自動搬送台車の経路探索問題に対する近似解法の提案

時田　陽輔　（沼田　一道　助教授）

１．はじめに

製品を製造する工場では，その過程で原材料，部品，半製品の移動が不可欠であり，そのための装置の設置・運用費用は，製造費用の大きな割合を占める．従来の少品種多量生産では，ベルトコンベア上に半製品を流し，それに沿って加工を加える形態が主流であった．しかし多様化した顧客の好みに合わせて，多品種少量生産を余儀なくされる現代の工場では，多用な加工の種類・順序に合わせて自由な搬送経路の設定や，加工対象に応じた経路の選択が必要となってきている．近年，このような要求に応えるものとして現れたのが，自動搬送台車システム（Automated Guided Vehicle System, 以下ＡＧＶシステムと呼ぶ）である．

本研究ではＡＧＶシステムにおける経路決定問題について考える．

２．ＡＧＶシステム

ＡＧＶシステムは，複数の加工場所，部品，半製品，半製品置き場をつなぐ走行線路と，指令所からの指令を受けてその上を走る台車から構成される．

２．１　台車

台車は荷台を持ち，そこに仕掛品などを載せて搬送する．動力源は内臓の蓄電池で，線路（誘導線の埋められた床面）上を指令に従って自由に走行できる．

２．２　線路

実際のＡＧＶシステムの構成は様々であるが，本研究では，図１のような格子状のネットワークを想定する．

点（ノード）は加工場所や置き場所，あるいは線路の分岐点を表す．アークは走行可能線路である．台車は各ノードにおいて，そこから出ているアークのどの方向へも移動することができる．複数の台車がネットワーク上の適当な位置に待機しており，搬送要求が生ずると，（指令により）出発ノードへ移動し，そこで荷を積んで目的ノードへ搬送し，適当な待機ノードへ戻る．

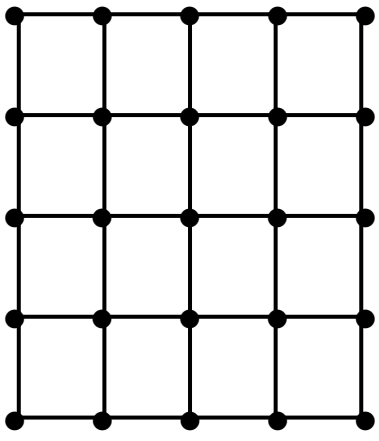


図１　経路の例

２．３　ブロッキングとデッドロック

一般に複数の搬送要求が発生したとき，指令センターは複数の台車に経路を割り当てる必要があるが，複数台の台車が同時に走行するため，経路の組合せによって台車同士が干渉する可能性がある．干渉は複数の台車が同時に同じノード，アーク上に存在してしまうことである．実際には，「ある時間（e）未満内に同一ノード，アーク上に存在すること」を禁止する．

台車間の干渉が起こるのは，つぎのいずれかの場合である[１]．

＜条件１＞２台以上の台車が，同一ノード上にeという時間差内で存在する場合（図２）．

＜条件２＞２台以上の台車が，同一アーク上を，同時に互いに向かい合って走行する場合（図３）．

条件１，２が経路の途中で起こる場合「当該ノードの手前で，一方が他方の通過を待つ（条件１）」

（図４），「共有パス（ノードとアークの列）の端点の手前のノードで一方が他方を待つ（条件２）」（図５）ことにより回避できる．これらは，一方が他方の経路を一時的に塞ぐもので，ブロッキングと呼ばれる．また，共有パスに両者の出発ノードが含まれるような場合には，干渉が回避不能になる.これはデッドロックと呼ばれ，この場合の経路の組合せは実行不能である．

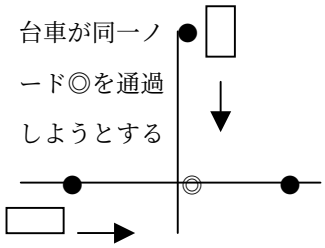
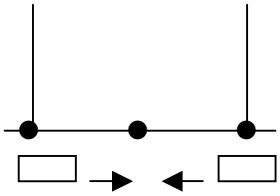
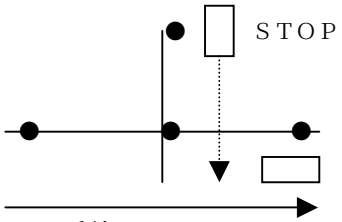
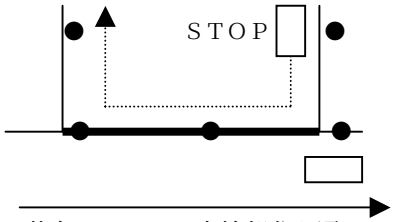


図２　条件１の場合


同一アーク上を向かい合って通過しようとする
図３　条件２の場合



手前のノードで待ち時間を設ける
図４　条件１の解消方法



共有パスである太線部分を通過するまで待ち時間を設ける
図５　条件２の解消方法

３．研究目的

ＡＧＶシステムにおいて複数個の搬送要求（出発，目的ノードの組）が与えられたとき，その個数分の台車に対する走行指令を決定（作成）する問題は，つぎのような最適化問題として定式化される．

