自動車車体塗装の制約を考慮した複数ロボットアームの進化的経路設計

永井　裕也[1,a)]　梶浦　梨央[1,b)]　中村　博光[2,c)]　東園　雄太[2,d)]　小野　智司[1,e)]

概要：自動車車体の塗装を行う複数のロボットアームの経路の設計は，技術者が多くの時間をかけて行っているため，自動化や設計時間の短縮が求められている．車体の塗装問題は，塗装工程に特有の制約条件を考慮して行う必要があるため，従来の溶接作業などを対象とした経路設計手法を直接応用することが困難である．このため，本研究は塗装経路設計を配送計画問題の類似問題と捉え，塗装工程に特有の制約条件を追加した組み合わせ最適化問題として定式化を行い，進化計算を用いた解法を提案する．提案手法を用いてアーム先端の経路を設計する 2 次元の簡易的な問題において実験を行い，有効性を検証した．

Evolutionary Path Design for Car Painting Robot Arms Considering Constraints

Abstract: Multiple path planning of robot arms for car painting requires engineers to spend a lot of time, and there is a strong demand for automation and reducing the manual work. Because this problem includes constraints specific to the painting process, it is difficult to directly apply conventional planning methods designed for welding robot arms. This study proposes a multi robot arm path planning method for car painting. The proposed method formulates it as a combinatorial optimization problem including constraints specific to car body painting process. and solves it using evolutionary computation. The effectiveness of the proposed method was verified through experiments on a simple two-dimensional problem.

1. はじめに

自動車の生産工場において自動車車体の塗装を行う工程は，複数のロボットアームが 1 台の車体に同時に塗装の吹き付けを行う．この工程におけるアームの動作設計は，アーム間の衝突を回避する，ライン上を流れる車体が塗装可能範囲にあるうちに作業を終える，仕上がりの品質を考慮して塗装順を制限するなどの塗装工程に独特の制約条件を勘案して行う必要がある．このため，熟練の専門技術者がシミュレータを用いながら数週間から数ヶ月の時間をかけて設計を行っている．

[1] 鹿児島大学
Kagoshima University, Korimoto, Kagoshima, Kagoshima 8900065, Japan
[2] トヨタ車体研究所
TOYOTA AUTO BODY Research & Development, Kokubu, Uenodan, Kirishima, Kagoshima 8994461, Japan
[a)] k5629320@kadai.jp
[b)] k0301280@kadai.jp
[c)] h-naka@rad.toyota-body.co.jp
[d)] y.higashizono@rad.toyota-body.co.jp
[e)] ono@ibe.kagoshima-u.ac.jp

複数ロボットアームを用いた経路設計の研究として，物体の把持を対象とした経路設計 [1], [2]，自動車車体の溶接作業を対象とした経路設計 [3], [4] がある．これらの研究が対象とするタスクはアームがある位置で停止して行う作業が多い．本論文で対象とする塗装作業のようにアームが定速で移動しながら行う作業の経路設計を対象とした研究はこれまであまり行われておらず，近年になって行われ始めているが [5]，溶接工程におけるアームの経路設計と比較してまだ十分に研究が行われておらず，特に塗装品質を保つための制約を考慮した研究は，著者らが確認する限り行われていない．

非線形計画法による最適化において一般的に，制約条件を考慮する際は目的関数と制約条件より定義した新たな関数の極値を求めるラグランジュの未定乗数法を用いることが多い．これに対して進化計算は，制約違反の量に基づき目的関数を加算（減算）するペナルティ関数を追加する方法のほか，特定の問題を解くために特殊な表現方式を用いて制約を考慮する，修復処理を用いて実行不能解を実行可

能解へと変換する，制約条件と目的関数を個別に扱い解を生成する方法など，制約条件を様々な方法で扱うことができる [6].

