

ニューラルネットワークによる複数機械の協調タスクを含んだ 並列機械スケジューリング

坂口 琢哉[†] 石崎 俊[‡]

[†]慶應義塾大学 SFC 研究所 [‡]慶應義塾大学環境情報学部

複数マシンによる同期・協調的な解決を要求するタスクを含む並列機械スケジューリング問題に言及し、ニューラルネットワークの組合せ最適化を応用したモデルを提案した。具体的には、ニューロンを 3 次元状に配置する事で、タスク、マシン、時刻の要素を一元的に管理し、従来手法と比較して時刻を直接的に決定できるモデルとした。スケジューリングの際の目的関数には総実滞留時間と納期ずれ時間と最小化の二つに言及し、一般的な問題で検証した結果、特に納期ずれにおいてモデルの有効性が示された。また、協調が必要なタスクを一定の割合で含んだ問題を複数設定し、それぞれに対する出力の比較検討を行った。

A Neural Network Model for Parallel Machine Scheduling Problems Considering Tasks with Machine Synchronization

Sakaguchi Takuya[†] and Ishizaki Shun[‡]

[†]Keio Research Institute at SFC, [‡]Faculty of Environmental Information, Keio University

In this paper, we proposed a kind of model for a parallel machine scheduling problem, considering cooperative tasks which are achieved with more than two machines synchronizing and cooperating with each other. Our model is based on neural network architecture, where neurons are displayed by three-dimensional structure to describe tasks, machines and times simultaneously. We adopted a value of total residence time and gap to delivery as goal functions, finding availability of our model especially for the latter one. Finally we applied our model to several patterns of scheduling problems including cooperative tasks, comparing their outputs of each other.

1.はじめに

並列機械スケジューリングとは、フローショップや JSP と並んで実践用途の高い NP 困難な問題であり[1]、特に能力の等しい複数のマシンを用いて大量のタスクを並列的に処理する際、タスクを「いつ」「どのマシンで」処理すべきかを決定するものである。タスクには処理時間や納期が設定されており、その上で納期遅れ時間や納期ずれ時間、総実滞留時間などの目的関数が最小化するようにスケジュールを決定する[2][3]。このとき、タスクによっては例えばコンピュータ間の通信テストなど、達成に複数のマシンの同期や協調が要求されるものも考えられ、本稿では特にこうした「協調タスク」を含むような並列機械スケジューリングの解決が可能な手法を提案する。協調タスクを扱うモデルでは、マシンの同期に対応するために時刻を高い精度でアジャストできる事が不可欠であると考えられる。

従来の並列機械スケジューリングに対する手法としては、GA を用いたモデル[2][4]やノード探索を用いたもの[5]などが提案されている。これらの多くは、先にタスクを担当するマシンとその順序だけを決定し、実際の処理開始時刻はマシンごとにスケジュールの先詰めや後詰めといったヒューリスティックな手法で確定させる。しかし協調タスクではマシンの時刻的な同期が要求されるため、各マシンが単独でスケジュールをずらす事ができず、これらの方法で合理的な処理開始時刻を求める事が難しい。

そこで本稿では、この問題に対しホップフィールド型ニューラルネットワーク[6][7][8]を用いる事で、タスクを処理するマシンと処理開始時刻を同時に決定する新しい手法を提案する。具体的には、ニューラルネットワークの構造においてニューロンを 3 次元的に配置する事で、タスク、マシン、処理時刻を一元的に表現し、これらの組合せ最適化問題として問題を解決

する。また、時刻を直接コントロールする事によるモデルの有効性を検討すると共に、実際に協調タスクを含むスケジューリング問題に取り組み、その応用可能性について考察する。

2.提案手法

2.1 問題設定

本稿で扱う問題では、範囲 $[1, t^{Max}]$ の離散時刻において g 種類のタスクと等性能の $h(<g)$ 個のマシンが与えられ、タスク i には必要処理時間 $c_i(>0)$ と納期 $d_i(\leq t^{Max})$ 、および処理に必要なマシン数 $e_i(\leq h)$ の情報が、一方マシン j には同時刻における複数タスクの処理禁止、進行中のタスク処理の中断禁止といった制約が定められている。本問題ではこれらの条件を満たしつつ、以下に示す二つの目的関数を考慮して、スケジュールを作成する。

