

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^{††}

本稿では並列機械スケジューリング問題の一種として、特に複数のマシンが同期・協調的に処理する「協調タスク」を含み、さらに目的関数として「滞留時間」と「納期ずれ時間」の双方を扱う問題に言及し、これを解決する手法としてリカレント型ニューラルネットワークの組合せ最適化を応用した新たなモデルを提案した。提案モデルは「タスク」「マシン」および「処理時刻」を符号化した3次元構造のスケジューリング層と、問題に内在する制約や目的関数を符号化した制約層から構成される構造とした。特に処理時刻を直接符号化したことで、協調タスクにともなうマシンの同期が容易になった。また、周囲に配置された複数の制約層が中央のスケジューリング層を制御する構造により、複数の制約や目的関数の同時並行的な処理が可能となった。評価実験ではタスク数とマシン数の増加に対する提案モデルの効率性を調査したほか、協調タスクを含まない場合と含む場合の双方のスケジューリングについて、既存手法との比較を行った。その結果、特に納期ずれ時間に関して従来手法より優れた結果が得られ、モデルの有効性が示された。

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

TAKUYA SAKAGUCHI[†] and SHUN ISHIZAKI^{††}

In this paper, we considered a kind of parallel machine scheduling problems with “cooperative tasks” accomplished with two or more synchronizing machines, mentioning to multiple objective functions of “total residence time” and “delivery time lag”. We proposed a novel model for this problem based on a recurrent type of neural network architecture for combinatorial optimization. It consists of one “Schedule Layer” of 3D structure representing combinations of tasks, machines and process time and several “Constraint Layer” representing each constraint or objective function. Direct encoding of process time simplifies a synchronization of machines for cooperative tasks, while the mechanism that peripheral constraint layers control a central schedule layer realizes parallel treatments of multiple constraints and objective functions. We evaluated the performance of our model with increasing the number of tasks and machines, compared it with traditional model for both situations with and without cooperative tasks and finally obtained the superior result of our model especially for delivery time lag.

1. はじめに

与えられたタスクを、限られた作業能力の中で合理的に処理する手順を決定するスケジューリング問題は、数的にも幅広く奥深い問題であり、従来より様々な枠組みについて研究が行われてきた。並列機械スケジューリングはそうした枠組みの1つであり、複数のマシンを用いて大量のタスクを並列的に処理していく手順を解決する問題である。ここで「マシン」とは機

械に限らず、コンピュータや人間である場合も考えられ、一方「タスク」に関しても様々なものが対象となりうるため、実践的な用途の広い問題であるといえる。また現在、本問題はNP-Hardであることが知られている¹⁰⁾。

並列機械スケジューリングでは一般的に、1台のマシンによるタスクの重複処理や処理の中断を許可しない。また、各タスクには固有の処理時間や納期などが与えられ、一方マシンについてはすべてが等性能であると仮定する。しかし、なかにはタスクの処理時間を曖昧にしたもの³⁾や、マシンによって加工能率が異なる設定のもの¹²⁾もあり、様々な発展形を考えることが可能である。一方、スケジューリングの目的関数と

[†] 慶應義塾大学 SFC 研究所

Keio Research Institute at SFC

^{††} 慶應義塾大学環境情報学部

Faculty of Environmental Information, Keio University

しては納期遅れ時間や滞留時間を用いる場合が多いが^(2),5),12)、これに関しても稼働機械数に言及したものの⁽¹¹⁾や、非単調関数である納期ずれ時間や採用したもの⁽⁸⁾、あるいはそれらの目的関数を組み合わせて複合的な解決を目的としたものなども知られている^(4),8)。

こうした様々な並列機械スケジューリングを扱う研究の一環として、本稿では特にタスクの中に「協調タスク」が含まれるような問題を対象とし、これを解決する手法を提案する。ここで協調タスクとは「処理の際に複数のマシンによる同期・協調が必要なタスク」と定義する。協調タスクの例としては複数の人間によるブレインストーミングやコンピュータ間の通信テストなどが考えられ、日常においてもきわめて現実性の高いタスクである。一方、目的関数としては「滞留時間」および「納期ずれ時間」の双方に言及し、複数の目的関数に対する同時並行的な解決を目指す。