本研究は，複数のロボットアームを用いた自動車車体の塗装経路設計を配送計画問題（Vehicle Routing Problem: VRP）の類似問題として捉え，組合せ最適化問題として定式化をして進化計算により解く手法を提案する．提案手法は，進化計算の利点を活かして，車体塗装問題に特有の条件を含む多様な制約条件をその特性に応じて最適化の各プロセスにおいて個別に扱う点に特徴がある．評価実験を行い，提案手法が多様な制約を考慮して 4 基のロボットアームの経路を設計することができ，専門技術者が設計した経路と類似する経路を設計することを確認した．

2. 関連研究

2.1 溶接作業を対象とした研究

溶接作業を対象とした複数ロボットアームを用いた経路設計に関する研究として Spensieri らの研究 [3] と Touzani らの研究 [7] が挙げられる．

Spensieri らは，経路設計の問題を 2 つの巡回セールスマン問題（Traveling Salesman Problem）の類似問題として分割し，繰り返し解くことでサイクルタイムが短い経路を設計する手法を提案した．この手法は，はじめに，ロボットアーム間の衝突を無視して各アームの経路とスケジュールを決定する min-max Multiple Generalized Traveling Salesman Problem（MGTSP）と捉えてその解を生成する．続いて MGTSP の解をもとに，ロボットアーム間の衝突を避けるため，アームが同期して動作する条件下で各アームのタスク順序とスケジュールを決定する Generalized Traveling Salesman Problem（GTSP）として，再度解を生成する．上記の部分問題は，要件を満たした経路が設計されるまで繰り返し解かれ，サイクルタイムが短いロボットアーム間の衝突を回避した経路を設計する．車体の溶接を行う 4 基のアームを含む環境で実験を行い，アーム間の衝突がない経路設計に成功した．

Touzani らは，複数ロボットのタスク順序割当問題に対して高品質な解を生成する反復的手法を提案した．この手法は問題を 4 段階に分けることで複雑性を解消した．1 段階目は各アームの可動範囲等の設定を考慮してタスク（訪れる点）の順序を決定する．2 段階目は 1 段階目で決定したタスクの順序をもとにアームの各点の関節角度などの構成を決定する．3 段階目は移動時間を最小化するために動作計画問題を解く．4 段階目はアーム間の衝突を避けるために，アームの待ち時間が最小となる同期信号を求める最適化問題を解く．同期信号を用いてアームを制御することでアーム間の衝突を回避する．4 段階目で生成された解とコストを記録し，再度解を生成する．この処理は停止基準に達するまで反復され，最終的に最もサイクルタイムが短い解が出力される．2 基のアームと 3 基のアームで実験を行い，衝突を回避したアームの軌道生成に成功した．しかし，これらは溶接作業を対象とした研究であるため，塗装作業と比較し，ある点で停止して作業を行う，作業順序に制限がないなど単純な条件となっている．

2.2 塗装作業を対象とした研究

塗装作業を対象とした複数ロボットアームの経路設計に関する研究 Zbiss らの研究 [5] が挙げられる．Zbiss らは，車体の CAD モデル，アームの設置位置，可動範囲等の設定を入力として経路を設計する手法を提案した．

この手法は塗装経路設計の問題を 3 段階に分けることで複雑性を解消した．1 段階目はアーム間の衝突が起こる可能性がある領域を算出し，衝突が起きない領域の 2 次元経路を決定する．2 段階目は衝突が起こり得る領域の 2 次元経路を決定する．1，2 段階で決定した経路において塗装漏れが確認された場合は，再度経路の決定を行う．3 段階目は 1，2 段階目の経路をもとに逆運動学ソルバを用いて 3 次元の経路を設計する．

Zbiss らは 6 基のアームで実験を行い，アーム間の衝突を回避した 3 次元経路設計に成功した．しかし，車体が一定速度で動く条件や塗装の品質を高めるための塗装順序の条件など，塗装問題特有の条件を考慮していない．

3. 提案手法

3.1 基本アイデア

本論文は，複数のロボットアームを用いた自動車車体の塗装経路設計を VRP の類似問題として定式化を行い，自動車生産現場で考慮されている一部の条件を制約として追加した組合せ最適化問題を進化計算を用いて解く手法を提案する．これにより，実用的な時間内に準最適解を発見できるメタヒューリスティクスを用いることが可能となる．提案手法は進化計算の利点を活かして最適化の各プロセスにおいて制約条件を個別に対処することにより，表 1 に示すように，従来手法よりも多くの制約条件を考慮することが可能となる．

本手法は組合せ最適化問題を解くことにより各アームが訪問する塗装箇所の割当てと順序を決定するソルバと，塗装箇所の割当てなどから各アームの詳細な経路を決定するするシミュレータから構成される．ソルバでは進化計算の代表的なアルゴリズムである遺伝的アルゴリズムを利用することで他の進化計算手法に比べ，問題の特性に依らず，大域的最適解に近い解を求めることが容易である．シミュレータでは，貪欲法に基づいて各アームの詳細な経路を決定し，アームの稼働時間を求める．すなわち，各アームは次に塗装を行う箇所に向かって直線状に移動することを繰り返すことで詳細な経路を生成する．また，ソルバとシミュレータで処理を分けることにより，制約条件を追加す

