



無人搬送車の運行管理法に関する研究

宝崎, 隆祐

(Degree)

博士 (学術)

(Date of Degree)

1991-03-31

(Date of Publication)

2014-01-28

(Resource Type)

doctoral thesis

(Report Number)

甲0986

(JaLCD0I)

<https://doi.org/10.11501/3057170>

(URL)

<https://hdl.handle.net/20.500.14094/D1000986>

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。



博士論文

無人搬送車の運行管理法に関する研究

平成 3 年 1 月

神戸大学大学院自然科学研究科

宝 崎 隆 祐

目次

第1章 緒論	1
第2章 無人搬送車の運行問題	8
2.1 緒言	8
2.2 経路・スケジュール決定問題の構成要素	8
2.3 経路決定問題	10
2.4 スケジュール決定問題	11
2.5 混合問題	12
2.6 無人搬送車の運行問題	13
2.6.1 干渉回避問題	14
2.6.2 搬送システムのモデル	15
2.6.3 評価尺度	16
2.6.4 割当て問題と運行経路決定問題	17
2.7 結言	17
第3章 一台車に対する運行経路決定問題	19
3.1 緒言	19
3.2 最短時間経路決定問題とタイムウィンドウ	19
3.2.1 台車の最短時間経路決定問題	19
3.2.2 通過可能時間帯	20
3.2.3 タイムウィンドウ	21
3.3 台車運行型SPPTW (Shortest Path Problem with Time Window)	23
3.3.1 モデルの設定	23
3.3.2 演算の定義	24

3.4	台車運行型SPPTWの解法アルゴリズム	25
3.4.1	修正進入可能時間帯と存在可能時間帯	25
3.4.2	解法アルゴリズム	26
3.4.3	実行可能解の性質	28
3.4.4	アルゴリズムの正当性に関する考察	29
3.5	下界評価法と解法の効率化	32
3.5.1	下界評価第1法, 第2法	33
3.5.2	緩和問題と下界評価第3法	34
3.6	数値例	36
3.6.1	搬送系モデル	37
3.6.2	解法アルゴリズムによる解法例	37
3.6.3	下界評価法の比較例	39
3.7	結言	43
第4章 台車運行型SPPTWの拡張		44
4.1	緒言	44
4.2	走行時間の非負性を仮定しない場合の一般解法	44
4.2.1	ネットワークの再構成	44
4.2.2	問題の変換と部分問題の定式化	46
4.2.3	最適経路の性質	48
4.2.4	部分問題の解法アルゴリズム	50
4.2.4.1	Bellman-Ford型の解法アルゴリズム	50
4.2.4.2	Yen型の解法アルゴリズム	53
4.2.4.3	分枝限定型の解法アルゴリズム	53
4.2.5	台車運行型SPPTWの一般解法	56
4.2.6	数値例	57
4.3	走行リスケジューリング問題	60
4.3.1	実行可能な走行スケジュールの表現法	61

4. 3. 2	最早到着時刻と最早出発時刻の導出	62
4. 3. 3	最遅出発時刻と最遅到着時刻の導出	64
4. 3. 4	四つの時刻の関係	67
4. 3. 5	走行スケジュールのずれとリスケジュールリング手順	68
4. 3. 6	数値例	68
4. 4	その他の現実的な運行経路決定問題への拡張	72
4. 4. 1	複数目的地から成る搬送要求に対する経路決定法	72
4. 4. 2	目的地において作業時間を有する場合の経路決定法	73
4. 5	結言	74
第5章 複数台車に対する運行経路決定問題		75
5. 1	緒言	75
5. 2	非同時走行問題	76
5. 3	非同時走行問題に対する運行経路決定法	77
5. 3. 1	経路決定アルゴリズム	77
5. 3. 2	待ち停止時間の決定	79
5. 3. 3	線形計画問題数の削減	82
5. 4	三つの経路決定法の特徴と比較	84
5. 4. 1	経路決定法の定性的特徴	84
5. 4. 2	シミュレーションの条件と検討項目	86
5. 4. 2. 1	条件設定	86
5. 4. 2. 2	検討項目	88
5. 4. 3	シミュレーションの結果とその分析	89
5. 4. 3. 1	解答不能率	89
5. 4. 3. 2	解の改善度	90
5. 4. 3. 3	計算所要時間	93
5. 4. 4	更新法における計算所要時間の短縮	99
5. 4. 5	現実的な経路決定手順に関する考察	100

5.5	最大走行完了時刻最小化問題に関する考察	101
5.6	ヒューリスティックな干渉回避法	102
5.6.1	非同時走行問題の解答不能性の原因と対策	102
5.6.2	ヒューリスティックな干渉回避法	105
5.6.3	三つの干渉回避法の特徴	107
5.6.4	数値例	110
5.7	結言	113
第6章 搬送要求の割当て問題		115
6.1	緒言	115
6.2	重複を許さない割当て問題	116
6.3	重複を許す割当て問題	120
6.4	動的割当て法	122
6.5	数値例	124
6.5.1	シミュレーションの条件と検討項目	124
6.5.2	シミュレーションの結果とその分析	126
6.6	結言	131
第7章 結論		133
謝辞		136
参考文献		137

第1章 緒 論

生産システムの構成サブシステムのなかで、搬送系システムは最も遅れて自動化、自律化に手がつけられた分野の一つである¹⁾²⁾。搬送系の自動化は、NC機械やロボットを用いた加工、組み立てシステム³⁾⁴⁾、CAD/CAMを用いた設計システム⁵⁾⁻⁷⁾、各種手法による生産計画・生産管理システム⁸⁾⁻¹¹⁾、自動倉庫等の物流在庫システム⁴⁾等、比較的自動化の容易なサブシステムの島¹²⁾¹³⁾を経て、ようやくそれらを結ぶ搬送系の自動化に到達したかのような感がある。1985年の市場規模調査の統計によっても、NC機械市場が約5,400百万ドル、ロボット市場が約2,300百万ドル、CAD/CAMが約1,000百万ドルであるのに対し、AGV (Automated Guided Vehicle) のそれは約20百万ドル程度である¹⁾。しかし、AGVシステムの導入数は1980年以降急増しており、その成長にはめざましいものがある²⁾。

1910年代に始まったフォード社の大量生産形態では、生産工程間の物流は工作、組み立て機械と一体となった柔軟性の希薄なコンベアが担っており、独立した搬送系と呼ぶことのできないものであった。生産工程に対し搬送系がその独立した機能を持つようになるには、近年の多品種少中量生産に対する自動化の高まりが必要であった。この生産形態に適合する生産システムとして登場してきたのが、FMS (Flexible Manufacturing System) である。この自動化生産システムを最初に登場させた米国のKearney & Trecher社は、次のようにFMSを定義した¹³⁾。「(1)数台の工作機械で構成されたいくつかの加工ステーションと、(2)部品搬送用のマテリアルハンドリングシステムを持ち、(3)特有のアルゴリズムを用いたコンピュータ制御のシステム」。この表現に現われているように、FMSでは独自の機能を有する搬送系システムが不可欠であることがうかがえる。このように、生産工程を柔軟にかつ効率よく結ぶ自動搬送システムは、1960年代後半のFMSの誕生とともに注目され始めてきた。以後、自動搬送システムの発展は、機能上の柔軟性を追求しながら現在に至っている。

搬送方向の自由度を規定する搬送パターンの形態については、直線形からループ形、ネット形、ツリー形、更にはランダムアクセス形へと柔軟性の高い形態への移行がみられる¹³⁾¹⁴⁾。もちろん産業分野により必要とする自動搬送システムの主体となるハードウェアは異なり、コンベアから、ロボット、軌条台車、クレーン、モノレール、無人搬送車といった多種多様なシステム

が共存している⁴⁾¹³⁾。また、搬送容量としては、自動車産業界において自動車のフレームを搬送する数トン容量のものから、近年の電子部品業界における数キログラム容量のもの¹⁾⁴⁾¹³⁾、ハンドリングできるワークの形状としては、角物形状、回転形状、平板形状から、さらにはサービス業界における規格化されていないサービス品の運搬等¹³⁾、それぞれの時代の産業界からの要請に応える様々な搬送システムが開発されてきている¹⁵⁾¹⁶⁾。

これら自動搬送システムをハード面から支える要素技術は早くからあったが、現在では、それらに先端技術と呼ばれるものの応用も研究されつつある。センサー技術におけるレーザー、レーダー、音響光学機器、磁気、画像処理技術、超音波、CCDカメラ等、誘導技術における慣性航法、衛星利用、マップマッチング技術等、通信技術におけるデータキャリアや移動体通信等の面で数多くの試みがなされている¹⁷⁾¹⁸⁾。このように極めて多種多様化する自動搬送システムの中で、現在注目されているのが無人搬送車システムである。搬送手段としての無人搬送車（AGV：Automated Guided Vehicle）は、他の手段と比較して、次のような利点を有する²⁾¹⁷⁾。

- 1) コースの設定、変更に対する柔軟性がある。
- 2) 走行路面や環境に対する変化に対し対応しやすい。
- 3) 障害物検出と回避機能をもたせることができる。
- 4) 高度な情報通信機能を保有できる。
- 5) 分散処理化された搬送システムへの変更も比較的容易である。
- 6) 複数搬送車の相互協調が可能である。
- 7) 搬送経路の選択に柔軟性があり、最適コースの自動選択も可能である。
- 8) 自律的、知能的な誘導技術の応用が可能である。

これらのことからわかるように、無人搬送車の特徴は、他の搬送手段にはない高い柔軟性にある。以下では、典型的な無人搬送車のハードウェアについて概説する¹⁷⁾¹⁹⁾。

誘導方式としては、電磁誘導方式が最もコマーシャルベースにのった方式である。工場床面の深さ数cmの溝にケーブルを埋設し、3～10kHzの電波を流す。この信号により生じる誘導電流をAGV本体の左右のピックアップコイルで感知することにより、ケーブルからの左右のずれを検知する。従って、通常は、埋設したケーブルに沿った経路のみを搬送路として使用できる。搬送路上には、AGVと加工機械や倉庫、各種バッファとの間で工作物、治工具等の交換を行うためのロード／アンロードステーション設備があ

る。また、電磁誘導方式のAGVは通常バッテリーからの電力を動力源としており、搬送路上にはバッテリー充電用の設備も設けられている。AGVは、これらの設備と物流システムとの接合点において停止できる他、稼働を終えたAGVの待機場所としてのホームポジションや搬送路の分岐点、制御用に予め決められた地点において停止可能である。

搬送システムを含めた生産システム全体の柔軟性を支える最も重要な技術は、コンピュータを用いた情報処理技術である。生産システムのソフトウェア体系を図示したのが、図1. 1¹³⁾である。多品種製品に対応する生産方式を変更する場合、ハードウェアを変更する方法には種々の制約が大きく、ソフトウェアの変更による簡便で効率的な方法が主流となった。搬送システムの制御・運行における柔軟性の実現にも、同様の理由により、そこで用いられるソフトウェアが重要である。しかし、柔軟性があるといわれるAGVシステムにおいても、種々のハードウェア的な制約があり、また、ソフトウェア内部で用いられているアルゴリズムそのものにも柔軟性が十分実現されているとは言えない現状も見逃せない。

ここでは、そのようなソフトウェアとして、AGVの運行管理法について考える。複数AGVが運行されている搬送系では、AGV間の干渉回避は最も重要な問題の一つである。AGV同士の干渉形態としては、一方が他方の経路を一時的に塞ぐブロッキングや、それが解消不能となるデッドロックがある。現状では、干渉回避操作を容易に行うために、制御は容易であるが選択しうる経路に柔軟性のないループ形の搬送パターンに一方通行の走行制御を用いているケースが多い^{21) 22)}。また、搬送系内をいくつかのブロックに分割し、一つのブロック内では一台のAGVのみしか走行させないで干渉回避を行うブロッキングゾーン制御や、搬送路の分岐点進入前の予め決められた地点で分岐点での干渉可能性を判定し、必要に応じて局所的に干渉を回避する方法等が多く採用されている^{13) 19) 20)}。また、AGVの運行経路決定法としては、干渉の発生については一切考慮せず、搬送ネットワーク上の最短経路をとりあえず採用し、途中の干渉回避は上述した局所的な走行制御に委ねるような方法が一般的に行われている。

このような運行制御は比較的規模の小さな搬送システムにおいては簡便で取扱い易いが、総合的な見地から干渉回避の最適化ができないため、AGVが本来有する運行の柔軟性を十分には引き出すことができない。従って、一つの搬送ネットワーク上に多くのAGVが投入された場合等の大規模化、複

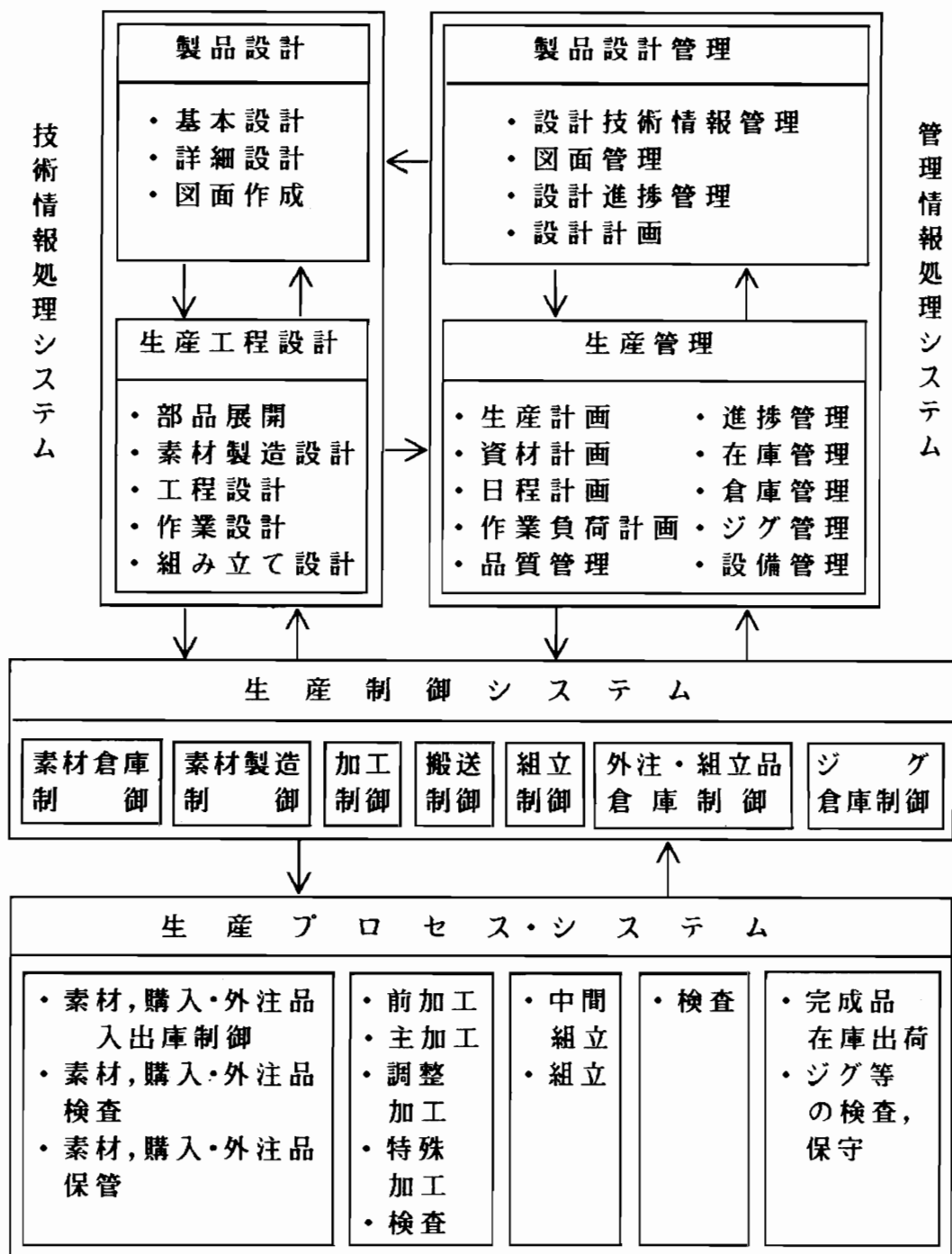


図 1. 1 生産システムのソフトウェア体系 ¹³⁾

雑化した搬送系においては干渉の多発化が予想され、これまでの運行制御方式ではこれに十分対応するのは困難であると思われる。表1. 1¹⁾は、工場におけるAGVシステムの導入実績を示したものである。表に見られる導入台数の増加は、今後も続くものと予想され、多発化の予想される干渉を効率よく解消する新しい運行管理法が望まれている。この問題に対する解決法を提案するのが本研究のねらいである。すなわち、ソフトウェアによりAGVの運行管理・制御を行い、柔軟で効率的な搬送を実現することである。ここで取り扱う運行管理法は、図1. 1の中では搬送制御に属するソフトウェア群に位置するものである。

表 1. 1 AGV 導入実績¹⁾

年	システム数	台車数
～76	430	736
'77	50	88
'78	49	94
'79	76	126
'80	83	155
'81	118	393
'82	191	423
'83	291	638
'84	321	686
'85	356	938
'86	347	755
計	2,312	5,032

最後に、現実的な問題としてのAGVシステム導入を考える。この問題に関しては導入による費用対効果の議論は避けて通れないが、それは、導入ラインの形態、環境、カスタマイズ化の程度により異なり、一般的に論じることとは不可能である。しかし、導入の際に考慮すべきことについて、いくつかの導入例に関するケーススタディが報告されている²³⁾。それによると、AGVメーカーの提供したシステムの大半が約3,000万円以下の価格であり、そのコスト構成は、AGV本体をはじめとするハードウェアに35%、通信制御関

連に30%、設置及びテスト費用に10%、その他のソフトウェアに25%となっている¹⁾。この数字から、ハードウェアそのもののコストだけでなく、通信制御、ソフトウェア関連のコストも高いことがわかり、現実システムとしてAGVシステムを構築するに際しては、高いコストパフォーマンスをもつソフトウェアが求められていることが認識できる。

本研究では、無人搬送車の運行管理法を研究対象とする。工場の搬送系内で生じたワーク等の搬送要求に対し、効率的にAGVを運行させる問題（運行問題）の解決に資するための提案を行うことを具体的な目標とする。

次章以降では、以下のことを議論する。

第2章においては、運行問題を伝統的な“経路・スケジュール決定問題”の中において位置付けし、あわせてこれまでの研究成果についてのサーベイを行う。また、運行問題特有の性質を明かにすることにより、この問題を研究対象とすることの価値について述べる。また、この問題を二つの問題（割当て問題と運行経路決定問題）に分割して定義する。

第3章～5章においては、運行経路決定問題に関する解法アルゴリズムの提案を行う。

第3章においては、まず、複数台車のうちの一台に搬送要求が与えられた場合、走行時間を最小化する運行経路決定問題を取り扱う。この問題をタイムウインドウをもつネットワークにおける最短時間経路問題（台車運行型SPPTW）にモデル化した後、これを分枝限定法を用いて解く方法を提案する^{24) - 28)}。その後、この解法で用いられる下界評価法を検討することにより、計算時間の面でより効率的な解法への改善を考える^{29) 30)}。

第4章においては、3章で取り扱った台車運行型SPPTWを他の問題に拡張することを試みる。まず、この問題をAGVの走行時間の非負性を仮定しない場合の問題に一般化することにより、3章で提案した解法を一般化するとともに、台車運行型SPPTWが古典的な最短経路問題と同じ構造をもつことを明らかにする^{31) - 33)}。次に、ここで提案した方法により決定された経路に沿って走行中のAGVに遅延が生じた場合、走行スケジュールを再スケジュールリングする手順に関する提案を行う³⁴⁾。4章の後半では、搬送要求に複数の目的地が含まれる場合や、目的地において作業時間のある場合の現実的な搬送要求に対する運行経路決定法についての考察を行う³⁵⁾。

第5章では、複数搬送車に与えられた搬送要求に対する運行経路決定法を、組合せ論的観点から提案する。決定に際して最適化する評価尺度としては、

全搬送車の総走行時間最小化と最大走行完了時刻最小化の二つを採用した。ここで提案する運行経路決定法は、搬送車の代替経路をグラフ論的手法により探索する過程と、線形計画問題を解くことにより代替経路上における搬送車の待ち停止時間を決定する過程とにより構成される³⁶⁾⁻⁴⁰⁾。上述したように、搬送車の運行経路決定に際しては、搬送車間の干渉回避を可能とすることが不可欠の条件である。5章の後半では、上で提案した経路決定アルゴリズム中に組み込んで使用するヒューリスティックな三つの干渉回避法を提案する⁴¹⁾⁻⁴³⁾。

第6章では、同時に発生した複数の搬送要求をどの搬送車に割り当てるかという割当て問題に対して、Min-Maxマッチング問題の解法を用いた割当て法を提案する。

第2章 無人搬送車の運行問題

2. 1 緒言

無人搬送車（以後、台車という）の運行問題は、“Routing and Scheduling Problem”（経路・スケジュール決定問題）という伝統的な問題のなかで論じることができる。この章では、まず経路・スケジュール決定問題とよばれる問題群について概観する。2. 6節以降において、本研究で取り扱う台車の運行問題を具体的に定義することにより、この問題を経路・スケジュール決定問題の中で位置付ける。また、運行問題に固有の性質について明かにし、この問題を研究対象とすることの意義について述べる。また、次章以降でこの問題を取り扱う準備として、問題のモデル化を行う。

2. 2 経路・スケジュール決定問題の構成要素

経路・スケジュール決定問題は、現実社会におけるロジスティックの問題においてしばしば現われる問題である。ロジスティックとは、製品やサービスを供給点から需要点に送る機能を意味する。例えば、部品や原材料を工場に運送する活動や工場から出荷された製品を小売店、卸に配送する活動、学校の生徒をスクールバスで送り届ける活動等きわめて広い分野を含む活動である。このロジスティック機能の中には、大きく分けて三つの意志決定のレベルが考えられる⁴⁴⁾。ストラテジック、タクティカル及びオペレーショナルな意志決定である。先の工場から小売店への出荷の例でいえば、第1のストラテジックな意志決定の中には、工場の立地場所や配送センターの位置を決定することが含まれる。第2のタクティカルな意志決定の中には、どの工場の製品をどの小売店にどれだけ配送するかという計画が含まれ、第3のオペレーショナルな意志決定の中には、何台のトラックで具体的にどのルートを通って配送するかという計画が含まれる。しかし、これらのレベルとそれらに属する意志決定の種類と手順について一般的に規定することは困難である。配送センターの位置は、具体的にどれだけ数のトラックによってどのルートによる配送が効率的かが検討された後に議論すべき種類のものかもしれない。逆に、他の外部要素によってセンターの位置は最初に確定され、その後具体的な配送計画が立案される手順が採用されるかもしれない。

多くの現実的問題のタクティカルなレベル及びオペレーショナルなレベルにおいてしばしば現われる問題が、経路・スケジュール決定問題である。この問題は、上述したような企業等の私的活動ばかりでなく、公的な場面でもいろいろな問題に現われており、ゴミ収集に際しての収集車の運行問題、バス等の大衆輸送手段の運行問題、都市における警察、消防署、緊急医療施設への輸送問題、老人・身体障害者の輸送の問題等様々な形態をもっている。表2. 1は、問題を構成する要素とその具体的な選択肢についてまとめたものである。但し、バスやトラックといった輸送、運行の主体となるものを一括してビークルと呼んでいる。

表2. 1 経路・スケジュール決定問題の構成要素

要素	選択肢
搬送ネットワーク	有向ネットワーク, 無向, ミックス, ユークリッド
ビークルのサイズ	1台, 複数台
ビークルのタイプ	同種, 異種, 特殊
ビークルの容量制約	あり (すべて同一, 異なる), なし
ビークルの格納場所	1カ所 (1デポ), 複数カ所 (マルチデポ)
需要の性質	確定的, 確率的, 分割可能
需要点の位置	ノード上, アーク上, 混在
最大ルート時間制約	あり (すべてのルートで同一, 異なる), なし
業務内容	集荷のみ, 配達のみ, 集荷・配達, 分割配達 (あり, なし)
コスト 評価尺度	変動費 (ルートコスト), 固定費, 回送費 トータルコスト最小化, 固定費と変動費の総和最小化, 使用ビークル数最小化, サービスの実体に関して設定した評価尺度の最適化, 顧客の満足度指数の最適化

問題の様相は異なるが、問題において決定すべき要素は同じであり、一つは経路 (Route) であり、他の一つはスケジュール (Schedule) である。従って、これらの問題は経路・スケジュール決定問題と呼ばれるのである。また、どの要素を決めるかによって、これらの問題は大きく三つの問題群に分類できる。すなわち、経路決定問題 (Routing Problem)、スケジュール決定問題 (Scheduling Problem) 及び両方の要素を同時に決める混合問題 (Combined Routing and Scheduling Problem) である。以下では、それぞれの分類に属する代表的な問題の概要を述べる。

2. 3 経路決定問題

経路決定問題は経路だけを決定する問題であり、次のような問題が含まれる。

- 1) The Traveling Salesman Problem (TSP)
- 2) The Multiple Traveling Salesman Problem (MTSP)
- 3) The Chinese Postman Problem (CPP)
- 4) The Vehicle Routing Problem (VRP)

TSPは、ネットワーク上のオフィスを表わすノードにいる一人のセールスマンが、オフィスを起点としてすべてのノードにいる顧客をただ1回訪問した後、再びオフィスに戻ってくる経路のなかで最短時間のものを見つける問題であり^{45) - 49)}、MTSPは、この問題を複数のセールスマンで手分けして顧客を回るように拡張した問題^{50) 51)}である。Karp⁵²⁾, Papadimitriou⁵³⁾ やGarrey⁵⁴⁾ が証明したように、TSPはNP完全な問題であり、大規模な問題に対しては、その最適経路を求める厳密解法よりは、少ない計算時間で準最適解を求めるヒューリスティックな解法^{47) 55) - 65)}の方に多くの研究がなされている。厳密解法としては、Held and Karp^{66) 67)} による分枝限定法を用いた解法が効率的であるといわれている。彼らは、問題を0-1整数計画問題に定式化し、これにラグランジェの緩和法⁶⁸⁾ 及び劣勾配法^{69) 70)} を適用することにより、有効な下界を利用する方法を採用している。MTSPは、基本的にはTSPに帰着できることは知られている^{71) 72)} が、実用的な計算時間内で実行可能解を求めることはTSP以上に困難である。TSPやMTSPは、需要点がノードにあるような多くの問題に応用できる。

CPPは、すべてのアークを少なくとも1回通過する経路の中で最短時間

のものを見つける問題である。この問題は、アークが無向または有向である場合多項式時間で解ける⁷³⁾⁻⁷⁵⁾が、無向アークと有向アークが混在したネットワークにおいてはNP完全となることが知られている。この問題は、家ゴミや工場廃棄物の収集、道路清掃の計画問題等、需要点のアークにあるような問題に応用できる。

VRPとしては、前節で述べた工場から小売店への品物の配達計画問題を例にとることができる。ある一つのデポから出発した異なる積荷容量をもつ複数台のトラックが、工場を經由し、すべての小売店にそれぞれの注文量を配達した後帰ってくるものとする。VRPは、コストとして運送費用のみが加算されるものとして、トータルコストが最小になるように各トラックの経路を求める問題である⁷⁶⁾。NP完全なこの問題に対して、Christofides⁷⁷⁾による厳密解法が提案されている。また、数多くのヒューリスティックな解法も提案されているが、その手法の主なものには次のものがある。

・Cluster First-Route Second Procedure: まずトラックに割り当てる小売店群を決め、次にそれぞれのトラックごとに最適な経路を決める方法⁵⁹⁾⁷⁸⁾

・Route First-Cluster Second Procedure: まずすべての小売店を巡回する経路を決め、次にそれを適当ないくつかの巡回経路に分ける方法⁷⁹⁾⁸⁰⁾

2. 4 スケジュール決定問題

スケジュール決定問題は、いわば時間的な制約のあるタスクをネットワーク上のノードに見立てて、その経路を決定する問題である。興味ある問題として、次のような問題がある。

1) The Vehicle Scheduling Problem (VSP)

2) The Crew Scheduling Problem (CSP)

VSPは以下のような問題である。開始時刻、終了時刻の与えられた仕事(タスク)の集合がネットワーク上のノードとして表現されている。また、アークは、それが結ぶ二つのタスクを連続して実行できることを意味しているものとする。いま、ある一つのノード(デポ)からタスクをその開始・終了時刻間の先行関係に矛盾を生じないようにつないでゆき、再びデポに戻る。このようないくつかの閉路により、すべてのタスクをカバーする。但し、一つの閉路は一つのビークルにより担当するものとする。VSPは、ビークルを割り当てることにかかる固定費用とノード間の運行費用との総費用を最小に

するスケジュールを求める問題である。この問題には、多項式オーダーの解法アルゴリズム⁸¹⁾が提案されているが、一つの閉路を構成するタスクの完了時刻に制約がある場合や複数のデポのある場合に拡張された問題は、NP困難になることが知られている⁸²⁾。

CSPは、文字どおり乗員や作業員をあるタスクに関しスケジュールする問題であり、興味ある問題として次のようなものがある。

・The Fixed Location Worker Scheduling Problem^{83),84)}：電話案内、交換サービスを行うための交換手の割当て問題である。1日の交換需要形態とそれに必要なサービス量とが予め分かっているものとし、また、交換手の勤務形態も既知であるものとした場合、各勤務形態に割り当てる交換手の人数を決定する問題である。

・The Mass Transit Crew/Vehicle Scheduling Problem⁸⁵⁾：路線バス運行のため、乗員の交替やバスへの割当てをスケジューリングする問題である。バスの運行経路と運行スケジュールは既知であるとした場合、食事の摂取や労働時間等いくつかの労働協約を遵守しながら、バスを運行させるための乗員の交代時間及び交代場所を決定する問題である。

・The Air Crew Scheduling Problem⁸⁶⁾：航空機運行のため、乗員の乗務をスケジューリングする問題である。各種の労働協約を遵守しながら、既知である航空機の運行スケジュールに沿ったクルーの乗務計画を立てる問題である。

2. 5 混合問題

混合問題は、経路及びスケジュールをともに決定する問題であり、あるネットワーク上での経路決定を、経路を構成するノードの先行関係や時間的な制約の下で行わなければならない場合の問題である。例えば、スクールバスによる生徒の送り迎えの問題において、スクールバスの運行経路を考えると、生徒を拾うノードは届け先の学校のあるノードよりも先に経由しなければならない（先行関係）。また、学校の始業時間までに学校に到着するような運行計画を立てなければならない（時間制約）。これに分類される問題の中で興味あるものとして、次のような問題がある。

- 1) スクールバス問題
- 2) トラクター・トレーラー問題

3) 家ゴミ収集問題

4) 航空機問題

5) ダイヤルーアンドーライド問題

スクールバス問題は、上述したように、スクールバスによる学校への生徒の送り迎えにおいて運行コストを最小化する問題である⁸⁷⁾⁸⁸⁾。トラクター・トレーラー問題は、トラクターにより牽引されるトレーラー内にあるノードで品物を集荷し、それをある目的ノードへ配達する問題である。もし、集荷地点においてトレーラーの積荷が常に満載になれば、集荷した後は必ず配達するという先行関係のある経路決定問題⁸⁹⁾になる。トレーラーが混載可能であれば、複数のノードにおいて集荷した後それぞれに対応する目的ノードへの配達を集荷とは逆の順序で行うという先行関係のある問題になる⁹⁰⁾。家ゴミ収集問題は、家庭から出されたゴミを複数の収集車で集めていく経路を求める問題であるが、道路の交通規則（駐・停車禁止時間帯等）を守りながら、不確定な量のゴミをできるだけ効率よく集めなければならない⁹¹⁾。航空機問題は、主に航空機による各都市からの貨物の集配を効率的に行う航空機の飛行ルートを、不確定な貨物量や速達便／普通便の区分を考慮に入れながら決定する問題である。ダイヤルーアンドーライド問題は、顧客からの要求に応じて車を配車する問題である。顧客は、彼の乗車場所と下車場所及び必要であれば乗車時間とを指定する。配車係は、緊急な顧客の要求に対してはリアルタイムな配車を実施し、予め予約している顧客に対しては事前の配車計画を立てなければならない。

2. 6 無人搬送車の運行問題

台車の運行管理や走行制御を行う上において最も重要な問題は、ワーク等を搬送ネットワーク上のある地点からある地点へと搬送する要求（搬送要求）に対し、なんらかの評価尺度のもとで、それを利用可能な台車に割当て（割当て問題）、最適な経路と走行スケジュールを搬送ネットワーク上で決定する（運行経路決定問題）ことである。以後、この二つの問題を総称して台車の運行問題と呼ぶ。

ここでは、台車の運行問題を一般的な経路・スケジュール決定問題の中で位置付けることにより、他の問題との共通性及びその固有の性質について議論する。

2. 6. 1 干渉回避問題

無人搬送システムのハードウェアの特徴については、第1章において述べた。その搬送路全体としては、あらかじめ埋設された制御用ケーブルに沿った搬送路網、または、たとえ自律走行可能な搬送システムであっても制御用の各種センサーの設置された搬送路網を想定すればよい。ある1本の搬送路上で複数の台車が行き会う場合、人間や自動車のようにお互いが衝突を回避するような動きをする高度に自律化された台車は、これまでのところ実用化されていない。従って、そのような衝突を防ぐためには、同じ搬送路上を複数の台車が行き会わないように運行を制御する必要がある。2. 3～2. 5節における経路・スケジュール決定問題では、このような局所的な干渉回避を陽に考慮する必要はなかった。それらの問題では、搬送路網として通常の道路を、運搬や搬送を担当する主体として人間や自動車を暗黙の内に仮定しており、搬送路上での干渉回避、例えば交差点等で衝突しないで行き会うことが自律的に可能であることが当然視されるような性質の問題であるからである。また、たとえ衝突回避のためにある時間を要するものとしても、運搬や搬送全体に要する時間からみればその時間は無視できるからである。

このような問題に対し、本論文で取り扱う工場内の無人搬送車システムにおいては、この搬送路上での干渉回避がシステムの運行効率に及ぼす影響は大である。なぜなら、搬送主体としての無人搬送車は、高度に自律的な干渉回避能力に欠けているのが普通であるからである。また、搬送路全体の規模はせいぜい工場全体を覆う程度であり、しかも、工場という製品の生産・組立のための数々の機器が集約的に集められた施設のなかにあつては、それら機器を結ぶ搬送系にはきわめて高密度の干渉発生点が潜在的に存在するものと考えられるからである。従って、この搬送システムにおける干渉をいかに効率的に解消するかということが、システム全体の搬送効率を決定する重要な要因になるのである。干渉回避のためには、1章で述べたように、ブロッキングゾーン制御方式のような簡便な干渉回避法や、搬送路を複線化したり待避路や迂回路を設定する等の搬送路網に柔軟性を与える方法が考えられる。本研究では、与えられた搬送網上で、できるだけ無人搬送車の柔軟性を引き出す運行ソフトウェア開発を目指し、簡便な干渉回避よりは、効率的な回避法を考える。搬送路網を変更する方法に関する問題は、搬送系の設計問題に属し、それだけで大きな問題であり、本論文では取り扱わない。

前述した干渉問題があるため、たとえ一台の台車に対し運行経路を決定す

る問題であっても、干渉が起こらないことを保証する運行経路の指示には、どの搬送路をどの順序で通過するかという経路指示だけではなく、他の台車との干渉発生を事前に防止するために、どの停止点でどれだけの時間待つかという待ち停止時間（または停止点での到着時刻、出発時刻）に関する指示を与える必要がある。前者はいわば経路の経由点を決定する問題であり、後者は走行スケジュールを決定する問題である。両者の問題を総称して“台車の運行経路決定問題”と呼ぶと、それは一般的な経路・スケジュール決定問題の中では、2.5節で述べた混合問題に分類されるべき問題である。

もし、複数の搬送要求に対し複数の台車が利用可能な状況にあれば、どの台車をどの要求に割り当てるかという“割当て問題”を考えなければならない。この問題は、2.3節のMTSP、2.4節の複数ビークルのある場合のVSPや2.5節の複数バスによるスクールバス問題といった2.3節～2.5節の経路・スケジュール決定問題の三つの分類のいずれに属する問題であっても、複数ビークルが存在する場合には必ず考慮しなければならない問題である。

台車の運行問題は、以上述べた運行経路決定問題と割当て問題を解く問題であるといつてよい。

2.6.2 搬送システムのモデル

ここでは、経路・スケジュール決定問題の構成要素の表2.1を用いて、無人搬送車の運行問題を定義する。第1章で述べたように、無人搬送車の搬送路網は、ロード／アンロードステーション、バッファ、ホームポジションや搬送路の分岐点のように台車の停止可能な地点をノードとし、ノードを結び台車の走行のみ可能な搬送路を無向または有向アークとするネットワークにモデル化できる。図2.1はその例である。ノードの集合を N 、アークの

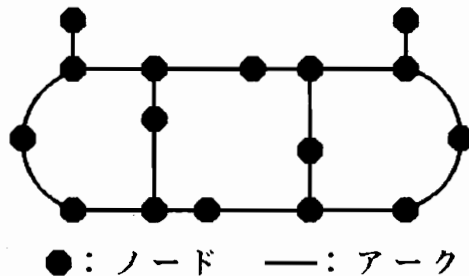


図2.1 搬送ネットワークモデル

集合をA、この搬送ネットワークをG(N, A)で表す。ノードiとjを結ぶアーク(i, j)上を走行する台車の走行時間はどの台車に対しても同一であると仮定し、これを実数 d_{ij} で表わす。

このネットワーク上を走行する複数台車間における衝突は、ある近接限界を越えた接近により起こるものとし、その台車間の近接限界を時間の単位をもつ ϵ （最小干渉許容時間）で表わす。従って、衝突の発生する条件として、次の二つの場合が考えられる。

1) 2台以上の台車が、同一ノード上で ϵ 未満の時間帯に存在する場合。

2) 2台以上の台車が、双方向走行可能な同一搬送路上を、同時に、かつ互いに向かい合って走行する場合。

複数台の台車が経路決定の対象となるが、ここでは、ビークル（台車）はすべて同種のものとし、また、積載容量も無制限と仮定することにより積載量に関わる問題を陽には考えないこととする。

工場内では、台車に対するワーク等の搬送要求が連続して生じる環境にあり、それに伴って台車の出発地点も到着地点も頻繁に変化するのが普通である。従って、運行問題においても、複数の台車の各出発・到着デポはそれぞれ異なるものとする。ビークルに対する搬送要求には、どこで、何を、どれだけ集荷し、どこへ、何を、どれだけ配達するかという情報が含まれている。しかし、積載容量の制約を無視することを前提としているため、上の情報のうち、場所に関するもののみを問題の対象とする。すなわち、どの場所を通過するかという情報のみから成る搬送要求を考える。また、工作物の搬送は、あらかじめ決まった加工順序で加工機械の設置されたノードに工作物を供給するためのものである場合が多い。従って、搬送要求はノードでのみ発生し、あらかじめ順序付けられた確定的な経路ノードから成るものとする。以上のように、本論文の対象とする無人搬送車の運行問題のモデルの具体的な構成は、表2. 1の構成要素を用いて述べることができる。

2. 6. 3 評価尺度

搬送システムにおいて最も重要視されるのは搬送時間である。工場における生産では、単位時間当りの生産性や製品の納期内出荷が重要であり、そのためには、加工・組立工程での工程時間短縮はもとより、搬送時間の短縮化が望まれる。従って、一台の台車の運行経路を決める場合には、搬送要求を満足する経路の内、最小走行時間をもつ経路が最も重要である。複数台車に

対する運行問題に際しては、走行時間の観点から、二つの最適化の評価尺度が考えられる。その第1は、すべての台車の走行時間の総和（総走行時間）を最小化すること。その第2は、すべての台車のうちの最も遅い走行完了時刻（最大走行完了時刻）を最小化することである。第1の評価尺度は、搬送を含めた全製造時間の短縮や、搬送車の一般的なエネルギー源である電力の消費量縮小による搬送のランニングコスト削減を図ることにつながるものである。第2の評価尺度は、搬送を含めた製造工程の完了時刻（メイクスパン）の短縮化につながるものである。この研究では、以上の評価尺度を取り扱う。

2. 6. 4 割当て問題と運行経路決定問題

無人搬送車の運行問題においては、複数の搬送要求のどの要求をどの利用可能な台車に割当て（割当て問題）、割り当てられた要求を台車がどのような具体的な経路で実現するか（運行経路決定問題）という二つのレベルの意志決定が必要であることは先に述べた。運行問題を全体の最適化を考慮しながら解くためには、この二つの問題を同時に最適化しなければならないが、2. 2～2. 5節で述べた他の多くの経路・スケジュール決定問題でもそうであるように、それは手法的にも計算時間的にも困難である。また、無人搬送車に関する限り、運行経路決定問題でさえこれまでほとんど取り扱われておらず、それを研究対象とする意義は大きい。それは、一般的な経路・スケジュール決定問題では、ビークルがある特定の経路を通るのに要する時間は簡単に求められるのに反して、無人搬送車に関する問題には干渉回避という要因が含まれるため、台車が特定の経路を通過すると仮定しても、その走行時間を求めることは容易でないからである。以上の事情を考え、本論文では、割当て問題と運行経路決定問題とをそれぞれ分離して論じることにする。このやり方は、2. 3節で述べた Cluster First-Route Second Procedure の手順を踏襲するものである。次章以降において、便宜上、まず運行経路決定問題を議論した後、割当て問題について述べることとする。また、運行経路決定問題は、錯誤のおそれがない場合、単に経路決定問題ということもある。

