

# 隊列走行可能な端末交通システムにおける 旅客輸送方式

長谷部 浩二 加藤 和彦 阿部 洋丈 川本 雅之

著者らは、電子連結により複数の車両が隊列走行可能な端末交通システムの研究開発を進めている。本論文では、こうした交通システムを対象として、車両の再編成により乗り換え動作を排した旅客の輸送方式を提案する。特にここでは、運転手の操縦にしたがって隊列の先頭を走行する車両と、先頭車に無人運転で追従する車両の 2 種類で構成されるシステムを考える。本提案手法の基本的なアイデアは、隊列内の車両ごとに目的地を設定し、交通システム内の路線網の各分岐点で目的地までの経路を同一とする先頭車両に後続車両を接続するというものである。また、ここで対象とするシステムでは、すべての旅客の輸送需要が中央の運行管理用サーバにリアルタイムで集約され、その情報をもとに最適化されたな運行スケジュールが動的に決定されることを想定している。しかしながら、このような中央集中型の制御方式では、中央で障害が発生するとシステム全体が停止してしまうという問題がある。本論文では、このような障害が発生した際にも運行サービスを継続させるために、暫定的な運行スケジュールに自動的に切り替える方法を提案する。

## 1 はじめに

2006 年末の「高齢者、障害者等の移動等円滑化の促進に関する法律」(通称「バリアフリー新法」)の施行以降、鉄道などの基幹的な公共交通や大規模施設等における移動の利便性は改善されつつあるが、それらの間をつなぐ端末交通網は未整備のままである。そのため、高齢者や障害者を含むすべての人々にとって高度な利便性を備えた新しい端末交通システムの創出と社会実装が強く求められている。

海外では、すでにいくつかの端末交通システムが実用運行されており、また実用化に向けたデモ走行も各所で行われている。その代表的なものとしては、オランダ・ロッテルダム近郊を走る ParkShuttle[1] やフランス・ラロシェル CityMobil2[2] などが挙げられ

る。しかしながら、端末移動ではしばしば乗降先が多様に分散しているため、小型バスをはじめとする従来型の車両や運行システムでは、すべての旅客を乗降需要のあるすべての停留所に連れ回すという問題が生じる。また、高齢者や障害者は乗降に時間を要するため、それぞれの停留所での滞留時間も長くなり、結果として速達性を大きく喪失するものになってしまうという課題も残されている。

以上を背景に、著者らは、電子連結により複数の車両が隊列走行可能な端末交通システムの研究開発を進めている。その最大の特徴は、運行経路上で随時車列を再編成することにより、乗り換え動作なしで旅客を目的地まで輸送できる点にある。また、乗降需要のある旅客の乗った車両だけが必要な停留所に留まり、他の旅客の移動時間に影響を与えないようにするという利点も備えている。一方、隊列走行可能な車両が端末交通システムにおいて利用された例はこれまでになく、実際にどのような運行方式によって上記の利点が得られるのかについては明らかではない。

そこで本論文では、隊列走行可能な車両による交通システムを対象として、車両の再編成により旅客の乗り換え動作を排した輸送方式を提案する。特にここで

A Passenger Transport Method for Last Mile Transportation with Platooning Vehicles.

Koji Hasebe, Kazuhiko Kato, Hirotake Abe, 筑波大学システム情報系, Faculty of Engineering, Information and Systems, University of Tsukuba.

Masayuki Kawamoto, 筑波大学国際産学連携本部, International Innovation Interface Organization, University of Tsukuba.

