タスクの終点と始点の接続性を考慮した複数台 AGV の配送スケジューリング

Transport scheduling for multiple AGVs considering connectivity between end and start points of tasks

○学　目野 航輝 (筑波大)
正　萬 礼応 (筑波大)　　大矢晃久 (筑波大)

Koki MENO, Tsukuba University, meno-k@roboken.cs.tsukuba.ac.jp
Ayanori Yorozu, Tsukuba University
Akihisa Oya, Tsukuba University

In this paper, a method for the AGV transportation scheduling problem is proposed. In actual workplaces, additions and changes to transportation tasks occur due to production line problems and delays. Therefore, it is necessary to quickly plan a feasible transportation schedule with no delays in specified time when tasks are added and changed. This study focuses on to quickly find a transportation schedule with no delay from the specified time and how to continue searching for a better solution during the execution of the transportation tasks. the structure of the solution that considers the order of tasks for each AGV and a scheduling method that considers the connectivity of end points of tasks and start points of next tasks are proposed. In order to verify the effectiveness of the proposed method, numerical simulations were carried out with assuming real transportation tasks.

***Key Words*:** multiple AGV, scheduling, optimization

1　緒言

無人配送車 (AGV: Automatic Guided Vehicle) とは運転操作を必要としない自動運転車のことで，主に物資を配送するために使われる．製造システム，コンテナターミナル，倉庫システムなどのサービス産業では，人手不足や作業の効率化などの理由から物資の配送に AGV を採用し，生産の柔軟性と効率を維持している．

工場や物流倉庫での AGV の配送では，指定の期間の間に指定の位置にある物資を指定の場所まで配送することが求められる．物資を指定された時間内に配送するなどの制約を満たしながら，複数台の AGV を効率的に運用するためには，配送タスクのスケジューリングが重要である．実際の現場では，生産ラインのトラブルや遅れなどによりタスクの追加や変更が起こるため，追加や変更が起こった場合に素早く実行可能解：配送要求時刻から遅れのない配送スケジュールを求める必要がある．また、タスクの追加や変更が起きた場合にも対応しやすくなるため、配送がより早く終わるスケジュールを求めることも重要である. 本研究では, 素早く実行可能解を求め、実行可能解が得られた後も配送がより早く終わるスケジュールを探索し続ける方法に着目する．

配送タスクの時間制約を考慮して複数台 AGV のスケジューリングを行う場合，タスクをどの AGV に割り当て，どの順番で実行するかを計画する必要がある．複数台 AGV のスケジューリング問題は NP 困難な問題であるため，ヒューリスティック手法である遺伝的アルゴリズムを用いた手法が多く提案されている [1,2]．しかし，従来手法の遺伝子の構造と AGV へのタスクの割当と順序を無作為に探索するアルゴリズムでは，指定された時間内に物資を配送可能な配送スケジュールを得ることさえ難しい場合がある．

本研究では，タスク間の回送時間を減らすことで素早く次のタスクを開始することができるため，タスクの終点と始点の接続性が重要であると考え，AGV ごとにタスクの実行順を考慮可能な解の構造とタスクの終点と始点の接続性を考慮したスケジューリング手法を提案し，シュミレーションで評価する．

2　問題設定

本研究では工場や物流倉庫における AGV の配送を想定しスケジューリングを行うことを考える．本研究で想定する環境を Fig. 1 に示す．全ての物資は棚に積まれて保管されており，棚は Fig. 1 の棚の保管場所と搬入搬出口のいずれかに置かれている．ここで，指定の物資が積まれた棚を指定の場所まで配送することを配送タスクとする．また，指定の物資が積まれた棚が置かれている場所をタスクの始点，配送先をタスクの終点とする．工場や物流倉庫などでは，必要な物資を必要なときに配送することが要求される．そのため，各配送タスクには指定の時刻から指定の時刻までに配送するように配送終了要求期間が与えられ，AGV は終了要求期間の間に物資を配送することが求められる．配送タスクには同じトラックが搬出する物資の配送や同じ作業に使う物資の配送などがあり、配送の種類ごとのグループに分けられる．各グループのタスクは一つでも遅れてしまうとその後の作業に影響が出てしまうため，終了要求期間はグループごとに与えられる．AGV は割り当てられた配送タスクに従って，終了要求期間の間に始点から終点まで棚を配送することが要求される．本研究ではこれらの条件の中で，あるタスク群を与えられた際に素早く実行可能解が求められ，固定の探索回数内で配送がより早く終わる解を得られるか検証する．

