機械式立体駐車場出庫スケジューリング*

阿瀬 始**・茨木 俊秀***・柳浦 睦憲***

Removal Scheduling of a Multi-Story Mechanical Parking Facility*

Hajime ASE**, Toshihide IBARAKI*** and Mutsunori YAGIURA***

Multi-story mechanical parking facilities are becoming popular in crowded cities, as they are space efficient. Each floor of such a facility consists of two rail-mounted automated carriers on a pair of rails running from one end to the other, and rows of cells along the rails, where each cell can accommodate one car. The number of rows varies from one to four. There are two elevators connecting all floors, which are dedicated to removal and placing of the cars, respectively. When many cars are waiting to enter and to leave, it is important to schedule the movement of carriers so that the required time is minimized. As a basic subproblem for an efficient scheduling of the total system, we consider here the problem of finding an optimal schedule when an ordered set of car removals is given. We present here an exact algorithm based on dynamic programming, which shows the problem can be solved in polynomial time even if it is a quite complicated combinatorial problem. Some computational results of nontrivial sizes are presented.

1. はじめに

車の数が非常に多い日本、とくに大都市等では、特定の目的のために車が集まる場所・地域には、数百台以上が収納できる大規模駐車場が必要となる。駐車場には大別して自走式と機械式¹⁾がある。土地の広さに対する収納台数という点では機械式が優り、大規模駐車場に適している。ただし、機械式では台車やエレベータの運行管理の巧拙が作業効率に大きく影響するので、運行管理のためのスケジューリングが重要となる。

本論文で考察するのは、Fig. 1に示すような機械式立体駐車場の出庫スケジューリングである。Fig. 1の駐車場は多階からなり、それぞれ一つずつある入庫口および出庫口は、それぞれ入庫エレベータおよび出庫エレベータにより各階と結ばれている。各階には台車が走るレー

Fig. 1 Mechanical multi-storied parking facility

ンの両側に、駐車室が一つずつレーンに沿って配置されている（ただし、入庫および出庫エレベータ室になっている室には車は駐車できない）。この型の駐車場を単列型駐車場という。レーン上には台車が2台あり、レーン上を自由に移動し、駐車室とエレベータ室間の車の出し

* 原稿受付 1998年8月10日

** NKK (株) NKK Corp.; 2-1 Suehiro-cho, Tsurumi ward, Yokohama city, Kanagawa 230-8611, JAPAN

*** 京都大学 大学院 情報学研究科 Graduate School of Informatics, Kyoto University; Yoshida Honmachi, Sakyo ward, Kyoto city, Kyoto 606-8501, JAPAN

Key Words: mechanical parking facility, scheduling, shortest path problem, mathematical programming, dynamic programming.

入れを行う。ただし、2台の台車は同一レーン上を走行するので、両台車はある距離(=δ)までしか接近できない。また、レーンは駐車場の端までしかないので、端から距離δ以内の駐車室にはその端の側の台車しか到達できない。両台車が距離δ以内に接近あるいは相手台車の向こう側に移動しようとする状態を干渉と呼び、干渉が発生するときにはどちらかの台車が干渉が生じない地点まで退避し待機するという干渉回避処理が必要となる。

駐車場で作業には入庫処理と出庫処理がある。入庫処理の時には空いている駐車室が任意に選択できるので、出庫処理の時には各出庫車の格納されている駐車室は決まっている。したがって、出庫要求待ち行列に対応して定まる出庫車の駐車室の位置と出庫順序に応じて、2台の台車のどちらに処理をさせるか、また干渉が生じたときにはどのような干渉回避処理をさせるかを適切に決めないと、作業効率が大きく低下することがある。特に出庫では、駐車場利用者が出庫要求後自分の車が出庫口から出てくるまでずっと待っているため、その時間を短縮することは利用者に対するサービスの向上になる。

この機械式駐車場の出庫スケジューリング問題は、台車間の干渉による複雑さのため問題の分析が容易でないこと、加えてスケジューリングに対する研究開発投資の大きさの割には効果が小さいと考えられていたなどの理由から、これまで理論的解析はなされていなかった。

しかしながら、これまで考えられていた効果は厳密な解析に基づいたものではないこと、駐車場の能力は駐車台数と全車の総出庫時間等の関係に基づいて評価されるので可能ならばその厳密な関係を把握すべきであること、また実際の運用に適用されるのは計算時間の理由から近似解を求めるアルゴリズムになるにしても、最適解を求めることができればその性能をきちんと評価できることから、筆者らは理論的解析を試みた。

本論文では、この機械式駐車場の出庫スケジューリング問題が、動的計画法の考え方を適用することによってある構造的特徴をもったグラフ上の最短経路問題としてモデル化できることを示し、最適解を求めるアルゴリズムを与えるとともに、その計算量が多項式オーダーであることを示す。入庫および入出庫スケジューリング問題については、別の論文で扱う予定である。

2. 出庫スケジューリング問題

2.1 出庫処理

出庫要求に対し、その出庫車が階 j の駐車室に格納されているとすると、出庫エレベータと階 j の台車は Fig. 2 のダイアグラムの上図と下図に示す一連の動作を行う。上図は階 j に格納されている車の出庫のため、出庫エレベータが $E_1 \sim E_4$ の四つの部分作業からなる処理

Fig. 2 Diagram of removing a car

を行う様子を示している。ただし、縦軸はエレベータの位置(階0と階 j)、横軸は時間である。

- 1) E_1 は出庫口(階0)から階 j への空移動、
- 2) その後台車の到着を待ち、 E_2 は台車から車の移載、
- 3) E_3 は車を積載して階 j から出庫口への移動、
- 4) E_4 は扉を開き車を出し扉を閉める、

