

Title	自動搬送車の動作計画問題に関する研究
Author(s)	山根, 毅史
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1360
Rights	
Description	Supervisor:平石 邦彦, 情報科学研究科, 修士

修 士 論 文

自動搬送車の動作計画問題に関する研究

指導教官 平石 邦彦 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

山根 毅史

2000年2月15日

目次

1	はじめに	1
1.1	背景、目的	1
1.2	本論文の構成	2
2	自動搬送車の動作計画問題	3
2.1	状況設定, 諸定義	3
2.1.1	自動搬送システム	3
2.1.2	軌道	3
2.1.3	AGV	3
2.1.4	ステーション	4
2.1.5	搬送要求	4
2.2	問題の定義	4
2.3	グラフへの表現	5
2.4	制約条件	5
3	アルゴリズム	7
3.1	アルゴリズムの流れ	7
3.2	入出力	7
3.3	アルゴリズム	8
3.3.1	Main	8
3.3.2	Sub1	11
3.3.3	Sub2	13
3.4	アルゴリズムの正当性について	14
3.4.1	文献 [3] の補題	14
3.4.2	アルゴリズムの正当性	14

3.5	計算量について	16
3.6	具体例	17
3.6.1	AGV が 1 台の場合	17
3.6.2	AGV が 2 台の場合	20
4	まとめ	24
4.1	結論	24
4.2	今後の課題	24

目 次

2.1	軌道	4
2.2	ステーションと交差点	5
3.1	経路グラフの作成	13
3.2	走行路グラフ (例 1)	17
3.3	経路グラフ (例 1)	17
3.4	経路グラフ 2(例 1)	18
3.5	グラフ \bar{G} (例 1)	19
3.6	走行路グラフ (例 2)	20
3.7	経路グラフ (例 2)	21
3.8	整理後の経路グラフ (例 2)	22

表 目 次

3.1 点集合 (例 1)	18
3.2 搬送要求 (例 2)	20

第 1 章

はじめに

1.1 背景、目的

近年，様々な分野の工場のオートメーション (FA) 化が進んでいる．例えば自動倉庫や，CAD，CAM による設計の段階におけるコンピュータの使用から，製造プラントのオートメーション化などの工場内の設備，装置を利用などが挙げられる．具体的には，自動搬送車，工作機械，産業ロボットなどにより，プラント間あるいはプラント内の製造物，荷物などの搬送や，部品の組み立て，検査，出荷など全工程において自動化が進んでいる．工場の FA 化を進めることで生産ラインの連携，あるいは統合化を図っている．こうした様々な部分の FA 化を行うことによるメリットは，作業の効率化，人件費削減などが挙げられる．

こうした流れの中で自動搬送車を利用した自動搬送システムの導入が工場のさまざまなラインで進められている．自動搬送車 (Automated Guided Vehicle - 以後 AGV と呼ぶ) は，ある敷設された軌道の上を自動的に移動して，物品の搬送や加工物の受け渡しなどを行う無人の台車である．部品や加工物の自動搬送は工場の自動化には欠かせない．AGV を利用した自動搬送システムは床面や地面などを車輪で走るタイプのもので，レールの上を走るものや，床面に張られた反射テープの上など誘導線を走ることで実現する．通常この自動搬送システムは複数の AGV が動作することで複数の搬送要求を満たす．従って AGV 同士の衝突が問題になってくる．この衝突を回避するために AGV が自律的に動くタイプのものでないものがある．搬送要求は当然のことながら，AGV 同士が衝突することなく仕事をこなさなければならない．前者は例えば各 AGV が他の AGV と衝突しないようにするために，AGV それぞれが衝突を事前に検出しそれを回避を行ったりするのに対して，後者は衝突を集中的に管理している装置で検出したり，あるいは事前に搬

送計画を立てスケジュール通りに動作させるといったことが挙げられる。

また，社会的なニーズとして多品種少量の生産が主流となる中で，それを実現するための複数の生産ラインや倉庫間の搬送形態は複雑になっており，それに対応するため，またそれに伴って各ラインの運転効率を下げることはないような搬送システムを構築する必要がある．ここでどのような搬送システムを組む上でも考慮しなければならないのは AGV 同士の衝突の検出とその回避方法である．現在ではそれを経験的な知識を基に，スケジュールを組んだり，経路決定や運行制御に関する問題をグラフなど数学的にモデル化を行い，スケジューリング問題として解決しようとする試みがなされたり [1][2] している．

本研究では，スケジュール通りに AGV を動作させるような搬送計画を事前に立てるアルゴリズムを提案することを目的とする．ここで述べる自動搬送システムは事前に「いつ，どこに」に移動しなければならないといった各 AGV の動作経路を決定しておき，その通りに AGV を動作させる．また，各 AGV は複数あるどの搬送要求を満足してもよく，目的地に到着後の AGV の動作は軌道上ならば制限はない．ただし，AGV の台数と搬送要求の数は一致しているものとする．

1.2 本論文の構成

第 2 章 本研究で用いている言葉の定義や扱う問題の定義，制約条件などについて述べている．

第 3 章 本研究で提案する自動搬送車の動作計画を求めるアルゴリズムをについて述べ，アルゴリズムの正当性，計算量の評価などについて述べている．

第 4 章 結論，今後の課題について述べている．

第 2 章

自動搬送車の動作計画問題

2.1 状況設定，諸定義

本研究で述べる自動搬送システム，運行規則等について述べる．

2.1.1 自動搬送システム

ここで述べる自動搬送システムは，ある定められた軌道の上を走る無人の搬送車 (AGV) を使って，その軌道とつながっている各プラントで受け取ったものを搬送し，目的のプラントまで届けるシステムである．AGV は軌道上を動き，動作中それぞれ衝突のないように，またブロッキングやデッドロックの起きないようにしなければならない．受け取る時間や届けなければならない時間はあらかじめ決められており，経路は動作前に計画する．また経路は，動作中に変更の起きない静的なものとする．

2.1.2 軌道

軌道は AGV がそれに沿って動くための経路である．また同じ軌道，例えば図 2.1 のように，ある方向に進む AGV がいて反対方向から同じ軌道に沿って他の AGV が進んできた場合，すれ違うことができない．

2.1.3 AGV

AGV はあらかじめ敷設されている軌道の上だけを移動することができる．また複数の AGV が動作する場合，どの AGV も同じ性能を持つ．つまり同じ時間内に同じだけの移



図 2.1: 軌道

動能力を持つ．ただしここでは荷物の量，重さによる移動量の変化は考えない．つまりどんな荷物も同じ時間で運ぶ．

2.1.4 ステーション

AGV が荷物を受け渡しする場所がステーションである．例えば工場で製造プラントで出来たものを倉庫に搬送するような場合製造プラントおよび倉庫がステーションになる．

2.1.5 搬送要求

AGV は決められた時間に指定のステーションに受取りに行き，決められた時間に送り届けなければならない．この時間とステーションの組み合わせに制限はなく，またどの AGV がどの仕事をするといった割り当てもない．搬送要求は次のような組の集合によって表す．

