

# 最適化手法を用いた移動体の順序付け方式 ——公平性を考慮した遅延最小化

澤田 めぐみ<sup>1,a)</sup> 尾崎 敦夫<sup>1</sup> 白石 将<sup>1</sup> 松村 寛夫<sup>1</sup>

受付日 2015年3月4日, 採録日 2015年8月12日

**概要:** 複数の移動体を扱う交通システムでは, 同一領域を移動体が共有して通過することが多い. この共有領域においては, 安全に効率良く, かつ, 移動体間の公平性ができるだけ損なわれないように移動を制御することが求められる. 本論文では, 移動体が一定時間, 待機可能な場所がある応用問題を対象に, 共有領域を複数の移動体が通過する際の順序を最適化する方式を提案する. 提案方式は, 遺伝的アルゴリズム (GA: Genetic Algorithm), タブーサーチ (TS: Tabu Search) を用いて移動体の順序付けを行うものであり, 移動体全体の遅延時間や公平性損失の最小化を図る. 計算機シミュレーションによる評価の結果, 遅延時間や公平性損失に関する解の評価値が, 先着順で移動体を通過させる従来方式と比べ, 提案方式は最大 60%程度に改善されることを確認した. また, 提案方式で用いる最適化手法として, TS は, GA と比べ, 得られる解の評価値が最大 10%程度優れていることを確認した.

**キーワード:** 交通システム, 最適化, 遺伝的アルゴリズム, タブーサーチ

## Ordering of Moving Objects by Optimization Method for Delay Minimization Considering Fairness

MEGUMI SAWADA<sup>1,a)</sup> ATSUO OZAKI<sup>1</sup> MASASHI SHIRAISHI<sup>1</sup> NOBUO MATSUMURA<sup>1</sup>

Received: March 4, 2015, Accepted: August 12, 2015

**Abstract:** In some transportation system which involves many moving objects, it is general that these objects share common regions which they pass. The movement of objects in the common regions should be controlled to smooth the traffic while maintaining safety and fairness among the objects. In this paper, we discuss how to schedule the objects utilizing a shared region with a holding area where each object can wait for constant time interval. The proposed method determines the permutation of moving objects which represents the scheduling order of the objects by introducing an optimization algorithm such as genetic algorithm (GA), tabu search (TS). The objective function to be optimized is determined by the total delay and fairness of the schedule. We confirmed that our method improved the objective function value by 60% compared to the conventional first-come first-served (FCFS) method in some simulations. Additionally, TS could generate the solution which were 10% better than that by GA.

**Keywords:** transportation system, optimization, genetic algorithm, tabu search

### 1. はじめに

飛行機, 船舶, 自動車などの移動体を構成要素として含む交通システムにおいては, 安全性が確保され, 効率が良

く, さらに, 移動体間の公平性が損なわれない交通の実現が望まれている. 安全性を確保するためには, 複数の移動体の間に適切な間隔 (「規定間隔」と呼ぶ) を確保する必要がある. 効率の良さは, 各移動体についての遅延時間や, 移動に要するエネルギー消費量などの観点に基づき定量化される. また, 公平性は, たとえば, 移動体間の追い越しの発生数に基づき定量化される.

<sup>1</sup> 三菱電機株式会社情報技術総合研究所  
Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Kanagawa 247-8501, Japan

<sup>a)</sup> Sawada.Megumi@ah.MitsubishiElectric.co.jp

一般に、同一方面に向かう移動体は、同一領域を共有して移動することが多い（この領域を「共有領域」と呼ぶ）。したがって、移動体の数が増えると、共有領域において、規定間隔を満たすために互いに干渉しあう渋滞状況が発生する。たとえば、飛行機では空港付近、船舶では港付近、自動車では合流付近などでこのような状況が発生する。このため、交通システムでは、複数の移動体間で規定間隔を保ち、いかに効率良く、公平性を損なわないように共有領域を移動するかが課題となる。渋滞の程度が低ければ、共有領域内で各移動体の速度や経路を変化させるだけで、規定間隔などの制約を満たす交通流が実現できる。しかし、渋滞の程度が高く、それだけでは対処不可能な場合に備え、移動体が一定時間待機することが可能な待機場所が用意される場合がある（たとえば、飛行機の待機旋回など）。本論文では、そのような待機場所を備えた交通システムを対象として、共有領域における移動体の通過順序付け方式を提案する。

通過順序付けとして最も単純なのは、先着順（共有領域への到着順）で、移動体を通過させる方式（「先着順方式」と呼ぶ）である。先着順方式は、移動体間の追い越しが発生しないため、公平性が確保できる。しかし、この方式では、移動体が短い間隔で連続して共有領域に到着する場合、ある移動体に遅延が発生すると、後続の移動体に遅延が伝播するため、交通全体として大きな遅延が発生する恐れがある。この問題を解決するためには、遅延の伝播を断ち切るような、先着順方式以外の順序付けが必要である。

本論文では、公平性を大きく損なわない範囲内で先着順からの順序入れ替えを許容し、移動体の遅延時間の総和を最小化する方式を提案する。そして、先着順方式と提案方式について、交通全体の遅延への影響を評価する。

本論文の構成は以下のとおりである。2章で、既存の移動体の順序付け方式を述べ、本研究の位置付けを示す。また、3章では、検討対象とする交通システムのモデルを示す。そして4章で、順序付けのための先着順方式および提案方式を説明する。5章では、提案方式の評価結果を示し、6章でその結果を考察する。最後に、7章で本論文の結論を示す。

## 2. 本研究の位置付け

本章では、複数の移動体の順序付けに関する、既存の方式について述べ、本研究の位置付けを示す。なお、ここでは、各移動体が同一の目標地点を目指して移動することを前提とする。そして、各移動体の目標地点への到着順序と到着時刻を定める処理をスケジューリングと呼び、各移動体の目標地点への到着時刻を確定した結果を、スケジュールと呼ぶ。また、静的なスケジューリングを前提とし、対象とする移動体に関する情報（目標地点への到着が可能な時間帯など）は、スケジューリングを実施する際に、すべ

て正確に分かっているものとする。

移動体の順序付けに関する標準的な問題設定では、各移動体について、目標地点への到着が可能な時間帯（「タイムウィンドウ」と呼ぶ）が制約条件として与えられる。さらに、安全性に基づく制約として、目標地点への到着時点での移動体間の距離間隔や時間間隔の下限、すなわち規定間隔が設けられる。規定間隔は、前後する移動体の種類に応じて異なる。以上の制約の下で、各移動体の目標地点への到着時刻を適切に設定することにより、各移動体の遅延時間の総和を最小化することや、到着時刻の最大値を最小化するような順序付け方式が検討されている [1]。しかし、このような問題設定は、多くの移動体が短い時間間隔で共有領域に到達した場合に、各移動体のタイムウィンドウが狭いと制約を満たすことが困難になるという課題がある。そこで、共有領域に待機場所が用意されている場合が検討されている [2]。