> 台車間の干渉（衝突）が起きないという制約の下で，総所要時間（各台車の走行時間と待ち時間の和）が最小となる経路と経路上のノードにおける待ち時間を各台車に割り当てる．

この問題に対して，[１]は待ち時間の決定に線形計画法を利用したアプローチを，また[２]はメタヒューリスティックスによる解法を提案している．本研究では，[１]に沿って，格子状ネットワークの場合の問題と解法を考える．

４．本研究で扱う経路決定問題

図１のような格子状ネットワーク上で，m 個の搬送要求 $q=1,2,\ldots,m$ が m 台の台車に与えられた状況を考える．

本研究では，つぎのような前提を置き，以下の記号を導入する．

[前提] ｉ．搬送要求は時刻０において，一斉に与えられる．ii．各台車は時刻０において，出発ノードに位置している．iii．各台車の速度は同一である．iv．各リンクの通過に要する時間はリンクと方向によらず一定（a 単位時間）である．ｖ．ノードの通過に要する時間は直進，右左折によらず無視できる．vi．経路は最短経路（回り道の無い経路）のみを対象とする．

[記号] 搬送要求番号　$q,r,\ldots$（$q,r,\ldots\in M=\{1,2,\ldots,m\}$）．

搬送要求　$Q: Q=\{(o_1,d_1),\ldots,(o_q,d_q),\ldots,(o_m,d_m)\}$　ただし o_q，d_q は第 q 搬送要求の出発点と目的点．

経路集合　$PATH_q$ ：o_q から d_q への経路全体の集合．$(q=1,2,\ldots,m)$

経路組合せ　$\pi=\pi_1,\ldots,\pi_q,\ldots,\pi_m$　ただし，$\pi_q\in PATH_q$．前提vi.より $PATH_q$ に属する経路の長さ（ノード数）は等しい．これを，L_q で表す．

経路通過ノード　$\pi_q(i)$ ：o_q から d_q へ向かう経路 π_q が i 番目に通過するノード．

ただし，　$\pi_q(0)=o_q$，　$\pi_q(L_q)=d_q$．

ノード休止時間　$x_q(\pi,i)$：経路組合せπのもとで，経路π_qがその第iノードで休止する時間.

以上により，「m個の搬送要求Qが与えられたとき，要求を満たし，走行時間と待ち時間の和を最小とする経路と待ちノード(待ち時間)をm台の台車に割り当てる」問題は次のように定式化される.

(問題)

$$\text{minimize} \quad \sum_{q=1}^{m}\sum_{i=0}^{L_q-1} x_q(\pi,i)$$

$$\text{sub.to} \quad \forall q,r \in M,\ 0 \leqq \forall i \leqq L_q,\ 0 \leqq \forall j \leqq L_r$$

$$\textbf{if}\ \pi_q(i)=\pi_r(j)\ \textbf{then}\ \left|\sum_{k=0}^{i-1} x_q(\pi,k)+ia-\sum_{l=0}^{j-1} x_r(\pi,l)-ja\right| \geqq e$$

$$i,j \in (i \neq L_q,\ j \neq L_r)$$

$$\textbf{if}\ \pi_q(i)=\pi_r(j)\ \textbf{then}\ \sum_{l=0}^{j-1} x_r(\pi,l)+ja \leqq \sum_{k=0}^{i-1} x_q(\pi,k)+ia-e \quad (i=L_q)$$

$$\textbf{if}\ \pi_q(i)=\pi_r(j)\ \textbf{then}\ \sum_{k=0}^{i-1} x_q(\pi,k)+ia \leqq \sum_{l=0}^{j-1} x_r(\pi,l)+ja-e \quad (j=L_r)$$

$$\pi_q \in PATH_q \quad (q=1,2,...,m),\quad x_q(\pi,i) \geqq 0 \quad (q=1,2,...,m\ ;\ i=0,1,2,...,L_q-1)$$

台車の走行時間は前提vi.より定数となるので，待ち時間だけを最小化している．[１]では，これと同様の問題を，経路対毎の共有パスの有無，通過待ちの優先順位で場合分けし，線形計画問題として扱うことを提案している．しかし，経路の選択，デッドロックの有無の判断，優先順位の場合分け等組合せ的探索を避けられないので，解くべき線形計画問題の数は極めて大きくなり，あまり実際的とは思われない．また，デッドロックの判別等については何も触れられていない.

そこで本研究では，待ち時間の設け方を含めて，近似的探索を行う解法を提案する.

５．提案する解法

簡単化のために，さらに次のような仮定，方針を採用する．①$a=e=1$とする．②ノードでの待ち時間はeの倍数とする．③m個の搬送要求の出発点と目的点は相異なる．④経路集合πの２経路内に，逆向きの共有アーク部分が２箇所以上ある場合には，この組合せを探索対象から除く.