$\{(出発時刻, 出発点), (到着時刻, 到着点)\}$

2.2 問題の定義

軌道の上にステーションがいくつかあり，ステーション間を AGV が移動して部品を搬送する，この自動搬送システム上で，

- 動作経路を決定する時の AGV の初期位置
- 搬送要求

が与えられた時に搬送要求を満たすような AGV の動作経路つまり各時刻における各 AGV の位置 (動作計画) を求める．

2.3 グラフへの表現

まず，AGV が移動する走行路はステーション間をつなぐ軌道を辺集合，ステーションを点集合とすることでグラフ化できる．AGV 同士が衝突するパターンとしてステーションあるいはステーション間の 2 つがある．そこでステーション間に交差点という仮想的な点を設けることにより，衝突するポイントを点として表現する．交差点は任意の 2 つのステーション間に必ず 1 つ存在するようにする．例えば下図の通りである．

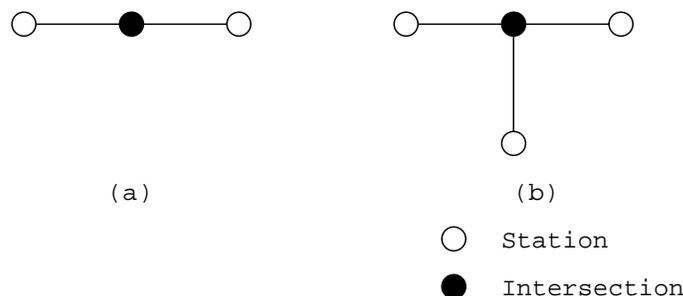


図 2.2: ステーションと交差点

図 2.2 の (a) はステーションが 2 つの場合，(b) はステーションが 3 つの場合である．(b) のような場合つまり，走行路が交わるような場合必ず交差点もしくはステーションが置かれる．ただし，実際の軌道ではステーション間の距離が同じとは限らないので，ダミーのステーションおよび交差点を挿入することで単位時間で移動できるようにする．

2.4 制約条件

自動搬送システムで軌道上を複数の AGV が移動する場合，それらが互いに衝突しないように何らかの制約を設けておく必要がある．

- 各 AGV の動作計画が

- (a) 時刻 0 で全ての AGV がステーションにいる
- (b) AGV が時刻 t でステーション s にいる時，時刻 $t+2$ までステーション s に留まるか，あるいは s に隣接する交差点 c に移動し，時刻 $t+2$ で c と隣接するステーションに移動する．

を満たす．

- 時刻は離散的な時間である .
- AGV はステーション間を単位時間で移動できるものとする .

この時 , 同一時刻に複数の AGV が同じ場所にいないような動作計画が存在すれば衝突は発生しない .

さらに , 問題を単純化するために , つぎの仮定を置く .

- AGV の台数と要求の数は一致しており , 各 AGV は唯 1 つの要求を満たす .

第 3 章

アルゴリズム

提案するアルゴリズムは搬送要求と AGV が移動する軌道をグラフ化したものから動作経路を決定する。

3.1 アルゴリズムの流れ

1. 時刻 0 から動作完了時刻までの経路グラフを作成する。経路グラフは各時刻における AGV の位置を点で、動作経路を有向辺で表したグラフである。サブルーチン Sub1 で後述する。
2. グラフに重みづけを行う
3. そのグラフから無限大ではない最小コストの disjoint パスを $3k$ 本見つける。見つからなければ動作計画は存在しない。
4. $3k$ 本の disjoint パスは初期位置から出発点への経路、出発点から到着点への経路、および到着点から動作完了時刻までの動作経路を含んでいる。これにより動作計画が決定する。

3.2 入出力

このアルゴリズムの入出力は以下の通りである。

- 入力
- AGV の台数 (k 台)
 - AGV の初期位置

- 軌道のグラフ
- 搬送要求

出力 各時刻における各 AGV の位置

3.3 アルゴリズム

3.3.1 Main

Step1 搬送要求の中から到着時刻の最も遅い時刻を t_1 とする .

Step2 Sub1 を使い , 時刻 0 から時刻 t_1 までのグラフを作成する . これを経路グラフと呼ぶ . 作成した経路グラフの中で

- 時刻 0 において各 AGV の初期位置に対応する点集合を V_{1s} とする .
- 搬送要求の出発点と出発時刻で一意に決まる点集合を D とする .
- 到着点と到着時刻で決まる点集合を A とする .
- 時刻 t_1 における全てのステーションに対応する点集合を V_{3t} とする .

Step3 点集合 D の要素数と同じ要素数をもつ 2 つの点集合 V_{2s} と V_{1t} をグラフに加える . 同様に点集合 A の要素数と同じ要素数をもつ点集合 V_{3s} と V_{2t} をグラフに加える . D と A の要素数は制約条件より等しい . なお , D の各点 d と V_{2s} の点 d_{2s} および V_{1t} の点 d_{1t} が対応しているとし , また , A の各点 a と V_{3s} の点 a_{3s} および V_{2t} の点 a_{2t} が対応しているとする .

Step4 各点集合について以下のような操作を行う .

- D の各点 d を始点とする各辺 $d \rightarrow d'$ に対し , 辺 $d_{2s} \rightarrow d'$ を加える .
- D の各点 d を終点とする各辺 $d'' \rightarrow d$ に対し , 辺 $d'' \rightarrow d_{1t}$ を加える .
- D の点とそれに接続する辺を削除する .

同様にして ,

- A の各点 a を始点とする各辺 $a \rightarrow a'$ に対し , 辺 $a_{3s} \rightarrow a'$ を加える .
- A の各点 a を終点とする各辺 $a'' \rightarrow a$ に対し , 辺 $a'' \rightarrow a_{2t}$ を加える .

- A の点とそれに接続する辺を削除する .

さらに , ソース s , シンク t , および , 以下の辺を加える .

- $s \rightarrow v$ ($v \in V_{1s} \cup V_{2s} \cup V_{3s}$) .
- $v \rightarrow t$ ($v \in V_{1t} \cup V_{2t} \cup V_{3t}$) .

この操作の結果 , 出来たグラフを G とする .

Step5 グラフ G の各辺のコストを決定する . 各辺のコストを $3k$ 次元の非負のベクトルである .

s と t に接続していない辺のコストは , 時刻 t の点から時刻 $t+2$ の点へ接続する辺のように , 経路グラフ上のその辺を通ることが AGV の動作しないことを意味する場合 ,

$$\underbrace{(0, 0, 0, \dots, 0, 0, 0)}_{3k \text{ 個}}$$

とする . それ以外は

$$\underbrace{(1, 1, 1, \dots, 1, 1, 1)}_{3k \text{ 個}}$$

とする . s と t に接続している辺のコストはそれぞれ次のように決められる .

$$E_{1s} : s \rightarrow V_{1s}$$

$$\underbrace{(0, 0, \dots, 0)}_{k \text{ 個}}, \underbrace{(\infty, \infty, \dots, \infty, \infty, \infty, \dots, \infty)}_{2k \text{ 個}}$$

のようなコストを全ての辺が持っている .

