

ファジーニューロによるアクティブスケジューリングの研究

旭川高専 ○宮川有樹、渡辺美知子、古川正志

要旨

多工程で仕事量の多い長期のジョブショップスケジューリング問題に対し、従来の静的な解法では長大な計算時間を要し、アクティブな工場の状態の変化に対応できない。そこで本研究ではファジーニューロを用いて時間経過と共に最適なディスパッチルールを随時選択し、アクティブスケジューリングの数値計算実験を行った。

1. はじめに

工場の状態を判断しながらスケジュールを決定していくアクティブスケジュールは、静的なスケジュールと比較してまだその方法論が確立していない。本研究では工場の典型的な状態に対するディスパッチルールをニューロで学習させ、工場の状態をファジーで判断させながらニューロからディスパッチルールを導くアクティブスケジューリング法を提案する。

2. ジョブショップスケジュール問題

サービスを待っている対象群にサービスを与える順序を定める問題はジョブショップスケジュール問題と呼ばれ、一般に製造工場内のモデルとして次のように述べられる。n個の仕事とm台の機械 M_q ($q=1, 2, \dots, m$) に対して各仕事を加工する機械の順序と、各仕事の各機械上での処理の時間が既知のとき、所与の目的関数を最適化するように全仕事を全機械にスケジュールする。

3. ディスパッチルールを結合したリアクティブスケジュール

3.1 ディスパッチスケジュール

作業時間等、各作業のある評価の大きさの順に作業を並べ、その順番によって、各機械の作業開始時刻を決めていく確定型手順のことを指す。この手順では、作業を機械にかける順番と同じ順番で決定が行われ、作業の選択をその作業を機械にかける寸前に行うことができる。

3.2 GA とディスパッチスケジュールの比較

GA (遺伝的アルゴリズム) や BAB (ブランチ・アンド・バウンド) 法を用いて静的なジョブショップスケジュール問題を最適化する方法はこれまでに種々提案されている。これらの方法の場合、仕事量の多い長期のスケジュールでは長大な計算時間を要し、アクティブな工場の状

態の変化に対応できない。また、通常のディスパッチルールでは計算時間は短いスケジュール的にはあまりよいものは得られない。

そこで工場の状態とその状態にあったディスパッチルールをニューロで学習し、学習済みのニューロを適用することで、スケジュールの最適化を計る。

3.3 適用ディスパッチルールの選択

以下のディスパッチルールを採用する。

SPT : 作業時間が最小の差し迫った作業から順に選択

LPT : 作業時間が最大の差し迫った作業から順に選択

SDT : 各仕事の全作業時間と差し迫った作業時間の割合が最小のものから順に選択

LDT : 各仕事の全作業時間と差し迫った作業時間の割合が最大のものから順に選択

SMT : 各仕事の全作業時間と差し迫った作業時間の積が最小のものから順に選択

LMT : 各仕事の全作業時間と差し迫った作業時間の積が最大のものから順に選択

SRPT : 各仕事の残り作業時間が最小の作業から順に選択

LRPT : 各仕事の残り作業時間が最大の作業から順に選択

3.4 ニューロによるディスパッチルールのファジー化と学習

ニューロへの入力として、以下の変数を用いて、3つの工場の状態を定義する。

M_i : i番目の機械 ($i=1, 2, \dots, n$)

O_i : M_i の作業数

$P_{i,j}$: M_i による j番目の仕事の加工時間

$E_{i,t}$: t時間における M_i の加工が終了した作業数

$F_{i,t}$: t時間における M_i の加工が終了した時間

(1) 残り作業数の割合 : Rn

$$Rn = \frac{\sum_{i=1}^n Oi - \sum_{j=1}^n Ej}{\sum_{i=1}^n Oi} \times 100[\%] \quad (1)$$

(2) 残り作業時間の割合 : Rm

$$Rm = \frac{\sum_{i=1}^n \sum_{j=1}^{Oj} Pi,j - \sum_{i=1}^n \sum_{j=1}^{Ej} Pi,j}{\sum_{i=1}^n \sum_{j=1}^{Oj} Pi,j} \times 100[\%] \quad (2)$$

(3) 最速完了時間の割合 : Smax

$$Smax = \frac{\max_{i=1}^n (Fi)}{\max_{i=1}^n (Fi + \sum_{j=B_{i+1}}^{Oj} Pi,j)} \times 100[\%] \quad (3)$$