協調タスクを扱うモデルでは、マシンの同期を実現させるためにタスクの処理時刻を高精度で調整できる仕組みが必要不可欠であると考えられる。また、複数の目的関数を同時に扱うためには、複雑な問題を記述できるだけの高い表現力がモデルに要求される。こうした問題をふまえ、本稿ではリカレント型ニューラルネットワークを用いた、新たなモデルを提案した。具体的には、ニューラルネットワーク内でニューロンを3次元に配置した「スケジュール層」を用意し、ここで「タスク」「マシン」「処理時刻」を一元的に扱うことで、これらの値を直接的に決定可能なモデルとした。さらにこれらの動作を、スケジュール層の外側に配置された複数の「制約層」によって制御することで、目的関数の高精度かつ多面的な達成を可能とした。

並列機械スケジューリングに対する一般的な数理モデルとしては、これまでにGAを用いたもの^(8),10)や局所探索によるもの^(9),11)、さらには自律分散型のスケジューリング⁽²⁾などが知られている。また、ニューラルネットワークを用いたモデルとしては、Parkら⁽⁵⁾やBonfimら⁽¹⁾によるものがある。これらの多くは、タスクに対してまず割当てとなるマシンとその処理順序のみを決定し、その後マシンごとにスケジュールの先詰めや後詰めを行うことで、実際の処理開始時刻を確定させる。しかし協調タスクを含むスケジューリングではマシン間の時刻的な同期が要求されるため、このようなマシン単独での調整が許されず、単純な方法でタスクの最適な処理開始時刻を求めることは難しい。これに対し提案モデルでは、タスクやマシンとの相互作用も含めて処理時刻を直接扱えるため、こうした同期を容易に実現可能である。

ニューラルネットワークは、スケジュール問題以外にも様々な組合せ最適化問題について応用が進められており⁽⁷⁾、また3次元構造を用いたモデルもTakefujiら⁽⁶⁾や吉池ら⁽¹³⁾による成果が報告されているため、本問題でもその有効性が期待される。

2. 問題設定

2.1 前提

まず、 p 種類のタスク T_i と、これを処理する等性能の q 個のマシン M_j を設定する。ただし、 $p \geq q$ である。次に、扱う時間の範囲として $[0, t^{\text{Max}}]$ の離散量を考え、この範囲内で各タスクは任意の処理時刻に任意のマシンで1度だけ処理されるものとする。その際、1つのマシンは同時刻に複数のタスクを処理できず、一度開始した処理が中断されることもない。

一方、タスク T_i には必要処理時間 c_i と納期 d_i 、および処理に必要なマシンの数 e_i が、それぞれ $0 < c_i \leq t^{\text{Max}}$ 、 $0 < d_i \leq t^{\text{Max}}$ 、 $0 < e_i \leq q$ で設定されている。これらのうち、必要処理時間 c_i と必要マシン数 e_i はスケジューリングにおいて絶対的に満たされるべき値であり、一方納期 d_i については、可能な限りこれに近い時刻で処理を終了させるものとする。なお、必要マシン数 e_i は本稿で新たに導入した概念であり、協調タスクは $e_i \geq 2$ によって定義される。

2.2 制約と目的関数

前節で述べた内容に基づき、以下の3つを制約として設定する。

制約1 マシン M_j が時刻 k において処理できるタスクの数は、1以下である。

制約2 タスク T_i は、連続した処理時間 c_i によって処理される。

制約3 タスク T_i は、制約2を e_i 個のマシンで同期的に満たすことで処理される。

一方目的関数としては、滞留時間と納期ずれ時間に言及したスケジューリングを行う。

滞留時間とはマシンの滞留時間の合計であり、ここでは各マシンが最初のタスク処理を開始してから、最後のタスク処理を終了するまでの時間の総和と定義する。

目的関数1 次の f_1 を最小化する。

$$f_1 = \sum_j \left(\max_i \nu_{ij} - \min_i (\nu_{ij} - (c_i - 1)) \right)^2 \quad (1)$$

ただし、 ν_{ij} はタスク T_i がマシン M_j によって処理される場合の終了時刻である。

一方納期ずれ時間や和は、各タスクの処理を設定された納期ちょうどに合わせて完了することを考えたものであり、各タスクの納期と実際の処理終了時刻との差の絶対値を合計することによって与えられる。

目的関数 2 次の f_2 を最小化する。

$$f_2 = \sum_j \sum_i (v_{ij} - d_i)^2 \tag{2}$$

上記の 2 つの目的関数は、特に納期が極端に早いタスクや遅いタスクにおいてトレードオフとなり、問題解決を困難なものにしている。納期ずれ時間や和を考慮し、これらのタスクを納期に合わせて処理すると、それだけ早い時刻や遅い時刻にマシンが動作することになり、滞留時間や和を増加させる原因になるからである。

