エレベータの呼びを導入した自動運転車の運行計画問題

永井　一輝[1]・中西　航[2]・朝倉　康夫[3]

[1]学生非会員　東京工業大学 環境・社会理工学院 土木・環境工学系（〒 152-8552 目黒区大岡山 2-12-1）
E-mail: k.nagai@plan.cv.titech.ac.jp

[2]正会員　東京工業大学 助教 環境・社会理工学院 土木・環境工学系（〒 152-8552 目黒区大岡山 2-12-1）
E-mail: nakanishi@plan.cv.titech.ac.jp

[3]正会員　東京工業大学 教授 環境・社会理工学院 土木・環境工学系（〒 152-8552 目黒区大岡山 2-12-1）
E-mail: asakura@plan.cv.titech.ac.jp

本研究では，事前予約が不要で運行可能な路線が限定されているといった条件下で自動運転車を最適に走行させる方法として平面エレベータを提唱する．その運行計画を求める問題の中で，1次元の路線上を車1台で運行する場合に着目して，最適化問題の定式化と解法を考案した．提案した3種類の解法の特性を理解するために，仮想的な条件設定の下で運行シミュレーションを行った．その結果，計算時間，客の待ち時間，乗車時間を考慮した場合，通常のエレベータの運行方式であるSelective Collectiveが非常に安定した方式であることが明らかになった．

Key Words : *Single Vehicle Horizontal Elevator, Vehicle Routing Problem, Autonomous Vehicle, Elevator Call, Horizontal Elevator Operation Problem*

1. はじめに

自動運転車は種々の交通問題を解決することが期待されており，近年大きな注目を集めている．多くの研究では，自動運転車は任意の道路ネットワークを自由に走行できる状態，すなわちSAE International [1]の定義ではレベル4または5相当が想定されている．しかし，井坪ら[2]による自動運転車の実証実験の結果分析によれば，自動走行中に手動介入が必要な状況が数多く発生している．そのため，それらのレベルでの自動運転の本格的な導入にあたっては，自動運転車が円滑に走行可能な道路空間の確保等，念入りな準備の必要性が示唆されている．ゆえに，現段階では，自動運転車が走行可能な路線を事前に限定した運用がより実現可能性の高い方法であると考えられる．路線が与件であるということは，車両が停車する停留所の集合とその隣接関係が与件ということであり，自動運転車両は停留所を順にめぐっていく．ただし，移動する停留所の順序はあらかじめ決まっているわけではなく，需要に応じてその順序が動的に決められる．

また，自動運転車の運用方法に関する既往研究の大半では，利用者の事前予約が想定されている．しかしながら，我々の日常生活では事前予約は必ずしも利便性が高いとはいえない．たとえば，外出の必要性や意欲が逐次発生するために事前に交通手段の予約が難しい場合や，病院の受診や店舗での買い物のように終了時刻，すなわち移動したい時刻が不確実な場合はしばしば発生する．ゆえに，事前予約を必要としない自動運転システムのほうが利便性が高い場合もあると考えられる．

事前予約が不要で路線は固定されているが停留所での停車の順序は利用者の呼びに応じて決まる既存の交通システムとしてエレベータが挙げられる．そこで，本研究では自動運転車の運用方法の1つとして，上記の特徴を満たす運用である平面エレベータ(Horizontal Elevator)を提唱し，その運用方法を最適化問題として定式化するとともに数値計算により解法の特性を調べることを目的とする．

2. 平面エレベータ運行計画問題

(1) 平面エレベータ(Horizontal Elevator)