文献 [2] では、飛行機の待機旋回を想定し、移動体は待機場所で、一定時間、もしくはその整数倍の時間、待機することが可能なモデルとなっている。すなわち、この問題拡張で、各移動体のタイムウィンドウは、待機時間に応じて一定時間間隔に並んだものとして表現される。そして、この順序付け問題に関し、待機回数が少なくなるように移動体をスケジューリングしていくことで、準最適解を得る方式が提案されている。具体的には、各移動体について同じ待機回数のタイムウィンドウを抽出し、そのなかで到着可能な上限時刻が早い順に移動体をスケジューリングしていくというものである。しかし、文献 [2] は、以下について考慮していない。

- 移動体間の公平性の確保が必要な場合。
- 規定間隔が移動体の種類に依存する場合。

これに対し、本論文では、上記事項も考慮した、待機場所を含む交通システムに関する移動体の順序付け方式について論じる。

さて、移動体の順序付け問題は、単一機械スケジューリング問題の一種であり、NP 困難であることが知られている [1]。よって、問題を混合整数計画問題に定式化し、分枝限定法などを用いて厳密な最適解を得る方式は、スケジューリング対象とする移動体の数が増えると処理時間が膨大となる。生成されたスケジュールを手で修正（対話型で最適化）する実運用時での利用を想定すると、処理時間は多くても数分以内である必要がある。たとえば、文献 [1] において、厳密解法は、約 100 台の移動体のスケジューリングに約 2 時間を要したという結果が示されている。厳密解法の処理時間は移動体の数に対して指数関数的に増加することから、数十台以上をスケジューリング対象として厳密解法を利用することは困難である。

そこで、文献 [1], [2] では、最適解とは限らないが、それなりに良い準最適解を得る方法が検討されている。文献 [2]

は、前述のとおり、ある評価基準に基づき、評価値が良い順に移動体をスケジューリングしていくことで1つの解を構築する方法である。これに対し、文献[1]は、メタヒューリスティック[3]を適用する方法であり、解を修正する手続きを反復することで準最適解を得る。このメタヒューリスティックを用いることで、先着順方式や文献[2]の方式から得た解を改善する解が得られる可能性がある。

メタヒューリスティックには様々なアルゴリズムが存在する。移動体の順序付け問題では、局所探索法を用いた方式[1]や遺伝的アルゴリズム(GA: Genetic Algorithm)を用いた方式[4],[5]が検討されているが、それらのアルゴリズムの比較は行われていない。本論文では、メタヒューリスティックを用いて順序付けの最適化を図る方式について、大域的な探索を行う傾向が強いGAを用いた場合と、局所的な探索を行う傾向が強いタブーサーチ(TS: Tabu Search)を用いた場合の両方についても論じる。

### 3. 想定する交通モデル

本章では、本論文で想定する交通モデルでの共有領域と移動体について説明する(図1)。

#### 共有領域のモデル

- 複数の移動体が存在し、これらが共通に利用する単一の共有領域が存在する。
- 共有領域には入域点と出域点(前述の「目標地点に相当」)がそれぞれ1つずつ存在する。各移動体は入域点から入り、共有領域内を入域点から出域点に移動後、出域点から出るものとする。
- 共有領域の入域点直前に移動体の待機場所が存在し、ある定められた待機時間か、その整数倍の時間だけ、そこで待機可能とする。ここで、待機場所を同時に利用する移動体数に制限はないものとする。

#### 移動体のモデル

- 各移動体が共有領域の入域点に到着する時刻はあらかじめ与えられるが、待機場所を利用することにより、前後の移動体との間隔の調整や、共有領域に進入する順序の入れ替えが可能である。
- 移動体は一定範囲内での速度調整が可能であり、共有領域の入域点から出域点への移動に要する時間は、所定の最短移動時間～最長移動時間の範囲で自由に設定できるものとする。

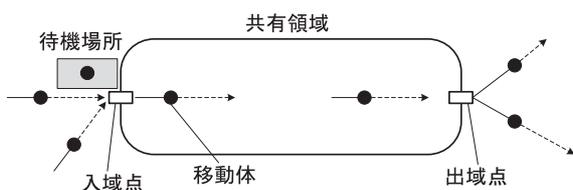


図1 交通モデル

Fig. 1 Traffic model.

- 安全性の確保のため、連続する移動体の前後に所定の規定間隔が保たれている必要がある。規定間隔は時間で与えられるものとし、「規定時間間隔」と呼ぶ。ここでは簡単のため、出域点の通過時刻の時間間隔が所定の閾値以上であれば、規定時間間隔の制約を満たしているものと見なす。

- 各移動体の遅延時間を、出域点通過時刻に基づいて定める。その際、他の移動体が存在しないと仮定した場合の最も早い出域点通過時刻を基準とする。したがって、移動体の遅延時間は、すべて0以上の値となる。

上記の条件の下で、各移動体を、公平性を大きく損なわない範囲内で、可能な限り小さな遅延で移動するようスケジューリングすることが目的である。スケジューリングにおいては、出域点において所定の規定時間間隔を満たすために、以下の手段を利用して出域点通過時刻を調整する。

- 待機場所の利用。
- 共有領域における速度調整。

本論文が対象とする上記問題も文献[1],[2]で示されるように混合整数計画問題に定式化することは可能と考える。しかし、2章で述べたとおり、計算時間の観点から厳密解法の利用は現実的ではない。なお、線形緩和を許容し、混合整数計画問題を線形計画問題として解くことも考えられるが、得られた解を実行可能解に修正することは困難である。以上より、本論文では、解法にメタヒューリスティックを採用する。

### 4. 順序付け方式

本章では、一般的な順序付け方式である従来方式(先着順方式)および、提案方式について述べる。これらの方式は、各移動体の入域点到着時刻が与えられた場合に、出域点で所定の規定時間間隔を満たすスケジュールを生成する。

両方式において、移動体の順列からスケジュールを生成するが、その際に、スケジューラを利用する。本スケジューラは、移動体を与えられた順番で1台ずつスケジューリングしていくことにより、最終的に全移動体のスケジュールを得るものである。具体的な処理内容については後述する。

#### 4.1 先着順方式

先着順、すなわち入域点への到着順に、移動体を共有領域に入域させる方式であり、最も単純かつ一般的な方式である。本方式では、移動体間の公平性を保つことができるが、遅延伝播が発生しやすいという問題がある。

先着順方式の処理手順を図2に示す。

スケジューラは、先着順の順列を入力として、スケジューリングを行う。前述のように、スケジューラは1台ずつ移動体の出域点通過時刻を決定していくが、この際、(直前に出域点通過時刻が決定された)先行する移動体との間で規定時間間隔の制約を満たすことが必要である。具体的に

は、以下の時刻範囲 [I] および [II] に共通に含まれる時刻範囲のうち、最も早い時刻を、スケジューリング対象とする移動体の出域点通過時刻として決定する。

[I] 規定時間間隔を満たす出域点通過時刻範囲. 具体的には、「先行する移動体の出域点通過時刻+規定時間間隔」以降の時刻範囲.