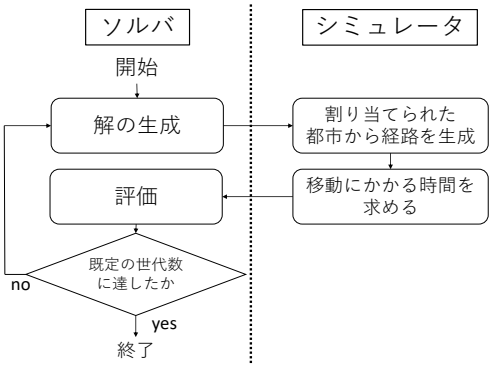

図 1 提案手法の処理手順

る際やシミュレータを変更する際に，最適化手法の変更が不要となる．

3.2 構成と処理手順

提案手法の構成と処理手順を図 1 に示す．本手法は関連研究 [5] と同様に，塗装経路設計の問題を部分問題に分割する．すなわち，塗装箇所（都市に相当）の訪問順ならびにアームへの割当てを決定する VRP に類似する組合せ最適化問題（上位問題）と，決定された塗装箇所の訪問順，アームへの割当てから詳細な経路を導出する問題（下位問題）を組合せることで塗装経路設計問題を表現する．本手法は都市の座標を入力として前者を解くソルバと，ソルバで生成された解候補を入力として後者を解くシミュレータから構成される．ソルバは遺伝的アルゴリズム（Genetic Algorithm: GA）による最適化を行い，シミュレータは貪欲法を利用する．上位問題における解候補の評価を行う際に，下位問題を解くことで詳細な経路を設計し，設計された詳細な経路をもとに，上位問題における解候補の評価値（適応度）を計算する．

加えて，本問題は塗装作業に特有の制約を含めて多様な制約条件を含むが，本方式は制約の特性を考慮し，ソルバやシミュレータ内の各プロセスにおいて個別に対処する点に特徴がある．例えば，塗装はアームが直線状に移動しながら塗料の吹き付けを行う必要がある．関連研究で対象としている溶接を行うアームの経路設計は溶接を行う箇所を都市とすることで VRP として直接的に表現することが可能であるが，本問題は，同様に考えることが難しい．このため，本研究は都市を線分として表現する．これにより，ムラになることを避けて直線状に移動しながら均一に塗装を行うことが可能となる．上位問題は都市の訪問順のみを決定し，各都市のどちらの端点から塗装を開始するかについては下位問題において決定する．

3.3 組合せ最適化としての定式化

3.3.1 設計変数

提案手法のソルバは，既存の進化計算アルゴリズムを応用できるように，塗装問題を VRP の類似問題として定式化し，VRP と同様の解表現を採用する [8], [9], [10]．すなわち，解候補に相当する個体 $\boldsymbol{x}$ は，訪問する順で並べられた都市番号の順列である．設計変数の総数は都市の総数 N と同じであり，変数の各要素 $\boldsymbol{x}_i$ は都市番号を割当てるため $1 \sim N$ の整数を用いる ($\boldsymbol{x} \in \{1, 2, \cdots, N\}$)．なお，VRP および本問題は，複数のトラックやアームの経路を決定する問題である．ここでは，説明を単純化するために，各アームが同数の都市を訪問することとし，下記により塗装位置（都市）の割当と訪問順の両方を表現することとする．

$$\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\},\ (x_i \neq x_j,\ \forall i, j \in \{1, \ldots, N\}) \quad (1)$$

すなわち，x_1 から $x_{N/n_{arms}}$ は第 1 アームが訪問する都市とその順序を表わし，$x_{N/n_{arms}+1}$ から $x_{2N/n_{arms}}$ は第 2 アームが訪問する都市とその順序を表わすといった具合に，$\boldsymbol{x}$ の要素の位置により各アームへの都市の割当を表現する．

