャップ (%)  

$$\frac{ODis}{HACO_{Dis}} = 100(\%).$$

$$\frac{HACO_{Dis} - AC}{HACO_{Dis}} \times 100(\%)$$
 (16)  
 ギャップ<sub>Cost</sub> (%)  
 は、HACOのコストとACOのコストの差を表しており、その式は

ギャップ (%)  

$$\frac{OCost}{HACO_{Cost}} = 100(\%).$$

$$\frac{HACO_{Cost} - AC}{HACO_{Cost}} \times 100(\%)$$
 (17)

表1に示すように、車両数が同じ場合、HACOの結果はACOの結果よりも優れており、距離を短縮し、コ

コストを削減することができる。しかし、表1の中には効果の悪いインスタンスが6つあり、インスタンスC102、C104、C107はコストを減らしながら走行距離を伸ばし、インスタンスR104、R105は距離を減らしながらコストを増やし、RC106は走行距離を伸ばすだけでなくコストも増やしています。うまくいかなかった6つのインスタンスを除いて、他のインスタンスは大幅に良い結果となりました。

また、表2から、ACOと比較して、HACOはほとんどのインスタンスで効果が高く、走行距離を減らすだけでなく、コストも減らすことができることがわかる。同様に、表7には効果の低い7つのインスタンスがあり、その中には、インスタンスR203、R210、RC202、RC206の走行距離が増加し、コストも増加しているもの、インスタンスC203とRC207は距離を短くする一方でコストを増加させているもの、インスタンスR211はコストを削減するが、距離を増加させているものがある。

### 他の知的アルゴリズムとの比較（ノード番号は同じ）

ソロモンの標準的な各インスタンスには、地理的に異なる分布をしている100人の顧客がいます。表3と表4では、顧客数が同じ場合、HACOと他の知的アルゴリズムを比較しています。太字は

**表3**  
タイプ1 (C1、R1、RC1) の  
比較結果

インスタ ンス	SA			GA			HACO		
	ビー	ディス	コスト	ビー	ディス	コスト	ビー	ディス	コスト
C101	15	2135.20	25210.00	14	1901.83	21140.00	13	<b>1262.53</b>	<b>14320.00</b>
C102	13	1812.56	21369.00	13	1789.27	19953.00	13	<b>1693.11</b>	<b>14627.00</b>
C103	11	1901.99	19219.00	11	1666.86	16628.00	11	<b>1530.39</b>	<b>12549.00</b>
C104	10	1639.66	16596.00	10	1335.73	12578.00	10	<b>1307.09</b>	<b>11382.00</b>
C105	11	2161.45	23301.00	11	1311.57	13187.00	11	<b>1244.97</b>	<b>12287.00</b>
C106	13	1997.92	19630.00	13	1481.56	15022.00	13	<b>1460.96</b>	<b>14496.00</b>
C107	11	1182.75	12384.00	11	1052.57	12216.00	13	1377.25	13521.00
C108	12	1742.48	17873.00	12	1312.65	13597.00	12	<b>1309.69</b>	<b>13447.00</b>
C109	11	1718.28	17199.00	11	1214.12	14391.00	11	<b>1199.90</b>	<b>13671.00</b>
R101	20	2082.06	32544.00	22	1921.79	30201.00	26	2550.81	34008.00
R102	21	1989.32	26675.00	20	1711.83	26046.00	23	2343.94	28536.00
R103	15	1680.94	17273.00	18	1518.13	16734.00	16	1848.63	21064.00
R104	13	1537.25	18006.00	13	1410.95	17425.00	13	<b>1398.59</b>	<b>16560.00</b>
R105	17	1779.36	20360.00	16	1510.71	21858.00	19	1870.37	25316.00
R106	15	1764.23	18175.00	15	1468.54	17951.00	16	1843.72	21038.00
R107	13	1712.83	18390.00	13	1766.38	19241.00	13	<b>1630.95</b>	<b>15791.00</b>
R108	12	1447.08	14514.00	12	1309.60	14966.00	12	<b>1285.82</b>	<b>14447.00</b>
R109	13	1666.98	17247.00	14	1294.83	21040.00	15	1696.26	18705.00
R110	14	1645.49	16490.00	14	1573.89	16710.00	14	<b>1566.01</b>	<b>15615.00</b>
R111	13	1504.64	15646.00	13	1394.25	16043.00	14	1510.99	16656.00
R112	12	1375.96	13759.00	12	1317.88	14156.00	12	<b>1272.96</b>	<b>13390.00</b>
RC101	17	2322.98	22840.00	16	1738.53	18180.00	21	2350.94	23750.00
RC102	16	2173.43	26259.00	16	2389.70	25786.00	16	<b>2132.71</b>	<b>21379.00</b>
RC103	13	1854.61	20733.00	13	1856.08	20253.00	13	<b>1791.28</b>	<b>17074.00</b>
RC104	14	1672.42	17354.00	12	1702.10	16569.00	12	<b>1643.07</b>	<b>14833.00</b>
RC105	18	2134.18	21365.00	17	1783.33	22928.00	19	2362.44	23636.00
RC106	14	1846.32	25178.00	15	1635.40	21140.00	15	1935.82	20221.00
RC107	14	1788.89	17953.00	14	1762.27	18133.00	14	<b>1733.07</b>	<b>16913.00</b>
RC108	13	1678.36	17042.00	13	1574.27	16503.00	13	<b>1567.73</b>	<b>15677.00</b>