3. 提案手法

3.1 モデルの構造

前章で定義した問題の解決手法として、リカレント型ニューラルネットワークを用いたモデルを提案する。モデルの全体的な構造は図 1 のような形をしており、生成中のスケジュールを表現するスケジュール層と、それらを目的関数に従ってコントロールする各種制約層との相互作用によって、適切なスケジュールを生成する。

次節以降で、各層の動作について解説する。

3.2 スケジュール層

スケジュール層に所属するスケジュールニューロン n_{ijk} は、内部電位 u_{ijk} と発火を表す 2 値の出力値 v_{ijk} を持ち、その発火は「タスク T_i がマシン M_j で時刻 k において処理されている」状態を意味する。これらがタスク軸、マシン軸、時間軸の 3 つの軸に沿って 3 次元に配置されることでスケジュール層を構成し、全体の発火分布で 1 つのスケジュールを表現する。

ここで制約 2 と制約 3 の充足を考えたとき、任意のタスク T_i は必ず連続した c_i 個の時間と e_i 個のマシンによって処理されなければならないため、スケジュールニューロン n_{ijk} は同一時間軸上に連続した c_i 個および同一マシン軸上の e_i 個のニューロンとつねに同期し、発火を $c_i \times e_i$ 個単位で連動させる必要がある。図 2 は、タスク T_i の必要処理時間 $c_i = 4$ 、必要マシン数 $e_i = 3$ としたときの、制約 2 と制約 3 を満たして発火した例である。

提案モデルでは、制約 2 と制約 3 をスケジュール層内で充足させるために、つねにこのような状態でニューロン群が発火することを考える。そこで、これらの制約を満たしたグループとして可能なニューロンの組合せをすべて考慮し、その中で内部電位の総和が最大で

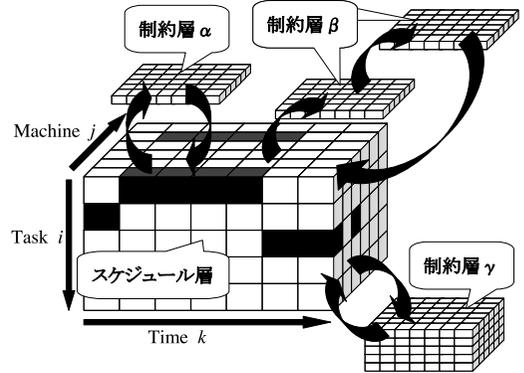


図 1 提案モデルの構造
Fig. 1 The structure of our model.

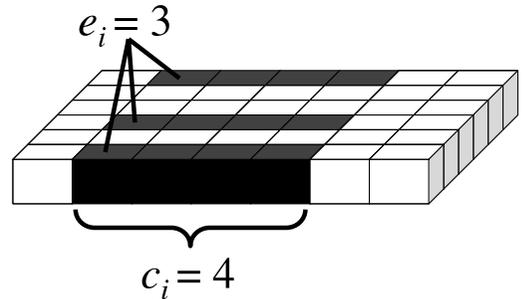


図 2 タスク T_i に関する発火イメージ
Fig. 2 Neuron fires on Task T_i .

あったグループに属するニューロンのみを、集団単位で発火させる。これを実現するために、スケジュールニューロン n_{ijk} の入出力関数を、そのグループに属するニューロンすべての内部電位の総和 U_{iJK} を用いて次式で定義する。

$$U_{iJK} = \sum_{j \in J} \sum_{k \in K} u_{ijk} \tag{3}$$

$$v_{ijk} = \begin{cases} 1 & (j \in J, k \in K, U_{iJK} \geq \max_{J', K'} U_{iJ'K'}) \\ 0 & (\text{otherwise}) \end{cases} \tag{4}$$

ただし、 J は任意の e_i 個の要素からなるマシン集合、 K は連続した c_i 個の要素からなる時間集合であり、いずれもグループの構成要素を表したものである。

スケジュールニューロンの発火は、基本的に“Winner takes all”の考えに基づくものであるが、その比較は個々のニューロン単位ではなく、式 (3) および式 (4) で示したようにグループ単位で比較する点特徴的である。