平面エレベータ(以降，HE)を，路線が限定された道路ネットワーク上を走行する自動運転車(以降，車)を用いてエレベータを模倣した運行を行う交通システムと定義

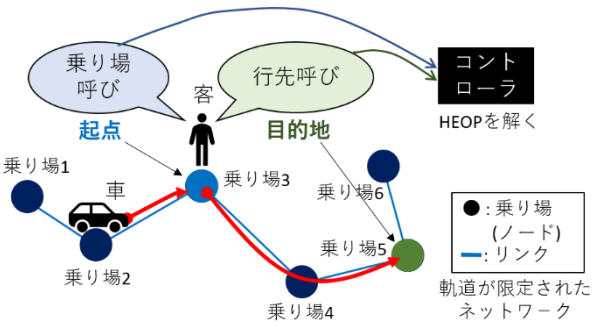

図-1　平面エレベータ(HE)のイメージ

する(図-1)．このシステムにおいて，利用者(以降，客)は，自身の起点となる停留所(以降，乗り場)で乗り場呼び(ボタンを押して起点をHEシステムに伝えること)と，行先呼び(ボタンを押して目的地をHEシステムに伝えること)を登録する．車は，客の呼びに応じて起点の乗り場で客を乗せ，目的地の乗り場まで輸送する．客の乗り場呼びの登録から乗車までの時間を待ち時間，その間の客の状態を待ち状態と定義する．同様に，客の乗車から降車までの時間を乗車時間，その間の客の状態を乗車状態と定義する．さらに，待ち時間と乗車時間の和(HEシステムの利用に要した時間)をシステム時間と定義する．また，トリップは，車がある地点から呼びを1つ処理できる乗り場へ移動することとして定義する．ただし，一度のトリップにおいて処理可能な呼びは1つである．同一の乗り場で複数個の呼びを処理する際には，最初1個の呼びが処理されたトリップの直後に，所要時間ゼロのトリップを残った呼びの個数回分だけ行うものとする．

(2)　平面エレベータ運行計画問題

平面エレベータ運行計画問題(Horizontal Elevator Operation Problem，以降，HEOP)はHEの車の運行予定の経路(以降，計画経路)を求めるための問題であり，組合せ最適化問題の一種である．

a)　問題設定

本研究においては以下の前提でHEOPを解くこととする．時刻tに新たな客の呼びが発生したときに，その呼びを含めた時刻tでシステムに存在する客($k = 1, 2, \ldots, N_s$; N_sは時刻tにおいて待ち状態または乗車状態の客数)の呼びを処理する順序を求める． HEの車は1台で，1次元ネットワーク上を走行するものとする．また，時刻は離散時刻で表され，客は乗り場呼びと行先呼びを同時に登録するものとする．事前予約を必要としないため，客の起終点や発生時刻はその客の呼びの発生まで未知である．したがって，HEシステムの状態(車の位置および発生中の呼び)は時刻の経過とともに動的に変化する．ゆえに，呼びが発生する度に瞬時に問題を解く必要があるため，

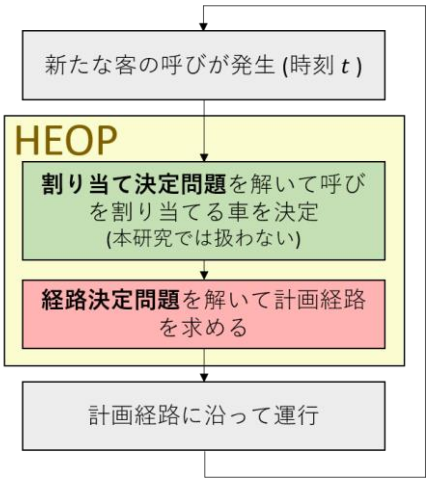

図-2　HEの運行におけるHEOPの位置づけ

計算の即時性はHEOPにおいて重要な要素である．

b)　決定変数

HEOPの決定変数は，発生中の呼びをいずれの車に割り当てるか，および各車が割り当てられた呼びをどの順序で処理するかである．ただし，本研究においては車が1台のHEを仮定しているため，決定変数の前者は考慮されない．ゆえに，後者の決定(以降，経路決定問題)のみを扱う．HEOPについて以上の事柄を図-2にまとめて示す．

c)　目的関数

エレベータ運行計画問題では，一般的に，待ち時間などの客の不満足度と，消費電力量などの運行コストが最小化される．本研究においては下式(1)で表されるように，待ち時間と乗車時間の重みづけ和を目的関数として設定する．

$$\sum_{k=1}^{N_s} [\alpha_1 WT_k + \alpha_2 RT_k] \qquad (1)$$

ただし，WT_k, RT_kはそれぞれ客kの待ち時間，乗車時間とし，α_1, α_2は非負実数の重みづけパラメータである．

この目的関数では待ち時間と乗車時間のみが評価されるが，他の要素も考慮することが可能である．例えば，長時間の待ち状態や迂回の多発は待ち時間および乗車時間の増加以上に客の不満足度を増加させると考えられる．また，車の混雑の度合い等も客の不満足度を増加させる要因として挙げられる．このような事柄は実用化の段階においては目的関数に加えることで幅広く考慮する必要がある．