は、運転手の操縦にしたがって隊列の先頭を走行する車両（以下で先導車と呼ぶ）と、先導車に無人運転で追従する車両（以下で後続車とよぶ）の2種類で構成されるシステムを想定する。本提案手法の基本的なアイデアは、隊列内の車両ごとに目的地を設定し、システム内の路線網の各分岐点で目的地までの経路を同一とする先頭車両が接続されるように車両を繋ぎ変えるというものである。これにより、経路の分岐点で旅客が乗り換える代わりに、車両を繋ぎ変えることで目的地まで移動が可能となる。

また、本論文で対象とするような端末交通システムにおいては、個々の旅客の移動需要をリアルタイムで把握し、それに基づいて動的に各車両の運行スケジュールを決定することが重要である。通常こうしたスケジューリングは、中央の運行管理用のサーバが必要に関する情報を一元的に集約し、そこから各車両に指示を出すという方法によって実現されることが多い。このような中央集中型の制御方式は、分散管理型の制御方式に比べて実装が容易であるという利点がある。しかしながら、中央のサーバが何らかの原因で機能不全に陥ってしまうと交通システム全体が停止してしまうという問題がある。本論文では、中央の制御システムに障害が発生しても旅客輸送サービスを継続させるために、障害時に各車両に対して暫定的な運行経路を自動で割り当てる方法についても提案する。

本論文の構成は以下の通りである。第2章では、対象とする電子連結車両の概要を示した上で、提案手法である旅客の輸送方式について述べる。第3章では、もう一つの提案手法である中央の運行指令システムの耐障害性向上のための方法を説明する。第4章では関連研究について述べ、最後に第5章で結論と今後の課題について述べる。

## 2 車両の再編成による旅客輸送方式

この章では、電子連結車両の概要とその旅客輸送方式について説明する。なお、この論文では一貫して旅客の輸送を目的とする交通システムに焦点を当てて議論を進めるが、ここで示す手法は、複数の車両が隊列を成して移動するという特徴を有するさまざまな輸送システムに応用が可能である。

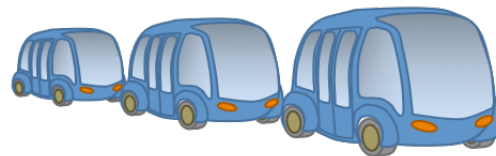


図1 電子連結車両の概念図

### 2.1 電子連結車両の概要

電子連結車両は、物理的な接点を持たずに複数の車両を電子的に制御しながら隊列走行させることが可能な車両である。その概念図を図1で示す。車両はその機能により以下の2つに分類される。

- 先導車：有人で運転しながら後続の車両を牽引する車両
- 後続車：先導車に追従して無人で走行することができる車両

車両の連結は、電子的な制御により任意に繋ぎかえることができ、先導車を先頭にあらかじめ決められた台数までの車両を縦走させることができる。また、いずれの種類の車両についても定員まで旅客を乗車させることができる。

### 2.2 電子連結車両の運行経路ネットワーク

#### 2.2.1 運行経路ネットワークの一般的な形態

ここで対象とする交通システムでは、既存の鉄道やバスのように、旅客の乗降地点である複数の停留所（駅）と、停留所間を結ぶ運行経路があらかじめ設定されている。また、運行経路ネットワーク上には、その地理的な位置によっていくつかのエリアが設けられており、停留所は1つもしくは複数のエリアに属している。

運行経路ネットワークの具体例を図2で示す。それぞれのエリアには複数の停留所があり（図中の丸および四角）、エリアごとにエリア内のすべての停留所を経由する運行経路が設定されている（図中の矢印）。また、図中の四角のように、停留所の中には複数のエリアにまたがって存在するものがある。これを結節点と呼ぶ（この例では、「1」と「2」で示された2つの結節点がある。）結節点は、エリアBのように1つのエリアに複数存在することもある。