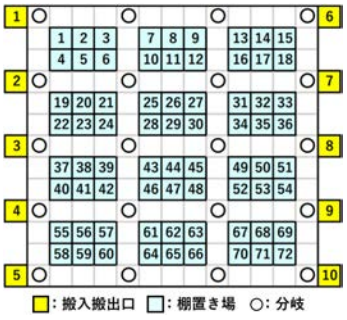

Fig.1 想定する環境

3　関連研究

複数台 AGV の配送スケジューリングは配送タスクの数や AGV の数，倉庫の規模が大きくなるにしたがって，実行可能なスケジュール数が指数関数的に増加するため，NP 困難な問題である．そのため，厳密解を求めることは困難であり，近似解を求める方法が研究されている．この近似解を求める方法として数理計画による手法 [3] とヒューリスティック手法がある．数理計画による手法では，今回のような離散的な問題に対して膨大な計算量が必要になるため，より効率的に解を求める方法としてヒューリスティック手法が研究されている [4]．AGV スケジューリングの研究にメタヒューリスティックアルゴリズムの一つである遺伝的アルゴリズムを用いた手法の研究が多く行われている [1,2]．遺伝的アルゴリズムは，最初に解の初期集合を作成しその集合に対して交叉，突

然変異，評価，選択を繰り返し行い近似解を求める手法である [5]. AGV の配送タスクスケジューリング問題に対しての解の構造として，タスクの実行順を表した列とタスクへの AGV の割り当てを表した列を合わせて一つの解を表す方法が用いられている [1,2]. [2] の解の構造を本研究の問題に適応した場合の例を Fig. 2 に示す. Fig. 2 の解の構造の例では AGV1 が 1，11，7，5，AGV2 が 6，4，2，12，AGV3 が 8，3，9，10 の順にタスクを実行することを表す．しかし Fig. 2 のような解の構造に交叉や突然変異を用いて解を生成した場合，親の性質を引き継いだ解が生まれにくく効率的に解を探索することができない.

そこで本研究では AGV ごとのタスクの実行順とタスクの終点と次に行うタスクの始点の接続性が重要であると考え，AGV ごとのタスクの実行順を考慮可能な解の構造と，タスクの終点と始点の接続性を考慮した入れ替えによる解の生成方法を用いた探索アルゴリズムを提案する．この解の構造と入れ替え用いることで親の良い性質を保ちながら解を生成でき，効率的な探索ができると考えられる.

タスクの実行順	6	1	4	8	3	11	9	2	7	5	12	10
AGVの割り当て	2	1	2	3	3	1	3	2	1	1	2	3

Fig.2 [2] の解の構造

4　提案手法

本論文ではタスクの AGV ごとの実行順を考慮可能な解の構造と，タスクの終点と始点の接続性を考慮した入れ替えを用いた探索アルゴリズムを提案する．提案する解の構造を 4.1 に示し，探索アルゴリズムのフローを 4.2 に示す．また，タスクの終点と始点の接続性を考慮して解を生成するための入れ替え方法を 4.3 で述べる.

4.1　AGV ごとの実行順を考慮可能な解の構造

本研究では AGV ごとに配送タスクを実行する順に並べた 2 次元配列によって解を構成する．タスクを AGV ごとに実行順に並べることで，入れ替えによって時間制約を満たす解を見つけやすくなる．また，タスクの始点と終点の接続性を考慮しながら入れ替えを行うことができる．AGV3 台に 6 つの配送タスクを割り当てた例を Fig. 3 に示す．この例では AGV1 はタスクを 4，3，1 の順に実行し，AGV2 はタスク 6 を実行し，AGV3 はタスクを 2，5 の順に実行する.