3.3.2 目的関数

解候補の評価値（適応度）は，シミュレータを呼び出して詳細な経路の設計を行い，その結果をもとに算出する．シミュレータは塗装経路の概略を表わす $\boldsymbol{x}$ をもとに，アーム $a (\in \{1, \ldots, n_{arms}\})$ の詳細な経路 $\boldsymbol{r}_a$ と作業時間 $t_a(\boldsymbol{x})$ を求める．$\boldsymbol{r}_a = \{\boldsymbol{r}_{a,1}, \boldsymbol{r}_{a,2}, \ldots\}$ はアーム a の u 秒ごとの座標である．上位問題における目的関数は t_a の最大値とし，これを最小化する．

$$\begin{aligned}
\text{minimize} \quad & f(\boldsymbol{x}) = \max_a t_a(\boldsymbol{x}) + p_a(\boldsymbol{x}) + p_c(\boldsymbol{x}) && (2)\\
\text{subject to} \quad & d(\boldsymbol{r}_{a,k}, \boldsymbol{o}_a) \leq R_a && (3)\\
& d(\boldsymbol{r}_{a,k}\ , \boldsymbol{r}_{b,k}) \geq R_{collision} \\
& \qquad (a, b \in \{1, \ldots, n_{arms}\},\ a \neq b) && (4)\\
& c(e(p_{c,l-1}) \geq e(p_{c,l})) \leq \tau && (5)
\end{aligned}$$

式 (3) から式 (5) は制約条件を表わしており，$p_a(\boldsymbol{x})$, $p_c(\boldsymbol{x})$ は制約違反に基づくペナルティ関数である．

3.3.3 制約条件

本手法は，アームの可動範囲に関する制約（式 (3)），アーム同士の衝突に関する制約（式 (4)），塗装順序に関する制約（式 (5)）も考慮して塗装経路の設計を行う（表 2）．アームの可動範囲に関する制約は，アーム a の時刻 k の座標が $\boldsymbol{r}_{a,k}$ $(k \in \{1, \ldots, t_a\})$, 可動中心 $\boldsymbol{o}_a$ から可動域 R_a 内の距離にあることとなる．式 (3) における d は距離を表わす．ロボットアーム同士の衝突に関する制約は，異なる 2 つのアーム間の距離が常に規定距離 $R_{collision}$ よりも大きくなることを要求する．

自動車車体の垂直面における塗装はアームを水平方向に往復させながら下から上に順次塗装を行うことが求められ，同一のパネルにおいて塗装順が逆転することは避けなければならない．これは，塗装の垂れを抑えるとともに，吹き付けを行った継ぎ目を目立たせないようにするために

表 1 各手法で考慮する条件

手法	考慮する制約条件				
	アームの可動範囲	アーム同士の衝突	塗装面積	作業対象が一定速度で移動	塗装順序
溶接作業関連研究 [3]	○	○			
塗装作業関連研究 [5]	○	○	○		
提案手法	○	○	○	○	○

必要となる．一方で，ロボットアームの設置条件より，車体下部については車体中部および上部よりも後に塗装を行う必要があり，上記の塗装順に関する制約をごく少ない回数のみ違反を認める必要がある．本手法は上記の塗装順序に関する制約を式 (5) によって表現する．車体を構成するパネル c における塗装箇所 $s_{c,1}, s_{c,2}, \ldots, s_{c,l}, \ldots$ は添字 l が大きいほど上部に位置するとし，塗装箇所 $s_{c,l}$ が塗装されたタイミングを $e(s_{c,l})$ とする．上下に隣接する塗装箇所 $s_{c,l-1}$，$s_{c,l}$ において，基本的に $e(s_{c,l-1}) < e(s_{c,l})$ でなければならないが，アームの設置状況に応じてパネル毎に τ 回までの違反を許容する．式（5）における $c(\mathcal{C})$ は条件 $\mathcal{C}$ が生じる回数を表す．

3.4 制約違反時の対処

提案手法は表 2 に示す様々な制約条件を，解候補の表現形式の工夫，制約違反に応じた修復処理の適用，およびペナルティを課す（表 3）ことで制約の解消を図る．制約 1 に対する対処方法を操作 1-1, 操作 1-2, 操作 1-3，制約 2 に対する対処方法を操作 2-1, 制約 3 に対する対処方法を操作 3-1 とする．

アームの可動範囲に関する制約（式 (3)）は，塗装対象車両が定速で移動していることを考え，3 通りの方法で対処する．まず，ソルバにおいて，全作業時間内にアーム a の可動範囲に入ることのない塗装箇所は，他のアームに割当てられた可動範囲に入る可能性がある塗装箇所と入替える修復処理によって，アーム a に割当てないこととする．加えて，シミュレータにおいて，アーム a がある都市を訪問すべき時間に塗装箇所がまだ対象範囲外である場合は，対象の塗装箇所が範囲内に入るまでアーム a を待機させる．また，本来その塗装箇所を訪れるべきタイミングで既に車体が流れていてアームの範囲外になってしまった場合は，シミュレータの仕様上待機状態のまま割当てられた塗装箇所をすべて訪れずに設計を終える．この際，割当てられたが訪れなかった塗装箇所の数と制約違反の時間に応じてペナルティ p_a を目的関数に加算することで違反の発生を抑制する．