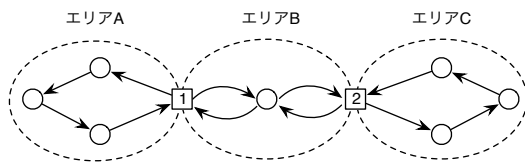


図 2 運行経路ネットワークの例

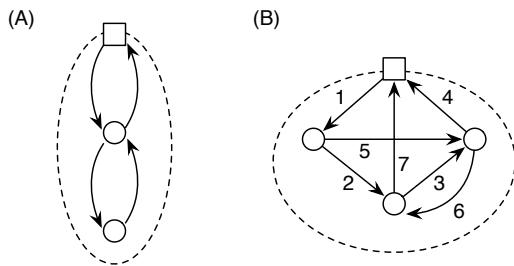


図 3 エリア内の経路の例

エリア内の経路は必ずしも異なる停留所のみを結ぶ必要はなく、例えば図 3 の (A) のように同じ停留所が複数回現れるような経路であってもよい。また、図 3 の (B) のように、エリア内の停留所を巡回する経路は一通りに定まっていなくてもよい。この図の例では、1-2-3-4 の順で周回する経路と、1-5-6-7 の順で周回する経路が存在する。

### 2.2.2 運行経路ネットワークの具体例

ここで運行経路ネットワークの典型例を 2 つ挙げる。1 つは、図 4 の (A) で示すように、複数のエリア内を周回する局所的なルートと、異なるエリア間を接続する幹線ルートの 2 種類の経路によって構成される運行ネットワークである。またもう 1 つは、図 4 の (B) で示すように、住宅地から主要駅の間を輸送するための端末交通ネットワークである。

### 2.3 車両の再編成

先導車両は、結節点で後続の任意の後続車両を切り離すことができる。また、結節点には切り離された無人運転車を一時的に停留させることができるスペース（車両プール）が備えられており、先導車両は接続台数のキャパシティを超えない範囲で車両プールの車両を接続して追従走行させることができる。

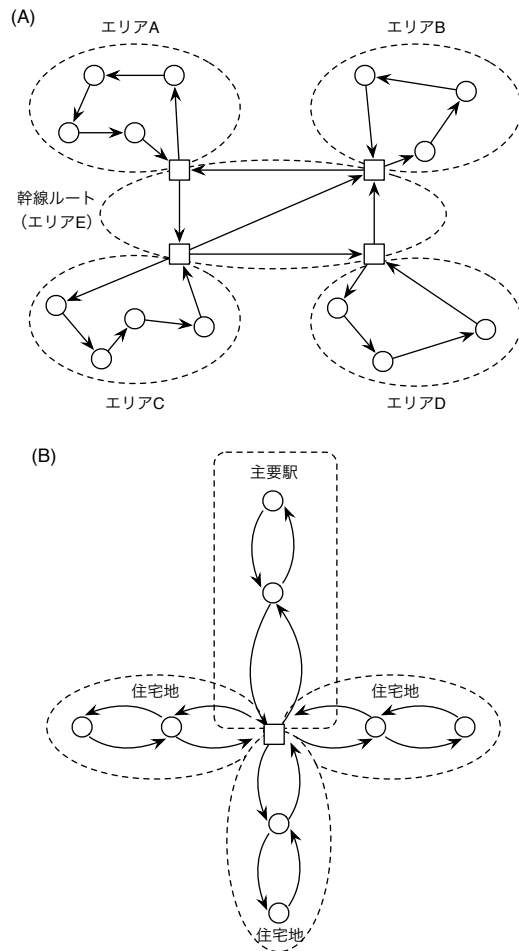


図 4 運行ネットワークの具体例

### 2.4 行先票と旅客の乗降

先導車両および後続車両はそれぞれ行先票を備えており、そこに運行経路ネットワーク内のエリア名のいずれか 1 つを表示することができる。また行先票には「回送」や「満車」など、旅客に乗車できないことを知らせる表示機能も備える。同一の車列にある車両が、それぞれ異なるエリア名を表示してもよい。また、行先票の表示は現在位置が以下の条件を満たすときにのみ変更することができる。