という部分作業である。これに対し、下図は出庫処理をする階 j の台車が $C_1 \sim C_4$ の四つの部分作業からなる処理を行う様子を示している。

- 1) C_1 は出庫車の格納されている駐車室への空移動、
- 2) C_2 は駐車室から車の移載、
- 3) C_3 は車を積載して出庫エレベータ室への移動、
- 4) C_4 は出庫エレベータに車の移載、

という部分作業である。ただし、縦軸は階 j におけるこの台車の水平位置を表している。Fig. 2 のダイアグラムにおいて、実線は車を積載している状態、破線は車を積載していない状態を表す。部分作業 E_2 と C_4 は同期して行われなければならないから、 E_1 と E_2 の間、あるいは C_3 と C_4 の間に通常は待ち時間が発生する。

複数台の出庫要求に対しては Fig. 2 の処理を繰り返すことになるが、台車は2台で処理するので、台車のダイアグラムは一般に干渉回避動作を含む複雑なものとなる。

2.2 出庫スケジューリング問題

Fig. 1 の駐車場では出庫エレベータは1台だけであるので、出庫作業時間を最短にするという観点からは、出庫エレベータの動かし方に対するスケジューリングの余地はない。一方、台車は2台あるので、各出庫をどちらの台車が処理するか、干渉が発生した場合にどちらの作業を優先するかを選択することができる。すなわち、出

庫スケジューリング問題における決定変数は台車割当てと干渉回避方法であって、これらは台車の作業のダイアグラムを決めることにほかならない。本論文で扱う出庫スケジューリング問題はつぎのように記述できる。

【定義 1】 出庫スケジューリング問題とは、時刻 $t=0$ において、出庫要求待ち行列 $1, 2, \dots, n$ と出庫エレベータ、各台車の初期位置が与えられたとき、出庫完了時間を最短とするような 2 台の台車の作業のダイアグラムを決定することである。

特に出庫要求数 n が総駐車台数の場合の出庫を全出庫といい、そのスケジューリング問題を全出庫スケジューリング問題という。全出庫とは、駐車場が満室状態から空の状態にする出庫のことであり、全出庫に要する時間は駐車場の能力評価を行ううえで重要な因子とされている。

問題の定式化のため以下の記号を導入する。

記号

- M : 各階のレーンの一番端の位置
- M_E : 出庫エレベータ室の位置
- T_E : 出庫エレベータの出口における作業時間
- T_S : 台車からエレベータへの車の移載時間
- T_R : 駐車室から台車への車の移載時間
- $x(k)$: 車 k が格納されている駐車室の位置
- $f(y, z)$: 台車が位置 y から位置 z まで移動するのに要する時間 ($f(y, z) = f(z, y)$)
- $g(a, b)$: エレベータが階 a から階 b まで移動するのに要する時間 ($g(a, b) = g(b, a)$)

3. 最短経路問題としてのモデル化

出庫 k が終了後、出庫 $k+1$ が終了するまでの作業時間を $U(k, k+1)$ とすると、

$$\text{出庫作業時間} = \sum_{k=0}^{n-1} U(k, k+1)$$

である。出庫 k の車の駐車室 $x(k)$ は階 $h(k)$ にあるとする。出庫 k の処理を終了後、出庫 $k+1$ の処理のため出庫エレベータが階 $h(k+1)$ に到着するまでの時間を $S_E(k+1)$ 、出庫 $k+1$ の車を積載した台車が出庫エレベータ室に到着するまでの時間を $S_C(k+1)$ とすると、

$$U(k, k+1) = \max\{S_E(k+1), S_C(k+1)\} + T_S + g(h(k+1), 0) + T_E$$

と書ける。さらに、

$$S_E(k+1) = g(0, h(k+1)) = g(h(k+1), 0)$$

であるから、

$$U(k, k+1) = \max\{0, S_C(k+1) - S_E(k+1)\}$$

$$+ 2g(h(k+1), 0) + T_S + T_E \quad (1)$$

したがって、出庫作業時間に対する次式を得る。

$$\begin{aligned} \text{出庫作業時間} &= \sum_{k=0}^{n-1} \max\{0, S_C(k+1) - S_E(k+1)\} \\ &\quad + \sum_{k=0}^{n-1} (2g(h(k+1), 0) + T_S + T_E) \\ &\geq \sum_{k=0}^{n-1} (2g(h(k+1), 0) + T_S + T_E) \quad (2) \end{aligned}$$

これより、最短出庫時間は (2) 式の右辺を下限として持つ。この右辺は、出庫エレベータが台車を待つことなくフル稼働する場合の所要時間を表している。(1) 式の右辺第一項は出庫エレベータが台車の到着を待つ待ち時間であるから、出庫作業時間最短問題は、出庫エレベータが台車の到着を待つ待ち時間の和を最小にする問題であるということができる。

一方、台車が出庫車 k を出庫エレベータに移載してから、次に出庫車 $k+1$ を出庫エレベータに移載するまでの時間を $W(k, k+1)$ とおくと、

$$\text{出庫作業時間} = \sum_{k=0}^{n-1} W(k, k+1) + g(h(n), 0) + T_E \quad (3)$$

とも書けるので、出庫作業時間最短問題は、 $W(k, k+1)$ の和を最小にする問題と見ることもできる。 $U(k, k+1)$ でなく $W(k, k+1)$ を用いると、 $U(k, k+1)$ の作業開始の時点では台車はすべて移動状態であるかもしれないのに対して、 $W(k, k+1)$ の作業開始の時点では、出庫 k の処理を済ませた台車は停止状態であるので、次の状態を考えやすいという利点がある。したがって、以下、 $W(k, k+1)$ の内容を詳細に検討する。