$$E_{2s} : s \rightarrow V_{2s}$$

$$\begin{aligned} &(\infty, \infty, \dots, \infty, 0, \infty, \infty, \dots, \infty, \infty, \infty, \dots, \infty) \\ &(\infty, \infty, \dots, \infty, \infty, 0, \infty, \dots, \infty, \infty, \infty, \dots, \infty) \\ &(\infty, \infty, \dots, \infty, \infty, \infty, 0, \dots, \infty, \infty, \infty, \dots, \infty) \\ &\vdots \\ &\underbrace{(\infty, \infty, \dots, \infty)}_{k \text{ 個}}, \underbrace{(\infty, \infty, \dots, \infty)}_{k \text{ 個}}, \underbrace{(0, \infty, \infty, \dots, \infty)}_{k \text{ 個}} \end{aligned}$$

のようなコストをそれぞれの辺で持つ .

$$E_{3s} : s \rightarrow V_{3s}$$

$$\underbrace{(\infty, \infty, \dots, \infty, \infty, \infty, \dots, \infty)}_{2k \text{ 個}}, \underbrace{(0, 0, \dots, 0)}_{k \text{ 個}}$$

のようなコストを全ての辺が持っている .

$$E_{1t} : V_{1t} \rightarrow t$$

$$\underbrace{(0, 0, \dots, 0)}_{k \text{ 個}}, \underbrace{(\infty, \infty, \dots, \infty, \infty, \infty, \dots, \infty)}_{2k \text{ 個}}$$

のようなコストを全ての辺が持っている .

$$E_{2t} : V_{2t} \rightarrow t$$

E_{2t} のそれぞれの辺の始点を決める (到着点, 到着時刻) の組と搬送要求で対になっている (出発点, 出発時刻) がある . これによって決まる点を終点にもつような E_{2s} の辺のコストを E_{2t} の各辺のコストとする .

$$E_{3t} : V_{3t} \rightarrow t$$

$$\underbrace{(\infty, \infty, \dots, \infty, \infty, \infty, \dots, \infty)}_{2k \text{ 個}}, \underbrace{(0, 0, \dots, 0)}_{k \text{ 個}}$$

のようなコストを全ての辺が持っている .

Step6 Step5 で重みづけの終わったグラフの各点に対し, 各辺の始点が終点より小さい数字を持つように番号付けを行う . そして, このグラフ G から s から t までの $3k$ 本の disjoint なパスを Sub2 を実行して見つける . そのパスが,

- 無限大のコストを持つ場合, 動作計画は存在しない .
- それ以外の場合, Step7 へ

Step7 Step6 で求められるグラフ \bar{G} 上の以下のようなパス

$$\begin{aligned} \langle s, \dots, s \rangle &\rightarrow \langle v_1^1, \dots, v_k^1, w_1^1, \dots, w_k^1, x_1^1, \dots, x_k^1 \rangle \rightarrow \\ &\rightarrow \langle v_1^2, \dots, v_k^2, w_1^2, \dots, w_k^2, x_1^2, \dots, x_k^2 \rangle \rightarrow \dots \\ &\rightarrow \langle v_1^m, \dots, v_k^m, w_1^m, \dots, w_k^m, x_1^m, \dots, x_k^m \rangle \rightarrow \langle t, \dots, t \rangle \end{aligned}$$

で,

$$\begin{aligned} v_1^1 &\rightarrow v_1^1 \rightarrow \dots \rightarrow v_1^m \\ &\vdots \\ v_k^1 &\rightarrow v_k^1 \rightarrow \dots \rightarrow v_k^m \\ w_1^1 &\rightarrow w_1^1 \rightarrow \dots \rightarrow w_1^m \\ &\vdots \end{aligned}$$

$$\begin{aligned}
w_k^1 &\rightarrow w_k^1 \rightarrow \cdots \rightarrow w_k^m \\
x_1^1 &\rightarrow x_1^1 \rightarrow \cdots \rightarrow x_1^m \\
&\vdots \\
x_k^1 &\rightarrow x_k^1 \rightarrow \cdots \rightarrow x_k^m
\end{aligned}$$

が G 上で $3k$ 本の disjoint なパスである。 $v_1^1, \dots, v_k^1, w_1^1, \dots, w_k^1, x_1^1, \dots, x_k^1$ は V_{1s}, V_{2s}, V_{3s} に、 $v_1^m, \dots, v_k^m, w_1^m, \dots, w_k^m, x_1^m, \dots, x_k^m$ は V_{1t}, V_{2t}, V_{3t} の点集合全体に対応している。それぞれ、

V_{1s}	v_1^1, \dots, v_k^1
V_{2s}	w_1^1, \dots, w_k^1
V_{3s}	x_1^1, \dots, x_k^1
V_{1t}	v_1^m, \dots, v_k^m
V_{2t}	w_1^m, \dots, w_k^m
V_{3t}	x_1^m, \dots, x_k^m

だとすると、上の $3k$ 本の disjoint なパスから k 本の以下のような disjoint パスを作る。

$$\begin{aligned}
v_1^1 &\rightarrow \cdots \rightarrow v_1^m \rightarrow w_1^2 \rightarrow w_1^3 \rightarrow \cdots \rightarrow w_1^m \rightarrow x_1^2 \rightarrow x_1^3 \rightarrow \cdots \rightarrow x_1^m \\
v_2^1 &\rightarrow \cdots \rightarrow v_2^m \rightarrow w_2^2 \rightarrow w_2^3 \rightarrow \cdots \rightarrow w_2^m \rightarrow x_2^2 \rightarrow x_2^3 \rightarrow \cdots \rightarrow x_2^m \\
&\vdots \\
v_k^1 &\rightarrow \cdots \rightarrow v_k^m \rightarrow w_k^2 \rightarrow w_k^3 \rightarrow \cdots \rightarrow w_k^m \rightarrow x_k^2 \rightarrow x_k^3 \rightarrow \cdots \rightarrow x_k^m
\end{aligned}$$

このパスの v_1^1, \dots, v_k^1 は初期位置、 w_1^2, \dots, w_k^2 は出発点、 x_1^2, \dots, x_k^2 は到着点、 x_1^m, \dots, x_k^m は動作完了時（時刻 t_1 ）の点に対応する。つまり時刻 0 から動作完了時刻 t_1 までの各位置が決まり、動作計画が決定する。

3.3.2 Sub1

ある時刻 t_1 から時刻 t_2 までの経路グラフを作る。図 3.1 は経路グラフの作成過程を示している。なお、この図では $time$ が偶数時間とする。

Step1 与えられた走行路のグラフを $t_2 - t_1$ 個複製する．それらをそれぞれ $G_t (t = 0, \dots, t_2 - t_1 + 1)$ とする．またパラメータ $time$ の初期値を $time = t_1$ とする．

Step2 もし $time = t_2$ だったら終了．そうでない場合

- $time$ が偶数時間の時 Step3 へ
- $time$ が奇数時間の時 Step4 へ

Step3 時刻 $time$ において AGV が存在可能なステーションを G_{time} 上の点で見つける．この点の集合を X とする．さらに X の各点に対し，隣接する交差点を G_{time+1} 上で見つけ， X の各点を始点， G_{time+1} 上で隣接する交差点を終点とする辺を加える．さらに， $time \leq t_2 - 2$ のときは， X の各点に対し，同じ位置を表すステーションを G_{time+2} 上で見つけ， X の各点を始点， G_{time+2} 上の同じ位置を表す点を終点とする辺を加える．

その後 Step5 へ

Step4 時刻 $time$ において AGV が存在可能な交差点を G_{time} 上の点で見つける．この点の集合を X とする．さらに X の各点に対し，隣接するステーションを G_{time+1} 上で見つけ， X の各点を始点， G_{time+1} 上で隣接するステーションを終点とする辺を加える．