2. 7 結言

この章では、これまでに研究されてきた経路・スケジュール決定問題の中で、無人搬送車の運行問題を位置づけることを試みた。その結果、混合問題

と呼ばれる問題群に分類できることがわかった。それは、無人搬送車に関する問題には干渉回避という特有な性質の要素が含まれるため、その経路決定には待ち停止時間の決定というスケジュール決定問題が必然的に伴われるからである。

第3章 一台車に対する運行経路決定問題

3.1 緒言

ここでは、複数台車中の搬送要求を受けた一台についてのみ新たに経路を決定するという基本的かつ単純な運行法を対象として、その場合の運行経路決定法について研究する。まず、この運行法に基づく台車の経路決定問題をタイムウィンドウを伴う最短時間経路問題としてモデル化する。この問題に対して、実行可能解の存在の有無を厳密に判定でき、実行可能解が存在する場合には最適経路を多項式時間で見つけることのできる分岐限定法を用いた解法アルゴリズムを提案する。次に、解法の中で用いられる下界評価法として新たな評価法を考案することにより、解法による計算時間の効率化を図る。最後に、提案した解法アルゴリズムを用いた具体例を述べる。また、シミュレーションにより、提案した三つの下界評価法の有効性について比較検討した例を述べる。

3.2 最短時間経路決定問題とタイムウィンドウ

3.2.1 台車の最短時間経路決定問題

台車による搬送系は、2.6.2節で述べたように、アークとノードから成るネットワークとしてモデル化できる。無向アークは互いに逆の方向をもつ2つの有向アークとして表現できるから、このネットワークは有向ネットワーク $G(N, A)$ であるとして差し支えない。 N はノードの集合、 A は有向アークの集合である。また、 d_{ij} はアーク (i, j) の走行時間であり、非負の実数で表されるものとする。

いま、 P 台から成る台車群 $Q = \{Q_i; i=1, \dots, P\}$ が存在する搬送系において、ノード s に停止している台車 Q_i に対して、そのノード（出発ノード）をある時刻（出発許可時刻）以降に出発し、ある一つの目的ノードへ到着せよという搬送要求が与えられた場合を想定する。但し、 $Q_i \in Q - \{Q_i\} = Q'$ である他の台車 Q_j は、停止中であるものを含めて既に決定されている走行ノード列（アーク列）及びノードでの待ち停止時間に従って走行しているものとし、それらの変更は考えないこととする。台車間の干渉は、2.6.2節の条件1), 2) の場合に生じるものとし、最小干渉許容時間は ε であるとする。

以上のような設定の下での台車の最短時間経路決定問題とは、台車 Q_k が出発ノード s を出発許可時刻 t_0 以降に出発し、他の台車との干渉を回避しながら目的ノード e に最も早い時刻（以後、最早到着時刻と呼ぶ）で到着できるようなノード列と各ノードにおける干渉回避のための待ち停止時間を求めることである。以下では、ノード列とノードでの待ち停止時間の両者を一対として解と呼ぶものとする。

3. 2. 2 通過可能時間帯

前節で述べた台車の最短時間経路決定問題に対し、次のような解法の手順が考えられる。

1) 各ノード、アークにおいて $Q_k \in Q'$ と干渉を起こさないような Q_k の通過可能な時間帯を求める。

2) 1) で求めた通過可能な時間帯の制約を満足するノード列及び待ち停止時間のうち、最早時刻で目的ノードに到着するものを探す。

上述の手順1)での通過可能な時間帯は、最小干渉許容時間 ϵ と Q' の台車のノード列及び待ち停止時間から次のように計算できる。以下においては、台車 $Q_k \in Q'$ のノード列を L_k 、アーク列を L'_k とし、ノード i が L_k 内にあることを $i \in L_k$ 、アーク (i, j) がアーク列 L'_k 内にあることを $(i, j) \in L'_k$ で表す。

(1) 台車 $Q_k \in Q'$ のノード列及び待ち停止時間から各ノードにおける Q_k の到着、出発時刻を求める。

(2) 台車 $Q_k \in Q'$ のノード $i \in L_k$ での到着、出発時刻を (t_a, t_d) とする。この時、時間帯 $(t_a - \epsilon, t_d + \epsilon)$ 間は、干渉のため他の台車は通過禁止となる。従って、他の台車のノード i における通過可能な時間帯（通過可能時間帯） $T^k(i)$ は、

$$T^k(i) = \{(-\infty, t_a - \epsilon], [t_d + \epsilon, \infty)\} \quad (3-1)$$

となる。同様に、台車 Q_k がノード i で停止中であり、時刻 t_d に出発する場合の通過可能時間帯は、

$$T^k(i) = \{[t_d + \epsilon, \infty)\} \quad (3-2)$$

となり、 Q_k がノード i へ時刻 t_a に到着以降そこで停止する場合の通過可能時間帯は、

$$T^k(i) = \{(-\infty, t_a - \varepsilon]\} \quad (3-3)$$

となる。

(3) 台車 $Q_k \in Q'$ がアーク $(i, j) \in L'_k$ を走行するとする。アーク (i, j) へ進入の禁止される時間帯は、その入口において $(t_d - \varepsilon, t_d + \varepsilon)$ で与えられるが、これは、ノード i における通過禁止の制約を守ることによって満たされる。

もし、 $(i, j), (j, i) \in A$ であり、これらのアークの表す搬送路が双方向走行可能であるならば、この搬送路を逆に走行する台車に対して制約を考えなければならない。台車 Q_k がアーク (i, j) を走行してノード j に到着する時刻は $t_d + d_{ij}$ である。また、台車 Q_k がノード i に到着する時刻 t_a 以前にノード j からアーク (j, i) を通ってノード i を通過するためには、他の台車は遅くとも時刻 $t_a - d_{ji}$ にはノード j を出発しなければならない。従って、 (j, i) の入口における進入禁止時間帯は $(t_a - d_{ji} - \varepsilon, t_d + d_{ij} + \varepsilon)$ となる。以上から、アークへの進入可能な時間帯（進入可能時間帯） $S^k(j, i)$ は、

$$S^k(j, i) = \{(-\infty, t_a - d_{ji} - \varepsilon], [t_d + d_{ij} + \varepsilon, \infty)\} \quad (3-4)$$

となる。

(4) 手順(2), (3)を $Q_k \in Q'$, $i \in L_k, (i, j) \in L'_k$ に対して行い、ノード、アーク毎に、得られた通過可能時間帯、進入可能時間帯の数直線上における共通部分をとることによって、台車 Q_k の通過可能時間帯 $T(i)$, 進入可能時間帯 $S(i, j)$ ($i, j \in N$) を求める。

3. 2. 3 タイムウィンドウ

前節の手順(4)で求めた各ノード、アークでの通過可能時間帯、進入可能時間帯は、これまで最短経路問題において研究されてきたタイムウィンドウ⁹²⁾の一種と考えることができる。

タイムウィンドウのある場合の最短経路問題⁹³⁾ (SPPTW: Shortest Path Problem with Time Window) は、当初、タイムウィンドウを伴う場合の巡回セールスマン問題やその他の経路・スケジュール問題^{94) - 97)}の部分問題として研究された⁹²⁾。例えば、巡回セールスマン問題において、各ノードに存在する顧客がセールスマンと面会するのに都合のよい時間帯をタイムウィンドウとして設定するもので、セールスマンは、タイムウィンドウ中に顧客

との面会を済ませられるように、アークでの所要時間を考慮してノードへの到着を計画しなければならない。従って、従来の研究におけるタイムウィンドウは、次のような性質を有していた⁹²⁾⁹⁴⁾。

(1) タイムウィンドウは、ノードのみに存在する。

(2) タイムウィンドウは、人間や自動車がこのノードを出発する時刻に対してのみ制約となり、到着時刻に対しては何ら制約とはならない。

(3) 各ノードでのタイムウィンドウは高々1つの時間区間しか持たない。

従来のSPPTWに対して、Desrosiersらは一般化した動的計画法による解法を提案している⁹²⁾。また、Desrochers and Soumis⁹⁸⁾は、ダイクストラ法を一般化したラベルセッティング法を用いて、さらに計算時間の速いアルゴリズムを提案し、ノード数2500、アーク数25000で構成されるネットワークにおける最短時間経路を求めている⁹²⁾。

これに対し、本研究では、各ノード、アークにおける台車の干渉を防止するための通過可能時間帯としてタイムウィンドウの導入を考える。従って、このタイムウィンドウは、次のような性質を有することとなる。

(1') タイムウィンドウは、ノードばかりでなくアークにも存在する。

(2') タイムウィンドウは、出発時刻ばかりでなく到着時刻に対しても制約となり、タイムウィンドウ中のある時間区間に到着した台車は、同じ時間区間内に出発しなければならない。

(3') 各ノード及びアークのタイムウィンドウは一般に複数個の時間区間から成る。

このような性質の違いのため、このタイムウィンドウを伴うSPPTWは、従来のそれにはなかった次のような解の複雑性を考慮に入れなければならない。

(a) 最短時間経路の部分経路が必ずしも最短時間経路とはならない。従って、部分経路の最適性に基づく動的計画法を用いた解法⁹⁹⁾は本問題には適用できない。

(b) 最短時間経路がループを形成する場合がある。

本論文では、タイムウィンドウの性質として前述の(1')～(3')を考え、その場合のSPPTWに対し分岐限定法を用いた解法アルゴリズムを提案する。以下では、従来のSPPTWと区別するため、ここで取り扱うSPPTWを台車運行型SPPTWと呼ぶことにする。

3.3 台車運行型SPPTW (Shortest Path Problem with Time Window)

3.3.1 モデルの設定

3.2.1節で定義した最短時間経路決定問題を考える。ノード*i*での台車*Q*の通過可能時間帯を表す有限個(n_i)の時間区間をノード*i*のタイムウィンドウとし、次式の $T(i)$ で定義する。

$$T(i) = \{T_{ik} = [t_1^{ik}, t_2^{ik}]; k=1, \dots, n_i\} \quad (3-5)$$

但し、

$$t_1^{ik} \leq t_2^{ik}, k=1, \dots, n_i, t_2^{ik-1} < t_1^{ik}, k=2, \dots, n_i \quad (3-6)$$

である。

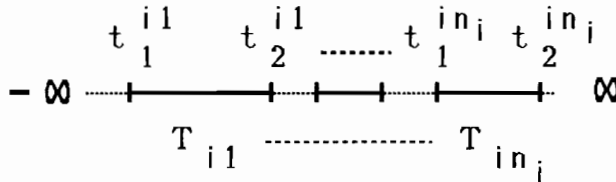


図3.1 数直線上の閉区間集合 $T(i)$

式(3-5)は、図3.1のように数直線上に並んだ複数の閉区間からなる閉区間集合を形成する。但し、 R を実数全体の集合とすると、最左端の区間として $(-\infty, a_1]$ 、最右端の区間として $[a_2, \infty)$ ($a_1, a_2 \in R$)を許すものとする。このようにして表される任意の閉区間集合よりなる集合族を K で表す。同様に、アーク(i, j)に対しては、そのアークの入口で、台車*Q*の進入可能時間帯を表す有限個(m_{ij})の時間区間から成るタイムウィンドウ $S(i, j)$ を、式(3-5)と同様の形式で次のように定義するものとする。

$$S(i, j) = \{S_{ij,k} = [s_1^{ij,k}, s_2^{ij,k}]; k=1, \dots, m_{ij}\} \quad (3-7)$$

但し、

$$s_1^{ij,k} \leq s_2^{ij,k}, k=1, \dots, m_{ij}, s_2^{ij,k-1} < s_1^{ij,k}, k=2, \dots, m_{ij} \quad (3-8)$$

以上のような設定の下での台車運行型SPPTWとは、台車が出発ノード s を出発許可時刻 t_0 以降に出発し、ノード、アークをそれぞれのタイムウィンドウの制約内に通過しながら目的ノード e へ最早時刻に到着するような経路（ノード列及び待ち停止時間）を求めることである。

3.3.2 演算の定義

図3.1のように表される区間の集合全体の集合族を K とする。また、 R を実数全体の集合とし、空でない X 、 U を、

$$\begin{aligned} r &\in R \\ X &= \{[x^{k_1}, x^{k_2}]; k=1, \dots, p\} \in K \\ U &= \{[u^{h_1}, u^{h_2}]; h=1, \dots, q\} \in K \end{aligned} \quad (3-9)$$

で表す。このとき、 K 、 R 上の単項演算子 $\underline{\min}$ 、 $\underline{\max}$ 、2項演算子 \oplus 、 \ominus 、 \otimes を次のように定義する。

$$\underline{\min} X = x^{1_1} \in R \quad (3-10)$$

$$\underline{\min} \emptyset = \infty \quad (3-11)$$

$$\underline{\max} X = x^{p_2} \in R \quad (3-12)$$

$$\underline{\max} \emptyset = -\infty \quad (3-13)$$

$$X \oplus r = \{[x^{k_1} + r, x^{k_2} + r]; k=1, \dots, p\} \in K \quad (3-14)$$

$$X \ominus r = \{[x^{k_1} - r, x^{k_2} - r]; k=1, \dots, p\} \in K \quad (3-15)$$

$$X \otimes U = \{[x^{k_1}, x^{k_2}] \cap [u^{h_1}, u^{h_2}]; k=1, \dots, p, h=1, \dots, q\} \in K \quad (3-16)$$

また、関係演算子 \gg を次のように定義する。

$$U \gg X \Leftrightarrow u^{1_1} > x^{p_2} \quad (3-17)$$

さらに、 $r \in R$ 、 $X \in K$ なる r 、 X に対して $r \in X$ であるとは、ある k に対して $r \in [x^{k_1}, x^{k_2}]$ であることを意味するものとする。

これらの演算子に対しては、次のような関係が成立する。

$$\underline{\min} (X \oplus r) = (\underline{\min} X) \oplus r \quad (\text{複号同順}) \quad (3-18)$$

$$\underline{\min} (X \otimes U) \geq \max (\underline{\min} X, \underline{\min} U) \quad (3-19)$$

$$\underline{\max} (X \otimes U) \leq \min (\underline{\max} X, \underline{\max} U) \quad (3-20)$$

3. 4 台車運行型SPPTWの解法アルゴリズム

3. 4. 1 修正進入可能時間帯と存在可能時間帯

ノードjにおける通過可能時間帯 $T(j)$ に対し、次のような変換を考える。

$(i, j) \in A$ である全ての $i \in N$ に対し、

$$T(j) \ominus d_{ij} \in K \quad (3-21)$$

式(3-21)で表される閉区間集合は、タイムウィンドウ $T(j)$ の制約内にノードjに到着するためのアーク (i, j) への進入時刻の制約を示したものである。従って、この制約と本来の進入可能時間帯 $S(i, j)$ の制約とを同時に表現すれば次のようになる。

$$S'(i, j) = S(i, j) \otimes [T(j) \ominus d_{ij}] \quad (3-22)$$

この $S'(i, j) \in K$ をアーク (i, j) の修正進入可能時間帯と呼ぶ。このようにすれば、ノードj及びアーク (i, j) における可能時間帯の制約は、修正進入可能時間帯 $S'(i, j)$ の制約で置き換えることができる。

次に、台車が時刻tにノードiを出発してノードjへ走行しようとしている場合を考える。この時、次の2つの場合が考えられる。

(1) $t \notin S'(i, j)$ ならば、走行は許されない。

(2) $t \in S'(i, j)$ ならば、時刻 $t + d_{ij}$ にノードjに到着する。

このことから、ある時間区間V内の任意の時刻にノードiに存在することを許された台車がアーク (i, j) を通過してノードjに移動する場合、台車は次式で表される時間帯の任意の時刻にノードjに到着し得る。

$$W_{ij} = [V \otimes S'(i, j)] \oplus d_{ij} \quad (3-23)$$

ノードjの通過可能時間帯 $T(j) = \{T_{jk}; k=1, \dots, n_j\}$ に対し、 $W_{ij} \otimes T_{jk} \neq \emptyset$ であれば、

$$c_k^{ij} = \min \{W_{ij} \otimes T_{jk}\} \quad (3-24)$$

はk番目の通過可能時間区間 T_{jk} に台車の到着できる最も早い時刻を示している。従って、この台車がノードjのk番目通過可能時間区間に存在できる

時間区間は、可能な限り早い時刻 c_{jk}^{ij} にこの時間区間に到着し、以後、この通過可能時間区間の許す限りノード j に滞在することで与えられ、

$$E_{jk} = [c_{jk}^{ij}, t_2^{jk}] \quad (3-25)$$

となる。ノード i において台車の存在できる時間区間の集合

$$E(i) = \{E_{ik} = [e_1^{ik}, e_2^{ik}]; k=1, \dots, n_i\} \in K \quad (3-26)$$

を存在可能時間帯と呼ぶこととし、上のような方法である存在可能時間区間 V を用いてその隣接ノード j に存在可能時間帯を作成することを、以下では V をノード j に移行するという。 V の移行前にノード j にすでに存在可能時間帯が作成されている場合は、その存在可能時間帯と V の移行により作られる存在可能時間帯との和集合によってより大きな存在可能時間帯を作成すればよい。なお、経路決定の対象となる台車 Q は、最初出発ノード s に存在しており、出発許可時刻 t_0 以降に出発してよとの仮定から、ノード s における存在可能時間帯 $E(s)$ は1つの時間区間から成る次のような集合になる。

$$E(s) = \{[t_0, t_2^{sk*}]; t_0 \in T_{sk*}\} \quad (3-27)$$

但し、 $t_0 \notin T_{sk}, k=1, \dots, n_s$ ならば、 Q 以外の台車が s を通過している時間帯に t_0 を設定することになり問題は無意味になるので、 t_0 は s のいずれかの通過可能時間帯中に設定されるものと仮定する。

3. 4. 2 解法アルゴリズム

ここで提案する解法アルゴリズムは、出発ノード s における式(3-27)の初期存在可能時間帯を次々に隣接ノードへ移行させていき、各ノードにおける存在可能時間帯を新たに作成または修正していく。そして、目的ノードに作成できた存在可能時間帯が以後修正されないことが確認できた時点で、それに含まれる最早時刻を目的ノードへの最早到着時刻とするものである。その最早到着時刻を実現する台車の経路及び待ち停止時間は、目的ノードに作成された存在可能時間帯の移行元の存在可能時間帯を次々にバックトラッキングすることにより求めることができる。もし、目的ノードにおける存在可能時間帯が作成できなければ、実行可能解は存在しない。

このような基本的な考え方に基づいて構成された解法アルゴリズムは次の

とおりである。

(STEP 1) 全ての有向アーク $(i, j) \in A$ に対し、 $S'(i, j)$ を (3-22) 式により作成する。この時、

$$S'(i, j) = \emptyset \quad (3-28)$$

ならば、台車がアーク (i, j) を通過できる時間帯は存在しないから、アーク (i, j) を開放除去する。

(STEP 2) 存在可能時間帯 $\{E(i); i \in N\}$ を式 (3-27) を用いて、次のように初期化する。

$$\begin{aligned} E(s) &= \{[t_0, t_2^{sk}]; t_0 \in T_{sk}\}, \mu_s = 0 \\ E(i) &= \emptyset, \mu_i = 0, i \in N - \{s\} \end{aligned} \quad (3-29)$$

但し、 μ_i は、ノード i から隣接ノードへ移行された存在可能時間帯 E_{ik} の最も大きな k の値を示す。

(STEP 3) 目的ノード e へ有向パスで連結されるノードの集合を U とし、

$$L = \{i \in U; E(i) \neq \emptyset\} \quad (3-30)$$

とする。すなわち、 L は U 中で存在可能時間帯をもつ集合を表す。もし $L = \emptyset$ ならば終了。この時、実行可能な経路は存在しない。

(STEP 4)

$$e_1^{i_0 k_0 + \nu}(i_0) = \min_{i \in L, E_{ik} \neq \emptyset} \{e_1^{ik + \nu}(i)\} \quad (3-31)$$

となる (i_0, k_0) を求める。但し、 $\nu(i) (i \in N)$ は、タイムウィンドウを一切考慮しない場合のノード i から e までの最小走行時間を表す。

もし、 $i_0 = e$ ならば終了。 $e_1^{i_0 k_0}$ が求める最早到着時刻である。

また、 $i_0 \neq e$ であり、かつ、

$$E_{i_0 k_0} \oplus \nu(i_0) \geq T(e) \quad (3-32)$$

ならば、実行可能解は存在せず終了する。

(STEP 5) 存在可能時間区間 $E_{i_0 k_0}$ を使用して、次のようにアークの除

去及び隣接ノードでの存在可能時間帯の更新処理を行う。

$j \in U, (i_0, j) \in A$ である任意のノード j に対し、もし、

$$E_{i_0 k_0} \geq S'(i_0, j) \quad (3-33)$$

ならば、アーク (i_0, j) を開放除去する。そうでないならば、(3-23)式及び(3-24)式を使って、次のようにノード j の存在可能時間帯を更新する。

$$W_{i_0 j} = \{E_{i_0 k_0} \otimes S'(i_0, j)\} \oplus d_{i_0 j} \quad (3-34)$$

さらに、

$$W_{i_0 j} \otimes T_{jk} \neq \emptyset, \mu_j < k \leq n_j \quad (3-35)$$

となる k に対し、

$$c_k^{i_0 j} = \min \{W_{i_0 j} \otimes T_{jk}\} \quad (3-36)$$

とする。

もし、 $E_{jk} = \emptyset$ 、または、 $E_{jk} \neq \emptyset$ かつ $c_k^{i_0 j} < e_l^{jk}$ ならば、

$$E_{jk} = [c_k^{i_0 j}, t_2^{jk}] \quad (3-37)$$

$$\gamma(j, k) = (i_0, k_0) \quad (3-38)$$

とする。 $\gamma(j, k)$ は、存在可能時間帯 E_{jk} がどの存在可能時間帯から移行されてきたかを示すラベルである。

$$(STEP 6) E_{i_0 k_0} = \emptyset, \mu_{i_0} = k_0 \quad (3-39)$$

として (STEP 3) へ行く。

3. 4. 3 実行可能解の性質

台車運行型 S P P T W の実行可能解の性質として、次の定理が成り立つ。

[定理 3. 1] ノード s, e 以外のあるノードにおいて、ある通過可能時間区間内から出発した後、この同一時間区間内に再び戻って来るようなループをもつ実行可能解が存在すれば、このループがなく、 e への到着時刻が同じ実行可能解もまた存在する。

(証明) あるノードの通過可能時間区間内の時刻 t に到着した後出発し、ループを通過した後この同一可能時間区間内に再到着して、時刻 t' に再び出発するような実行可能解が存在するとする。この時、この解における時間区間 $[t, t']$ の部分をこのノードで台車が待ち停止するように修正した解もまた実行可能であり、かつ e への到着時刻は変わらない。(証明終)

3. 4. 4 アルゴリズムの正当性に関する考察

3. 4. 2 節で述べたアルゴリズムの正当性に関して以下のことが言える。

[補題3. 1] 目的ノード e への到着時刻の下界を表す(3-31)式の値は(STEP3)~(STEP6)の繰り返しに対し非減少である。

(証明) 式(3-31)により (i_0, k_0) が求めた時点では、 $i \in L$ 、 $E_{ik} \neq \emptyset$ である i, k に対し、 $e_1^{i_0 k_0} + \nu(i_0)$ は最小となる。故に、(STEP5)において次に新しく更新される時間区間 E_{jk} に対し、

$$e_1^{i_0 k_0} + \nu(i_0) \leq e_1^{jk} + \nu(j) \quad (3-40)$$

であることを言えばよい。

式(3-34)の $W_{i_0 j}$ 及び式(3-36)の $c_k^{i_0 j}$ に対し、

$$e_1^{i_0 k_0} + d_{i_0 j} \leq \min W_{i_0 j} \leq c_k^{i_0 j} \quad (3-41)$$

となる。故に、式(3-37)で新しく更新される時間区間 E_{jk} に対し、

$$e_1^{jk} \geq e_1^{i_0 k_0} + d_{i_0 j} \quad (3-42)$$

となるので、

$$\begin{aligned} e_1^{jk} + \nu(j) &\geq e_1^{i_0 k_0} + d_{i_0 j} + \nu(j) \\ &\geq e_1^{i_0 k_0} + \nu(i_0) \end{aligned} \quad (3-43)$$

但し、最後の不等式においては、タイムウィンドウを考慮しない場合の i_0 と e とを結ぶ最短経路上にノード j が存在する時、等式が成立する。(証明終)

(STEP2)の式(3-29)及び(STEP5)の式(3-36), (3-37)から、存在可能時間区間は、必ずある通過可能時間区間に含まれる形で形成され、しかも、含まれる存在可能時間区間の数は高々1つである。従って、このアルゴリズムでは、ノードにおける存在可能時間区間を示す添字と通過可能時間区間を示す添字とは同じものを用いている。

[補題3. 2] 最適解導出までにある通過可能時間区間が式(3-31)によって選ばれる回数は、高々1回である。

(証明) (STEP6)から明らかのように、 μ_j は、ノードjにおいて(STEP4)の式(3-31)で選ばれた直前の k_0 値を示している。但し、一度も選ばれたことのないノードについては0に設定されている。従って、 i_0 の隣接ノードjに対して、式(3-35)の $\mu_j < k$ でよいことを証明できれば、式(3-31)で (j, μ_j) が選ばれ式(3-39)で $E_{j\mu_j} = 0$ となった後再び $E_{j\mu_j}$ が作成されることはないから、式(3-31)で (j, μ_j) が再び選ばれることはない。

通過可能時間区間の指数kは正の整数であり、 $\mu_j = 0$ の場合 $\mu_j < k$ となるのは当然である。従って、 $\mu_j > 0$ とする。(STEP5)での存在可能時間帯の更新において、式(3-35)の $\mu_j < k$ を考慮しないアルゴリズムを考える。この場合も[補題3. 1]は証明も含めそのまま成立する。 $\mu_j > 0$ であるから、今、式(3-31)で (i_0, k_0) が選ばれる以前において (j, μ_j) が選ばれたはずである。その時の存在可能時間区間を $E_{j\mu_j}^* = [e_{1}^{*j\mu_j}, e_{2}^{*j\mu_j}]$ とすると

[補題3. 1]より、

$$e_{1}^{*j\mu_j} + v(j) \leq e_{1}^{i_0k_0} + v(i_0) \leq e_{1}^{i_0k_0} + d_{i_0j} + v(j) \quad (3-44)$$

従って、存在可能時間区間が通過可能時間区間上に作成されること及び式(3-41)から、 $k < \mu_j$ なるkに対して、

$$t_{2}^{jk} < t_{1}^{j\mu_j} \leq e_{1}^{*j\mu_j} \leq e_{1}^{i_0k_0} + d_{i_0j} \leq \min W_{i_0j} \quad (3-45)$$

故に、 $W_{i_0j} \otimes T_{jk} = \emptyset$ となる。 $k = \mu_j$ の時、 $W_{i_0j} \otimes T_{jk} \neq \emptyset$ とすると、式(3-45)の $e_{1}^{*j\mu_j} \leq \min W_{i_0j}$ 及び $\min W_{i_0j} \leq c_k^{i_0j}$ より、 $E_{j\mu_j}^*$ に対し、式(3-37)で新たに更新される E_{jk} は前者が後者を包含する関係にある。

存在可能時間帯としてはより大きな $E_{j\mu_j}^*$ だけを考えればよく、式 (3-31) で一度選ばれたことのある通過可能時間区間については、以後、その更新を考える必要はない。特に、新しく更新される E_{jk} が以前作成された $E_{j\mu_j}^*$ から移行されてきたものである場合は、[定理3.1]におけるループが作られたことを意味しており、上で述べたことは、このようなループのある経路を考える必要のないことを言っているのと同じである。以上から、式(3-35)において $\mu_j < k$ とすればよい。従って、ある通過可能時間区間を式(3-31)で選ぶのは、高々1回で十分である。(証明終)

(STEP5)の式(3-35)で $\mu_j < k$ とできる正当性が[補題3.2]で同時に証明された。

[補題3.1], [補題3.2] から、本アルゴリズムによる解の導出について次のことが言える。

[定理3.2] このアルゴリズムにより、台車運行型SPPTWを次式のオーダをもつ多項式時間で解くことができる。

$$O(|A| \cdot m_0 + |N| \cdot |A| \cdot n_0 + |N|^2 \cdot n_0 \cdot (n_0 + m_0)) \quad (3-46)$$

但し、 n_0 , m_0 は1つのノード、1つのアークのタイムウィンドウに含まれる最大時間区間数をそれぞれ表す。

(証明) [補題3.2]によつて、あるノードのある通過可能時間区間上に作られた存在可能時間区間は高々1回しか選ばれないので、このアルゴリズムにおける(STEP3)~(STEP6)の繰り返し回数は高々 $D = \sum_{i \in N} n_i$ である。

3.4.1節での説明で明らかのように、存在可能時間帯がアークを通過して隣接ノードへ移行できるということは、すなわち、台車そのアーク及び隣接ノードのタイムウィンドウの制約を満足して走行できることに他ならない。従つて、もし最短時間経路が存在すれば、存在可能時間帯が目的ノード e に作成され、高々 D 回の繰り返しのうちに、式(3-31)で $i_0 = e$ となる。一度 $i_0 = e$ となれば、 $v(e) = 0$ であること及び[補題3.1]から、この時点で最早到着時刻が見つかったことになる。もし最短時間経路が存在しなければ、 e に存在可能時間帯は作成できず、高々 D 回の繰り返しの内に(STEP3)で

$L = \emptyset$ となる。

このアルゴリズムにおいて使用される基本算法は比較と加減法のみであり、その計算回数によって計算量を推定できる。 n_o, m_o をそれぞれ1つのノード、アークのタイムウィンドウに含まれる最大時間区間数とすると、(STEP1)における計算量は $O(|A| \cdot (m_o + n_o))$ である。また、(STEP1)でダイクストラ法^{100) - 103)}を用いてあらかじめ $\gamma(j) (j \in N)$ を計算しておくものとする、その計算量は $O(|N|^2)$ で与えられる。(STEP2), (STEP3), (STEP4)及び(STEP5)における計算量は、それぞれ $O(n_o), O(|A| + |N|), O(|N| + n_o), O(|N| \cdot (n_o + m_o))$ となる。(STEP6)の計算量は0である。上述したように、(STEP3)~(STEP6)は高々 $D (\leq |N| \cdot n_o)$ 回繰り返されるので、全体の計算量は $O(|A| \cdot m_o + |N| \cdot |A| \cdot n_o + |N|^2 \cdot n_o \cdot (n_o + m_o))$ となる。

(証明終)

このアルゴリズム中、式(3-28), (3-33)によるアークの除去、(STEP3)における集合 U の作成、式(3-32)による実行可能性の判定の各手続きは、最適解を正しく導出するというアルゴリズムの正当性にとっては必ずしも必要ではない。それらは、(STEP3)~(STEP6)が繰り返される毎に計算にかかわるネットワークの規模を小さくし、アルゴリズムの実行効率を高める。また、最適解の経路は、目的ノード e への最早到着時刻が(STEP4)での終了によって求められた後、ラベル $\gamma(\cdot, \cdot)$ を用いて見つけることができる。

3. 5 下界評価法と解法の効率化

前節までは、複数台車のうちの一台の経路決定に限定した場合の経路決定法を、グラフ論的立場から提案した。それは、問題をタイムウィンドウをもつネットワーク上での最短時間経路問題(台車運行型SPPTW)としてモデル化し、分枝限定法を用いて解くアルゴリズムである。一般の分枝限定法においては、最適化したい目的関数値の下界の評価法をうまく工夫することが、計算効率にとって重要なことであると言われている^{104) 105)}。このことは本問題に対しても言えることである。ここでは、比較的計算効率の良い下界評価法を提案することによって、3. 4. 2節の解法アルゴリズムの改善を図る。ここで提案する方法は、本来のタイムウィンドウ制約の一部を緩和した問題を解くことによって、下界を求めようとするものである。

3. 4. 2節で述べた解法アルゴリズムでは、分枝限定法を用いている。(STEP4)における(3-31)式によってノード*e*における最早到着時刻の下界を評価し、最小の下界をもつ存在可能時間帯を選択する。(STEP5)が分枝の手続きに相当し、(STEP4)で選択された存在可能時間帯を隣接ノードへと移行させ、隣接ノードにおいて存在可能時間帯を新しく作成または修正する。

(STEP4)における存在可能時間帯の区間 E_{ik} の下界は、その存在可能時間帯から台車が出発すると仮定した場合の目的ノード*e*における最早到着時刻の下界になっていれば、基本的にはどんなものでもかまわない。以下では、式(3-31)による下界を含め、存在可能時間区間 E_{ik} に対する三つの下界評価法 $LB_m(E_{ik})$ ($m=1,2,3$)を考える。3. 4節の解法アルゴリズムでは、これら三つの下界に対して(3-31)式のみを次のように修正すればよい。

$$LB_m(E_{i_0k_0}) = \min_{i \in L, E_{ik} \neq \emptyset} \{LB_m(E_{ik})\} \quad (3-47)$$

3. 5. 1 下界評価第1法, 第2法

ノード*i*の存在可能時間帯 $E(i)$ の時間区間 E_{ik} に関する最も単純な下界評価法 $LB_1(E_{ik})$ は、

$$LB_1(E_{ik}) = e_1^{ik} \quad (3-48)$$

である。すなわち、 E_{ik} 中に存在する台車は早くとも e_1^{ik} にしか出発できず、また、それ以後非負の走行時間の後に目的ノードに到着するから、(3-48)式は明かに目的ノードへの最早到着時刻の一つの下界である。(3-48)式による下界評価法を、下界評価第1法とよぶ。

E_{ik} を出発した後の目的ノード*e*までの走行時間は、タイムウィンドウ制約が全くないと仮定した場合の走行時間 $v(i)$ 以上になるはずであることを考慮すると、次式による下界も一つの下界となることは明かである。

$$LB_2(E_{ik}) = e_1^{ik} + v(i) \quad (3-49)$$

これは(3-31)式で使用されている下界であり、この評価法を下界評価第2法とよぶ。

最適解を少ない計算時間で導出するうえでの下界評価法の有効性は、次の点に依存する。その第1は、目的ノードへの実際の最早到着時刻を、この下

界によってどれだけ誤差なく評価できるかという点である。その第2は、下界評価そのものに要する計算時間をどれだけ少なくできるかという点である。以上の観点から考察すると、下界評価第1法、第2法は、計算効率に関して次のような特徴をもつといえる。

(1) 第1法は、存在可能時間帯に含まれる最も早い時刻を下界としているので、計算時間は事実上ないに等しい。しかし、存在可能時間帯を出発した後の目的ノードまでの台車の走行時間を全く考慮していないために、下界の誤差は大きい。

(2) 第2法では、任意のノード i からノード e までの古典的な最短走行時間 $\psi(i)$ をあらかじめ計算しておく必要がある。 $\psi(i) (i \in N)$ の導出においてダイクストラ法¹⁰⁰⁾を用いれば、その計算量はオーダー $O(|N|^2)$ となる。もし、問題の最適経路がタイムウィンドウを考慮しない場合の最短経路に近似している場合には、下界の誤差も小さくなるであろうと予想される。その反面、この最適経路の近似があまりよくない場合には、下界の誤差も大きくなると思われる。特に、問題に実行可能解の存在しない場合には、任意のノード i の存在可能時間帯から目的ノード e までの走行時間は事実上無限大であるのに対し、ノード i から目的ノード e へ連結性が保たれている限り $\psi(i)$ は有限値をとり、下界の誤差は極めて大きいといえる。すなわち、実行可能解の存在しない問題に対しては、計算効率がよくないであろうことが予想される。

通常、より誤差の少ない下界評価法にしようとする、より大きな計算時間を要する。従って、誤差と計算時間との間で均衡をうまくとり、全体として計算効率の良い下界評価法を見つけることが肝要である。以下で提案する下界評価法は、第2法の欠点を補間するように考案された評価法である。

3. 5. 2 緩和問題と下界評価第3法

3. 2. 3節で述べた台車運行型SPPTWにおけるタイムウィンドウ制約は、次のとおりである。

(1) あるノードにおける台車の出発・到着時刻は、そのノードにおけるタイムウィンドウ内になければならない。

(2) タイムウィンドウ中のある通過可能時間区間に到着した台車は、同じ時間区間内に出発しなければならない。

いま、ノード i の存在可能時間帯の区間 E_{ik} から出発する台車が、上のタ

タイムウインドウ制約のうち (1) のみを満たすように走行して、目的ノード e に到達するものとする。この場合の目的ノードにおける最早到着時刻は、制約 (2) を考慮しない分だけ原問題での最早到着時刻より大きくなることはなく、これが下界を与えることは明かである。また、タイムウインドウ制約を全く考慮しない(3-49)式による下界よりは誤差が少ないことも容易にわかる。制約 (2) が緩和されているとはいえ、(1) の制約によって、実行可能解の集合は制約をうける。特に、ノード i からノード e への連結性が保たれているにもかかわらず、区間 E_{ik} からノード e への実行可能な経路がこの緩和問題において見い出せない場合もある。その場合には、原問題においても区間 E_{ik} から e への走行は実行不可能であると判断でき、解の探索において存在可能時間区間 E_{ik} を考慮する必要は全くない。このことが、前節の (2) で述べた下界評価第 2 法の欠点を補間する性質となっている。

この緩和問題の解を用いた下界評価法を下界評価第 3 法とよぶ。すなわち、ノード i を時刻 t に出発し、タイムウインドウ制約のうち (1) のみを満足しながら目的ノード e に到達するとした場合の最早到着時刻を $\kappa(i, t)$ とし、存在可能時間帯 E_{ik} の下界を次により定義する。

$$LB_3(E_{ik}) = \kappa(i, e_{1}^{ik}) \quad (3-50)$$

以下では、タイムウインドウ制約 (2) を緩和した緩和問題¹⁰⁴⁾¹⁰⁵⁾ を解く解法アルゴリズムについて議論する。ここでの緩和問題は、次のとおりである。あるノード i を時刻 t に出発し、タイムウインドウ制約 (1) のみを満足しながら目的ノード e に到達する台車の最早到着時刻 $\kappa(i, t)$ を求める。

いま、この緩和問題におけるノード j ($\neq e$) への最早到着時刻を t_j とする。このとき、台車がノード j から隣接ノード k へ向かう場合、時刻 t_j 以降タイムウインドウ $T(j)$ 内の任意の時刻に出発可能であるが、それはまたアーク (j, k) のタイムウインドウ制約を満たし、かつ、ノード k でのタイムウインドウ制約内に到着しなければならない。従って、(3-22)式を用いると、ノード j から出発可能な時間帯は、 $[t_j, \infty] \otimes S'(j, k)$ となる。結局、台車がノード j から k に移動すると仮定した場合、ノード k への最早到着時刻 t_k は、次式で与えられる。

$$t_k = \min \{ [t_j, \infty] \otimes S'(j, k) \} + d_{jk} \quad (3-51)$$

ノード i を時刻 t に出発するから、 $t_i = t$ は明かである。この初期の最早

到着時刻からはじめて、(3-51)式を適用しながら、次々に他のノードにおける最早到着時刻を求めていき、最終的に t_e を見つければよい。この手順は、古典的な最短経路問題におけるダイクストラ法¹⁰⁰⁾¹⁰²⁾をはじめとする逐次的な解法手順と全く同様に実行できる。以下では、ラベリング法¹⁰⁶⁾を用いた $\kappa(i, t)$ の導出アルゴリズムを提案する。

(RX1) ノード i にテンポラリラベルをつける。

$$l_i = t \quad (3-52)$$

他のノードに対しては、次により、テンポラリラベルのないことを示す。

$$l_j = \infty, \quad j \neq i \quad (3-53)$$

(RX2) テンポラリラベルのついたノードの集合を B とする。

$$B = \{j; l_j < \infty\} \quad (3-54)$$

もし、 $B = \emptyset$ ならば終了。実行可能解は存在しない。

$B \neq \emptyset$ であれば、

$$l_k = \min \{l_j; j \in B\} \quad (3-55)$$

とし、 l_k をパーマネントラベルとする。

もし、 $k = e$ ならば終了。ノード e への最早到着時刻 $\kappa(i, t)$ は l_k となる。

(RX3) ノード k の隣接ノードのうち、パーマネントラベルのついていない任意のノード h に対し、(3-51)式を用いて l_h を修正する。

$$l_h = \min \{l_h, \min [[l_k, \infty] \otimes S'(k, h)] + d_{kh}\} \quad (3-56)$$