ロボットアーム同士の衝突に関する制約（式（4））は，複数のアームが十分に間隔を空けて設置された環境を対象としており，本制約に対する違反が発生する頻度が比較的低いことから，本制約の違反が確認された場合は，違反が生じている時間に応じてペナルティ p_c を目的関数に加算することで違反の発生を抑制する．

塗装順序に関する制約（式（5））において 制約違反が確認された場合，各アームが連続する領域を塗装するように都市の割当と訪問順を同時に変更する修復処理を行う．

3.5 最適化における操作

本手法は GA の最適化オペレータとして，トーナメント戦略（Tournament Selection）[11], [12]，順序交叉（Order Crossover）[13], [14]，反転操作（Inversion Operation）[15] を用いる．トーナメント戦略は母集団からランダムに選ばれた複数の個体でトーナメントを行い，評価値（適応度）が高い個体を選択する．順序交叉は 2 つの親個体から 2 つの子個体を生成する．一方の親個体から一部の遺伝子を引き継ぎ，それ以外の部分は，他方の親個体から相対的な順序を保持して遺伝子を受け継ぐ．反転操作は一定確率で個体中のランダムな 2 つの要素を入替える．

3.6 シミュレータ

シミュレータは，ソルバで生成された個体をもとにアームの詳細な経路を作成し，作業時間を求めてソルバに返す．シミュレータの処理手順を図 2 に，経路を決定するルールを表 4 に示す．各アームの詳細な経路は割当てられた塗装箇所を順番に訪問することで決定するが，一般的な VRP とは異なり，塗装箇所は線状の形状を持ち，いずれの端点に到着するかを決定する必要がある．また，塗装対象となる車体が等速で移動する点も VRP とは異なる．訪問先の都市の両端がアームの可動範囲に入るまでアームはその場に待機し，都市の両端が可動範囲に入った時点で移動を開始する．各アームが塗装箇所に到着した場合は一方の端点から反対の端点へ移動する．端点へ移動を終えた段階で当該箇所の塗装は終えたものとして次の塗装箇所へ移動する． このとき，移動先の塗装箇所の両端のうち移動距離が短くなる端点へ移動する．

また，シミュレータにおいて詳細な経路を設計する際，可動範囲外に塗装箇所が存在するためにアームが待機する時間，割当てられたが訪問できなかった塗装箇所の数，2 つのアーム間の距離が規定値 $R_{collision}$ を下回った時間，各塗装箇所の塗装が開始された時間を記録する．塗装箇所の塗装された時間をもとに塗装順序に関する制約の違反を判定する．違反が確認された場合，アームの相対的な位置を固定する修復処理を施し制約違反を解消する．修復処理

表 2 考慮する制約条件

番号	条件	対応箇所
制約 1	アームが可動中心から可動域内の距離にある.	ペナルティ，修復処理
制約 2	2 つのアームの距離を規定距離以上に保つ	ペナルティ
制約 3	下から上へ順次塗装を行う．ただし，一定回数の違反は許容する.	ペナルティ，修復処理
制約 4	車体のすべての塗装箇所を塗装する.	解表現
制約 5	車体が定速で直線状を移動する.	シミュレータ
制約 6	各塗装箇所は，あらかじめ決められた直線または曲線に沿って，あらかじめ決められた速度 v_{sp} で一度に塗装する.	シミュレータ
制約 7	アームが車体と衝突してはならない.	シミュレータ
制約 8	塗装を行っていない際のアームの移動速度は v_{mv} 以下とする.	シミュレータ

表 3 ペナルティ関数および修復操作による制約違反への対処

操作	対処方法	適用位置	処理内容
操作 1-1	ペナルティ	評価値算出時	可動範囲外にいる時間 ×500 を評価値に加算する.
操作 1-2	ペナルティ	評価値算出時	割当てられたが訪問しなかった塗装箇所の数 ×10000 を評価値に加算する.
操作 1-3	修復処理	個体生成時	可動範囲に入らない都市を可動範囲に入る都市と入替る.
操作 2-1	ペナルティ	評価値算出時	2 つのアームの距離が規定距離よりも短い時間 ×1000 を評価値に加算する.
操作 3-1	修復処理	評価値算出後	アームの相対的な位置を固定する.