3.1 1 階だけの場合

$W(k, k+1)$ の計算にあたり、台車による出庫車 k の出庫エレベータへの移載が終了した時点時刻 0 とする。出庫車 $k+1$ を処理する台車を決め、相手台車にはそのとき処理する出庫車の番号を与えると、両台車がそれぞれの作業を開始する時刻と位置が既知であれば、干渉を無視した場合のダイアグラムは確定する。また、干渉が発生するかどうかはこれらの情報からわかり、干渉が発生する場合にはどちらの台車を優先するかを決めれば両台車のダイアグラムは確定する。それゆえ、両台車が作業を開始する時刻と位置の与え方が重要である。出庫車 k の処理を担当した台車は、時刻 0 で位置 M_E にいて、部分作業 C_1 を開始しようとしている。もう一方の台車については以下のことがわかる。

(1) 出庫 k の期間において干渉があった場合

- (a) 時刻0で車を積載していなければ, M_E から δ 離れた位置で待機しており, 次の部分作業 C_1 を開始しようとしている.
- (b) 時刻0で車を積載している場合はつぎのどちらかである.
 - (イ) 時刻0で M_E から δ 離れた位置で待機しており, 次の部分作業 C_3 を開始しようとしている.
 - (ロ) 積載した車の駐車室の位置に時刻0より前に停止しており, 次に部分作業 C_3 を開始しようとしている. この開始時刻を時刻 $(-t)$ とする.

(2) 出庫 k の期間において干渉がなかった場合

この台車が現在の処理 (の部分作業 C_1) を開始した時点時刻 $(-t)$ とする. このときの位置は, この出庫処理が最初の出庫処理であればスケジューリングの初期位置であり, 2回目以降の出庫処理であれば M_E である.

以上の考察から, 出庫 k の処理が終わった時点に対応して状態 $k(a, b, c, d)$ を導入する.

【定義2】 状態 $k(a, b, c, d)$

出庫番号 k に対する状態 $k(a, b, c, d)$ は, 以下で与える a, b, c, d の四つ組で定義される.

- a : 車 k を出庫した台車番号, $a = 1, 2$
- b : 相手台車が処理中の出庫車の番号
 $k+1 \leq b \leq n$ または $b = \phi$ (作業しない)
- c : 干渉の有無と相手台車の積載状態
 $= 0$: 干渉なし
 $= 1$: 干渉が有り相手台車は車 b を積載している
 $= 2$: 干渉が有り相手台車は車を積載していない
- d : 相手台車に対する上述の時刻 $(-t)$ の t を時間刻み Δt で離散化した値

状態の中に時刻 $(-t)$ での相手台車の位置や次の部分作業が陽には入っていないが, 上で考察したようにそれらは b, c, d の値から決定できるのである.

状態 $k(a, b, c, d)$ を節点 $k(a, b, c, d)$ に対応させ, また, スケジューリング開始時刻における台車1と2の初期位置を表す特別な節点を導入し, これを始点とする. 各節点から移行可能な節点を枝で結び, これらの節点と枝からなるグラフ G を作る. ある状態Aからある状態Bへ移行可能かどうかは, 両台車の初期条件が出発節点Aの情報から与えられ, 両台車の作業に対する情報が到着節点Bから与えられるので, この状態の移行を実現する台車のダイヤグラムが存在するかどうかでわかる. 終点は最終の出庫 n の処理終了に対応する節点である. 最後は

どちらかの台車が出庫 n の処理を行い, 相手台車は処理をしないので, $n(1, \phi, 0, 0)$ と $n(2, \phi, 0, 0)$ が終点である.

Fig. 3 にグラフ G を図示する. Fig. 3 において, 各節点から出る枝には3種類あり, 破線はある節点から出る複数の枝が時刻, 位置, 次の部分作業の情報から1本だけに決まる枝であることを表し, 波線は干渉時の優先台車の選択に対応する枝であることを表し, それらはすべて存在可能な枝である. さらに, 太実線は相手台車が処理する車の選択に対応する枝であることを表し, それらもすべて存在可能な枝である. このグラフ G は以下に示すいくつかの顕著な性質をもつ.

- (1) 節点 $k(a, b, c, d)$ から移行可能な節点を節点 $k'(a', b', c', d')$ とするとき, $k' = k+1$ である.
- (2) 節点 $k(a, b, c, d)$ と節点 $k+1(a', b', c', d')$ が枝で結ばれているとき, $a' = a$ ならば $b' = b$ である.
- (3) 節点 $k(a, b, c, d)$ と節点 $k+1(a', b', c', d')$ が枝で結ばれているとき, d' は一意的に決まる.
- (4) 節点 $k(a, b, c, d)$ から出る枝は, $b' = \phi$ のときは1本, $b' \neq \phi$ のときは各 b' に対し, 干渉がないか干渉があっても作業時間最短の観点からダイヤグラムが確定する場合は1本, 干渉があつてそのときの優先台車の選択に二通りある場合は2本ある.