3. 平面エレベータ運行計画問題の解法

本研究では，経路決定問題の解法として，Selective Collective(以降，SC)，Dynamic Programing for Dynamic Cases(以降，DPDC)，All Route Search Optimization(以降，ARSO)の3方式を提唱する．それぞれ，計算の負荷と最適性において異なる特徴を有する．

(1) Selective Collective (SC)

SCはエレベータの運行に用いられているヒューリスティクスの一種である．SCの運行では，まず，(a) HEシステムが発生中の呼び，車の現在地および進行方向を認識し，(b) 車の進行方向前方で発生中の呼びのみを選択する．次に，(c) 車は選択された呼びに対して現在地から近い順に応答していき，(d) (b)で選択された呼びをすべて処理し終わったら反対方向に方向転換するという手順を繰り返すことによって計画経路が定まる．車は逆走(乗車中の乗客の旅行方向と反対方向に走行すること)が禁止されており，一度方向転換をした場合，新しい進行方向前方の呼びをすべて処理し終えるまで再度の方向転換が不可能である．また，SCで計画経路を求める場合は探索経路数が1であるため，計算負荷が非常に小さく，計算の即時性が期待出来る．

(2) Dynamic Programing for Dynamic Cases (DPDC)

DPDCは動的計画法(Dynamic Programing，以降，DP)を用いた経路決定問題の解法であり，Psaraftis (1980, 1983)[3][4]のDial-a-Ride ProblemのDPを用いた解法をHEOP向けに応用したものである．以下のa) - e)の5段階の手順を経て計画経路が決定されるが，その際にHEシステムの状態は(2)式で定義される状態ベクトル $\boldsymbol{s}$ を用いて表される．状態ベクトル $\boldsymbol{s}$ は，発生中の任意の呼び e と N_s 人の客の状態 $s_k\ (k = 1, 2, \dots, N_s)$ の $N_s + 1$ 個の成分から構成される．

$$\boldsymbol{s} = \left(e, s_1, s_2, \dots, s_{N_s}\right) \tag{2}$$

第1成分 e は，一度のトリップで処理される呼びを示しており，第2成分以降はその呼びの処理が完了した時点での各客の状態を示している．なお，客 k の状態を表す成分 s_k は，$s_k = 3$ のときは客 k が待ち状態，$s_k = 2$ のときは乗車状態，$s_k = 1$ のときは目的地に到着して降車済みであることを表す．

a) 現在状態の特定

新たな呼びが発生した時刻 t における状態を現在状態と定義すると，現在状態はベクトル $\boldsymbol{s_0}$ によって表される．

$$\boldsymbol{s_0} = \left(l_0, s_{1,0}, s_{2,0}, \dots, s_{N_s,0}\right) \tag{3}$$

ただし，l_0 は車の現在地をHEシステムに伝える呼び，$s_{k,n}$ は計画経路のトリップ n 完了時の客 k の状態 $(n = 0, 1, 2, \dots, N_c$，ただし N_c は時刻 t に発生中の呼びの個数) を表す．

b) 実現可能状態の特定

呼び $e(N_c$ 通り)，客の状態 $(s_1, \dots, s_k)(3^{N_s}$ 通り) の全ての組合せを生成し，状態ベクトルを全 $N_c \cdot 3^{N_s}$ 通り生成する．これに現在状態 $\boldsymbol{s_0}$ を加えた $N_c \cdot 3^{N_s} + 1$ がDPDCにおける状態の最大探索数である．生成された状態のうち以下の条件式(4)-(6)を満たすもののみが実現可能な状態であり，そうでないものは実現不可能な状態として削除される．

$$s_k \le s_{k,0} \tag{4}$$

$$\begin{cases} s_k = 2 & (\text{if} \quad e = h_k) \\ s_k = 1 & (\text{if} \quad e = c_k) \end{cases} \tag{5}$$

$$\begin{cases} G \le Q & (\text{if} \quad e \in H) \\ G \le Q - 1 & (\text{if} \quad e \in C) \end{cases} \tag{6}$$