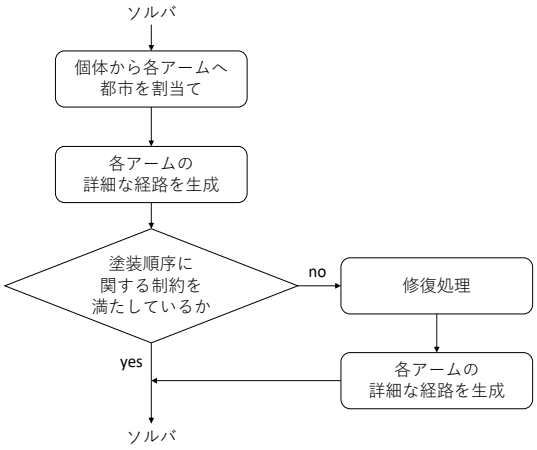

図 2 シミュレータの処理手順

表 4 シミュレータにおいて経路を決定するルール

番号	ルール
ルール 1	塗装箇所の両端が可動範囲に入るまでは待機する.
ルール 2	塗装箇所の両端が可動範囲に入った時点で移動を開始する.
ルール 3	塗装箇所に到達後は一方の端点から反対の端点へ移動する.
ルール 4	塗装箇所の移動を行う際は移動先の都市の両端のうち移動距離が短くなる端点へ移動する.
ルール 5	割当てられたすべての塗装箇所を訪問したら待機する.

は個体を書き換えるため，再度，各アームの詳細な経路の設計を行う．最終的なアームの作業時間をソルバへ渡しシミュレータの処理が終了する.

4. 評価実験

提案手法の有効性を検証するために，簡易的な 2D シミュレータを用いて経路の設計を試みた.

4.1 実験設定

本実験は，車体側面の塗装を 4 基のアームで行う経路設計を試みた．経路は 2 次元座標の順列で構成されるものとし，本シミュレータは 1/100 秒毎に座標を算出し（$u = 1/100$），アーム先端の姿勢は塗装面に対して常に垂直に正対するものとした．4 基のアームは実際の生産現場と同様の間隔に設置し，可動範囲ならびに移動速度 v_{sp}，v_{mv} を実際に使用されているアームの仕様にもとづいて設定した．各アームに割当てる塗装箇所（都市）数は，偏りが生じないように固定した．3 車種 M_1, M_2, M_3 について本手法を適用し，得られた経路を，専門技術者が設計した経路と比較する形で評価した.

また，本手法における GA の設定は，母集団を 400 個体，突然変異確率は 2%とした．150 世代を上限とし，車種毎に 5 試行ずつ経路の設計を試みた．塗装順序に関する制約（式（5））における違反を認める回数は，人手で設計した経路をもとに $\tau = 1$ とした．また，各制約違反時のペナルティとして可動範囲に関する制約では，

$$p_a = t_{vio} \times 500 + n_{unvisits} \times 10^4 \quad (6)$$

$$p_c = t_{cor} \times 10^3 \quad (7)$$

として目的関数に加算した．ここで，t_{vio} は違反が生じている時間であり，$n_{unvisits}$ は，割当てられたが訪れなかった塗装箇所数，t_{cor} はロボットアーム同士の衝突に関する制約の違反が生じている時間を表わす.

4.2 実験結果

提案手法を適用した際の，初期集団における解の例と最適化終了時の最良解の経路を図 3 に示す．初期集団における解はランダムに塗装箇所を移動し，車体の一部のみ塗装を行う経路が設計されているが，最適化終了時の解では，