出庫 k を終了後, 節点 $k(a, b, c, d)$ の状態から節点 $k+1(a', b', c', d')$ の状態に移るまでの作業時間 (以下に述べるように, 台車のダイヤグラムから求める) をその2節点間の枝の重み $W(k(a, b, c, d), k+1(a', b', c', d'))$ とする. そうすると, 最短時間スケジュールは, この重みつき有向グラフ G の始点から終点に至る最短経路を求めることによって決定できる.

つぎに, 枝の重み W の計算法を説明する. 出庫エレベータが再び階1に戻ってくる時刻は $2g(1, 0) + T_E$ であるから, ψ を出庫 $k+1$ の処理をする台車が出庫車を積載して出庫エレベータ室に到着する時刻を与える関数とすると, 一般に W は次式の表現で与えられる.

$$W(k, k+1) = \max\{2g(1, 0) + T_E, \psi\} + T_S \quad (4)$$

ψ は干渉の有無, 干渉回避の方法等に依存する. これらをすべて調べ上げ整理すると, ψ はつぎの $\psi_1, \dots, \psi_9$ のどれかになる.

$$\begin{aligned} \psi_1 &= f(M_E + \sigma(a)\delta, M_E) \\ \psi_2 &= 2f(M_E, x(k+1)) + T_R \\ \psi_3 &= f(M_E + \sigma(a)\delta, x(k+1)) + T_R + f(x(k+1), M_E) \\ \psi_4 &= -d\Delta t + 2f(M_E, x(k+1)) + T_R \\ \psi_5 &= -d\Delta t + f(M_E, x(k+1)) \\ \psi_6 &= f(M_E, x(b) - \sigma(a)\delta) + f(x(b) - \sigma(a)\delta, \end{aligned}$$

Fig. 3 Graph G

Fig. 4 Examples of diagrams of two carriers

$$\begin{aligned}
 & x(k+1) + f(x(k+1), M_E) + 2T_R \\
 \psi_7 &= f(M_E, x(b')) + T_R + f(x(b') + \sigma(a)\delta, M_E) \\
 \psi_8 &= f(M_E, x(b')) + T_R + f(x(b') + \sigma(a)\delta, \\
 & x(k+1) + T_R + f(x(k+1), M_E) \\
 \psi_9 &= f(M_E, x(k+1) - \sigma(a)\delta) + T_R + f(x(k+1), \\
 & x(b') + \sigma(a)\delta) + T_R + f(x(b') + \sigma(a)\delta, M_E)
 \end{aligned}$$

Fig. 3のグラフ G の枝に対応する ψ を示した。ただし、台車1と台車2の初期位置は M_E あるいは $M_E \pm \delta$ とは限らないので、始点から出る枝に対応する ψ は $\psi_1, \dots, \psi_9$

にはならない。干渉条件や ψ の導出を含むより詳細な議論については参考文献2)を参照されたい。

台車のダイヤグラムの例として、 $k(1, k+1, 0, d)$ から $k+1(2, b', c', d')$ に移行するときの3ケース ($c'=0, 1, 2$) を Fig. 4に示す。Fig. 4では、出庫 k は台車1が処理し、出庫 $k+1$ は台車2が処理している。Fig. 4の中央の図は $c'=1$ に対応し、出庫 $k+1$ において干渉があって出庫 $k+1$ の終了時に台車1が車を積載している場合を示している。この例では、出庫 $k+1$ のとき、台車1の出庫車も台車2の出庫車も駐車室の位置は M_E より

も台車2の側にあり、 $d'=0$ であるから台車2が干渉時に退避処理をするダイアグラムとなる。ただし、台車2が車を積載してから退避する場合と、退避してから車を積載する場合の2通りの可能性がある。このような場合には、 $W(k, k+1)$ を最小にするダイアグラムが採用され、Fig. 4の中央の図は前者の場合であったことを示している。

3.2 多階の場合

出庫スケジューリングにおいて、駐車場が多階の場合には1階の場合と異なるつぎの特徴がある。

- (1) 各階での台車の作業が並行してできる。
- (2) 各階間の相互作用は出庫エレベータの作業を通してのみ発生する。

1階の場合に対する最短経路問題としてのモデル化を多階の場合に拡張しようとするとき、上記の特徴(1)より各階ごとの独立したグラフを作成し、それらに特徴(2)を考慮した何かを付加することによってできそうであるが、実はそれほど単純ではない。ある階である出庫 k_1 が終了した状態から、同じ階の次の出庫 k_2 が終了するまでの作業時間を計算しようとするとき、 $k_2 = k_1 + 1$ ならば、 k_2 の出庫エレベータが到着する時刻は出庫 k_1 の状態に関する情報から計算できるが、そうでない場合は他階での出庫作業(つまり他階におけるグラフのパス)に依存する。この依存性を処理するために、他階の状態もすべて同時に考慮することができるよう、出庫 k の処理が完了した時点の状態をつぎのように定義した。

【定義3】 状態 $k((a_1, b_1, c_1, d_1, e_1), \dots, (a_N, b_N, c_N, d_N, e_N))$

a_j : 出庫車 k が階 j の場合は車 k を出庫した台車番号。出庫階以外の階 j においては最近の出庫処理を行った台車番号。 $a_j = 1, 2$

b_j : 階 j における相手台車が処理中の出庫車の番号。
 $k+1 \leq b_j \leq n$ または $b_j = \phi$ (作業しない)

c_j : 階 j における干渉の有無と相手台車の積載状態。
 $= 0$: 干渉なし

$= 1$: 干渉が有り相手台車は車 b_j を積載している

$= 2$: 干渉が有り相手台車は車を積載していない

d_j : 時刻 $(-t)$ の t を時間刻み Δt で離散化した値

e_j : 出庫車 k を処理する階 j においては0。そうでない階においては、出庫 k の作業終了時刻を時刻0としたとき、階 j の最近の出庫の作業終了時刻 $(-t')$ の t' を時間刻み Δt で離散化した値。ただし、階 j の出庫がすべて終了なら0とする。