- **納期ずれ時間 and の最小化**

各タスクを、できるだけ与えられた納期に合わせて完了させる。

- **総実滞留時間 and の最小化**

各マシンについて、最初のタスクの処理開始時刻から最後のタスクの処理終了時刻までの期間を、できるだけ短縮させる。

2.2 提案モデル

本モデルで扱うニューラルネットワークは、スケジュールを表現するスケジュールニューロンと、それを制御する制御ニューロンの二種類から構成される。

スケジュールニューロン

発火によって「タスク i がマシン j で時刻 $t=k$ で処理されている」事を表わし、本稿では n_{ijk} と表記する。これをタスク軸、マシン軸、時刻軸に沿って 3 次元に配置し、全体の発火分布で一つのスケジュールを表現する(図 1 参照)。

n_{ijk} は内部電位 u_{ijk} と出力値 v_{ijk} を持ち、その発火はそれぞれのタスク軸平面上でニューロンの内部電位を比較する事で決定されるが、そ

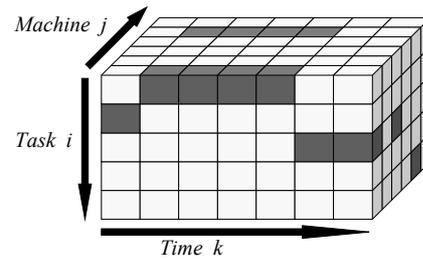


図 1: ニューラルネットワークの構造

の際に個人単位ではなく、常にグループ単位で電位比較や発火を行う点が特徴的である。

具体的には、各ニューロン n_{ijk} は個人の内部電位 u_{ijk} の他に、時間軸正方向に連続した c_i 個分のニューロンの内部電位を加算した u'_{ijk} を保持し、この値がタスク軸平面 i において最大であるニューロン n_{ipq} が発火する。その際、 n_{ipq} だけでなく、 u'_{ijk} の上昇に貢献した $n_{ip(q+1)} \sim n_{ip(q+c_i)}$ のニューロンも連動して発火する。

$e_i \geq 2$ である場合、ニューロンの発火はより複雑になり、マシン軸方向においても連動する。その場合は u'_{ijk} を更にマシン軸方向に上位 e_i 個分だけ加算した u''_{ik} を考え、その最大値 u''_{iq} に貢献したニューロンが最終的に発火する。この結果、それぞれの平面 i において、常に c_i 個分の連続したニューロン列が e_i 本発火している事になり、タスクに対する処理の連続性やマシンの同期といった条件が保障される。

制約ニューロン

スケジュールニューロンを制御する存在として、3次元構造の外部に $m^{(1)}_{jk}$, $m^{(2a)}_{jk}$, $m^{(2b)}_{jk}$ および $m^{(3)}_{ijk}$ という別のニューロンを用意し、それぞれの内部電位と出力値を同じ添字の x と y で表現した以下での動作式で定義する。

$$y_{jk}^{(1)} = \begin{cases} x_{jk}^{(1)} - 1(\text{for} : x_{jk}^{(1)} > 1) \\ 0(\text{otherwise}) \end{cases} \quad (3)$$

$$y_{jk}^{(2a)} = \begin{cases} 1(\text{for} : x_{jk}^{(2a)} > 0) \\ 0(\text{otherwise}) \end{cases} \quad (4)$$

$$y_{jk}^{(2b)} = x_{jk}^{(2b)} \quad (5)$$

$$y_{ijk}^{(3)} = \begin{cases} \frac{(k - (d_i - (c_i - 1)))(k - d_i)}{(t^{Max})^2} & (for: x_{ijk}^{(3)} > 0) \\ 0 & (otherwise) \end{cases} \quad (6)$$

これらは前節で定義した制約や目的関数に対応しており、 $m^{(l)}$ は「同一マシンの同時刻におけるタスク重複」を、 $m^{(2b)}$ は「総実滞留時間」を、 $m^{(3)}$ は「納期ずれ時間」をそれぞれ抑制する。また、 $m^{(2a)}$ は n から $m^{(2b)}$ への結合を束ねる中継ノードの役割を果たし、従来 $(t^{Max} + 1)gh$ t^{Max} 本必要だった結合数を $(2g + t^{Max})ht^{Max}$ 本に抑えている。