は、HACOが比較アルゴリズムよりも良い結果を得ていることを示しています。

表3、4では比較アルゴリズムとしてタブー・サーチ・アルゴリズムと遺伝的アルゴリズムを使用しています。Veh、Dis、Costは上記の意味と一致しています。表3によると、顧客数が同じ場合、HACOは比較アルゴリズムよりも62.07%良い結果が得られています。中でも、顧客が地理的に集中している場合（セットC1）、HACOは比較アルゴリズムよりも88.89%良い結果を得ている。地理的に顧客の分布が分散している場合（セットR1）、HACOは41.67%の良い結果しか得られていない。また、顧客が地理的に分散している場合と集中している場合では、HACOは62.50%の良い結果となり、実現可能である。同様に、表4によると、顧客数が同じ場合、HACOは比較アルゴリズム

よりも60.71%良い結果となっている。その中で、HACOは

顧客が比較的集中している場合（セット C2）には、比較アルゴリズムよりも 75% 良い結果が得られます。顧客が分散している場合（セット R2）には、HACO は 54.55% 良い結果しか得られず、比較アルゴリズムで得られた結果の方が良い結果となります。顧客が集中している場合と分散している場合では、HACO は比較アルゴリズムよりも 62.50% 良い結果となります。

### 他の知的アルゴリズムとの比較（お客様のニーズは同じです）

ソロモンの標準的なインスタンスでは、各インスタンスには、ニーズの異なる100人の顧客がおり、ある人はより多く、ある人はより少なくなっている。各インスタンスにおいて、同じまたは類似した顧客の要求を分類し、顧客の要求の観点から提案アルゴリズムの性能と収束性を評価する。の性能を十分に分析するために



**表4**  
タイプ2 (C2,R2,RC2) の比較  
結果

	インスタンス			GA			HACO		
	ビー	ディス	コスト	ビー	ディス	コスト	ビー	ディス	コスト
C201	3	1395.73	4212.00	3	888.34	4382.00	3	<b>591.56</b>	<b>3603.92</b>
C202	3	1580.84	6150.00	4	958.11	5483.00	4	<b>905.92</b>	<b>5202.31</b>
C203	4	1294.96	4967.00	4	965.19	3684.00	4	876.94	5302.75
C204	3	1282.14	5947.00	3	1028.03	5851.00	3	<b>979.53</b>	<b>5155.46</b>
C205	4	1127.15	6029.00	3	705.49	4994.00	4	<b>669.25</b>	<b>4688.43</b>
C206	3	1341.12	4702.00	3	760.77	4087.00	3	<b>753.34</b>	<b>3996.40</b>
C207	3	1462.22	5175.00	4	1028.50	5092.00	4	<b>715.31</b>	<b>4985.96</b>
C208	4	1372.35	4576.00	4	741.75	2053.00	4	720.31	4785.73
R201	4	2045.47	8873.00	4	1218.40	8819.00	4	2018.20	8100.00
R202	4	1238.68	2940.00	4	1151.54	3429.00	4	1977.10	7862.00
R203	3	1578.86	5788.00	6	881.41	5158.00	3	1761.75	5846.80
R204	3	1267.85	4714.00	5	1218.74	4484.00	3	<b>1195.32</b>	<b>4371.00</b>
R205	4	1542.28	7088.00	7	1475.60	6495.00	4	<b>1470.54</b>	<b>5717.00</b>
R206	3	1455.18	5417.00	4	1509.65	5150.00	3	<b>1455.13</b>	<b>4693.30</b>
R207	3	1344.84	5156.00	5	1348.00	5630.00	3	<b>1333.29</b>	<b>4489.50</b>
R208	3	1292.03	3278.00	4	1186.22	4043.00	3	1044.05	4222.50
R209	3	1513.12	6739.00	6	1475.50	6403.00	3	<b>1440.30</b>	<b>5602.70</b>
R210	3	1451.76	5863.00	6	941.74	5272.00	3	1561.41	5670.90
R211	4	1249.35	4356.00	6	1200.82	4709.00	3	<b>1189.92</b>	<b>4195.10</b>
RC201	6	2415.57	9047.00	7	2374.40	8769.00	5	<b>2298.07</b>	<b>8360.24</b>
RC202	4	21740.97	6145.00	7	1136.82	5284.00	4	2046.15	7467.31
RC203	4	1705.41	6812.00	5	1715.39	6095.00	4	<b>1684.92</b>	<b>5764.29</b>
RC204	3	1336.28	4417.00	5	1248.27	4645.00	3	<b>1245.31</b>	<b>4378.33</b>
RC205	4	1625.37	5250.00	8	1374.65	5224.00	5	2096.55	7233.18
RC206	4	1689.13	5523.00	6	1245.04	5442.00	4	1748.73	5814.34
RC207	4	1735.33	6902.00	7	1803.47	6918.00	4	<b>1618.64</b>	<b>5740.28</b>
RC208	4	1373.24	4784.00	5	1407.83	5031.00	3	<b>1306.25</b>	<b>4431.72</b>