- 現在の位置が結節点である。
- 現在位置するエリアと現在の行先票で示されているエリアが一致している。

ただし、車両が満車になったときは、任意の停留所で

行先票の表示を「満車」に切り替えることができる。

旅客は、ネットワーク内の任意の停留所で乗車することができ、車両によって目的地とする停留所に移動することができる。ここで、旅客が乗車するときは、目的地とするエリアが行先票に表示された車両にのみ乗ることができるものとする。特に、出発地と目的地が同一のエリア X にあるときは、X が表示された車両にのみ乗ることができる。また、行先票が「回送」もしくは「満車」と表示された車両には、誰も乗ることができない。

## 2.5 車両の制御

本論文で提案する輸送方式を実現するために、各車両に対して運行スケジュールの決定と指示を行う必要がある。ここでは中央集中型の制御を想定する。具体的には、システムの中央に車両の運行管理用のサーバ（以下、中央サーバと呼ぶ）を設置し、スマートフォンなどを利用した旅客からの移動需要の収集、旅客の待ち時間の最小化などを目的とした最適な運行スケジュールの決定、各先導車両に対する指示を行う。また、中央サーバに加えて各結節点にもサーバ（以下、結節点サーバと呼ぶ）を設置するものとする。中央サーバで決定された車列の再編成や運行経路は、ネットワーク通信により一旦結節点サーバを介して各車両に伝えられる。

## 2.6 交通システムの具体例

以上で述べた旅客輸送方式の具体例を 1 つ示し、提案手法によって乗り換え動作なしで旅客を目的地まで輸送できることを説明する（図 5 を参照のこと）。

この例では、エリア A, B, C と、それらを繋ぐ幹線ルートからなるエリア D の 4 つからなる運行経路ネットワークを考える。A, B, C の各エリアは地理的に離れていて、それを先導車両が高速に走行する幹線ルートが繋ぐことにより「快速」の役割を果たす。4 つのエリアのそれぞれについて、先導車両は自身のいるエリアの外には出ないものとする。なお、図中で「A」「B」「C」のいずれかの記号が示された四角は車両を表す。このうち灰色の四角は先導車を、白色の四角は後続車を表す。また、それぞれの四角内のエリア

名はその車両が示している行先票のエリアを意味している。

エリア A, B, C において、それぞれの運行経路は複数の停留所を周回する経路によって構成されており、その上を先導車が周回する（ここでは簡単のために各エリアで 1 台のみ考える。また、図中で個々の停留所を省略している。）また、幹線ルートはエリア A, B, C のそれぞれと繋がる結節点のみからなる周回ルートにより構成されており、その上を右回りと左回りの 2 方向で先導車が周回する（ここでは、「R1」～「R3」と記された右まわりの先導車と、「L1」～「L3」と記された左回りの先導車の計 6 台があると仮定する。）

以上の運行経路ネットワークにおいて、以下のような方法で旅客を輸送するものとする。まずステップ 1 において、エリア A, B, C のそれぞれにおいて、先導車の行先票を自身の属するエリアとし、また後続車の行先票を外部のエリアとする。例えばエリア A では、先導車の行先票を「エリア A」とし、後続車の行先票を「エリア B」と「エリア C」とする。各エリアを周回する間、旅客は自身の目的地のあるエリアの行先票を掲げた車両に乗車する。

次にステップ 2 において、各エリアで結節点から出発した車列が 1 周すると、後続車を結節点で切り離す（図では可読性に配慮し、エリア A の結節点での車両の動作のみを示している。）切り離された車両は、幹線ルートを走行する先導車に接続され、それぞれの目的地のエリアに向かう。

ステップ 3 において、目的地のあるエリアの結節点に到着した後続車は切り離され、新たにそのエリアの先導車に接続される（図では幹線ルートの車両を省略している。）

さらにステップ 4 において、後続車は次の客を運ぶために、車両の行先票を外部のエリアに切り替え、現在のエリアを周回する。例えばステップ 3 において、エリア B および C から幹線ルートを経て運ばれてきた後続車は、ステップ 4 において新たに行先票をエリア B および C に切り替える。このとき注意すべきことは、新たに行先票がエリア B および C に変更された後続車に乗車している旅客は、エリア A のどこ