ただし，h_k, c_k, H, C, G, Q はそれぞれ，客 k の乗り場呼び，行先呼び，発生中の乗り場呼びの集合，行先呼びの集合，状態 $\boldsymbol{s}$ にて乗車状態の客数，車の定員である．

c) 実現可能状態の分類

実現可能と特定された任意の状態 $\boldsymbol{s}$ が何番目のトリップ完了時点の状態を表すものであるかを特定する．B_0 を現在状態 $\boldsymbol{s_0}$ の全客の状態成分 $s_{k,0}$ の総和，B を状態 $\boldsymbol{s}$ の全客の状態成分 s_k の総和(ともに $k = 1, \dots, N_s$) とする．このとき，トリップ番号 n は $B_0 - B$ によって求められ，$B_0 - B = n$ となる状態 $\boldsymbol{s}$ はトリップ n 完了時に実現可能な状態は状態空間 $\boldsymbol{S_n}\ (n = 0, 1, 2, \dots, N_c)$ に属する．これにてトリップ番号 n が特定されたため，これ以降の状態ベクトル $\boldsymbol{s}$ の呼び成分は e ではなく $l_n\ (n = 0, 1, 2, \dots, N_c)$ で表すものとする．

d) 実現可能トリップの特定

計画経路は $\boldsymbol{s_0}, \boldsymbol{s_1}, \boldsymbol{s_2}, \dots, \boldsymbol{s_n}, \dots, \boldsymbol{s_{N_c}}$ のような状態遷移の連鎖で表される．ただし，$\boldsymbol{s_n}$ は計画経路のトリップ n 完了時の状態ベクトル $(n = 0, 1, 2, \dots, N_c)$ であり，現在状態 $\boldsymbol{s_0}$ と同様に(7)式のように表せる．

$$\boldsymbol{s_n} = \left(l_n, s_{1,n}, s_{2,n}, \dots, s_{N_s,n}\right) \tag{7}$$

1回のトリップは1つの状態遷移と同値である．任意のトリップ n での状態遷移が実現可能であるか否かに関する条件式を次のように定義する．状態 $\boldsymbol{s} = \left(l_{n-1}, s_1, s_2, \dots, s_{N_s}\right)\ (\in \boldsymbol{S_{n-1}})$ から，次の任意の状態 $\boldsymbol{s'} = \left(l_n, s'_1, s'_2, \dots, s'_{N_s}\right)(\in \boldsymbol{S_n})$ への状態遷移は，

$$s'_k = \begin{cases} s_k - 1 & (\text{if } \ l_n = h_k, c_k) \\ s_k & (\text{otherwise}) \end{cases} \ (k = 1, 2, \dots, N_s) \quad (8)$$

を満たすとき実現可能である$(n = 1, 2, \dots, N_c)$.

e) 最適化

U_n を計画経路の現在状態$\boldsymbol{s_0}$からトリップn完了時の状態 $\boldsymbol{s_n}$ に至るまでの目的関数値とする．また，$\tau(l_{n-1}, l_n)$，g_n およびG_n をそれぞれ，呼びl_{n-1} の乗り場から呼びl_n の乗り場までの車での所要時間，トリップn における待ち状態の客数，同乗車状態の客数とすると，DPにおける部分問題は (9), (10) 式のように定義される．

$$U_n = \begin{cases} 0 & (n = 0) \\ \min\limits_{s \in \boldsymbol{FS_n}} [\tau(l_{n-1}, l_n) \cdot (\alpha_1 g_n + \alpha_2 G_n) + U_{n-1}] & \\ & (n = 1, 2, \dots, N_c) \end{cases} \quad (9)$$

$$\boldsymbol{s_n} = \arg\min_{s \in \boldsymbol{FS_n}} [\tau(l_{n-1}, l_n) \cdot (\alpha_1 g_n + \alpha_2 G_n) + U_{n-1}] \quad (n = 1, 2, \dots, N_c) \quad (10)$$