(RX4) (RX2)に行く。

3. 6 数値例

ここでは、これまで述べてきた台車運行型SPPTWの解法を用いた簡単な数値例を述べる。また、3. 5節で提案した下界評価法を比較し、それらの計算効率に対する特性について検討する。

3.6.1 搬送系モデル

図3.2で示される搬送ネットワークを考える。このネットワーク上には40個のノードがあり、ノードの横にはノード番号が表示されている。ネットワークの48個のアーキは双方向走行可能であり、いずれのアーキ上での走行時間も3.0単位時間であるとする。また、干渉を起こさない最小の台車間の時間距離（最小干渉許容時間）を1.0時間単位とする。以下で実施した数値例では、いずれも計算機としてIBM3083 (VM/CMS) を使用した。

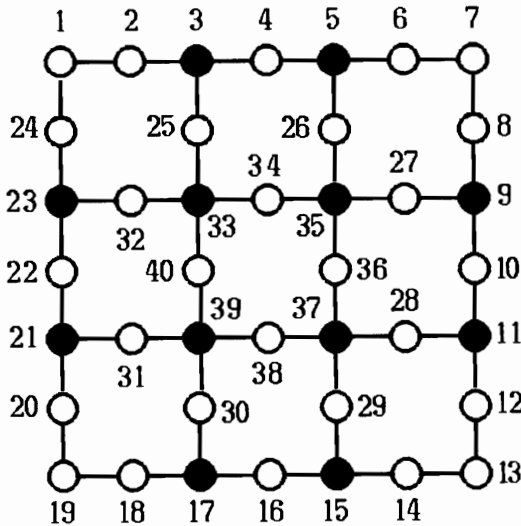


図3.2 搬送ネットワーク

3.6.2 解法アルゴリズムによる解法例

図3.2の搬送系を走行する6台の台車を考える。6台の台車 ($Q_1 \sim Q_6$) の現在位置と将来の予定経路は、表3.1のノード列で示されている。台車 Q_6 はノード25で待機中である。また、台車 Q_2 のみはアーキ(37, 36)上をノード36に向かって走行中である。台車 Q_2 が時刻0においてノード37から時間2.0の地点を走行中であること及び台車 Q_5 がノード28において時間1.0だけ待ち停止することをノード番号37, 28の横の括弧の中に記した-2.0, 1.0で示している。

表 3 . 1 台車の予定経路

台車	経路
Q 1	31-39-38-37-36-35-34
Q 2	37(-2.0)-36-35-34-33-32
Q 3	36-35-26
Q 4	34-33-40-39-30
Q 5	10-11-28(1.0)-37-38-39-31
Q 6	25 (待機中)

以上のような状況の下で、台車 Q_6 に対しノード29への搬送要求が生じたとする。この時、時刻0以降ノード25を出発し、ノード29に最早時刻に到着する最短時間経路を求める。

本問題は、 $N=(1, \dots, 40)$, $|A|=96$, $d_{ij}=3.0(i, j \in N)$, $t_0=0.0$, $s=25$, $e=29$, $Q_0=Q_6$, $Q=\{Q_i; i=1, \dots, 6\}$, $\epsilon=1.0$ とした場合の台車運行型SPPTWである。ノード、アークでの通過可能時間帯、進入可能時間帯は、最小干渉許容時間 ϵ 及び表3.1に示した $Q'=\{Q_i; i=1, \dots, 5\}$ の走行経路を用いて、3.2.2節の手順により計算した。例えば、出発ノードにおける通過可能時間帯は $T(s)=\{(-\infty, \infty)\}$, ノード39でのそれは $T(39)=\{(-\infty, 2], [4, 8], [10, 15], [17, \infty)\}$ となり、ノードでの通過可能時間区間の平均個数は1.5となった。有向アーク当りの進入可能時間区間の平均個数は1.2となったが、さらに、解法アルゴリズム(STEP1)を適用して得られる修正進入可能時間帯のそれは1.6となった。

この問題を3.4.2節の解法アルゴリズムにより解いた時、次のような結果を得た。

(1) 台車 Q_6 の最短時間経路は、25(1.0)-33-40(7.0)-39-38-37-29であり、ノード29への最早到着時刻は26.0となった。但し、()内はノードでの待ち停止時間を示す。この解は、台車 Q_6 に対し、台車 Q_4 がノード33に進入した直後にこのノードに入るようにノード25で待ち停止させ、また、 Q_5 が経路37-38-39を通過するのをノード40で待ち合わせさせた後、この経路を逆に走行させるような解となっている。

(2) 最適解導出までのCPU時間は55ミリ秒であった。また、解法アルゴリズムにおける(STEP3)~(STEP6)の総繰り返し回数は14回であった。この問題では、[定理3.2]の(証明)の中で述べたDの値は60となり、最悪の場合、問題を解くまでに60回の繰り返しが必要である。一方、上の(1)項で述べたQ₀の最短時間経路のノード数は7であり、この解を導出するまでの最小繰り返し回数は7回である。

3.6.3 下界評価法の比較例

図3.2の内、黒丸で表わされた12個のノード(ノード番号:3,5,9,11,15,17,21,23,33,35,37,39)の前には自動工作機械が設置されており、これらのノード間のみを無人搬送車が工作物や治工具等を搬送要求に従って搬送する場合を考える。この搬送系に投入する台車数は、2台から10台まで変化させた。

シミュレーションは、各台車数毎に次の手順により実施した。

(i) 投入された全台車を、12個の黒丸のうち異なるノードにランダムに待機させる。

(ii) 全台車のうち、搬送が最も早く終了した台車(要求待台車)に対して、黒丸のうちからランダムに一つの目的ノードを選択し、そのノードへの搬送要求を与える。もし、搬送の最も早く終了する台車が複数台あれば、台車番号の小さな台車を要求待台車とする。

(iii) 要求待台車以外の台車の走行経路と最小干渉許容時間からタイムウィンドウを計算し、台車運行型SPPTWを作成する。この問題を一つの下界評価法LB_m(\cdot)を用いたアルゴリズムによって解き、要求待台車の最短時間経路を決定する。

(iv) 上の手順(ii),(iii)を300回繰り返す。

(v) 手順(ii)~(iv)をm=1,2,3について実施する。その際、目的ノード番号を与える乱数は、各mに対して全く同じものを発生させるものとする。

以上のようなシミュレーションは、実際の工場においては、搬送要求が休みなく台車に与えられ、各台車の稼働率が100%となる場合に相当する。また、工作機械の設置場所を搬送路の交差点としたのは、やや現実的ではない。しかし、これにより台車が交差点上で停止するケースが多くなり、他の台車の走行の障害となることで実行可能解のない台車運行型SPPTWを多く

作成できる。このことが、三つの下界評価法の特徴を明確にするのに役立つ。

以上のようなシミュレーションにより、次のような結果が得られた。

3. 5. 1 節でも述べたように、分枝限定法を用いたアルゴリズムの計算効率に対する下界評価法の影響は、下界の誤差と下界計算に要する時間の二つの要素から判断できる。三つの下界評価法の有効性の比較は、解導出までに要した計算時間から判断できるが、下界の誤差に関する考察を行うため、解導出までの分枝回数についてのデータも取得した。

(1) 実行可能解の存在しない問題数の割合

ここでは、厳密な意味で実行可能解の存在しない問題を実行不能問題とよぶ。表3. 2は、各下界評価法毎、実行不能となった問題数の全問題数に対する割合（実行不能率）を示したものである。

表3. 2 実行不能率 (%)

台車数	2~4	5	6	7	8	9	10
第1法	0	11	26	48	71	77	91
第2法	0	11	26	47	71	77	91
第3法	0	11	28	47	71	77	91

前節のシミュレーション手順で述べたように、要求待台車に対して発生させた目的ノードは、各下界評価法では全く同じものとした。同じ問題に対しては、どの下界評価法を使用しても目的ノードへの最早到着時刻は同じになるものの、それを実現する最適経路としては異なる経路が得られる可能性がある。従って、その後発生する目的ノードが同一であっても、異なったタイムウィンドウが作成され、台車型SPPTWとして異なるものが得られることがある。発生した目的ノード列が同一であるにもかかわらず、表3. 2において、各下界評価法に対し異なった実行不能率が得られたのはこのことによる。但し、結果の違いはほとんどなく、どの評価法においても、解の存在しない問題は台車数5台から発生し始め、10台になると実行不能率は90%を越える結果となった。

(2) 繰り返し数

分枝限定法を用いた3.4.2節の解法アルゴリズムにおいて、(STEP3)～(STEP6)の繰り返し回数は、問題を解くまでの分枝回数でもある。下界の誤差の程度は、分枝回数によってある程度判断できる。すなわち、通常、誤差の小さい下界評価法は分枝回数は少なく、誤差の大きいものは分枝回数も多い。図3.3は、台車数に対する問題一題当りの平均繰り返し回数を示したものである。図3.3(a)は、実行可能な問題一題当りの結果を示したものであり、図3.3(b)は、実行不能問題一題当りの結果である。また、図3.3(c)は、(a),(b)を総合して、実行不能であるか否かを問わず問題一題当りの平均繰り返し回数を示している。

これらの結果から、アルゴリズムの繰り返し回数に関して、次のことがいえる。

(ア) 下界評価第1法は、ほとんどの場合、他の評価法よりも繰り返し回数が多い。

(イ) 第2法は、実行可能な問題に対しては第3法と同程度の繰り返し回数であるが、逆に、実行不能問題に対する回数は第1法と同程度に多い。従って、図3.3(c)からわかるとおり、台車数が少なく容易に最適解の求められそうな状況に対しては、第3法と同様に効率の良い方法であるが、台車数が多くなり台車間で複雑な干渉の予想される状況に対しては、その効率は第3法に比べて良くない。この結果は、第3.5.1節で述べた第2法の欠点に関する考察を裏づけするものとなっている。

(ウ) 第3法は、繰り返し回数に関して、三つの評価法のなかで最も効率の良い方法である。

(3) 計算時間

図3.4は、問題一題当りの平均計算時間を示したものである。図3と同様に、図3.4(a)は実行可能な問題のみに対する結果、図3.4(b)は実行不能問題のみに対する結果、図3.4(c)は全ての問題に対する結果である。これらの結果から、次のことがいえる。

(ア) 実行可能な問題においては、第2法が最も有効な下界評価法である。これは、繰り返し回数に関する考察から判断されるように下界誤差もそれほど大きくはなく、また、下界の計算時間もそれほど多くは要しないことによるものと思われる。この有効性は台車数が少ない場合に大きい。しかし、台車数が多くなると、この有効性も他の評価法とほぼ同程度となる。これは、

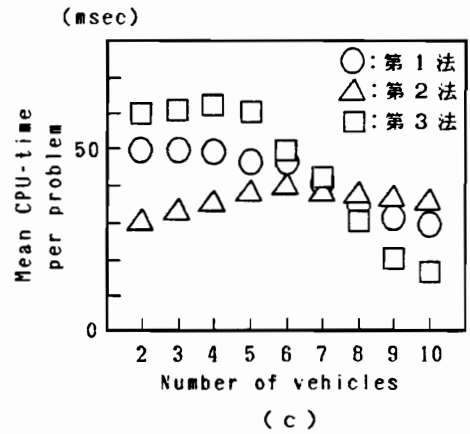
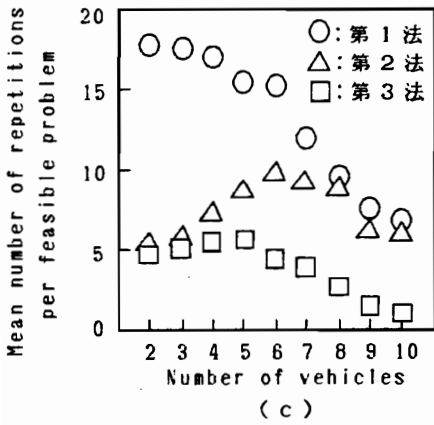
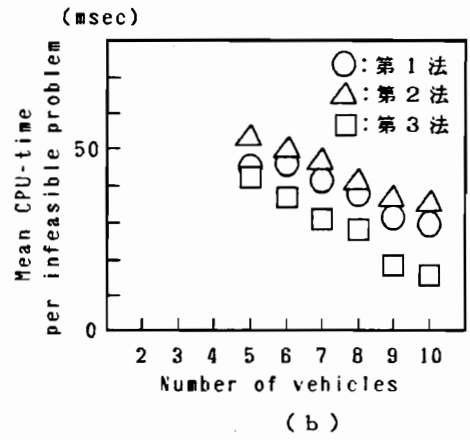
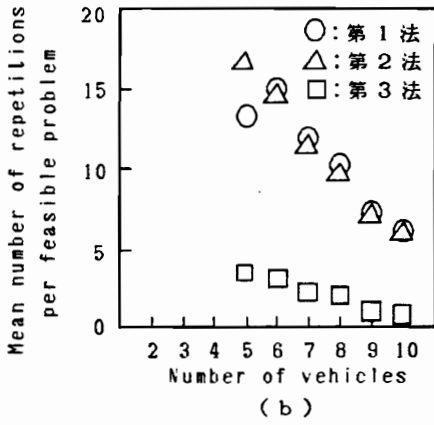
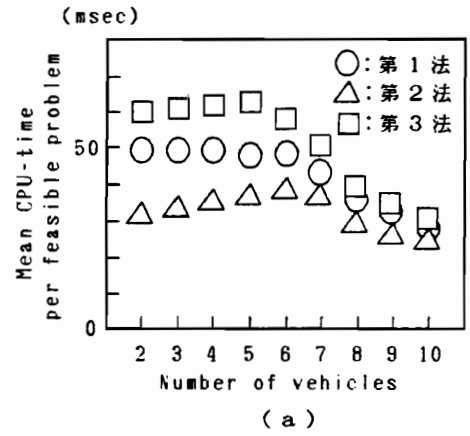
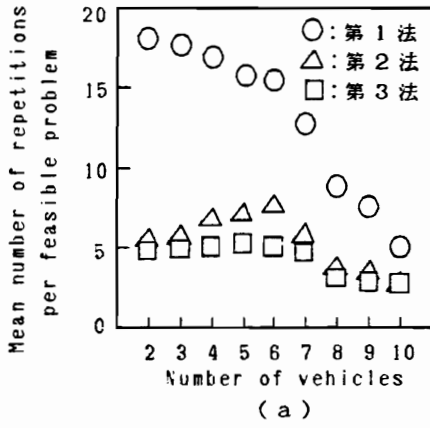


図3.3 平均繰り返し回数

図3.4 平均計算時間

台車数が少なく最適経路がほぼ古典的な最短経路に近いと予想される状況においては、第2法の下界誤差は小さいが、台車数が多くなり、台車間の複雑な干渉のためにその近似があまり成立しない状況においては、下界誤差も大きくなるためであると思われる。

(イ) 実行不能問題においては、第3法が最も有効である。

(ウ) 実行不能であるか否かを区別しない場合、上述した(ア)(イ)の特徴と台車数が多くなるにつれ実行不能問題も多くなるという傾向とにより、台車数が少ない場合は第2法が有効であり、台車数が多い場合は第3法が有効であるといえる。

3. 7 結言

この章では、工場における複数無人搬送車のうちの一台に対する運行経路決定問題を取り扱った。3. 2節及び3. 3節では、この問題を搬送ネットワークのノード、アークに到着・出発時刻制約のタイムウィンドウが存在する場合の最短経路問題(台車運行型SPPTW)に定式化した。3. 4節においては、存在可能時間帯という概念を導入し、この問題に対し、分枝限定法を用いた多項式計算時間の解法アルゴリズムを提案した。この解法の計算効率の良しあしは、一般の分枝限定法でもいえるように、最適化したい目的関数値の下界評価に左右される。3. 5節では、三つの下界評価法(第1~3法)を提案し、これらの有効性をシミュレーションにより検討した結果、次のような特徴のあることがわかった。

(1) 第2法は、他の下界評価法に比べ、実行可能な経路が存在し、かつ、複雑な干渉形態の予想されない問題に対して有効な下界評価法である。

(2) 実行可能な経路の存在しない問題に対しては、その実行不能性をいち早く判断できる下界評価法として第3法が有効である。

第4章 台車運行型SPPTWの拡張

4.1 緒言

この章では、3章で提案した台車運行型SPPTWの解法アルゴリズムを他の問題に拡張することを考える。まず、台車運行型SPPTWを台車の搬送時間の非負性を仮定しない場合の問題に一般化することにより、古典的な最短経路問題と台車運行型SPPTWが等価な構造をもつことを明かにする。また、台車の現実的な運行問題として、走行途中において走行スケジュールに遅滞を生じた場合の再スケジューリング手順について考察する。最後に、搬送要求が複数の目的地を含む場合や、目的地において作業時間をもつような場合への運行経路決定法について考察する。

4.2 走行時間の非負性を仮定しない場合の一般解法

ここでは、台車運行型SPPTWを走行時間が必ずしも非負ではないアークをもつ一般化されたネットワークにおける問題に拡張し、問題の一般的な構造について議論する。そして、この問題が基本的にはタイムウィンドウをもたないネットワークにおける最短経路問題¹⁰²⁾¹⁰⁷⁾と同じ構造をもつことを明かにする。さらに、このことを用いて、二段階の手順から成る一般解法を提案する。まず、対象とするネットワークをタイムウィンドウに注目して再構成し直した後、原問題をその再構成したネットワークでの最短時間経路問題の部分問題に分割する。次に、この部分問題を最適性の原理¹⁰⁸⁾⁻¹¹¹⁾を用いてBellman-Ford法と同型の漸化式によって解く。部分問題の解法としては他に二つの方法を提案し、数値例を用いて各方法の実行効率を比較検討する。

4.2.1 ネットワークの再構成

3.3.1節で述べたタイムウィンドウをもつネットワークを考える。次のような手続きによりこのネットワークを再構成する。

1) ノード $i \in N$ に対して、その通過可能時間区間の数だけの新しいノード i_k ($k=1, \dots, n_i$) を作成し、各々の i_k に1つの通過可能時間区間 T_{i_k} を与える。

2) 有向アーク $(i, j) \in A$ に対して、

$$T_{ik} \otimes S(i, j) \otimes (T_{jl} \ominus d_{ij}) \neq \emptyset \quad (4-1)$$

ならば、ノード i_k から j_l への有向アークを作成し、そのアークの入り口の通過可能時間帯として $S(i, j)$ を与える。

この手続きによって変換したネットワークを以下修正ネットワークと呼ぶ。台車運行型 SPPTW に関して、原ネットワークと修正ネットワークが等価であることを以下で述べる。

原ネットワークにおいて台車のノード i の通過可能時間区間 T_{ik} からノード j の通過可能時間区間 T_{jl} への移動は、修正ネットワークにおいてはノード i_k からノード j_l への移動に対応する。また、台車がノード i からノード j へ移動する場合、まずノード i のタイムウィンドウ $T(i)$ 及びアーク (i, j) の入口におけるタイムウィンドウ $S(i, j)$ の制約がある。さらに、ノード j へそのタイムウィンドウ $T(j)$ 中に到着するためには、ノード i を時間帯 $T(j) \ominus d_{ij}$ 中に出発しなければならない。従って、手順 2) の (4-1) 式は、修正ネットワークにおけるノード i_k から j_l への台車の移動の可能性を考慮した式である。また、原ネットワークの同一ノード i から作成された二つのノード間においては、ノード i_k からノード i_l ($k \neq l$) への移動は、原ネットワークにおいて台車が通過可能時間区間 T_{ik} に到着し、別の時間区間 T_{il} から出発することに対応し、3. 2. 3 節で述べたタイムウィンドウの性質 (2') により許可されない。従って、修正ネットワーク上では、これらのノード間にアークを作成する必要はない。

この修正ネットワークにおけるアーク数に関しては、次の定理が成り立つ。

[定理 4. 1] 修正ネットワークへの変換の手続き 2) によって、ノード i_k ($k=1, \dots, n_i$) とノード j_l ($l=1, \dots, n_j$) の間において作成されるアーク数は高々 $n_i + n_j - 1$ である。従って、修正ネットワークで新たに作成される有向アークの総数は高々次の式で与えられる。

$$\sum_{(i, j) \in A} (n_i + n_j - 1) \quad (4-2)$$

(証明) 式(4-1)が成り立てば、

$$T_{ik} \otimes (T_{jl} \ominus d_{ij}) \neq \emptyset \quad (4-3)$$

である。従って、 $k=1, \dots, n_i$ と $l=1, \dots, n_j$ の組合せに対して、上式が高々 $n_i + n_j - 1$ 組成立することを証明すればよい。 $T(i) = \{T_{ik}; k=1, \dots, n_i\}$ 及び $T(j) \ominus d_{ij} = \{T_{jl} \ominus d_{ij}\}$ は数直線上に並ぶ n_i 個と n_j 個の閉区間の集合であり、上式の成立する組数はその二つの集合の積集合がもつ閉区間の数と同じである。従って、数直線上に互いに重複なく並ぶ m 個の閉区間集合と同様な n 個の閉区間集合の積集合がもつ閉区間数を $K[m, n]$ で表わすと、

$$K[m, n] \leq m + n - 1 \quad (4-4)$$

であることをいえばよい。

$m=1, n=1$ の場合、 A と B の積集合のもつ最大区間数が 1 となることは明かである。故に、このとき式(4-4)は成立する。

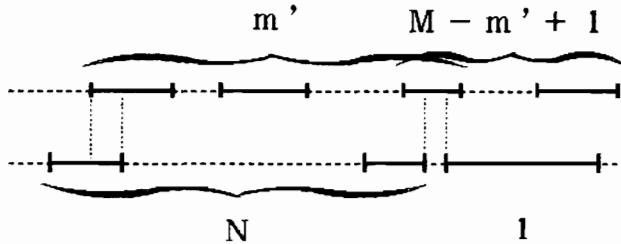


図 4 . 1 区間の重なり

$m \leq M, n \leq N$ に対し式(4-4)が成立していると仮定する。 $K[M, N+1]$ は、図4.1のような場合に最大となり、

$$\begin{aligned} K[M, N+1] &\leq \max_{1 \leq m' \leq M} (K[m', N] + K[M-m'+1, 1]) \\ &\leq \max_{1 \leq m' \leq M} ((m'+N-1) + (M-m'+1)) = M + (N+1) - 1 \end{aligned} \quad (4-5)$$

となる。同様に、 $K[M+1, N] \leq (M+1) + N - 1$ となる。従って、任意の自然数 m, n の組に対し式(4-4)が成立する。(証明終)

4. 2. 2 問題の変換と部分問題の定式化

前節で述べたことから、原ネットワークにおける台車運行型 SPPTW は、修正ネットワークにおける次のような問題に置き換えることができる。

$t_0 \in T_{s_p}$ ($1 \leq p \leq n_s$) となる一つの p に対して、ノード s_p を時刻 t_0 以降に出発し、ノード、アークをそのタイムウィンドウ制約内に通過しながら、ノード e_k ($k=1, \dots, n_e$) のいずれかに最早時刻に到着する経路を求める。この問題は、目的ノードを個別に設定することより、 n_e 個の部分問題（ノードにおけるタイムウィンドウが一つの通過時間区間から成るネットワーク上で、一つの出発ノード s と一つの目的ノード e_k をもつ台車運行型 SPPTW）に分割することができる。この部分問題を、以後部分台車運行型 SPPTW とよぶ。原ネットワークにおけるノード e の通過可能時間区間は(3-17)式の定義による大小関係 $T_{e_1} \leq T_{e_2} \leq \dots \leq T_{e_{n_e}}$ をもつことから、必ずしもこれらの部分問題をすべて解く必要はない。すなわち、目的ノード e_1, e_2, \dots をもつ部分問題を順に解いていき、解が求まった時点でその解を原問題の最適解とすればよい。

以下では、ノードにおけるタイムウィンドウが一つの時間区間から成るようなネットワークでの部分台車運行型 SPPTW を取り扱う。修正ネットワーク上でのノード数を n とし、ノード番号をあらためて付与しなおし、出発ノードをノード 1、目的ノードをノード n としても一般性を失わない。また、対象とするネットワークのノードの集合、アークの集合、ノード i 及びアーク (i, j) におけるタイムウィンドウは、原問題と同じ記号 $N, A, T(i), S(i, j)$ を用いるものとする。但し、 $T(i)$ が一つの時間区間から成ることを明示するため、

$$T(i) = [a_i, b_i] \quad (4-6)$$

で表わす。

部分台車運行型 SPPTW は、次のように混合整数計画問題に定式化される。

$$\begin{aligned} & \text{minimize} \\ & f(X, V, U) = \sum_{(i,j) \in A} d_{ij} \cdot x_{ij} + \sum_{k \in N} (v_k - u_k) + t_0 \end{aligned} \quad (4-7)$$

subject to

$$\sum_j x_{ij} - \sum_k x_{ki} = \begin{cases} 1 & (i=1) \\ 0 & (i \neq 1, n) \\ -1 & (i=n) \end{cases} \quad (4-8)$$

$$x_{ij}=1 \Rightarrow \begin{cases} v_i + d_{ij} = u_j & ((i, j) \in A) \\ v_i \in S(i, j) \end{cases} \quad (4-9)$$

$$u_i, v_i \in T(i), u_i \leq v_i \quad (i \in N) \quad (4-11)$$

$$u_s = t_0 \quad (4-12)$$

$$x_{ij} \in \{0, 1\} \quad (4-13)$$

但し、 $X = \{x_{ij}; (i, j) \in A\}$, $U = \{u_i; i \in N\}$, $V = \{v_i; i \in N\}$ とする。上の式中、変数 x_{ij} は台車の経路がアーク (i, j) を通る場合は1を、そうでない場合は0をとる0-1変数である。 u_i 、 v_i は、それぞれ、ノード i における台車の到着時刻、出発時刻を表わす。

式(4-8)は、台車の経路がノード1から始まり、ノード n で終る経路の連続性を保証する式である。式(4-9)、(4-10)は、経路がアーク (i, j) を通るならば、ノード j での到着時刻はノード i からの出発時刻にアーク (i, j) での走行時間 d_{ij} を加えたものに等しく、また、ノード i からの出発時刻はアーク (i, j) のタイムウィンドウ制約を満たさなければならないことを示している。式(4-11)は、ノード i での到着及び出発時刻がノードのタイムウィンドウ制約を満たし、後者は前者より大きいことを示している。式(4-12)は、出発ノード s での到着時刻を t_0 とおくことにより、 s からの出発が時刻 t_0 以降に起こることを保証する。式(4-7)の第1項は経路のアーク上での総走行時間であり、第2項は経路上におけるノードでの総待ち停止時間である。従って、目的関数 $f(X, U, V)$ は経路全体での走行が終了する時刻を示しており、問題はこの値を最小化することである。しかしながら、整数計画問題を解くことは一般に難しく、本研究でも上式をすなおに解くことはせず、グラフ論的アプローチによる解法を考える。

4. 2. 3 最適経路の性質

部分台車運行型SPPTWにおいては、次の定理が成り立つ。

[定理4. 2] 問題に最適経路があるならば、ループを含まない最適経路が存在する。

(証明) 最適経路のノード列をノード j においてループのあるノード列 $\{l_1=1, l_2, \dots, l_k=j, \dots, l_{k'}=j, \dots, l_t=n\}$ とし、経路上の任意のノード l_h における(到着時刻, 出発時刻)を (u_h, v_h) とする。部分台車運行型 SPP TW においては各ノードのタイムウィンドウは一つの時間区間から成るから、式(4-6)の記号及びループの走行時間の非負性により、

$$a_j \leq u_k \leq v_k \leq u_{k'} \leq v_{k'} \leq b_j \quad (4-14)$$

となる。従って、経路のノード列が $\{l_1=1, l_2, \dots, l_k, l_{k'+1}, \dots, l_t=n\}$ であり、各ノードでの(到着時刻, 出発時刻)が $(u_1=t_0, v_1), (u_2, v_2), \dots, (u_k, v_{k'}), (u_{k'+1}, v_{k'+1}), \dots, (u_t, *)$ (目的ノードにおいては到着時刻のみ)である解は、タイムウィンドウ制約を満足し、かつ最適解と同じ時刻に目的ノードに到着する。つまり、最適経路のループ部分を削除した最適経路もまた存在する。 (証明終)

[定理4.3] 問題に最適解が存在するならば、出発ノードからその任意の途中ノードまでの部分経路が最適となる最適解もまた存在する。

(証明) 最適解がノード列 $\{l_1=1, l_2, \dots, l_k=j, \dots, l_t=n\}$ 及び任意のノード l_h における(到着時刻, 出発時刻) (u_h, v_h) から成るものとする。この経路の前半の任意の部分経路 $\{l_1=1, \dots, l_k\}$ に関して、ノード l_1 を時刻 t_0 以降出発し、ノード l_k へ最早時刻に到着する最適解を、ノード列 $\{l'_1=1, l'_2, \dots, l'_p=l_k\}$ 及び $\{(到着時刻, 出発時刻)\} = \{(u'_1=t_0, v'_1), (u'_2, v'_2), \dots, (u'_p, *)\}$ とする。このとき $u'_p \leq u_k$ であるから、ノード列 $\{l'_1=1, l'_2, \dots, l'_p, l_{k+1}, \dots, l_t=n\}$ 及び $\{(到着時刻, 出発時刻)\} = \{(u'_1=t_0, v'_1), (u'_2, v'_2), \dots, (u'_p, v_k), (u_{k+1}, v_{k+1}), \dots, (u_t, *)\}$ である解もまた最適解となる。 (証明終)

定理4.2及び定理4.3によつて、最適経路としてループがなく、その任意の部分経路が最適である経路のみを探索すればよいことが証明された。また、これらの定理は最適経路を求める直接の方法を示している。すなわち、定理4.2は最適経路は高々全ノード数 n 個のノード列から成ることを意味しており、定理4.3はあるノードまでの最適経路を求めるにはそのノードの隣接ノードまでの最適経路を用いて探索すればよいことを意味している。

4. 2. 4 部分問題の解法アルゴリズム

前節で述べた定理によって、動的計画法を用いた部分台車運行型SPPTWの解法アルゴリズムが提案できる。

4. 2. 4. 1 Bellman-Ford型の解法アルゴリズム

定理4. 3によって、ノード*l*を時刻 t_0 以降出発しあるノード*i*に最早時刻に到着する最適解は、 $(k, i) \in A$ となるすべてのノード*k*までの最適解を用いることによって求めることができる。求めるノード*i*への最早到着時刻を u_i とし、 $(k, i) \in A$ なる任意のノード*k*における最早到着時刻及び出発時刻を u_k, v_k とする。4. 2. 2節における(4-10)式及び(4-11)式の制約と(4-9)式から、

$$\begin{aligned} u_i &= \min_{(k,i) \in A} \{ \min_{v_k \in S(k,i), u_k \leq v_k \leq b_k} (v_k + d_{ki} \in T(i); \\ &= \min_{(k,i) \in A} \{ \min_{v_k \in S(k,i), u_k \leq v_k \leq b_k} (v_k + d_{ki}; a_k \leq v_k + d_{ki} \leq b_k, \\ &= \min_{(k,i) \in A} \{ \min_{v_k \in S(k,i), v_k \in [u_k, b_k]} (v_k + d_{ki}; v_k \in T(i) \ominus d_{ki}, \\ &= \min_{(k,i) \in A} \{ \min_{v_k \in ((T(i) \ominus d_{ki}) \otimes S(k,i) \otimes [u_k, b_k])} (v_k + d_{ki}; \\ &= \min_{(k,i) \in A} \{ \min B_{ki} + d_{ki} \} \end{aligned} \quad (4-15)$$

但し、最後の式への変形では、

$$S'(k, i) = (T(i) \ominus d_{ki}) \otimes S(k, i) \quad (4-16)$$

$$B_{ki} = S'(k, i) \otimes [u_k, b_k] \quad (4-17)$$

とおいた。

式(4-15)によってノード*i*への最早到着時刻は算出できるが、それを実現する経路は、 $u_i = (\min B_{ki}) + d_{ki}$ となる適当なノード*k*までの最適経路

にノード i を付加すればよい。

以上のことから、次のような漸化式によって任意のノード $i \in N$ への最早到着時刻を導出できる。

$$i) \quad u_1^{(0)} = t_0, \quad u_i^{(0)} = \infty \quad (i \neq 1) \quad (4-18)$$

ii) すべての $i \in N$ に対し、

$$u_i^{(m+1)} = \min \left\{ u_i^{(m)}, \min_{(k,i) \in A} \left(\min_{ki} B_{ki}^{(m)} + d_{ki} \right) \right\} \quad (4-19)$$

但し、

$$B_{ki}^{(m)} = S'(k, i) \otimes [u_k^{(m)}, b_k] \quad (4-20)$$

$u_i^{(m)}$ は、高々 $(m+1)$ 個のノード数をもつ有向パスによって実現できるノード i への最早到着時刻を表している。上の漸化式は、 $u_i^{(m)}$ ($i \in N$) が変化しなくなったとき終了し、そのとき、目的ノード n への最早到着時刻のみならず、すべてのノードへの最早到着時刻が求められていることになる。また、定理4.2より、 m の値が高々 $(n-1)$ となるまでには必ず終了する。式(4-19)では、比較計算と加法計算のみが必要であり、各ノード及び各アークの通過可能時間区間数の最大数をそれぞれ n_0 、 m_0 とすると、その計算量は係数を省略せずに表すとオーダー $O(2 \cdot (n_0 + m_0) \cdot n^3)$ となる。

問題において、特に、

$$T(i) = \{ [-\infty, \infty] \}, \quad i \in N \quad (4-21)$$

$$S(i, j) = \{ [-\infty, \infty] \}, \quad (i, j) \in A \quad (4-22)$$

とすれば、台車は各ノード、各アークを常時通過可能であることになり、問題はタイムウィンドウ制約を考慮しないネットワークにおける古典的な最短経路問題となる。このとき、漸化式(4-19)は次のようになる。

$$u_i^{(m+1)} = \min \left\{ u_i^{(m)}, \min_{(k,i) \in A} (u_{ki}^{(m)} + d_{ki}) \right\} \quad (4-23)$$

上の解法は古典的 shortest path 問題における Bellman-Ford の解法^{112) 113)} に他ならない。従って、式(4-18)～(4-20)による解法を以後 Bellman-Ford 型の解法

(B-F型解法)とよび、式(4-23)を用いたBellman-Fordの解法(B-F解法)と区別する。

B-F型解法による解は、タイムウィンドウの制約があるため、B-F解法により計算される解より大きな値になる。

[定理4.4] B-F型解法による解 $u^{(m)}_i$ と、同じネットワークにおいてタイムウィンドウがないと仮定した場合のB-F解法による解 $w^{(m)}_i$ に関して次の不等式が成立し、 $w^{(m)}_i$ は $u^{(m)}_i$ の下界値を与える。

$$u^{(m)}_i \geq w^{(m)}_i \quad (4-24)$$

(証明) (3-19)式及び(4-20)式から、

$$\begin{aligned} \min_{ki} B_{ki}^{(m)} &= \min (S'(k,i) \otimes [u_k^{(m)}, b_k]) \\ &\geq \max (\min S'(k,i), u_k^{(m)}) \geq u_k^{(m)} \end{aligned} \quad (4-25)$$

となる。式(4-19)と上式から、

$$u^{(m+1)}_i \geq \min \left\{ u^{(m)}_i, \min_{(k,i) \in A} (u_k^{(m)} + d_{ki}) \right\} \quad (4-26)$$

である。

$m=0$ のときは、 $w^{(0)}_i$ と $u^{(0)}_i$ は初期値であり、式(4-18)により同じ値をとる。従って、式(4-24)が成立する。

いま、ある m のとき式(4-24)が成立すると仮定する。式(4-26)より、

$$\begin{aligned} u^{(m+1)}_i &\geq \min \left\{ u^{(m)}_i, \min_{(k,i) \in A} (u_k^{(m)} + d_{ki}) \right\} \\ &\geq \min \left\{ w^{(m)}_i, \min_{(k,i) \in A} (w_k^{(m)} + d_{ki}) \right\} \\ &= w^{(m+1)}_i \end{aligned} \quad (4-27)$$

となり、 $m+1$ のときにも式(4-24)は成立する。従って、数学的帰納法により式(4-24)は常に成り立つ。(証明終)

4. 2. 4. 2 Yen型の解法アルゴリズム

古典的な最短経路問題に関しては、式(4-23)を用いたB-F解法に対して、使用する記憶容量を少なくし、かつ計算量を減少させる改善案をYen¹¹⁴⁾が提案している。このYenの修正法は、台車運行型SPPTWに関するB-F型解法式(4-18)~(4-20)に対しても直接適用でき、その結果は次のようになる。

$$i) u_1^{(0)} = t_0, u_i^{(0)} = \infty (i \neq 1) \quad (4-28)$$

ii) mが偶数である場合は $i=1, \dots, n$ の順に、mが奇数である場合は $i=n, \dots, 1$ の順に次式を計算する。

$$u_i^{(m+1)} = \begin{cases} \min \{ u_i^{(m)}, \min_{k < i, (k,i) \in A} (\min B_{ki}^{(m+1)} + d_{ki}) \} \\ \quad (m \text{ が偶数のとき}) \quad (4-29) \\ \min \{ u_i^{(m)}, \min_{k > i, (k,i) \in A} (\min B_{ki}^{(m+1)} + d_{ki}) \} \\ \quad (m \text{ が奇数のとき}) \quad (4-30) \end{cases}$$

これらの式を用いた台車運行型SPPTWの解法をYen型解法とよぶこととする。

上の計算は、B-F型解法と同様に $u_i^{(m)}$ が変化しなくなったとき終了し、その時点ですべてのノードへの最早到着時刻が求められていることになる。その計算量はオーダー $O((n_0 + m_0) \cdot n^3 / 2)$ となる。

4. 2. 4. 3 分枝限定型の解法アルゴリズム

B-F型及びYen型の解法アルゴリズムは、式(4-15)を基本算法として各ノードkの $u_k^{(m)}$ を用いてその隣接ノードの最早到着時刻を修正してゆき、最適値に収束させようとするものである。従って、式(4-15)の基本算法を用い、最早到着時刻の修正が正当になされることを保証しながら、より早い計算速度を実現する計算手順を工夫することが上述した解法の他にも可能である。

ここで提案する解法は、基本算法を用いて計算する際、式(4-19), (4-29)のようにあるノードiの最早到着時刻 $u_i^{(m)}$ の更新を、 $(k, i) \in A$ のようにiに対し後方に隣接するすべてのkにおける $u_k^{(m)}$ から計算するのではなく、

あるノード i^* を選択し、その $u^{(m)}_{i^*}$ を用いて、 $(i^*, k) \in A$ のように i^* に
 対し前方に隣接するすべてのノード k の最早到着時刻を基本算法によって更
 新するものである。すなわち、

$$u_k^{(m+1)} = \min \{ u_k^{(m)}, \min_{i^*} B_{i^*k}^{(m)} + d_{i^*k} \} \quad (4-31)$$

とする。 $u_k^{(m)}$ が式(4-31)の $\{ \}$ 中第2項目の値によって新しい値 $u_k^{(m+1)}$
 に更新された場合、“ノード i^* の最早到着時刻がノード k へ移行された”と
 呼ぶことにする。このような移行を式(4-18)を初期値として実行してゆくこ
 とが式(4-19), (4-29)による漸化式の計算とまったく同じであることは明かで
 ある。そして、すべてのノードでの最早到着時刻が更新されなくなったとき
 最適解が得られる。

もし、選んだ $u^{(m)}_{i^*}$ が以後更新されない値、すなわちノード i^* の本来の
 最早到着時刻であれば、 $u^{(m)}_{i^*}$ による上式の計算は再び行う必要はない。そ
 して、そのようなノード i^* として目的ノード n が選択されれば、その時点で
 計算を終了することができる。このような i^* の選択方法を以下で議論する。

実数全体を R 、ノード全体を N とし、ノードと時刻の上で定義される実数
 関数 $f(\cdot, \cdot)$

$$f : N \times R \rightarrow R \quad (4-32)$$

を考える。 f は、次のような性質をもつものとする。

$$f(i, t) \leq f(j, t + d_{ij}), \quad (i, j) \in A \quad (4-33)$$

$$f(i, t) < f(i, t'), \quad t < t' \quad (4-34)$$

この f を用いて、次式を満たす一つのノード i^* を選択する。

$$f(i^*, u_{i^*}^{(m)}) = \min_{j \in N} f(j, u_j^{(m)}) \quad (4-35)$$

このとき、次の定理が成り立つ。

[定理4.5] 式(4-35)によって選択したノード i^* における $u^{(m)}_{i^*}$ は、
 以後、式(4-31)での計算によっては更新されない最早到着時刻である。

(証明) 式(4-35)で選択された $u^{(m)}_{i^*}$ が、選択後において $u^{(m)}_j$ から更新
 されるものとする。このとき、 $u^{(m)}_j$ が隣接ノードへ次々に移行していき、