この状態定義は、基本的に1階の場合の状態の N 階分の直積であり、さらに各階における基準時刻0がずれているため、時刻を調整するための新たな変数 $e_1, \dots, e_N$

を導入したものである。この状態によって、多階の場合も1階の場合と同様に、全階の台車の作業ダイアグラムが節点 $k((a_1, b_1, c_1, d_1, e_1), \dots, (a_N, b_N, c_N, d_N, e_N))$ から節点 $k+1((a'_1, b'_1, c'_1, d'_1, e'_1), \dots, (a'_N, b'_N, c'_N, d'_N, e'_N))$ への移行にともなうて決定されるので、移行可能な2節点間を枝で結び、この間の処理時間を枝の重みとする重みつき有向グラフ G を作る。多階の場合の最短時間スケジューリングは、この重みつき有向グラフ G の始点から終点に至る最短経路を求めることによって決定できる。

4. DPによる解法と計算量

4.1 1階だけの場合

3.1でモデル化した最短経路問題は、枝の重みがすべて枝の両端節点の情報だけから計算でき、過去に通過したパスの情報には依存しないので、動的計画法(DP)³⁾に基づくアルゴリズムで解くことができる。すなわち、

$$\Phi(k(a, b, c, d)) = \text{節点 } k(a, b, c, d) \text{ から終点に至る最短パス長}$$

とおき、節点 $k(a, b, c, d)$ と枝で結ばれている節点の集合を $V = \{k+1(a', b', c', d')\}$ とすると、

$$\Phi(k(a, b, c, d)) = \min_{v \in V} \{W(k(a, b, c, d), v) + \Phi(v)\} \quad (5)$$

であるから、これを k の小さいものから大きいものへ順番に求めてゆけばよい。

本アルゴリズムの計算量を評価するため、まず節点の数を求める。Fig. 3のグラフ G の構造を考慮すると節点 $k(a, b, c, d)$ の個数およびその節点から出る枝の本数は、Table 1のようにまとめることができる。節点 $k(a, b, c, d)$ において、 $c=0$ の場合の d の最大個数 D_1 と $c=1$ の場合の d の最大個数 D_2 はそれぞれ、

$$D_1 = [(2\max\{f(1, M_E), f(M, M_E)\}) + T_R] / \Delta t + 0.5 \quad (6)$$

$$D_2 = [(\max\{f(1, M_E), f(M, M_E)\})] / \Delta t + 0.5 \quad (7)$$

で与えられる。ここで、

$$D = \max\{D_1, D_2\} = D_1 \quad (8)$$

とおく。Table 1より、各 k に対する節点 $k(a, b, c, d)$ の最大個数を $Q(k)$ 、 $k=1, \dots, n$ とすると、

$$Q(k) = 2\{2(n-k)(D+1)+1\} \quad (9)$$

と書ける。始点は1個であるから、

Table 1 Number of nodes and edges (one floor type)

k	node				number of nodes	number of edges
	form	a	b	d		
n	$n(a, \phi, 0, 0)$	1, 2	ϕ	0	2	0
$n-1$	$n-1(a, \phi, 0, 0)$	1, 2	ϕ	0	2	1
$1 \leq k \leq n-2$	$n-1(a, n, 0, d)$	1, 2	n	$1, \dots, D_1$	$2D_1$	1
	$n-1(a, n, 1, 0)$	1, 2	n	0	2	1
	$n-1(a, n, 1, d)$	1, 2	n	$1, \dots, D_2$	$2D_2$	1
	$n-1(a, n, 2, 0)$	1, 2	n	0	2	1
	$k(a, \phi, 0, 0)$	1, 2	ϕ	0	2	1
	$k(a, b, 0, d)$	1, 2	$k+2, \dots, n$	$1, \dots, D_1$	$2(n-k-1)D_1$	1
	$k(a, b, 1, 0)$	1, 2	$k+2, \dots, n$	0	$2(n-k-1)$	1
	$k(a, b, 1, d)$	1, 2	$k+2, \dots, n$	$1, \dots, D_2$	$2(n-k-1)D_2$	1
	$k(a, b, 2, 0)$	1, 2	$k+2, \dots, n$	0	$2(n-k-1)$	max 2
	$k(a, k+1, 0, d)$	1, 2	$k+1$	$1, \dots, D_1$	$2D_1$	max $2(n-k-1)+1$
	$k(a, k+1, 1, 0)$	1, 2	$k+1$	0	2	max $2(n-k-1)+1$
	$k(a, k+1, 1, d)$	1, 2	$k+1$	$1, \dots, D_2$	$2D_2$	max $2(n-k-1)+1$
	$k(a, k+1, 2, 0)$	1, 2	$k+1$	0	2	max $2(n-k-1)+1$
0	start node				1	max $2(2(n-1)+1)$

$$\begin{aligned} \text{総節点数} &\leq 1 + \sum_{k=1}^n Q(k) = 1 + 2 \sum_{k=1}^n \{2(n-k)(D+1)+1\} \\ &= 1 + 2(D+1)n(n-1) + n = O(Dn^2) \quad (10) \end{aligned}$$