[II] スケジューリング対象とする移動体のタイムウィンドウ. これは、当該移動体の入域点到着時刻, 最短移動時間, 最長移動時間, また、待機場所における待機時間によって定まる.

スケジューラによる移動体の出域点通過時刻決定に関し、移動体の順列が「..., A, P, ...」の順であり、移動体 A がすでにスケジューリングされ、次に移動体 P をスケジューリングする状況下での 3 通りの例を図 3(a)~(c) に示す. ここで、図 3(a)~(c) は、それぞれ以下の場合を示す.

- 図 3(a)：移動体 P が、当該移動体のタイムウィンドウの最も早い時刻にスケジューリングされた場合（共有領域内の速度調整および、待機場所を利用しなかった場合）.

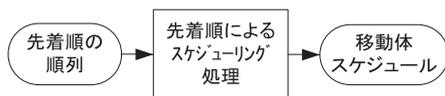


図 2 到着順方式による処理手順

Fig. 2 Flow of FCFS (first-come first-served) method.

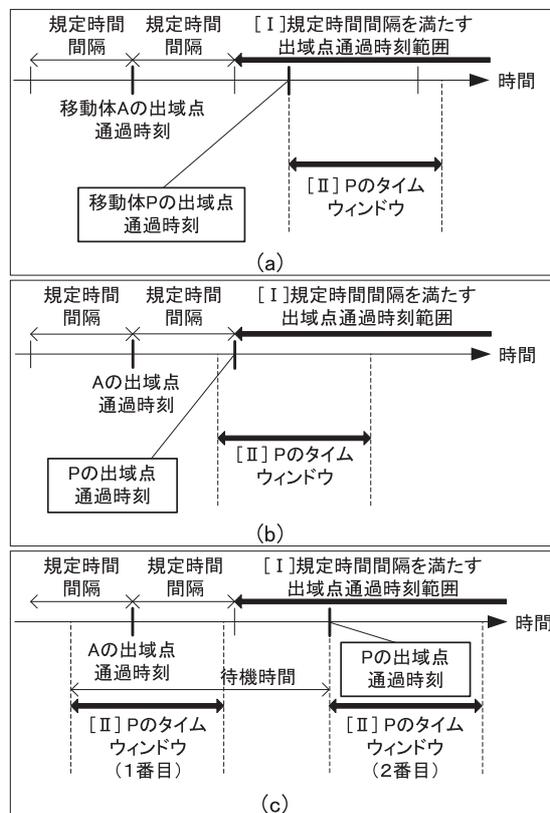


図 3 到着順方式のスケジューリング例

Fig. 3 Scheduling example of FCFS method.

- 図 3(b)：移動体 P を、当該移動体のタイムウィンドウにおいて、規定時間間隔を満たすように速度調整し、スケジューリングした場合（待機場所の利用は不要）.
- 図 3(c)：移動体 P が、当該移動体のタイムウィンドウにおいて、規定時間間隔を満たすように待機場所を利用した調整を行い、スケジューリングした場合.

## 4.2 提案方式

移動体の共有領域からの出域順序を、待機場所を利用して到着順から入れ替えることにより、遅延伝播を解消し、遅延時間の最小化を図る方式である. 提案方式において、順序付けの決定には最適化手法を用いる.

提案方式に関し、まず、最適化手法の適用について述べた後に、処理手順を示す.

### 4.2.1 最適化手法の適用検討

最適化手法として、メタヒューリスティックの GA, TS を用いる. メタヒューリスティックを適用する際には、解をどのように表現するかが課題となる. 順序付け問題の場合、各移動体の目標地点への到着時刻を定めることが目的なので、各移動体の到着時刻の組によって、解を表現することが考えられるが、タイムウィンドウや規定間隔などの制約を満たさない解が高頻度で発生し、最適化が非効率になる恐れがある. なお、制約違反をペナルティとして定量化して目的関数に組み込むことで、無制約として最適化を進める場合も同様と考える.

上記の問題を解決するために、解を順列表現で表す方式が提案されており、衛星観測計画の生成などに適用されている [6]. 移動体順序付けに適用する場合、解は移動体の順列で表されるが、この順列はスケジューリングの順序を意味する. すなわち、順列の先頭から、順に移動体を 1 台ずつ選択し、制約を満たすよう目標地点への到着時刻を確定していくことで、全移動体に関するスケジュールを得る. この方法によれば、解の修正において、制約を満たすスケジュールを得やすいとの利点があるため、本論文でもこの解表現を用いたメタヒューリスティックを採用する.

### 4.2.2 処理手順

提案方式の処理手順を図 4 に示す.

本方式では、以下の処理手順から構成されるループ処理を規定回数だけ反復し、得られたスケジュールの中で評価値が最も良いものを最終的な解として採用する.

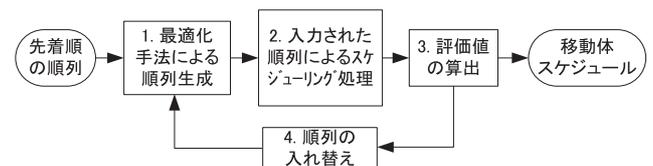


図 4 提案方式による処理手順

Fig. 4 Flow of proposed method.