3.3 制約層

提案モデルには制約層 α 、制約層 β 、および制約層 γ の 3 つの制約層が存在し、それぞれ符号化された

制約や目的関数に従ってスケジュール層の発火を制御する．

このうち制約層 α は制約 1 を符号化した層であり，そこに含まれる制約ニューロン m_{jk}^α は，マシン M_j が時刻 k において複数のタスクを処理しようとした場合に発火してこれを抑制する．以下に m_{jk}^α の内部電位 x_{jk}^α と出力値 y_{jk}^α に対する，動作式と入出力関数を示す． y_{jk}^α はランプ関数であり，2 つ以上のスケジュールニューロンの発火に対して線形に増加する．

$$x_{jk}^\alpha = \sum_i v_{ijk} \quad (5)$$

$$y_{jk}^\alpha = \begin{cases} x_{jk}^\alpha - 1 & (x_{jk}^\alpha > 1) \\ 0 & (\text{otherwise}) \end{cases} \quad (6)$$

一方制約層 β は，目的関数 1 である滞留時間 and の抑制に関係し，同一マシン上で時間的に離れたタスクの処理に対して強く反応する．制約層 β はその中でさらに 2 層構造をとっており，第 1 層の制約ニューロン $m_{jk}^{\beta 1}$ がマシン M_j の時刻 k における全スケジュールニューロンの発火を代表するのに対し，第 2 層の制約ニューロン $m_{jk}^{\beta 2}$ は，第 1 層で発火した任意のニューロンの時刻 k' と自らの時刻 k との差を合計して出力値とする．

この一連の流れに基づき， $m_{jk}^{\beta 1}$ の内部電位 $x_{jk}^{\beta 1}$ と出力値 $y_{jk}^{\beta 1}$ ，および $m_{jk}^{\beta 2}$ の内部電位 $x_{jk}^{\beta 2}$ と出力値 $y_{jk}^{\beta 2}$ の関係は，それぞれ次式のように設定される．

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

$$y_{jk}^{\beta 1} = \begin{cases} 1 & (x_{jk}^{\beta 1} > 0) \\ 0 & (\text{otherwise}) \end{cases} \quad (8)$$

$$x_{jk}^{\beta 2} = \sum_{k'} y_{jk'}^{\beta 1} \left(\frac{k - k'}{t^{\text{Max}}} \right)^2 \quad (9)$$

$$y_{jk}^{\beta 2} = x_{jk}^{\beta 2} \quad (10)$$

最後に，制約層 γ は目的関数 2 である納期ずれ and の抑制に貢献し，その制約ニューロン m_{ijk}^γ はタスク T_i の納期から大きく外れた時刻でのタスク処理を抑制する．その内部電位 x_{ijk}^γ と出力値 y_{ijk}^γ に対する動作式と入出力関数は，次のように記述される．

$$x_{ijk}^\gamma = v_{ijk} \quad (11)$$

$$y_{ijk}^\gamma = \begin{cases} \frac{(k - (d_i - (c_i - 1)))(k - d_i)}{(t^{\text{Max}})^2} & (x_{ijk}^\gamma > 0) \\ 0 & (\text{otherwise}) \end{cases} \quad (12)$$

そして，これらの結果得られた y^α ， $y^{\beta 2}$ および y^γ がスケジュールニューロン n_{ijk} にフィードバックさ

れ，その内部電位 u_{ijk} が次式によって修正される．

$$\Delta u_{ijk} = -w_1 y_{jk}^\alpha - w_2 y_{jk}^{\beta 2} - w_3 y_{ijk}^\gamma \quad (13)$$

ただし， w_1 ， w_2 ， w_3 はそれぞれ正の結合係数である．

なお， y_{jk}^α と $y_{jk}^{\beta 2}$ に関してはニューロンの出力値であっても 2 値的な値をとらず，タスク重複数や時間差によって連続的な値をとりうることに留意されたい．

3.4 アルゴリズム

前節で示した動作式と入出力関数に従い，モデルが問題解決を行う手順を示す．

- step1 すべての u_{ijk} を $[0, 1)$ の一様乱数で初期化
- step2 式 (3)，(4) により v_{ijk} を計算
- step3 式 (5) により x_{jk}^α を，式 (7)，(8)，(9) により $x_{jk}^{\beta 2}$ を，式 (11) により x_{ijk}^γ を，それぞれ計算
- step4 式 (6) により y_{jk}^α を，式 (10) により $y_{jk}^{\beta 2}$ を，式 (12) により y_{ijk}^γ を，それぞれ計算
- step5 すべての y_{jk}^α について値が 0 だった場合，動作終了
- step6 式 (13) により u_{ijk} を更新
- step7 step2 に戻って繰り返す