**表5** 小規模インスタンス C101 において、顧客要求が同じ場合の HACO と他の知的アルゴリズムとの比較

	DemandNHACORReference [35]					]Reference [42].							
	ビー	ディス	コスト	時間		ビー	ディス	コスト	時間	ビー	ディス	コスト	時間
10	52	8	<b>1307.76</b>	<b>11526.00</b>	142.48	8	1322.22	11566.00	165.01	8	1373.71	11540.00	156.70
20	26	5	939.16	6942.64	<b>48.89</b>	5	939.16	6942.64	52.82	5	967.71	7025.98	52.75
30	13	4	537.29	4575.96	<b>18.78</b>	4	537.29	4575.96	20.04	4	537.29	4575.96	20.35
40	7	2	<b>289.93</b>	<b>2308.37</b>	<b>9.53</b>	2	308.27	2349.06	10.70	2	308.27	2349.06	10.92
50	2	1	39.82	1039.87	<b>2.24</b>	1	39.82	1039.87	2.74	1	39.82	1039.87	2.59
	DemandNReferenceです。 [18]									]GASA			
	ビー	ディス	コスト	時間		ビー	ディス	コスト	時間	ビー	ディス	コスト	時間
10	52	8	1328.31	11561.00	154.97	9	2212.29	11931.00	87.60	8	1352.08	11902.00	143.89
20	26	5	939.16	6942.64	54.33	6	967.71	7025.89	91.81	5	943.91	6993.66	49.83
30	13	4	537.29	4575.96	20.90	4	621.66	4634.73	76.63	5	462.38	4155.65	60.02
40	7	2	308.27	2349.06	10.76	2	329.37	3353.68	49.58	2	291.393	2518.37	28.53
50	2	1	39.82	1039.87	2.62	1	39.82	1039.87	17.25	1	39.82	1039.87	21.70

表6 同じ顧客要求を持つ小インスタンス R101 に対する HACO と他の知的アルゴリズムとの比較

Demand	NHACORReference [35]		Reference [42].							
	VehDisCost	時間	ビーデ イエス	コスト	時間	ビー	ディス	コスト	時間	
$0 \leq D < 1036$	101108.0613196.00	84.97	9	101045.7	13155.00	86.41	10	989.62	12112.00	87.14
$10 \leq D < 2039$	<b>111066.2414120.00</b>	<b>94.53</b>	9	111084.9	15136.00	96.60	12	1103.99	15145.00	95.61
$20 \leq D < 3019$	<b>7612.137616.43</b>	<b>35.04</b>		7616.11	7630.85	36.82	7	666.87	8678.45	36.50
$30 \leq D < 405$	3242.553242.68	<b>6.43</b>		3242.55	3242.68	6.87	3	242.55	3242.68	6.87
$D \geq 401$	146.041046.07	<b>0.92</b>		146.04	1046.07	1.02	1	46.04	1046.07	1.00
ndN	Dema参考文献[18].		GA				SA			
	VehDisCost	時間	ビーデ イエス	コスト	時間	ビー	ディス	コスト	時間	
$0 \leq D < 1036$	101052.4213183.00	88.63	2	131262.4	15177.00	110.39	5	682.18	12291.00	32.17
$10 \leq D < 2039$	111072.2214188.00	98.46		111144.9	14175.00	95.70	11	1086.08	14740.00	95.32
$20 \leq D < 3019$	7616.117630.85	35.28		7619.16	7646.65	73.19	7	615.91	7622.27	37.30
$30 \leq D < 405$	3242.553242.68	6.82		3303.65	4324.81	58.36	3	242.55	3242.68	27.35
$D \geq 401$	146.041046.07	30.96		146.04	1046.07	10.57	4	46.04	1046.07	32.27
$10 \leq D < 2041$	<b>91115.0812460.00</b>	<b>101.43</b>		91143.91	12475.00	104.27	9	410.00	1303.9513	103.68
$20 \leq D < 3026$	<b>7926.778930.23</b>	<b>51.84</b>		7936.50	9115.98	53.00	8	4.66	938.41912	56.38
$30 \leq D < 4011$	5533.966591.03	<b>17.03</b>		5533.96	6591.03	18.01	5	1.62	566.58661	18.12
$D \geq 405$	3271.833271.85	<b>6.21</b>		3271.83	3271.85	7.15	3	1.85	271.83327	6.72
DemandN	参考文献[18].		GA				SA			
	VehDisCost	時間	ビーデ イエス	コスト	時間	ビー		DisCost	時間	
$0 \leq D < 1017$	4539.155562.93	30.53		4434.49	5514.32	101.07	5	4.89	420.66434	30.26
$10 \leq D < 2041$	91143.9112475.00	102.20		91301.4512427.00	10586.00	105.86	11	619.00	1149.8612	121.38
$20 \leq D < 3026$	7926.778930.23	55.26		7945.63	9376.25	92.29	9	0.57	937.78941	53.05
$30 \leq D < 4011$	5533.966591.03	6.75		5612.50	7691.09	59.73	5	1.03	533.96659	29.73
$D \geq 405$	3271.833271.85	6.75		3271.83	3271.85	18.36	3	1.85	271.83327	27.56