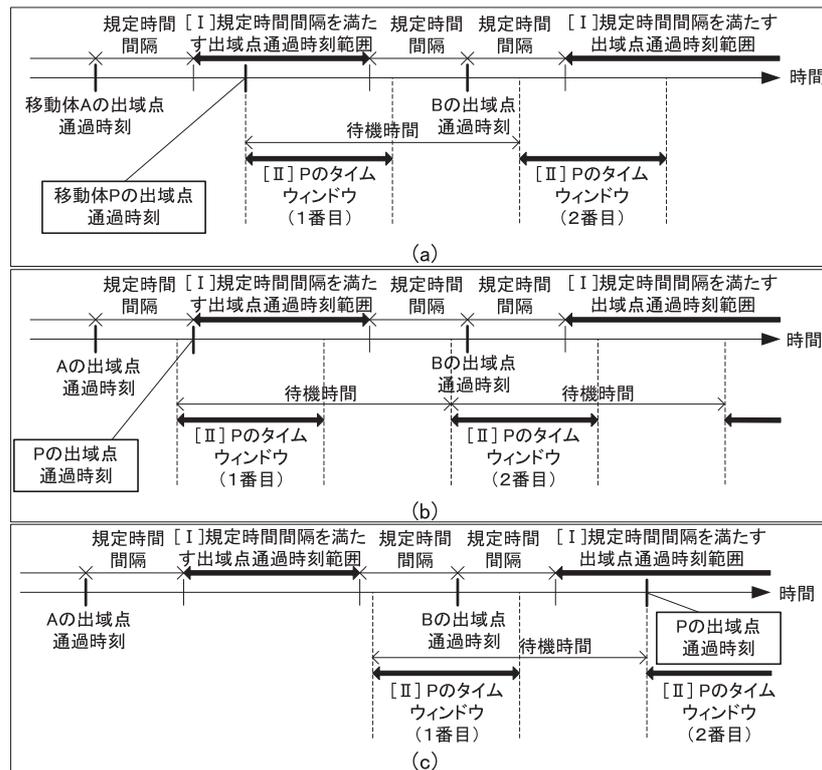


図 5 提案方式のスケジューリング例

Fig. 5 Scheduling example of proposed method.

1. 直前のループで生成した移動体の順列および対応する評価値に基づき、最適化手法を用いて、新たな順列を複数生成する。なお、最初のループでは、先着順を含む、複数の順列を生成するものとする。
2. 最適化手法が出力する各順列に関し、スケジューラは、順列が示す順序で、1台ずつ移動体の出域点通過時刻を決定していく。個々の移動体をスケジューリングする際は、すでにスケジューリングされた移動体との間で規定時間間隔の制約を満たすようにする。先着順方式と異なり、スケジューリング済みの複数の移動体の出域点通過時刻の間を縫って、スケジューリング対象の移動体の出域点通過時刻を決定する。
3. 生成された移動体スケジュールについて、それぞれ、評価値を算出する。
4. 移動体の順列を、評価値が最も良い移動体スケジュールを生成した順列に入れ替える。

以下、上記の提案方式の処理手順における「出域点通過時刻の計算」と「移動体スケジュールの評価値」の内容を説明する。

#### 出域点通過時刻の計算

提案方式の処理手順 2. における、個々の移動体のスケジューリングでは、以下の時刻範囲 [I] および [II] に共通に含まれる時刻範囲のうち、最も早い時刻を、スケジューリング対象とする移動体の出域点通過時刻として決定する。

[I] 規定時間間隔を満たす出域点通過時刻範囲。具体的に

は、「すでにスケジューリングされた各移動体の出域点通過時刻 ± 規定時間間隔」の範囲内を除いた時刻範囲。

[II] スケジューリング対象とする移動体のタイムウィンドウ。

スケジューラによる移動体の出域点通過時刻決定に関し、移動体の順列が「 $\dots, A, \dots, B, \dots, P, \dots$ 」の順であり、移動体 A, B がすでにスケジューリングされ、移動体 P をスケジューリングする状況下における 3 通りの例を図 5 (a)~(c) に示す。なお、この際、P の最初のタイムウィンドウが、A および B の出域点通過時刻付近であるものとする。

図 5 (a)~(c) において、移動体をスケジューリングするための共有領域での速度調整および、待機場所の利用に関する必要有無は、それぞれ、図 3 (a)~(c) と同様の場合を示す。図 3 と図 5 で異なる点は、規定時間間隔を満たす通過時刻範囲であり、先着順方式の図 3 (a)~(c) では、「先行する移動体の出域点通過時刻 + 規定時間間隔」以降の時刻範囲となるのに対し、提案方式の図 5 (a)~(c) では、「すでにスケジューリングされた各移動体の出域点通過時刻 ± 規定時間間隔」の範囲内を除いた時刻範囲となる。

#### 移動体スケジュールの評価値

提案方式の処理手順 3. において、移動体スケジュールの評価値 F は、公平性も考慮した総遅延時間の最小化を行うために、式 (1) に示す目的関数を利用して算出する。

$$F = (\text{総遅延時間}) + \alpha \times (\text{公平性の損失}) \quad (1)$$

式 (1) の各項の意味は以下のとおりである。

- 総遅延時間：各移動体の遅延時間の総和
- 公平性の損失：先着順からの順序遅れの最大値
- $\alpha$ ：最適化項目トレードオフ調整用のパラメータ

ここで、公平性の損失における「先着順からの順序遅れ」は、「先着順方式での出域順序を基準とした、提案方式での出域順序の差」を示す。たとえば、先着順方式で生成された移動体スケジュールでは 2 番目に出域予定であった移動体が、提案方式で生成された移動体スケジュールでは 5 番目に出域予定となった場合は、順序遅れは 3 となる。また、トレードオフ調整用パラメータである  $\alpha$  の値は、大きくするほど、公平性を保つことを考慮する度合いが大きくなる。 $\alpha$  の値が 0 の場合は、総遅延時間のみを考慮した単一指標の最適化となる。なお、公平性の損失は、最大遅延時間や最大コスト（遅延をコストに換算した値）などを用いることも考えられる [1]。本論文では、目的関数に遅延を示す項を含んでいるため、公平性の損失には、先着順から順序が大きく遅れる移動体の発生を防ぐための「先着順からの順序遅れの最大値」を用いるものとした。

## 5. 評価

本章では、提案方式の評価結果を示す。前述のように、提案方式では、スケジュールリング順序を表す移動体の順列を、最適化手法を用いて修正する処理を反復する。以下では、代表的な最適化手法として、メタヒューリスティックで大域的な探索を行う傾向が強い GA を利用した場合と、局所的な探索を行う傾向が強い TS を利用した場合についてそれぞれ評価する。

GA は生物進化に基づく確率的な最適化アルゴリズムの 1 つであり、複数の解（個体群）を保持、その個体群への交叉・突然変異・淘汰の遺伝的操作の適用を反復することにより、解の評価値を向上する方向に個体群を進化させる。

TS は、解を 1 つ保持し、その解を少し変化させた近傍解集合の最良解を選択することを繰り返すことで評価値の良い解の探索を行う。その際に、探索が後戻りして同じ解ばかりを探索すること（サイクリングと呼ばれる）を防ぐため、選択した最近の解の生成方法に関する情報（タブー属性と呼ばれる）をタブーリスト（短期記憶と呼ばれる）に記憶し、一定期間、その属性を持つ解の選択を禁止する。このようにして、局所最適解に陥ることを防ぎ、探索の多様化を図る。TS は、下記の 2 通りの方式を評価する。

- TS1：先着順を初期値として、TS により探索を行う。
- TS2：TS1 と同様、先着順を初期値として、TS により探索を行うが、探索途中で最良解が一定回数未更新となった場合は、初期解（先着順の順列）から探索を再スタートする。

表 1 移動体種別 A, B の入域点到着パターン (シナリオ (b))  
Table 1 Arrival pattern of moving objects (scenario (b)).