上式において $\boldsymbol{FS_n}$ は状態 $\boldsymbol{s_{n-1}}$ から遷移可能な状態 $\boldsymbol{s} \in \boldsymbol{S_n}$ の空間を表す．また，$\tau(l_{n-1}, l_n) \cdot g_n$ および $\tau(l_{n-1}, l_n) \cdot G_n$ の項はそれぞれ，トリップn における待ち時間の総和，乗車時間の総和を意味する．

(3) All Route Search Optimization (ARSO)

ARSOは経路決定問題の解法の1つで，経路を全探索することで最適解を求める方式である．現状で既知の情報に基づいた最適解が得られる．呼びの順列で呼びを処理する順序(すなわち経路)を表す．以下の3段階の手順を経て計画経路が決定される．

a) 現在状態の特定

車の現在地，発生中の呼びの特定が行われ，その結果が現在状態としてHEシステムに把握される．

b) 全実現可能経路の生成

現在状態で特定されたN_c 個の呼びの順列を全$N_c!$ 通り生成する．この値がARSOにおける最大探索経路数である．これらのうち，客 k $(k = 1, \dots, N_s)$ の乗り場呼びと行先呼びの順序が逆転している経路，および車の定員を上回る状態が発生する経路を表す順列は実現不可能であり，削除される．残った順列は実現可能な経路を表し，その経路は実現可能な経路の集合R に属する．

c) 最適化

$V_{r,n}$ を任意の経路 r $(\in R)$ の現在状態からトリップn 完了時の状態に至るまでの目的関数値とする．また，経路r のトリップn 完了予定時刻を$T_{r,n}$ とすると，$V_{r,n}$ は (11) 式のように定義され，計画経路r^* は (12) 式のように決定される．なお，$T_{r,n}$ は (13) 式のように定義される．ただし，$L(l_n), RTS, TS$ はそれぞれ，呼びl_n の乗り場，時刻t における車の残り停車時間，車の乗り場での毎回の停車時間を表す．

$$V_{r,n} = \begin{cases} 0 & (n = 0) \\ \min\limits_{r \in \boldsymbol{R}} [(T_{r,n} - T_{r,n-1}) \cdot (\alpha_1 g_n + \alpha_2 G_n) + V_{r,n-1}] & \\ & (n = 1, 2, \dots, N_c) \end{cases} \quad (11)$$

$$r^* = \arg\min_{r \in R} V_{r,N_c} \quad (12)$$

$$T_{r,n} = \begin{cases} t & (\text{if } \ n = 0) \\ T_{r,0} + 1 & (\text{if } \ n = 1,\ L(l_0) = L(l_1)) \\ T_{r,0} + RTS + \tau(l_0, l_1) & (\text{if } \ n = 1,\ L(l_0) \neq L(l_1)) \\ T_{r,n-1} & (\text{if } \ 2 \leq n \leq N_c,\ L(l_{n-1}) = L(l_n)) \\ T_{r,n-1} + TS + \tau(l_{n-1}, l_n) & (\text{if } \ 2 \leq n \leq N_c,\ L(l_{n-1}) \neq L(l_n)) \end{cases} \quad (13)$$

4. 数値実験例

前章で述べた3つの方式を用いて，**図-3** のような1次元ネットワーク(各リンクに添えられている数字は所要時間)を有するHEの運行シミュレーション実験を行った．シミュレーション実験では，客は確率的に発生し，車はHEOPで得られた計画経路に沿って運行する．客は (14) 式のようなポアソン分布に従って発生するものとする．ただし，κ は非負整数，$\lambda_{o,d}$ は乗り場o から乗り場d へ移動する客の100タイムステップあたりの平均発生数，$P_{o,d}(\kappa)$ は同期間中に乗り場 o から乗り場 d へ移動する客がκ 人発生する確率を表している．

$$P_{o,d}(\kappa) = \frac{\lambda_{o,d}^{\kappa} \exp(-\lambda_{o,d})}{\kappa!} \quad (14)$$

