遺伝的アルゴリズムと優先規則の融合による 動的スケジューリング* (納期余裕に着目した優先規則の影響)

江口 透^{*1}, 道垣内 将司^{*2}
大場 史憲^{*1}, 村山 長^{*3}

Dynamic Scheduling Using the Mixture of a Genetic Algorithm and a Priority Rule (The Effect of Priority Rule with Consideration to the Due-Date Allowance)

Toru EGUCHI^{*4}, Masashi DOUGAKIUCHI,
Fuminori OBA and Takeshi MURAYAMA

^{**} Graduate School of Engineering, Hiroshima University,
1-4-1 Kagamiyama, Higashi-Hiroshima-shi, Hiroshima, 739-8527 Japan

We have proposed a dynamic scheduling method using the mixture of a genetic algorithm and a priority rule. In the method, a dynamic scheduling problem is assumed to be a series of static problems and scheduled on a rolling basis. In this paper, we examine the effect of priority rule used in the method. For this purpose, two types of the ATC rules having two parameters are used. The first type is the rule whose parameters are tuned for the dynamic scheduling environment. The second type is the rule whose parameters are tuned for the imminent static scheduling problem. Numerical experiments show the followings: The proposed scheduling method searches effective schedules guided by the priority rule and the rule has important effect to the scheduling performance. For the dynamic job shop scheduling, the scheduling method using the first type of ATC rule greatly outperforms the one using the second type of ATC rule. The reason can be explained by considering the due-date allowance contained in the optimized schedules.

Key Words: Scheduling, Production Planning, Production System, Dynamic Scheduling, Rolling Scheduling, Genetic Algorithm, Priority Rule, Job Shop, Tardiness Cost

1. 緒 言

小規模で静的なスケジューリング問題に対しては、さまざまな最適解探索手法が適用可能で、ある程度の計算時間をかけることで性能の良いスケジュールを作成することができる。将来に渡ってどのようなジョブがいつ到着するか既知ではない動的スケジューリング問題に対しても、近未来の情報が既知であると仮定できるならば、動的問題を静的問題の連続と見なし、最適解探索手法をローリングベースで適用するスケジューリング法が利用可能である⁽¹⁾⁽²⁾⁽³⁾など。しかし、この方法における近未来の情報に基

づく最適化はあくまでも部分的な情報に基づく最適化であり、それが最終的なスケジューリング性能の向上に結びつく保障はない。この意味で、最適解探索手法は動的環境になじみにくい側面があり、その適用には注意を要する。

一方、動的スケジューリング問題に対して最も実用的なスケジューリング法として、ディスパッチングルールなどの優先規則を用いる方法がある。優先規則には、動的環境において有効な知識を埋め込むことが可能である。例えば納期遅れコスト(重みつき納期遅れ時間)最小化を目的とする場合に有効な優先規則⁽⁴⁾⁻⁽⁶⁾は、納期遅れ最小化を目的としてスケジューリングを行うにもかかわらず、あるジョブが納期に遅れているからという理由でそのジョブの優先度を上げない方がよいという、長期的全体的な視点に立った知識を利用して動的スケジューリングを行う⁽⁷⁾。

* 原稿受付 2005年3月17日。

^{*1} 正員、広島大学大学院工学研究科(〒739-8527 広島県広島市山1-4-1)。

^{*2} 広島大学大学院工学研究科。

^{*3} 正員、広島大学歯学部(〒734-8553 広島県広島市南区霞1-2-3)。

E-mail: eguchi@hiroshima-u.ac.jp

筆者らは、最適解探索手法と優先規則が持つこのような特徴を考慮し、遺伝的アルゴリズム (GA) による確率的探索と優先規則とを融合した動的スケジューリング法を提案している⁽⁸⁾。その論文では、納期遅れコスト最小化に対して動的環境で性能の良い一つのルールを優先規則として用い、短時間で効率よく定期的な最適化を行うという観点から提案手法の有効性を議論した。しかし、利用した優先規則が動的スケジューリングの性能向上に果たしている役割や影響は明らかになっていない。そこで本論文では、その方法において、融合する優先規則の性質が動的環境での最適化に与える影響を調べることを目的とする。そのため、動的環境で性能の良い優先規則を用いた場合と、近未来の情報に基づく静的問題の最適化に適した優先規則を用いた場合の比較実験を行う。これによって、融合する優先規則は最適化の性能に大きな影響を与え、その性質の違いによって動的環境でのスケジューリング性能は大きく変わることを示す。さらに、将来投入されてくるであろうジョブが割り込めるような余裕時間の量に着目することでその性能の差が生じる原因を説明できることを示す。

以下、第2章から4.2節までは既報⁽⁸⁾と重複するが、論文の独立性を保障するために必要な説明を行っている。4.3節で本論文の目的である融合する優先規則の有効性を調べるために用いるルールとその利用法について述べ、第5章で数値実験を行う。第6章で結論を述べる。