このサイクルを繰り返すことで，3 次元構造のスケジュールニューロン群によって表現されるスケジュールの内容が徐々に改善されていき，最終的にすべての m_{jk}^α について出力値が 0，すなわちタスクの重複が完全に解消された段階で，モデルは動作を完了する．

4. 実験結果

4.1 準備

実験に際して，タスク数 p とマシン数 q の組合せ，およびタスクの必要マシン数 e_i は，目的に応じて様々な値を設定した．また実験を通じて，必要処理時間 c_i と納期 d_i はそれぞれ $[1, 5]$ ， $[c_i, t^{\text{Max}}]$ の領域でランダムに決定し，一方 $t^{\text{Max}} = 50$ とした．最後に， y_{jk}^α ， $y_{jk}^{\beta 2}$ ， y_{ijk}^γ の値域や収束のしやすさなどを考慮し， $w_1 = 0.01$ ， $w_2 = 0.0001$ ， $w_3 = 1$ とした．

4.2 問題規模の拡大と目的関数値

タスク数 p とマシン数 q を増加した場合に，目的関数値がどのように推移するかを検証した．ここでは協調タスクを考慮せず， $p \in \{3, 4, 5\}$ ， $q \in \{10, 15, 20, 25\}$ とした場合のすべての組合せについて，各 10 試行を行った．図 3 に，タスク数 p とマシン数 q の推移に対する，滞留時間 f_1 と納期ずれ時間 f_2 の平均を表す．なお，両目的関数における全体的な標準偏差は， f_1 で 15.8， f_2 では 35.4 であった．

図 3 から，マシン数の増加に対しては滞留時間 f_1 と納期ずれ時間 f_2 のともに増加傾向にあり，一方タスク数

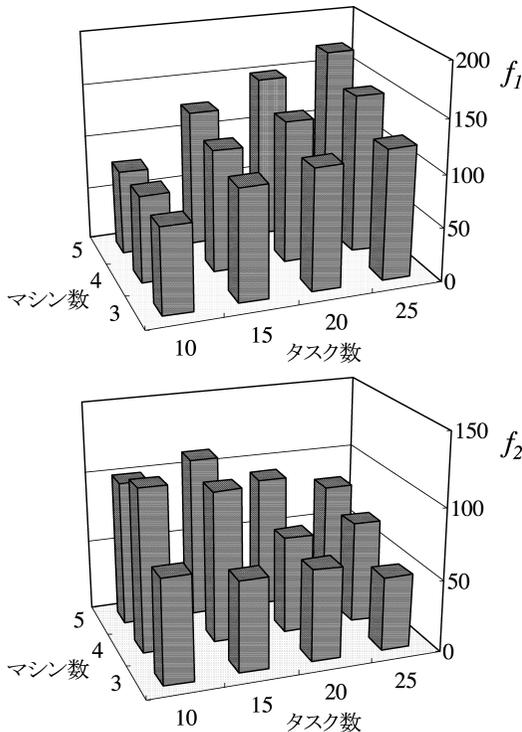


図3 タスク数とマシン数の増加にともなう滞留時間(上)/納期ずれ時間(下)

Fig.3 Total residence time (Upper)/Delivery time lag (Lower) with increasing number of tasks and machines.

の増加に対しては滞留時間(上)は増加するものの、納期ずれ時間(下)については大きな影響がないことが明らかになった。

このうちタスク数の増加を考えた場合、滞留時間(上)に関してはマシンの絶対的な処理時間が増えるため増加はやむをえないが、一方納期ずれ時間(下)に関してはタスク数が増えてもマシンへの割当てを工夫することで、増加をある程度抑制することが可能であり、提案モデルではそれを効率的に実現できているといえる。

一方マシン数の増加を考えた場合、複数のマシンでタスク処理を分散できるようになるため、一般的には滞留時間(上)、納期ずれ時間(下)ともに減少すると予測されるが、ここでは逆の結果となっている。この原因として、マシン数を増やしてスケジュールの自由度を上げたことで逆に問題が簡単になりすぎてしまい、その結果スケジュール生成の過程で目的関数の影響が十分に反映される前に、モデルが収束してしまったことが考えられる。これらをより正確に検証するためには、スケジュールの自由度を小さくし、モデル収束までの時間を十分にとった実験が必要であると思われる。