た変異操作と適応戦略の有効性を証明するために、文献[18,35,42]で改良されたアントコロニーアルゴリズム、および他のアルゴリズムGA、SAと比較した。文献[18]では、アントコロニーアルゴリズムをベースにローカルサーチ法を追加し、文献[35]では、新しいフェロモン更新式を提案し、文献[42]では、適応戦略と交換メカニズムを導入しています。

表5、6、7、8、9、10では、同じ顧客要求を持つ異

は顧客の需要、 $N$ は顧客の数、 $Veh$ は車両の数、 $Dis$ は総移動距離、 $Cost$ は車両の総コスト、 $Time$ はアルゴリズムの実行時間を表します。また、太字は、HACOが比較アルゴリズムよりも良い結果を得ることを示しています。表5、表6、表7は、提案するHACOと比較アルゴリズムを用いて、異なる小インスタンス（C101、R101、RC101）の最適な結果をそれぞれ計算し、計算結果とアルゴリズムの性能を分析したものである。表5、表6、表7によると、使用する車両数が同じか類似している場合、HACOは15回の比較のうち少なくとも8回は良い結果を得ることができ、走行距離を短縮するだけでなく、車両の総コストを削減することができる。車両台数と総コストが同じ場合

表8 ミディアムインスタンス C201 において、顧客要求が同じ場合の HACO と他の知的アルゴリズムとの比較

	DemandNHACORReference [35]					]Reference [42].							
	ビー	ディス	コスト	時間		ビー	ディス	コスト	時間	ビー	ディス	コスト	時間
10	52	2	1219.30	5284.26	1484.69	3	1288.72	5304.53	150.56	2	1219.44	5286.27	152.97
20	26	2	<b>977.33</b>	<b>3039.87</b>	<b>51.85</b>	3	1014.29	4135.72	53.67	2	1039.82	4296.27	54.91
30	13	1	658.58	2574.51	21.30	3	578.35	1614.26	22.49	1	658.58	2574.51	23.10
40	7	1	213.97	1214.06	10.04	2	206.63	1182.07	11.67	1	213.97	1214.06	10.88
50	2	1	39.82	1039.82	<b>2.01</b>	1	39.82	1039.82	2.38	1	39.82	1039.82	2.63
	DemandNReference [18]									JGASA			
	ビー	ディス	コスト	時間		ビー	ディス	コスト	時間	ビー	ディス	コスト	時間
10	52	3	1234.61	5299.47	152.91	3	470.35	4133.76	186.64	3	652.46	2552.16	133.60
20	26	2	1010.32	4038.54	53.45	3	1091.43	4365.19	96.65	4	998.92	3416.61	54.18
30	13	1	658.58	2574.51	22.30	1	594.08	1658.22	69.71	2	361.04	1549.73	29.49
40	7	1	213.97	1214.06	10.85	1	206.63	1182.07	42.34	2	206.63	1182.07	27.98
50	2	1	39.82	1039.82	2.33	1	39.82	1039.82	18.66	1	39.82	1039.82	24.30

表9 大規模インスタンス R201 で顧客要求が同じ場合の HACO と他の知的アルゴリズムの比較

Demand		NHACORReference [35]		]Reference [42].							
		VehDisCost	時間	ビーデ イーエス	コスト	時間	ビー	ディス	コスト	時間	
$0 \leq D < 1036$		2948.103027.11	91.55	2987.21	3047.96	94.88	2	1001.76	3068.24	94.04	
$10 \leq D < 2039$		2904.873903.03	100.43	2943.22	3927.11	101.24	2	937.75	3921.33	101.15	
$20 \leq D < 3019$		2499.012562.93	36.12	2502.17	2640.05	36.41	2	546.25	2816.44	36.48	
$30 \leq D < 405$		1270.861270.92	6.61	1270.86	1270.92	6.78	1	270.86	1270.92	36.48	
$D \geq 401$		146.041046.03	1.00	146.04	1046.03	1.10	1	46.04	1046.03	1.03	
Demand	N	参考文献[18].		GA			SA				
		VehDisCost	時間	ビーデ イーエス	コスト	時間	ビー	ディス	コスト	時間	
$0 \leq D < 1036$		2987.213047.96	92.25	6966.24	3172.28	94.85	4	965.44	3121.19	92.71	
$10 \leq D < 2039$		2918.133911.52	101.12	41150.74	3949.41	797.88	4	987.91	3939.11	132.32	
$20 \leq D < 3019$		2544.342803.26	36.57	3505.15	2685.01	79.07	3	543.92	2647.26	36.89	
$30 \leq D < 405$		1270.861270.92	6.66	1270.86	1270.92	6.78	1	270.86	1270.92	36.48	
$D \geq 401$		146.041046.03	0.99	146.04	1046.03	19.53	1	46.04	1046.03	15.49	