2. 動的ジョブショップスケジューリング問題

本研究で対象とする動的スケジューリング問題は次のように記述される。 M 台の機械により構成されるジョブショップで断続的に到着するジョブを処理する。各ジョブ J_i (i はジョブにつけられた番号) は、 n_i 個の工程 $\{o_j \ (j=1,2,\dots,n_i)\}$ を持ち、工程番号 j の増加順に処理される。各工程 o_j の処理時間を p_j とする。この処理時間や工程順序はジョブ毎に異なっているものとする。どのジョブがいつ到着するかは事前には分からないが、投入が決まった時点ではその工程順序や処理時間は既知であるとする。なお本論文では、段取り時間やジョブの搬送時間、機械の故障は考慮しない。

本研究でのスケジューリングの目的は、次式で表されるジョブの平均納期遅れコスト TC を最小化することである。

Fig.1 Rolling scheduling

$$TC = \sum_{i=1}^N v_i \times \max(0, C_i - d_i) / N \quad (1)$$

ただし、 C_i, d_i, v_i は、それぞれジョブ J_i の処理完了時刻、納期、納期遅れ単位時間あたりのコストであり、 N は評価対象のジョブ総数である。 $\max()$ は括弧内の数値の大きい方の値を示す。

本論文では、各ジョブは平均到着率 λ のポアソン過程でショップに到着し、直ちに投入されるとする。すなわち、本論文で扱うスケジューリング問題は、ジョブが漸続的にショップに到着し、将来に渡ってどのジョブがいつ到着するか事前にはわからない動的スケジューリング問題である。ただし、次章で述べるように近未来に到着するジョブに限ってはその情報が既知とする。

3. 動的スケジューリング問題のローリングスケジューリング

第2章で記述した動的スケジューリング問題を、静的問題の連続としてとらえて、ローリングスケジューリングを行う。すなわち図1において、現時点から時間長さ ΔF の近未来までに投入されるジョブの情報が既知であるとし、その情報を基に最適スケジュールを探索し、得られたスケジュールをある時間長さ ΔT だけ実行する。 ΔT 時間経過したら、その時点から ΔF 将来までの情報を更新して最適スケジュールを探索する。これを繰り返すことによって動的問題のスケジューリングを行う。ここで、 ΔT が ΔF より短く設定されているのは、より新鮮で正確な情報を使ってスケジュールを作成するためである⁽¹³⁾。

いま $J_{mq(l)}$ を、最適化する l 番目の期間でショップ内に存在しているジョブと、将来の ΔF までの時間内に投入が予約されているジョブの集合とすると、各期間での最適化は、 $J_{mq(l)}$ のジョブの情報 (到着時刻 r_i , 工程数 n_i , 各工程 o_j を処理する機械、処理時間 p_j ,

Fig.2 Coding and crossover

納期 d_i , 納期遅れ単位時間あたりのコスト v_i など) を使ってそれらの平均納期遅れコスト TC_i (式(2)) を最小化するように行われる。

$$TC_i = \sum_{i \in J_{\text{shop}}(t)} v_i \times \max(0, C_i - d_i) / |J_{\text{shop}}(t)| \quad (2)$$

ただし, $|J_{\text{shop}}(t)|$ は最適化される対象のジョブの個数を意味する。また, 新たな期間のスケジュールを探索する際には, 未処理工程のみを探索の対象とし, 前の期間で作業に着手済みであるがまだ終了していない工程もそのまま処理を続行するものとして, 探索の対象外とする。このようにして得られたスケジュールを実行するという操作を時間間隔 ΔT 毎に繰り返す。

また, 最適スケジュールを探索する計算時間に制限を設け, その時間を Δt とする。すなわち, スケジュール探索を開始する時刻は, 現在のスケジュールを実行している期間 ΔT が終了する時刻の Δt 前とし, Δt の間で計算された結果に基づいて次の新たな期間 ΔT のスケジュールを実行する。

4. 遺伝的アルゴリズムと優先規則の融合によるスケジュールの探索

4.1 個体の表現と適応度の評価 GA のコーディング法を図2に示す。最適化の対象となるジョブ群 $J_{\text{shop}}(t)$ の各工程 o_{ij} にそれぞれ一つの遺伝子 $gene_{ij}$ を与える。この $gene_{ij}$ は 0 から 1 の間の実数で表現する。これらの遺伝子をまとめて染色体とし, これを一つの個体と考える。

Fig.3 An example of next job selection

個体の適応度は, 各工程に割り当てられた遺伝子 $gene_{ij}$ をジョブ選択の優先度とみなしてスケジューリングシミュレーションを行うことで計算される。このスケジューリングシミュレーションは遅れなしのフォワードスケジューリングであり, 機械がある処理を終えて遊休状態になった時, その機械での処理を待つジョブについて, 対応する工程 o_{ij} の遺伝子 $gene_{ij}$ の値を参照し, その値が最も大きいジョブを選択する。これにより, 各個体に対しそれぞれ一つのスケジュールが作成でき, そのスケジュールの TC_i (式(2)) の値に基づいて個体の適応度を定める。この TC_i が小さいほど適応度が高いと定義する。