Step5 G_{time} 上で他の複製したグラフへ辺の接続がない点を削除する．

Step6 $time = time + 1$ とし．Step2 へ戻る．

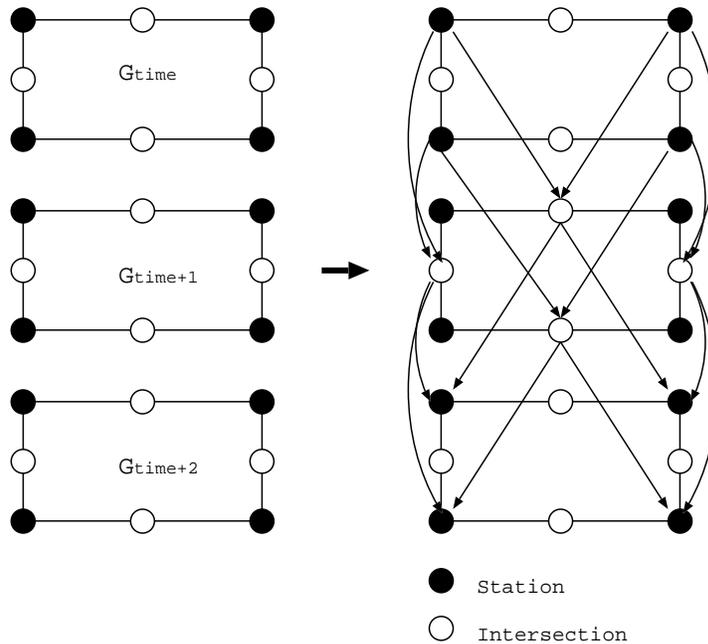


図 3.1: 経路グラフの作成

3.3.3 Sub2

Step1 グラフ G から, 以下のようなグラフ \bar{G} を作る .

点 V' G 上の点の $3k$ 個の組 $\langle v_1, v_2, v_3, \dots, v_k \rangle$ を \bar{G} の点とする . ただし, s あるいは t を除いて点の組には同じ要素は含まれない .

枝 A' : 以下を満たす時に限り, 次のような辺

$$\langle v_1, v_2, v_3, \dots, v_k \rangle \rightarrow \langle v_1, v_2, v_4, \dots, v_k \rangle$$

が存在する : $v_j \rightarrow \hat{v}_j$ がグラフ G の辺であり, かつ, $v_j = \min\{v_1, \dots, v_k\}$.

この辺を「 j 番目の次元の辺」とよぶ .

この辺のコストとして, グラフ G の辺 $v_j \rightarrow \hat{v}_j$ のコストの j 番目の要素を与える .

Step2 G' から Dijkstra のアルゴリズム [5] 等により点 $\langle s, \dots, s \rangle$ から $\langle t, \dots, t \rangle$ への最小コストのパスを見つける .

- 見つかる場合, Main の Step5 へ戻る
- 見つからない場合, 動作計画は存在しない .

3.4 アルゴリズムの正当性について

3.4.1 文献 [3] の補題

アルゴリズムの正当性は文献 [3] に示されている結果に基づいている．まず，準備として，ソース s とシンク t をもつ有向で非循環なグラフ G 上の各辺 $v \rightarrow w$ に対し， k 次元非負ベクトルのコスト

$$\langle c_{vw}^{(1)}, c_{vw}^{(2)}, \dots, c_{vw}^{(k)} \rangle$$

が与えられているとする． G 上に s から t への k 本の有向パス P_1, \dots, P_k が存在するとき， j 番目のパスのコスト $c_{P_j}^{(j)}$ をつぎのように定める．

$$c^{(j)}(P_j) = \sum_{v \rightarrow w \text{ on } P_j} c_{vw}^{(j)}.$$

目的は，vertex-disjoint な k 本のパスで，コストの合計 $\sum_{j=1,k} c^{(j)}(P_j)$ を最小にするものを求めることである．

補題

コスト $c^j(P_j) = l_j (j = 1, \dots, k)$ をもつ s から t への k 本の vertex-disjoint なパス P_1, \dots, P_k が G に存在するならば，かつそのときに限り， j 番目の次元の辺のコストの合計が $l_j (j = 1, \dots, k)$ であるような $\langle s, \dots, s \rangle \rightarrow$ から $\langle t, \dots, t \rangle$ へのパス \bar{P} が \bar{G} に存在する．

証明は Appendix に示した．この補題より， G 上のコスト最小の vertex-disjoint な k 本のパスを求める問題は， \bar{G} 上の最小コストのパスを求める問題に帰着されることがわかる．

3.4.2 アルゴリズムの正当性

まず Main の Step5 で作成されるグラフ G は非循環で有向なグラフであり，シンクとソースを一つずつ持っている． G から Sub1 で \bar{G} が作成される．この \bar{G} から， $\langle s, \dots, s \rangle \rightarrow \langle t, \dots, t \rangle$ へのパスが見つかる場合，補題より G に $3k$ 本の vertex-disjoint なパスが存在する．さらに，Step4 で重みづけを行い，Sub1 で G' 中のパスで最小コストなものを見つけることで，コストが ∞ 以外の場合は V_{1s} から V_{1t} へのパス， V_{2s} から V_{2t} へのパス， V_{3s} から V_{3t} へのパスそれぞれ対応がとれる．つまり $3k$ 本のパスの中で例えば V_{1s} に含まれる点を通るパスが V_{2t} に含まれる点を通るということはない．よって 初期位置 \rightarrow 出発

点 \rightarrow 到着点 への AGV の移動中どの時刻においても衝突しないような動作経路が見つかる, コストが ∞ の場合は上の例で通ることを意味するので, 衝突しない動作経路は存在しない.

逆に動作計画が求まれば経路グラフ上に k 本の vertex-disjoint パスが存在する. その k 本のパスはそれぞれ $V_{1s}, V_{2s}, V_{3s}, V_{1t}, V_{2t}, V_{3t}$ を通っている. 時刻 0 から時刻 t_1 のどの時刻においても共有する点はないので V_{1s} から V_{1t} , V_{2s} から V_{2t} , V_{3s} から V_{3t} それぞれのグループで k 本の vertex-disjoint なパスが存在する. さらに補題より, \bar{G} 上に $\langle s, \dots, s \rangle \rightarrow \langle t, \dots, t \rangle$ へのパスが存在する.

3.5 計算量について

問題のサイズを決定するアルゴリズムの入力パラメータとして,

1. 搬送要求の数 = AGV の台数 = k
2. 走行路グラフの点の数 n , 辺の数 m
3. 搬送要求の中で最も遅い到着時刻 t

がある. 経路グラフの点の数は $O(tn)$, 辺の数は $O(tm)$ である. グラフ G は経路グラフに $2k + 2$ 個の点および $6k$ 個の辺を加えたものである. また, 文献 [3] の結果により, $G = (V, A)$ において $3k$ 本の vertex-disjoint なパスを求める計算時間は $O(|V|^{3k-1}|A|)$ である. このため, このアルゴリズムで動作計画を求めるには最悪 k の指数時間かかることになる.