入域点到着順序	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
移動体種類	A	A	A	A	A	A	A	B	B	A	B	A	B	A	A
入域点到着順序	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
移動体種類	B	B	B	B	A	B	A	B	A	B	A	B	A	B	B
入域点到着順序	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
移動体種類	B	B	B	B	A	B	A	B	A	B	A	B	A	B	A
入域点到着順序	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
移動体種類	A	B	A	A	B	B	A	A	B	B	A	B	A	B	B

表 2 移動体間の規定時間間隔

Table 2 Required time interval between moving objects.

前方の移動体	後続する移動体	規定時間間隔
A	A	105
A	B	155
B	A	55
B	B	55

以下、評価条件としてシナリオ設定および、GA と TS のパラメータ設定について述べた後に、評価結果を示す。

### 5.1 評価条件

#### シナリオ設定

シナリオ設定を以下に示す。ここでは、我々が扱うアプリケーションにおいて、典型的なシナリオ・パラメータを採用した。なお、時間は任意の単位に基づく値で表す。

- 移動体：60 台
- 移動体の入域点到着時間間隔：90
- 待機場所での待機時間：1 回の待機あたり 240
- 移動体が共有領域の移動に要する時間：830~920 の範囲で調整可能（タイムウィンドウの長さは 90）

また、移動体の到着状況として、以下の 2 つのシナリオ (a), (b) を評価する。

- (a) 1 種類の移動体のみが登場するシナリオ。出域点における規定時間間隔は 125 とする。
- (b) 2 種類の移動体 A, B が登場するシナリオ。移動体 A, B は、入域点に、ランダムに生成した表 1 に示すパターンで到着するものとする。また、出域点における規定時間間隔は、前後の移動体の種類に応じて、表 2 に示す値に設定する。必要な規定時間間隔が大きいほど、移動体の間で遅延が伝播しやすい状況となる。

#### GA のパラメータ設定

- 世代数（最適化ループ数）：1,000
- 1 世代あたりの個体数（解の数）：400
- 初期解集合：「先着順」と「ランダムに生成した順列（複数）」を利用
- 遺伝的操作：
  - ▷ 交叉：順列で表される解に適した代表的な交叉法である循環交叉 (CX: Cycle Crossover) を利用する。

表 3 評価結果 (シナリオ (a))

Table 3 Evaluation Result (scenario (a)).

方式	α の設定値	総遅延時間		公平性の損失			最大遅延時間			
		初期解	GA	TS1	初期解	GA	TS1	初期解	GA	TS1
提案方式	0	83725	66510	64629	1.00	35.26	33.14	2883.0	5199.1	4942.3
	100		66280	63858		8.36	7.71		2801.7	2677.4
	1000		67223	65057		4.25	3.82		2502.0	2420.3
	5000		77810	66672		1.09	2.14		2685.2	2356.1
	10000		78808	67953		1.00	1.92		2696.4	2386.3
先着順方式		139117		0.00			4635.0			
待機最小化方式 <sup>2)</sup>		67160		56.00			7680.0			

方式	α の設定値	評価値(式(1)のF)			収束回数		処理時間[秒]	
		初期解	GA	TS1	GA	TS1	GA	TS1
提案方式	0	83725	66510	64629	951.2	914.5	86.5	50.8
	100	83825	67116	64548	957.3	886.3	69.5	39.9
	1000	84725	71475	68877	895.1	826.8	61.4	35.6
	5000	88725	83264	77372	819.2	650.3	58.7	27.3
	10000	93725	88808	87153	817.6	678.3	59.3	28.5

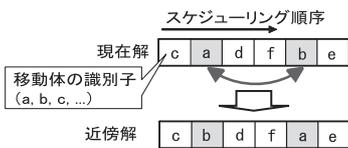


図 6 TS における近傍解生成方法

Fig. 6 Modification of solution in TS.

- ▷ 突然変異：与えられた確率（突然変異率と呼ぶ：本評価では 0.01 に設定）で、逆位を適用する。ここで逆位は、順列で表された解において、ランダムに選ばれた 2 点間の要素の順序を逆転する操作を表す。
- ▷ 淘汰方法：エリート選択とルーレット選択の両方を実施し、選択された個体は次世代へ残す一方、選択されなかった個体は次世代へ残さずに淘汰する。具体的には、まず、次世代へ残す個体を、評価値が良い順に、設定された数（エリート数と呼ばれる：本評価では 30 に設定）だけ選択する（エリート選択）。そして、エリート選択によって選択された個体を除き、次世代へ残す個体数に達するまで、評価値に応じた確率で個体を選択していく（ルーレット選択）。

TS のパラメータ設定

- 最適化ループ数：1,000
- 1 ループあたりの近傍解の生成数：400
- 初期解：先着順
- 近傍解生成操作：以下の方法を適用。
  - ▷ 近傍解生成方法：現在解のスケジューリング順序をランダムに 1 組交換することで生成する。図 6 に、近傍解の生成例を示す。図 6 の場合、近傍解は、現在解のスケジューリング順序の 2 番目 (a) と 5 番目 (b) の交換により生成されている。ここでは、現在解のすべての近傍解を生成するのではなく、あらかじめ設定した数の近傍解をランダムに生成する方法を利用する。

- ▷ 短期記憶の記憶方法：現在解を更新する際に選択した「スケジューリング順序の交換位置の組」を短期記憶に登録する。たとえば、図 6 の近傍解で現在解を更新した場合、「スケジューリング順序の 2 番目と 5 番目の組」を短期記憶に登録する。短期記憶に登録した交換位置の組は、登録してから一定期間（ループ回数で指定され、短期記憶長と呼ばれる：本評価では 20 に設定）が経過すると破棄される。
- ▷ 探索時の短期記憶の利用方法：現在解を更新する際に、「短期記憶に登録されている交換位置の組」によって生成された近傍解（以後、タブー解と呼ぶ）での更新を禁止する。たとえば、図 6 の近傍解で現在解を更新した場合、それから一定期間は、スケジューリング順序の 2 番目と 5 番目の交換により生成される近傍解での更新を禁止する。ただし、タブー解が「今まで探索した解の最良解の評価値」よりも良い評価値を持つ場合、タブー解での更新を許容する。

