

# シミュレーション技術によるマルチロボット経路最適化

## Multi-robot Path Optimization by Simulation Technology

阿瀬 始 制御技術部 統括スタッフ 工博  
勘定 義弘 エンジニアリング研究所 生産技術研究部 主査

Hajime Ase  
Yoshihiro Kanjo

複雑なスケジューリングや生産計画システムの開発において、何らかの意味で最適化を目指すにはシミュレーション技術が欠かせない。マルチロボット経路最適化においては、遺伝アルゴリズムによって最適解を得る過程での適応度計算のところでシミュレータを用いている。このシミュレータは経路が与えられたときに作業完了時間やデッドロック回数を求めるものであるが、その内容と適用結果について論じる。

*In developments of complex scheduling and planning systems, simulation technology is necessary in order to aim the optimization in some sense. In multi-robot path optimization, a simulator is used in the calculation of fitness in the genetic algorithm. This simulator finds job completion time and number of deadlocks. This paper discusses the construction of the simulator and its application results.*

### 1. はじめに

スケジューリングや生産計画システムの開発においてシミュレーション技術が利用される目的をまず論じる。スケジューリングを例にとり、スケジューリング対象（機械、工程など）のデータが与えられてから結果が得られるまでのフローの概念を抽象化して Fig.1 に示す。図において、スケジューラ F は与えられたデータ  $(x, p)$  に対しスケジュール  $u$  を作成する。スケジュール  $u$  をスケジューリング対象であるプロセスに適用すると結果である性能指標（納期、コスト、スループットなど）が得られる。ここに、 $x$  は確定的なデータ、 $p$  は不確定的なデータである。この過程においてシミュレーション技術が利用される状況には大別して以下の3つが考えられる。

- (1) スケジュールの事前評価
- (2) スケジューラの性能評価
- (3) スケジューラ的设计

(1) はスケジュール  $u$  あるいはそれを適用して得られる結果  $y$  を事前に評価するために用いる場合である。 $u$  と比較すべき別の  $u'$  を用意し、 $u, u'$  をシミュレータ  $\textcircled{G}$  に適用し得られた結果  $y, y'$  を比較して  $u$  の優位性を検証することが目的である。(2) はすでに何らかの方法によって与えられ

たスケジューラ F の性能を検証する場合である。実際には実施できないようなさまざまなケース（データ  $(x, p)$  の多くの組）を与え、スケジューラ F、シミュレータ  $\textcircled{G}$  を適用して得られた結果から平均的性能、最悪時の性能などを検証するのが目的である。(3) はスケジューラ F の設計時にシミュレータを用いる場合である。プロセス G が特別な性質を持っていたりきれいな形で定式化されたりしている場合には、スケジューラ F が厳密解法あるいは近似解法というような形で構成できる場合がある。しかしながら、現実の多くの問題ではそうはいかず、そういう場合にはスケジューラ F の設計過程でシミュレータ  $\textcircled{G}$  を使って結果を評価しながら、最適な F あるいは最適な  $u$  を求めなければならない。非線形計画法における直接探索法<sup>5)</sup>や組み合わせ最適化におけるメタヒューリスティクス<sup>6)</sup>などはそのような例である。

本稿は、最後の(3)の例としてマルチロボット経路最適化にシミュレーション技術を適用した結果を論じる。

### 2. マルチロボット溶接システムにおけるシミュレーション

橋梁や造船などの重工業においては、近年ロボットによる省力化が積極的になされてきた。さらに生産性の向上を目的として、マルチロボットの形態のシステムが増えつつある。Fig.2 に造船におけるマルチロボット溶接システムを示す。本システムは10台のロボットからなり、各ロボットは9軸の多関節天吊り型で、定盤（縦8m、横16m）の上に配置された Fig.3 に示すようなワークと呼ばれる溶接対象部材の中のいくつかある溶接箇所を分担して溶接するシステムである<sup>1)</sup>。生産性向上の目的から10台の構成になっているので、Fig.4 に示すように定盤が10分割されてい

