

自動車車体塗装における様々な制約を考慮した複数ロボットアームの経路設計 Path Planning of Multiple Robot Arms Considering Various Constraints of Car Painting

永井 裕也¹⁾ 中村 博光²⁾ 新町 成人²⁾ 東園 雄太²⁾ 小野 智司¹⁾
Yuya Nagai Hiromitsu Nakamura Narito Shinmachi Yuta Higashizono Satoshi Ono

1 はじめに

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ライン上を流れる車体に対して、予め決められた経路を辿り塗料の吹付を行う。この工程におけるロボットアームの経路設計は、アーム間の衝突を回避するなどの一般的な制約条件に加えて、仕上がりの品質を考慮して塗装順を制限するなど、塗装工程に独特の制約条件も勘案して行う必要がある。現在、塗装経路設計は熟練の専門技術者が数週間から数ヶ月の時間をかけて行っているため、設計時間の短縮や自動化が求められている。

複数のロボットアームの経路を設計する研究は多く行われている [1, 2]。しかし、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動作を行う作業が多い。近年、塗装作業のようにアームが移動しながら行う作業を対象とした研究も行われ始めているが [3]、溶接作業におけるロボット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いない。特に塗装の品質を保つための制約条件は実用化にあたり重要な制約条件であるが、この制約条件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。

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配送計画問題 (Vehicle Routing Problem: VRP) の類似問題と捉え、組合せ最適化問題として定式化を行い、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複数基のロボットアームの 3 次元経路を設計することができることを確認した。

2 関連研究

溶接作業を対象とする複数の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 [1] と Touzani らの研究 [2] が挙げられる。これらの研究は、衝突を無視して経路設計を行った後、衝突が発生した場合は別途処理を行うことで稼働時間を減らしつつ衝突も回避した。しかし、自動車車体の溶接工程は、アームがある地点で停止して作業を行うため、これを塗装作業に応用することは難しい。

塗装作業を対象とする複数のロボットアームを用いた経路設計に関する研究も行われている [3, 4]。Zbiss らは、ロボットアームの可動範囲から予め衝突が起こり得る領域を取得し、衝突が起こり得る領域と起きない領域それぞれで経路を設計することで、衝突を回避した塗装経路を設計した [3]。しかし、実用化において重要となる塗装の品質は考慮していない。永井らは自動車の生産現場で考慮されている一部の条件も制約条件として考慮

1) 鹿児島大学 Kagoshima University

2) トヨタ車体研究所

TOYOTA AUTO BODY Research & Development

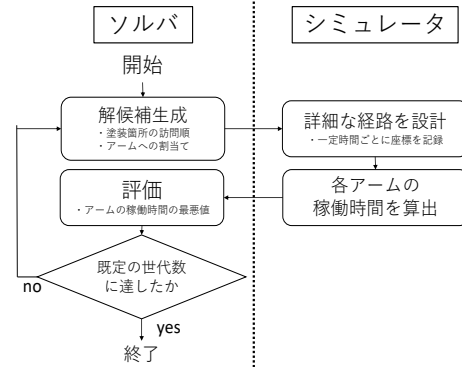


図 1 提案手法の処理手順

した経路設計手法を提案し、車体のフード、側面、バックドアの片側を塗装する 3 次元経路の設計が可能であることを示した [4]。

3 提案手法

本研究では、先行研究 [4] で提案された手法をもとに、車体全体を塗装する経路を設計する手法を提案する。本手法は、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し、自動車の生産現場で考慮されている一部の条件を制約条件として追加した組合せ最適化問題として扱う。また、この組合せ最適化問題を階層的な問題として定式化し、上位の問題を進化計算により、下位の問題を貪欲法によって解く。提案手法は、階層型最適化および進化計算の利点 [5] 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、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。また、上位問題およびその解法を変更することなく、下位問題側のみを変更することで、従来手法 [4] を、車両全体を塗装する手法に拡張することが可能である。

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 [3] 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 (顧客に相当) の訪問順ならびにロボット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、ロボット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。上位問題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 (Genetic Algorithm: GA) をソルバとして利用する。これにより、他の進化計算手法に比べて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となる。下位問題は、貪欲法に基づいて各ロボットアームの詳細な経路を決定するシミュレータを作成することで解き、ロボットアームの稼働時間を求める。すなわち、各ロボットアームは次に塗装を行う箇所に向かって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。

提案手法のソルバは、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 [6, 7]. すなわち、解候補に相当する個体 \mathbf{x} は、塗装（訪問）する順で並べられた塗装箇所（顧客）番号の順列である. 設計変数の総数 N は塗装箇所の総数 N_p 以上とし、変数の各要素 x_i は塗装箇所番号 $\{1, \dots, N_p\}$ または塗装を行わないことを示すダミー値 $\{N_p + 1, \dots, N\}$ とする ($\mathbf{x} \in \{1, \dots, N\}$). これにより、解候補 \mathbf{x} は以下のような形で塗装箇所の割当てと塗装順の両方を表現できる.

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

変数の一部にダミー値を割り当てることで、アームに割り当てる塗装の数を可変とすることができる.