各ニューロンの結合式を、以下に示す。

$$x_{jk}^{(1)} = \sum_i v_{ijk} \quad (7)$$

$$x_{jk}^{(2a)} = \sum_i v_{ijk} \quad (8)$$

$$x_{jk}^{(2b)} = \sum_l y_{jl}^{(2a)} \left(\frac{k - l}{t^{Max}} \right)^2 \quad (9)$$

$$x_{ijk}^{(3)} = v_{ijk} \quad (10)$$

$$u_{ijk} = -w_1 y_{jk}^{(1)} - w_2 y_{jk}^{(2b)} - w_3 y_{ijk}^{(3)} \quad (11)$$

ただし、 w_1 , w_2 , w_3 は結合係数である。

これらの結合を通じて、

- (1) スケジュールニューロンが発火し、
 - (2) その状態が制約ニューロンに伝達され、
 - (3) 問題に反応して制約ニューロンが発火し、
 - (4) その結果がフィードバックされ、
 - (5) スケジュールニューロンが電位を修正
- というサイクルが成立する。そしてこの過程を繰り返す事で、ネットワークが表現するスケジュールの内容が改善されていき、全ての $m^{(l)}$ が 0 となった段階で、問題を解決したとして動作を終了する。

3.結果

3.1 使用したパラメータ

提案モデルを実装し、実験を行った。その際、タスク数 g とマシン数 h 、必要マシン数 e_i は、目的に応じて様々な値を設定した。また各タスクの必要処理時間 c_i と納期 d_i は、それぞれ $[1, 5]$, $[c_i, t^{Max}]$ の領域で乱数化し、 $t^{Max} = 50$ とした。最後に、 $w_1 = 0.01$, $w_2 = 0.0001$, $w_3 = 1$ とした。

3.2 既存手法との比較

モデルの有効性を検証するため、既存手法による結果との比較を行った。ここでは、玉置らが実験に用いた問題[2]を適用し、本モデルが出力したスケジュールの納期ずれ時間と総実滞留時間を計算した。実験はモデルが 100 通りのスケジュールを作成するまで繰り返し行い、作業完了までの時間は約 10 分であった。

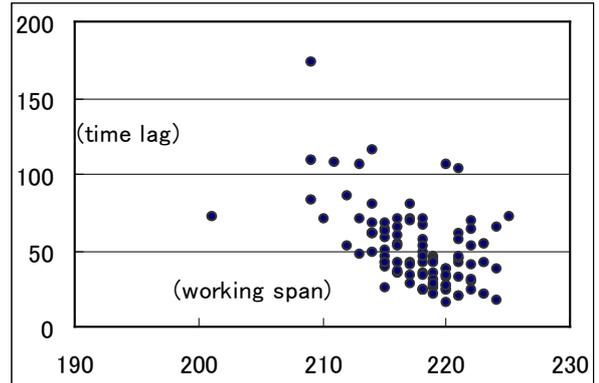


図2: 各出力における納期ずれ時間と総実滞留時間

上に示した結果を玉置らの報告と比較した場合、滞留時間は既存手法が優れている一方、納期ずれ時間についてはほぼ同等か、本手法がやや優れている(図2参照)。この要因としては、既存手法がタスクの処理順序を固定した後でその開始時刻を決定する段階的な方法であるのに対し、本手法ではタスクの処理時刻を直接扱うため、納期に対してよりアジャストされたスケジュールを作成しやすい事が考えられる。しかし一方で納期に固執し過ぎてタスク間に空き時間が出来る場合も多く、滞留時間の改善については今後の検討課題と言える。

3.3 協調タスクに関する検証

本稿において、各タスクは処理に同期が必要なマシン数 e_i の情報を持ち、 $e_i \geq 2$ で協調タスクと定義される。これを含むスケジューリングの問題設定として、本稿ではタスクの延べ処理時間の総和に留意しつつ、(a)~(g)まで7パターンを用意した(表1参照)。

表1: パターンにおけるタスク数(個数)

Pattern	(a)	(b)	(c)	(d)	(e)	(f)	(g)
$e_i=1$	5	20	2	2	0	7	9
$e_i=2$	5	2	11	2	6	1	9
$e_i=3$	5	2	2	8	6	7	1