5.2 評価結果

先着順方式、GA を用いた提案方式、TS1 を用いた提案方式、および、文献 [2] の待機回数が少なくなるように移動体をスケジューリングしていくことで準最適解を得る方式（「待機最小化方式」と呼ぶ）を評価した結果を表 3（シナリオ (a)）、表 4（シナリオ (b)）に示す。これらの表で、提案方式は、公平性の考慮度合いを調整するパラメータである α を 0, 100, 1000, 5000, 10000 の 5 通りに設定して評価した結果を示す。また、提案方式の初期解を評価した結果も示す。提案方式の初期解は、スケジューリング順序を先着順に設定したものであり、メタヒューリスティックを適用する前の解に相当する。なお、提案方式のスケジューラは出域順序を先着順から入れ替え可能であるため、スケジューリング順序が先着順であっても、提案方式の初期解は、出域順序を先着順とする先着順方式とは異なる

表 4 評価結果 (シナリオ (b))  
Table 4 Evaluation Result (scenario (b)).

方式	$\alpha$ の設定値	総遅延時間		公平性の損失			最大遅延時間			
		初期解	GA	TS1	初期解	GA	TS1	初期解	GA	TS1
提案方式	0	27027	8165	7289	2.00	9.38	7.75	965.0	985.5	826.1
	100		8011	7326		5.49	4.66		656.2	602.2
	1000		8149	7709		3.26	3.03		516.7	497.0
	5000		11012	10197		2.00	2.00		517.3	517.0
	10000		11073	10302		2.00	2.00		530.5	518.8
先着順方式		51709		0.00			1530.0			
待機最小化方式 <sup>2)</sup>		23625		53.00			5520.0			

方式	$\alpha$ の設定値	評価値(式(1)のF)			収束回数		処理時間[秒]	
		初期解	GA	TS1	GA	TS1	GA	TS1
提案方式	0	27027	8165	7289	740.6	554.9	31.1	13.1
	100	27227	8561	7792	707.4	534.6	27.8	12.4
	1000	29027	11411	10734	747.5	665.0	25.1	15.2
	5000	37027	21012	20197	715.0	667.8	23.4	14.4
	10000	47027	31073	30302	718.4	662.1	23.7	14.2

るスケジュールになりうる。表 3, 4 では、各方式で得られた移動体スケジュールの総遅延時間、公平性の損失（先着順からの順序遅れの最大値）、最大遅延時間（各移動体に対して求められた遅延時間の最大値）を示す。さらに、提案方式に関しては、解の評価値（式 (1) の値）、収束回数（最良解を得たループ回数）、処理時間（計算機環境 AMD AthlonTM 64 X2 Dual Core Processor 4600+, 2.39 GHz, 2.12 GB RAM）も示す。また、GA, TS は確率的なアルゴリズムであるため、提案方式に関する数値は 100 回試行の平均値を示している。ここで、TS2 を用いた場合の評価結果（100 回試行の平均値）は、TS1 を用いた場合とほぼ同様の結果となったため、掲載は割愛する。しかし、100 回試行の標準偏差は、TS1 と TS2 で異なる結果となったため、TS1 と TS2 に関し、シナリオ (b) の各試行で得た解の評価値の分布を図 7 に示す。なお、シナリオ (a) でも同様の傾向は確認されたが、その効果はシナリオ (b) よりも小さいため、掲載は割愛する。これらの評価結果から、以下の事項が確認できた。

先着順方式と提案方式の比較 (表 3, 4 参照)

- 提案方式は、 $\alpha$  の値にかかわらず、先着順方式と比べ、総遅延時間が短い。また、提案方式で用いた初期解も、先着順方式と比べ、総遅延時間が短い。
- 提案方式は、公平性を考慮しない設定 ( $\alpha = 0$ ) では、先着順からの順序遅れが大きく、遅延時間の大きな移動体が存在しており、公平性が大きく損なわれていた。これに対し、公平性を考慮する設定 ( $\alpha \geq 100$ ) では、 $\alpha$  の増加に従い、公平性が改善された。
- 総遅延時間と公平性損失に関する解の評価値は、提案方式の方が、 $\alpha$  の値にかかわらず、先着順方式よりも優れていた。 $\alpha = 10000$  に設定した提案方式は、解の評価値を、最大で先着順方式の 60%程度に改善した。

なお、GA を用いた提案方式と先着順方式の比較評価を、表 1, 2 の設定が異なる他のシナリオを用いて行った場合も、上記と同様の傾向が確認された [8], [9], [10]。

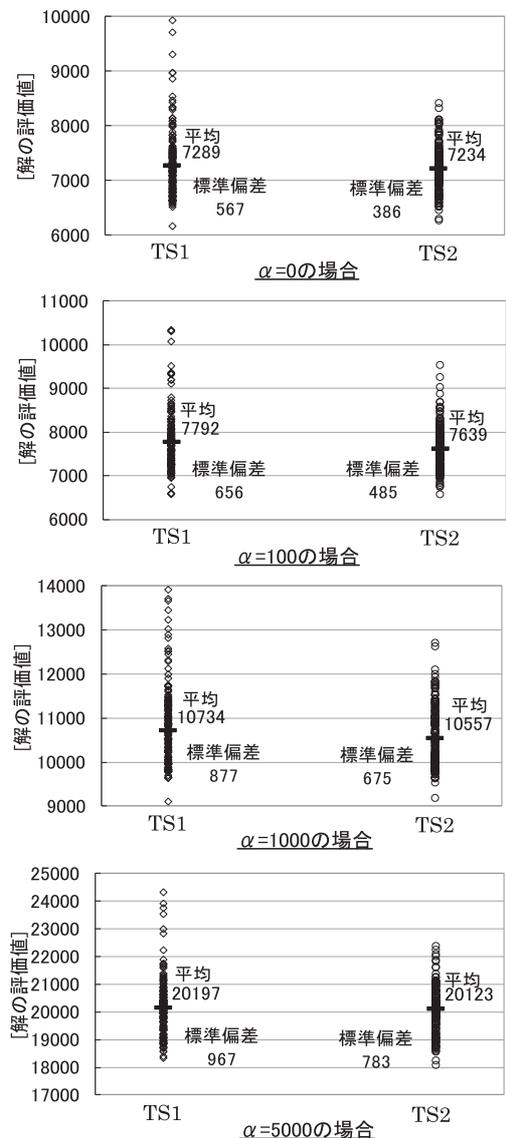


図 7 TS1, TS2 を用いた提案方式で得られた解の評価値の分布 (シナリオ (b))

Fig. 7 Distribution of evaluation values utilizing TS1 and TS2 (scenario (b)).