入力：搬送要求　$OD_1=(o_1,d_1), OD_2=(o_2,d_2),...,OD_m=(o_m,d_m)$　o…始点　　d…終点

出力：デッドロックの有無，ブロッキングを回避した走行計画（o_1からd_1への経路＋待ちノード＋待ち時間）

[解法]Step1．各搬送要求（OD_1～OD_m）についてo_q，d_q内の最短経路を全て求める.

OD_qに対する最短経路の集合をSP_qとする.

Step2．$minTWT \leftarrow \infty$とする.

Step3．$\forall \pi=(\pi_1,\ldots\pi_m)$, $\pi_q \in SP_q$について以下を行う.

優先順位（$\pi_{p1},\ldots\pi_{pm}$），$(\forall p1,...,pm) \in S_m$について，以下を行う.

①π_{pk}（$k=2,...,m$）について，デッドロックの判定を行いながら，

(π_{p1}～π_{pk-1})と干渉しないようにπ_{pk}に待ち時間を設ける.

②待ち時間が加えられたら，①を繰り返す.

③待ち時間の追加が無くなったとき，

$TWT \leftarrow$現在の経路集合の優先順位における総待ち時間

P ←走行計画

if $minTWT > TWT$ **then** $minTWT \leftarrow TWT$, $bestP \leftarrow P$

Step4．終了．　$minTWT$ …最小待ち時間，$bestP$ …最良走行計画

６．プログラムの作成

$m = 3$ とし，４×４，５×５の格子状ネットワークで上記の解法を実行するプログラムを作成した．プログラムには Delphi３.１を用いた．

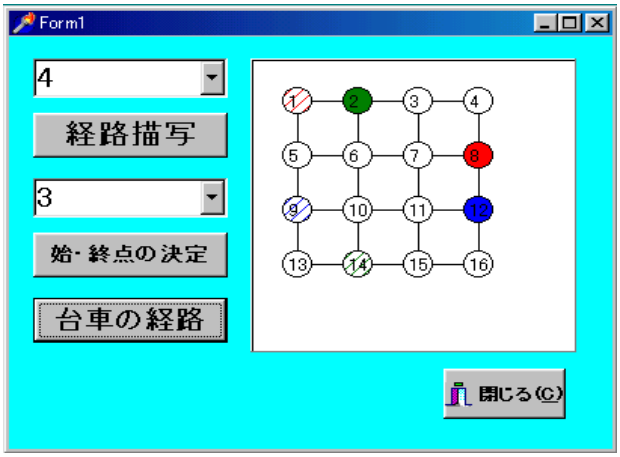


図６　実行画面

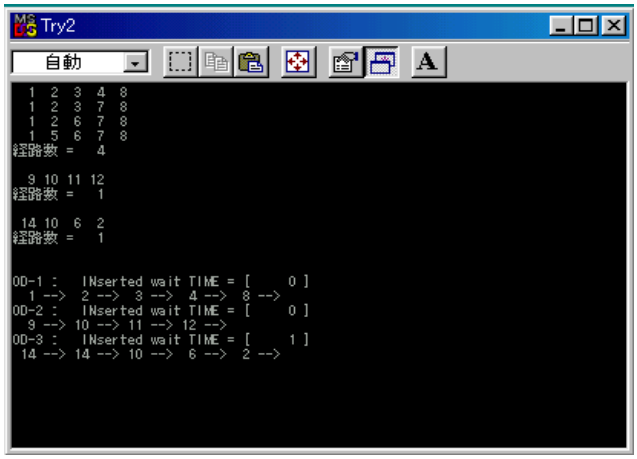


図７　実行結果

７．結果と考察

例題を繰り返し解いた結果，デッドロックは完全に検出された．ブロッキングについては，１ヶ所待ち時間を設けることで回避できる場合でも，更に他の場所で待ち時間を設けてしまうことがある．これは，待ち時間を２台の台車間で考えているために起こるものと推測される．待ち時間を設け終えた後，全体でもう一度考慮するべきであった．精度を上げるためにアルゴリズムを再検討する必要がある．

現時点では完全なシステムであるとは言えないが，始・終点の入力のみで，短時間で結果を出力できるという結果から，[１]の決定法よりも実際的であると思われる．

８．まとめ

本研究では，格子状のネットワークに限定して，ＡＧＶシステムの経路決定問題を考え，近似解法を提案し，プログラムを作成した．ネットワークのサイズや台車数など限られた条件のもとではあるが，始終点を入力するだけで，ほぼ同じ走行計画が得られるという点で，試作品として意義あるものと考える．例題を通して発見された問題点を改善し，どんな搬送要求にも対応できるシステムを作り上げることが課題である．また今後は更なる実用性を考え，回り道をして干渉を回避する方法や，連続した搬送要求に対応できるアルゴリズムを検討する必要もあると考える．

【参考文献】

[１] 藤井進，三道弘明，宝崎隆祐：自動搬送台車の経路決定法；日本機械学会論文集（Ｃ編），５５巻，５１４号（１９８９－６）

[２] 遠藤真一郎，小西正躬，森脇俊道，吉田正義：大規模搬送システムにおける遺伝的アルゴリズムを用いた移動ロボットの経路検索；システム制御情報学会論文誌，Vol１３，No.３，pp.１１５－１２３（２０００）