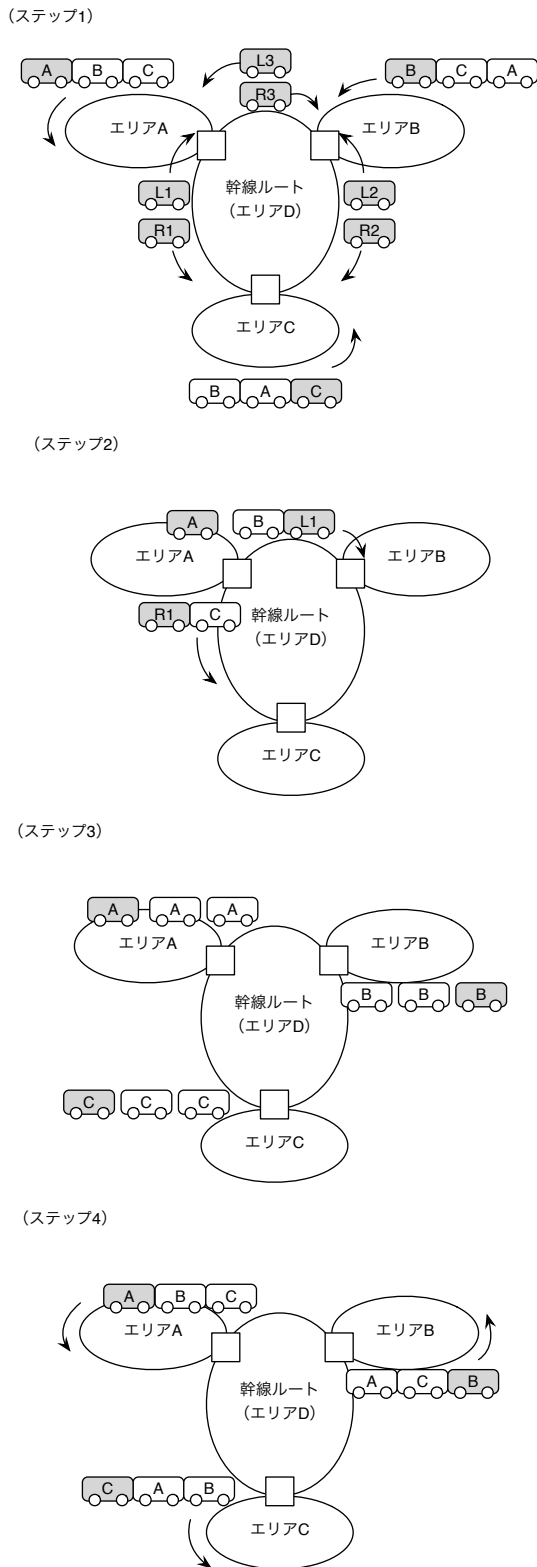


図 5 旅客輸送方式の具体例

かの停留所を目的地としていることである。したがって、エリア A を周回する間に、既に後続車に乗っている旅客を降車させながら、エリア A 内の新たな旅客を乗車させることになるため、効率的に車両を利用することができる。

以上の例で示した通り、旅客は自身の目的と同一の行先票を表示する車両に乗車すれば、乗り換え動作を行わなくても目的地に到着することができる。

### 3 障害発生時の運行経路切り替えの方式

この章では、結節点サーバが中央サーバの障害を検知した際に、先導車に対して障害時用にあらかじめ定められた暫定的な運行経路（以下、暫定運行経路と呼ぶ）に切り替える方法を提案する。具体的には、障害発生時の走行経路（複数あってよい）と、どの経路に何台の先導車を割り当てるのかをあらかじめ決めておくというものである。これにより、中央サーバからの経路の指示が途絶している間も、各先導車は自らに割り当てられた静的なスケジュールに基づいて走行し続けることができる。