が同値であれば、HACO は比較アルゴリズムよりも最適な結果を計算するのにかかる時間が短くなります。さらに、これらの3つの表は、小さなインスタンスを解くとき、顧客が地理的にどのように分布していても、HACOの結果は比較アルゴリズムよりも優れていることも示しています。

表8、表9、表10は、提案したHACOと比較アルゴリズムを用いて、中規模インスタンス(C201)と大規模インスタンス(R201とRC201)の最適な結果をそれぞれ計算し、その結果を分析することで、提案したHACOの性能をさらに説明したものである。表8に示すように、提案したHACOは、中規模インスタンスを解くときには良い結果が得られず、顧客要求が20のときにのみ

モデルを使用しています。同じ車両数の場合、HACOは比較アルゴリズムよりも総距離と総コストで良い結果、または同等の結果を得ており、実行時間も他の知的アルゴリズムよりも短いです。

表5、6、7、8、9、10から、顧客需要が10～30の場合、中程度のインスタンスC201を除いて、HACOは他の知的アルゴリズムに比べて、小インスタンスと大インスタンスで良い結果を得ていることがわかります。参考文献[43]で提案されているハイブリッド・アルゴリズムも、大規模インスタンス問題の解決に適しており、貪欲アルゴリズムと可変近傍探索を組み合わせている。初期解の精度は向上するが、収束性は他の知的アルゴリズムと比較して劣る。文献[44,45]で提案されているアルゴリズムは、実用的な

表 10 顧客要求が同じ場合の大規模インスタンス RC201 における HACO と他の知的アルゴリズムとの比較

Demand		NHACO Reference [35]		Reference [42].					
		VehDisCost		時間	ビーデ イエス	コスト	時間	ビー	DisCost 時間
$0 \leq D < 1017$		2512.272517.23		28.31	2517.21	2535.17	28.92	2	517.2125 28.76
$10 \leq D < 2041$		<b>31183.364303.26</b>		<b>106.42</b>	31188.90	4323.45	107.49	3	35.17 1214.904 107.93
$20 \leq D < 3026$		<b>2867.322997.92</b>		<b>55.34</b>	2871.54	3091.32	55.42	2	374.58 891.9330 56.43
$30 \leq D < 4011$		2501.222530.98		<b>17.28</b>	2501.22	2530.98	18.15	2	63.48 531.4626 18.11
$D \geq 405$		1270.431270.48		<b>6.49</b>	1270.43	1270.48	6.64	1	16.03 270.4312 6.77
									70.48
Demand		参考文献[18].		GA				SA	
		VehDisCost		時間	ビーデ イエス	コスト	時間	ビー	DisCost 時間
$0 \leq D < 1017$		2517.212535.17		29.14	3408.32	2311.93	98.98	4	375.4814 30.33
$10 \leq D < 2041$		31214.904374.58		109.53	61238.52	4390.58	107.33	5	45.28 1218.884 113.28
$20 \leq D < 3026$		2872.773024.85			3896.95	3126.22	79.08	4	385.01 891.1034 59.31
$30 \leq D < 4011$		2501.222530.98			1642.08	2633.74	18.75	3	23.62 507.7925 28.99
$D \geq 405$		1270.431270.48			1270.43	1270.48	23.16	1	33.47 270.4312 26.83
									70.48

図10 最適経路の収束曲線

の問題を解決します。しかし、解の質を向上させることを前提に、図10に示すように、本稿で提案したハイブリッド・アルゴリズムの収束性についても分析している。

図10は、最適経路の収束を示したもので、サブグラフ(a)、(b)は、それぞれTWVRPを解いたときの異なる比較アルゴリズムの収束曲線を表しています。

(b) TWVRPにおけるHACO, SA, GA, ACOの比較。

(a) HACOの比較、Ref.[35]、Ref.[42]、およびRef.[18]をTWVRPで使用した場合。



## 実用化

提案したHACOの有効性と実用性を検証するために、実際の例に適用する。寧夏回族自治区銀川市星清区の生鮮果物・野菜の物流を例にとります。青果物を倉庫（仙鳳青果流通センター）から13の流通拠点

（興慶区内の13のスーパーマーケット）に送る必要があります。図11は、倉庫（0）と13のスーパーマーケットの位置関係を地図上に示したものです。<sup>2</sup>また、各2点間の距離を表11に示す。

配送センターから各配送拠点に商品を配送する際、輸送車両は7つの部品で構成されています。

<sup>2</sup> <https://map.baidu.com/@11815234,4627365,13z>.