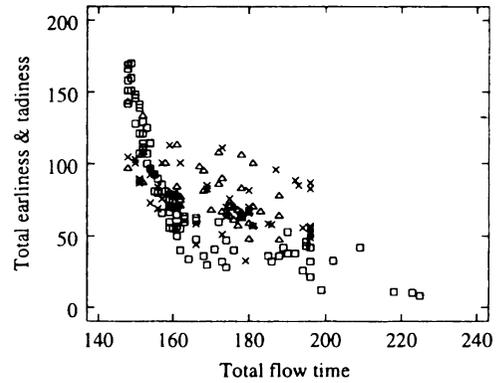


図4 既存手法の結果における目的関数値分布(文献8)より引用)

Fig.4 Distributions of objective functions with traditional model (Ref.8).

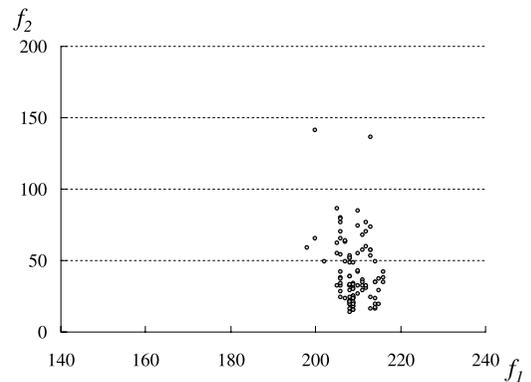


図5 提案モデルの結果における目的関数値分布

Fig.5 Distributions of objective functions with our model.

4.3 協調タスクを含まない問題の評価

ここでは提案モデルの評価実験として、協調タスクを含まない通常の並列機械スケジューリングを対象に既存手法との比較を行った。具体的には、玉置らが実験で用いた $p = 30, q = 5$ の規模の問題⁸⁾を対象とし、提案モデルを適用したスケジューリングを行った。なお、玉置ら自身はこの問題を遺伝的アルゴリズムとヒューリスティクスとの組合せによって解決しており⁸⁾、その結果と提案モデルによる結果とを比較することで評価を行った。

図4および図5に、その結果を示す。図4のグラフが玉置ら自身による既存手法の結果⁸⁾であり、図5が提案モデルによる結果である。各グラフとも、横軸に滞留時間(上)、縦軸に納期ずれ時間(下)をとり、生成したスケジュールをそれぞれの値に従ってプロットしたものである。グラフの左端にあるものほど滞留時間(上)の短いスケジュールであり、一方下側にあるものほど納期ずれ時間(下)の短いスケジュールである。なお、両

目的関数は一部においてトレードオフの関係になっていることに留意されたい。

これらのグラフから、提案モデルが作成したスケジュールは、既存手法のそれに対しほぼ右下端に位置することが分かる。このことは、滞留時間関については既存手法に及ばないものの、納期ずれ時間関については提案モデルの方が優れた結果であることを表している。特に、既存手法では $f_2 < 40$ であるようなスケジュールをほとんど生成できていないのに対し、提案モデルによるスケジュールの多くはこれを満たしており、納期ずれ時間関における有効性を示したものとなっている。

このような結果になる要因としては、既存手法がタスクの処理順序を固定した後でその処理時刻を決定するという段階的な方法であるのに対し、提案モデルでは時刻をニューロンに直接符号化して解探索を行うため、納期に対してより忠実なスケジュールの解を発見できることが考えられる。しかしその一方で納期に固執しすぎるあまり、タスク間に本来なら短縮できる空き時間ができてしまう場合も多く、滞留時間関が改善されない要因にもなっている。

4.4 協調タスクを含む問題の評価

提案モデルでは、スケジュール層においてニューロンが同期的に発火する仕組みを備えているため、協調タスクのようなマシンの同期が要求される問題に対しても対応が可能である。そこでこれを検証するために、様々な協調タスクを含むパターンを用意し、それぞれに対してモデルによるスケジューリングを行った。

具体的には、必要マシン数 $e_i \in \{1, 2, 3\}$ であるような3種類のタスクを考え、それらを一定の割合で混合したパターン $P_1 - P_7$ を用意した。各パターンのタスク含有個数を表1に示す。