3.6 具体例

このアルゴリズムを使用して実際に動作計画を求める具体例を挙げる。

3.6.1 AGV が 1 台の場合

最も簡単な例を挙げる。まず，図 3.2 のような点一つで表されるステーションだけの軌道があり，AGV が 1 台そこにいる。この搬送システムに 1 つ搬送要求があり，点の名前を 1 とすると

$$\{(1, 1), (2, 1)\}$$

である。



図 3.2: 走行路グラフ (例 1)

Step1 搬送要求の中で到着時刻の最も遅い時刻は 2 である。

Step2 経路グラフは以下の図のようになる。

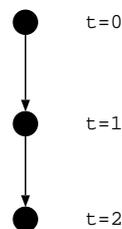


図 3.3: 経路グラフ (例 1)

Step3 各点集合は以下のようになる。

Step4 G は以下の図 3.4 のようになる。右側は左側の図を整理したものである。

Step5 重み付けを行ったものを 3.4 に示す。ただし， inf は ∞ のコストを示している。

表 3.1: 点集合 (例 1)

V_{1s}	1
V_{2s}	2
V_{3s}	3
V_{1t}	2
V_{2t}	3
V_{3t}	3

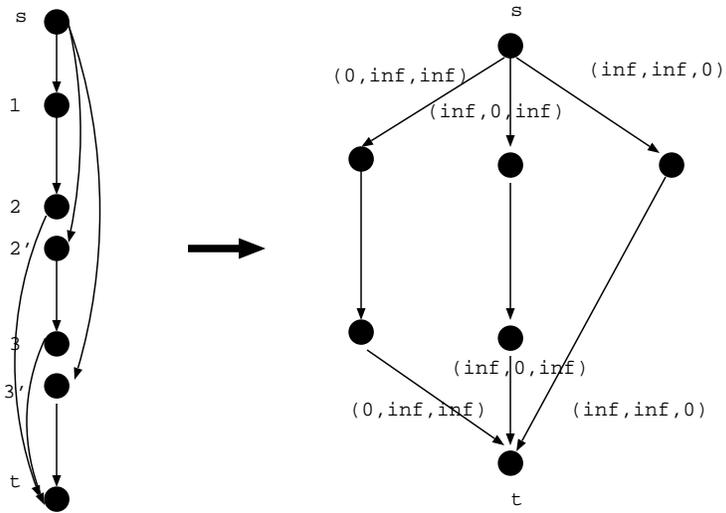


図 3.4: 経路グラフ 2(例 1)

Step6 この図 3.5は Sub2 を実行して生成されるグラフである . 重みづけは A から左側の辺は 1 からつまり s から出ている辺のつながりなので , $(0, \infty, \infty)$, $(\infty, 0, \infty)$, $(\infty, \infty, 0)$ のいずれかである . この対応は s とつながっている点によって決まる . つまり元のグラフ G で s とそれぞれつながっている点のコストがそれである . A と B の間にある辺は全て $(1, 1, 1)$ である . B より右の辺は $(0, \infty, \infty)$, $(\infty, 0, \infty)$, $(\infty, \infty, 0)$ のいずれかである .

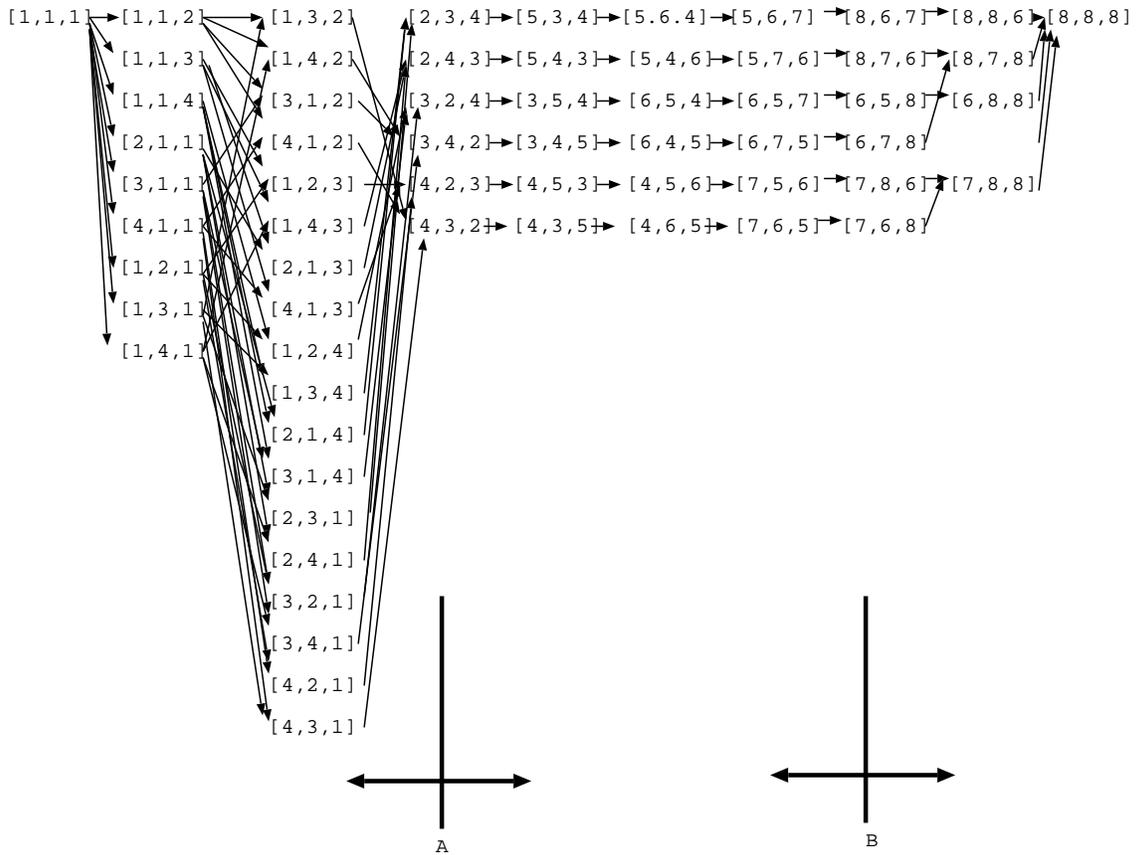


図 3.5: グラフ \bar{G} (例 1)

このグラフにおいてパスが見つかる．この場合どのパスを選んでも有限で最小コスト $(0 + 0 + 0 + 1 + 1 + 1 + 0 + 0 + 0 = 3)$ であるので，

$$\begin{aligned} &\langle 1, 1, 1 \rangle \rightarrow \langle 1, 1, 2 \rangle \rightarrow \langle 1, 3, 2 \rangle \rightarrow \langle 4, 3, 2 \rangle \rightarrow \langle 4, 3, 5 \rangle \rightarrow \\ &\rightarrow \langle 4, 6, 5 \rangle \rightarrow \langle 7, 6, 5 \rangle \rightarrow \langle 7, 6, 8 \rangle \rightarrow \langle 7, 8, 8 \rangle \rightarrow \langle 8, 8, 8 \rangle \end{aligned}$$

のようなパスを選択する．よって

初期位置から出発点へ $1 \rightarrow 4 \rightarrow 7 \rightarrow 8$

出発点から到着点へ $1 \rightarrow 3 \rightarrow 6 \rightarrow 8$

到着点からその後 $1 \rightarrow 2 \rightarrow 5 \rightarrow 8$

のようなパスをグラフ G にある．さらに 3, 5 と 4, 6 は同じ点であるのでリナンバリングする前の番号で $1 \rightarrow 2 \rightarrow 3$ となる．これはつまりそれぞれ時刻 0 の時 1 番，時刻 1 の時 1 番，時刻 2 の時 1 番にいればよいことが分かる．

3.6.2 AGV が 2 台の場合

図 3.6 のような走行路があり，AGV が 2 台 1 と 6 にいる．各搬送要求は以下の通りである．

表 3.2: 搬送要求 (例 2)

出発時刻	出発点	到着時刻	到着時刻
2	4	4	6
2	3	4	1

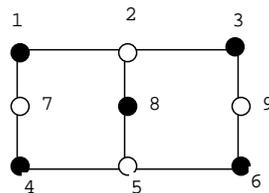


図 3.6: 走行路グラフ (例 2)

Step1 t_1 は 4 である .