つぎに、枝の数を求める。Table 1 より、 $k(a, b, c, d)$, $k=1, \dots, n-1$ の形の節点から出る枝の最大本数を $R(k)$ とすると、

$$R(k) \leq \begin{cases} 4D+6 & ; k=n-1 \\ 2(n-k-1)(6D+7)+4D+6 & ; k=1, \dots, n-2 \\ 4n-2 & ; k=0 \end{cases} \quad (11)$$

となる。したがって、

$$\begin{aligned} \text{総枝数} &= \sum_{k=0}^{n-1} R(k) \\ &\leq 4n-2 + \sum_{k=1}^{n-2} \{2(n-k-1)(6D+7)+4D+6\} \\ &\quad + 4D+6 \\ &= O(Dn^2) \quad (12) \end{aligned}$$

ここで、計算に必要なメモリの大きさを考察する。DP を適用して最短経路を求めるにあたっては、始点から最短経路を求めていく前向き解法と、終点に至る最短経路を逆向きに求めていく後向き解法のどちらも可能である。計算にあたって、すべての節点の情報をあらかじめ生成して

おく必要はないが、たとえば後向き解法で解くとすると、節点 $k(a, b, c, d)$ から終点に至る最短経路を求めるときに、節点 $k+1(a', b', c', d')$ から終点に至る最短経路と最適値の情報が必要なので、これらがメモリに保存されていなければならない。そして、節点 $k(a, b, c, d)$ の形のすべての節点の計算が終了するまでは、節点 $k+1(a', b', c', d')$ の情報を残しておかなければならない。ゆえに、これらの情報をたとえば配列構造でメモリに保存しようとする、最適値については、節点 $k(a, b, c, d)$ の形の節点の最大個数 $\times 2 \times 4$ バイト必要である。最短経路については、組 (c, d) を 1 変数で記憶することになると組 (a, b, c, d) の記憶に 3 変数が必要になるので、節点 $k(a, b, c, d)$ の形の節点の最大個数 $\times 2 \times 3 \times$ 出庫数 $\times 4$ バイト必要である。

節点 $k(a, b, c, d)$ の最大個数

$$= \max Q(k) = 2\{2(n-1)(D+1)+1\}$$

であるから、

$$\text{最適値保存用メモリ} = 16\{2(n-1)(D+1)+1\}$$

$$\text{最短経路保存用メモリ} = 48n\{2(n-1)(D+1)+1\}$$

バイトとなる。32 メガバイトの主記憶をもつ計算機では、これらの情報の記憶に約 20 メガバイト程度が消費できるとして、 $D=100$ のとき出庫数 $n=45$ 位まで、 $D=10$ のとき $n=138$ 位までの計算が可能である。

4.2 駐車場が多階の場合

$\xi_j(k)$ を出庫 k までに階 j から出庫した車の数を表すとし、時刻 $e_1, \dots, e_N$ の最大個数を E とする。出庫 k ($1 \leq k \leq n$) に対する階 j の状態 (a_j, b_j, c_j, d_j) の最大個数を $Q_j(k)$ とすると、(9) 式において k を $\xi_j(k)$ に n を n/N に置き換えることにより、

$$Q_j(k) = 2\{2(n/N - \xi_j(k))(D+1) + 1\} \quad (13)$$

が得られる。 $e_1, \dots, e_N$ のうち一つは必ず 0 であるから、多階の場合の各 k に対する節点 $k((a_1, b_1, c_1, d_1, e_1), \dots, (a_N, b_N, c_N, d_N, e_N))$ の最大個数を $Q(k)$ とすると、

$$Q(k) = E^{N-1} \prod_{j=1}^N Q_j(k) \quad (14)$$

である。 $\xi_j(k)$ にはつぎの関係

$$\sum_{j=1}^N \xi_j(k) = k \quad (15)$$

が成り立つから、

$$\begin{aligned} Q(k) &= 2^N E^{N-1} \prod_{j=1}^N \{2(n/N - \xi_j(k))(D+1) + 1\} \\ &\leq 2^N E^{N-1} \left(N^{-1} \sum_{j=1}^N \{2(n/N - \xi_j(k))(D+1) + 1\} \right)^N \\ &= 2^N E^{N-1} [N^{-1} \{2(n-k)(D+1) + N\}]^N \end{aligned} \quad (16)$$

と書ける。始点は 1 個であるから、

$$\begin{aligned} \text{総節点数} &\leq 1 + \sum_{k=1}^n Q(k) \\ &= 1 + 2^N E^{N-1} \sum_{k=1}^n \{[N^{-1} 2(n-k)(D+1) + N]\}^N \\ &= O(D^N E^{N-1} n^{N+1}) \end{aligned} \quad (17)$$

なお、 E は

$$\begin{aligned} E \Delta t &= 2f(1, M - \delta) + \max\{f(1 + \delta, M_E), f(M - \delta, M_E)\} \\ &\quad + 2T_R - \{g(0, 1) + g(0, m) + T_E\} \end{aligned} \quad (18)$$

で与えられる。この E より大きな e に対応する節点に対しては、その節点から出る枝の重みはすべて $e = E$ の場合と同じであり、かつ必ず干渉が発生して相手台車を優先しても出庫エレベータの到着までに出庫車を出庫エレベータ室に運ぶことができるから、その枝の行き先節点も $e = E$ の場合と同じである。したがって、 E より大きな e に対応する節点は不要である。