任意の2つの物流拠点および倉庫間の距離(ノード)	13	0
		0
	12	1009.73
		0
	11	2338.99
		1209.91
		0
	10	4130.70
		4384.20
		2758.12
		0
	9	2723.34
		992.84
		442.14
		0
		2526.65
		2244.09
		428.76
	8	3129.07
		0
		2765.47
		4627.54
		14202.78
	7	18601.23
		16912.56
		0
		14486.57
		11354073.60
	6	3774.74.79
		3066.44
		0
		1317.07
		3847.98
	5	3998.29
		707.59
		3345.50
		3651.14
		0
		3187.91
		1566.84
	4	3398.16
		1075.87
		3207.87
		3468.77
		2410.09
	3	3309.50
		4285.24
		3630.99
		788.51
		716.72
		2204.81
		2588.19
	2	0
		3918.52
		3671.38
		6496.76.01
		4099.16
		4693.87.93
		1556.27
		3194.93.15
	1	0
		2493.50.96
		3423.51
		4086.02.62
		2389.60
		4803.85.24
		751.54
		4316.47.84.93
		1219.43
		3600.23.96
	0	3839.21
		2196.49.24
		110.95
		636.44.77
		3042.28
		1405.41.36
		2257.03
		4755.93
		339.73
		266.15
		3068.41
		3053.16
		2666.65
		7082.25
		2473.58
		452.76
		6962.62

表12 各配信ポイントのタイムウィンドウと需要

ポイン ト	アイ	バイ	需要(トン)	サービス時間( 分)
0	9:40	11:40	0	0
1	10:10	10:25	15	20
2	10:00	10:15	16	15
3	9:50	10:05	15	15
4	10:20	10:45	10	13
5	11:00	11:25	5	8
6	10:50	11:15	7	9
7	10:35	10:55	5	8
8	10:00	10:30	25	17
9	10:10	10:35	22	19
10	10:45	11:10	26	21
11	10:05	10:40	18	11
12	10:25	10:50	13	14
13	10:20	10:45	16	12



図11 配信センターと配信ポイントの地理的位置関係

同型のトラックを使用しています。トラックの最大積載量は40トンで、平均速度は時速60キロです。各トラックの固定費は600元/km、輸送費は5元/kmです。表12は、図11のマーキングの順序に従って、配送センターとスーパーマーケットの具体的な状況を示したものである。表12からわかるように、スーパーマーケットは商品を購入する際、交通量の多い区間を避けようとしており、スーパーマーケットの時間帯はほぼ9時40分から11時40分の間であることがわかる。したがって、本稿の研究では、交通渋滞を考慮しておらず、車両はスーパーマーケットが指定した時間枠内で商品を配送することができる。

実際の問題の具体的な条件は、表11と12によって

知ることができます。を減らすことが目標です。

表14 HACOの結果  
TWVRPインスタンスの解決

ビー	ルート	距離(m)	需要の総和 (トン)
1	0-8-1-0	10415.41	40
2	0-11-9-0	6037.71	40
3	0-13-12-7-5-0	21808.75	39
4	0-3-2-6-0	13435.75	38
5	0-4-10-0	3677.46	36
合計		55375.08	193

表13

の結果と

ビー	ルート	距離(m)	需要の総和 (トン)
1	0-1-11-0	10301.00	33
2	0-13-12-7-0	19966.17	34
3	0-3-2-0	12287.54	31
4	0-9-4-0	3435.48	32
5	0-8-6-0	9083.86	32
6	0-10-5-0	2929.77	31
合計		58003.82	193

TWVRPインスタンス

揮発性因子を固定定数から動的な値に変更して解くACO

ということは、この問題の最適解は  
の間、常に数値を調整しながら見つけていきます。  
の収束性を向上させるために、アルゴリズムの演算を行  
います。  
アルゴリズムを採用しています。ACOのフェロモン更新  
式は  
改良された、適応性のある揮発性の設備の参加を得て。  
ーターは、フェロモン濃度を増減させることができます

その道の長所と短所に応じて  
HACOの性能を検証するために、Solomon  
のインスタンスを作成し、その結果、提案したアルゴ  
リズムは  
は良い性能を持っています。今後の研究の方向性とし  
ては  
は、多目的グリーンビークルルーティング問題を考えま  
す。

とができます。

は、車両数を最小化し、車両の総走行距離を短縮  
することで、物流センターのトータルコストを削減する  
。そこで、TWVRPインスタンスを解くためにACOと  
HACOを適用する。2つのアルゴリズムの結果を比較  
することで、提案するHACOの実用性と有効性を分析  
・検証し、その結果を表13と表14に示す。

表13から、ACOを用いてTWVRPインスタンスを解  
くには6台の車両が必要であることがわかるが、この  
6台の車両は十分に活用されていない。最後の行によ  
ると、総距離は 58003.82 m  
であり、すべての配送を完了するためには 3890.02  
元の費用が必要である。しかし、表14のTWVRPの例  
をHACOで解くと、5台の車両が必要となり、表13の6  
台の車両に比べて、この5台の車両はほぼフルに活用  
されている。このときの総距離は 55375.08 m  
であり、表 13 の総距離よりも 2628.74 m  
少ない。すべての配送を完了するための車両のコス  
トは3276.88元で、613.14元から減少している。した  
がって、例の適用と分析を通じて、提案されたHACO  
が効果的であり、実用的な意義があることを示すこ