Step2 時刻 0 から時刻 4 までの経路グラフは以下の通りになる . D は時刻 2 の時の 3, 4 の点 , A は時刻 4 の時の 1, 6 の点である .

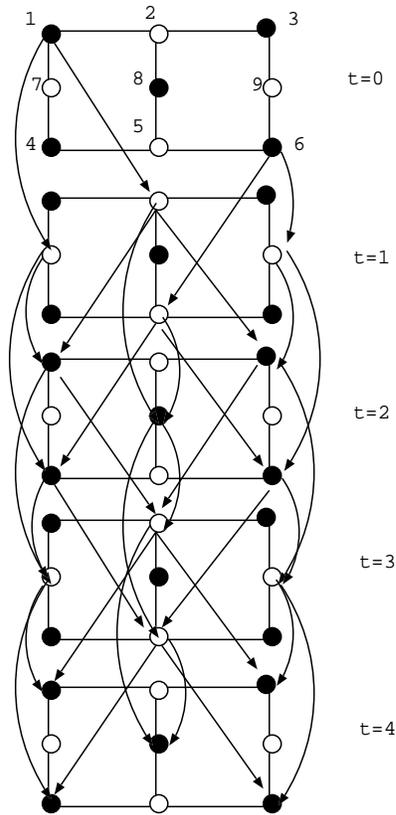


図 3.7: 経路グラフ (例 2)

Step3 V_{1s} は時刻 0 の時の AGV がいる点つまり 3, 4 で , V_{1t} は時刻 4 の時の点つまり 1, 6 である .

Step4 s, t, V_{2s}, V_{1t} をグラフに加えて , s と t に接続する辺を示し , 整理したものが図 3.8 である .

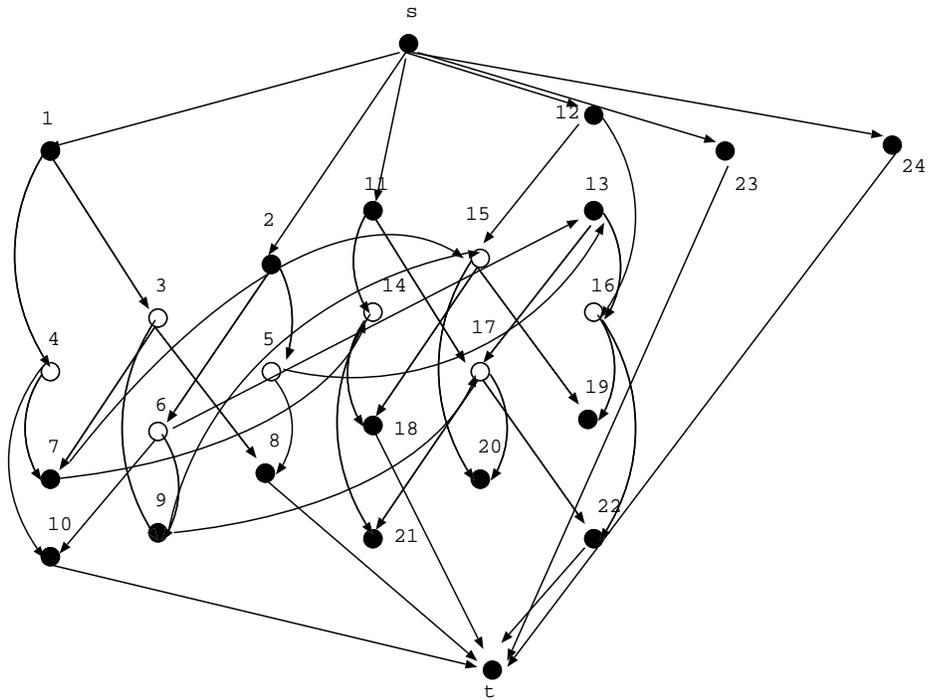


図 3.8: 整理後の経路グラフ (例 2)

Step5 各辺のコストは以下の通りである . s と t にはそれぞれ $0, 25$ の番号が割り振られ , リナンバリングされたものは図 3.8 に示してある .

$$\left\{ \begin{array}{ll} 0 \rightarrow 1 & \langle 0, 0, \infty, \infty, \infty, \infty \rangle \\ 0 \rightarrow 2 & \langle 0, 0, \infty, \infty, \infty, \infty \rangle \\ 0 \rightarrow 11 & \langle \infty, \infty, 0, \infty, \infty, \infty \rangle \\ 0 \rightarrow 12 & \langle \infty, \infty, \infty, 0, \infty, \infty \rangle \\ 0 \rightarrow 23 & \langle \infty, \infty, \infty, \infty, 0, 0 \rangle \\ 0 \rightarrow 24 & \langle \infty, \infty, \infty, \infty, 0, 0 \rangle \\ 8 \rightarrow 25 & \langle 0, 0, \infty, \infty, \infty, \infty \rangle \\ 10 \rightarrow 25 & \langle 0, 0, \infty, \infty, \infty, \infty \rangle \\ 21 \rightarrow 25 & \langle \infty, \infty, \infty, 0, \infty, \infty \rangle \\ 22 \rightarrow 25 & \langle \infty, \infty, 0, \infty, \infty, \infty \rangle \\ 23 \rightarrow 25 & \langle \infty, \infty, \infty, \infty, 0, 0 \rangle \\ 24 \rightarrow 25 & \langle \infty, \infty, \infty, \infty, 0, 0 \rangle \end{array} \right.$$

上は s もしくは t に接続している辺のコストである . それ以外は全て $\langle 1, 1, \dots, 1 \rangle$ である .

Step6,Step7 G 上における 6 本のパスは以下ようになる .

$1 \rightarrow 4 \rightarrow 10$
 $2 \rightarrow 5 \rightarrow 8$
 $11 \rightarrow 17 \rightarrow 22$
 $12 \rightarrow 15 \rightarrow 18$
 23
 24

ここで点集合はそれぞれ ,

V_{1s}	1,2
V_{2s}	11,12
V_{3s}	23,24
V_{1t}	8,10
V_{2t}	18,22
V_{3t}	23,24

であるから , 以下の 2 本の disjoint パスを作る .