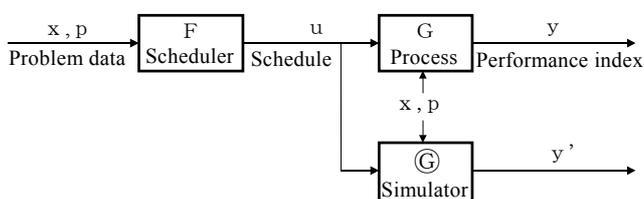


Fig.1 Abstract concepts of scheduling flow

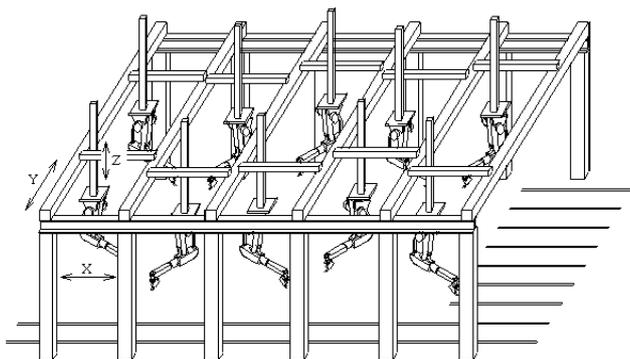


Fig.2 Schematic diagram of multi-robot welding system

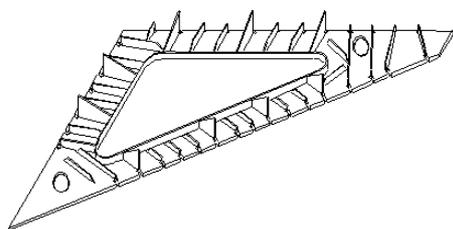


Fig.3 Typical figure of trans panel

る。ただワークには複雑な形状のものがあ、その中の溶接線も多様な方向を向いていることから、ロボットはさまざまな姿勢をとらなければならない。そのため Fig.4 に示す分割領域は完全に排他的なものではなく、実線の境界線を中心として破線で囲まれた帯状領域の隣接ロボットのいずれもがアクセス可能な境界領域を持っている。この境界領域が存在することから、隣接ロボット間の干渉という問題が発生する。干渉が発生すると先に動作を開始したロボットが優先で、遅れて動作を開始しようとしたロボットは干渉状態が解消されるまで一時停止して待たなければならない。干渉の中には、2 台以上のロボットにより互いに干渉状態を解消できない状況が発生することもある。このような状態をデッドロックと呼び、デッドロックが発生すると自動では解消できないので現場担当者による手動介入が