## 結論

本稿では、主に時間窓のある車両ルーティングの問題  
を取り上げ、車両の輸送距離と輸送コストを削減する  
ために、ハイブリッド・アントコロニーアルゴリズム  
を提案する。基本的なアントコロニーアルゴリズムと比  
較して、HACOはフェロモンを変更します。



は、車両の距離やコストを最適化するだけでなく、サービスに対する顧客満足度を最適化目標とする必要がある。また、近年の車両経路問題の研究では、車両の排気ガスを削減することが新しいアイデアとなっているため、二酸化炭素の排出量も最適化目標としてとらえる必要があります。

**Open Access** この記事はCreative Commons Attribution 4.0 International

Licenseに基づいてライセンスされています。このライセンスは、原著者と出典に適切なクレジットを与え、Creative Commonsライセンスへのリンクを提供し、変更が加えられたかどうかを示す限り、あらゆる媒体や形式での使用、共有、翻案、配布、および複製を許可します。この記事に掲載されている画像やその他の第三者の素材は、その素材のクレジットラインに別段の表示がない限り、その記事のクリエイティブ・コモンズ・ライセンスに含まれています。本記事のクリエイティブ・コモンズ・ライセンスに含まれていない素材で、あなたの意図する使用が法令で認められていない場合、または認められた使用を超える場合は、コピーライト保有者から直接許可を得る必要があります。このライセンスのコピーを見るには、<http://creativecommons.org/licenses/by/4.0/>をご覧ください。

## リファレンス

1. Zhang HZ, Zhang QW, Ma L, Zhang ZY, Liu Y (2019) A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows.インフォサイ490:166-190
2. Ralphs TK, Kopman L, Pulleyblank WR, Trotter LE (2003) On the capacitated vehicle routing problem.Math Progr 94(2-3):343-359
3. Hernandez F, Feillet D, Giroudeau R, Naud O (2016) Branch-and-price algorithms for the solution of the multi-trip vehicle routing problem with time windows.Eur J Oper Res 249(2):551-559
4. Schneider M (2016) The vehicle-routing problem with time windows and driver-specific times.Eur J Oper Res 250(1):101-119