ノード i^* の最早到着時刻を更新する。この移行が行われるノード j から i^* へのノード列を $L = \{j, l_1, l_2, \dots, l_p, i^*\}$ とする。移行は式(4-31)の第2項によって行われるが、(4-29)式より、一般的に、

$$\min_{kl} B_{kl}^{(m)} + d_{kl} \geq u_k^{(m)} + d_{kl} \quad (4-36)$$

である。また、 L に沿った $u_j^{(m)}$ の移行の結果ノード i^* に作られる最早到着時刻は、 $u^{(m)}$ より小さいはずである。故に、

$$u_{i^*}^{(m)} > u_j^{(m)} + (d_{jl_1} + d_{l_1 l_2} + \dots + d_{l_p i^*}) \quad (4-37)$$

となる。従って、式(4-33), (4-34)によって、

$$\begin{aligned} f(i^*, u_{i^*}^{(m)}) &> f(i^*, u_j^{(m)} + (d_{jl_1} + \dots + d_{l_p i^*})) \\ &\geq f(l_p, u_j^{(m)} + (d_{jl_1} + \dots + d_{l_{p-1} l_p})) \\ &\geq \dots \geq f(j, u_j^{(m)}) \end{aligned} \quad (4-38)$$

となり、式(4-35)に反する。 (証明終)

定理4. 5によって、一度式(4-35)によって選択されたノード i^* の最早到着時刻は以後更新されることはなく、 i^* から隣接ノードへの最早到着時刻の移行が式(4-31)によって行われた後は再び行われる必要はないことがわかる。従って、以降の計算ではノード i^* についてまったく無視しても構わない。

式(4-33), (4-34)を満たす具体的な関数 $f(\cdot, \cdot, \cdot)$ としては、例えば次のような関数がある。

$$f(i, t) = t + \mu(i, n) \quad (4-39)$$

$\mu(i, n)$ はノード i から目的ノード n への古典的な最短走行時間であり、Yenの解法を使ってオーダー $O(n^3/4)$ で求めることができる。

以上の議論から、式(4-39)を使った部分台車運行型 SPPTW の解法が以下のように提案できる。但し、記号 E はそれまでの計算で式(4-35)によって選択されたことのあるノード i^* の集合を示し、 I はそれまでの計算で有限の最早到着時刻をもつノードの集合を示すものとする。

i) Yenの解法¹¹⁴⁾により、ノード $i \in N$ から目的ノード n への最短走行時間 $\mu(i, n)$ を求める。

$$\text{ii) } u_1^{(0)} = t_0, u_i^{(0)} = \infty (i \neq 1), E = \emptyset, I = \{1\} \quad (4-40)$$

iii) $I - E = \emptyset$ ならば終了。解は存在しない。

$I - E \neq \emptyset$ ならば、式(4-39)の関数 f に対して、式(4-35)を用いて、

$$f(i^*, u_{i^*}^{(m)}) = \min_{j \in I-E} f(j, u_j^{(m)}) \quad (4-41)$$

とする。 $i^* = n$ ならば終了。ノード n への最適経路が求められた。

そうでないならば、 $(i^*, k) \in A, k \notin E$ となるすべての k に対し(4-31)式を用いて $u_k^{(m)}$ の更新を行う。この更新の結果、無限大から有限値の $u_k^{(m)}$ をもつようになったノード k の集合を S とする。

$$\begin{aligned} \text{iV) } E &= E \cup \{i^*\} \\ I &= I \cup S \end{aligned} \quad (4-42)$$

とする。 $m = m + 1$ としてステップiii)へ戻る。

上のアルゴリズムは、手続きiii)における $u_k^{(m)}$ の更新が分枝限定法の分枝にあたり、式(4-41)による関数 f の評価が分枝限定法の下界評価にあたることから、B-F型及びYen型の解法と対比するため、この解法アルゴリズムを以後分枝限定型の解法アルゴリズムと呼ぶ。このアルゴリズムによる計算量は、オーダー $O((n_0 + m_0) \cdot n^2 + n^3/4)$ となる。分枝限定型の解法アルゴリズムでは、他の二つの解法アルゴリズムと異なり、ステップiii)において $i^* = n$ となることにより終了でき、必ずしも目的ノード以外のすべてのノードでの最早到着時刻を求めずに計算を終えることができる。

4. 2. 5 台車運行型SPPTWの一般解法

4. 2. 4節で提案した解法アルゴリズムは、目的ノードが一つの場合の部分台車運行型SPPTWを対象とした解法であった。元の台車運行型SPPTWの一般解法では、4. 2. 2節で述べたように、修正ネットワークにおいて複数の目的ノードをもつ問題を解かなければならない。B-F型及びYen型ではすべてのノードへの最早到着時刻を計算するので、複数の目的

ノードが存在しそのいずれかへ到着すればよいのであれば、それらのノードにおける最早到着時刻を比較して最も小さいものを最適解として採用すればよい。また、分枝限定型の場合には、式(4-41)により選択したノード i^* がいずれかの目的ノードと一致したとき終了し最適解とすればよい。

これまでの議論を総合すると、アークの走行時間に非負性を仮定しない台車運行型SPPTWの一般化された解法アルゴリズムを、次の二段階の手順で構成することができる。

1) 4. 2. 2節で述べた手順によってネットワークを再構成し、ノードにおけるタイムウィンドウが一つの通過可能時間区間のみから成るようにする。このとき、複数の目的ノードの生成も許す。

2) 目的ノードが一つの場合は、B-F型、Yen型または分枝限定型の解法アルゴリズムを用いて台車運行型SPPTWを解く。なお、複数の目的ノードの存在する場合は、上で述べたように修正した解法を用いる。

本研究では、アーク上の走行時間が必ずしも非負ではない一般化されたネットワークにおいて、台車運行型SPPTWに対する三つの解法アルゴリズムを提案した。4. 2. 4節から明かなように、いずれの解法でもノード数 $|N|$ の3乗のオーダーの計算量を要する。一方、3. 4. 2節では同じ問題に対して走行時間の非負性を仮定した下での解法を提案した。その解法アルゴリズムは基本的にはここで提案した分枝限定型の解法と同じであり、その計算量は $|N|$ の2乗のオーダーである。両者の計算量は、古典的最短経路問題においてアークの非負性を仮定しない場合のBellman-FordやYenの解法、非負性を仮定した場合のダイクストラ法¹⁰⁰⁾における計算量と基本的には一致しており、その対応関係はタイムウィンドウを考慮した場合の式(4-19)と考慮しない場合の式(4-23)が基本的に同じ構造をもっていることに由来するものである。

4. 2. 6 数値例

ここでは、数値例により4. 2. 4節で提案した三つの解法の実行効率について考察する。

対象とする搬送ネットワークは3. 6. 1節の図3. 2と同じであるが、有向アーク(33,25), (35,27), (37,29), (39,31)には-2の負の走行時間を与えるものとする。また、それらと逆の方向をもつ有向アーク(25,33), (27,35), (29,37), (31,39)は存在しないものとする。このネットワークは40個のノー

ドと92個の有向アークから成る。また、台車間における最小干渉許容時間は $\epsilon = 1.0$ とする。

このネットワーク上で、次の手順により次々に台車運行型SPPTWを作成しては解いていくシミュレーションを実施した。このネットワークへの投入台車数 q は、2台から2ずつ増加させながら12台まで変化させた。

(1) q 台の台車に対し出発ノード、目的ノードをランダムに選ぶ。但し、各台車の出発ノード、目的ノードは重複しないように選択する。この問題を文献115)で提案されている組合せ論的方法によって衝突のない経路を導出し、初期経路として各台車に割り当てる。

(2) q 台の台車の内、最も早く走行の終了する一台の台車(要求待台車)を選ぶ。その到着ノードをあらためて出発ノード s とし、そのノードへの到着時刻 t_0 を出発可能時刻とする。さらに、この台車に対するあらたな要求としての目的ノード e を他の台車の到着ノード以外のノードからランダムに選ぶ。その他の台車の経路には変更は加えず固定し、これを用いることにより、3.2.2節で述べた方法でタイムウィンドウを計算する。以上により、出発可能時刻を t_0 、出発ノードを s 、目的ノードを e とする台車運行型SPPTWを作成し、これを4.2.4節の解法アルゴリズムを用いて解く。

(3) 上の手順(2)を、B-F型、Yen型及び分枝限定型の解法アルゴリズムの各々に対して、連続して400回ずつ繰り返す。但し、400題の問題においてランダムに与える400個の目的ノードは、各解法に対して同一なものとする。

表4.1 平均通過可能時間区間数

台車数	2	4	6	8	10	12	14	16	18	20
ノード	1.1	1.3	1.5	1.5	1.6	1.8	1.6	1.5	1.8	1.7
アーク	1.0	1.1	1.2	1.2	1.3	1.3	1.3	1.2	1.3	1.3

表4.1は、台車数 q に対して、作成された最短時間経路問題の通過可能時間区間数を問題一題当り、ノードまたはアーク一つ当りで平均をとった結果である。台車数の増加と共に、作成されるタイムウィンドウの通過時間区間数もほぼ増加している。

図4.2は、各解法により、最適経路導出までにそれぞれの漸化式におい

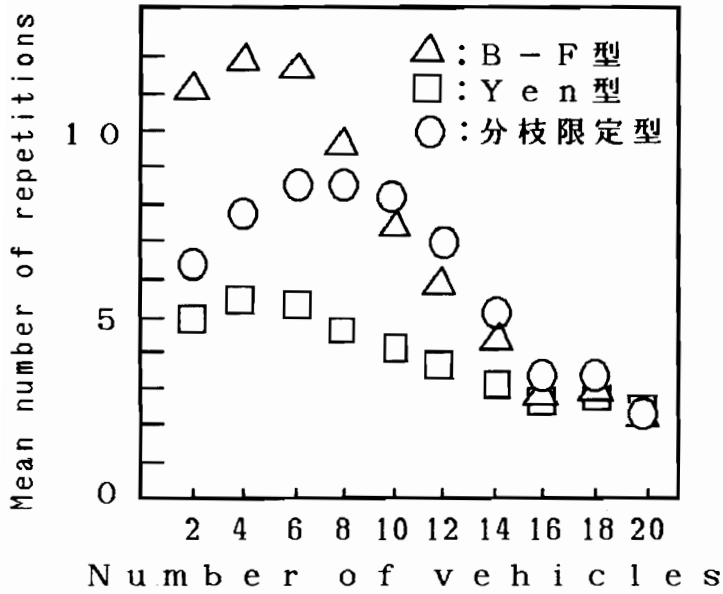


図 4 . 2 繰り返し回数

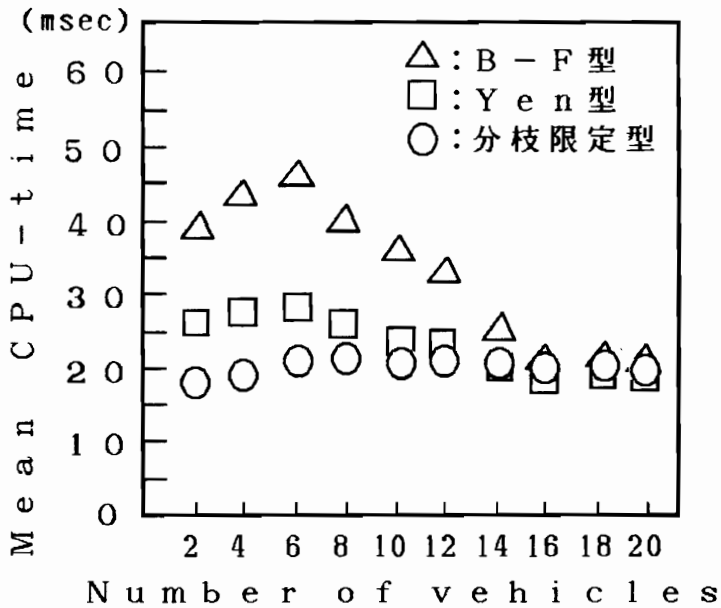


図 4 . 3 計算時間

て繰り返された計算の繰り返し回数 (m 値) の平均値を、また、図4. 3は各解法における平均の計算時間を示したものである。使用した計算機は、IBM3083 (VM/CMS) である。図中の三角、四角及び丸は、それぞれB-F型、Y e n型及び分枝限定型の解法アルゴリズムによるデータを表わしている。

それぞれの解法における繰り返し回数の表わす意味は異なる。計算量の面からみると、B-F型とY e n型では一回の繰り返し毎にすべてのノードに対しそれぞれ式(4-19)、(4-29)の基本算法を計算しなければならないという意味では同じである。この両者の比較ではY e n型の方がB-F型より繰り返し回数は常に少ない。この繰り返し回数の大小関係がそのまま計算時間にも反映しており、Y e n型の解法による計算時間の方がB-F型より常に小さい。これに対し、分枝限定型の解法では、一回の繰り返しにつき式(4-41)によるノード i^* の選択と i^* の隣接ノードの数だけ式(4-31)による計算を必要とする。従って、他の解法との繰り返し回数の違いがそのままには計算時間の違いに反映せず、繰り返し回数の多さに比べると計算時間はそれほど大きくない。計算時間の点では、分枝限定型が他の解法にくらべ最も優れているといえる。

4. 3 走行リスケジュールリング問題

3. 4節及び4. 2節では、台車運行型SPPTWを分枝限定法や動的計画法を用いて多項式計算時間で解く解法アルゴリズムを考えた。これらの経路決定法により、目的ノードに最早時刻に到着できる経路(ノード列)と各ノードにおける到着・出発時刻を確定的に決めることができる。しかし、台車が実際に走行する際には、台車の走行時間に種々の不確定要素が介在することから、計画された時刻どおりには到着または出発できない可能性がある。このように台車の実際の走行と当初の計画との間にずれが生じた際には、そのずれを補間するためのリスケジュールリングが必要である。特に、実際の走行に遅延が発生した場合には、以後の走行においてこの遅れを取り戻すようなリスケジュールリングが望まれる。

ここでは、計画された走行スケジュールの経由ノードにおける到着・出発時刻に対し、実際の走行に遅れが生じた場合のリスケジュールリング手順について考察する。まず、4. 2節で提案された解法を用いて台車型SPPTW

を解くことにより、最も早く目的ノードへ到着する走行スケジュールを立案する。その後、目的ノードへの最早時刻到着を保証するための各ノードからの最遅出発時刻を動的計画法に基づく漸化式によって求める。そして、この最遅出発時刻を用いることにより、実際の走行において遅延が発生した場合、その遅延の大きさに応じて、できるだけ遅れを取り戻すような走行リスケジュールリング手順を提案する。4. 2. 2節で明らかになったように、台車運行型SPPTWは、それと等価ないくつかの部分台車運行型SPPTWに分割できる。従って、以後の問題においてはこの部分台車運行型SPPTWのみを対象とする。

4. 3. 1 実行可能な走行スケジュールの表現法

ノード1を時刻 t_0 以降に出発しノード n に到るあるノード列 $H = \{h_1 = 1, h_2, \dots, h_p = n\}$ の実行可能性を考える。図4. 4はこのノード列のタイムウィンドウとそれらのノードをつなぐアークのタイムウィンドウを縦に並べたものであるとする。

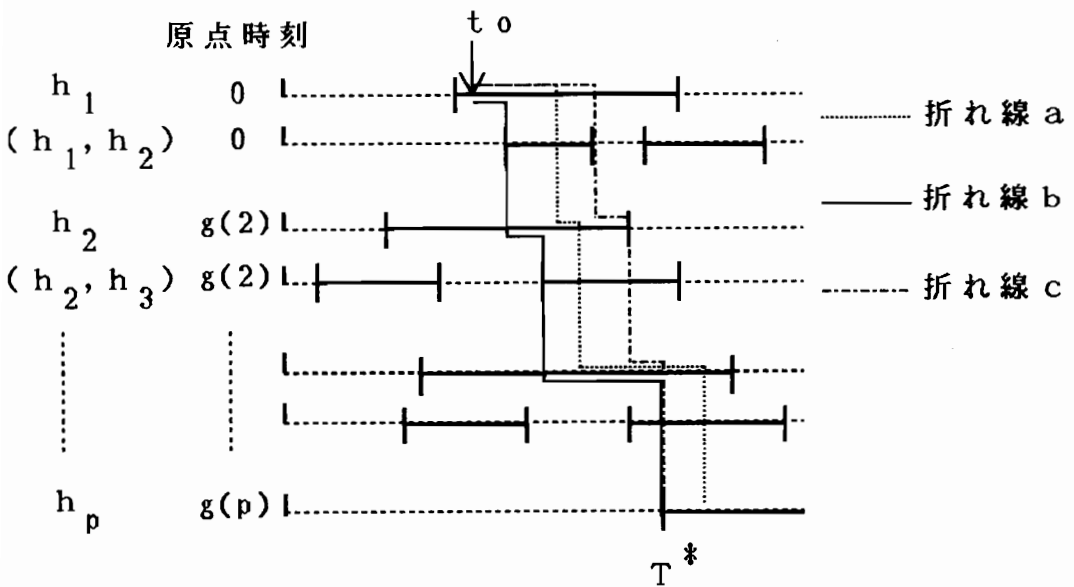


図4. 4 実行可能な走行スケジュール

但し、ノード h_1 とアーク (h_1, h_2) のタイムウィンドウの原点は時刻0

であるが、ノード h_2 とアーク (h_2, h_3) では時刻 $d_{h_1 h_2}$ としており、一般にノード h_k とアーク (h_k, h_{k+1}) では次の時刻を原点にとっている。

$$g(k) = \sum_{j=1}^{k-1} d_{h_j h_{j+1}} \quad (4-43)$$

もし台車がノード h_k のタイムウィンドウを時刻 t に出発したならば、ノード h_{k+1} には時刻 $t + d_{h_k h_{k+1}}$ に到着するから、ノード h_k からの出発とノード h_{k+1} への到着をノード h_k と h_{k+1} のタイムウィンドウ間をつなぐ垂直線で表現できる。また、各ノード上での停止はそのノードのタイムウィンドウ内での横線で表現できる。折れ線 a は、時刻 t_0 にノード 1 を出発した台車のある実行可能な走行スケジュールをこの方法で表わしたものである。

この表現法を用いれば、走行スケジュールが実行可能である必要十分条件は、“ノード 1 のタイムウィンドウ内の時刻 t_0 から始め、横線を引くことはノードのタイムウィンドウ内でのみ許され、その他の部分はすべて垂直線で結ばれた右下がりの連続な折れ線がすべてのタイムウィンドウを通るように描けること”である。

4. 3. 2 最早到着時刻と最早出発時刻の導出

前節で述べた必要十分条件から、あるノード列 H に沿って目的ノード n へ最早時刻に到着する実行可能な走行スケジュールは、次のような描き方によって求めることができる。

- 1) まず、ノード 1 のタイムウィンドウ中の時刻 t_0 から折れ線を描き始める。 t_0 をノード 1 への到着時刻とみなす。
- 2) ノード k への到着まで折れ線が描けたならば、次の要領でノード k と次の隣接ノード i のタイムウィンドウを結ぶ垂線をおろす。ノード k での到着時刻以降にアーク (k, i) のタイムウィンドウを通り、かつ i のタイムウィンドウ中の最も早い時刻に到着するように垂線を引く。この垂線は、ノード k からの出発時刻、ノード i への到着時刻を表わす。

ノード k への到着時刻と出発時刻を横線で結ぶ。

- 3) 手順 2) を目的ノード n への到着時刻が決定されるまで繰り返す。

この方法では、ノード列 H のすべてのノードでの最早到着時刻が同時に計算されていく。図 4. 4 の折れ線 b がそのように描かれた走行スケジュール

である。

ノード k への最早到着時刻 u_k^E から隣接ノード i の最早到着時刻 u_i^E を求める手順 2) の計算式は、3. 3. 2 節の演算子を用いて、次式で表わされる。

$$S'(k, i) = (T(i) \ominus d_{ki}) \otimes S(k, i) \quad (4-44)$$

$$B_{ki} = [u_k^E, b_k] \otimes S'(k, i) \quad (4-45)$$

$$u_i^E = \underline{\min} B_{ki} + d_{ki} \quad (4-46)$$

(4-15)～(4-17)式からわかるように、この計算式は、あるノードへの最早到着時刻をそれに接続するノードの最早到着時刻から動的計画法により求める方法がある特定のノード列 H に対して適用したものである。手順 1)～3)の方法は、4. 2. 4. 1 節で提案した解法を折れ線表示によって説明するものである。4. 2. 4. 1 節の $B-F$ 型解法では、次のような漸化式によって任意のノード $i \in N$ への最早到着時刻を導出する。

$$i) \quad u_1^{(0)} = t_0, \quad u_i^{(0)} = \infty, \quad v_i^{(0)} = \infty \quad (i \neq 1) \quad (4-47)$$

ii) すべての $i \in N$ に対し、

$$u_i^{(m+1)} = \underline{\min} \left\{ u_i^{(m)}, \underline{\min}_{(k,i) \in A} \left(\underline{\min} B_{ki}^{(m)} + d_{ki} \right) \right\} \quad (4-48)$$

上式において、

$$u_i^{(m+1)} = \underline{\min} B_{k^*i}^{(m)} + d_{k^*i} \quad (4-49)$$

となる k^* によって $u_i^{(m)}$ が更新されたならば、

$$\zeta(i) = k^* \quad (4-50)$$

$$v_{k^*}^{(m+1)} = \underline{\min} B_{k^*i}^{(m)} \quad (4-51)$$

とする。但し、

$$B_{ki}^{(m)} = [u_k^{(m)}, b_k] \otimes S'(k, i) \quad (4-52)$$

この漸化式は $u_i^{(m)}$ ($i \in N$) が変化しなくなったとき終了し、その時点

で各ノードにおける最早到着時刻が求められたことになる。目的ノード n の最早到着時刻を与える最適なノード列は、 $q_0=n, q_1=\zeta(q_0), q_2=\zeta(q_1), \dots$ としてノード1まで逆に辿っていくことにより得られる。また、 $v^{(m)}_{q_1}, v^{(m)}_{q_2}, \dots$ によつてノード q_1, q_2, \dots からの出発時刻が得られる。これらの時刻を最早出発時刻とよぶことにする。

4. 3. 3 最遅出発時刻と最遅到着時刻の導出

前節と同様にすれば、目的ノードへ最早時刻に到着するために遅くとも出発しなければならない時刻（最遅出発時刻）を導出できる。図4. 4において目的ノード n への最早到着時刻 T^* を実現するためのノード列 H の各ノードにおける最遅出発時刻を求める折れ線の描き方は、次のとおりである。折れ線は左上がりに描かれていく。

- 1) まず、ノード n の最早到着時刻 T^* から折れ線を描き始める。 T^* から隣接ノード h_{p-1} のタイムウィンドウに垂線を上に引き、 h_{p-1} からの出発時刻を求める。
- 2) ノード n からノード k における出発時刻まで折れ線が描けたならば、次の要領でノード k と前方に隣接するノード i のタイムウィンドウを結ぶ垂線を描く。ノード k での出発時刻以前に、アーク (i, k) のタイムウィンドウを通り、かつノード i のタイムウィンドウ中の最も遅い時刻に出発するように垂線を上へ引く。この垂線は、ノード i からの出発時刻、ノード k への到着時刻を表わす。

ノード k への到着時刻と出発時刻を横線で結ぶ。

- 3) 手順2) を出発ノード1からの出発時刻決定まで繰り返す。

この手順により描いたものが、図4. 4の折れ線 c である。ノード k の最遅出発時刻 v^L_k から前方へ隣接するノード i の最遅出発時刻 v^L_i を求める上の手順2) の計算は、次式により表現できる。

$$S^n(i, k) = (T(i) \otimes S(i, k)) \oplus d_{ik} \quad (4-53)$$

$$C_{ik} = S^n(i, k) \otimes [a_k, v^L_k] \quad (4-54)$$

$$v^L_i = \max C_{ik} - d_{ik} \quad (4-55)$$

ノード列Hに対するこの計算法を応用すれば、次の漸化式によって、目的ノードnへの到着時刻 T^* を実現するための任意のノード $i \in N$ における最遅出発時刻が導出できる。 T^* は、式(4-47), (4-48)により求められた目的ノードnの最早到着時刻であるとする。

$$i) \quad v_n^{(0)} = T^*, \quad v_i^{(0)} = -\infty, \quad u_i^{(0)} = -\infty \quad (i \neq n) \quad (4-56)$$

ii) すべての $i \in N$ に対し、

$$v_i^{(m+1)} = \max \left\{ v_i^{(m)}, \max_{(i,k) \in A} \left(\max C_{ik}^{(m)} - d_{ik} \right) \right\} \quad (4-57)$$

上式において、

$$v_i^{(m+1)} = \max C_{ik^*}^{(m)} - d_{ik^*} \quad (4-58)$$

となる k^* によって $v_i^{(m)}$ が更新されたならば、

$$\eta(i) = k^* \quad (4-59)$$

$$u_{k^*}^{(m+1)} = \max C_{ik^*}^{(m)} \quad (4-60)$$

とする。但し、

$$C_{ik}^{(m)} = S^n(i, k) \otimes [a_k, v_k^{(m)}] \quad (4-61)$$

この漸化式は $v_i^{(m)}$ ($i \in N$) が変化しなくなったとき終了し、その時点で各ノードにおける最遅出発時刻が求められたことになる。ノードiにおける $u_i^{(m)}$ を最遅到着時刻とよぶ。

式(4-56), (4-57)による最遅出発時刻 $v_i^{(m)}$ に関して、次の定理が成り立つ。

[定理4.6] タイムウィンドウを考慮しない場合のノードiからnまでの最短走行時間を $\mu(i, n)$ とすると次式が成り立つ。

$$v_i^{(m)} \leq T^* - \mu(i, n) \quad (4-62)$$

(証明) まず、次の漸化式で計算される $w_i^{(m)}$ に対して、

$$v_i^{(m)} \leq w_i^{(m)}, \quad i \in N \quad (4-63)$$

が成り立つことを証明する。

$$i) w_n^{(0)} = T^*, w_i^{(0)} = -\infty \quad (i \neq n) \quad (4-64)$$

ii) すべての $i \in N$ に対し、

$$w_i^{(m+1)} = \max \left\{ w_i^{(m)}, \max_{(i,k) \in A} (w_k^{(m)} - d_{ik}) \right\} \quad (4-65)$$

式(4-56)と(4-64)より、 $m=0$ のとき式(4-63)が成立する。 m の値に対して式(4-63)が成立すると仮定する。

式(3-20),(4-61)より $\max C^{(m)}_{ik} \leq v_k^{(m)}$ であるから、式(4-57)より、

$$\begin{aligned} v_i^{(m+1)} &\leq \max \left\{ v_i^{(m)}, \max_{(i,k) \in A} (v_k^{(m)} - d_{ik}) \right\} \\ &\leq \max \left\{ w_i^{(m)}, \max_{(i,k) \in A} (w_k^{(m)} - d_{ik}) \right\} \\ &= w_i^{(m+1)} \end{aligned} \quad (4-66)$$

となり、常に式(4-63)が成り立つ。

以下において、この $w_i^{(m)}$ が $T^* - \mu(i, n)$ に収束することを証明する。 $h_i^{(m)} = T^* - w_i^{(m)}$ とおくと、式(4-65)から次のようになる。

$$\begin{aligned} h_i^{(m+1)} &= T^* - \max \left\{ T^* - h_i^{(m)}, \max_{(i,k) \in A} (T^* - h_k^{(m)} - d_{ik}) \right\} \\ &= \min \left\{ h_i^{(m)}, \min_{(i,k) \in A} (h_k^{(m)} + d_{ik}) \right\} \end{aligned} \quad (4-67)$$

また、式(4-64)から

$$h_i^{(0)} = 0, h_j^{(0)} = \infty \quad (j \neq n) \quad (4-68)$$

となる。式(4-68),(4-67)は、任意のノード $i \in N$ から n までの最短経路を求める Bellman-Ford の解法^{112) 113)} に他ならない。従って、 $h_i^{(m)}$ は $\mu(i, n)$

に収束し、故に、 $w^{(m)}_i$ は $T^* - \mu(i, n)$ に収束する。また、式(4-65)から、 $w^{(m)}_i \leq w^{(m+1)}_i$ である。以上から、

$$v_i^{(m)} \leq w_i^{(m)} \leq w_i^{(m+1)} \leq \dots \leq T^* - \mu(i, n) \quad (4-69)$$

となる。 (証明終)

4. 3. 4 四つの時刻の関係

4. 3. 2節及び4. 3. 3節における漸化式によって求められるノード i の最早到着時刻、最早出発時刻、最遲到着時刻及び最遲出発時刻を u^E_i , v^E_i , u^L_i 及び v^L_i で表わす。

最早到着時刻は、ノード 1 を時刻 t_0 以降出発する限りこれ以上早くは到着できないという意味で真に最早であり、最遲出発時刻は、ノード n に最早到着時刻 T^* に到着するためにはこれ以上遅くは出発できないという意味で真に最遲である。しかし、あるノードでの最早出発時刻は、式(4-49), (4-51)からわかるとおり、ある隣接ノードに最早到着に到着するためにそのノードを出発する時刻であり、最遲到着時刻は、式(4-58), (4-60)からわかるとおり、ある隣接ノードを最遲出発時刻に出発した場合にそのノードに到着する時刻を表わしているにすぎないことは注意を要する。

4. 3. 2節及び4. 3. 3節の漸化式により有限値として導出される各ノードの u^E_i 及び v^L_i の大小関係は、走行スケジューリングに対して次のような意味をもつ。

- 1) u^E_i が得られない場合：出発ノードからの時刻 t_0 以降の出発によつてはこのノードへの到達は不可能である。
- 2) v^L_i が得られない場合：このノードを出発して、目的ノードに最早到着時刻 T^* までに到着できるような走行スケジュールは存在しない。
- 3) $u^E_i \leq v^L_i$ の場合：このノードを通過する最適走行スケジュールが存在する。ノード i において時間区間 $[u^E_i, v^L_i]$ の間台車の待ち停止が可能であり、以後のスケジュールに遅延が生じない限り、目的ノードには最早到着時刻までには到達できる。
- 4) $v^L_i < u^E_i$ の場合：このノードを通過する最適な走行スケジュールは存在しない。

4. 3. 5 走行スケジュールのずれとリスケジューリング手順

目的ノードへの最早時刻到着という観点から、一つの最適走行スケジュールを決定するような場合は、式(4-47)～(4-52)の漸化式から得られる走行スケジュールを採用すればよい。しかし、実際の走行が不確定な要素により計画どおりに進まないような状況では、4. 3. 4節で議論したスケジューリング情報を利用して柔軟なリスケジューリングを実施してやる必要がある。多くの搬送システムにおいてはできるだけ早く搬送することが重要な評価尺度であり、実際の走行において遅延の発生する場合にはその遅延を回復するようなリスケジューリング法が望まれる。そのような方法として次のようなリスケジューリング法が考えられる。

まず、式(4-47)～(4-52)により最適走行経路 L と最早到着時刻 u^L_i を求め、式(4-56)～(4-61)により最遅出発時刻 v^L_i を求める。また、4. 3. 3節の手順1)～3)を用いて、この走行経路 L に沿うように限定した最遅出発時刻 v^L_i を求める。

最適経路 L に沿って搬送台車を走行させ、走行の途中ノード j において遅延が生じた場合は、その現時刻 t の大小により次のような決定を行う。

- 1) $t \leq v^L_j$ ならば、目的ノードへの最早時刻到着には走行経路の変更は必要でなく、そのままの走行経路を保持する。
- 2) $v^L_j < t \leq v^L_j$ ならば、走行経路 L は既に最適ではない。 v^L_i を求めた際に導出した式(4-59)の $\eta(\cdot)$ を用いて、以後の走行経路をノード列 $q_1 = \eta(j), q_2 = \eta(q_1), \dots, n$ に変更することにより、目的ノードへの最早時刻到着をめざす。
- 3) $v^L_j < t$ ならば、いずれの経路をとっても目的ノードへの最早時刻到着は期待できない。従って、新たに、出発ノードを j 、出発許可時刻 t_0 を t として、式(4-47)～(4-52)、式(4-56)～(4-61)によるリスケジューリングを行う。

4. 3. 6 数値例

図4. 5のような搬送ネットワークを考える。ノードの中の数字はノード番号を表わす。すべての搬送路は双方向走行可能であり、相反する方向をもつ二つの有向アークで置換する。搬送路の横に描かれている数字は走行時間である。7台の搬送車が搬送ネットワーク上にあり、その内の6台に対しては表4. 2のような走行スケジュールで走行することが既に決定されている

ものとする。いま、7台目の台車に対し、ノード1を時刻 $t_0=0$ に出発し、目的ノード11にできるだけ早く到着するような搬送要求が与えられたとする。このとき、7台目の台車の走行スケジュールを決定する問題を考える。

以上のような状況においては、すべてのノードにおけるタイムウィンドウは一つの時間区間から成る。従って、4.2.2節で述べたネットワークの再構成は必要でなく、出発ノードを1、目的ノードを11とする台車運行型SPPTWに対し、4.2.4節の解法アルゴリズムを直接適用できる。

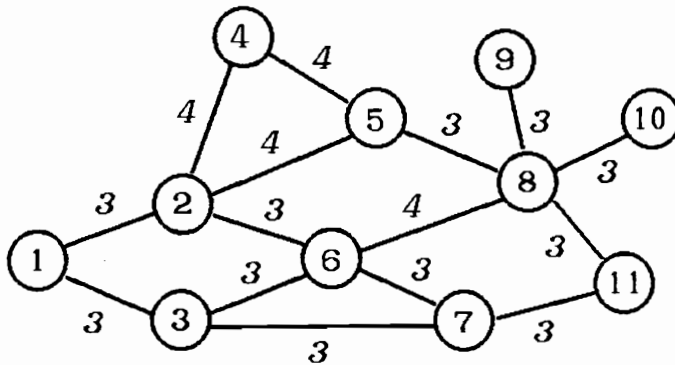


図4.5 搬送ネットワーク

表4.2 台車の走行スケジュール
ノード番号 (到着時刻, 出発時刻)

台車	走行スケジュール
1	3(*,0),1(3,*)
2	2(*,4),4(9,9),5(13,*)
3	6(*,11),3(14,*)
4	8(*,14),10(17,*)
5	9(*,15),8(18,*)
6	11(*,20),7(23,*)

式(4-47), (4-48)により導出される最適経路及び到着(出発)時刻を“ノード番号(最早到着時刻, 最早出発時刻)”の表記法で表わすと、 $1(t_0=0, 1)$,

2(4,8),5(8,11),8(14,17),11(20,*)となり、ノード11への最早到着時刻は $T^*=20$ となる。この最適経路Lを通ることに限定した場合の(最遅到着時刻,最遅出発時刻)は、4.3.3節の手順1)~3)を用いると、1($t_0=0,3$),2(6,9),5(13,13),8(16,17),11($T^*=20,^*$)となる。それらを図4.4のような折れ線表示で示したものが図4.6である。図にはLのノードにおけるタイムウィンドウが描かれており、最早到着(出発)時刻は破線の折れ線に、限定された最遅到着(出発)時刻は一点鎖線の折れ線によって表現されている。Lを構成するアークにおけるタイムウィンドウはすべて $(-\infty, \infty)$ となり、走行に対しなんら制約とはならないので表示していない。また、式(4-56)及び(4-57)により導出される真の最遅出発(到着)時刻とそれを実現するノード列Hは、1($t_0=0,3$),2(6,10),6(13,13),8(17,17),11(20,*)となる。このノード列 $H=\{1,2,6,8,11\}$ に限定した場合の最早到着(出発)時刻と真の最遅出発(到着)時刻を折れ線表示したものが図4.7の破線及び一点鎖線である。このLを構成するアークにおけるタイムウィンドウもすべて $(-\infty, \infty)$ であり、図の中には表示していない。この例題においては、上で述べた二つのノード列LとH以外のノード列には実行可能な走行スケジュールを許すものは存在しない。

いま、ノード列Lに沿って走行中、ノード2においてなんらかの遅延が生じたとする。このとき、4.3.5節の手順1)~3)を用いて、次のような走行スケジュールの立案が可能である。

- 1) 現時刻 t が、 $t \leq 9$ ならば、ノード列Lに沿った走行を続ける。
- 2) $9 < t \leq 10$ ならば、以後ノード列 $\{2,6,8,11\}$ に沿った走行に変更することにより、目的ノードへの最早時刻到着をめざす。ちなみに、この場合、ノード列Lに沿った走行が実行可能でなくなることは、4.3.1節の必要十分条件を満たすような折れ線が図4.6において描けないことからわかる。
- 3) $10 < t \leq 11$ ならば、目的ノードへの時刻 $T^*=20$ での到着は不可能であり、以後、ノード列 $\{2,6,8,11\}$ に沿って走行することによって、到着時刻 $t+d_{26}+d_{68}+d_{811}=t+10 (>20)$ での到着が可能となることが図4.7から読み取れる。
- 4) $11 < t$ ならば、目的ノードに到達する実行可能な走行スケジュールを立案することはできない。

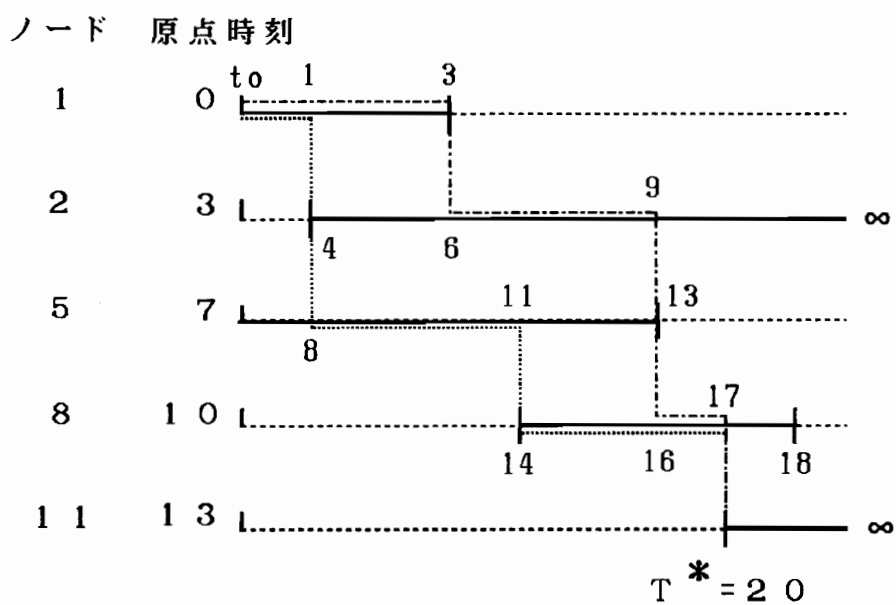


図 4 . 6 最早到着時刻

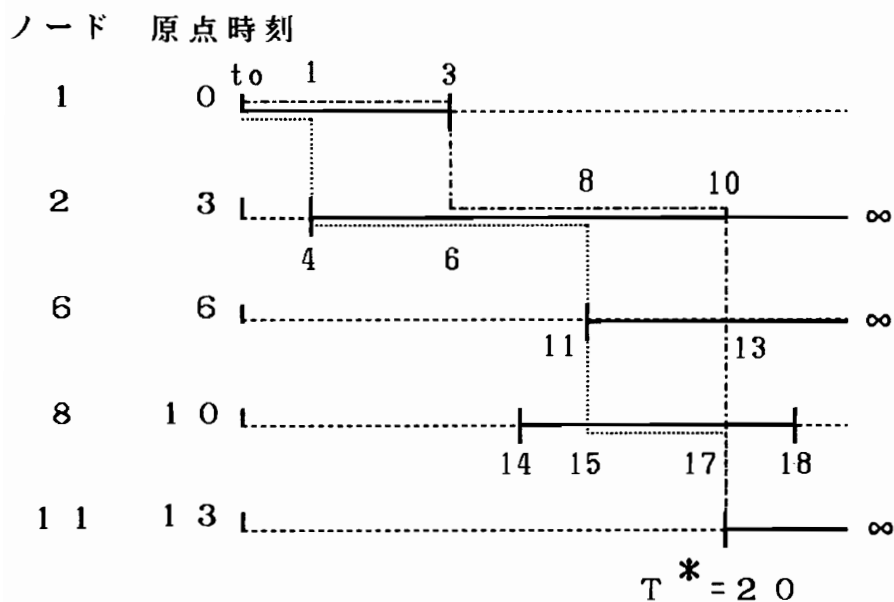


図 4 . 7 最遅出発時刻

4. 4 その他の現実的な運行経路決定問題への拡張

前節までは、搬送要求が一目的地からなる場合の最短時間経路の決定法についての議論を行った。また、その経路決定法は複数台車のうちの一台に対するものであった。ここでは、一台の台車に対する経路決定に限定するが、工場内で生ずる現実的な搬送要求に対する経路決定法をこれまでの方法の拡張として考察する。

その第1は、複数目的地から成る搬送要求を取り扱うこと、その第2は、目的地において作業時間のある搬送要求を取り扱うことである。加工フロアにおいて台車がある一つの工作物を受け持ったとき、通常、その加工順序に従って加工機械のあるステーションを巡回しなければならない。第1の要請は、このように順序付けられた複数目的地のある場合の経路決定が必要であることからくる。第2の要請は、加工機械等における加工時間待ちを、台車による搬送の中に加味する場合に必要であることからくる。このどちらの要請に対しても、これまでの方法を拡張し、適用できる。ここでは、走行時間を非負と仮定した3. 4節の問題における拡張の方法を考えるが、提案する方法は非負性を仮定しない場合の問題にも容易に適用できる。この二つの要請は、3. 4. 1節における存在可能時間帯の中に組み込むことができ、複数目的地をもつ問題や作業時間のある問題に対する解法アルゴリズムに拡張できる³⁶⁾。