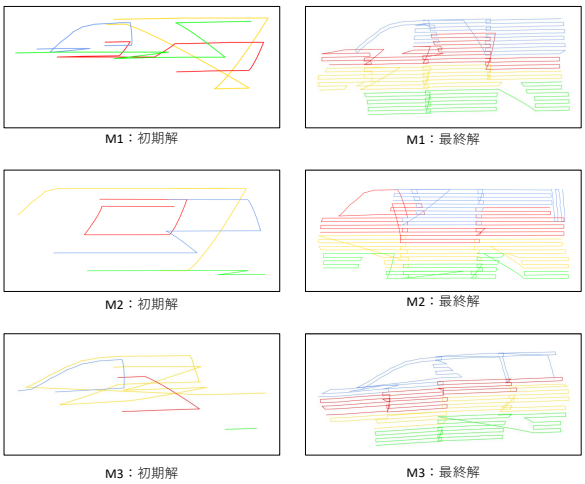

図 3　提案手法によって設計された経路

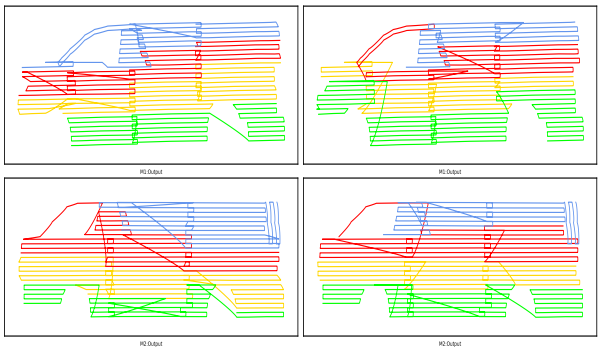

図 4　提案手法で設計した経路と技術者が設計した経路比較

アームを水平方向に往復させながら下から上に順次塗装を行う経路が設計されていることがわかる．提案手法によって，M_1，M_2，および M_3 におけるすべての試行において，すべての制約を満たした経路を設計することに成功した．

また，M_1，M_2 において，提案手法で設計した経路と実際に技術者が設計した経路の比較を行った（図 4）．提案手法により設計された経路を専門技術者に提示したところ，「総じて，人手で設計したアーム割当て，経路に類似した解」であるとのコメントを得た．

4.3　考察

本手法が類似した経路を設計した大きな要因として塗装順序に関する制約違反に対する修復処理が挙げられる．本修復処理は，技術者が設計した経路を参考に塗装箇所の割当てと訪問順を変更するため，人が設計する経路に類似した経路を生成することができる．

一方で本実験では，各アームに割当てる塗装箇所数を固定しており，これにより解空間の全てを探索していないこととなる．アームに割当てる塗装箇所の数を可変にし，4 基のアームの作業時間の平均を最小化する第二目的関数を追加し多目的最適化へと変更するなどの更なる工夫により，より良好な解を発見できると考える．

表 5　M_1：各アームの稼働時間比較

経路	アーム 1[s]	アーム 2[s]	アーム 3[s]	アーム 4[s]
提案手法				
1 回目	32.62	31.21	32.31	24.61
2 回目	31.04	31.53	29.61	36.74
3 回目	30.57	29.38	31.61	24.61
4 回目	30.30	29.26	30.13	24.61
5 回目	30.89	32.51	30.74	36.74
技術者	31.67	30.77	21.28	42.63

表 6　M_2：各アームの稼働時間比較

経路	アーム 1[s]	アーム 2[s]	アーム 3[s]	アーム 4[s]
提案手法				
1 回目	31.81	31.86	31.55	30.82
2 回目	30.40	29.85	28.35	30.18
3 回目	30.19	31.86	31.55	30.90
4 回目	30.40	29.85	28.35	30.18
5 回目	30.19	31.86	31.55	30.90
技術者	27.03	37.70	23.41	34.68

また，最適化を行った際の最良解の評価値の変化を図 5 に示す．最適化の中盤から終盤において，評価値は変化していないが生成された経路が変化することがあることを確認した．これは，適応度としてアームの稼働時間の最悪値を設定したためである．本問題は，ライン上を流れる車体が塗装可能範囲にあるうちに作業を終える必要があるため，アームの稼働時間の最悪値を評価値としたが，最悪値以外のアームの経路に変化がある場合，その変化を考慮することができない．各アームの微小な経路の変化を考慮できる評価値に変更することで，より各アームの稼働時間が短い経路を設計できると考える．

表 5 および 6 に，M_1，M_2 において本手法で設計した経路と技術者が設計した経路における各アームの稼働時間を示す．アームの稼働時間の最悪値を最小化する経路設計の観点では，本手法は人手で設計された経路よりも良好な目的関数値の経路を設計した．ただし，これは技術者はアームの稼働時間の最悪値以外にも様々な指標に着目して設計を行っているためであり，本手法においても今後はより詳細な評価項目を導入する必要があると考える．