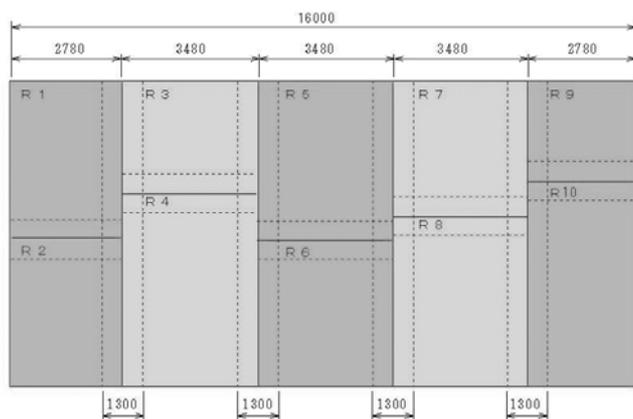


Fig.4 Working area and its division

必要となる。干渉による一時停止やデッドロックが発生すると、意図したほどの生産性が得られないということも起こりうる。それゆえ、デッドロックの回避はもちろんのこと干渉の発生もできるだけ少なくして高い生産性を確保できるシステムの開発が囑望されていた。この課題に対して、当社は遺伝アルゴリズムを適用し、マルチロボットに割り当てる溶接線単位の作業順序、すなわち、溶接経路を最適化することにより、溶接完了時間を最小化するシステムを開発した<sup>2),3)</sup>。また、実操業中に作業の進行が計画から多少ずれたとしても、それ以後も有効なロバスタなスケジュールを作成するシステムも開発した<sup>4)</sup>。たとえば溶接完了時間最小化問題はネットワーク上の最短経路問題と見なすことができるが、アークを通過する時間がこれまでに通過したパスに依存するというタイプの問題になり<sup>2)</sup>極めて解くのが難しい問題である。加えてロボット間に干渉があるので各ロボットの動きをあらかじめ陽に記述することができず、厳密な最適解を求めるアルゴリズムは期待できない。そこで、1 章で述べたように与えられた溶接順序に対し、溶接完了時間やデッドロック回数を導出するようなシミュレータが必要となるのである。

このようなシミュレータの実現方法としては、一定の時間ステップごとに逐次その時点での状態を更新するという方法がある。マルチロボットシステムでは、各ロボットの動作は、(1) 溶接開始点への移動、(2) 下降、(3) センシング、(4) 溶接、(5) 上昇、(6) 移動、(7) ワイヤカット+ノズル清掃/交換+旋回、という7つの基本動作に分解でき、溶接本数分だけこれらを繰り返すのであるが、各基本動作はそれが開始されればその終了時刻がわかり、あえて途中の状態を計算する必要はない。そこで、以下に示す事象駆動型システムで実現した。これは離散系に対してはよく用いられる手法である。

#### ステップ1：初期設定

各ロボットの初期位置、最初の溶接番号、動作番号を設定する。各ロボットの動作開始可能時刻の小さい順にソートし、開始待ち行列を作る。

デッドロック回数を0とする。

#### ステップ2：終了判定

開始待ち行列の中に要素がなければ終了、あればステップ3へ行く。

#### ステップ3：干渉領域設定

開始待ち行列の先頭要素であるロボット番号  $k$  と時刻  $T_k$  を取り出し、開始待ち行列からは削除する。ロボット  $k$  のこれからの溶接番号と動作番号と現在位置から、干渉領域  $D_k$  を張る。

#### ステップ4：干渉判定

ロボット  $k$  と干渉可能性のあるロボット  $m$  に対し、1つ前の動作の干渉領域  $D_m$  を求め、 $D_k$  と  $D_m$  の重なりを調べる。

## (1) 重なり (干渉) がある場合

干渉相手のロボット番号をロボット  $k$  の干渉相手メモリ  $\{M^{(i)}(k)\}$  に格納する。

$M^{(1)}(k) := m, i := 1$  とする

(1.1) ロボット  $M^{(i)}(k)$  の干渉相手メモリからロボット番号  $k'$  を取り出す。

(1.2)  $k'=0$  なら(1.3)へ行く。 $k' \neq 0$  かつ  $k' \neq k, M^{(1)}(k), \dots, M^{(i-1)}(k)$  であれば  $i := i+1, M^{(i)}(k) := k'$  として(1.1)へ戻る。そうでなければ(1.4)へ行く。

(1.3) 干渉が解消される時間  $T'_k$  を求め、ロボット  $k$  の動作開始可能時刻を  $T'_k$  に置き換え、 $k$  と  $T'_k$  を開始待ち行列の中の適切な位置に挿入する。ステップ 3 に戻る。

(1.4) (デッドロック発生) デッドロック回数を 1 増やす。(2)へ行く。

## (2) 重なり (干渉) がない場合

ロボット  $k$  の動作終了時刻  $T''_k$  を求める。ロボット  $k$  の全動作が終了であればステップ 2 に戻る。そうでなければ、ロボット  $k$  の溶接番号と動作番号と現在位置を更新し、 $k$  と  $T''_k$  を開始待ち行列の中の適切な位置に挿入する。ロボット  $k$  の干渉相手メモリを 0 クリアし、ステップ 3 に戻る。

ロボット間の干渉判定方法を Fig.5 を用いてもう少し詳しく説明する。ロボットがある基本動作を行うとき、その動作軌跡を囲む干渉領域を生成する。干渉領域を生成する時点でのロボットの位置と動作の最終位置の間のロボットの軌跡が線分  $s$  であれば、干渉領域はつぎのような長方形になる。