$1 \rightarrow 4 \rightarrow 11 \rightarrow 17 \rightarrow 24$
 $2 \rightarrow 5 \rightarrow 12 \rightarrow 15 \rightarrow 23$

これが経路グラフ上での動作計画の経路である . 走行路のグラフの点に置き換えると ,

各 AGV の初期位置	$t = 0$	$t = 1$	$t = 2$	$t = 3$	$t = 4$
1	1	7	4	5	6
6	6	9	3	2	1

となる .

第 4 章

まとめ

4.1 結論

アルゴリズム 搬送要求数と AGV の台数が一致していて、1 台につき 1 つの要求しか満たさないとした様々な制約条件の中で、AGV 同士がお互い干渉し合わない動作計画を立てるアルゴリズムを提案した。このアルゴリズムによって得られた動作計画は単に出発点から到着点への経路だけではなく、初期位置から出発点への経路が決定されているため、要求の割り当てを行うことができ、また到着点についての AGV が動作計画が完了するまで、その後の経路が決定されているため、AGV 同士が干渉し合わないで全ての AGV がその仕事を完遂できる。

計算量 アルゴリズムの計算量は最悪、搬送要求数および AGV の台数 k の指数時間かかる。

4.2 今後の課題

本研究で提案したアルゴリズムはさまざまな仮定の元に行われる。実際の搬送システムでは、様々な条件があるため、その仮定をなるべく取り除いたものを今後も考える必要がある。

搬送要求の数と AGV の台数 ここで提案したアルゴリズムは、搬送要求の数が AGV の台数が一致している場合についてのみ成立する。AGV の台数が搬送要求の数より少ない場合、1 台の AGV が複数の要求を満たす場合がある。その場合、このアルゴリズムで仕事の割り当てができない。また AGV の台数が搬送要求の数より多い

場合も同様に余った AGV をどうするか，あるいはどの AGV を使えば要求を満たすことが出来る場合と出来ない場合があるのかを考慮する必要がある．

動的な動作計画 ここでは動作計画を AGV が移動する前にあらかじめ決定し，その計画が途中で変更が加えられないことを仮定している．従って，実際の搬送システムを考慮すると動的な計画変更に耐えうるアルゴリズムを考える必要がある．

計算量 アルゴリズムで動作計画を求めるには最悪 k の指数時間かかる．このため，AGV の台数あるいは搬送要求の数によって実時間で解ける問題は限られてしまう．従って実際にこのアルゴリズムを使って動作計画を立てるためには，多項式時間で解けるようにアルゴリズムを工夫，あるいは創作する必要がある．

Appendix

補題の証明

(\Rightarrow)

k 本の s から t への vertex-disjoint なパス $P_1 \cdots P_k$ が与えられるものとする．それぞれのパスは

$$P_i = u_1^{(i)} \rightarrow u_2^{(i)} \rightarrow \cdots \rightarrow u_{r_i}^{(i)} \\ i = 1 \cdots k$$

と表す．但し $u_1^{(i)} = s, u_{r_i}^{(i)} = t$ である．

ここで， $P_1 \cdots P_k$ 上の arc を並べ替える．並べ替えのルールは arc の tail を小さい順にする．但し，tail が同じ場合その arc が存在するパスの番号が小さい順に並び替える．そして，並べ替えた順にそれぞれの arc を

$$w_1 \rightarrow z_1, w_2 \rightarrow z_2, \cdots, w_\rho \rightarrow z_\rho$$

とする．ここで k 本のパスが与えられているから $w_1 \cdots w_k$ までは全て s である．なぜなら一番小さい tail の組の要素は s であり，それが k 個存在するから上で述べたような並べ替えを行なえば，

$$s = w_1 = w_2 = \cdots = w_k$$

となる．また $w_{k+1} \cdots w_\rho$ にはソースは存在せず，また vertex-disjoint なパスが与えられているから同一のものは存在しない．よって

$$s = w_1 = w_2 = \cdots = w_k < w_{k+1} < \cdots < w_\rho$$

である．また ρ は arc の数に等しい．その値はパス 1 本につき $r_i - 1$ 本の arc があるから k 本では各パスの arc の数の summation に等しいので，

$$\rho = \sum_{i=1}^k (r_i - 1)$$

である .

ここで \bar{G} 上に \bar{P} というパスを定義する . \bar{P} を定義するために \bar{P} に存在する arc の $a_1 \cdots a_\rho$ を定義する .

$a_1 \cdots a_k$ について

$$\begin{aligned} a_1 &= \langle w_1, s, \cdots, s \rangle && \rightarrow \langle z_1, s, \cdots, s \rangle \\ a_2 &= \langle z_1, w_2, \cdots, s \rangle && \rightarrow \langle z_1, z_2, \cdots, s \rangle \\ &\vdots && \vdots \quad \vdots \\ a_k &= \langle z_1, z_2, \cdots, w_k \rangle && \rightarrow \langle z_1, z_2, \cdots, z_k \rangle \end{aligned}$$

$a_{k+1} \cdots a_\rho$ について

$$\begin{aligned} a_i &= \langle x_1, \cdots, x_{j-1}, w_i, x_{j+1}, \cdots, x_k \rangle \rightarrow \langle x_1, \cdots, x_{j-1}, z_i, x_{j+1}, \cdots, x_k \rangle \\ i &= k+1, \cdots, \rho \end{aligned}$$

ここで $\langle x_1, \cdots, x_{j-1}, x_j, x_{j+1}, \cdots, x_k \rangle$ は a_{i-1} の head である . なぜなら \bar{P} は a_1, \cdots, a_ρ で表されるパスであるから , a_{i-1} の head が a_i の tail でなければ \bar{P} はパスにならない . よって a_{i-1} の head は a_i の tail である .

また , $w_i = \min\{x_1, \cdots, x_k\}$ である . なぜなら , まず , a_1 については , その tail は全て s の要素からなる組であり , どの要素もその組で最小であり , $w_1 = s$ であるから w_1 は a_i の中に含まれており , 最小である .

$a_i (i = 1, \cdots, i-1)$ はその tail の組の中に w_i が含まれていて一番小さく , その head は w_i を z_i に置き換えたものであると仮定する . ここで x_1, \cdots, x_k つまり a_i の tail の組の要素はパス上にある点のどれかであるから ,

$$x_1, \cdots, x_k \in \{w_1, \cdots, w_\rho, t\} \quad (4.1)$$

である . なぜならパス上にある arc の tail は w_1, \cdots, w_ρ で表され , 唯一 tail になりえない t も含まれるからである . また , a_{i-1} の tail で w_{i-1} は最小であるから , それを z_{i-1} に置き換えた a_{i-1} の head のどの要素も w_{i-1} よりも大きい . 従って a_{i-1} の head は a_i の tail であるから a_i の tail つまり

$$x_1, \cdots, x_k > w_{i-1} \quad (4.2)$$

である .