これを実現するための最も単純な方法としては、各先導車に対して障害時にどの経路を走行すべきかを事前に決めておくというものが考えられる。しかしながら、交通システムの経路全体が広範囲におよぶ場合、障害発生時の現在地と走行すべき運行経路との距離が大きいと、割り当てられた経路にたどり着くまでに時間がかかってしまうという問題が生じる。

また、この問題を回避するための他の方法としては、障害発生時に結節点サーバが互いに情報交換し合いながら、自らの結節点に到着した先導車に対して動的に経路の割り当てを行うというものが考えられる。しかしながら、障害が結節点サーバ間のネットワークにもおよぶ場合には、結節点サーバ間での通信もできなくなってしまうため、どの経路に何台の先導車を割り当てたかが分からなくなるという問題が残る。

以上の問題を解決しながら目的の経路割り当てを実現するための方法として、本論文では、障害が発生したら各結節点がそこに到着した車両に対して順次割り当てを行うという方式を提案する。ただし各結節点には、どの運行経路に何台の車両を割り当てるのか

があらかじめ決められているものとする。この割り当ての配分を適切に設定することにより、結節点サーバ間で情報交換することなく、システム全体で意図した通りの割合になるよう制御することができる。

### 3.1 障害に関する仮定

以下で、本論文で想定する障害に関する仮定を述べる。

まず、ここで想定するのは中央サーバの故障のみである。特に、中央サーバからの情報がどの結節点サーバにも届かない場合のみを想定する。したがって、中央サーバが一部の結節点サーバのみと通信できるといった部分的な障害は考えないものとする。また、先導車は中央サーバに障害が発生すると、それを直ちに検知できるものとする。さらに、ここで想定する障害は、一度発生するとしばらくはその状態が継続するよなものを考える。

### 3.2 事前の設定事項

障害発生時に備え、以下の1~4を事前に設定する。

1. 障害発生時に先導車に割り当てられる暫定運行経路を設定する。このとき暫定運行経路は以下の2つの条件を満たすものとする。

(ア) 各エリアにおいて、少なくとも1つの暫定運行経路はエリアを周回する。

(イ) 1つもしくは複数の暫定運行経路をたどることにより、任意の2結節点間の一方から出発して他方に至る経路を得ることができる。

2. 各結節点において、結節点に新たな車列が到着した際の後続車の切り替えに関する規則を設定する。すなわち、先導車が次に向かうべき結節点まで牽引すべき後続車を、これまで牽引してきた後続車と車両プールにある後続車の中からどのように選定するのかを決定する規則を設定する。その具体例としては、これまで牽引してきた後続車および車両プールに停留している後続車のうち、乗客の待ち時間が最も長いものから順に牽引可能な上限の台数まで接続するといった規則が考えられる。

3. 各結節点において、結節点に新たな車列が到着した際の行先票の変更方法に関する規則について設定する。ただしここで行先票の変更の対象となる車両は、前述の通り、現在位置する結節点の属するエリアと現在の行先票で示されているエリアが一致している車両のみである。ここで、行先票の変更に関する規則は、以下の条件を満たす必要がある。

(ウ) 交通システム内のどのエリアについても、定期的に行先票に現れる(すなわち、十分な時間が経過してもどの車両の行先票にも表示されないようなエリアは存在しない。)