干渉領域 = 線分  $s'_1$  と  $s'_2$  を向かい合う辺とする長方形  
ここに、線分  $s'_1$  と  $s'_2$  は、線分  $s$  を前後に  $\delta_1$  ずつ延長した線分  $s'$  を  $s'$  と直交する方向に  $\pm \delta_2$  平行移動してできる線分である。こうしてできた長方形は  $D_k$  と表されている。この干渉領域  $D_k$  が遅れて動作を開始する他のロボット  $m$  の干渉領域  $D_m$  と重なっている状況を示している。基本動作(2), (3), (5), (7)においてはその動作が終了するまで干渉領域  $D_k$  は不変である。それに対し、基本動作(1), (4), (6)においてはロボットが  $x-y$  平面内で移動するので、ロボ

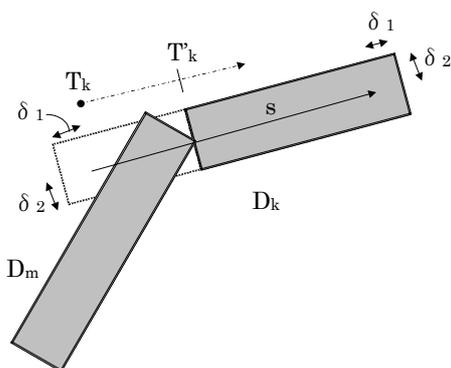


Fig.5 Collision detection and elimination of waiting

ットが移動するにつれて干渉領域  $D_k$  は縮小していく。

Fig.5 は時刻  $T_k$  で動作を開始した後、時刻  $T'_k$  で干渉領域  $D_k$  の破線部分が消滅し、2つの干渉領域の重なりがなくなったことにより時点で干渉が解消されることを示している。

## 3. 溶接経路最適化への応用

溶接完了時間最小化問題はつぎのように定式化できる。

目的関数

$$\max_k C(k) \rightarrow \min_k$$

制約条件

$$N_D = 0$$

ここに、 $C(k)$  はロボット  $k$  の溶接完了時間、 $N_D$  はデッドロック回数である。デッドロック回数をペナルティ化することによりつぎの目的関数式を設定することができる。

$$J \equiv \max_k C(k) + \lambda \times N_D \rightarrow \min_k \quad \dots (1)$$

この最適化問題を遺伝アルゴリズムで解く手順はつぎのようになる<sup>2)</sup>。なお最適化変数は各ロボットの溶接順序(経路)であるので、各ロボットの溶接番号の順序列の集合体が個体の染色体表現である。

ステップ 1: 初期設定

初期個体からランダムに初期集団を生成する。

世代数: =1 とする。

ステップ 2: 交叉と突然変異

元の個体集団の個体から交叉および突然変異により新しい個体を生成する。

ステップ 3: 適応度計算

新しい個体集団の個体の適応度を計算しランクづけする。

ステップ 4: 淘汰

ランク戦略により次世代の個体集団を選択する。

ステップ 5: 世代交代

収束条件を満たしているか世代数が上限に達していれば終了。そうでなければ、世代数: =世代数+1 としてステップ 2 に戻る。

ステップ 3 の適応度計算のところで目的関数式(1)の値が必要になる。目的関数式(1)の値を計算するには各ロボットの溶接完了時間およびデッドロック回数が必要である。そこで 2 章で述べたシミュレータが使われる