4.1,4.2より ,

$$x_1, \cdots, x_k \in \{w_i, \cdots, w_\rho, t\} \quad (4.3)$$

であると言える．ここで $j < k < i$ であるような a_j, a_k の時にまた a_j の head が a_k の tail である場合，

$$\begin{aligned} a_j &= \langle \cdots w_j \cdots \rangle \rightarrow \langle \cdots z_j \cdots \rangle \\ a_k &= \langle \cdots w_k \cdots \rangle \rightarrow \langle \cdots z_k \cdots \rangle \end{aligned}$$

のように a_j の tail の要素である w_j が z_j に置き換えられる．この時その $z_j = w_i$ であった場合， $a_i (i = 1, \dots, i-1)$ では， w_i が置き換えられるから w_i は a_i まで置き換えられることはない．従って， w_i は a_i の tail に存在する．すなわち，

$$w_i \text{ が } x_1, \dots, x_k \text{ に存在する.} \quad (4.4)$$

4.3, 4.4より $w_i < w_{i+1} < \cdots < w_\rho < t$ であることから w_i は x_1, \dots, x_k に含まれ，すなわち w_i は a_i の tail に含まれ，その中で最小である．よって $a_i (i = 1, \dots, \rho)$ の tail において w_i は最小であるといえる．

以上により a_1, \dots, a_ρ は定義され， \bar{P} は \bar{G} 上にある $\langle s, \dots, s \rangle \rightarrow \langle t, \dots, t \rangle$ のようなパスであり， $l_j (j = 1, \dots, k)$ のような j 番目の dimensional な arc の length の合計をコストにもつ．なぜなら， \bar{P} 上にある各 arc のコストは組の要素を置換するのに使う A 上の arc のコストに等しく，またパスのコストは各 arc のコストの合計であるからである．

(\Leftarrow)

\overline{G} 上にパス

$$\overline{P} \equiv \langle x_1^{(1)}, \dots, x_k^{(1)} \rangle \rightarrow \langle x_1^{(2)}, \dots, x_k^{(2)} \rangle \cdots \rightarrow \langle x_1^{(\rho)}, \dots, x_k^{(\rho)} \rangle$$

が存在することを仮定する .

ここで , \overline{G} の定義から $i = 1, \dots, \rho$ についてまた $j \in \{1, \dots, k\} \setminus \{\hat{j}\}$ であるようななどの j についても

$$\begin{aligned} x_{\hat{j}}^{(i)} &\rightarrow x_{\hat{j}}^{(j+1)} \in A \text{ かつ} \\ x_{\hat{j}}^{(i)} &= x_{\hat{j}}^{(i+1)} \end{aligned}$$

となるような $j \in \{1, \dots, k\}$ が存在することが言える .

$x_{\hat{j}}^{(i)} \rightarrow x_{\hat{j}}^{(j+1)} \in A$ であるのは ,

今 \overline{P} のパス上の arc は \overline{A} の定義からその tail の組の 1 要素を置換した組を head として
いる . その 1 要素が \hat{j} 番目の点であった場合 , arc の tail の \hat{j} 番目の要素は $x_{\hat{j}}^{(i)}$ で , head
の \hat{j} 番目の要素は $x_{\hat{j}}^{(i+1)}$ である . \overline{A} の定義から $x_{\hat{j}}^{(i)}$ から $x_{\hat{j}}^{(i+1)}$ への arc が A に存在するの
で , $x_{\hat{j}}^{(i)} \rightarrow x_{\hat{j}}^{(i+1)} \in A, i = 1, \dots, \rho, j \in \{1, \dots, k\}$ のような \hat{j} は存在する .

さらに $x_{\hat{j}}^{(i)} = x_{\hat{j}}^{(i+1)}$ であるような \hat{j} が存在するのは

\hat{j} 番目の要素だけを変えた組から組への arc で \overline{P} は存在しているから組の \hat{j} 番目以外の要素
はそのままであるから $x_{\hat{j}}^{(i)} = x_{\hat{j}}^{(i+1)}$ となるような \hat{j} も存在する .

以上により , $\overline{P} \equiv \langle x_1^{(1)}, \dots, x_k^{(1)} \rangle \rightarrow \langle x_1^{(2)}, \dots, x_k^{(2)} \rangle \cdots \rightarrow \langle x_1^{(\rho)}, \dots, x_k^{(\rho)} \rangle$ は G 上の s から t までの vertex-disjoint なパスである . なぜなら ,
まず先に述べたように , \overline{G} の定義から

$$\begin{cases} x_m^{(i)} \rightarrow x_m^{(i+1)} \in A \\ i = 1, \dots, \rho \\ m = 1, \dots, k \end{cases}$$

のような arc が存在する . つまり ,

$$\begin{aligned} x_m^{(1)} &\rightarrow x_m^{(2)} \\ x_m^{(2)} &\rightarrow x_m^{(3)} \\ &\vdots \\ x_m^{(\rho)} &\rightarrow x_m^{(\rho+1)} \\ m &= 1, \dots, k \end{aligned}$$

のような arc が存在する . また A の定義より $x_m^{(i)} < x_m^{(i+1)} (m = 1, \dots, \rho)$ であるから
 $x_m^{(1)} \leq x_m^{(2)} \leq \cdots \leq x_m^{(\rho)}$ である .

ここで P_1, \dots, P_k のパスが vertex-disjoint ではない場合つまり下のような x が再び置換される場合を考える .

$$\langle \dots x \dots \dots \rangle \quad (4.5)$$

⋮

$$\langle \dots \dots x \dots \rangle \quad (4.6)$$

4.6のような組で , その前の組から x に置換されたということは 4.5から 4.6の間では x より小さい要素が置換されてきたことになる . ということは 4.5における x は置換される要素ではないはずなので 4.6まで保存されるから 4.6で x が 2 つ存在することになる . これは \bar{V} の定義に反するため , P_1, \dots, P_k は G 上の s から t までの vertex-disjoint なパスである .

さらに $c^{(j)}(P_j)$ は $j = 1, \dots, k$ について \bar{P} の j 番目の dimensional な arc の length の合計である . なぜなら \hat{A} に存在する arc の length の定義は $x_j^{(i)} \rightarrow x_j^{(i+1)} \in A$ のコストに等しい . よってパス P_j のコストは $x_j^{(i)}, \dots, x_j^{(\rho)}$ までの arc のコストの合計である .

謝辞

最後に本研究を進めるにあたり，指導教官であり，北陸先端科学技術大学院大学情報科学研究科の平石 邦彦助教授には，終始様々な助言，御指導を頂きました．また同研究室の宋 少秋助手，高島 康裕助手にも適切な助言，指導を頂きました．ここに深く感謝の意を表します．

参考文献

- [1] 佐々木 淳, 増山 繁, 山川 栄樹, ”AGV システムにおける最悪移動完了時間の理論的解析” 電子情報通信学会論文誌, A Vol.J79-A No.8 pp.1433-1443 1996.8
- [2] 佐々木 淳, 増山 繁, 山川 栄樹”AGV システムにおける許容台車数の理論的解析”, 電子情報通信学会論文誌 A Vol.J78-A No.10 pp.1341-1347 1995.10
- [3] Chung-Lun Li, S.Thomas McCormick, David Simchi-Levi: ”Finding Disjoint Pathes with Different Path-Costs: Complexity and Algorithms”, NETWORKS , Vol.22 ,(1992)653-667
- [4] 宝崎 隆祐, 藤井 進, “物流システムにおけるスケジューリング” システム制御情報学会誌 Vol.37,No.6 pp.344-349 1993
- [5] JAN VAN LEEUWEN , “コンピュータ基礎理論ハンドブック I アルゴリズムと複雑さ” 丸善株式会社 p.554-557