また、これらの状態を数値のままニューロへ入力した場合、収束しないことが予想される。よって、このデータをファジー化し、大、中、小の3つの状態に変更し、入力する。

各状態での最適なディスパッチルールは実際にシミュレーションを行った結果をGAの結果と比較し、時間毎に状態を決定し、学習させた。

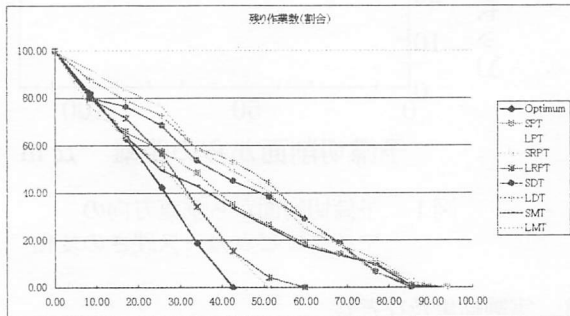


Fig. 1 ディスパッチルールとGAの比較

4. 数値計算実験

単位時間経過毎に状態を計算し、ニューロに入力させその状態での最適なディスパッチルールを選択し、スケジューリングを行った。10機械、10仕事のデータをランダムに作成し、実際にスケジューリングを行った結果を以下に示す。

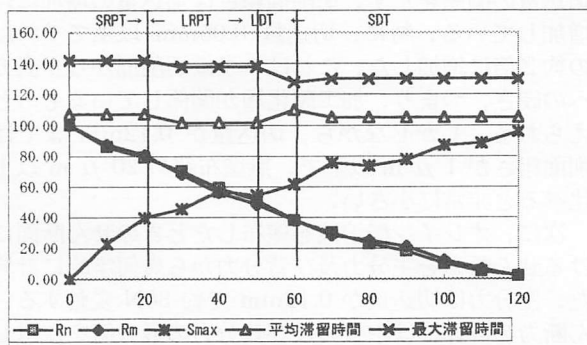


Fig. 2 時間経過による各状態と適用ディスパッチルール

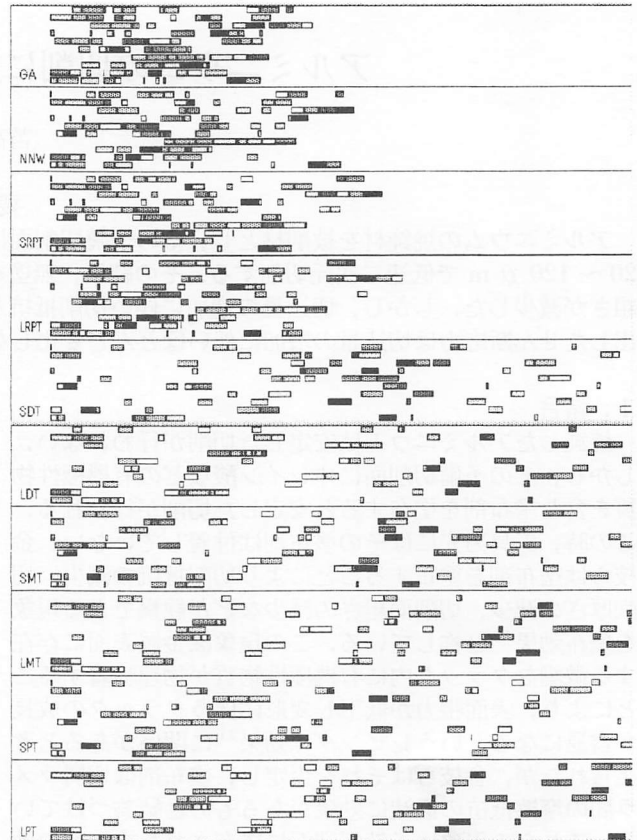


Fig. 3 GA, NNW, 各ディスパッチルールのガントチャート

Table. 1 スケジューリング結果

ルール	GA	NNW	SPT	LPT	SDT	LDT	SMT	LMT	SRPT	LRPT
平均滞留時間	87.99	105.25	267.66	250.67	269.58	273.28	226.97	272.60	107.27	114.83
最大滞留時間	112.49	130.03	297.85	274.41	293.38	296.07	256.89	296.32	142.06	150.94

5. おわりに

今回の実験では一連の仕事の作業に対して1機械1作業に基づいて行った。この為、ディスパッチルールの特徴からSRPTやLRPTのスケジューリングの良さが目立った。また、状態の把握の方法が完全でない為、入力データによってはニューロでの結果がディスパッチルールよりも悪いことがあった。今後はより正確な状態を検討し、ニューロによるリアクティブスケジュールの実用化を計りたい。

6. 参考文献

- 1) 関根智明: スケジューリングの理論、日刊工業新聞社、(1971)
- 2) 鍋島一郎: スケジューリング理論、森北出版、(1974)
- 3) 坂和正敏: ファジ理論の基礎と応用、(1990)
- 4) 中野馨: ニューロコンピュータ、(1989)