4. 4. 1 複数目的地から成る搬送要求に対する経路決定法

いま、搬送要求が目的ノード群 $\{l_1, l_2, \dots, l_h\}$ から成り、台車はこの順序ですべての目的ノードを通過しなければならないとする。3. 4. 2節の解法アルゴリズムをこの問題に適用する上において考慮すべき点は、次のとおりである。

(1) 通過可能時間区間における存在可能時間区間の複数化

3. 4. 1節で述べたように、あるノード i における存在可能時間帯とは、そのノードにおいて台車の存在可能な時間帯であった。ノード i のタイムウィンドウ $T(i)$ の個々の通過可能時間区間 T_{ij} 内におけるその存在可能時間区間は、 T_{ij} への台車の最早到着時刻を始端としてもつ。3. 4. 1節の解法アルゴリズムは各ノードでの最早到着時間を漸近的に計算していくことに他ならなかった。

この解法による計算途中で、ノード i において、目的ノード l_k まで通過した後の最早到着時刻と目的ノード $l_{k'}$ ($k < k'$) まで通過した後の最早到着時刻 t' とが、 $t' < t$ の関係を満たしているとする。3. 4. 1 節で述べたとおり、すべての存在可能時間帯は出発ノード s での初期存在可能時間区間の移行の結果作成される。この移行は台車のノード間の移動を表しており、その過去の移行を辿ることにより存在可能時間帯を可能にする台車の動きも同時に把握できる。従って、 t' の最早到着時刻をもつ存在時間区間は、 t をもつ存在可能時間区間よりは、より多くの目的ノードを通り、かつ、ノード i へより早い時刻での到着が可能であるから、 t をもつ存在時間区間は以後の計算においては必要はない。逆に、 $t < t'$ の関係があった場合、 t の存在可能時間区間からの出発が、その後目的ノード k' まで通過した後、再びノード i へ時刻 t' より早い到着を可能にするかもしれない。従って、以後の計算においては、両者の存在可能時間区間とも必要である。

このことから、3. 4. 2 節の解法アルゴリズムによる計算過程においては、あるノードの一つの通過可能時間区間につき、一つの目的ノードも通過していない場合、 l_1 まで通過している場合、...、 l_{n-1} まで通過している場合のそれぞれに関する存在可能時間区間を同時に格納しておく必要がある。

(2) 目的ノードの通過

3. 4. 1 節で述べたような存在可能時間の移行を考える場合、台車の目的ノードの通過を考慮しなければならない。目的ノード l_k の通過は、次の条件によって決定すればよい。すなわち、存在可能時間帯が目的ノード l_{k-1} まで通過済みであること及びその存在可能時間帯が現在目的ノード l_k にあることの二つである。

3. 4. 2 節の解法アルゴリズムに対し、この二つの点を考慮して修正すれば、複数目的地をもつ搬送要求に対する経路決定法に拡張できる。

4. 4. 2 目的地において作業時間を有する場合の経路決定法

複数目的ノード $\{l_1, l_2, \dots, l_n\}$ の各 l_k において、作業時間 t_k を要するものとする。3. 4. 2 節の解法アルゴリズムを適用する上において考慮すべき点は、出発可能時間帯の推定と目的ノードの通過に関する次のことである。

3. 4. 1 節の存在可能時間帯は、タイムウィンドウ制約を満たせば、その時間帯の任意の時間から隣接ノードへ出発可能であった。すなわち、存在

可能時間帯は出発可能な時間帯に等しい。存在可能時間帯の隣接ノードへの移行の操作は、このことに基づいたものであり、移行の本来の意味からすれば、移行の式(3-34)の $E_{i_0 k_0}$ は出発可能時間帯で置き換えられるべきである。

目的ノードでは作業時間が設定されているため、到着時刻から少なくともこの作業時間の間は出発できない。従って、目的ノード l_k における最早到着時刻が t であれば、出発は早くとも $t+t_k$ 以降であり、出発可能時間帯は存在可能時間帯の始端を t_k だけずらせたものに等しい。この t_k のずらしが可能であれば、台車が k 番目の目的ノードを通過したとみなすことができる。

4. 5 結言

この章では、3章で述べた台車運行型SPPTWを他の問題に拡張することを考えた。

4. 2節では、台車の走行時間の非負性を仮定しない場合の一般化された台車運行型SPPTWを議論し、この問題の一般的な構造について考察した。その結果、この問題が最適性の原理をもつ部分問題に分割できること、この部分問題が古典的な最短経路問題と同じ構造を有するためBellman-Ford型及びYen型の解法アルゴリズムが提案できることが明らかになった。さらに、分枝限定型の解法アルゴリズムも提案した。これら三つの解法アルゴリズムをシミュレーションにより比較した結果、分枝限定型の解法が最も優れていることが明らかになった。

4. 3節では、台車の走行に遅滞が生じた場合の走行リスケジュール手順についての提案を行った。そこでは、目的ノードへ最早時刻に到着するために、経由ノードにおいて台車が遅くとも出発しなければならない時刻（最遅出発時刻）を導出するアルゴリズムを動的計画法を用いることにより考えた。そして、この最遅出発時刻を用い、経由ノードにおいて遅延が発生した場合、遅延の大きさに応じてその遅れをできるだけ取り戻すリスケジュール手順を提案した。

4. 4節では、複数目的地をもつ搬送要求や、作業時間の指示のある現実的な搬送要求に対する経路決定法について考察した。その結果、3章の台車運行型SPPTWの解法を部分的に修正することによって、これらの経路決定法を提案できることがわかった。

第5章 複数台車に対する運行経路決定問題

5.1 緒言

第3, 4章では、干渉をあらかじめ回避し、かつ走行時間最小となる経路決定法について、複数台車のうちの一台の経路決定に限定して議論した。その結果、問題は古典的な最短経路問題と同様な構造をもっており、動的計画法^{108) - 111)}や分枝限定法^{104) 105)}を用いた解法によって、多項式計算時間で解を求めることができることを証明した。ここでは、複数台車の経路の同時決定について考える。複数台車の経路決定においては、一台のそれに比べ、次のような事項を新しく考慮に入れなければならない。

その一つは、経路決定法の手法そのものの困難さである。台車間の干渉があるため、ある台車の経路は他の経路に影響を与え、個々の台車の経路を順次に決定してゆくような方法では、全体的な最適化は望めず、従ってすべての台車の経路決定を同時に行う必要がある。この章では、組み合わせ論的な手法を用いてこの問題に取り組む。

複数台車で考慮すべきいま一つの問題は、評価基準の問題である。走行時間を最小化する問題を考えるにしても、複数台車の場合、すべての台車の総走行時間最小化の問題と、すべての台車の中で最も遅い走行完了時刻（最大走行完了時刻）を最小化する問題が考えられる。以下で提案する解法では総走行時間最小化の問題を主に取り上げ、それを最大走行完了時刻最小化の問題に応用する場合についての考察については、5.5節で述べることとする。

実際の工場において、複数台車に対する搬送要求が発生する場合には、次の二つの状況が想定できる。一つは、搬送システムが始動し始める際に起こるように、すべての台車は走行しておらず、待機中である状態で、搬送要求が複数の台車に与えられる場合である。このような搬送系の起動時における複数台車の経路決定問題を、同時走行問題と呼ぶ。また、搬送システムが稼働中である場合には、いくつかの台車は走行中であり、他の台車は停止中であるような場面で、搬送要求がいくつかの台車に与えられるという状況が生じる。このような搬送システムの稼働時にみられる経路決定問題を、非同時走行問題と呼ぶ。同時走行問題に対しては、藤井ら¹¹⁵⁾や澤田²¹⁾が、実際の台車の走行に先だって干渉を予見し、総走行時間を最小化するような経路決

定法を提案した。

この章の5.4節までは、藤井らの方法を一般化することにより、非同時走行問題に対する三つの解法を提案し、数値例によりそれらの方法の比較検討を行う。5.5節では、ここで提案した解法の最大走行完了時刻最小化を評価基準とした問題に対する応用について考察する。搬送ネットワークに多くの台車が投入された場合には、干渉を起こさない経路（実行可能経路）そのものを求めることさえ困難になってくる。5.6節では、このような状況下における実行可能経路を求めるためのヒューリスティックな三つの方法を提案する。

5.2 非同時走行問題

ここで対象とする搬送系モデル及び台車間の干渉の条件は、2.6.2節で定義したものと同一であるとする。また、目的関数としては、総走行時間を一般化して、次のような重み付きの総走行時間を考える。

いま、時刻 t_0 で、あるノードにおいて待機中である台車（待機中台車）と、搬送要求を満たすべく走行中である幾つかの台車（走行中台車）の混在するような非同時走行状態にある搬送システムを考える。そして、待機中台車の幾つか（要求待台車）に搬送要求が生じたとする。この搬送要求は、ある重み係数で示される優先度を持ち、順序付けられたノード列で表される。台車は、示された順序で、示されたノードを経由することを要求される。この時、以後の全台車の走行に関し、互いに干渉を起こさず、かつ、台車別走行時間に優先度をかけたものの総和（重み付き総走行時間）を小さくするように、経路及び経路内の各ノードにおける待ち停止時間を決定することが問題となる。但し、時刻 t_0 でアークを走行中の台車は、そのまま、走行方向に直近のノードまでは走行するものとし、引き返したり停止したりしないものとする。非同時走行問題の経路決定においては、以前に決定されたが t_0 時点でまだ走行の終了していない走行中台車の経路及び待ち停止時間（以後、残りパスという）の取り扱いに関して、次の三つの方法が考えられる。

(1) 残りパスは、一切変更せず、走行中台車の経路と待ち停止時間をそのまま固定してしまう。（完全固定法）

(2) 残りパスの経路は変更しないが、待ち停止時間だけは再計算する。（経路固定法）

(3) 残りパスを一切考慮せず、要求待台車のみならず走行中台車の経路、待ち停止時間も再計算する。(更新法)

時刻 t_0 以前に与えられ、いまだ到着していない走行中台車の搬送目的地を、残り目的地または残り目的ノードと呼ぶ。また、時刻 t_0 からあるノードの到達時刻まで、決まった経路及び待ち停止時間に従うようある台車が義務付けられている場合、この経路及び待ち停止時間を、その台車の固定パスと呼ぶこととする。走行中台車に対し、完全固定法では、残りパスがそのまま固定パスとなる。

図5.1は、以上のことを例示したものである。横軸は時間軸である。台車1は搬送要求の発生した要求待台車、台車2, 3は走行中台車である。

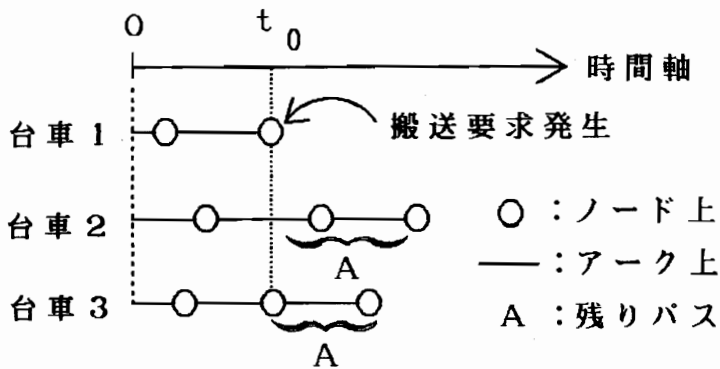


図5.1 非同時走行問題

以下では、残りパスに対する三つの方法による非同時走行問題の解法アルゴリズムについて説明する。

5.3 非同時走行問題に対する運行経路決定法

5.3.1 経路決定アルゴリズム

同時走行問題に関する研究に、固定パスの考え方を付加することにより得られた非同時走行問題に対する経路決定アルゴリズムは、以下のとおりである。このアルゴリズムは、その主要部分として、グラフ理論の第 k 最短経路探索法¹⁰⁶⁾と、待ち停止時間決定のための線形計画法から成る。なお、ここでは、台車数を n 台であるとする。

(STEP 1) 各台車に対し、次のような経路と待ち停止時間を求める。

搬送要求を受けた各要求待台車に対しては、第 k 最短経路探索法¹⁰⁶⁾を用いて、待機中のノードから搬送目的地を経由する最短経路を第 K 番目まで K 個求め、これを基本経路と呼ぶ。この基本経路においては、干渉回避のための待ち停止時間を考慮しない。

他の走行中台車に対しては、

(a) 完全固定法：残りパスの経路及び待ち停止時間を固定し、1 個の基本経路とする。

(b) 経路固定法：残りパスの経路のみを固定し、1 個の基本経路とする。

(c) 更新法：時刻 t_0 での現在地点から進行方向に直近のノードを出発点とし、残り目的地を経由する第 K 番目までの最短経路を求め、この経路のみを K 個の基本経路とする。

さらに、各々の基本経路に対し、その走行時間として、全アーク上の走行時間と固定した待ち停止時間の和をとる。

(STEP 2) $g = 1, T^* = \infty$ とする。

(STEP 3) 各台車の基本経路を要素とする基本経路の組み合わせのうち、未決定の待ち停止時間を考慮しない場合の重み付き総走行時間が第 g 最小となる値を t_g とし、その時の基本経路の組み合わせを $R_g = \{R^1_g, \dots, R^n_g\}$ で表わす。但し、 R^i_g は台車 i の基本経路である。

(STEP 4) $T^* \leq t_g$ ならば、(STEP 8)へ行く。そうでなければ、(STEP 5)へ行く。

(STEP 5) R_g に対して基本経路間での干渉の可能性の有無を判断し、可能性のある場合は、干渉回避の制約を付けた線形計画問題に定式化することにより、未決定分の待ち停止時間を決める。この際、変数を未決定分の待ち停止時間にとり、目的関数を待ち停止時間を考慮した重み付き総走行時間とする。なお、この詳細は後述する。干渉の可能性のない場合は、未決定分の待ち停止時間をすべて 0 とする。この時の待ち停止時間を考慮した重み付き総走行時間を t'_g とし、 R_g に待ち停止時間を付加した経路の組を R^*_g とする。

新しく決定された待ち停止時間がすべて 0 となったならば、(STEP 8)へ行く。そうでなければ、(STEP 6)へ行く。

(STEP 6) $T^* > t'_g$ ならば、 $R^* = R^*_g, T^* = t'_g$ とする。

(STEP 7) $g = g + 1$ として (STEP 3) へ行く。

(STEP 8) 終了。最適な経路及び待ち停止時間の組は、 R^* であり、その時の重み付き総走行時間は T^* となる。

本アルゴリズムは、(STEP3)～(STEP7)の手順を繰り返しながら解の改善を逐次的に行う。そして、(STEP4)において、それまでに求められていた待ち停止時間を考慮した重み付き総走行時間 T^* が待ち停止時間を考慮しない重み付き総走行時間 t_g 以下となった場合、または、(STEP5)において決定された待ち停止時間がすべて0となった場合には計算を終了する。

5. 3. 2 待ち停止時間の決定

本節で使用する記号は以下のとおりである。

n : 台車数

d_{ij} : ノード i, j 間の時間距離

M_q : 台車 q の基本経路に含まれるノード数

m_q : 台車 q の固定パスに含まれるノード数

n_{qi} : 台車 q の基本経路の i 番目ノード番号

x_{qi} : n_{qi} ノードでの待ち停止時間 (変数)

T_{qi} : 台車 q の固定パスの i 番目ノードでの待ち停止時間

ϵ : 最小干渉許容時間

W_q : 台車 q に与えられた搬送要求の優先度を示す重み付け係数

5. 3. 1 節 (STEP3)において得られた基本経路の組み合わせ R_g に対し、台車間の干渉を回避し、かつ重み付きの総待ち停止時間を最小にする問題は、以下の制約(5-2)～(5-7)の下で、次式で表される目的関数を最小にする線形計画問題¹⁰⁷⁾¹¹⁶⁾として定式化できる。

$$\sum_{q=1}^n \sum_{i=m_q}^{M_q} W_q \cdot x_{qi} \quad (5-1)$$

(1) 固定パス条件

$$x_{qi} = T_{qi}, \quad q=1, \dots, n, \quad i=1, \dots, m_q-1 \quad (5-2)$$

この x_{qi} は、もはや変数でなくなる。

(2) 非負条件

$$x_{qi} \geq 0, \quad q=1, \dots, n, \quad i=m_q, \dots, M_q \quad (5-3)$$

(3) 非干渉条件

2つの台車 q, r の基本経路の共通部分（共有パス）における非干渉条件は次のとおりである。

(i) 基本経路間に同方向の共有パスが存在する場合

図5. 2のように、ある I, J が存在し、 $n_{qI+h} = n_{rJ+h}, h=0, \dots, u$ が成立するならば、次のいずれかが成り立つ。

但し、 $1 \leq I, I+u \leq M_q, 1 \leq J, J+u \leq M_r$ とする。

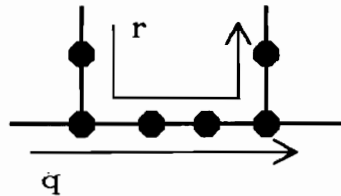


図5. 2 共有パス（同方向）

(ア) 台車 q が r より先に該当パスを通過する条件

$$\begin{aligned} & \sum_{i=1}^{I+h-1} (d_{n_{qi}n_{qi+1}} + x_{qi}) + x_{qI+h} \\ & \leq \sum_{j=1}^{J+h-1} (d_{n_{rj}n_{rj+1}} + x_{rj}) - \epsilon, \quad h=0, \dots, u \end{aligned} \quad (5-4)$$

(イ) 台車 r が q より先に当該パスを通過する条件

$$\begin{aligned} & \sum_{j=1}^{J+h-1} (d_{n_{rj}n_{rj+1}} + x_{rj}) + x_{rJ+h} \\ & \leq \sum_{i=1}^{I+h-1} (d_{n_{qi}n_{qi+1}} + x_{qi}) - \epsilon, \quad h=0, \dots, u \end{aligned} \quad (5-5)$$

特に、 $u = 0$ の場合は、共有パスではなく、共有ノードにおける非干渉条件となる。

(ii) 基本経路間に逆方向の共有パスが存在する場合

図5.3のように、ある I, J が存在し、 $n_{qi+h} = n_{rj-h}, h=0, \dots, u$ が成立するならば、次のいずれかが成り立つ。

但し、 $1 \leq I, I+u \leq M_q, 1 \leq J-u, J \leq M_r$ とする。

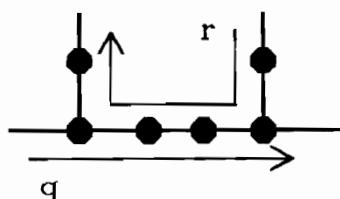


図5.3 共有パス（逆方向）

(ア) 台車 q が r より先に当該パスを通過する条件

$$\begin{aligned} & \sum_{i=1}^{I+u-1} (d_{n_{qi}n_{qi+1}} + x_{qi}) + x_{qI+u} \\ & \leq \sum_{j=1}^{J-u-1} (d_{n_{rj}n_{rj+1}} + x_{rj}) - \epsilon \end{aligned} \quad (5-6)$$

(イ) 台車 r が q より先に当該パスを通過する条件

$$\begin{aligned} & \sum_{j=1}^{J-1} (d_{n_{rj}n_{rj+1}} + x_{rj}) + x_{rJ} \\ & \leq \sum_{i=1}^{I-1} (d_{n_{qi}n_{qi+1}} + x_{qi}) - \epsilon \end{aligned} \quad (5-7)$$

以上のように、 n 台の台車のうち任意の2台を考えた時、これらの基本経路に共有パス（ノード）が存在すれば、いずれか一方の台車を先に通過させるという制約の下で1つのLP問題が与えられる。従って、解くべきLP問題の数は、2（全共有パス（ノード）数）である。これらの解の中で、(5-1)式を最小にするものを、与えられた基本経路の組の最適な待ち停止時間として採択する。

しかし、上述のLP問題の数は、共有パス（ノード）数に対し2のべき乗で増加し、現実的な計算を実施する上で極めて大きな障害となる。以下においては、これに対する対策としてLP問題数の削減方法について考察する。

5. 3. 3 線形計画問題数の削減

(1) 実行不可能問題の検出

以下に述べるように、台車 q , r の走行に関する幾何学的な判断によって、現実の台車運行において起こり得ないケースを抽出できる。

(i) 図5.4(a),(b)のような場合には、台車 q が先に共有パスを通過することは不可能である。

(ii) 図5.4(c),(d)のような台車の運行は、実行不可能である。

(iii) 図5.4(e)のような場合には、台車 q が共有パスAで先に通過し、かつ、Bで後に通過することは不可能である。

(iv) 図5.4(f)のような場合には、共有パスに先に到達する固定パスを持った台車が先に通過しなければならない。

(v) 図5.4(g)のような場合で、台車 q の基本経路の各ノードでの待ち停止時間を0と仮定した時の共有パスへの到着時刻が、台車 r より遅い場合は、台車 q が先に通過することは不可能である。

なお、これらと同様のケースは台車の進行方向を変えることにより他にも幾つか存在し、また、これらを組合わせた種々のケースを考えることもできる。以上に述べたようなそれぞれの場合に対応するLP問題は明らかに実行不可能であり、実行可能解をもたない。従って、このようなLP問題を幾何学的判断によって事前に削減すれば、経路決定において解かなければならないLP問題数を削減することができる。

(2) 有効なLP問題の検出

5.3.1節(STEP5)における線形計画法を解く手間は、変数の数、制約式の数が多い場合、実際の計算時間の面から無視できない要素となる。しかも、その計算によって待ち停止時間を求めた結果、それまでに得られた解を改善できないような無駄な場合に終ることも有り得る。そこで、実際に線形計画法を解く前に、あらかじめ、解の改善が起こりうるかどうかの可能性を考えておくことによって、無駄な計算を避け得る場合がある。

5.3.1節における、基本経路の組 R_g を考える。(5-3), (5-5)式より、次式が成り立つ。

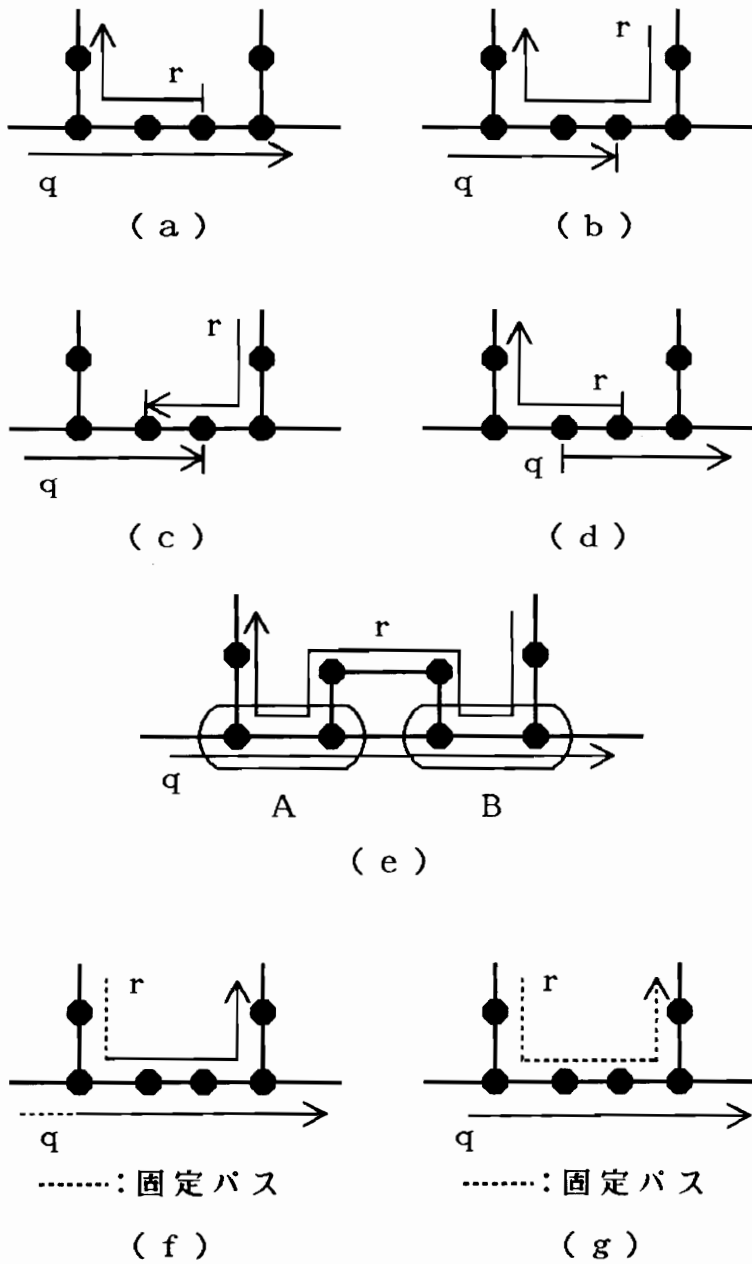


図 5 . 4 実行不可能問題の検出

$$\begin{aligned}
\sum_{i=1}^{M_q} x_{qi} &\geq \sum_{i=1}^{I-1} x_{qi} \\
&\geq \sum_{j=1}^{J-1} (d_{nr_j nr_{j+1}} + x_{rj}) + x_{rJ} - \sum_{i=1}^{I-1} d_{n_{qi} n_{qi+1}} + \epsilon \\
&\geq \sum_{j=1}^{J-1} d_{nr_j nr_{j+1}} - \sum_{i=1}^{I-1} d_{n_{qi} n_{qi+1}} + \epsilon \tag{5-8}
\end{aligned}$$

全ての共有ノード（パス）に対し、(5-8)式右辺の最大値を A_{qr} とすると、

$$\sum_{q=1}^n \sum_{i=1}^{M_q} W_q \cdot x_{qi} \geq \sum_{q=1}^n W_q \cdot \max_r \{0, \max_r A_{qr}\} = B \tag{5-9}$$

固定パスを考慮すると、

$$\sum_{q=1}^n \sum_{i=1}^{M_q} W_q \cdot x_{qi} \geq B - \sum_{q=1}^n \sum_{i=1}^{m_q-1} W_q \cdot T_{qi} = C \tag{5-10}$$

従って、Cの値はこの基本経路の組における重み付き総待ち停止時間の下限値を与える。故に、5.3.1節(STEP5)における t'_g に対し、 $t'_g \geq t_g + C$ が成立する。ここで、 $t_g + C \geq T^*$ であれば、5.3.1節(STEP6)における解の改善は期待できず、実際にLP問題を解く必要はない。

また、台車の全ての組 q, r に対し、 $A_{qr} \leq 0$ ならば、待ち停止時間はすべて0となることが容易にわかる。

5.4 三つの経路決定法の特徴と比較

ここでは、前節で提案した三つの経路決定法を比較検討する。まず、それらの定性的な考察を行い、次にシミュレーションによる定量的な分析を行う。最後に、これら三つの方法を組み合わせた現実的な経路決定手順について考察する。

5.4.1 経路決定法の定性的特徴

三つの経路決定法は、5.2節に示した基本的な考え方及び5.3.1節の具体的なアルゴリズムによって、次のような定性的特徴を有することがわ

かる。

(1) 経路固定法の実行可能解の集合は、完全固定法のそれを含み、更新法の実行可能解の集合は、経路固定法のそれを含む。また、5.3.1節(STEP1)のK値に関しては、大きな値を設定した方法による実行可能解の集合は、小さな値を設定した方法によるそれを含む。このことから、次のことがいえる。

(ア) 同一の非同時走行問題に対しては、その目的関数である重み付き総走行時間は、完全固定法、経路固定法、更新法の順に小さくなる。また、小さなK値の方法より、大きなK値の方法の方が、重み付き総走行時間は小さくなる。

(イ) 更新法によって解の求まらなかった問題は、経路固定法でも解は求まらない。この関係は、経路固定法に対する完全固定法、大きなK値の方法に対する小さなK値の方法に関しても成り立つ。

(2) 計算時間に関しては、次のことがいえる。

(ア) 基本経路を作成する場合の手間は、完全固定法、経路固定法では、搬送要求を受けた台車に対して、要求に含まれる目的地の数だけ第k最短経路探索法を用いることとなるが、更新法では、その他に、全ての走行中台車に対して、その残り目的地の数だけ使用しなければならない。

(イ) 5.3.1節(STEP3)で作られる基本経路の組合せ数は、完全固定法、経路固定法の場合、最大で $K^{(\text{要求待台車数})}$ であるのに対し、更新法では、最大 $K^{(\text{台車数})}=K^n$ となる。

(ウ) 5.3.2節の線形計画問題で、変数として待ち停止時間をもちうるノードは、完全固定法では、搬送要求を与えられた台車の固定パスを除いた基本経路の部分のノードであるが、経路固定法、更新法では、その他に、走行中台車の固定パスを除いた基本経路の部分のノードも考えなければならない。

また、完全固定法では、走行中台車間の干渉は、それらの残りパスにすでに付与されている待ち停止時間によって解消されているのに対し、経路固定法、更新法では、これらの台車間においても、5.3.2節(3)の非干渉条件を考慮する必要がある。

以上に述べたように、探索を必要とする経路数、解くべきLP問題数、LP問題当りの変数の数を考慮すると、求解に要する計算時間は一般に完全固

定法、経路固定法、更新法の順に大きくなる。

5. 4. 2 シミュレーションの条件と検討項目

三つの経路決定法の特徴を比較するためにシミュレーションを実施した。以下では、その条件設定と検討項目について述べる。

5. 4. 2. 1 条件設定

(1) ネットワークモデル

対象としたネットワークモデルは、図5. 5のネットワークA, B及びCである。台車は、各アークで双方向走行可能であるとする。各々のノード数(アーク数)は16(18), 20(26), 24(32)であり、ネットワークの規模はA, B, Cの順に大きい。

ネットワークの規模が大きいほど出発ノードと目的ノードが与えられた時の両者を結ぶ経路の代替数が多いことから、同じ台車数が投入されたとしても、ネットワークCの方がAよりもブロッキングやデッドロックの回避は容易になることが予想される。

(2) パラメータ値

台車数は $n=2\sim 5$ 台で変化させ、最小干渉許容時間 $\epsilon=1.0$, 基本経路の候補数 $K=5$, 優先度の重み係数は全て同じ $W_k=1.0(k=1, \dots, n)$ とする。

本シミュレーションでは、パラメータ ϵ , K , W_k を上のように固定し、台車数とネットワーク規模を変化させることによって各経路決定法の特徴について検討する。なお、使用した計算機はIBM3083 (VM/CMS) である。

(3) シミュレーション手順

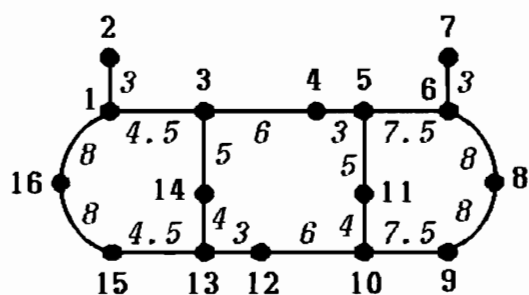
シミュレーションは、台車数($n=2\sim 5$)とネットワーク(ネットワークA~C)の組合せの各々に対して次のように実施する。

(ア) 全台車を待機中にし、その各々に対し一つの出発地、一つの目的地を同時に、かつランダムに与えることにより作成した同時走行問題を文献115)の解法を用いて解き、次の非同時走行の初期状態とする。

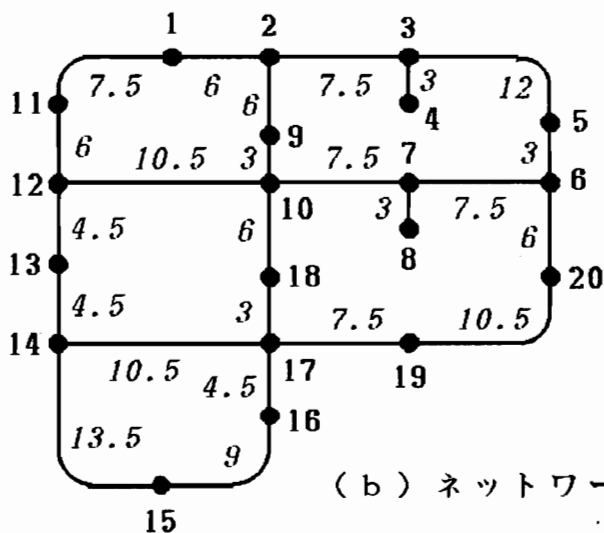
(イ) (7)で解の求まった場合について、搬送を最も早く終了した台車に対し新たに一つの搬送目的地をランダムに与え、非同時走行問題を作成する。

(ウ) 得られた同じ非同時走行問題を、完全固定法、経路固定法、更新法の各々の方法を用いて解く。

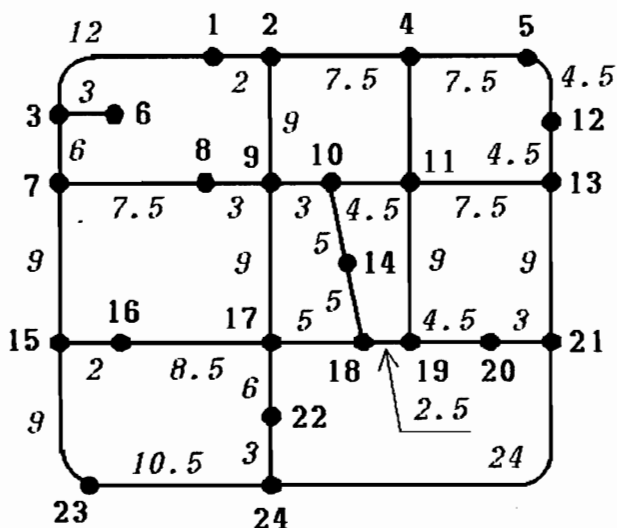
(エ) 手順(7)(イ)(ウ)を100回繰り返す。



(a) ネットワーク A



(b) ネットワーク B



(c) ネットワーク C

図 5. 5 ネットワークモデル

このような搬送要求の与え方は、実際の工場においては、加工機械等の設備が全てのノードに設定され、その間を優先度の等しい搬送要求が休みなく台車に与えられることにより、すべての台車が稼働率100%で運行されている場合に相当する。このような状況は台車の経路決定を行う運行管理システムにとってはかなり厳しい状況であり、現実の生産システムにおいてはより緩やかな状況にある。しかし、ここでは3種類の経路決定法の性能評価を目的としていることから、このような過酷な状況を想定した。

5.4.2.2 検討項目

(1) 解答不能率

本シミュレーションでは、5.4.2.1節(イ)の非同時走行問題において解の得られないことがある。以下では、本アルゴリズムで実行可能解が得られた問題を解答可能問題、そうでなかったものを解答不能問題と呼ぶことにする。解答不能問題の生じる原因としては、次が考えられる。

(a) 台車の目的地をランダムに与えるため、得られた問題が本来実行可能解の存在しない問題となった。すなわち、3.6.3節で述べた実行不能問題であった場合である。

(b) 実行可能解が存在するにもかかわらず、台車当りの基本経路数 K をある有限値に設定したために、実行可能解を得るに到らなかった。

上の(b)は、本アルゴリズムを実計算機上を実現する際の記憶容量の制約により、 K 値を有限値に設定したことによる。但し、この K をより大きな値に設定できれば実行可能解を得ることも可能である。また、5.3.1節のアルゴリズムにおける基本経路の作成法及び可能な基本経路の組合せの大きさから判断できるように、同じ非同時走行問題に対する実行可能解の集合は、完全固定法、経路固定法、更新法の順に包含される関係にある。従って、解答不能率(全問題数に対する解答不能問題数の割合)はこの順に小さくなる。前述したように、本シミュレーションでは過酷な台車運行を想定したため、各方法における解答不能率の違いが明確になった。

もし、一台しかない空き台車に与えられた搬送要求に対してある経路決定法が解答不能となると、他の台車の空きを待つか、ある時間の経過を待つて、この搬送要求の再割当てを行わなければならない。その間、空き台車の稼働は停滞することになる。この意味で、解答不能率は、経路決定法の実際の搬送システムでの台車の運行効率を評価する上で極めて重要な尺度である。

(2) 解の改善度

解法アルゴリズムでは、基本経路の組合せを逐次作成しながら、目的関数である重み付き総走行時間に関し改善を行う。ここでは、この基本経路の組合せの探索回数（ g 値）の変化を調べ、また、初期実行可能解と最適解における目的関数値を比較することにより各経路決定法の特徴を明らかにする。最適解の g 値は、最適解となった基本経路の組合せを見いだすまでにどれだけの基本経路の組合せが探索されたかを示しており、(STEP4)で解の最適性を確認し計算が終了するまでに探索された基本経路の組合せ総数は、最大の g 値で示される。また、初期実行可能解と最適解における目的関数値を比較することで、目的関数に関する解の改善効果がわかる。ここでは、両者の目的関数値の差を初期実行可能解における目的関数値で割ったものを改善率として定義する。

(3) 計算所要時間

アルゴリズムの性能評価に関しては、解導出までに費やされる計算時間も重要な尺度となる。ここでは、本アルゴリズムを構成する主要なプロセスにおける個々の計算所要時間を検討することにより、計算速度の面から各経路決定法の特徴を明らかにする。

5. 4. 3 シミュレーションの結果とその分析

5. 4. 3. 1 解答不能率

本アルゴリズムで作成される基本経路の組合せの最大数は、更新法では、 K^n ($=5^n$)、固定法では K ($=5$) である。このことは、固定法と更新法での解の探索領域の大きさの違いを意味し、これが前述した実行可能解の集合の包含関係を形成する。本節では、このことを考慮した上で、各方法の解答不能率について考察する。

図5.6は、各経路決定法のシミュレーション結果における解答不能率を示したものである。ネットワークの規模が同じであれば台車数が多くなるほど、また、台車数が同じであればネットワーク規模が小さくなるほど干渉形態は複雑化し、解答不能となる割合も増加する。また、前述した各方法の実行可能解の包含関係より、同一ネットワーク、同一台車数では必ず完全固定法、経路固定法、更新法の順に解答不能率が小さくなる。特に、ネットワークAに5台の台車を投入するような極めて複雑な干渉形態の予想される状況

においては、図5. 6(a)で示されているように更新法の有効性は明かであり、完全固定法に比べ約74%もの改善がみられる。以上をまとめると、次のようになる。

- (1) ネットワークの規模を固定した場合、台車数とともに解答不能率は増加する。
- (2) 台車数を固定すると、ネットワークの規模が小さくなるにつれ解答不能率は増加する。
- (3) 完全固定法，経路固定法，更新法の順に解答不能率は減少する。

5. 4. 3. 2 解の改善度

表5. 1は、最適解到達まで、及びその最適性を確認するまでの基本経路の組合せ探索回数（g値）を、解答可能問題一題当りの平均値で示したものである。

表5. 1 基本経路の組合せ探索回数（g値）
（上段：最適解，下段：最大g値）

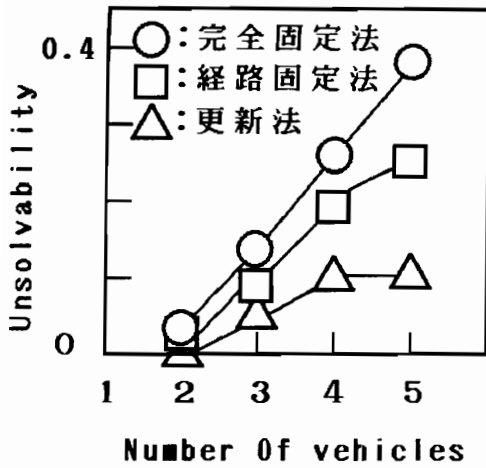
(a) 完全固定法 (b) 経路固定法 (c) 更新法

		ネットワーク		
		A	B	C
台 車 数	2	1.21	1.21	1.07
		1.23	1.23	1.07
	3	1.09	1.04	1.31
		1.36	1.08	1.40
	4	1.30	1.15	1.27
		1.46	1.29	1.33
	5	1.08	1.41	1.35
		1.44	1.53	1.57

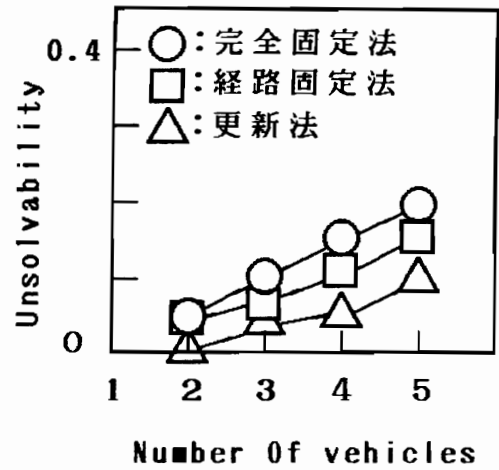
		ネットワーク		
		A	B	C
台 車 数	2	1.21	1.21	1.07
		1.23	1.23	1.11
	3	1.09	1.05	1.27
		1.51	1.20	1.47
	4	1.33	1.16	1.30
		1.73	1.47	1.80
	5	1.19	1.39	1.42
		2.34	1.86	2.02