ここで、先述の1で示した先導車の運行経路に関する2つの条件(ア)および(イ)と、行先票の変更規則に関する条件(ウ)について注意を述べる。これら3つの条件は、当該の障害発生時の運行方式が、交通システムの任意の2停留所間の移動を旅客の乗り換え動作無しで実現可能とするための十分条件となっている。その理由は、まず条件(ア)および(イ)を満たすことにより、旅客に対して十分な数の後続車があれば、車両の接続を適切に切り替えることにより(より具体的には、目的地により近い結節点に向かう先導車に接続・牽引されることにより)、任意の2停留所間を後続車が移動可能となる。さらに条件(ウ)を満たすことにより、各停留所において、どのエリアを目的地とする後続車もいつか必ず訪れるためである。

4. 各結節点サーバに対して、どの経路に何台割り当ててるのかを設定する。このとき、各結節点サーバが割り当てを担当する車両の台数の総和が、システム全体の先導車の台数と一致している必要がある。

以上の1~4の設定のもとで、障害が発生した際には、各結節点で経路の割り当てが行われていない先導車が到着するごとに、自身の担当分が完了するまで経路の割り当てを行う。これにより、結節点間で情報交換することなく意図通りの割り当てが可能となる。

なおここで、あるエリアに車両が偏って存在する場合、ある結節点サーバは自身の担当する割り当てがす

ぐに終わっているのに、別のサーバは先導車が訪れないために割り当てが終わらないことがある。このような場合には、既に割り当てが終わった結節点サーバに先導車が到着したら、まだ割り当てが終わっていない結節点に向かうよう指示するものとする。

### 3.3 結節点サーバによる暫定運行経路の割り当ての手順

中央サーバの障害により運行スケジュールに関する通信が途絶えると、各結節点サーバは障害を検知し、下記の障害発生時の動作モードに移行する。この動作モードにいる間、結節点に先導車が到着する度に、サーバは以下の Step 1 から Step 3 の手順に従って先導車に指示を出し、障害時の暫定運行経路に切り替えさせる。

Step 1. 先導車（以下、 $X$  と呼ぶ）が結節点に到着したら、次の動作を行う。

Case 1:  $X$  にまだ暫定運行経路が割り当てられていないとき

Case 1-1: 結節点サーバが担当分の割り当てを完了していないとき

Case 1-1-1:  $X$  が障害発生前に指示された経路を走行し終わっているとき

Case 1-1-1-1: 結節点サーバの持つ割り当て可能な暫定運行経路のうち、現在地を経路に含むかもしくはその近くを通るものがあれば、そのうちの 1 つを  $X$  の暫定運行経路として割り当てる（Step 2 へ）

Case 1-1-1-2: そうでなければ、 $X$  に対して別の結節点に向かうよう指示する。このとき、どの結節点に向かうのかについては、割り当ての効率を考慮した行き先が事前に決められているものとする（最初に戻って次の先導車の到着を待つ。）

Case 1-1-2:  $X$  が障害発生前に指示された経路を走行し終わっていないとき

Case 1-1-2-1: 結節点サーバの持つ割り当て可能な暫定運行経路のうち、障害発生前に指示された最終目的地を経路に含むかもしくはその近くを通るものがあれば、そのうち

の 1 つを  $X$  の暫定運行経路として割り当てる（Step 2 へ）

Case 1-1-2-2: そうでなければ、Case 1-1-1-2 と同様の処理をする。

Case 1-2: 結節点サーバが担当分の経路の割り当てを既に完了しているときは、Case 1-1-1-2 と同様の処理をする。

Case 2:  $X$  に既に経路が割り当てられているときには、 $X$  に対し現在の経路の走行を継続させる（Step 2 へ）

Step 2. 先導車が現在接続している後続車ならびに結節点に停車している後続車の中から、旅客の待ち時間の長い順に車両を選び出し、それらを牽引するよう先導車に指示を出して、車両を繋ぎかえる（Step 3 へ）

Step 3. これから走行するエリアで客がすべて降車する場合には、あらかじめ決められた方法で行先票を変更する。このときの変更規則は事前に定められており、先述のように「交通システム内のどのエリアについても、定期的に行先票に現れる（すなわち、十分な時間が経過してもどの車両の行先票にも表示されないようなエリアは存在しない）」という条件が満たされているものとする。（最初に戻って次の先導車の到着を待つ）