つぎに、枝の数を考える。出庫 k , $1 \leq k \leq n-1$, に対し、出庫車 $k+1$ が出庫される階を m とすると、節点 $k((a_1, b_1, c_1, d_1, e_1), \dots, (a_{m-1}, b_{m-1}, c_{m-1}, d_{m-1}, e_{m-1}), (a_m, b_m, c_m, d_m, e_m), (a_{m+1}, b_{m+1}, c_{m+1}, d_{m+1}, e_{m+1}), \dots, (a_N, b_N, c_N, d_N, e_N))$ から枝で接続される節点は、 $k+1((a_1, b_1, c_1, d_1, e'_1), \dots, (a_{m-1}, b_{m-1}, c_{m-1}, d_{m-1}, e'_{m-1}), (a'_m, b'_m, c'_m, d'_m, 0), (a_{m+1}, b_{m+1}, c_{m+1}, d_{m+1}, e'_{m+1}), \dots, (a_N, b_N, c_N, d_N, e'_N))$ という形の節点である。階 m における状態 (a_m, b_m, c_m, d_m) から状態 (a'_m, b'_m, c'_m, d'_m) への遷移の可能性は Table 1 に示した 1 階だけの場合と同じである。階 m におけるこの遷移が決まると $w(k, k+1)$ が確定するので、階 $j \neq m$ における e'_j が e_j から一意的に決まる。したがって、1 階だけの場合の $k(a, b, c, d)$ の形の節点から出る枝の最大本数をあらためて $\bar{R}(k)$ とし、 $k((a_1, b_1, c_1, d_1, e_1), \dots, (a_N, b_N, c_N, d_N, e_N))$ の形の節点から出る枝の最大本数を $R(k)$ とすると、

$$\begin{aligned} R(k) &\leq \bar{R}(k) E^{N-1} \prod_{j=1}^N Q_j(k) \\ &= \bar{R}(k) (Q_m(k))^{-1} E^{N-1} \prod_{j=1}^N Q_j(k) \end{aligned} \quad (19)$$

となる。 $\bar{R}(k) = O(Dn)$, $Q_m(k) = O(Dn)$ であるから、(16) 式の場合と同じ計算により、

$$\begin{aligned} R(k) &\leq \bar{R}(k) (Q_m(k))^{-1} 2^N E^{N-1} \\ &\quad [N^{-1} \{2(n-k)(D+1) + N\}]^N \\ &= O(D^N E^{N-1} n^N) \end{aligned} \quad (20)$$

となる。したがって、

Table 2 Parking room where the k -th removal car is stored (one floor type)

removal order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
room number	4	16	10	12	9	13	15	2	17	1	6	5	8	3	18	14

Fig. 5 Parking rooms of removal cars (one floor type)

Fig. 6 Diagram of optimal schedule (one floor type)

$$\text{総枝数} = \sum_{k=0}^{n-1} R(k) \leq O(D^N E^{N-1} n^{N+1}) \quad (21)$$

計算に必要なメモリの大きさは、たとえば2階の場合、1階の場合と同様の考察から、

$$\text{最適値保存用} = 8[2\{2(n-1)(D+1)+1\}]^2 E$$

$$\text{最短経路保存用} = 56n[2\{2(n-1)(D+1)+1\}]^2 E$$

バイト必要になる。32メガバイトの主記憶をもつ計算機では、 $D=100$ 、 $E=100$ のときにはメモリオーバーのため計算不能である。 $D=10$ 、 $E=10$ のときでも、 $n=2$ の場合しか計算できない。

5. 最適解の例と考察

5.1 1階の場合

実際の規模よりはかなり小さいが、得られた結果が考察しやすいことから、駐車室が16室（および出庫と入庫エレベータ室）の全出庫スケジューリング問題を例としてとり上げる。Table 2は出庫車が出庫順序ごとにどの室に格納されているかを示し、Fig. 5は各駐車室に何番の出庫車が格納されているかを示したものである。この駐車場の例では、 $T_E=11.0$ 、 $T_R=T_S=9.5$ 、 $g(1,0)=5.0$ 、および、 $f(a,b)=f_1(|a-b|)$ であり、 $f_1(1)=9.8$ 、 $f_1(2)=12.5$ 、 $f_1(3)=14.6$ 、 $f_1(4)=16.3$ 、 $f_1(5)=17.9$ 、...である。この例に対する最適解は、

$$\text{最短出庫作業時間} = 592.0 \text{ 秒 (下限値} = 488.0 \text{ 秒)}$$

であり、最適解に対応する台車の運行ダイアグラムはFig. 6に示すようになる。Fig. 6のダイアグラムから、たとえばつぎのようなことがわかる。

1) 台車1と2で処理する領域を分担するのではなく、必要に応じて相手台車の領域の処理も行っている（例：出庫3,6,11,13）。

2) 台車が交互に出庫処理をするのではなく、出庫車の駐車室の位置等により、さらに先の出庫処理に進む場合もある（例：台車1の出庫1から5、台車2の出庫4から7）。

3) 必ずしもその時点での出庫処理を優先させるのではなく、順序は先の出庫であってもその車をまず積載しておく場合が有利なこともある（例：台車1の出庫5は台車2の出庫2を待たせている）。

階が一つだけの場合の出庫スケジューリング問題は一見簡単に見えるが、Fig. 6の結果からもわかるように、最適解は決して自明なものではない。これらの解を単純なヒューリスティックで得るのは困難と考えられる。

なお、90MHzのPentium搭載パーソナルコンピュータで、 $D=50$ のとき、この例の計算に要した時間は約5秒であった。駐車室が60室の場合の計算には約5分要した。