各パターンの特徴としては、 P_1 が様々なタスクをまんべんなく含んだ問題であるのに対し、 P_2, P_3, P_4 はそれぞれ必要マシン数が1, 2, 3であるタスクだけが突出して多く、反対に P_5, P_6, P_7 についてはいずれかのタスクだけが相対的に少ないパターンとした。なお、パターンによってスケジュールの密度が極端に異なることを防ぐために、パターンに含まれる全タスクの必要マシン数の合計が一定になるよう配慮した。

これらのパターンに対しマシン数 $q = 4$ として、提案モデルによるスケジューリングを行った。実験は各パターンにつき20試行を行い、出力されたスケジュールに対する滞留時間関 f_1 と納期ずれ時間関 f_2 の平均値を算出した。さらに、同問題を Earliest Due Date first rule (EDD) に基づいたヒューリスティクスによ

表1 各パターンにおけるタスクの含有個数
Table 1 Number of tasks for each pattern.

パターン	e_i	P_1	P_2	P_3	P_4	P_5	P_6	P_7
通常タスク	1	5	20	2	2	0	7	9
2-協調タスク	2	5	2	11	2	6	1	9
3-協調タスク	3	5	2	2	8	6	7	1
タスク数合計		15	24	15	12	12	15	19
必要マシン数合計		30	30	30	30	30	30	30

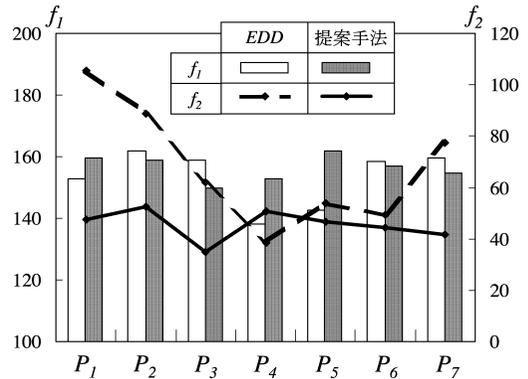


図6 各パターンにおける目的関数値の比較
Fig. 6 The comparison of objective functions for each pattern.

り解決した場合との比較を行うことで、提案モデルの有効性を評価した。

図6において、無地と点線のグラフがヒューリスティクスによる滞留時間関と納期ずれ時間関を、一方網掛け模様と実線のグラフが提案モデルによる滞留時間関と納期ずれ時間関を、それぞれ表している。いずれのグラフも下辺に近づくほど目的関数値の低いスケジュールを実現できていることになり、評価は高くなる。

この図から、EDDはタスクの絶対数が少ないパターン P_4 と P_5 では良い結果を出しているものの、タスク数の多いパターンではその内容にかかわらず精度を落としていることが分かる。それに対し、提案モデルはタスク数に関係なく安定した結果を残しており、特にパターン P_1, P_2, P_3, P_7 においては、モデルの有効性がうかがえる。

また、パターン P_1 においてはそれほどタスク数が多くないにもかかわらず、従来手法では納期ずれ時間関である f_2 の値が非常に高く、評価の低いものとなっている。これは、様々なタスクが混在するスケジューリングの難しさを示唆しているが、提案モデルではこのパターンに対しても良い結果を得ている。この原因としては、EDDなどでタスクの処理順序を先に決定した場合、その制約が後々納期ずれなどの改善の妨げとなることのあるのに対し、提案モデルでは処理順序

に縛られないため、より柔軟性の高いスケジュールを生成できることが考えられる。

一方、パターン間の差異に言及すると、特にパターン P_3 において提案モデルの納期ずれ時間、滞留時間、時間相和が相対的に減少していることが分かる。パターン P_3 は 2 マシンによる協調タスクの含有率が突出したパターンであるが、これはマシン数 $q = 4$ であることとあわせて考えた場合、同時刻にちょうど 2 つの 2 マシン協調タスクを過不足なく処理できることから、パターンとマシン数との相性の良さが反映された結果であると考えられる。

5. おわりに

本稿では、並列機械スケジューリングを扱う新たな手法として、リカレント型ニューラルネットワークを利用したモデルを提案した。具体的には、目的関数を滞留時間最小化と納期ずれ時間最小化に設定したうえで、制約とあわせてそれぞれ制約層に符号化した。さらに、スケジュールニューロンの符号化と発火の入出力関数について工夫し、協調タスクに対応可能なモデルとした。実験の結果、特に納期ずれ時間、時間相和に関して、問題規模の拡大に対する提案モデルの有効性が示された。また、協調タスクを含まない場合のスケジューリングについて、既存手法である遺伝的アルゴリズムによる結果と比較し、特に納期ずれ時間、時間相和についてその優位性を示すことができた。一方、協調タスクを含む場合のスケジューリングについては、その種類や含有率の異なる様々なパターンを用意し、各パターンについて EDD によるヒューリスティクスとの比較を行うことで、提案モデルの有効性を明らかにした。