		ネットワーク		
		A	B	C
台 車 数	2	1.42	1.39	1.31
		1.44	1.48	1.48
	3	1.45	1.09	1.47
		3.13	1.49	2.87
	4	3.58	2.10	1.51
		5.73	3.28	4.16
	5	5.38	2.47	2.30
		18.1	4.90	6.96

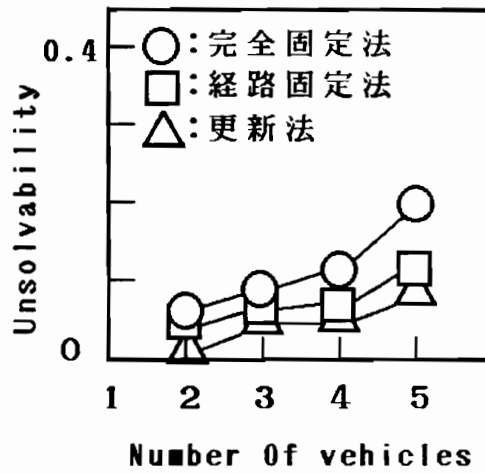
表5. 1より、完全固定法，経路固定法は、それぞれ解答可能問題一題当たり1回強の探索回数で最適解に達し、その最適性も1～3回の探索で確認されていることがわかる。但し、台車数が多くなり干渉形態が複雑になると、経路固定法は、完全固定法に比べ、最適解のg値、最大g値とも大きくなる傾向が認められる。これは、経路固定法では、走行中台車の基本経路の待ち停止時間が固定されていないため、待ち停止時間の固定された完全固定法では解の存在しないような複雑な干渉のある問題に対しても解の求められる場



(a) ネットワーク A



(b) ネットワーク B



(c) ネットワーク C

図 5. 6 解答不能率

合が発生し、そのような複雑な問題では、一般的に大きな g 値をもつ基本経路の組み合わせまで解が探索される傾向にあるからである。

一方、更新法においては、他の方法よりも最適解の g 値、最大 g 値ともに大きく、台車数の増加に伴って大幅に増加する。また、最適解の g 値は、ネットワークの規模が小さいほど、また、台車数が多いほど大きくなる傾向もみられる。これは、更新法では、その解答不能率が5、4、3、1節で考察したように固定法のそれよりも小さく、干渉の形態が複雑で固定法では解の得られないような問題に対しても解が得られることを基本経路の組合せ探索回数から説明しているといえる。すなわち、干渉が複雑になるほどより多くの基本経路の組合せが試され、その中から最適解が見いだされるようになるのである。また、更新法では、台車数が多いほど最大 g 値と最適解の g 値との差が大きくなり、最適性の確認に費やす探索の回数が多くなっていることがわかる。これは、次節で考察するように計算所要時間を大きくする原因の一つとなっている。

表5. 2は、各経路決定法において、最適解の初期実行可能解に対する目的関数値の平均改善率を示している。

表5. 2 改善率

(a) 完全固定法

(b) 経路固定法

(c) 更新法

		ネットワーク		
		A	B	C
台車数	2	.0	.005	.0
	3	.004	.0	.002
	4	.007	.001	.0
	5	.0	.017	.002

		ネットワーク		
		A	B	C
台車数	2	.0	.008	.001
	3	.002	.0	.018
	4	.006	.007	.022
	5	.0	.051	.053

		ネットワーク		
		A	B	C
台車数	2	.0	.007	.001
	3	.031	.004	.023
	4	.009	.006	.030
	5	.053	.063	.080

表5. 2に示した改善率については、ほぼ、完全固定法、経路固定法、更新法の順に大きくなっていることがわかる。しかし、最も改善の期待される更新法においても高々10%程度である。このことと先に述べたことを考え合わせると、更新法を用いて、最適解まで求めずに何らかの実行可能解の取得のみを考えることによって、少ない探索回数でかつ満足のいく解が得られると期待できる。

5. 4. 3. 3 計算所要時間

本アルゴリズムにおいて、計算所要時間に影響を及ぼす要因は、次の5つである。

- ・基本経路の探索
- ・最適性が判明するまでの基本経路の組合せ探索
- ・各台車の基本経路間で共有されるパス（共有パス）における干渉回避の可能性等に関する判定
- ・線形計画法による待ち停止時間の割り当て
- ・その他

以下に各項目について考察する。

(1) 基本経路の探索

図5. 7(a), (b), (c)は、ネットワークA, B, Cの基本経路探索に要した計算時間を示したものである。但し、経路固定法による結果は、完全固定法とほとんど同じであるので省略した。

固定法では、搬送要求の与えられた台車に対してのみ第k最短経路探索法が用いられるのに対し、更新法では、全ての台車に対して用いられる。従って、計算所要時間は、固定法では台車数に関係なくほぼ一定の値をとり、更新法では台車数に比例する増加がみられる。

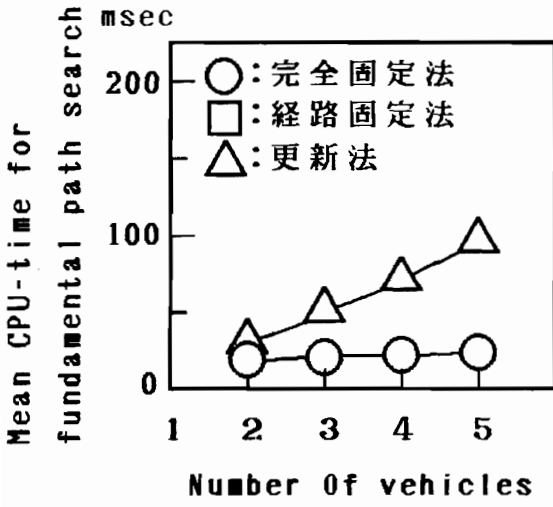
また、第k最短経路探索法のアルゴリズムの性質より、ネットワークの規模が大きくなるにつれ基本経路探索に要する計算時間は増加する。ネットワークAにおいては、固定法の台車数2～5での平均計算所要時間は20 msecであり、更新法による平均計算所要時間の傾きは20 msec/台である。これらの値は、ネットワークBでは、30 msec, 23 msec/台、ネットワークCでは、40 msec, 28 msec/台である。

(2) 最適性が判明するまでの基本経路の組合せ探索

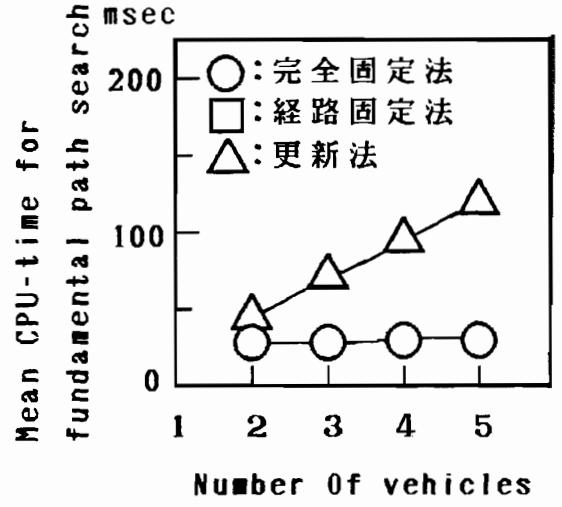
この組合せ探索回数(最大g値)については、5. 4. 3. 2節で考察したとおりである。以下に挙げる(3)～(5)の項目は、基本経路の各組合せに対してあてはまることであるため、探索すべき組合せ数が増加すれば、それぞれの項目に関連した計算所要時間も増加する。

(3) 各台車の基本経路間の共有パスにおける干渉回避の可能性等に関する判定

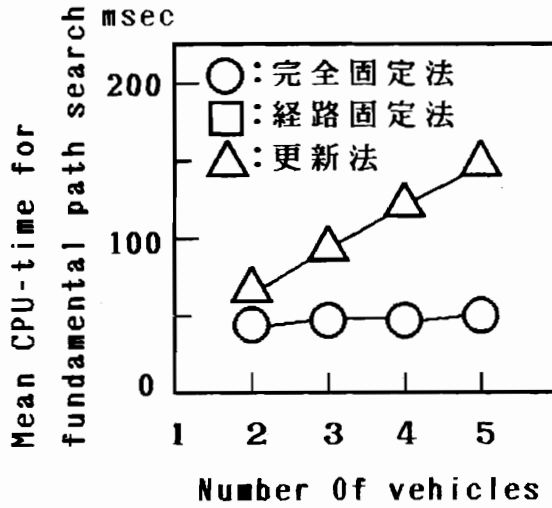
本アルゴリズムにおいては、干渉可能性の判定箇所は極めて多く、また、複雑であるため、この判定に要する計算時間のデータを独立して求めること



(a) ネットワーク A



(b) ネットワーク B



(c) ネットワーク C

図 5. 7 基本経路探索に要する平均計算時間

は困難である。ここでは、基本経路の一つの組合せ当り判定の対象とされた共有パスの平均数をひとつの目安として取り上げる。

図5. 8 (a), (b), (c)は、ネットワークA, B, Cについて、この値を示したものである。完全固定法では、走行中台車間の干渉は、各台車の残りパスに固定された待ち停止時間によってすでに解消されているので、判定の対象外となる。しかし、経路固定法、更新法では待ち停止時間が未定であるので、走行中台車間の干渉の可能性を考慮に入れなければならない。従って、完全固定法は、他の二つの方法に比べ干渉可能性の判定対象とされる平均共有パス数は少ない。

(4) 線形計画法による待ち停止時間の割り当て

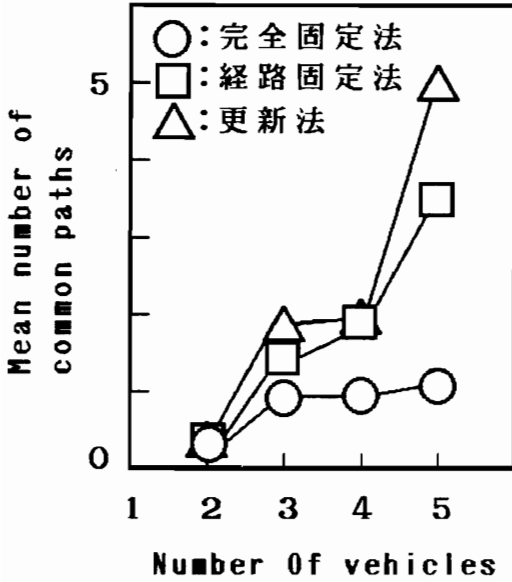
本アルゴリズムでは、基本経路の各組合せごとに、待ち停止時間を割り当てるためのいくつかのLP問題を解く。これに要する計算時間は、LP問題数及びLP問題の規模によって左右される。一つの基本経路の組に対し、2台の台車の干渉可能性のある箇所においてどちらが先に通過するかにより二つの制約不等式が考えられる。実際のアルゴリズム中では幾何学上、数式上の判断によりこの制約不等式数を削減しているが、制約不等式の組合せの数だけLP問題が生成されることになる。従って、(3)項における考察からわかるように、LP問題数は、一般に完全固定法では少なく、経路固定法、更新法では多くなる。また、完全固定法では、走行中台車の残りパスにおける待ち停止時間は定数となるのに対し、経路固定法、更新法では変数として取り扱い、走行中台車間の非干渉条件をも定式化しなければならない。従って、経路固定法及び更新法におけるLP問題の規模は、完全固定法のそれに比べ大きくなる。以上から、更新法及び経路固定法は、完全固定法に比べLP問題を解くための計算所要時間は大きくなる。

図5. 9 (a), (b), (c)は、ネットワークA, B, Cについて、一つの基本経路の組合せ当りのLP問題の解を求めるのに要する平均計算時間を示したものである。経路固定法と更新法には、それほどの差は認められない。

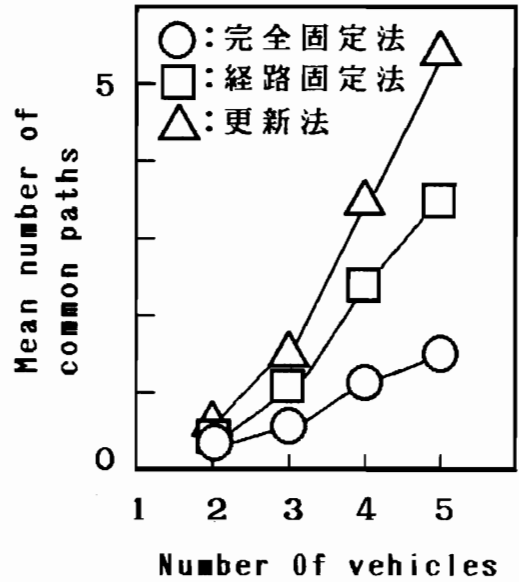
(5) その他

以上の項目の他に、本アルゴリズムを計算機のプログラム上で実行する場合必要なすべての手続きが計算所要時間に影響を及ぼすが、それを予測することは困難である。

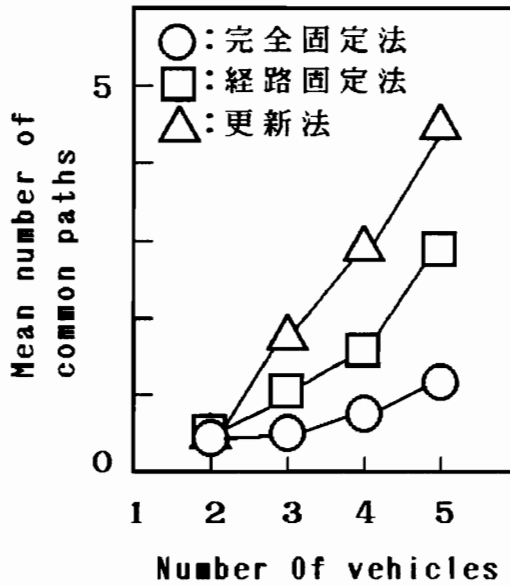
ネットワークA, B, Cにおける解答可能問題一題当りの計算所要時間は、図5. 10 (a), (b), (c)のとおりであり、上記の要因(1)～(5)の累積さ



(a) ネットワーク A

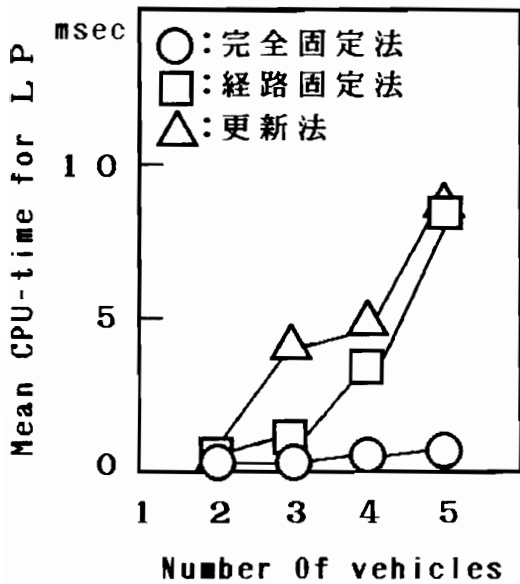


(b) ネットワーク B

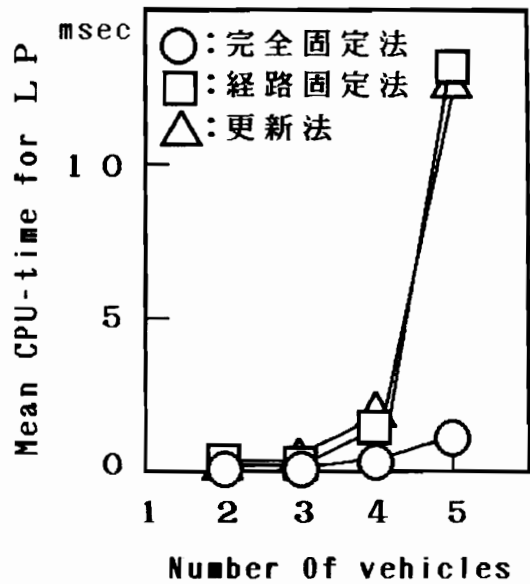


(c) ネットワーク C

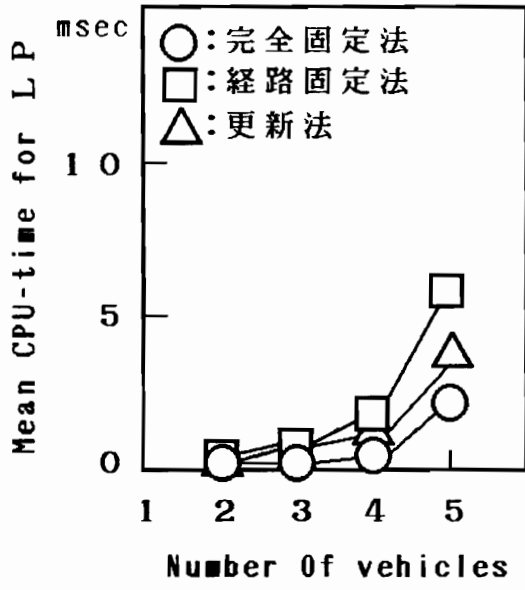
図 5 . 8 平均共有バス数



(a) ネットワーク A

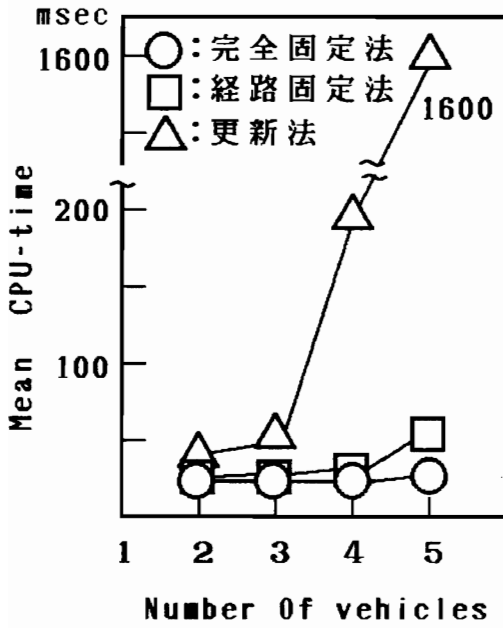


(b) ネットワーク B

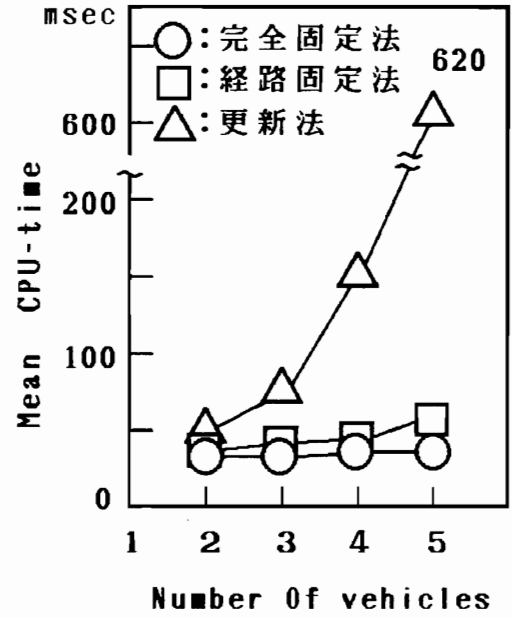


(c) ネットワーク C

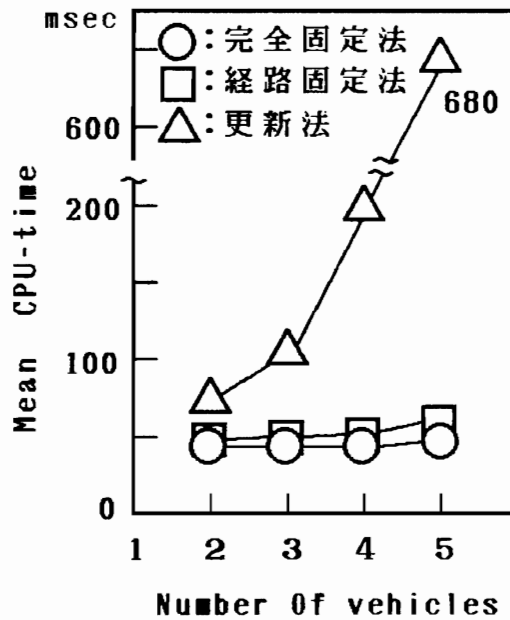
図 5. 9 LP 問題の求解に要する平均計算時間



(a) ネットワーク A



(b) ネットワーク B



(c) ネットワーク C

図 5. 10 平均計算所要時間

れた結果と考えることができる。

図では、更新法に対する計算所要時間が台車数に関して指数関数的に増加しているが、これは次のように説明できる。前にも述べたように基本経路探索のための計算時間は台車数に比例するが、台車数の増加に伴って、解の最適性を確認するための基本経路の組合せ探索回数は急激に増加する。このため全体としての計算所要時間は指数関数的な増加を示すのである。

5. 4. 4 更新法における計算所要時間の短縮

更新法は解答不能率の面で長所を有するが、反面、台車数の多い場合は計算所要時間が極端に大きくなる。しかし、5. 4. 3. 2節でも考察したように、最適解の初期実行可能解に対する目的関数値の改善率がそれほど大きくないことを考えあわせると、初期実行可能解のみの取得を考えれば、計算所要時間は短縮されしかも十分満足な解が得られると期待できる。

表5. 3は、更新法に対するこの近似解法を用いて、5. 4. 2. 1節と

表5. 3 近似解法の探索総数 (最大g値)

		ネットワーク		
		A	B	C
台車数	2	1. 4 2	1. 3 8	1. 3 0
	3	1. 2 5	1. 0 7	1. 2 9
	4	3. 3 7	2. 0 6	1. 3 6
	5	3. 7 3	2. 2 8	1. 7 1

表5. 4 近似解法の計算所要時間短縮率 (単位 %)

		ネットワーク		
		A	B	C
台車数	2	2. 3	2. 1	1. 8
	3	35. 8	4. 2	9. 2
	4	22. 8	14. 0	28. 2
	5	70. 9	52. 2	59. 0

同じ条件下でシミュレーションを行った場合の基本経路の組合せ探索総数（最大g値）である。表5.4は、近似解を用いない場合に対する近似解を用いた場合の計算所要時間の短縮率である。この表と表5.1(c)を比べてみればわかるように、近似解法を用いることにより探索回数の減少及びそれに伴う計算所要時間の短縮率は、台車数の多い場合に特に有効であることが確認できる。

5.4.5 現実的な経路決定手順に関する考察

実際に台車の運行管理を行う場合の経路決定に関しては、5.4.2.2節(1)で述べたように、実行可能解が求まり経路が決定できるか否かは台車の稼働率を考える上で重要な問題である。解答不能率に関する5.4.3.1節での考察から次のことがいえる。

(1) 実行可能な経路を決定できる能力という点では、更新法が最も優れている。特に、ネットワークの大きさに比べ台車数が多い等の複雑な干渉形態の予想される場合に、他の方法に対するこの優位性が顕著である。

さらに、実行可能解がある場合に走行時間の面で最適な解まで求めねばならないかどうかは、目的関数値に関する解の改善率や付加される計算時間の大小に依存するが、5.4.3.2節、5.4.4節での考察より次のことがいえる。

(2) 初期実行可能解の目的関数値に対する最適解のその改善率は、いずれの経路決定法に関してもそれほど大きくない。更新法においては、解の導出を初期解までで止めることで、解答不能率を変えずに計算時間を大きく短縮できる。

また、リアルタイムな運行管理を行うため、より少ない計算時間の経路決定法が望まれる。計算所要時間に関する5.4.3.3節での考察から次のことがいえる。

(3) 計算時間の面では、完全固定法は極めて優れており、ネットワークの大きさや台車数の変化に対して計算時間に大きな増減は生じない。

以上のことを考慮すると、完全固定法の計算時間の短さと更新法の解答不能率の低さというそれぞれの長所を活かした次のような現実的な経路決定手順が考えられる。

a) 固定法を用いて経路決定を試み、実行可能であるならば、その最適解を台車の経路として用いる。

b) a)で解答不能となった場合には、初期解のみを求めるように修正した更新法を用いて、他の台車も含めた総合的な経路決定を行う。

5. 5 最大走行完了時刻最小化問題に関する考察

5. 2～5. 4節までは、(重み付き)総走行時間を評価尺度とした経路決定問題を議論した。ここでは、先に提案した解法アルゴリズムを、5. 1節で述べたもう一方の評価基準、最大走行完了時刻最小化を用いた場合に適用することについて考察する。問題の設定は、5. 2節と全く同じ非同時走行問題であるとする。この問題に対し、搬送ネットワーク上にある複数台車の走行完了時刻の中の最も遅い完了時刻(最大走行完了時刻)を最小化する経路決定法を提案する。この経路決定問題に対しては、5. 3節で提案した解法アルゴリズムの部分的な手直しにより対応でき、その基本的な解法手順を変える必要はない。

以下では、5. 3節の解法アルゴリズムの修正箇所についてのべる。

(1) 経路決定アルゴリズムの修正

5. 3. 1節の経路決定アルゴリズムを次のように修正すればよい。

最大走行完了時刻を最小化するためには、最大走行完了時刻をもつ台車が、出発ノードから目的ノードを経由する経路の内、走行時間が最小になるような経路を通ることが最も望ましい。従って、(STEP1)で各台車の代替経路として走行時間の点から第K最小までのK個の基本経路を作成することは、この評価基準でも望ましい。また、(STEP3)における t_g を、基本経路のすべての組み合わせの内、待ち停止時刻を考慮しない場合の第g最小の最大走行完了時刻とする。また、その時の基本経路の組合せを R_g とする。さらに、(STEP5)における t'_g を、決定した待ち停止時間を含めた最大走行完了時刻を意味するものとする。

(2) 待ち停止時間決定問題の修正

(STEP5)における式(5-1)を、最大走行完了時刻を最小化する目的関数で修正しなければならない。すなわち、最大走行完了時刻を表す変数Gを用いて、次式で定式化できる。

$$\text{minimize } G \quad (5-11)$$

また、各台車の走行完了時刻に関しては、次の制約式が成立しなければな

らない。

$$\sum_{i=1}^{M_q-1} (x_{qi} + d_{n_{qi}n_{qi+1}}) \leq G, \quad q=1,2,\dots,n \quad (5-12)$$

以上から、待ち停止時間決定のための線形計画問題としては、5. 3. 2 節の(5-1)式を(5-11)式で置き換え、制約式を(5-2)～(5-7)式及び(5-12)式とすることにより、最大走行完了時刻を最小化する待ち停止時間を求めることができる。

5. 6 ヒューリスティックな干渉回避法

5. 2～5. 4 節では、待機中の台車と走行中の台車の混在する搬送システムの稼働状況に対して三つの経路決定法（完全固定法、経路固定法、更新法）を提案した。

現実の搬送要求に対しては、要求への即応性、台車の稼働率等の面から、可能な限り迅速に効率的な干渉回避経路を求めることが要求される。干渉回避経路を決定できる可能性が大きいという点では、更新法が他の方法に比べて常に優れている。しかし、台車数が増加したり、干渉の形態が複雑になったりすると、更新法によっても経路の求まらない場合がある。経路決定の可能性を増すためには、台車間の干渉が基本的にある台車の停止による他の台車の通行妨害及びそれらの相互作用によって発生することを認識し、そこに注目した干渉回避法を考えることが必要となる。

この節では、更新法によっても干渉回避経路が求まらない非同時走行問題に対し、その決定不可能性の原因を干渉形態の観点を含めて考察し、更新法による経路決定可能性を改善するヒューリスティックな三つの方法を提案する。また、これらの方法を適用した数値例を考えることにより、その有効性について考察する。また、ここで考える経路決定問題の前提条件は、すべて5. 2 節における非同時走行問題に同じであるとする。

5. 6. 1 非同時走行問題の解答不能性の原因と対策

非同時走行問題に対して固定法だけでなく更新法のアルゴリズムでも解の求まらない（解答不能な）場合が存在するが、その原因は大きく次のように分類できる。ここでは、それらを回避するためのいくつかのヒューリスティ

ックな対策もあわせて考察する。

(a) その問題に本来実行可能解が存在しない場合

図5. 11は、ノード1とノード3に台車A, Bが停止中である時、ノード2に存在する台車Cに対し他のノードへの搬送要求が与えられた例である。この搬送要求を満たす走行経路は存在しない。このような状況は、あるノードでの台車の停止が他の台車の通過を禁止することにより生じる。このような状況はデッドロックと呼ばれるが、このようにある台車の通過が禁止されるノードを以後この台車のボトルネックノードと呼ぶ。

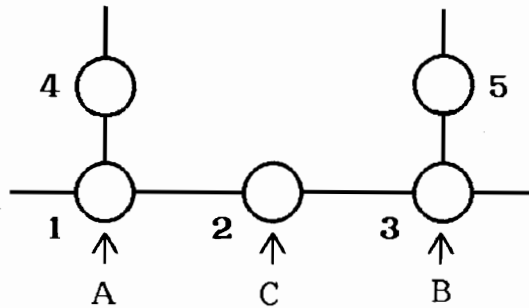


図5. 11 デッドロック

工場における台車は、本来、ワークや治工具等の搬送を目的としており、台車が工作機械のある地点やロード／アンロードステーション等で停止するのは、搬送のためにこれらの搬送物の積み込み、積み卸し等を行うためである。目的ノードでこれ以外の状態にあり、次の搬送要求を待ち受けて無駄な遊休時間をもつような場合には、ノードでの停止によって他の台車のボトルネックノードを作るような状況は可能な限り回避することが望ましい。回避するための一つの方法は、停止台車が待避行動を行い、一時的にデッドロックを解消してやることである。図5. 11において、停止している台車Aがノード4へ、または台車Bがノード5へ待避した後戻ってくることにより、一時的に台車Cのデッドロックを解くことが可能である。この方法を「停止ノードからの一時的待避法」と呼ぶ。

(b) 正数K値の有限性による場合

非同時走行問題に実行可能な経路が存在する場合、K値が無限であれば、各台車についてその(残り)目的ノードを経由するすべての走行ノード列が

基本経路として得られるはずである。従って、必ず実行可能な走行ノード列の組合せが求められる。しかし、アルゴリズムを実際の計算機上で実行する場合、記憶容量の制約からK値を有限に設定しなければならない。このため、走行ノード列の組合せ空間が実行可能解を含むに到らず、実行可能解が存在するにもかかわらずこれが得られない場合がある。

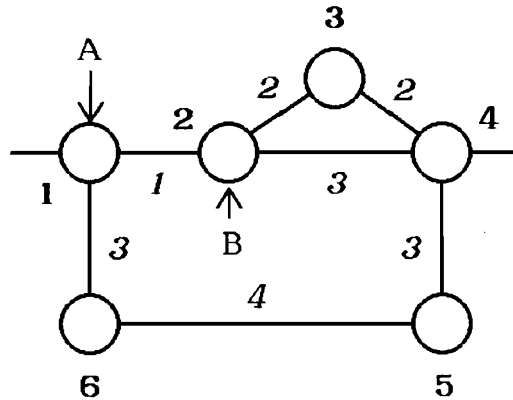


図 5 . 1 2 基本経路の探索

図 5 . 1 2 はそのような例である。アークの横の数字は、そのアークの走行時間を表す。台車 A はノード 1 に、台車 B はノード 2 にそれぞれ停止中であり、台車 A にノード 4 への搬送要求が与えられたとする。この時、 $K=2$ ならば、台車 A のノード 1 から目的ノード 4 への基本経路はノード列 $\{1, 2, 4\}$ 及び $\{1, 2, 3, 4\}$ となりパス $1-6-5-4$ を通る走行ノード列は含まないため、実行可能な走行ノード列は見いだせない。

この例は、台車 A のボトルネックノード 2 を含まない基本経路を見つけることができなかつたために解答不能となつた例である。更新法のアルゴリズムで採用している第 k 最短経路探索は、評価尺度としてアークでの走行時間を用いている。記憶容量の制約から K 値が固定されたものとした場合、基本経路となる走行ノード列を探索してゆくこの走行時間方式を変えることにより、ボトルネックノードを回避できる基本経路を見つけられる可能性がある。

その第 1 の方法は、評価尺度として走行ノード列のノード数を用いることである。この方法は、走行時間とノード数が比例するように設定された搬送ネットワークでは、従来の第 k 最短経路探索法と全く同じものになる。しか

し、実際の搬送ネットワークの設定では、台車の幾何学的な動きから搬送路の分岐点や台車の停止を必要とする地点にのみノードを設け、直接には走行時間とノード数が比例しないような場合が多い。このような場合には、上述した評価尺度によって、従来の走行時間尺度で求めた基本経路とは異なったものが見つけられると期待できる。この方法を「ノード数を評価尺度とした基本経路探索法」と呼ぶ。

第2の方法は、ある台車の基本経路を探索する際、その台車のボトルネックノードを推定し、それを除去した搬送ネットワーク上で第 k 最短経路を求めることである。この方法を「ボトルネックノードを除去する基本経路探索法」と呼ぶ。この方法では、ボトルネックノードを正確に推定することが重要である。もしこの推定が不正確で、誤って実行可能な経路上のノードをボトルネックノードに指定することになると、この実行可能な基本経路の探索が不可能となる。また、ある台車に対する推定ボトルネックノードを増やすと、その台車にとってネットワークの連結性が希薄になり、基本経路を探索する上での柔軟性に欠ける結果になる。

5. 6. 2 ヒューリスティックな干渉回避法

非同時走行問題に対する更新法の解答不能性を回避する対策は、そのまま台車間で干渉のない実行可能経路を探索する方法として応用できる。以下では、更新法で解答不能となった問題に対し、ヒューリスティックな三つの干渉回避法を考え、それを5. 3. 1節の経路決定アルゴリズムへ組み込む方法について述べる。

(1) 停止ノードからの一時的待避法

複数の台車の基本経路間の共通ノード（アーク）列を、それらの台車の共有パスと呼ぶ。台車の基本経路から、どの台車がある共有パスを先行すべきか判断できる場合がある。図5. 13(a), (b)はその例を示したものである。いずれの例でも、台車Aが共有パスを先行しなければならない。この制約によって、実行可能な基本経路の組合せも制約を受ける。このような制約により基本経路の走行ノード列の組合せ空間が実行可能な部分集合をもたない場合、問題は解答不能となる。

最終の停止ノードから一時的に待避させる台車としては、このような制約に多くかかわる台車を選ぶのが適当である。5. 3. 1節のアルゴリズムの(STEP5)において、基本経路の組 R_g に対して図5. 13(a)のパターンが検出

される毎に台車 B に付加した計数（障害度計数）を 0 から 1 ポイントずつ加算する。そして、問題が解答不能となった時には、まず障害度計数が正である台車を見つける。そして、その台車の基本経路を次のように作成し直す。すなわち、(STEP1)の(a), (b), (c)の手順中で、その（残り）目的ノード列の最終ノードから一旦待避した後再び戻ってくるようなループをもつ第 K 最短経路を探索することにより基本経路を作成する。その後、(STEP2)から再実行させる。ここでは、最終ノードにおけるループの作成は各台車に対して高々 1 回を限度とし、正の障害度計数をもつ台車が存在しない場合または実行可能解が求まった時アルゴリズムの実行を終了させる。

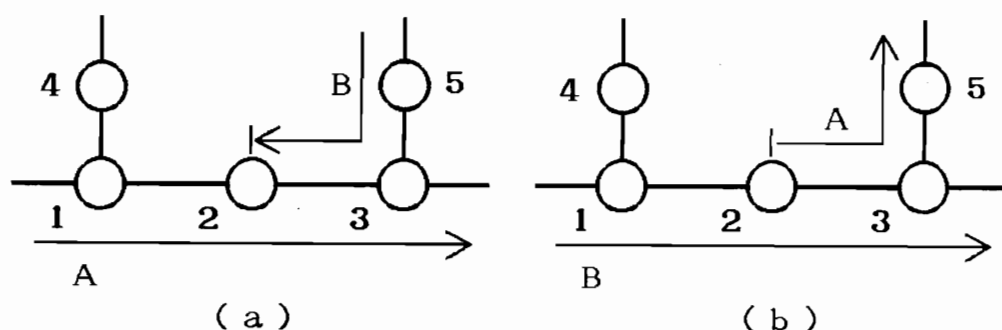


図 5. 1 3 共有バス

(2) ノード数を評価尺度とする基本経路探索法

非同時走行問題が更新法によっても解答不能となった時には、基本経路の作成の際ノード数を評価尺度とする第 k 最短経路探索法を用いるものとし、5. 3. 1 節のアルゴリズムの (STEP1) から再実行させる。但し、実行可能解が得られた場合はその時点でアルゴリズムの実行を終了する。

(3) ボトルネックノードを除去する基本経路探索法

除去すべきボトルネックノードの推定は次のように行う。5. 3. 1 節のアルゴリズムの (STEP5) において、基本経路の組 R_g に対して図 5. 1 3 (a) のパターンが検出される毎に、組（台車 A, ノード 3）に付加した計数（ネック計数）を 0 から 1 ポイントずつ加算し、図 5. 1 3 (b) のパターンの場合は組（台車 B, ノード 2）の計数を加算する。問題が解答不能となった時には、ネック計数が最大となった組の台車及びノードによって 1 つの台車とそのボト

ルネックノードを推定する。そして、このノードを除去したネットワークを基にして、選択した1つの台車の基本経路を作成した後、(STEP2)から再実行させる。5.6.1節(b)で述べたように、ボトルネックノードを増加させると基本経路探索の柔軟性が欠如するので、ボトルネックノードの選択は各台車に対し多くて1回とし、正のネック計数をもつ台車が存在しないかまたは実行可能解が求まった時点でアルゴリズムの実行を終了する。

(4) 三つの干渉回避法を組み合わせた方法

以後、上述した「停止ノードからの一時的待避法」, 「ノード数を評価尺度とする基本経路探索法」, 「ボトルネックノードを除去する基本経路探索法」をそれぞれ第1法, 第2法, 第3法と呼ぶ。三つの干渉回避法を単独で使用する場合の他に、これらを組み合わせた方法も考えられる。但し、次の点に留意する必要がある。第1法, 第3法では、台車またはノードを選択する際に、前問題を解く過程で得られた障害度計数及びネック計数という統計量を用いるのに対し、第2法ではその必要はない。

5.6.3 三つの干渉回避法の特徴

ここでは三つのヒューリスティックな干渉回避法(第1法~第3法)の定性的な特徴を、具体例を用いることにより明らかにする。

(1) 搬送系モデル

ここで使用する搬送ネットワークモデルは、図5.14で示したノード数12(アーク数13)をもつネットワークAである。

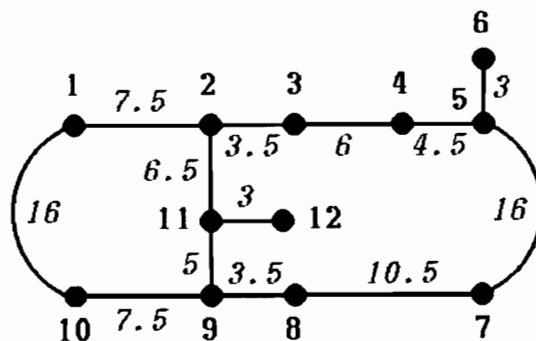


図5.14 ネットワーク A

図中のノードの横の数字はノード番号を、アークの横の数字はそのアークの走行時間を示す。また、最小干渉許容時間を $\epsilon = 1.0$ 、基本経路の候補経路数を $K=5$ とする。搬送要求の優先度はすべて 1.0 とする。以下の例では、台車数はすべて5台であるとする。

(2) 停止ノードからの一時的待避法 (第1法)

表5. 5 (a)は、時刻 t_0 での全台車の位置とそれ以降の残りパスの一例をノード列で示したものである。ノード番号の横の () 中の数字は、そのノードでの待ち停止時間を表す。但し、待ち停止時間が0の場合は省略してある。また、負の数字は、台車が t_0 時点でそのノードから数字の絶対値で示される時間距離だけアーク上を走行した地点にいることを表す。右肩に * 印のついたノードは残り目的ノードである。

表5. 5 第1法の適用例

(a) 残りパス

台車	残りパス
A	8(waiting)
B	10(-14.5)-1*
C	7(-4)-5-4*
D	5(7)-6-5-7*
E	2(-3)-3*

(b) 走行経路

台車	走行経路
A	8-7-5-6-5
B	10(-14.5)-1*
C	7(-4)-5-4-3-4*
D	5(7)-6-5-4(5.5)-5-7*
E	2(-3)-3(16)-2-3*