客の発生時刻は計画期間内において一様分布に従って決定される．また，100タイムステップあたりの平均客数はN_{p-ave} $(= \sum_o \sum_d \lambda_{o,d})$ で表される．さらに，より実状況に近づけるために，**図-3**のように各交通流に平均客数の配分を行う．ここで交通流は incoming，outcoming，inter-flow の3種類を定義する．それぞれ，主要乗り場から他の乗り場への移動，他の乗り場から主要乗り場への

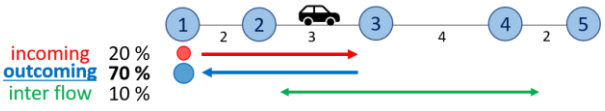

図-3 HEのネットワークと乗客数の配分

移動，主要でない乗り場間の移動を意味する．これは，中心市街地と周辺の住宅地のような状況を想定していることと同値である．以下の実験では，主要乗り場を乗り場1に設定し，客数の配分を incoming : outcoming : inter-flow = 2 : 7: 1 とする．その他の数値は，計画期間長 $T_p = 500$，車の定員 $Q = 20$，停車時間 $TS = 2$，パラメータ $\alpha_1 = \alpha_2 = 1$，車の初期位置は乗り場 1 とする．

ここで，上記入力事項のもと生成した1つの計画期間中の全客のODと発生時刻の組合せのことを1つの交通パターンと定義する．$N_{p-ave} = 5.0, 7.5, 10.0, 12.5, 15.0, 20.0, 25.0, 30.0$ の各 N_{p-ave} に対してそれぞれ10個の交通パターンを作成し，各交通パターンに対してシミュレーション実験を行った．その結果の平均値を図-4 に示す．なお，$N_{p-ave} > 15.0$ (DPDC) と $N_{p-ave} > 7.5$ (ARSO) に関しては計算の規模が非常に大きくなったため，計算不可能であった．

待ち時間(以降および図-4中にて，WT)に関しては，DPDCが最も良い結果となった．これは，DPDCの運行では Anti-SC Turn(以降，ASCT)と Reverse Pickup(以降，RP)が可能であることと，DPDCの定式化の性質上，車は呼びが発生している最も近い乗り場を訪れる傾向があるからである．ASCTはSCの運行に反する方向転換，RPは車がある乗り場で客を乗車させた直後にその客の旅行方向とは反対方向に出発してしまい迂回が生じる乗車のことである．これらはSCの運行では許容されないがDPDC, ARSOの運行では許容される．したがって，限られた状況でしか方向転換できないSCの運行ではWTは大きくなるという妥当な結果が得られている．また，ARSOのWTはDPDCとほぼ同値の結果となった．これは，DPDCと同様にASCTとRPが許容されていることと，DPDCよりもWTの分散が小さく，長時間の待ちの発生が少ないことに由来している．

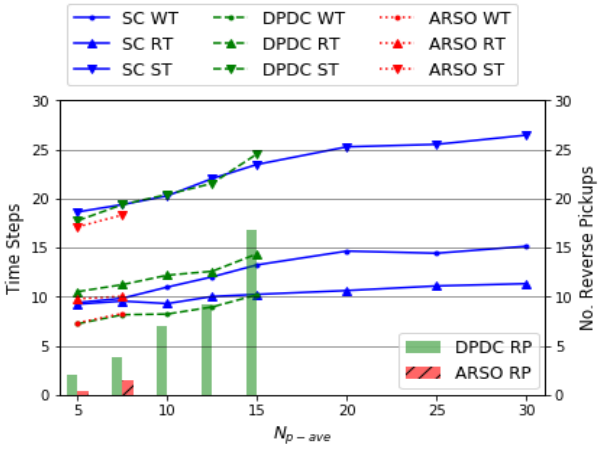

図-4　シミュレーション実験の結果

乗車時間(以降および図-4中にて，RT)に関しては，SCが3方式の中で最も良い結果となった．SCでは進行方向に沿って客を乗せているので迂回が生じず，乗車時間の増加は途中停車回数の増加にしか依存しない．一方，ARSOのRTもSCに匹敵する結果となった．一方，DPDCはRTがやや大きくなっている．これは先述の傾向によって迂回が生じやすいことに起因している．