搬送タスク

AGV	[1]	[2]	[3]	[4]	[5]	[6]
[1]	4	3	1			
[2]	6					
[3]	2	5				

Fig.3 提案する解の構造

4.2　探索アルゴリズムのフロー

提案する探索アルゴリズムのフローを Fig. 4 に示す．最初に，タスクの入力を受け取り解の集合を作成する．このとき一つ一つの解は 4.1 に示したタスクの実行順を考慮可能な解の構造を用いて構成する．作成した解の集合を初期集合として，入れ替え，評価，選択を繰り返し行い解を探索する. まず，4.3 に示すタスクの始点と終点の接続性を考慮した入れ替え方法を用いて集合に含まれる解を入れ替え，新しい解を作成する. そして，作成した各解の評価値を 4.5 に示す解の評価関数を用いて計算する．入れ替えによって生まれた解と入れ替える前の解を合わせた集合の中から，各解の評価値をもとに 4.6 に示す解の選択方法に従って，次の集合を構成する解を選択する．ここで実行可能解が含まれていれば，選択した解の中で一番評価値の良い解を出力する．これを繰り返しおこない，評価値の良い解を探索する，

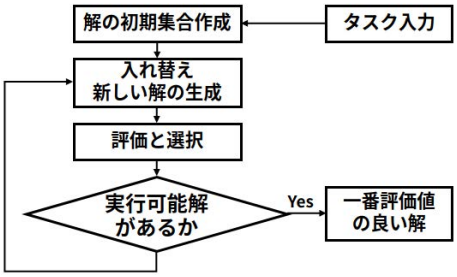

Fig.4 探索アルゴリズムのフロー

4.3　タスクの始点と終点の接続性を考慮した入れ替え

入れ替えアルゴリズムについて述べる．まず，配列の中から入れ替える点を 2 点選択する．この時，2 つのタスクの位置を交換するだけではなく，選択されたタスクの前後にタスクを挿入する操作も加えたいため，各タスクとその前後の点の中からランダムに選択することとする．2 点を選択した例を Fig. 5 に示す．Fig. 5 の左の例では二つのタスクが選択されており，この 2 つのタスクの位置を交換する操作を行う．Fig. 5 の右の例では一つのタスクとタスクの間の点が選択されており，選択されたタスクをタスクの間の点に挿入する操作を行う.

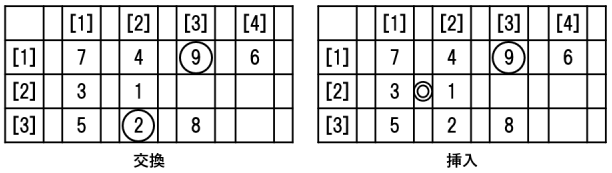

Fig.5 選択

選択されたタスクの位置を単純に入れ替えるだけではタスク間の接続性がよかった場合その関係が崩れてしまうため，前後のタスクとの接続性の良さを結合度として表し，結合度をもとに前後のタスクを一緒に入れ替えることを考える．前のタスクの終了地点から次のタスクの保管場所までの回送時間が短いほど次のタスクの実行時間が短くなるため，この時間の短さを結合度として表す．ここで配送タスクの番号を i，j，配送タスク i を T_i，T_i のの終了要求期間の開始時刻，終了時刻を S_i，E_i，配送での経路の長さの最大値を M，T_i の終了地点から T_j の棚までの移動時間を D_{ij}，T_i の棚の置き場所から T_i の配送先までの移動時間を D'_i，T_i の次に T_j を実行するとき T_j にかかる合計時間を $TD_{ij}(TD_{ij} = D_{ij} + D'_j)$ とする．T_i の次に T_j を実行するとき，結合度の大きさを変えるパラメータを p とし T_i，T_j の結合度を C_{ij} とすると，結合度 C_{ij} は (1) の式で与える.

$$C_{ij} = \begin{cases} 0 & if\ S_i + TD_{ij} > E_j \\ \left[1 - \frac{D_{ij} + \{S_j - (E_i + TD_{ij})\}}{M}\right] p & if\ S_j > E_i + TD_{ij} \\ (1 - \frac{D_{ij}}{M})p & otherwise \end{cases} \quad (1)$$

D_{ij} を M で割った割ることによって時間の長さを正規化し，1 から引くことで短さを 0 以上 1 以下で表現している．また，$S_i + TD_{ij}$ が E_j より大きいときタスク T_j は必ず終了要求期間の終了時刻 E_j を超えてしまうので，この順に実行するのは好ましくないため，結合度は 0 とする．S_j が $E_i + TD_{ij}$ よりも大きい場合は，AGV はタスク i を終えた後必ず $\{S_j - (E_i + TD_{ij})\}$ 以上の時間待ってからタスク j を開始するため，D_{ij} に $\{S_j - (E_i + TD_{ij})\}$ を足して計算する.