この時、ノード8で待機中の台車Aに対し、目的ノード5への搬送要求が生じた場合の経路決定問題を第1法を組み込んだ更新法アルゴリズムで解いたところ、表5. 5 (b)のような走行経路が得られた。5. 2. 1節で述べたように、更新法アルゴリズムでは走行中台車の残りパスは固定しない。走行中台車B~Eのうち表5. 5 (a)の残りパスと同じ走行経路が得られたのは、他の台車の走行経路と共有パスを有しない台車Bだけであり、他の走行中台車では、干渉回避のため複雑で巧妙な走行経路に修正されている。台車Dは、一時的にノード6に待避することにより、台車Cがノード5から4へ通過することを許す。台車Dがノード7へ向い、台車Aがそれとは逆方向にノード5へ向かうことを可能とするため、台車Aはノード6で、台車Dはノード4でお互いを待ち合わせる。さらに、台車Dのノード4での待ち合わせを可能とするため、台車Cは残り目的ノード4に到着した後ノード3へ一時的に待

避する。さらに、この台車Cの待避を可能とするため、台車Eはノード2に一時的待避をした後残り目的ノード3に戻る。

この数値例は、第1法を用いて得られた台車の一時的待避が巧妙に組み合わせられることによって実行可能解が得られた例である。なお、この問題に対して第2法または第3法を使用しても解は得られない。

(3) ノード数を評価尺度とする基本経路探索法 (第2法)

表5.6(a)は、 t_0 時点での全台車の残りパスの一例を示している。ノード12で待機中の台車Aに与えられた目的ノード5への搬送要求に対し、第2法を組み込んだ更新法アルゴリズムを用いて解くと、走行経路は表5.6(b)のようになる。台車Aがパス12-11-2-3-4-5を通過することを可能にするため、台車Bが残り目的ノードでないノード2からノード1へ一時的に待避する。さらに、台車Bの一時的待避を可能にするため、台車Eがノード1からノード10へ一時的待避を行う。

表5.6 第2法の適用例

(a) 残りパス

(b) 走行経路

台車	残りパス	台車	走行経路
A	12(waiting)	A	12-11-2-3-4-5
B	2-3-4*	B	2-1-2-3-4*
C	7-8-9-11-2*	C	7(15)-8-9-11-2*
D	6-5-7-8*	D	6-5-7-8*
E	1-2-3*	E	1-10-1-2-3*

この例は、(残り)目的ノードでないノードからの一時的待避が有効である場合である。第2法ではノード数を基本経路探索の評価尺度とするため、走行時間が大きいにも関わらずアーク(1,10)、(1,2)が台車E、Bの待避用のパスとして採用され実行可能解が得られた。また、評価尺度がノード数であれば一時的待避行動はアークの走行時間に関わりなく、さらに、その行動の起点が(残り)目的ノードか否かに関わりなく可能である。なお、この問題に対し(残り)目的ノードからの待避を主眼とした第1法または第3法を使用しても実行可能解は得られない。

(4) ボトルネックノードを除去する基本経路探索法 (第3法)

t_0 時点での全台車の残りパスの一例及びその時ノード3で待機中の台車A

に与えられた目的ノード5への搬送要求に対し第3法を組み込んだ更新法により求めた走行経路は、それぞれ表5.7(a), (b)のとおりである。台車Aは、ノード3から大きく迂回するようにパス3-2-11-9-8-7-5を通してノード5へ到達する。また、台車C及び台車Dは、台車Aを通過させるために一時的な待避行動を行う。

表5.7 第3法の適用例

(a) 残りパス

台車	残りパス
A	3(waiting)
B	11(-1)-12-11*
C	9(-1)-11-2*
D	9-8*
E	3(-2)-4*

(b) 走行経路

台車	走行経路
A	3(8)-2-11-9-8-7-5
B	11(-1)-12(14)-11*
C	9(-1)-11-2-1-2*
D	9(9)-10-9-8*
E	3(-2)-4*

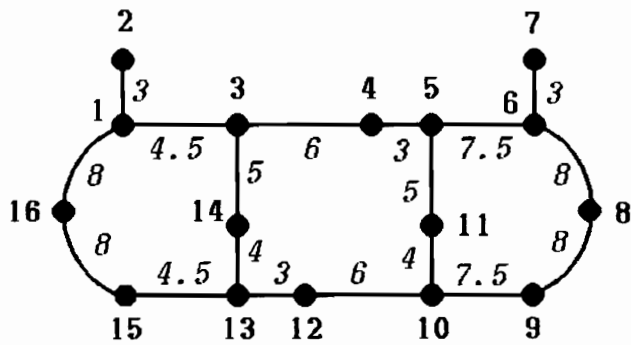
この例は、ノード4が台車Aのボトルネックノードと推定され、出発ノードから目的ノードへ大きく迂回する基本経路が探索されたことにより実行可能解が得られた例である。第1法、第2法を用いて解こうとした場合、基本経路の探索は、走行時間が小さくかつノード数の少ないパス3-4-5近辺の経路探索に終始し、結局実行可能解を得ることはできない。

5.6.4 数値例

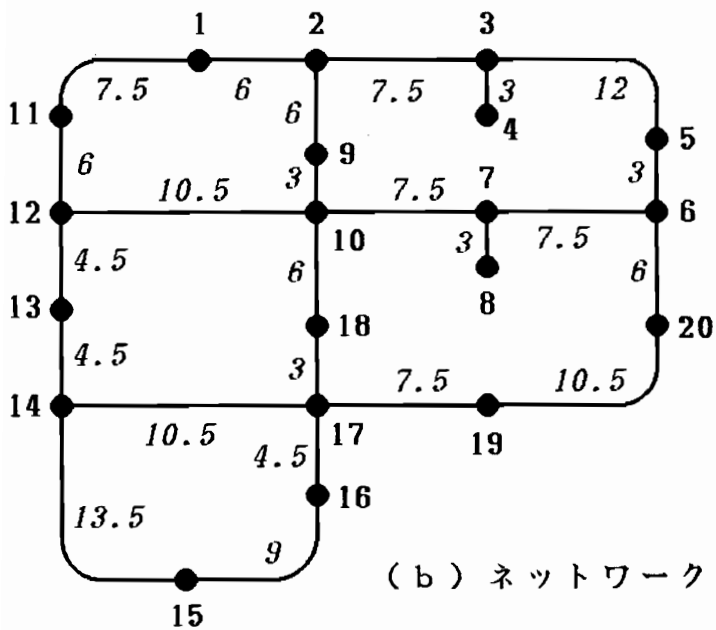
解答不能率を全問題数に対する解答不能となった問題数の割合で定義する。ここでは、従来の更新法による解答不能率が、第1法～第3法を用いることによりどの程度改善されるかを、簡単なシミュレーションを実施して検討する。

(1) 搬送系モデル

ここで用いる搬送系ネットワークは、前節で用いたネットワークAの他、図5.15(a), (b)のネットワークB, Cである。ネットワークB, Cのノード数(アーク数)は、16(18), 20(26)であり、ネットワークの規模は、A, B, Cの順に大きい。ネットワークの規模が大きいほど出発ノードと(残り)目的ノードを結ぶ経路の代替数が多くなることから、同じ台車数では、より大きな規模のネットワークほどデッドロックは生じにくくなることが予想され



(a) ネットワーク B



(b) ネットワーク C

図 5 . 15 ネットワークモデル

る。本シミュレーションにおける最小干渉許容時間，基本経路の候補経路数，搬送要求の優先度は、それぞれ $\varepsilon = 1.0$, $K = 5$, $W_k = 1.0$ ($k = 1, \dots, n$) である。台車数 n は 2 ~ 5 台で変化させる。

(2) シミュレーションの手順

台車数 ($n = 2 \sim 5$) , ネットワーク (ネットワーク A ~ C) の組合せの各々に対して、次のような手順でシミュレーションを行う。

(ア) 全台車に対し、一つの出発ノード、一つの目的ノードをランダムに与える。このような同時走行問題を、文献115)の経路決定アルゴリズムにより解が得られるまで繰り返し作成する。この解を、次の非同時走行問題のための初期状態とする。

(イ) 上の手順で得られた台車の走行状態において、搬送を最も早く終了した台車に対し、新たに一つの目的ノードをランダムに与え、その非同時走行問題を更新法によって解く。

(ウ) 問題が解答不能となった場合には、同じ問題を第1法、第2法及び第3法を用いてそれぞれに解く。

(エ) 上の(ア)(イ)(ウ)の手順を100回繰り返す。

(3) シミュレーション結果

表5. 8 (a), (b), (c)は、ネットワークA, B, Cに対するシミュレーション結果を解答不能率 (全問題数100に対する解答不能であった問題数の割合) で表したものである。

表中には、比較の便宜上、前述したシミュレーション手順の(ウ)を省いた更新法のみによる解答不能率及び第1法、第2法を組合せた方法による解答不能率も併記している。但し、組合せ方は次のように行った。更新法の次に第1法、第2法をそれぞれ単独に用いる。さらに、第1法と第2法を同時に用いたアルゴリズム、すなわち評価尺度をノード数とし、(残り)目的ノードからの待避のある基本経路を探索する更新法アルゴリズムを実行する。もちろん、アルゴリズムのどの実行段階にあっても、実行可能解が得られた時点で実行を終了するものとする。

表5. 8より、解答不能率の点から次のことがいえる。

(ア) 三つの干渉回避法 (第1法 ~ 第3法) の間に明確な優劣はない。

(イ) 干渉回避法を組み合わせることにより、各方法を単独で用いた場合よりもかなり大きな解答不能率の改善効果がみられる。これは、5. 6. 3節で述べたように、各方法の適用が有効となる干渉形態が異なっており、これ

らの方法を組み合わせることによって、種々の干渉形態に対し有効な回避法となったためであると考えられる。

表5. 8 解答不能率

(a) ネットワーク A

台車数	更新法	第1法	第2法	第3法	1,2法の組合せ
2	0.07	0.03	0.03	0.00	0.02
3	0.13	0.11	0.09	0.06	0.04
4	0.16	0.13	0.07	0.09	0.03
5	0.24	0.21	0.14	0.15	0.06

(b) ネットワーク B

台車数	更新法	第1法	第2法	第3法	1,2法の組合せ
2	0.00	0.00	0.00	0.00	0.00
3	0.05	0.02	0.04	0.00	0.01
4	0.10	0.05	0.05	0.03	0.00
5	0.09	0.04	0.08	0.05	0.02

(c) ネットワーク C

台車数	更新法	第1法	第2法	第3法	1,2法の組合せ
2	0.03	0.01	0.02	0.00	0.00
3	0.05	0.03	0.02	0.02	0.00
4	0.03	0.01	0.01	0.00	0.00
5	0.10	0.09	0.03	0.04	0.02

5. 7 結 言

この章では、複数台車に対する経路決定法について考察した。

5. 2節では、評価尺度を搬送要求の優先度の重みを付けた総走行時間とした場合の経路決定問題を非同時走行問題と呼ぶ問題にモデル化した。この問題に対し、5. 3節では、搬送要求の発生時点で、目的地へ走行中の台車の経路情報をそのまま用いるか、部分的に用いるか、全く使用しないかによ

って、三つの経路決定法（完全固定法，経路固定法，更新法）を提案した。5. 4節では、これらの経路決定法の特徴を、解答不能率、解の改善度及び計算所要時間の観点から各種シミュレーション結果を用いて明らかにした。その概要は次のとおりである。

（1）完全固定法は短時間で解が得られるという長所があるが、複雑な干渉形態の予想される場合には実行可能解が得られない場合も多い。

（2）更新法は、複雑な干渉形態の予想されるような場合でも解の取得が十分期待できる。しかし、台車数の多い場合には、最適解を求めようとすると計算所要時間がかかなり大きくなる。但し、初期実行可能解を近似解とすれば、計算時間は大きく短縮され、しかもその近似解は十分に実用的であるといえる。

（3）経路固定法は、完全固定法と更新法の中間的性質を有する。また、これらの特徴を活かした現実的な経路決定手順についての考察も加えた。

5. 5節では、5. 3節で提案した経路決定法を、最大走行完了時刻最小化を評価尺度とした場合の問題に応用する方法について考察した。

5. 6節では、更新法でさえも実行可能な経路を見いだせない問題に対し、干渉回避経路探索のための三つのヒューリスティックな方法（第1法、第2法及び第3法）を提案し、シミュレーションを用いてその有効性について考察した。その結果、それらの有効性の特徴について以下のような知見を得た。

（a）第1法「停止ノードからの一時的待避法」は、ある台車の最終目的地での停止が他の台車の通過の障害となり、停止台車の一時的な待避が有効である場合に適した方法である。

（b）第2法「ノード数を評価尺度とする基本経路探索法」は、ある台車の走行途中でのノードからの一時的待避が他の台車の通過にとって有効である場合に適した方法である。

（c）第3法「ボトルネックノードを除去する基本経路探索法」は、ある台車にとって出発ノードから目的ノードへ大きく迂回する経路への走行が有効である場合に適した方法である。

以上のような各方法のもつ固有の有効性のほか、解答不能率の点では、これらの方法を組合せることがさらに有効であるというシミュレーション結果を得た。

第6章 搬送要求の割当て問題

6.1 緒言

前章までは、搬送要求の台車に対する割り当てがすでに決まったものと仮定して、要求に対する最適経路の導出について考えた。ここでは、搬送要求の台車への割当て問題について考察する。2.6節で述べたように、複数のビークルをもつ一般的な経路・スケジュール決定問題においては、割当て問題も同時に解かなければならない部分問題として、本来その中に含まれている。しかし、多くの大規模な経路・スケジュール決定問題においては、実際的な計算時間で実行可能解を導出しようとする場合には、問題の複雑さゆえに、問題を二つの部分問題、すなわち、割当て問題とビークルの割当てが確定した場合の経路・スケジュール決定問題の二つに階層的に分割して解く方法が多く採用されており、無人搬送車の運行問題においても事情は同じである。以下で提案する搬送要求の台車への割当て方法は、その大部分がこの分割手法を用いており、従って、全体としての最適性が必ずしも保証されていないことは注意を要する。また、それらの分割算法では、まず搬送要求を台車に割り当てた後、前章までに提案した経路決定法により、それらの要求を満たす最適な経路を求めるといった計算手順を踏むことになる。

ここでは、以下のような状況のもとにおいて、ある時刻に同時に生じた複数の搬送要求を台車に割り当てる問題について考察する。

P 台の台車 V_1, V_2, \dots, V_P が存在している。台車 V_q はノード s_q へ運行中であり、 s_q への到着時刻 r_q までは他の搬送要求には応じられないが、到着の後は空きとなる。いま、 M 個の搬送要求 J_1, J_2, \dots, J_M が生じたとする。要求 J_k は、ワークの積み込みノード a_k 、ワークの加工順序に従って経由すべき加工機械のあるノード列 L_k 及びそれぞれの機械での加工時間、ワークの荷降しノード b_k から成る。ノード a_k から L_k を経由し、ノード b_k に到達する最短経路での走行時間に加工時間を加えた時間を t_k とする。この時、すでに空きとなっている台車ばかりでなく現在搬送中のものも含めて、各要求をどの台車に割り当てればよいかを決定することが問題となる。

この基本問題に対し、台車が複数のワークを同時に積載可能であるような場合には、ある要求のワークを積み込んだ後別の要求のワークを積んで、その後それぞれのワークの荷降ろしを行うというように、荷降ろしは積み込み

の後にいうという先行関係さえ満足すれば、複数要求の a_k, L_k, b_k を混合して経由することが可能である。これに対し、台車が一個積みであるような場合には、ある要求に含まれる a_k, L_k, b_k をすべて経由した後に他の要求を受けつけるという手順をとることになる。多くの無人搬送車の積載形式は、パレット上に固定したワークをパレットごと積み降ろしする一個積みの形式が多く、この章では一個積みの場合の基本問題について考察する。また、問題において最適化すべき評価尺度として、代表的な二つの尺度、総走行完了時刻最小化及び最大走行完了時刻最小化を考える。

6. 2 重複を許さない割当て問題

ここでは、前節で述べた割当ての基本問題において、各台車における搬送の負荷や可動率を平均化するため、一台の台車 V_q に割り当てる要求は高々一つの要求 J_k のみであり、複数要求を重複して割り当てることはしない場合を考える。

(1) 完全列挙法

M 個の搬送要求を P 台の台車へ割当ててる場合の組合せ総数は ${}^P C_M \cdot M!$ であり、もしこの値がそれほど大きくない場合には、次のような全数列挙的な手順により、最適な割当て及び経路を求めることができる。

i) 各割当てに対し、第3～5章で提案した経路決定法を用いて最適経路及び最適な目的関数値（総走行時間または走行完了時刻）を求める。

ii) すべての割当てのなかで最適な目的関数値の得られた割当て、経路を最適解とする。

この単純な完全列挙法は、搬送要求の割当てと経路の最適性を同時に達成できる方法であるが、予想される計算時間 ${}^P C_M \cdot M! \times$ （経路決定問題一題当りの求解に要する計算時間）が実用上差し支えない程度となることを前提とするものであり、小規模の問題にのみ適用可能である。

(2) 総走行完了時刻最小化問題

搬送要求 J_k を台車 V_q に割り当てた場合の走行完了時刻 w_{kq} は、その経路が最短経路であると仮定すると、

$$w_{kq} = r_q + \mu(s_q, a_k) + t_k \quad (6-1)$$

となる。但し、 $\mu(i, j)$ は、ノード i から j への最短走行時間とする。

この仮定の下で総走行完了時刻を最小化する割当て問題は、一般の割当て問題¹⁰²⁾¹⁰⁴⁾¹⁰⁵⁾と同様に次のように定式化できる。

$$\underset{\tilde{x}}{\text{minimize}} \quad f(\tilde{x}, \tilde{w}) = \underset{\tilde{x}}{\text{min}} \sum_{k=1}^M \sum_{q=1}^P w_{kq} \cdot x_{kq} \quad (6-2)$$

subject to

$$x_{kq} = \begin{cases} 1 \\ 0 \end{cases}, \quad 1 \leq k \leq M, \quad 1 \leq q \leq P \quad (6-3)$$

$$\sum_{q=1}^P x_{kq} = 1, \quad 1 \leq k \leq M \quad (6-4)$$

$$\sum_{k=1}^M x_{kq} \leq 1, \quad 1 \leq q \leq P \quad (6-5)$$

(6-2)式の変数 \tilde{x} 、定数 \tilde{w} は、それぞれベクトル $\{x_{kq}: 1 \leq k \leq M, 1 \leq q \leq P\}$ 、 $\{w_{kq}: 1 \leq k \leq M, 1 \leq q \leq P\}$ を表すものとする。 x_{kq} は、要求 k が台車 V_q に割り当てられるとき1をとり、そうでないとき0をとるような変数である。

この問題は、グラフ論では、2部グラフにおける重み付きマッチング問題(The Weighted Matching Problem)¹⁰²⁾¹¹⁷⁾や最大流最小コストフロー問題(The Minimum Cost Flow Problem)¹⁰²⁾¹⁰⁷⁾¹¹⁸⁾に置き換えることができる。

問題を重み付きマッチング問題に置き換えると、図6. 1のような搬送要求を示すノード群と台車を示すノード群とが2部グラフを構成し、要求 k のノードと台車 q のノードを結ぶアークの重みが w_{kq} であるとき、その中に含まれるアークの重みの総和が最小となるマッチングを求める問題になる。ちなみに、マッチングとは、それに含まれるどの二つのアークも同一ノードを共有しないようなアークの集合をいう。この問題に対しては、ハンガリアンメソッドとよばれる方法¹¹⁹⁾¹²⁰⁾により、計算量がオーダー $O(M^3 \cdot P)$ である解法が知られている。問題を最大流最小コストフロー問題に置き換えると、図6. 2のように、アークの最大容量が1であり、アーク (J_k, V_q) に流れる単位フロー当りのコストが w_{kq} であるネットワークを考え、ソースノード s からシンクノード t への最大流量で最小コストとなるような流れを求めるフロー問題になる。図6. 2のアークの横に記されている数字は、(最大

容量, 単位フロー当りのコスト) を表している。この問題は、Klein¹²¹⁾の考
 えを用いた解法アルゴリズムにより、オーダー $O(M \cdot (M+P)^3)$ の計算時間
 で解を求めることができる¹⁰²⁾。

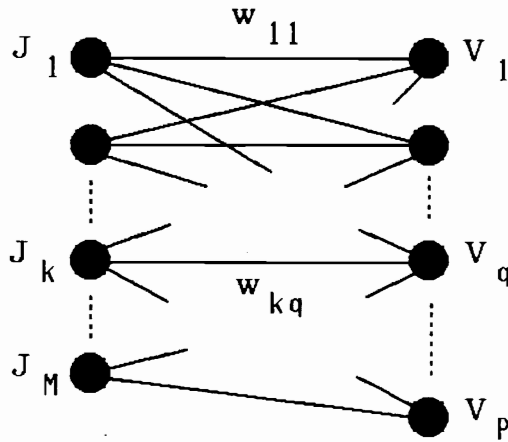


図 6 . 1 重み付きマッチング問題

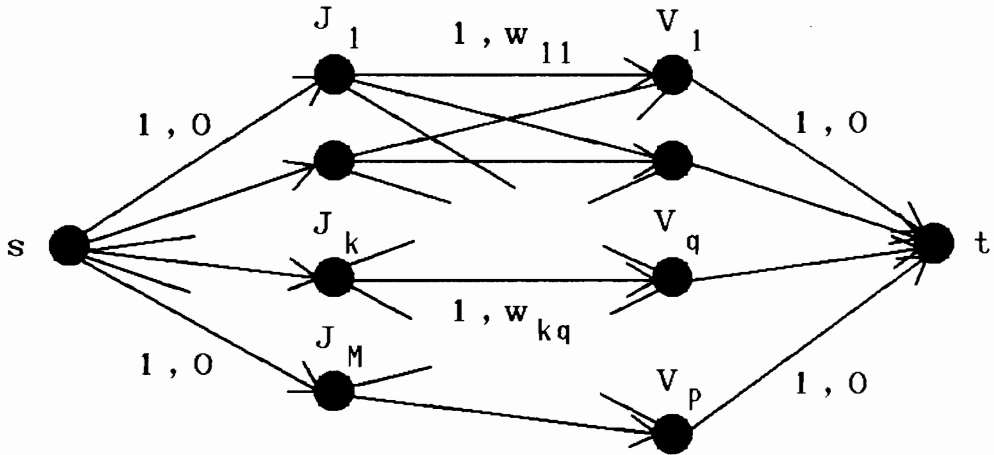


図 6 . 2 最大流最小コスト問題

(3) 最大走行完了時刻最小化問題

6. 1節の前提の下で、搬送要求 J_k を台車 V_q へ割り当てた場合の走行完了時刻は(6-1)式の w_{kq} で表され、要求割当てのない台車 V_q のそれは r_q となる。従って、最大走行完了時刻最小化問題では、次式を最適化することになる。

$$\begin{aligned} \underset{\tilde{x}}{\text{minimize}} \quad & g(\tilde{x}, \tilde{w}, \tilde{r}) \\ & = \min_{\tilde{x}} \max_{1 \leq q \leq P} \{ r_q + (w_{kq} - r_q) \cdot x_{kq} \} \end{aligned} \quad (6-6)$$

但し、 \tilde{r} はベクトル $\{r_q : 1 \leq q \leq P\}$ を表す。また、変数 x_{kq} の満たすべき制約式は、(6-3)~(6-4)と同じである。

この問題は、図6. 3で示されるMin-Maxマッチング問題¹⁾²⁾ (The Min-Max Matching Problem、またはThe Bottleneck Problemとも呼ばれる)に変換できる。

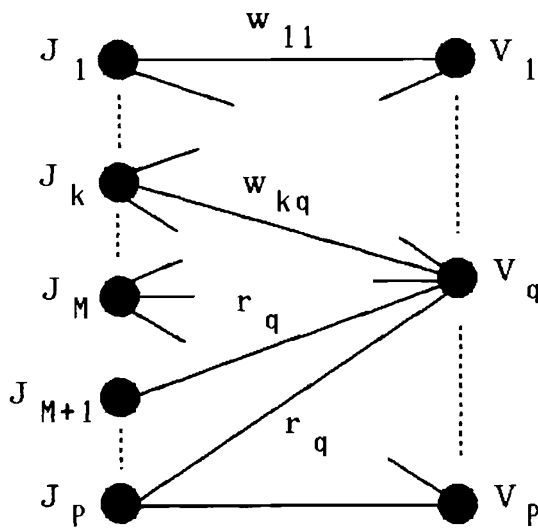


図6. 3 Min-Maxマッチング問題

図中の搬送要求を示すノード J_k と台車を示すノード V_q との間のアークには、重み w_{kq} が付与されている。搬送要求を示すノード群の中にはダミーの

要求 J_{M+1}, \dots, J_P が置かれており、そのノード数を台車のノード群のノード数と同じにしている。その任意のダミーノードと台車 V_q とを結ぶアークは、台車に対し搬送要求を割り当てないことを示すものでその重みは r_q である。Min-Max マッチング問題とは、マッチングに含まれるアーク数が最も多い最大マッチングのうち、それに含まれるアークの重みの最大を最小化するようなマッチングである。この問題に Lawler が提案したしきい値法による解法¹⁰²⁾ を適用すれば、オーダー $O(P^3)$ の計算量で最適解を求めることができる。

6. 3 重複を許す割当て問題

6. 2 節では、一台の台車に割り当てられる搬送要求は、高々一つであるとした。ここでは、台車に複数の搬送要求が割当て可能な場合、すなわち割当てに重複の許される問題について考える。

(1) 完全列挙法

重複の許される場合の完全列挙法も、6. 2 節(1)の方法と同様に、搬送要求の台車に対するすべての割当てを列挙して最適解を求めようとするものである。M 個の要求を P 台の台車に重複を許して割り当てる場合の割当て総数は P^M である。また、ある一台の台車に s 個の要求が割り当てられた場合、台車がこれらの要求に対する搬送サービスを実施する順序付け総数は $s!$ である。以上のすべての組合せに対して、3~5 章で提案した経路決定法によりその最適経路、最適目的関数値を求め、その内最もよい目的関数値（総走行完了時刻または最大走行完了時刻）の得られた割当て、経路を最適解とするものである。この方法は、考慮すべき組み合わせ数が 6. 2 節(1)の完全列挙法よりも多く、また、各台車には複数の搬送要求が与えられるから一つの経路決定問題を解くための計算時間も 6. 2 節(1)よりは多くなると考えられる。従って、この方法は、6. 2 節(1)の完全列挙法の適用よりも、さらに小規模な問題にのみ適用可能である。

(2) 総走行完了時刻最小化問題

台車 V_q が最初に要求 J_k の搬送を行うとした場合の搬送終了時刻は、 $r_q + \mu(s_q, a_k) + t_k$ である。また、要求 J_1 の後に要求 J_k の搬送を行うとして、 J_1 の搬送終了から J_k の搬送終了までに要する時間は、どの台車に対しても $\mu(b_1, a_k) + t_k$ である。搬送要求を示す M 個のノード $J_k (k=1, \dots, M)$ と P 台の台車の出発ノード $s_q (q=1, \dots, P)$ だけから成り、ノード i と j の間のアーク

の長さ $a(i, j)$ が次式で定義されるグラフを考える。

$$a(i, j) = \begin{cases} r_q + \mu(s_q, a_k) + t_k & \text{for } i = s_q, j = J_k \\ \mu(b_1, a_k) + t_k & \text{for } i = J_1, j = J_k \end{cases} \quad (6-7)$$

このとき、総走行完了時刻を最小化する場合の割当て問題は、次のようなグラフ問題に変換できる。すなわち、ループがなくかつ互いに共有するノードがなく、それぞれがノード s_q ($1 \leq q \leq P$) から出る P 本のパスの内、すべてのノードがいずれかのパスに属し、パスに含まれるすべてのアークの長さの和が最小となるパスの組を見つける問題である。各々のパスに含まれるアークの長さの和は、その出発ノードからパスに含まれるノード順に搬送要求を与えられた場合の台車の搬送完了時刻に等しい。従って、すべてのパスについてのそれらの総和は、総走行完了時刻に等しい。台車をセールスマンとし、出発ノードをオフィスとすれば、この問題は、次の点を除いて、必ずしも同じ位置ではない P 個のノード s_q ($q=1, \dots, P$) のそれぞれにいるセールスマンが、要求を示す M 個のノードを手分けして巡回する多重巡回セールスマン問題^{46) 48)}となる。

a) 各台車 (セールスマン) の出発ノードは必ずしも同じでない。

b) 各台車 (セールスマン) は、最後には、割り当てられた搬送要求の最終ノード (最終顧客のいるノード) にとどまり、出発ノードには帰らない。

2. 3節で述べたように、この問題を実時間で解くことの困難さを考えると、ヒューリスティックな方法が重要であり、それには巡回セールスマン問題の研究で提案されている種々の手法が役立つものと思われる。

(3) 最大走行完了時刻最小化問題

最大走行完了時刻最小化する割当て問題に関しても、厳密解法により実時間で解く場合の困難性は前節と同様であり、NP完全な問題となる¹²²⁾。この問題に関しては、Chenら¹²²⁾が分枝限定法を用いた厳密解法を提案している。彼らは、搬送ネットワークのノード間の走行時間が各台車ごとに異なり、目的ノードにおける作業時間が設定されている問題を取り扱っている。しかし、そのネットワークモデルとして三つのノード間での走行時間に三角不等式が成り立つユークリッド的なネットワークを用いており、また、すべての台車は同時に出発し、それぞれの出発ノードに再び帰ってくることを前提としており、実用上の適用範囲に限られる。また、実用的な時間の範囲で計算

が終了することは必ずしも保証されていない。

6. 4 動的割当て法

前節までに述べたことから、割当て問題に対し実計算時間内で厳密解を求めることの困難性がわかる。また、現実の搬送システムにおいては、搬送要求群が離散時間に時系列的に生じる。従って、現実の台車の運行管理にとって、搬送全体の最適化を図るための搬送要求の割当ては、6. 2～6. 3節で述べた一時刻における要求の発生に関する問題よりさらに困難な問題となる。ここでは、時系列的に生じる搬送要求の発生時点ごとに局所的な最適化を図りながら、動的に台車を割り当てていく一つの割当て法を提案する。

まず、工場内における搬送要求の発生の現状とこれまでよく用いられている典型的な割当て方法について概観する。問題において、台車は一個積み台車であると仮定する。

工場内には、在庫・倉庫設備、加工・組立設備、検査設備、入荷・出荷設備等種々の設備が存在し、それらの設備において搬送要求が発生する。例えば、加工設備において、ある加工品の加工が終了し、次工程への搬送を必要とする場合には、まず当該加工機械へ加工品受取のための搬送要求が生じ、台車への加工品搭載後は次の工程までの加工品の搬送要求が発生する。このような搬送要求は、搬送を請け負う空き台車の存在や搬送先設備の受け入れ可能等、要求遂行のための条件が整うまで、台車運行管理システムの搬送要求の待行列内に登録される。この際、なんらかの評価基準や優先度により、待行列内の要求の並び替えが行われ、空き台車があり次第、待行列内から順に要求が取り出され台車に割り当てられる。この時、複数台の空き台車が存在する場合には、ここでもなんらかの基準に基づき一台の台車が選択されることになる。このような要求の待行列における優先順位や台車の引き当ての際の決定規則は、ディスパッチングルールと呼ばれる。

以上のような典型的な割当て法で用いられるディスパッチングルールは、通常、加工機械に対し加工作業を表すジョブを割り当てるスケジューリング問題¹²³⁾においてよく研究されており、ジョブの待行列における優先順位を決定するルールとしては、次のようなものが考案されている¹²⁴⁾⁻¹²⁶⁾。

- (1) F I F O (First In First Out) : 先着優先
- (2) S P T (Shortest Processing Time) : 最小加工時間優先

- (3) LPT(Longest Processing Time): 最長加工時間優先
- (4) LWKR(Least Work Remaining): 残り加工時間最小優先
- (5) MWKR(Most Work Remaining): 残り加工時間最大優先
- (6) EDD(Earliest Due Date): 納期の最も早いもの優先
- (7) Slack: 納期までの余裕最小優先
- (8) Random: ランダムに選択

これらのディスパッチングルールを搬送要求の台車に対する割当て問題に用いた場合の有効性に関する研究も、いくつかなされている²²⁾¹²⁷⁾⁻¹²⁹⁾。

ここでは、上で述べた典型的な割当て法に、一般的な台車の運行管理ではよく用いられる次のようなディスパッチングルール²²⁾¹²⁷⁾¹²⁹⁾を採用したものを考える。

- (a) 搬送要求は、その発生順に待行列から取り出される (FIFO)。
- (b) 複数台の空き台車がある場合は、搬送要求に含まれる目的ノードに最も早く到着できる台車を割り当てる (最短距離優先)。このルールはSPTに相当する。

文献22)では、生産シミュレーションにより検討し、ルール(b)が他のルールに比較して概ね有効であるとの結論を得ている。

以上のような、一般的なディスパッチングルールによる割当て法に対し、6.2節(3)で述べた最大走行完了時刻最小化を評価尺度とした解法を利用して、次のような台車の割当て法を提案する。

- i) 搬送要求が発生した場合、搬送先設備が受け入れ可能であれば、現時点における台車の空き状況を問わず、要求を前もって台車に割り当てることとする。すなわち、台車が必ずしも空きの状態でなくても、その台車をあらかじめ予約することができる。但し、予約は以後変更できないものとする。予約された台車は、現在遂行中の搬送が終わり次第、直ちに予約された要求に対する搬送を開始するものとする。
- ii) 発生した要求の台車への割当ては、6.2節(3)で述べたMin-Maxマッチング問題をLawlerの解法¹⁰²⁾を用いて解くことにより行う。この問題において、台車 q が走行中であれば、ノード s_q は現在搬送中の最終目的ノードであり、 r_q は現時点からノード s_q に到着するまでの走行時間とする。また、台車が待機中であれば、 s_q を待機ノードとし、 r_q をゼロとする。
- iii) 上で決定した割当てに基づき、5章で提案した完全固定法を用いて、各台車の走行経路、走行スケジュールを求める。ここで決定した経路は、以後

変更しない。

iv) 上記i)～iii)の手順を搬送要求が生じるごとに実施する。

この割当ては、搬送要求の発生ごとに動的に台車への割当てを行うことから、動的割当て法と呼ぶ。この割当て法は、時系列的に発生する搬送要求列の中のある時点で発生した要求に対し、将来発生するであろう要求は一切考慮せずに、発生時点だけの局所的な最適化（最大走行完了時刻最小化）を図りながら、全体の最適化を目指すものである。

6.5 数値例

ここでは、前節の(a)，(b)のディスパッチングルールを用いた方法と動的割当て法との有効性に関する比較を、簡単なシミュレーションを用いることにより行う。特に、全搬送要求に対する搬送の完了時刻及び一回の搬送に要する平均時間に対する搬送要求の発生頻度の影響についての感度分析を行う。

6.5.1 シミュレーションの条件と検討項目

(1) モデルの設定

搬送ネットワークモデルとして、36個のノードと38本の双方向走行可能なアークをもつ図6.4のようなネットワークを考える。

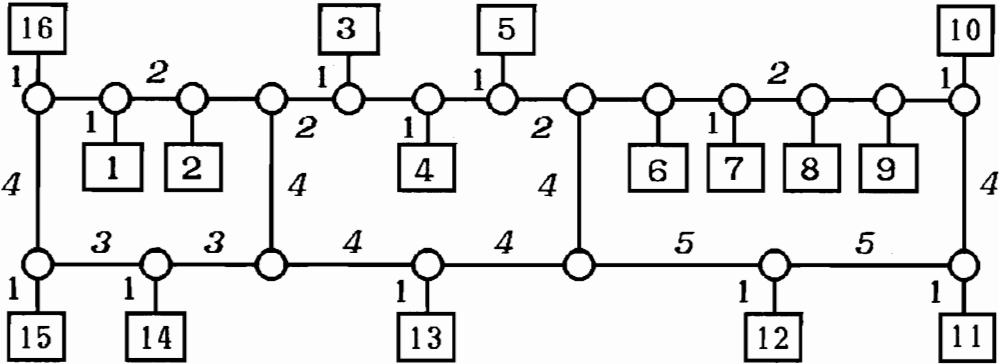


図6.4 搬送ネットワークモデル

アークの横には走行時間が記されている。加工・組立機械、倉庫等はノード1～16（これらを設備ノードとよぶ）に設置しており、ジョブショップ型の加工組立工程を構成している。

台車はこれらのノード間のワークの搬送に従事するものとする。台車間では、最小干渉許容時間 $\epsilon = 0.5$ 以上の時間距離があれば、干渉は起こらないものとする。設備ノードにおいて発生する搬送要求は、単一の搬送要求、すなわち一つの目的ノードをもつ搬送要求のみを考える。また、要求の発生間隔は指数分布に従うものとする。1～3個の搬送要求が同時に発生可能であり、その個数はランダムにとるものとする。各要求に含まれる一つの目的ノードは設備ノードの中からランダムに選択する。

(2) 検討項目

次の二つを検討項目とする。

- ・全搬送要求に対する搬送完了時刻
- ・一回の搬送要求に要する平均搬送時間

(3) シミュレーションの手順

n 台の投入搬送台車は、初期時点でランダムに選ばれた設備ノードのいずれかに待機している。その後、平均発生時間間隔 λ の指数分布に従った搬送要求が発生するものとする。すなわち、搬送要求の発生と発生との間の時間間隔 T は、次の確率に従うものとする。

$$Pr(T \leq t) = 1 - \exp(-t/\lambda) \quad (6-8)$$

また、一度の搬送要求の発生では、(1)項で述べたように1～3個の間でランダムにとられる複数の要求が発生する。発生回数は、合計50回である。

このような搬送要求に対し、一般的なディスパッチングルールによる割当て法（以後、一般的割当て法とよぶ）と動的割当て法を適用する。どちらの方法に対しても、発生する要求は全く同一とする。すなわち、要求の発生時間間隔、各発生時点における搬送要求数、その要求の目的ノードは、どちらの割当て法に対しても同じものを与える。また、一般的割当て法により搬送要求を割り当てられた台車の経路は、動的割当て法と同じく、5章で提案した完全固定法を用いて決定するものとする。このような50回の時系列的な搬送要求発生を一回のシミュレーションとして、これを10回繰り返すことにより、(2)項で述べた二つの検討項目に関する出力を平均値として得る。以上のようなシミュレーションにより、パラメータ n を3, 4台、 λ を2～

20で変えながら、感度分析を実施した。

6. 5. 2 シミュレーションの結果とその分析

図6. 5 (a), (b)は、それぞれ台車数3, 4の場合の搬送要求発生時間間隔（横軸）に対する平均搬送完了時刻（縦軸）を示したものである。“○”は一般的割当て法によるデータを、“△”は動的割当て法によるデータを示している。但し、データの横軸の値は、パラメータ λ ではなく、実際に発生した要求の平均要求発生時間間隔でとっている。

図からわかるとおり、ほとんどの場合、動的割当て法が一般的割当て法より有効であるが、その有効性の特徴として、次のことが指摘できる。

(1) 動的割当て法が有効であるのは、主に要求発生間隔が小さい場合すなわち発生頻度の大きな場合であり、発生間隔が大きく発生頻度の小さな場合には、二つの割当て法の有効性の間には明確な差は認められない。

(2) 要求発生時間間隔が小さい場合には、どちらの割当て法についても、その搬送完了時刻にそれほど変動がない。要求発生時間間隔が大きい場合には、どちらの割当て法についてもその搬送完了時刻に差はなく、しかも、発生間隔が大きくなるにつれ直線的に大きくなる。

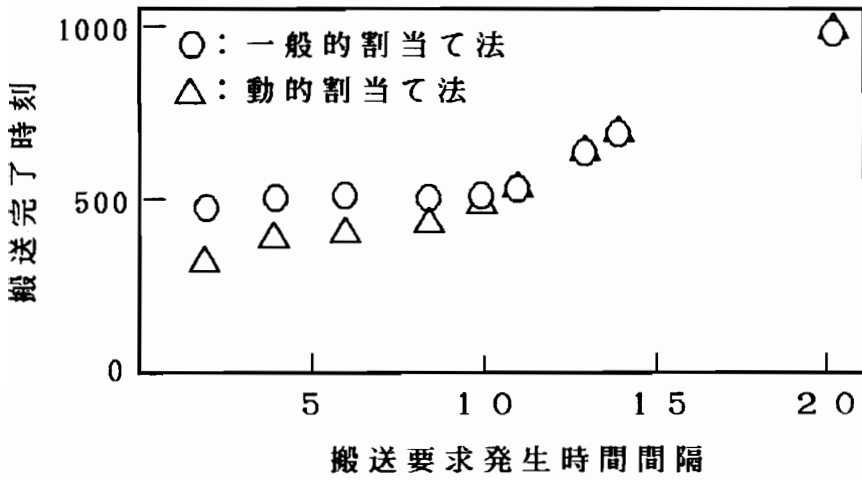
図6. 6 (a), (b)は、それぞれ台車数3, 4の場合の要求発生時間間隔（横軸）に対する台車の一回の搬送に要する平均搬送時間の変化を示したものである。一般的割当て法に比較した動的割当て法の有効性がこの図でも見られ、次のような特徴をもつ。