溶接完了時間最小化は生産性向上に直結するので重要な要求であることは明らかであるが、シミュレータの中にはロボットの動作モデルが入っているので、

(1) モデルが実際と完全には合っていない

(2) 作業中にトラブルが発生することがある

などの理由から、実作業における作業の進行がスケジュール結果とずれてくることがある。あまりに大きいずれの場

合はしかたがないにしても、軽微なずれであれば得られた計画どおりに作業を続けることができると望ましい。そのためには、ずれによって新たな干渉の発生ができるだけ少ないようなスケジュールが良いことになる。このようなスケジュールあるいはスケジューリングシステムは、Fig.1で言うと、不確定なデータ  $p$  があるクラス  $P$  の中で変動したときの性能指標  $y'$

$$y' = G(x, p')u \quad ; p' \in P$$

を変動がなかったときの性能指標  $y$

$$y = G(x, p)u$$

からそれほど劣化させないものであるということになる。すなわち、この意味でこのようなスケジュールをロバストスケジュールということにする。当社はすでにロバストスケジューリングシステムも開発しており<sup>4)</sup>、溶接完了時間最小とロバスト性最大という2目的最適化問題を、Pareto保存戦略<sup>7)</sup>を組み込んだ遺伝アルゴリズムによって解いている。

#### 4. 結果のビジュアル化

溶接完了時間最小化の結果は、最小化された溶接完了時間と0になっているデッドロック回数が数値的に確認できる。しかしながらこれだけでは情報に乏しい。溶接完了時間最小化システムでは、最適解を求める過程の情報として

- (1) 各世代における個体のランク順での目的関数値
- (2) 最良個体（ランク1位）の目的関数値の世代ごとの推移

を提示している。さらに、最終的に得られた最適解に対し、実際に各ロボットがどのような順序で溶接作業を行うのか、途中で干渉発生はどの程度あるのかなどを視覚的に確認できるようにするため、Fig.6に示すようなビジュアル化システムも開発した。最適解（溶接順序）に対応してシミュレータにより、各ロボットの各動作の（開始時刻、開始位置）、（終了時刻、終了位置）、待ち時間、干渉相手などの情報がファイルに保存されているので、このファイルを読み込み、全ロボットの溶接定盤上での動作が再現できるようになっている。

Fig.6は定盤上に配置された3枚のワークの溶接作業を示している。矢印のついた細い線が溶接線を表す。10個ある正方形がロボットの現在位置を示している。図では左の2つを除く残り8つの正方形からすべてどこかの溶接線に棒状のものが出ているが、これはロボットが溶接トーチを出していることを示しており、これが出ているとそのロボットは先端が当たっている溶接線の溶接作業中であるということになる。溶接線の溶接が済んだ箇所は色が変わっている。基本動作7つに対応してロボットの位置を表す正方形の枠の色が変わり、二次元表現である図では

表しきれない上昇や下降といった動作もわかるようになっている。また、干渉待ちやデッドロックが発生したときも色が変わるとともに、発生時刻や相手ロボット番号などの情報が表示される。

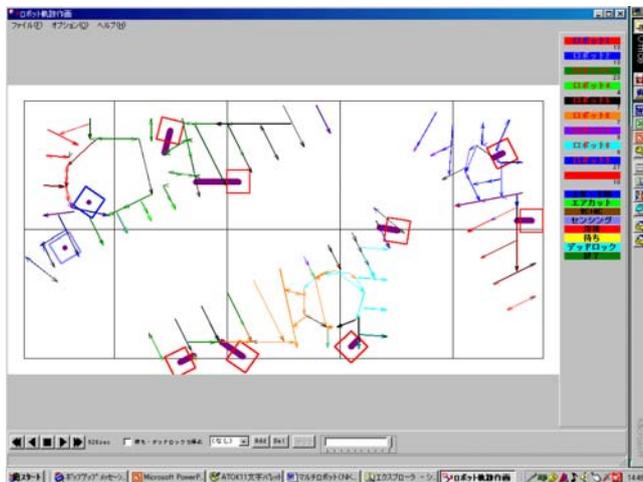


Fig.6 Visualization of simulation results

Fig.7はロバストスケジュールの結果である。スケジュールのロバスト性を検証するため、Fig.6に加えて、各ロボットが境界領域にアクセスする時間帯のガントチャートを合わせて表示している（Fig.7の下半分）。Fig.4で示した境界領域に番号をつけ、色分けして表示している。時間を動かすとFig.7の上半分に見えている図では、各ロボットが溶接作業をしたり移動したり、干渉待ちをしたりという動作を進めていく。一方、下のガントチャートでは画面が左にスクロールされていき、縦線（Fig.7下図では中央付近にある）により現在の時点の作業をしているかがわかるようになっている。同じ時間帯に複数ロボットの同一境界領域のガントが存在すればそこでは干渉が起きていることがわかり、同一境界領域のガントがロボット間で十分な時間