なお、中央サーバが障害から復旧し、結節点サーバに対して再度運行指令の通信が送られるようになったら、各結節点サーバは結節点に到着した先導車に対して再び中央サーバから送られてきたスケジュールに従うよう指示を出すものとする。

## 4 関連研究

本論文の冒頭でも言及したように、端末交通システムの利便性を向上することは現代社会における重要な課題の一つであり、デマンドバスなどの端末交通に関する研究が数多く行われている。特に、旅客の待ち時間の最小化などを目的とした運行スケジュールの最適化に関するさまざまな手法が、数理最適化などの分野の応用として提案されている。これらは、配送問題[8] もしくはその変形である Dial-a-Ride 問題[5]、もしくはスクールバス経路問題[7] をもとにして

いる。このうち、本論文の交通システムに近いものを扱っている研究としては、複数車両の動的な計画を行うための配送問題を扱ったもの[3]や、旅客の車両間での乗り換えを許す場合のスケジュール最適化手法[6][4]などが挙げられる。しかしながら、これらの研究分野においても隊列走行可能な端末交通システムについては、これまでほとんど扱われてこなかった。

本論文では、具体的な運行スケジューリングの最適化についてまで考察が及ばなかったものの、ここで提案した旅客輸送方式の実用化のために、こうした最適化問題を考えることが重要であるといえる。

## 5 結論と今後の課題

本論文では、電子連結により複数の車両が隊列走行可能な端末交通システムを対象に、車両の再編成によって乗り換え動作を排した旅客の輸送方式を提案した。その基本的なアイデアは、運行経路ネットワーク上の各分岐点（結節点）において、先導車と行き先が同一となる後続車を車列に結合するというものであり、これにより旅客を目的地まで車両ごと移動することが可能となった。また、停留所をいくつかのエリアに分割し、各車両に設けた行先票を結節点で適切に切り替えることにより、旅客の輸送を効率よく行う方法を示した。さらに、中央の運行制御用のサーバの障害発生時に、各車両が暫定的な運行経路に自動的に移行することにより、旅客の輸送サービスを継続する方式を提案した。

今後の課題として最も重要なものの一つは、第4章

でも言及したように、本論文で提案した運行方式に対するスケジュールの最適化手法の開発である。特に、旅客の待ち時間（最短の移動時間と実際にかかった移動時間との差）の最小化や車両の移動コストの最小化などは、実用化を考える上で重要なものと考えられる。

また、本論文では中央の運行管理用のサーバや各結節点のサーバを仮定したが、耐障害性の観点から、こうしたサーバの設置を排し車両間だけで運行スケジューリングを決定する自律分散型の制御方式を考へることも有益であると考えられる。

## 参考文献

- [1] <http://www.2getthere.eu/projects/rivium-grt/>
- [2] <http://www.citymobil2.eu/>
- [3] A. Attanasio, J-F. Cordeau, G. Ghiani, and G. Laporte. Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, vol.30, issue 3, pp.377-387, 2004.
- [4] M. Bögl, K. F. Doerner, S. N. Parragh. The School Bus Routing and Scheduling Problem with Transfers. *Networks*, vol.65, no.2, pp.180-203, 2015.
- [5] J-F. Cordeau and G. Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, vol.153, pp.29-46, 2004.
- [6] C. E. Cortés, M. Matamala, and C. Contardo. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, vol.200, pp.711-724, 2010.
- [7] J. Park and B. Kim. The school bus routing problem: A review *European Journal of Operational Research*, vol.202, pp.311-319, 2010.
- [8] P. Toth and D. Vigo (eds.) *Vehicle Routing: Problems, Methods, and Applications (2nd ed.)*, Society for Industrial and Applied Mathematics, 2014.