(a) 動的割当て法による平均搬送時間は、要求発生時間間隔の変化に対し、概ね一定である。

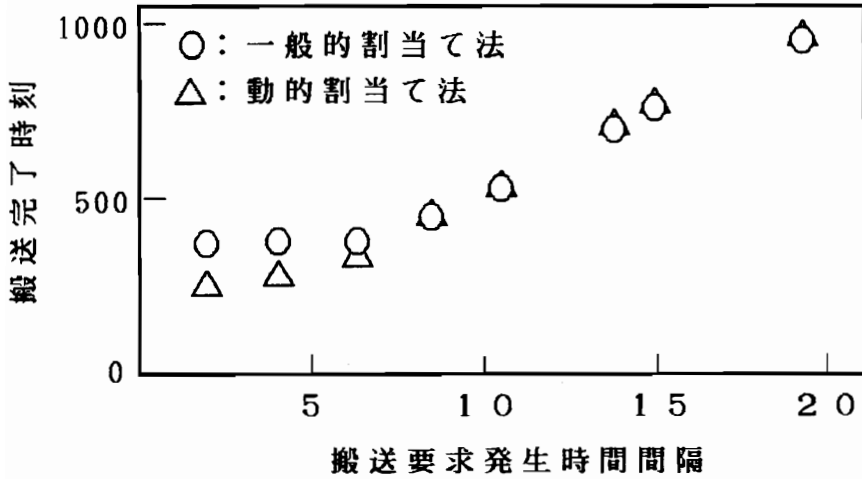
(b) 一般的割当て法による平均搬送時間は、要求発生間隔が小さいほど大きくなる傾向がある。

50回の搬送要求が出尽くすまでは搬送は完了しないから、搬送完了は、全搬送要求の発生と、発生終了時点に待行列に残った搬送要求に対する搬送を完了する作業との二つから成ると考えてよい。従って、搬送完了時刻に対する割当て法の良しあしは、全要求の発生終了時点において残っている要求をできるだけ少なくするように途中の搬送要求を処理できるかという点と、待行列に残った要求を要求発生終了時点以後においてどれだけすばやく処理できるかによると考えられる。

もし、要求の発生間隔が十分大きく、要求の発生と発生との間に要求に対

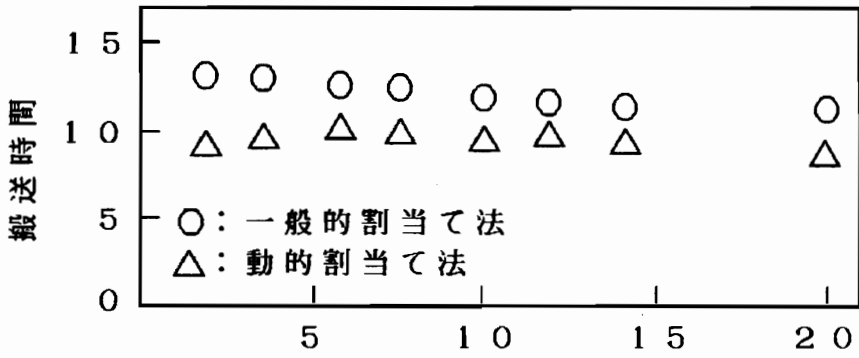


(a) 台車数=3の場合



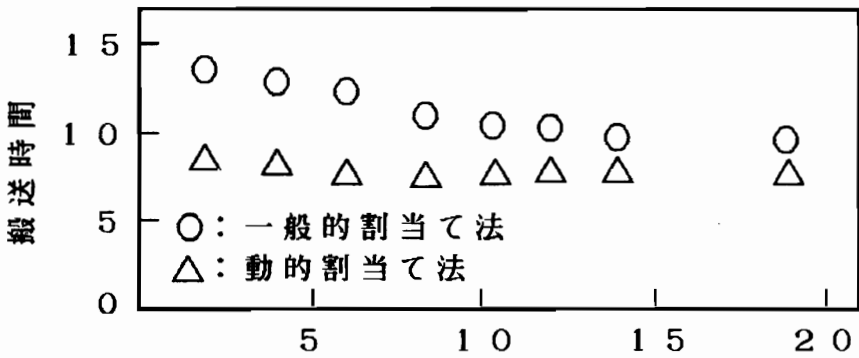
(b) 台車数=4の場合

図6.5 搬送完了時刻



搬送要求発生時間間隔

(a) 台車数=3 の場合



搬送要求発生時間間隔

(b) 台車数=4 の場合

図 6 . 6 搬送時間

する搬送を完了可能であるならば、搬送完了時刻はすべての要求が発生し尽くすまでの時間と最後に発生した要求を処理する時間の和で近似できると考えられる。この時、搬送完了時刻 y は、要求の平均発生間隔 ν と一つの要求に対する平均搬送時間 t とを用いて、次の式で見積ることができる。

$$y = \nu \cdot (50 - 1) + t \quad (6-9)$$

図6. 5 (a)の一般的割当て法の平均搬送完了時刻を y で割ると、要求発生時間間隔が10より大きな場合にはその値が概ね1.0となり、その場合には動的割当て法の有効性がほとんど現われていないことがわかる。逆に、要求発生間隔が10より小さな場合には、値は1.0より大きくなり、その場合には動的割当て法の有効性が生じていることがわかる。(6-9)式の y に対する平均搬送完了時刻の比が1.0を境にして、動的割当て法の有効性に変化が生じるのは、図6. 5 (b)でも同様である。 ν が大きい場合における搬送完了時刻の ν に対する直線性は、(6-9)式に由来する。

このことは、図6. 6の平均搬送時間についても当てはまり、比が1.0より大きくなる要求発生間隔の小さな場合には、一般的割当て法の平均搬送時間が増加しており、逆に動的割当て法のそれはほとんど変化していない。

台車数 n 、要求の平均発生間隔 ν 及び一回の平均要求発生数 m を用いると、一台の台車に与えられる搬送要求一個当りの発生間隔 w を次式で見積ることができる。

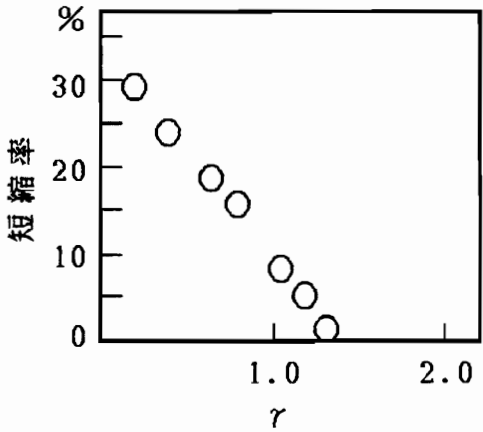
$$w = n \cdot \nu / m \quad (6-10)$$

この値を平均搬送時間 t で割った値 $\gamma = w / t$ は、台車の一回の搬送時間を単位として要求の発生間隔を表わした無次元の値となる。この値が1より大きければ、台車は次の要求を受け付けるまでに現在請け負っている搬送を終了でき、待機時間をもつ余裕のあることを表わし、1より小さければ、搬送を終了する余裕もなく次の要求が与えられることを表わすといえる。

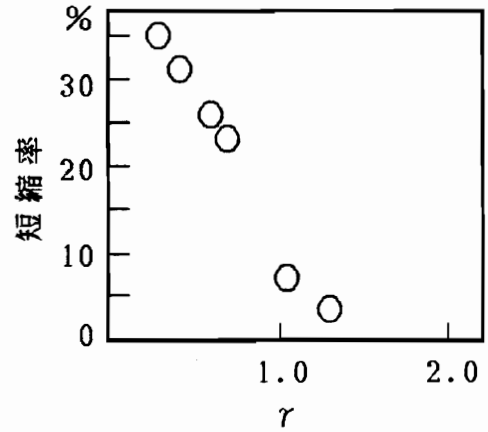
図6. 5 (a), (b)をそれぞれ描き直し、横軸に一般的割当て法に関する指標 γ を、縦軸に一般的割当て法に対する動的割当て法による平均搬送完了時刻の短縮率をとると、次のような特徴をもつ図6. 7 (a), (b)が得られる。

(ア) γ が1.0より小さな場合には、比較的大きな短縮率をもち、動的割当て法の有効性が強く現われる。短縮率は γ が大きくなるにつれ直線的に減少し、 γ が1.0と2.0の間において0となる。

(イ) γ が2.0より大きい場合には、動的割当て法による短縮効果は全くない。



(a) 台車数=3の場合



(b) 台車数=4の場合

図6.7 短縮率

以上のシミュレーション結果から、動的割当て法の有効性に関し次のことが言える。

i) 動的割当て法は、一般的なディスパッチングルールによる割当て法に比べ、常に有効である。割当て法の有効性は、台車の搬送時間に比べ要求の発生頻度が高く、要求を効率よく台車に割り当てることが必要となる場合に特に重要である。すなわち、 γ が1.0より小さい場合であり、そのときには動的割当て法は高い有効性を有している。

ii) 動的割当て法は、Min-Maxマッチング問題の多項式計算時間の解法を用いているため、台車数や搬送要求数が多い場合にもその計算時間における実用性に問題はない。

i) で述べたように、動的割当て法の有効性は γ の値に左右されるが、搬送システムに含まれる種々の要素が γ に及ぼす影響を考慮することにより、動的割当て法の評価を総合的に考察することが可能である。平均搬送時間に影響を及ぼす要因として、搬送ネットワーク網の形態（ループ形、ランダムアクセス形等）、ネットワークの干渉回避に関する工夫の有無（待避路や迂回路の有無等）、台車数、最小干渉許容時間、経路決定法の良しあし等が考え

られる。例えば、3～5章で提案したような効率的な経路決定法により、台車の平均搬送時間を小さくすることができる。台車数は、搬送時間に対し相反する二つの影響を与える。すなわち、台車数が多い場合は、搬送要求の発生した設備の近くで利用可能となる台車が多くなる結果、搬送時間が小さくなるように働く。図6.6の(a),(b)ではこの影響が見られ、台車数の増加により搬送時間の若干の短縮効果が現われている。その反面、台車数が増えると、台車の経路が錯綜し、台車間の干渉が複雑になる結果、干渉回避のための待ち停止時間が多くなり、それが搬送時間を大きくするように働く。

γ は搬送要求の平均発生間隔を平均搬送時間で割ったものであるから、上の例で述べたようになんらかの要因により搬送時間が小さくなると、 γ の値が大きくなる。その結果、図6.7からわかるとおり、同じ搬送要求の発生時間間隔に対しては、割当て法の良しあしの差がそれほど重要でなくなる。経路決定法との関連で言えば、効率的でない割当て法であっても、経路決定法が効率的であれば、その短所が埋められるのである。逆に、効率の悪い経路決定法であれば、搬送時間が大きくなり γ の値が増加する結果、割当て法の良しあしが重要となる。すなわち、経路決定法の効率の悪さを割当て法の効率のよさでカバーしなければならない。

台車の搬送時間に変化がなくとも、生産形態がより多品種化、少量化し、作業工程が細分化されるようになると、搬送要求の発生頻度が高くなり、その結果 γ の値が小さくなる。そのことが、割当て法の良しあしが効率的な搬送に大きく影響するような状況を生むことになる。

工場の搬送系の大規模化、複雑化や台車数の必要以上の増加、また、上で述べた多品種少量化といったような生産形態の変化により、従来の経路決定法の効率性が相対的に低下するような状況においては、動的割当て法のようなより有効な割当て法が重要となる。

6.6 結言

この章では、搬送要求の台車への割当て問題について考えた。この問題は、台車の運行問題全体を最適に解くための部分問題であり、本来、もう一方の部分問題である運行経路問題を考慮して最適化すべき問題である。しかし、他の経路・スケジュール決定問題に関するこれまでの研究から、台車の運行問題に関しても、全体としての厳密解をリアルタイムに導出することの困難

性が指摘できる。従って、台車の運行問題全体を大きく二つに分割し、その一つを搬送要求の割当て問題、いま一つを割当てがなされたとした場合の運行経路を決定する問題として、別々に最適化を図る階層的な解法手法を採用したのである。後者の問題は、第3～5章ですでに議論した。ここでは、一個積みの台車を想定し、総走行完了時刻最小化または最大走行完了時刻最小化の評価尺度をもつ割当て問題を議論した。割当て問題に関しては、一台の台車に一つ以上の要求の割当てをしない（重複を許さない）ものとし、台車の経路が干渉のない最短経路を取り得るとすると、この問題は2部グラフのマッチング問題や最大流最小コスト問題と見なすことができる。この問題に対して、多項式の計算時間をもつ解法アルゴリズムが考えられることを6.2節で示した。6.3節では、重複の許された問題が多重巡回セールスマン問題と同程度にその解法が困難であることを明かにした。

以上のことを踏まえ、6.4節においては、台車の割当て問題を重複を許さない問題に置き換えて、Min-Maxマッチング問題の解法を用いたヒューリスティックな解法アルゴリズム（動的割当て法）を提案した。工場における搬送要求の発生は本来時系列的であるが、この要求群全体の割当てを最適化することは極めて困難である。ここで提案した動的割当て法は、時系列的に発生する搬送要求をその発生時点だけで最適化することにより、全体の最適化を目指す方法である。6.5節において、この割当て法と一般的なディスパッチングルールを用いた典型的な割当て法を簡単なシミュレーションを用いて比較検討した結果、動的割当て法が次のような有効性を有することが明らかになった。

(1) 動的割当て法は、典型的な割当て法に比べ、常に有効である。特に、台車の搬送時間に比べ要求の発生頻度が高く、効率よく台車を要求に割り当てる必要がある場合に、動的割当て法は有効である。

(2) 動的割当て法は、Min-Maxマッチング問題の多項式計算時間の解法を用いているため、台車数や搬送要求数が多く問題の規模が大きい場合にもその計算時間における実用性に問題はない。

第7章 結 論

近年多品種少中量生産における生産の効率化が望まれており、工場においては、柔軟的で効率的な搬送手段である無人搬送車を用いた搬送システム作りが注目を集めている。しかし、これまでのところ、従来の無人搬送車の運行管理方法では、無人搬送車に求められている柔軟で効率的な搬送機能を十分に引き出しているとは言いがたい。

本研究は、効率的な搬送の実現を目指して、無人搬送車（以後、台車という）の運行管理ソフトウェアとして、搬送系内で生じたワーク等の搬送要求に対する運行問題の解法を提案することを具体的な目標とした。本論文では、この問題を割当て問題と運行経路決定問題に分割して考察した。以下、本論文の成果について、各章ごとに要約して述べる。

第2章においては、まず、一般的な“経路・スケジュール決定問題”を経路決定問題、スケジュール決定問題及び混合問題に分類し、それぞれの分類に含まれる具体的な問題に関するサーベイを行った。台車の運行問題の特徴を考察した結果、この問題は、複数台車間の干渉という現象を考慮しなければならぬため、上の分類における混合問題に位置することがわかった。以上の考察により、台車の運行問題の独自性が明かになり、この問題を対象とすることの価値が明確になった。

第3章においては、複数台車のうちの一台に一目的地からなる搬送要求が与えられた場合、走行時間を最小化する運行経路決定問題を議論した。この問題をタイムウィンドウをもつ最短時間経路問題（台車運行型SPPTW）にモデル化し、分枝限定法を用いた解法アルゴリズムを提案した。

一般の分枝限定法でも言えるように、この解法の計算効率はそこで用いられる下界評価法に大きく左右される。3章の後半では、三つの下界評価法（第1～3法）を提案し、それらの有効性について数値例により検討した。その結果、第2法は問題に実行可能解の存在する場合の解の導出に効果的であり、第3法は実行可能解の存在しない場合にその実行不能性を効果的に判断するのに有効であることがわかった。一方、第1法は、比較の基準として考えた方法であり、他の方法に比べ簡便な評価法であるが、評価の誤差が大きいため全体としての有効性に欠けるという結果が得られた。

第4章では、台車運行型SPPTWを拡張した種々の問題について考察した。まず、台車の走行時間に非負性を仮定しない場合に一般化した台車運行

型SPPTWを考えた。その結果、この問題がタイムウィンドウ制約のない古典的な最短経路問題と同じ構造をもつことが明らかになった。この性質を用いることにより、3章で提案した解法を、古典的な最短経路問題に対するBellman-Fordの解法やYenの解法と同じ構造の解法に一般化することができた。

次に、実際の台車の走行において遅延が生じた場合、走行スケジュールをどのように再スケジューリングするかの手順に関する考察を行なった。4章の後半では、各ノードにおける台車の最早到着時刻と最遅出発時刻を導出するアルゴリズムを考え、これら二つの時刻を用いたリスケジューリング手順の提案を行なった。

4章の最後では、搬送要求が複数目的地からなる場合や、目的地における作業時間をもつ場合の運行経路決定問題に対して、第3章で提案した解法を拡張する方法についての考察を行なった。

第5章では、3章、4章が一台車に対する運行経路決定問題を取り扱ったのに対し、全台車の総走行時間最小化を評価尺度として、複数台車の運行経路決定問題に対する解法アルゴリズムを組合せ論的立場から提案した。解法は、各台車に対しいくつかの代替経路をグラフ論的手法により探索する過程と、代替経路上の待ち停止時間を線形計画法を用いて決定する過程とから成る。この解法では、搬送要求が与えられた台車以外の台車の経路の取扱い方により、三つの経路決定法（完全固定法、経路固定法、更新法）を提案した。さらに、数値例を用いることにより検討した結果、これら三つの方法の有効性に関して、次のような知見を得ることができた。

(1) 実行可能な経路決定という点では、更新法が最も優れている。特に、ネットワークの大きさに比べ台車数が多い等の複雑な干渉形態の予想される場合に、他の方法に対するこの優位性が顕著である。更新法は、計算時間が比較的大きいが、解の導出を初期実行可能解を導く段階までで止めることで、計算時間を大きく短縮できる。

(2) 計算時間の面では、完全固定法は極めて優れており、ネットワークの大きさや台車数の変化に対して計算時間に大きな増減は生じない。

また、評価尺度を最大走行完了時刻最小化とした場合の解法についての考察も行なった。

無人搬送車の経路決定では、複数台車の衝突回避を可能にする経路（実行可能経路）を見つけることが必要不可欠の条件である。5章の後半では、上で提案した更新法によっても問題の実行可能経路が求められなかった場合に、

その解法アルゴリズム中に組み込んで使用するヒューリスティックな三つの干渉回避法（第1～3法）を提案した。また、これらの方法の特徴について、数値例により検討した。その結果、次のような特徴があることがわかった。

（a）第1法は、停止中台車が他の台車の通行の妨害になっている場合、この停止中台車を一時的にほかの場所に待避させることが有効な状況に適する方法である。

（b）第2法は、ある台車がある走行途中において一時的に待避行動をとることが、他の台車の通行にとって有効な状況に適する方法である。

（c）第3法は、複数台車の通過により混雑する地点を迂回することが有効である状況に適する方法である。

（d）第1～3法のそれぞれの干渉回避法が有効となる経路決定問題の特徴が異なっているため、三つの方法を組み合わせることにより、さらに有効な干渉回避法となる。

第6章では、同時に複数個発生した搬送要求をどの台車に割り当てるかという割り当て問題に対して、Min-Maxマッチング問題の解法を利用したヒューリスティックな割り当て法（動的割り当て法）を提案した。さらに、シミュレーションにより、この方法を典型的なディスパッチングルールを用いた割り当て法と比較した結果、次のことが明らかになった。

搬送完了時刻及び一回の搬送に要する平均搬送時間の面で、ディスパッチングルールを用いた割り当て法に比べ、動的割り当て法の方が有効である。特に、搬送要求の発生頻度が高く、効率的な割り当ての望まれる状況において、その有効性は高い。

謝 辞

本研究の全過程を通じて、理解あるご指導とご鞭撻を賜った神戸大学大学院自然科学研究科システム科学専攻 藤井進教授に、心から感謝の意を表す。また、親身なご教示と研究に対する有益なご助言をいただいた流通科学大学商学部経営学科 三道弘明助教授に心からお礼申し上げます。

本論文をまとめるにあたり、ご指導、ご助言をいただいた神戸大学大学院自然科学研究科システム科学専攻 平井一正教授、前川禎男教授、神戸大学工学部機械工学科 上田完次教授に心から感謝申し上げます。

本研究の遂行に際し、貴重なご教示とご助言をいただいた岡山大学工学部情報工学科 石井博昭教授、京都大学工学部数理工学科 茨木俊秀教授、米国ジョージア工科大学 Stuart J. Deutsch教授に厚くお礼申し上げます。また、本研究に対して有益なご示唆をいただいた神戸大学大学院自然科学研究科 綿谷泰典助手に心からお礼申し上げます。

日頃から有益なご助言と励ましの言葉をいただいた姫路独協大学経済情報学部 松本隆一教授、神戸大学工学部システム工学科 多田幸生助教授、神戸大学大学院自然科学研究科非常勤講師 廣嶋孝策氏に厚くお礼申し上げます。

この研究を進めるにあたり、種々の面でご協力いただいた神戸大学大学院自然科学研究科学生 関陽君、韓尚秀君、神戸大学工学部システム工学科第五講座 松田秀雄講師、第一講座 田地野悦子助手、近藤敦技官並びに在学生、卒業生諸氏に対して心からお礼申し上げます。

参考文献

- 1) T.Takahashi, "State of the art of AGVs in FMS in Japan", Proc. 6th Int. Conf. Automated Guided Vehicle Systems, pp.51-60, 1988.
- 2) 中村明德, "最新AGVの動向・安全対策と組立ラインへの導入例", 自動化技術, 20(1), pp.38-41, 1988.
- 3) 本田康悟, "NC技術とFA", 計測と制御, 26(7), pp.25-30, 1987.
- 4) 篠塚元雄, 吉谷明次, "自動倉庫・搬送システム", 情報処理, 25(4), pp.317-325, 1984.
- 5) 中村史朗, "エンジニアリング・データベース", 情報処理, 25(4), pp.349-363, 1984.
- 6) 川辺真嗣, "機械設計生産におけるモデリング技術の応用", 情報処理, 25(4), pp.355-363, 1984.
- 7) 木村文彦, "FAにおける情報処理技術の役割", 情報処理, 25(4), pp.355-363, 1984.
- 8) 岡田和保, "生産管理トータルシステム", 情報処理, 25(4), pp.371-377, 1984.
- 9) 渡辺真, "生産管理", 計測と制御, 26(7), pp.556-562, 1987.
- 10) 鍋島一郎, "スケジューリング理論", 数学ライブラリー, 森北出版, 1974.
- 11) 村松林太郎, "生産管理の基礎", 国元書房, 1979.
- 12) 岩田一明他, "生産システム学", 精密工学講座13, コロナ社, 1982.
- 13) 伊東詮, 岩田一明, "フレキシブル生産システム", 日刊工業新聞社, 1984.

- 14) 樋口俊郎, "FMSにおける搬送システムと搬送装置", 日本機械学会誌, 86(775), pp.653-657, 1983.
- 15) 竹本靖, "クリーンルーム検査ロボットとその適用", 自動化技術, 20(1), pp.148-152, 1988.
- 16) 須田英之, "自動倉庫に対し柔軟性に優れる無人フォークリフト・システム", 自動化技術, 20(1), pp.171-176, 1988.
- 17) 津村俊弘, "最近の無人搬送車 (AGV) の開発現況", 自動化技術, 20(1), pp.34-37, 1988.
- 18) 津村俊弘, "無人搬送車とその制御", 計測と制御, 26(7), pp.43-48, 1987.
- 19) 石田英夫, "小規模システムに最適な計量級無人搬送車", 自動化技術, 20(1), pp.153-157, 1988.
- 20) 篠塚, 関, "FAにおける無人搬送と貯蔵の自動化", 計測と制御, 22(11), pp.931-936, 1983.
- 21) 澤田清, "FMSにおける搬送台車の経路決定に関する研究", 神戸大学大学院工学研究科修士論文, 1985.
- 22) 河内素子, "シミュレーションによる工場内自動搬送システムの計画と運用に関する研究", 神戸大学大学院工学研究科修士論文, 1987.
- 23) P.Kruse, "Experience with a large AGV system", Proc. 6th Int. Conf. Automated Guided Vehicle Systems, pp.131-150, 1988.
- 24) 宝崎隆祐, 藤井進, 三道弘明, "到着および出発時間制約のタイムウィンドウを伴う最短時間経路問題に対する一解法", 電子情報通信学会論文誌, J73-A(5), pp.979-986, 1990.
- 25) S.Fujii, H.Sandoh and R.Hohzaki, "A routing control method of automated guided vehicles by the shortest path with time window", Proc. 10th ICPR (to be published).

- 26) R.Hohzaki, S.Fujii and H.Sandoh, "A routing method of automated guided vehicles in FMS by the time-windows constrained shortest path", Proc. 1990 Japan-U.S.A. Symposium on Flexible Automation, 2, pp.485-492, 1990.
- 27) 宝崎隆祐, 藤井進, 三道弘明, "一般化されたTime-windowのある場合の最短経路問題", 日本OR学会1989年度春季研究発表会77`ストラク集, pp.39-40, 1989.
- 28) 藤井進, 宝崎隆祐, 三道弘明, "無人搬送車の最短時間経路問題 (タイムウィンドウを用いたグラフ論的アプローチ)", 第34回システム制御情報学会研究発表講演会講演論文集, pp.461-462, 1990.
- 29) 宝崎隆祐, 藤井進, 三道弘明, "タイムウィンドウをもつ最短経路問題の解法における下界評価法について", 神戸大学大学院自然科学研究科紀要 (校正中).
- 30) 宝崎隆祐, 藤井進, 三道弘明, "一般化されたTime-windowをもつ最短経路問題の効率的解法について", 日本OR学会1989年度秋季研究発表会77`ストラク集, pp.124-125, 1989.
- 31) 宝崎隆祐, 藤井進, 三道弘明, "タイムウィンドウのある一般化されたネットワークにおける最短時間経路問題", 電子情報通信学会論文誌 (投稿中).
- 32) 宝崎隆祐, 藤井進, 三道弘明, "タイムウィンドウをもつ一般化されたネットワークでの最短時間経路問題", 日本OR学会1990年度春季研究発表会77`ストラク集, pp.194-195, 1990.
- 33) 宝崎隆祐, "タイムウィンドウのあるネットワークにおける最短時間経路について", 第25回SSOR予稿集, pp.105-110, 1990.
- 34) 宝崎隆祐, 藤井進, 三道弘明, "タイムウィンドウを用いた無人搬送車の走行リスケジューリングに関する一考察", 日本OR学会1990年度秋季研究発表会77`ストラク集, pp.92-93, 1990.
- 35) 西川明伸, 藤井進, 宝崎隆祐, 三道弘明, "複数目的地及び滞在

- 時間を考慮したタイム・ウィンドウをもつ最短時間経路問題”，日本OR学会1990年度秋季研究発表会77ストラクト集，pp.94-95，1990.
- 36) 藤井進，三道弘明，宝崎隆祐，“自動搬送台車の経路決定法(第1報，経路決定アルゴリズム)”，日本機械学会第68期全国大会講演会講演論文集，pp.261，1989.
 - 37) 藤井進，三道弘明，宝崎隆祐，“自動搬送台車の経路決定法(第1報，経路決定アルゴリズム)”，日本機械学会論文集，55-C(514)，pp.1568-1573，1989.
 - 38) S.Fujii, H.Sandoh and R.Hohzaki, "Routing control of automated guided vehicles in FMS", Proc. 1988 U.S.A.-Japan Symposium on Flexible Automation, pp.629-636, 1988.
 - 39) 宝崎隆祐，藤井進，三道弘明，“無人搬送台車の経路決定法に関する一考察(数値例による経路決定法の定量的分析)”，神戸大学大学院自然科学研究科紀要，8-B，pp.95-102，1990.
 - 40) 藤井進，三道弘明，宝崎隆祐，“FMSにおける自動搬送台車の非同歩走行アルゴリズム”，日本OR学会1988年度春季研究発表会77ストラクト集，pp.25-26，1988.
 - 41) 宝崎隆祐，藤井進，三道弘明，“無人搬送車の経路決定におけるヒューリスティックな干渉回避法”，日本機械学会論文集(掲載決定)。
 - 42) S.Fujii, R.Hohzaki and H.Sandoh, "Heuristic methods for collision-free routing of automated guided vehicles", Proc. 1990 Pacific Conference on Manufacturing (to be appeared).
 - 43) 宝崎隆祐，藤井進，三道弘明，“無人搬送車の経路決定におけるヒューリスティックな干渉回避法”，関西支部第64期定時総会講演会講演概要集，D(990-59)，pp.434-436，1990.
 - 44) L.Bodin, B.Golden, A.Assad and M.Ball, "Routing and scheduling of vehicles and crews", Comput. & Ops. Res.,

- 10(2), pp.63-211, 1983.
- 45) M.Bellman and G.Nemhauser, "The traveling salesman problem: a survey", *Ops. Res.*, 16, pp.538-558, 1968.
 - 46) B.Gavish and K.Srikanth, "An optimal solution method for large-scale multiple traveling salesman problems", *Ops. Res.*, 34(5), pp.698-717, 1986.
 - 47) J.Norback and R.Love, "Geometric approaches to solving the traveling salesman problem", *Management Sci.*, 23(11), pp.1208-1223, 1977.
 - 48) R.Russell, "An effective heuristic for the M-tour traveling salesman problem with some side conditions", *Ops. Res.*, 25(3), pp.517-524, 1977.
 - 49) H.Crowder and M.Padberg, "Solving large-scale symmetric traveling salesman problems to optimality", *Management Science*, 26(5), pp.495-509, 1980.
 - 50) B.Gavish and K.Srikanth, "An optimal solution method for large-scale multiple traveling salesmen problems", *Ops. Res.*, 34(5), pp.698-717, 1986.
 - 51) R.Russell, "An effective heuristic for the M-tour traveling salesman problem with some side constraints", *Ops. Res.*, 25(3), pp.517-524, 1976.
 - 52) R.Karp, "Reducibility among combinatorial problems", *Complexity of Computer Computations*, pp.85-104, Plenum Press, New York, 1972.
 - 53) C.Papadimitriou, "The euclidean traveling salesman problem is NP-complete", *Theor. Comp. Sci.*, 4(3), pp.237-244, 1977.
 - 54) M.Garey, R.Graham and D.Johnson, "Some NP-complete geometric problems", *Proc. of 8th SIGACT symp. on the Theory of Computing*, pp.10-22, 1976.

- 55) D.Rosenkrantz, R.Sterns and P.Lewis, "An analysis of several heuristics for the travelling salesman problem", SIAM J. Comp., 6, pp.563-581, 1977.
- 56) B.Golden, "A statistical approach to the TSP", Networks, 7, pp.209-225, 1977.
- 57) N.Christofides, "Worst-case analysis of a new heuristic for the traveling salesman problem", Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, 1976.
- 58) G.Cornuejols and G.Nemhauser, "Tight bounds on Christofides' traveling salesman heuristic", Math. Prog., 14, pp.116-121, 1978.
- 59) R.Karp, "Probabilistic analysis of partitioning algorithms for the traveling salesman problem in the plane", Math. Ops. Res. 2, p.209-224, 1977.
- 60) S.Lin and B.Kernighan, "An effective heuristic algorithm for the traveling salesman problem", Ops. Res., 21, pp.498-516, 1973.
- 61) S.Akl, "The minimal directed spanning graph for combinatorial optimization", Austral. Comput. J., 12(4), pp.132-136, 1980.
- 62) A.Frieze, G.Galbiati and F.Maffioli, "On the worst-case performance of some algorithms for the asymmetric traveling salesman problem", Networks, 12(1), pp.23-39, 1982.
- 63) P.Kanellakis and C.Papadimitriou, "Local search for the asymmetric traveling salesman problem", Ops. Res., 28, pp.1219-1222, 1980.
- 64) R.Karp, "A patching algorithm for the nonsymmetric traveling salesman problem", SIAM J. Comp., 8,

pp.561-573, 1979.

- 65) P.Van der Cruyssen and M.Rijchaert, "Heuristic for the asymmetric travelling salesman problem", J. Ops. Res. Soc., 30, pp.697-701, 1978.
- 66) M.Held and R.Karp, "The traveling salesman problem and minimum spanning trees", Ops. Res., 18, pp.1138-1162, 1970.
- 67) M.Held and R.Karp, "The traveling salesman problem and minimum spanning trees: part II", Mathematical Programming, 1, pp.6-25, 1971.
- 68) A.Geoffrion, "Lagrangian relaxation and its uses in integer programming", Math. Programming study, 2, pp.82-114, 1974.
- 69) P.Wolfe, "A method of conjugate subgradients for minimizing non-differentiable functions", Math. Prog. Study, 3, pp.145-173, 1975.
- 70) 今野浩, 山下浩, "非線形計画法", ORライブラリー, 日科技連, 1978.
- 71) A.Brellman and S.Hong, "Transformation of multi-salesman problem to the standard traveling salesman problem", J. ACM, 21, pp.500-504, 1974.
- 72) C.Orloff, "Routing a fleet of M vehicles to/from a central facility", Networks, 4, pp.35-64, 1974.
- 73) R.Bellman and K.Cooke, "The Konigsberg Bridges problem generalized", J. Math. Anal. Appl., 25, pp.1-7, 1969.
- 74) J.Edmonds and E.Johnson, "Matching Euler tours, and the chinese postman", Math. Prog., 5, pp.88-124, 1973.
- 75) E.Beltrami and L.Bodin, "Networks and vehicle routing for municipal waste collection", Networks, 4, pp.65-94,

1974.

- 76) B.Golden, T.Magnanti and H.Nguyen, "Implementing vehicle routing algorithms", *Networks*, 7, pp.113-148, 1977.
- 77) N.Christofides, A.Mingozzi and P.Toth, "Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations", *Math. Prog.*, 20, pp.255-282, 1981.
- 78) B.Gillett and L.Miller, "A heuristic algorithm for the vehicle dispatch problem", *Ops. Res.*, 22, pp.340-349, 1974.
- 79) L.Bodin and S.Kursh, "A detailed description of a street sweeper routing and scheduling system", *Comput. & Ops. Res.*, 6, pp.181-198, 1979.
- 80) L.Bodin and S.Kursh, "A computer-assisted system for the routing and scheduling of street sweepers", *Ops. Res.*, 26(4), pp.525-537, 1978.
- 81) G.Danzzig and D.Fulkerson, "Minimizing the number of tankers to meet a fixed schedule", *Naval Res. Logistics Quart.*, 1, pp.217-222, 1954.
- 82) J.Lenstra and A.Kan, "Complexity of vehicle routing and scheduling problems", *Networks*, 11(2), pp.221-227, 1981.
- 83) L.Bodin, D.Risebfuekd and A.Kydes, "Scheduling and estimation techniques for transportation planning", *Comp. Ops. Res.*, 8, pp.25-38, 1981.
- 84) M.Segal, "The operator-scheduling problem: a network-flow approach", *Ops. Res.*, 22, pp.808-823, 1974.
- 85) A.Wren, "Computer scheduling of public transport: Urban passenger vehicle and crew scheduling", North-Holland, Amsterdam, 1981.

- 86) J.Rubin, "A technique for the solution of massive set covering problems with application to airline crew scheduling", *Transportation Sci.*, 7, pp.34-48, 1973.
- 87) R.Newton and W.Thomas, "Bus routing in a multi-school system", *Comput. & Ops. Res.*, 1, pp.213-222, 1974.
- 88) L.Bodin and L.Berman, "Routing and scheduling of school buses by computer", *Transportation Sci.*, 13(2), pp.113-129, 1979.
- 89) M.Ball, B.Golden, A.Assad and L.Bodin, "Planning for truck fleet size in the presence of a common carrier option", *Decision Science*, 14(1), 103-120, 1983.
- 90) A.Assad, M.Ball, L.Bodin and B.Golden, "Combined distribution routing and scheduling in a large commercial firm", *Proc. of 1981 Northeast AIDS Conf.*, pp.99-102, 1981.
- 91) R.Russell and W.Igo, "An assignment routing problems", *Networks*, 9, pp.1-17, 1979.
- 92) M.Solomon and J.Desrosiers, "Time window constrained routing and scheduling problems", *Transportation Sci.*, 22(1), pp.1-13, 1988.
- 93) S.Pallottino, "Shortest-path methods: complexity, interrelations and new propositions", *Networks*, 14, pp.257-267, 1984.
- 94) J.Desrosiers, F.Soumis and M.Desrochers, "Routing with time windows by column generation", *Networks*, 14(4), pp.545-565, 1984.
- 95) M.Solomon, "Algorithms for the vehicle routing and scheduling problems with time windows constraints", *Ops. Res.*, 35(2), pp.254-265, 1987.
- 96) M.Solomon, "On the worst-case performance of some

- heuristics for the vehicle routing and scheduling problem with time window constraints", *Networks*, 16, pp.161-174, 1986.
- 97) A.Kolen, A.Kan and H.Trienekens, "Vehicle routing with time windows", *Ops. Res.*, 35(2), pp.266-273, 1987.
- 98) M.Desrochers and F.Soumis, "A generalized permanent labelling algorithm for the shortest path problem with time windows", *Information Systems and Operational Research*, 26(3), pp.191-213, 1988.
- 99) M.Desrochers and F.Soumis, "A reoptimization algorithm for the shortest path problem with time windows", *European Journal of Operational Research*, 35(2), pp.242-255, 1987.
- 100) E.Dijkstra, "A note on two problems in connection with graphs", *Numerische Mathematik*, 1, pp.269-271, 1959.
- 101) T.Hu, "Combinatorial algorithms", Addison-Wesley Publishing Company, Massachusetts, 1982.
- 102) E.Lawler, "Combinatorial optimization: networks and matroids", Holt, Rinehart and Winston, New York, 1976.
- 103) H.Wagner, "Principles of operations research", Prentice Hall Inc., New Jersey, 1975.
- 104) 茨木俊秀, "組合せ最適化", 講座・数理計画法 8, 産業図書, 1983.
- 105) 今野浩, "整数計画法", 講座・数理計画法 6, 産業図書, 1981.
- 106) D.Shier, "On algorithm for finding the K shortest paths in a network", *Networks*, 9, pp.195-214, 1979.
- 107) T.Hu, "Integer programming and network flows", Addison-Wesley Publishing Company, Massachusetts, 1969.

- 108) R.Bellman, "Dynamic programming", Princeton University Press, New Jersey, 1957.
- 109) R.Bellman, "Dynamic programming and modern control theory", Academic Press, New York, 1965.
- 110) 鍋島一郎, "動的計画法", 森北出版, 1968.
- 111) 坂口実, "動的計画法", 近代数学新書, 至文堂, 1968.
- 112) R.Bellman, "On a routing problem", Quart. Applied Math., 16, pp.87-90, 1958.
- 113) L.Ford, Jr., "Network flow theory", The RAND Corp., P-923, 1956.
- 114) J.Yen, "An algorithm for finding shortest routes from all source nodes to a given destination in general network", Quart. Appl. Math., 27, pp.526-530, 1970.
- 115) S.Fujii and H.Sandoh, "A routing algorithm for automated guided vehicles in FMS", Proc. of 9th ICPR, 2, pp.2261-2267, 1987.
- 116) 古林隆, "線形計画法", 講座・数理計画法 2, 産業図書, 1980.
- 117) 尾崎弘, 白川功, "グラフとネットワークの理論", コロナ社, 1973.
- 118) M.Iri, "A new method of solving transportation-network problems", J. Ops. Res. Soc. Japan, 3, pp.27-87, 1960.
- 119) H.Kuhn, "The hungarian method for the assignment problem", Naval Res. Logist. Quart., 2, pp.83-97, 1955.
- 120) H.Kuhn, "Variants of the hungarian method for assignment problems", Naval Res. Logist. Quart., 3, pp.253-258, 1956.

- 121) M.Klein, "A primal method for minimal cost flows",
Management Sci., 14, pp.205-220, 1967.
- 122) C.Chen, C.Lee and C.McGillem, "Task assignment and load
balancing of autonomous vehicles in a flexible
manufacturing system", IEEE J. Robotics and Automation,
RA-3(6), pp.659-671, 1987.
- 123) 関根智明, "スケジューリングの理論", 日刊工業新聞社,
1971.
- 124) 中村豊, 長尾勝紀, 田村担之, "納期を考慮したルール表現に
よるFMSスケジューリング", 第1回インテリジェントFA
シンポジウム講演論文集, pp.13-16, 1987.
- 125) 渡部透, 城戸行右, 藤井六郎, 稲上健一, "ジョブショップス
ケジューリングの発見的優先ルールと自己最適化", 第2回イ
ンテリジェントFAシンポジウム講演論文集, pp.45-46,
1989.
- 126) 渡部透, "FMSのオンラインスケジューリングと適応制御",
セミナー「FMSとそのソフトウェア」テキスト,
自動制御協会, pp.58-71, 1984.
- 127) 藤井進, 三道弘明, 河内素子, 綿谷泰典, 高田実, "搬送シ
ミュレータを用いたカート運用法に関する基礎的研究",
第1回インテリジェントFAシンポジウム講演論文集,
pp.3-4, 1987.
- 128) 伊藤治生, "汎用FMSシミュレータにおける自動搬送台車の
非干渉走行制御機能の開発", 神戸大学工学部卒業論文,
1988.
- 129) 河内新治, "無人搬送車を中心としたFMSシミュレーショ
ン", セミナー「FMSの計画・設計とその評価」テキスト,
精機学会, pp.38-46, 1985.