システム時間(以下および図-4中にて，ST)に関しては，ARSOが最も小さく，DPDCは N_{p-ave} が小さい場合においてSCに劣らない結果となった．ただし，DPDCとARSOは客数が増えると計算が不可能になったため，需要が大きい状況下での実用には向かないと言える．

ここで，RPについて詳しく述べる．シミュレーション実験の結果より，RPには，効率的RPのみに該当するもの，非効率的RPのみに該当するもの，それら両方に該当するもの，いずれにも該当しないものの4種の性質のものが存在していることが分かった．効率的RPは，両方向の客を同時に乗降させることで停車回数をSCの運行よりも少なくすることができる現象である．これに対し，非効率的RPは，ある客Aの乗車がRPに該当し，かつ既に乗車済みの客Bが客Aよりも先に降車する際に発生し，客BのSTを増加させてしまう現象である．ARSOのRPの回数はDPDCの半分以下であり，発生したRPのほとんどは，非効率的RPのみに該当するもの以外のRPである．

以上のことから，STの増加を抑えた運行を行うためには，ASCTとRPを利用してWTの増加を抑えながら，効率的RPを行って非効率的RPを行わない運行が望ましいことが分かった．そのような運行計画を求める解法は，SCを基にASCTと効率的RPを許容した解法，DPDCの部分問題を1トリップ単位ではなく複数トリップ単位で解く方法等が考えられるが，前者はSCよりも，後者はDPDCよりも計算の負荷が大きくなるため，それらの手法がHEの運行により適したものであるとは限らない．

5.　結論

本研究では，事前予約を不要とし，運行可能な路線を限定することで利便性および実現可能性を高めた自動運転車の運用方法として平面エレベータ(HE)を提案した．さらに，平面エレベータ運行計画問題(HEOP)の解法を3種考案し，HEの運行シミュレーション実験を行った．その結果，次のことが明らかになった．第一に，SCはSTに関して，全探索を行うARSOに極端に劣るわけではなく，計算の負荷・即時性の面で安定した優秀な方法である．第二に，DPDCは動的計画法(DP)を用いたものの，

客数(N_{p-ave})が大きくなると結局は計算が不可能になってしまう．第三に，車1台が1次元ネットワーク上を走行するHEの運行においてはSCで良いが，Anti-SC Turn (ASCT) および Reverese Pickup (RP)を上手く取り入れることでシステム時間(ST)を短縮可能な低計算負荷の解法が考案できればより良いと言える．

本研究では，車1台が1次元ネットワーク上を走行するHE (1-dimensional Single Vehicle Horizontakl Elevator) を想定した．しかしながら，提唱した手法を実状況に応用するためには，車が複数台のHEOP (Multi-Vehicle HEOP)，および2次元ネットワーク上での運行方法の考案が必要である．そのため，今後の課題として，本研究で取り組んだ経路決定問題のみならず，新しく発生した呼びをどの車に割り当てるかを決定する割り当て決定問題も解く必要がある．また， SCを2次元ネットワークにそのまま利用することはできないため，SCのような計算負荷の低い手法を考案する必要がある．

参考文献

1) SAE International : Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicle, SAE International J3016, 2016.
2) 井坪慎二，馬渡真吾，岩里泰幸，関谷浩孝，澤井聡志：実証実験を通じた中山間地域における自動運転の課題と対応についての分析，第 60 回土木計画学研究発表会・講演集， Vol. 60， 2019.
3) Psaraftis, H. N. : A Dynamic Programing Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem, Transportation Science, Vol. 14, No. 2, pp. 130-154, 1980.
4) Psaraftis, H. N. : An Exact Algorithm for the Single Vehicle Many-to-Many Dial-A-Ride Problem with Time Windows, Transportation Science, Vol. 17, No. 3, pp. 351-357, 1983.

(2020. 3. 8 受付)

AN OPERATION PROBLEM OF AUTONOMOUS VEHICLE WITH ELEVATOR CALLS

Kazuki NAGAI, Wataru NAKANISHI and Yasuo ASAKURA