今後の課題としては、パラメータの最適化による精度の向上や大規模問題への対応、新たな目的関数の導入などがあげられる。また、協調タスクに関する検証もさらに掘り下げて、その可能性を追求していく。さらに、ここでは等性能として扱ったマシン間に能力差や相性といった関係性の要素を付加し、より実践的な問題への対応を目指したい。そして最後に、モデルを拡張することで、フローショップ問題やジョブショップ問題などにも適用させていきたい。

参考文献

1) Bonfim, T.R. and Yamakami, A.: Neural Network Applied to the Coevolution of the Memetic Algorithm for Solving the Makespan Minimization Problem in Parallel Machine Scheduling, *Proc. VII Brazilian Symposium on*

Neural Networks, p.197 (2002).

- 2) 飯間 等, 福井敦彦, 三宮信夫: 並列機械問題における自律分散型スケジューリングシステムの一構成法, システム制御情報学会論文誌, Vol.9, No.2, pp.97-99 (1996).
- 3) 董 彦文, 北岡正敏: ファジィ作業時間をもつ非一様型並列機械スケジューリング問題, 日本経営工学会論文誌, Vol.51, No.1, pp.10-16 (2000).
- 4) 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, 日本経営工学会論文誌, Vol.49, No.5, pp.294-302 (1998).
- 5) Park, Y., Kim, S. and Lee, Y.: Scheduling Jobs on Parallel Machines Applying Neural Network and Heuristic Rules, *Computers and Industrial Engineering*, Vol.38, No.1, pp.189-202 (2000).
- 6) Takefuji, Y. and Lee, K.C.: A Parallel Algorithm for Tiling Problems, *IEEE Trans. Neural Networks*, Vol.1, No.1, pp.143-145 (1990).
- 7) Takefuji, Y.: *Neural Network Parallel Computing*, Kluwer Academic Publishers (1992).
- 8) 玉置 久, 西野悦雄, 阿部重夫: 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的アルゴリズムによる解法, 計測自動車制御学会論文集, Vol.35, No.5, pp.662-667 (1999).
- 9) Tian, Y., Sannomiya, N., Nakano, T. and Tu, Z.: A Local Search Based on Simulations of Fish School Behavior and its Application to an Optimal Scheduling Problem, *Proc. IEEE International Conference on Systems, Man and Cybernetics*, Vol.5, pp.684-689 (1999).
- 10) 当麻淳治, 飯間 等, 三宮信夫: 作業割当問題における遺伝的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.24-33 (1996).
- 11) 山名隆史, 飯間 等, 三宮信夫: 稼働機械数の最小化を目指した並列機械スケジューリング問題に対する局所探索法の適用, 電気学会論文誌 C 編, Vol.123, No.8, pp.1486-1492 (2003).
- 12) 柳川佳也, 宮崎茂次, 平方 敏, 寺戸信幸: 加工能率の異なる並列機械スケジューリング, 日本機械学会論文集 C 編, Vol.62, No.595, pp.346-352 (1996).
- 13) 吉池紀子, 北端美紀, 武藤佳恭: ニューラルコンピュータの「いろは歌」作成への応用, 情報処理学会論文誌: 数理モデル化と応用, Vol.42, No.SIG5 (TOM4), pp.99-106 (2001).

(平成 17 年 7 月 28 日受付)

(平成 17 年 10 月 21 日再受付)

(平成 17 年 11 月 17 日採録)



坂口 琢哉

1977年生．1999年慶應義塾大学環境情報学部卒業．2001年同大学大学院政策・メディア研究科修士課程修了．2004年同博士課程単位取得退学．現在同大学 SFC 研究所訪

問所員．ニューラルネットワーク・コンピューティング，人工知能，認知科学等に興味を持つ．博士（政策・メディア）．



石崎 俊（正会員）

1970年東京大学工学部計数工学科卒業．同学科助手を経て1972年通産省工業技術院電子技術総合研究所勤務．1985年推論システム研究室長．自然言語研究室長を経て1992

年から慶應義塾大学環境情報学部教授．1994年から政策・メディア研究科教授兼任．ニューロインフォマティクス，認知科学，自然言語処理等に興味を持つ．