選択されたタスクは (1) の式を使って前後のタスクとの結合度を計算し，結合度を確率として一緒に入れ替えるかを決定する．一緒に交換する場合は，同じく一緒に入れ替えるタスクの前後の

タスクとの結合度を計算し，一緒に入れ替えるかの決定を再帰的に行う．

最後にタスクの入れ替えを行う．交換を行った例を Fig. 6 に示す．Fig. 6 では選択されたタスクの前後のタスクとの結合度を計算してタスク 9 と 6，タスク 5 と 2 を一緒に交換するタスクとしている．

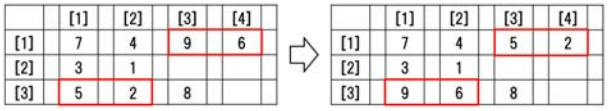

Fig.6 交換の例

4.4 スケジュールへの変換

配列をスケジュールに変換することを考える．先頭のタスクから順に開始時刻と終了時刻を計算し後尾まで順に割り当てていくことを各 AGV で行い，2 次元配列からスケジュールを得る．T_i が AGVa の p 番目に割り当てられたとし，T_j が $p+1$ 番目に割り当てられたとき，T_i の開始時刻を ST_i，T_i の終了時刻を ET_i とすると T_j の開始時刻 ST_j は (2) の式で与える．

$$ST_j = \begin{cases} 0 & if\ p = 1 \\ S_j - TD_{ij} & if\ S_j > ET_i + TD_{ij} \\ ET_i & otherwise \end{cases} \tag{2}$$

また，T_j の終了時刻 ET_j は (3) の式で与える．

$$ET_j = ST_j + TD_{ij} \tag{3}$$

各タスクの開始時刻は前のタスクの終了時刻と等しく，終了時刻は開始時刻に実行時間を足した時間とする．また，終了要求期間の制約を考慮するため，前のタスクの終了時刻と T_j の実行時間 TD_{ij} の和が終了要求期間の開始時刻を過ぎていない場合は，終了要求期間の開始時刻に配送が終わるようにタスクの開始時刻を割り当てる．

4.5 解の評価と選択

解の評価値の計算方法と選択方法について述べる．解の評価値は搬出するグループごとに終了要求期間からの遅れ時間を計算し，その和を評価値とする．グループは全ての物資が揃わなければ出発できないため，グループごとの終了要求期間からの遅れ時間はグループの中で終了時間の一番遅いタスクの遅れ時間よって与える．グループ s によって与えられる終了要求期間の終了時刻を TE_s，グループ s が搬出する物資を配送するタスクの番号の集合を X_s，グループ s が搬出する物資を配送するタスクの番号を $x_s(x_s \in X_s)$ とすると，グループ s の終了要求期間の終了時刻からの遅れ時間 DT_s は (4) の式で与える．

$$DT_s = \begin{cases} Max(ET_{x_s}) - TE_s & if\ Max(ET_{x_s}) - TE_s \geq 0 \\ 0 & otherwise \end{cases} \tag{4}$$

また，終了要求期間からの遅れ時間が 0 となった解は，より早く搬送が終わるスケジュールを求めたいため，評価値として終了要求期間の終了時刻とグループの全て物資の配送が終わった時刻との差の合計を評価値とする．グループの数を N とすると解の評価値 E は (5) の式で与えられる

$$E = \begin{cases} \sum_{k=1}^{N}(TE_k - Max(ET_{x_k})) & if\ \sum_{k=1}^{N} DT_k = 0 \\ \sum_{k=1}^{N} DT_k & otherwise \end{cases} \tag{5}$$