5. Koc C, Bektas T, Jabali O, Laporte G (2015) A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. *Comput Oper Res* 64:11-27
6. Bertsimas D, Simchi-Levi D (1993) A new generation of vehicle routing research: robust algorithms addressing uncertainty. *Oper Res* 44(2):286-304
7. Ombuki B, Ross BJ, Hanshar F (2006) Multi-objective genetic algorithms for vehicle routing problem with time windows. *Appl Intell* 24(1):17-30
8. Ghoseiri K, Ghannadpour SF (2010) Multi-objective vehicle routing problem with time windows using goal programming and genetic algorithm. *Appl Soft Comput* 10(4):1096-1107
9. Wang KG, Gao YL (2019) Application of differential evolution algorithm based on mixed penalty function screening criterion in imbalanced data integration classification. *数学* 7(12):1237
10. Wang KG, Gao YL (2019) Topology structure implied in  $\beta$ -Hilbert space, ハイゼンベルグ不確定性量子特性、DEアルゴリズムの数値シミュレーション. *数学* 7(4):330
11. Ho SC, Haugland D (2004) A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Comput Oper Res* 31(12):1947-1964
12. Belhaiza S, Hansen P, Laporte G (2014) A hybrid variable neighborhood tabu search heuristic for the vehicle routing problem with multiple time windows. *Comput Oper Res* 52:269-281
13. Wang Z, Li Y, Hu XP (2014) A heuristic approach and a tabu search for the heterogeneous multi-type fleet vehicle routing problem with time windows and an incompatible loading constraint. *Comput Ind Eng* 89:162-176. <https://doi.org/10.1016/j.cie.2014.11.004>
14. Chiang WC, Russell RA (1996) Simulated Annealing metaheuristics for the Vehicle routing problem with time windows. *Ann Oper Res* 63(1):3-27
15. Deng AM, MAO C, Zhou YT (2009) Optimizing Research of an improved simulated annealing algorithm to soft time windows vehicle routing problem with pick-up and delivery. *Syst Eng Theory Pract* 29(5):186-192
16. Wang C, Mu D, Zhao F, Sutherland JW (2015) A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup delivery and time windows. *Comput Ind Eng* 83:111-122
17. Yu B, Yang ZZ (2011) An ant colony optimization model: the period vehicle routing problem with time windows. *Transp Res Part E: Log Transp Rev* 47(2):166-181
18. Ding QL, Hu XP, Sun LJ, Wang YZ (2012) An improved ant colony optimization and its application to vehicle routing problem with time windows. *Neurocomputing* 98:101-107
19. Wang Y, Wang L, Chen GC, Cai ZQ, Zhou YQ, Xing LN (2020) An improved ant colony optimization algorithm for the periodic vehicle routing problem with time window and service choice. *Swarm Evolut Comput*. <https://doi.org/10.1016/j.swevo.2020.100675>
20. Geng N, Chen ZT, Nguyen QA, Gong DW (2021) Particle Swarm Optimization algorithm for the optimization of rescue task allocation with uncertain time constraints. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-020-00252-2>
21. Ünal AN, Kayakutlu G (2020) Multi-objective particle swarm optimization with random immigrants. *Complex Intell Syst* 6(3):635-650
22. Qin SF, Sun CL, Zhang GC, He XJ, Tan Y (2020) A modified particle swarm optimization based on decomposition with different ideal points for many-objective optimization problems. *Complex Intell Syst* 6(2):263-274
23. Liu F, Zhang JW, Liu T (2020) A PSO-algorithm-based consensus model with the application to large-scale group decision-making. *Complex Intell Syst* 6(2):287-298
24. Ebrahimnejad A, Enayattab M, Motameni H, Garg H (2021) Modified artificial bee colony algorithm for solving mixed interval-evaluationされたファジー最短経路問題. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-021-00278-0>
25. Yang Y, Duan Z (2020) An effective co-evolutionary algorithm based on artificial bee colony and differential evolution for time series predicting optimization. *Complex Intell Syst* 6(2):299-308
26. Wang H, Wang WJ, Zhou XY, Zhao J, Wang Y, Xiao SY, Xu MY (2020) Artificial bee colony algorithm based on knowledge fusion. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-020-00171-2>
27. Pandey AC, Tikkiwal VA (2021) Stance detection using improved whale optimization algorithm. *Complex Intell Syst*. <https://doi.org/10.1007/s40747-021-00294-0>
28. Enayattab M, Ebrahimnejad A, Motameni H (2019) Dijkstra algorithm for shortest path problem under interval-valued fuzzy environment. *コンプレックス・インテル・シスト* 5(2):93-100
29. Colnani A, Dorigo M, Maniezzo V (1991) Ant coloniesによる分散型最適化. である. *ECAL91-European Conference on Artificial Lifeの議事録*.
30. Bullnheimer B, Hartl RF, Strauss C (1997) Applying the ant system to the vehicle routing problem. である. *メタヒューリスティクスに関する国際会議* 1-12
31. Yu B, Yang ZZ, Yao B (2009) An improved ant colony optimization for vehicle routing problem. *Eur J Oper Res* 196(1):171-176
32. Mavrouniotis M, Yang S (2011) A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. *Soft Comput* 15(7):1405-1425
33. Wang XY, Choi TM, Liu HK, Yue XH (2016) A novel hybrid ant colony optimization algorithm for emergency transportation problems during post-disaster scenarios. *IEEE Trans Syst Man Cybern Syst* 48(4):545-556
34. Jabir E, Panicker V, Sridharan R (2017) Design and development of a hybrid ant colony-variable neighbourhood search algorithm for a multi-depot green vehicle routing problem. *トランスプ・レス・パートDトランスプ・エンバイロン* 57:422-457
35. Li YB, Soleimani H, Zohal M (2019) An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. *J Clean Prod* 227:1161-1172
36. Dorigo M (1992) 最適化学習と自然なアルゴリズム. *イタリア・ミラノ工科大学 (Politecnico di Milano)*
37. El-Sherbeny NA (2010) Vehicle routing with time windows: an overview of exact, heuristic and meta-heuristic methods. *J King Saud Univ Sci* 22(3):123-131
38. Dorigo M, Maniezzo V, Colnani A (1996) The ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B* 26(1):29-41
39. Dorigo M, Bonabeau E, Theraulaz G (2000) Ant algorithms and stigmergy. *Fut Gen Comput Syst* 16(8):851-871
40. Dorigo M, Blum C (2005) Ant colony optimization theory: a survey. *Theor Comput Sci* 344(2-3):243-278. <https://doi.org/10.1016/j.tcs.2005.05.020>
41. Ghannadpour SF, Noori S, Tavakkoli-Moghaddam R, Ghoseiri K (2014) A multi-objective dynamic vehicle routing problem with fuzzy time windows: model, solution and application. *Appl Soft Comput* 14:504-527
42. Huang SH, Huang YH, Blazquez CA, Paredes-Belmar G (2018) Application of the ant colony optimization in the resolution of the bridge inspection routing problem. *Appl Soft Comput*. <https://doi.org/10.1016/j.asoc.2018.01.034>
43. Zhou YZ, Huang JC, Shi JM, Wang R, Huang KH (2021) The



- 
- ~~electric vehicle routing problem with partial recharge and~~  
vehicle recycling.Complex Intell Syst.  
<https://doi.org/10.1007/s40747-021-00291-3>
44. Feng GQ, Xu P, Cui DL, Dai XW, Liu H, Zhang Q (2021)  
Multi-stage timetable rescheduling for high-speed railways: a  
dynamic programming approach with adaptive state  
generation.Complex Intell Syst.  
<https://doi.org/10.1007/s40747-021-00272-6>

45. Jie J, Liu KR, Zheng H, Wang BB, Dai R (2021) Double phases PSOに基づくジェスチャー認識のためのマルチチャンネルs EMGの高次元特徴データの削減。Complex Intell Syst. <https://doi.org/10.1007/s40747-020-00232-6>

# **出版社の注意事項 Springer**

Natureは、出版された地図や所属機関における法律上の主張に関して、中立を保っています。