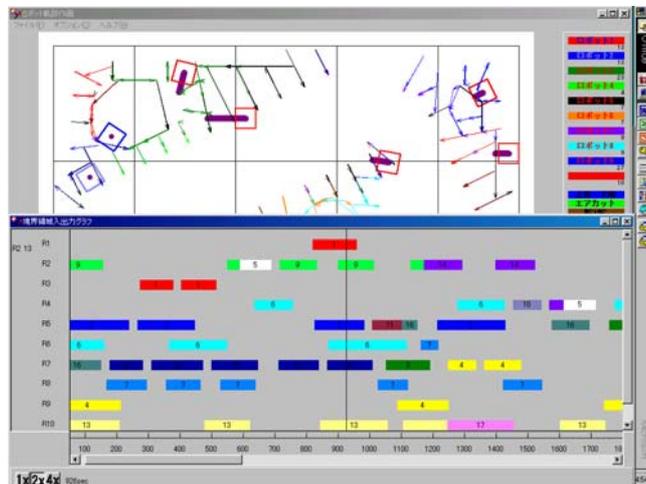


Fig.7 Visualization of robust schedules

離れていると、そこでは多少のずれが発生しても新たな干渉発生の危険性が小さいということであるから、ロバスト性が高いということがわかる。

## 5. おわりに

スケジューリングや生産計画におけるシミュレーション技術の役割を述べ、スケジューリングシステムの設計過程でのシミュレーションの例としてマルチロボットシステムの経路最適化への応用について論じた。シミュレーションと言っても目的に応じてその表現レベルは異なってくる。マルチロボットシステムにおいても、経路最適化に必要なのは溶接完了時間やデッドロックの回数であり、それらを求める上で必要な基本要素（各ロボットの基本動作の開始時刻、終了時刻、干渉の有無、干渉待ち時間など）が計算できるモデルで十分である。溶接トーチの旋回や位置決めなどのメカニカルな細かい動作をモデル化する必要はない。そうは言ってもメタヒューリスティックスなどの反復解法の中でシミュレーションを行う場合、1回のシミュレーションにかなりの計算時間がかかるような場合は、計算時間上の制約からモデルの精度を粗くする必要がある場合もある。そのような場合はスケジューリングシステムの設計にはやや粗いシミュレータ、最終的な性能の検証用には精度の良いシミュレータというような使い分けが必要であろう。

## 参考文献

- 1) 杉谷ほか. “造船小組立マルチロボット溶接システム”. NKK 技報. No.153, pp.72-80(1996).
- 2) 勘定ほか. “マルチロボット溶接における溶接経路の最適化”. システム制御情報学会論文誌. Vol.15, No.4, pp.194-202(2002).
- 3) 勘定ほか. “マルチロボット溶接経路最適化”. 溶接学会第 172 回溶接法研究委員会. pp.1-17(2000).
- 4) 勘定ほか. “マルチロボット溶接におけるロバストスケジューリング”. スケジューリングシンポジウム'2001. pp.59-64(2001).
- 5) ジャコビほか. 非線形最適化問題の反復解法. 培風館(1976).
- 6) 柳浦ほか. “組合せ最適化-メタ戦略を中心として-”. 朝倉書店(2001).
- 7) 玉置ほか. “遺伝アルゴリズムを用いたパレート最適解集合の生成法”. 計測自動制御学会論文集. Vol.31, No.8, pp.1185-1192(1995).

<問い合わせ先>

制御技術部

Tel. 045 (505) 7672 阿瀬 始

E-mail address : aseh@eng.tsurumi.nkk.co.jp