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 \mathbf{x} をもとに、ロボットアーム $a (a \in \{1, \dots, n_{arms}\})$ の詳細な経路 \mathbf{r}_a と作業時間 $t_a(\mathbf{x})$ を求める. $\mathbf{r}_a = \{r_{a,1}, r_{a,2}, \dots\}$ はロボットアーム 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.

$$\text{minimize} \quad f(\mathbf{x}) = \max_a t_a(\mathbf{x}) + p_a(\mathbf{x}) + p_c(\mathbf{x}) \quad (2)$$

$$\text{subject to} \quad d(\mathbf{r}_{a,k}, \boldsymbol{\theta}_a) \leq R_a \quad (3)$$

$$d(\mathbf{r}_{a,k}, \mathbf{r}_{b,k}) \geq R_{collision}$$

$$(a, b \in \{1, \dots, n_{arms}\}, a \neq b) \quad (4)$$

$$c(e(s_{c_g, t-1}) \geq e(s_{c_g, t})) \leq \tau \quad (5)$$

式 (3) から式 (5) は制約条件を表わしており、 $p_a(\mathbf{x})$, $p_c(\mathbf{x})$ は制約違反に基づくペナルティ関数である [4].

本研究では、車体の両側面を含む全体を塗装する 3 次元の経路を導出するよう従来手法 [4] を拡張する. 従来手法は車体の片側のフード、側面、バックドアを塗装する経路を導出する手法であったため、提案手法は、ソルバにおいては従来手法と同様に車体片側の塗装範囲割り当てと塗装順決定を行うこととし、ソルバが決定する車体片側の塗装順をシミュレータにより車体両側の塗装経路に展開することで車体全体を塗装する 3 次元の経路を導出する.

このとき、基本的には左右対象となるように経路を決定するが、車体のフードやバックドア部分等は左右対称な経路にしまうとアーム同士の衝突が発生するため、パネル単位に以下の 3 種類の両側展開ルールをあらかじめ与えておくこととする.

- (1) 左右対象
- (2) 並行移動
- (3) 並行移動 + 片側遅延

(2) はバックドアなどにおいて、ロボットアーム同士が一定距離を保ちながら平行に塗装を行う. (3) はフードに適用されるルールであり、2 基のロボットアームが長時間平行に塗装を行うことで互いの機体に塗料がかかることを防ぐために、片方のアームの動き出しを遅らせることでアーム間の距離が一定にならないように塗装を行う. 上記のルールにより他の経路に大きな影響を与えることなく衝突を回避した両側の経路を設計することが可能になる.

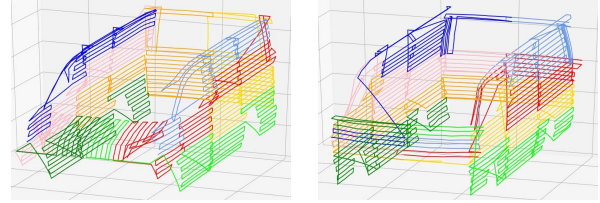


図 2 提案手法により設計された塗装経路

4 評価実験

提案手法の有効性を検証するために、全 2 車種 M_1, M_2 について車体のフード、側面、バックドアの塗装を左右 4 基、計 8 基のロボットアームで行う経路の設計を試みた. 本実験における経路は 3 次元座標の順列で構成されるものとし、アームは生産現場に基づいて設置した. また、各アームに割り当てる塗装箇所数を可変にするため、 $N = 1.25N_p$ とした.

考慮すべき全ての制約条件を満たした経路を設計可能であることを確認するため、各車種 4 試行ずつ経路の設計を行った. 提案手法によって得られた経路を図 2 に示す. 2 車種ともに全ての制約条件を満たした経路が設計可能であることを確認した.

5 結論

本研究は、複数のロボットアームを用いた自動車車体の 3 次元塗装経路設計を自動で行う手法を提案した. 2 車種に対して本手法を適用し、全ての制約条件を満たした経路を設計可能であることを確認した. 今後、設計された経路を生産現場で用いられているシミュレータに適用することで、提案手法の実用性を検証する.

参考文献

- [1] 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.
- [2] Hicham Touzani, Nicolas Seguy, Hicham Hadj-Abdelkader, Raul Suarez, Jan Rosell, Leopold Palomo-Avellaneda, and 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.
- [3] 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.
- [4] 永井裕也, 田島彩音, 中村博光, 東園雄太, 小野智司. 自動車車体塗装用複数ロボットアームの経路設計における複数の制約対処法の併用. 人工知能学会全国大会論文集 第 37 回 (2023), pp. 1B4GS201–1B4GS201. 一般社団法人人工知能学会, 2023.
- [5] Asuka Hisatomi, Hitomi Koba, Kazunori Mizuno, and Satoshi Ono. Elthon: An escher-like tile design method using hierarchical optimization. *Applied Soft Computing*, Vol. 112, p. 107771, 2021.
- [6] Christian Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & operations research*, Vol. 31, No. 12, pp. 1985–2002, 2004.
- [7] Barrie M Baker and MA1951066 Ayeche. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, Vol. 30, No. 5, pp. 787–800, 2003.