5.　結論

本論文は，複数のロボットアームを用いた自動車車体の塗装経路設計を自動で行う手法を提案した．2D シミュレータを用いた評価実験を行い，本手法が多様な制約条件を満たし，専門の技術者により設計された経路に類似した経路を設計することを確認した．

今後，各アームの微小な変化を考慮した適応度の設定や，車体の側面以外も含めた経路設計について検討する．また，実際の塗装経路を設計する際に利用される 3D シミュレータと組合せ，技術者の労力軽減を実現する．

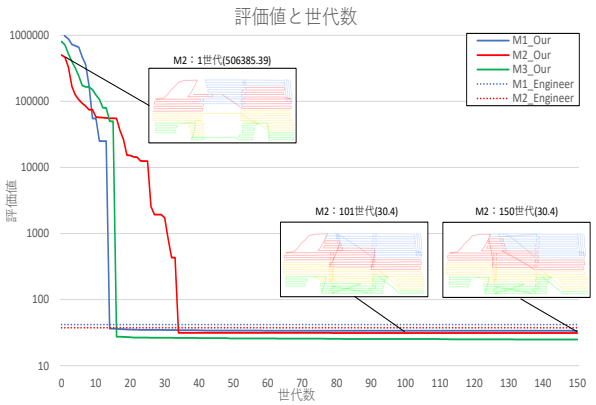

図 5 世代数と評価値

参考文献

[1] A Tika, N Gafur, V Yfantis, and N Bajcinca. Optimal scheduling and model predictive control for trajectory planning of cooperative robot manipulators. *IFAC-PapersOnLine*, Vol. 53, No. 2, pp. 9080–9086, 2020.

[2] Rahul Shome. Roadmaps for robot motion planning with groups of robots. *Current Robotics Reports*, Vol. 2, No. 1, pp. 85–94, 2021.

[3] Domenico Spensieri, Johan S Carlson, Fredrik Ekstedt, and Robert Bohlin. An iterative approach for collision free routing and scheduling in multirobot stations. *IEEE Transactions on Automation science and Engineering*, Vol. 13, No. 2, pp. 950–962, 2015.

[4] Hicham Touzani, Hicham Hadj-Abdelkader, Nicolas Séguy, and Samia Bouchafa. Multi-robot task sequencing & automatic path planning for cycle time optimization: Application for car production line. *IEEE Robotics and Automation Letters*, Vol. 6, No. 2, pp. 1335–1342, 2021.

[5] K Zbiss, Amal Kacem, Mario Santillo, and Alireza Mohammadi. Automatic collision-free trajectory generation for collaborative robotic car-painting. *IEEE Access*, Vol. 10, pp. 9950–9959, 2022.

[6] Carlos A Coello Coello. Constraint-handling techniques used with evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1310–1333, 2022.

[7] Hicham Touzani, NicolasS 辿 guy, Hicham Hadj-Abdelkader, Ra 炭 lSu 叩 rez, Jan Rosell, Leopold Palomo-Avellaneda, Samia Bouchafa. Efficient industrial solution for robotic task sequencing problem with mutual collision avoidance & cycle time optimization. *IEEE Robotics and Automation Letters*, Vol. 7, No. 2, pp. 2597–2604, 2022.

[8] Christian Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & operations research*, Vol. 31, No. 12, pp. 1985–2002, 2004.

[9] Barrie M Baker and MA1951066 Ayechew. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, Vol. 30, No. 5, pp. 787–800, 2003.

[10] Habibeh Nazif and Lai Soon Lee. Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling*, Vol. 36, No. 5, pp. 2110–2117, 2012.

[11] Melanie Mitchell. *An introduction to genetic algorithms*. MIT press, 1998.

[12] Vladimir Filipović. Fine-grained tournament selection operator in genetic algorithms. *Computing and Informatics*, Vol. 22, No. 2, pp. 143–161, 2003.

[13] Lawrence Davis, et al. Applying adaptive algorithms to epistatic domains. In *IJCAI*, Vol. 85, pp. 162–164, 1985.

[14] Kusum Deep and Hadush Mebrahtu. New variations of order crossover for travelling salesman problem. *International Journal of Combinatorial Optimization Problems and Informatics*, Vol. 2, No. 1, pp. 2–13, 2011.

[15] Kyu-Yeul Lee, Seong-Nam Han, and Myung-Il Roh. An improved genetic algorithm for facility layout problems having inner structure walls and passages. *Computers & Operations Research*, Vol. 30, No. 1, pp. 117–138, 2003.