選択ではこれらの式で与えられた評価値に従い，前の世代で残された解と入れ替えによって生まれた解の集合から，評価値の小さい順に選択して次の解の集合とする．

5 検証

5.1 検証条件

Fig. 1 に示す環境で，40 個のタスクを各種法に与え性能を比較する．AGV の数は 3 台とする．各種法 1000 回の探索を行い，それぞれ 50 回試行する．遅れのないスケジュールを探索する早さとどれだけ早く配送を終えられる解を得られるかが重要であるため，探索実行中の評価値の遷移，実行可能解：評価値 0 以下の解を得た時の探索回数，探索終了時の評価値を比較する．ここで評価値が 0 以下である解は終了要求期間から遅れのない配送スケジュールを表している．

5.2 提案する解の構造の効果検証

まず提案する解の構造の効果を検証するために，Fig. 2 の解の構造での探索と本研究で提案する解の構造での探索を比較する．解の構造の違いによる影響を検証するため，$p = 0$ とし結合度は考慮しないようにする．比較する手法を表 1 に示す．Method1 は Fig. 2 の解の構造を用いた遺伝的アルゴリズムである．タスクの実行順を表した列の交叉には順序交叉，突然変異には転座突然変異を用い，AGV の割当を表した列には通常の交叉の 2 点交叉，突然変異には通常の突然変異を用いた．交叉率は 0.6，突然変異率は 0.05 とした．Method2 は結合度を常に 0 として，結合度を考慮せず選択したタスクのみを入れ替えるようにした提案手法のアルゴリズムである．どちらも初期集合はランダムに作成し，Method2 は一つの解から入れ替えによって生成する解は 2 個とする．Method1 と Method2 では新しい解を作成するときに解を生成する個数が異なるため，Method1 の初期集合の解の数は 300 個，Method2 の数は 100 とし選択する際の解の母体数が等しくなるようにする．

Table 1 解の構造の効果検証における比較手法

手法	解の構造	探索方法	結合度のパラメータ p
Method1	Fig. 2	遺伝的アルゴリズム	-
Method2	Fig. 3	入れ替え，評価，選択	0

タスク 40 個を与えた場合の探索実行中の評価値の遷移を Fig. 7，実行可能解を得た時の探索回数を Table. 2，探索終了時の評価値を Table. 3 に示す．Fig. 7 は探索回数毎の評価値の平均値をプロットしたものである．

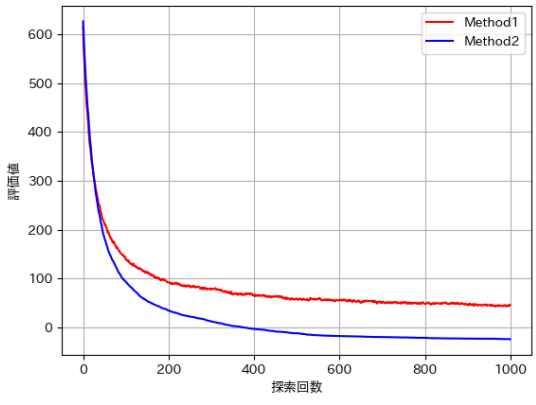

Fig.7 探索回数毎の評価値の平均の遷移

タスク 40 個を与えた結果では Table. 2，Table. 3 から，Method1 より Method2 のほうが実行可能解を少ない探索回数で求められ，低い評価値の解を発見できている．また，Method1 はタスク 40 個のときは 3 回しか実行可能解を探索できていないのに対し，Method2 は 47 回探索できている．これらのことから，本研究で扱う問題において提案する解の構造を用いた手法は従来手法よりも早く実

Table 2 実行可能解を得た時の探索回数

手法	平均値	標準偏差	最大値	最小値	探索成功回数
Method1	865.67	92.20	996	797	3
Method2	406.15	152.86	886	210	47

Table 3 探索終了時の評価値

手法	平均値	標準偏差	最大値	最小値
Method1	45.84	32.90	194	12
Method2	-26.72	12.00	8	-47

行可能解の探索でき，低い評価値の解を得られると言える．これは各タスクに期限が与えられ期限からの遅れを考える問題において，AGV ごとにタスクの実行順を考えながら探索することが重要であるからと考えられる．

5.3 結合度導入の効果検証

次に，タスクの終点と始点の接続性による影響を確認したいため，5.1 に示す条件で結合度を考慮した場合としない場合の結果を比較する．比較する手法を表 4 に示す．Method3 は $p = 1.0$ として結合度を考慮するようにした提案アルゴリズムである．Method2, Method3 の初期集合はランダムに作成し，初期集合の解の数は 100，一つの解から入れ替えによって生成解は 2 個とする．