### 提案方式の初期解と最適化結果の比較 (表 3, 4 参照)

提案方式は,  $\alpha$  の値にかかわらず, 初期解よりも良い評価値となる解を生成した. なお, 提案方式の初期解は,  $\alpha$  の値にかかわらず, 移動体スケジュールの内容 (総遅延時間, 公平性の損失, 最大遅延時間) が同じとなる. 一方, 解の評価値は,  $\alpha$  の値に応じて異なる値となる.

### 提案方式での $\alpha$ の設定値に関する比較 (表 3, 4 参照)

$\alpha = 100, 1000$  における総遅延時間は,  $\alpha = 0$  の場合と大きな差異がない. 一方,  $\alpha = 5000, 10000$  における総遅延時間は,  $\alpha = 0$  の場合と比べ, 最大で 35% 程度増加した.

### 提案方式と待機最小化方式 [2] の比較 (表 3, 4 参照)

- 提案方式は  $\alpha = 0$  であっても, 待機最小化方式 [2] と比べ, 公平性の損失が少なく, 最大遅延時間が短い. この差は, 提案方式が  $\alpha \geq 100$  となるとさらに大きくなる.
- シナリオ (a) では, 提案方式は,  $\alpha$  の値にかかわらず, 待機最小化方式 [2] と総遅延時間に大きな差異がない. 一方, シナリオ (b) では, 提案方式は,  $\alpha$  の値にかかわらず, 待機最小化方式 [2] と比べ, 総遅延時間が短い.

### 提案方式での最適化手法 GA と TS1 の比較 (表 3, 4 参照)

- TS1 は, GA と比べ, 総遅延時間が短く, 公平性の損失および最大遅延時間も, おおむね優れていた. 解の評価値についても, TS1 の方が GA よりも優れており, 最大 10% 程度の改善が確認された.
- TS1 は, 収束回数と処理時間も GA よりも優れていた. なお, GA の交叉法に関して, CX 以外の交叉法も評価したが, 大きな差は見られなかった [7].

### 提案方式での最適化手法 TS1 と TS2 の比較

上述のように, 100 回試行の平均値からは, TS1 と TS2 の性能の差は確認できなかったが, 図 7 より, 探索の収束時に再スタートを行う TS2 の方が, TS1 と比べ, 得られる解の評価値の分散を減少させる効果があることが確認できた.

## 6. 考察

提案方式は最適化手法に GA, TS いずれを用いた場合でも, 公平性を考慮することで, 先着順方式に対し, 最大順序遅れが小さく, かつ, 最大遅延時間および総遅延時間が短い移動体スケジュールを生成できることが分かった. この際, 提案方式は初期解でも, 先着順方式より総遅延時間が短い. これは, 提案方式のスケジューラが出域順序を先着順から入れ替え可能となっており, スケジューリング順序が先着順であっても遅延伝播の解消が起るためである. さらに, 提案方式は, 初期解に対して GA や TS を用いて順序の修正処理を繰り返すことで, 解の評価値を改善できることが分かった. また, 提案方式の処理時間は 90 秒以内であり, 実運用で利用可能であることも確認できた.

提案方式における  $\alpha$  の値の設定は,  $\alpha$  の値がある程度大きくなる (たとえば, 表 3, 4 の場合,  $\alpha = 1000$ ) までは, 公平性を考慮しない  $\alpha = 0$  の場合と比べて, 総遅延時間が同程度でありながら, 先着順からの最大順序遅れが小さく, 最大遅延時間も短い移動体スケジュールが生成された. しかし,  $\alpha$  の値をさらに大きくする (たとえば, 表 3, 4 の場合,  $\alpha = 5000, 10000$ ) と, 先着順からの最大順序遅れがより小さく, 最大遅延時間もより短くなるものの,  $\alpha = 0$  の場合と比べて総遅延時間が大きくなる. このことから, 総遅延時間が大きくなるまでは,  $\alpha$  の値を大きく設定した方が良いといえる. なお, 総遅延時間の増大を許容し,  $\alpha$  の値をどの程度まで大きく設定するかについては, 運用者が「公平性の損失」と「総遅延時間の増加」のトレードオフを考慮して決定するものとする.

提案方式は, 待機最小化方式 [2] と比べ, 最大順序遅れが小さく, かつ, 最大遅延時間が短い移動体スケジュールを生成できることが分かった. ここで, 提案方式は  $\alpha = 0$  であっても, 待機最小化方式 [2] と比べ, 最大順序遅れが小さく, 最大遅延時間が短い. これは, 待機最小化方式 [2] が, 遅延伝播を断ち切るにあたり, いずれかの移動体に待機場所の使用回数が偏りやすい動作となっているのに対し, 提案方式は, 待機場所を使用する移動体を確率的に選択するためであるとする. また, 提案方式は, 待機最小化方式 [2] と比べ, 総遅延時間が同程度または短い移動体スケジュールを生成できることも分かった. 提案方式によるスケジュールが, 待機最小化方式 [2] よりも短い総遅延時間となったシナリオは, 前後の移動体の種類に応じて規定時間間隔が異なるシナリオである. これは, 提案方式が遅延伝播を断ち切るだけでなく, 移動体間の規定時間間隔が短くなるようにスケジュールを最適化した結果と考える.

提案方式で用いる最適化手法は, 本問題に関し, TS の方が GA よりも探索性能 (得られる解の質, 収束回数, 処理時間) が優れていた. この理由を, GA を用いた提案方式で得られた最良解の内容, すなわちスケジューリング順序の観点で述べる. シナリオ (a) について, GA を用いた提案方式で得られた最良解の内容を図 8 に示す. 図 8 の各グラフにおいて, 横軸はスケジューリング順序, 縦軸は移動体番号 (移動体に対して先着順に振られた番号) を表す. 図 8 から分かるように, 最良解に対応するスケジューリング順序は, 先着順に類似した順列になっており, これらの順列の間の類似度は, 公平性の考慮度合い  $\alpha$  が増加するにつれ, 増加している. この傾向は, シナリオ (b) および TS を用いた場合も同様であった. 以上より, 大域的な探索を得意とする GA と比較して, TS の方が先着順に近い順列を集中的に探索することができたため, 本論文で対象とした問題では, TS の方が GA よりも探索性能が優れていたと考える. また, 探索途中で最良解が一定回数未更新となった場合に初期解 (先着順の順列) から探索を再ス