5.2 2階の場合

4.2で述べたように、2階の場合の最適解の計算は現在の標準的能力の計算機では難しい。そこで、2階の場合にも何とか最適解を求めるため、以下の二つの工夫を行った。ただし、 $D=10$ 、 $E=10$ とした。第一の工夫は最適経路すべてを保存するのではなく、最適経路の一つ前の節点のみを保存することにしたことである。そのため、終点から後向きに解いていって始点に到達しても、最適値は求められているが最短経路が保存されていない。そこで今度は、最短経路上の一つ前の節点を始点として、再び計算をし直す。これを繰り返し、再設定される始点を終点に近づけて最適経路を求めるのである。すなわち、メモリの不足を計算時間の消費に転嫁させたのである。第

		M_E															
first floor		1	3	5	7	9	11	13	15	17	19						
		2	4	2	1	8	In	1	2	6	Out	13	2	9	3	6	
		No. 1															
		1	0	1	4	1	7	3	0	2	8	1	8	1	2	1	6
second floor		2	4	6	8	1	0	1	2	1	4	1	6	1	8	2	0
		1	3	5	7	9	1	1	1	3	1	5	1	7	1	9	
		3	4	6	In	2	2	9	Out	2	2	7	3	4			
		No. 2															
		5	3	5	3	2	3	3	1	7	1	5	2	5	2	0	2
		2	4	6	8	1	0	1	2	1	4	1	6	1	8	2	0
		No. 1															
		5	3	5	3	2	3	3	1	7	1	5	2	5	2	0	2

Fig. 7 Parking rooms of removal cars (two floor type)

Table 3 Parking floor and room where the k -th removal car is stored (two floor type)

removal order	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
floor number	1	2	2	2	2	2	2	1	2	1	1	1	1	1	2	1	2	1
room number	9	15	1	3	2	5	12	5	11	2	18	14	15	4	14	16	6	12
removal order	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
floor number	1	2	1	2	2	1	2	1	2	1	1	1	2	2	2	2	2	1
room number	20	18	3	9	20	1	16	11	17	10	17	8	10	6	8	19	4	19

Fig. 8 Diagram of optimal schedule (two floor type)

二の工夫は、節点 $k((a_1, b_1, c_1, d_1, e_1), (a_2, b_2, c_2, d_2, e_2))$ において、相手台車が処理する出庫 b_1, b_2 は各階における現在の出庫からたかだか5出庫先までに限定したことである。厳密に言えば、これは最適解の探索範囲を絞ったことになるため、得られた解の最適性は保証できなくなる。しかしながら、相手台車が随分先の出庫を早目に準備することは、1階の最適解の計算結果からみても、ほとんど生じないと考えられる。これら二つの工夫によりつぎの計算結果を得た。

1階あたり18室で2階の駐車場の出庫スケジューリングの例をとりあげる。5.1で与えたデータに加え、 $g(2,0)=6.5$ である。Table 3は出庫順序ごとにどの室に格納されているかを示し、Fig. 7は各駐車室に何番の出庫車が格納されているかを示したものである。この例に対する最適解は、

最短出庫作業時間 = 1235.7秒 (下限値 = 1152.0秒)

であり、最適解に対応する台車の運行ダイアグラムはFig. 8に示すようになる。

なお、90MHzのPentium搭載パーソナルコンピュータで、この例の計算に要した時間は約1時間15分であった。

6. おわりに

本研究は、機械式立体駐車場の出庫スケジューリング問題がグラフの最短経路問題としてモデル化できること、およびそのアルゴリズムの計算量が出庫数 n の多項式オーダーであることを明らかにしたものである。また、実際にそのアルゴリズムを実装し、最適解の例を与えた。ただし、多階の場合には、出庫エレベータの階間の相互作用を考慮しなければならないため、状態として各階の状態の直積を用いており、それゆえ状態数は階数 N に対しては指数関数となっている。各階の状態の直積を用いずに、出庫エレベータの階間の相互作用を考慮することができ、状態数が階数 N の多項式であるようなモデル化が期待されるが、これは今後の研究課題である。

応用面から見ると、機械式駐車場といっても構造的により複雑なタイプのものもあり、またスケジューリングには出庫の場合だけでなく入庫も絡んだより複雑なスケジュー

リングが存在する。これらも今後の研究課題である。

計算面から見ると、本研究で得られたアルゴリズムは多項式オーダーであるといっても、実際には多大なメモリを必要とし、かつ非常な計算時間を要する。そのため、現在のハードウェア環境のもとでは、最適解を求めることのできる駐車場の規模に制限がある。そこで、問題規模が大きくなっても、実用に耐え得るような計算効率の良い近似解法が求められる。すでにこの近似解法の開発も行っており、各種の近似解法の性能を比較評価するうえで本論文で考察したDPによる最適解が大いに役立った。

最後に、本研究で行ったモデル化は機械式立体駐車場だけでなく、同一レーン上を走行するマシン間に干渉のある他の設備やシステムにも応用可能であると考えられる。これも今後の研究課題である。

参 考 文 献

- 1) 三星ほか：立体駐車場設備；建築と社会，Vol. 77, No. 4, pp. 27-65 (1996)
- 2) 阿瀬，茨木，柳浦：機械式立体駐車場出庫スケジューリング；京都大学大学院情報学研究科数理工学専攻テクニカルレポート
- 3) ベルマン：ダイナミック・プログラミング，東京図書 (1973)