Table 4 結合度導入の効果検証における比較手法

手法	解の構造	探索方法	結合度のパラメータ p
Method2	Fig. 3	入れ替え，評価，選択	0
Method3	Fig. 3	入れ替え，評価，選択	1.0

タスク 40 個を与えた場合の探索実行中の評価値の遷移を Fig. 8，実行可能解を得た時の探索回数を Table. 5，探索終了時の評価値を Table. 6 に示す．

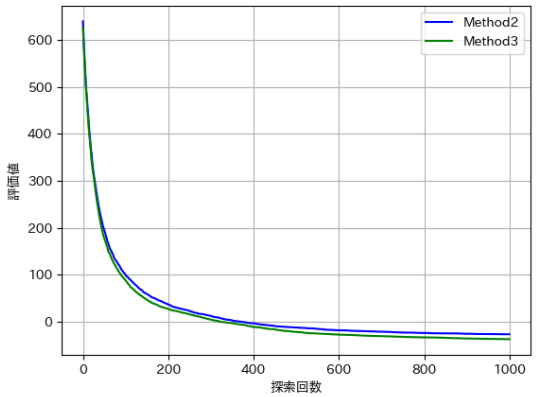

Fig.8 探索回数毎の評価値の平均の遷移

Method2 と Method3 では探索回数毎の評価値の平均の遷移，実行可能解を得た時の探索回数の平均，探索終了時の評価値の平均を比較すると大きな差は見られなかった．しかし，Table. 5 をみると，Method3 のほうが実行可能解の探索失敗回数が少なく，標準偏差も小さいことがわかる．このことから結合度を考慮した探索では結合度を考慮しない探索と比べ，無作為に行う解の初期集合の作成や入れ替えからでも，固定の探索回数での探索において安定して実行可能解を探索できると言える．

Table 5 実行可能解を得た時の探索回数

手法	平均値	標準偏差	最大値	最小値	探索失敗回数
Method2	406.15	152.84	886	210	3
Method3	367.48	78.69	530	242	0

Table 6 探索終了時の評価値

手法	平均値	標準偏差	最大値	最小値
Method2	-26.72	11.96	8	-47
Method3	-37.20	6.50	-22	-53

6 結言

素早く実行可能解を求め，配送がより早く終わるスケジュールを探索するために，タスクの AGV ごとの実行順を考慮可能な解の構造とタスクの終点と次に行うタスクの始点の接続性を考慮した入れ替えによる探索手法を提案し，数値シミュレーションにより検証を行った．検証では，先行研究の解を用いた手法より，提案する解の構造を用いた方が より少ない探索回数で実行可能解を得られ，より早く搬送ができる解を探索できることを確認した．また，接続性を考慮した探索の方が安定して遅れのないスケジュールを探索できることを確認した．今後はタスクの規模，AGV の台数，タスクの与えられ方など，条件を変えて提案手法の効果を検証していきたい．

参考文献

[1] Umar, U. A., Ariffin, M. K. A., Ismail, N., Tang, S. H. (2015). Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment. International Journal of Advanced Manufacturing Technology, 81(912), 21232141. https://doi.org/10.1007/s00170-015-7329-2

[2] Lyu, X., Song, Y., He, C., Lei, Q., Guo, W. (2019). Approach to Integrated Scheduling Problems Considering Optimal Number of Automated Guided Vehicles and Conflict-Free Routing in Flexible Manufacturing Systems. IEEE Access, 7, 7490974924. https://doi.org/10.1109/ACCESS.2019.2919109

[3] 平中雄一朗, 西竜志, 乾口雅弘. (2007). ラグランジュ緩和とカット生成による生産工程と複数台搬送車の同時スケジューリング問題に対する分解法. システム制御情報学会論文誌, 20(12), 465474. https://doi.org/10.5687/iscie.20.465

[4] 古川正志. ”数理計画問題—メタヒューリスティクス解法の基礎—.” 精密工学会誌 82.8 (2016): 735-739.

[5] 棟朝 雅晴 (2008). 遺伝的アルゴリズム その理論と先端的手法. 森北出版