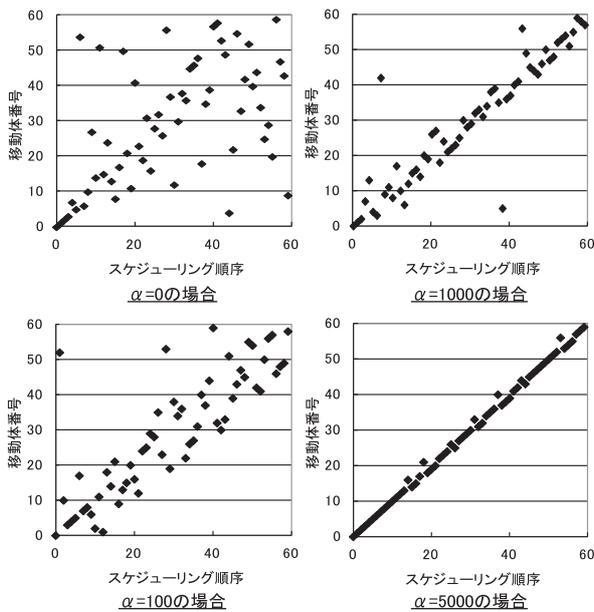


図 8 シナリオ (a) について GA を用いた提案方式で得られた最良解の内容

Fig. 8 Optimal solution obtained by GA (scenario (a)).

スタートする TS2 は、再スタートの機構を組込む前の TS1 と比べ、得られる解の評価値の分散が減少したという結果が確認された。これに関しても同様に、先着順に近い順列を集中的に探索することが重要であったためと考える。

## 7. むすび

本論文では、交通システムにおいて、移動体が一定時間、待機可能な場所がある場合に関し、移動体の順序入れ替えを許容する順序付け方式を提案した。提案方式は、スケジューリング順序を表す移動体の順列を解表現として、最適化手法を適用するものである。最適化の対象とする目的関数は、順列に従って移動体を 1 台ずつ、制約を満たすようスケジューリングした結果から算出される遅延時間および公平性の劣化の度合いに基づいて評価するものである。提案方式における最適化手法として GA を用いた場合と、TS を用いた場合について、我々が扱うアプリケーションの典型的なシナリオ設定で計算機シミュレーションを行った。評価の結果、先着順で移動体を通過させる従来方式、待機回数が少なくなるように移動体をスケジューリングしていく方式 [2] と比較して、提案方式は評価値の良い解が得られることを確認した。また、得られた解を調査した結果、共有領域への移動体の到着順序（先着順）に近い傾向があった。そのため、提案方式における最適化手法としては GA ではなく、先着順の近くを集中的に探索する TS を用いる方が、得られる解の評価値および収束の速さが向上することを確認した。今後の課題は、提案方式を様々なシナリオに対して検証し、その汎用性を確認することである。

## 参考文献

- [1] Soomer, M.J.: Runway Operations Scheduling using Airline Preferences, Doctoral thesis, Vrije Universiteit Amsterdam (2008).
- [2] Roy, K., Bayen, A. and Tomlin, C.: Polynomial time algorithms for scheduling of arrival aircraft, *Proc. AIAA Conference on Guidance, Navigation and Control*, pp.1849-1866 (2005).
- [3] Sait, S.M. and Youssef, H.: *Iterative Computer Algorithms with Applications in Engineering – Solving combinatorial optimization problems*, IEEE Computer Society (1999). 白石洋一 (訳): 組合せ最適化アルゴリズムの最新手法, 丸善 (2002).
- [4] Abela, J., Abramson, D., Krishnamoorthy, M., De Silva, A. and Mills, G.: Computing optimal schedules for landing aircraft, *Proc. 12th National ASOR Conference*, pp.71-90 (1993).
- [5] Beasley, J.E., Sonander, J. and Havelock, P.: Scheduling aircraft landings at London Heathrow using a population heuristic, *Journal of the Operational Research Society*, Vol.52, pp.483-493 (2001).
- [6] Globus, A., Crawford, J., Lohn, J. and Pryor, A.: A Comparison of Techniques for Scheduling Fleets of Earth-Observing Satellites, *Proc. 14th International Conference on Automated Planning and Scheduling* (2004).
- [7] 澤田めぐみ, 白石 将, 尾崎敦夫, 松村寛夫: 移動体の通過順序付けにおける遺伝的アルゴリズムの交叉法の比較評価, *情報科学技術フォーラム講演論文集*, Vol.9, No.1, pp.143-144 (2010).
- [8] 松村寛夫, 白石 将, 澤田めぐみ, 尾崎敦夫: 遅延を最小化する移動体順序付け, *電子情報通信学会 SANE, 宇宙・航行エレクトロニクス*, Vol.109, No.426, pp.35-40 (2010).
- [9] 松村寛夫, 尾崎敦夫, 白石 将, 澤田めぐみ: 最適化手法を用いた移動体の順序付け, *電子情報通信学会総合大会講演論文集* (2010).
- [10] Sawada, M., Shiraiishi, M., Ozaki, A. and Matsumura, N.: Object ordering for delay minimization considering fairness, *SICE Annual Conference 2011*, pp.2888-2892 (2011).



澤田 めぐみ

2008 年東京電機大学大学院工学研究科情報メディア学専攻修士課程修了。同年三菱電機株式会社に入社。同社情報技術総合研究所にて、最適化に関する研究開発に従事。2014 年電子情報通信学会学術奨励賞受賞。電子情報通

信学会会員。



尾崎 敦夫 (正会員)

1988年九州工業大学工学部情報工学科卒業。1990年同大学院電気工学研究科計算機コース博士課程前期修了。同年三菱電機株式会社に入社。以来、並列分散処理およびモデリング&シミュレーションに関する研究開発に従事。

現在、同社情報技術総合研究所・電子システム技術部・センサ処理基盤グループマネージャー。1996年ESS96 (8th European Simulation Symposium) Best Paper Award, 2006年電子情報通信学会コンカレント工学研究会優秀論文賞各受賞。博士(情報工学)。電子情報通信学会, IEEE 各会員。



白石 将 (正会員)

1994年東京大学大学院工学研究科電子工学専攻修士課程修了。同年三菱電機株式会社に入社。同社情報技術総合研究所にて、最適化, データ解析, 並列分散処理に関する研究開発に従事。



松村 寛夫

2006年早稲田大学大学院理工学研究科物理学及応用物理学専攻博士課程修了。同年三菱電機株式会社に入社。同社情報技術総合研究所にて、最適化等の研究開発に従事。博士(理学)。