なお、本実験では $h=4$ に固定し、結果は20試行の平均とした。以下に、出力結果から得られた目的関数の値を示す(図3参照)。

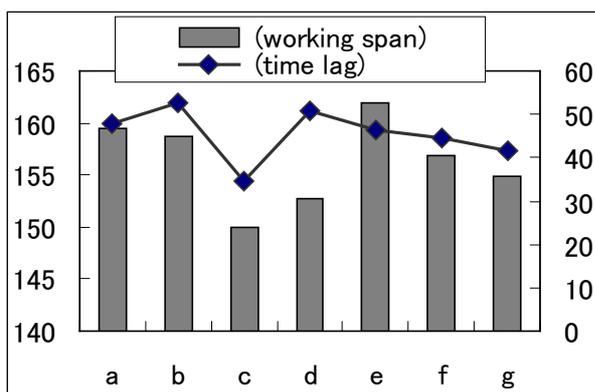


図3: 協調タスクの割合に対する目的関数の値

上図から、特にパターン(c)で納期ずれ時間と、滞留時間ともに大きく減少している事が分かる。(c)は $e_i=2$ の割合が突出した組合せのパターンであるが、これは $h=4$ の状況下においてタスク間の噛み合わせが極めて良く、余剰マシンによるロスが生まれにくい。こうした条件の良さが実験結果にも反映されたと考えられる。

一般に、協調タスクを多く含むスケジューリングほどマシンが頻繁に同期する必要があるため、問題の解決はより困難である。本節で行った実験は比較的小規模なものであったため、モデルはほとんどの場合において短時間で収束した。必要マシン数 e_i を本実験以上に増加さ

せた場合や、多種多様なタスクが混在する場合の検証については、今後の検討課題である。

4.おわりに

本稿では、協調タスクを含む並列機械スケジューリングの手法としてニューラルネットワークを導入し、タスクの順序ではなく時刻を直接決定するモデルを提案した。また、目的関数を総実滞留時間最小化と納期ずれ時間最小化としたところ、特に後者について高速で高い評価が得られ、提案モデルの有効性が示された。更に、協調タスクを含む問題パターンを幾つか用意し、それぞれの結果を比較した。

今後の課題としては、パラメータの最適化や大規模問題への対応の他、新たな目的関数の導入を検討する。一方、協調タスクに関しても、マシン間の能力差や相性などの要素を付加し、実践的な問題への対応を目指したい。そして最後に、本モデルを拡張する事で、フローショップ問題やJSPへの応用にも挑戦したい。

参考文献

- [1] E.L.Lawler, J.K.Lenstra, A.H.G.R.Kan, "Recent Development in Deterministic Sequencing and Scheduling", A Survey, Deterministic and Stochastic Scheduling, D.Reidel Pub. Comp., pp.35-73, 1982.
- [2] 玉置久, 杉本竜彦, 荒木光彦, "非正規目的関数を含む多目的並列機械型スケジューリング問題のモデル化とその遺伝アルゴリズムによる解法", 計測自動車制御学会論文集, 35, 5, pp.662-667, 1999.
- [3] A.Federgruen and G.Moscheiov, "Heuristics for Multimachine Minmax Scheduling Problems with Earliness and Tardines Costs", Management Science, 44, pp.287-299, 1997.
- [4] 当麻淳治, 飯間等, 三宮信夫, "作業割当問題における遺伝アルゴリズムの構成法", システム制御情報学会論文誌, Vol.9, No.1, pp.24-44, 1996.
- [5] 小坂田貴文, 柳川佳也, 宮崎茂次, "並列機械に対する総実滞留時間最小化のもとでの最大総加工時間最小化スケジューリング", 日本経営工学会誌, 49, 5, 293-302, 1998.
- [6] Hopfield,J.J., Tank,D.W., "Neural Computation of Decisions in Optimization Problems", Biological Cybernetics, Vol.52, pp.141-152, 1985.
- [7] Y.Takefuji, K.C.Lee, "A parallel algorithm for tiling problems", IEEE Trans. on Neural Networks, 1, 1, pp.143-145, 1990.
- [8] Takefuji,Y., "Neural Network Parallel Computing", Kluwer Academic Publishers, 1992.