さらにこの時, ジョブ選択優先度を遺伝子の値のみで決定するのではなく, ディスパッチングルールなどの優先規則による優先度も考慮して決定することを考える。すなわち, 各個体の適応度を決定するためにスケジューリングシミュレーションが実行されるが, その際のジョブ選択場面においてある優先規則を適用した時の優先度を pri_{ij} とするとき, ジョブ選択優先度を次式で計算する (図3)。

$$gene_{ij} \times pri_{ij} \quad (3)$$

ただし, $pri_{ij} > 0$ とする。なお, 文献(8)では $gene_{ij}$ と pri_{ij} の影響の大きさをコントロールするパラメータ α を導入しているが, 本論文では, 議論を簡単にするため, 単純に式(3)を用いる。

4.2 選択・交叉・突然変異 各個体の適応度の評価は遺伝子と優先規則によって算出される優先度を利用したスケジューリングシミュレーションによって行われるが, GA の遺伝的操作は遺伝子のみに対して行われる。本研究では, 限られた計算時間での性能改善を重視し, 世代交代は, 現世代の個体のうち適応度が上位 20% の個体を次世代にコピーし, 残り 80% は適応度に比例して現世代から選択・交叉させ新たに作成することで行う。交叉は,

図2に示すように選択された二つの個体の同じジョブの同じ工程の間で行われる。それぞれ同じ工程に対応した遺伝子について、70%の確率で一方の個体から、30%の確率でもう一方の個体から遺伝子を引き継ぐ。突然変異は、確率5%でランダムに遺伝子の値を変更する。

なお、全ての遺伝子の値が0.5である個体一つ初期集団に入れておき、最良個体には突然変異操作は行わない。これによって、 ΔT 毎に最適化するスケジュールに関しては、少なくともここで使用した優先規則によるスケジュールと同じ性能のスケジュールが得られることが保障されている。また、世代交代を進めるに従って各世代における最良個体の性能が悪化することはない。

スケジュールの実行は、探索によって得られた最良スケジュールの ΔT までの処理順序をそのまま実行することで行われる。

4.3 融合する優先規則 本論文の目的は、GAと融合する優先規則の影響を調べることである。既報⁽⁸⁾では融合する優先規則を動的環境で良い性能を持つ一つの優先規則に固定していた。これはその優先規則の動的環境での有効性がGAと融合した後も有効であることを期待したからである。しかし、前節までに述べたスケジューリング法の直接的な最適化は近未来のジョブ群 $J_{\text{next}(t)}$ に対して行われるので、その最適化を効率良く行える優先規則と融合して探索を行った方がより有効である可能性もある。ここで、動的環境で有効な優先規則と、近未来のジョブ群 $J_{\text{next}(t)}$ のみの最適化に有効な優先規則の性質は大きく異なると考えられるので、この二通りの優先規則を利用した比較実験を行うことにより、GAと融合する優先規則が動的環境での性能に及ぼす影響を調べる。すなわち、次の二通りの優先規則を用いた比較実験を行う。

- (1) 動的環境で単独で用いる場合に有効な優先規則。
- (2) 近未来のジョブ群 $J_{\text{next}(t)}$ の最適化に対して有効な優先規則。

本論文では、この比較実験を行うための優先規則として、式(4)で表されるATC(Apparent Tardiness Cost)⁽⁴⁾ルールを用いる。

$$pri_{ij} = \frac{v_i}{p_{ij}} \exp \left\{ -\max \left[\frac{(d_i - t - \sum_{m=j}^n p_{im} - \sum_{m=j+1}^n b \times p_{im})}{k\bar{p}}, 0 \right] \right\} \quad \dots\dots(4)$$

この優先規則では、式(4)の pri_{ij} の値が最も大きいジョブ J_i の工程 q_j を次に処理する。ただし、式(4)中の t は個体の適応度を計算する際のスケジューリングシミュレーションにおいてそれぞれのジョブ選択が行われる時刻である。また、 $\bar{p}$ はジョブのその工程での平均処理時間である。式(4)に含まれる $\sum_{m=j+1}^n b \times p_{im}$ は、予測

される待ち時間を意味し、 $(d_i - t - \sum_{m=j}^n p_{im} - \sum_{m=j+1}^n b \times p_{im})$

は、納期までの正味の余裕時間の予測値を意味する。よってこのルールは、 v_i が大きく、 p_{ij} が小さく、納期までの正味の余裕時間の予測値が正の値の場合にその値が小さいジョブに、より高い優先度を与える。この意味でATCルールは、既報⁽⁹⁾で利用した $W(CR+SPT)^{(9)}$ ルールと似た性質を持つルールであるといえる。また、筆者らの数値実験においても生産条件によっては $W(CR+SPT)$ ルールよりも優れたルールであることが確認されている⁽⁹⁾⁽⁷⁾。

さて、このATCルールには二つのパラメータ b と k があり、これによってこのルールの性能は変化する。 b については、この値が大きいくほど待ち時間が長くと予測される。将来に渡って投入されるジョブの情報が完全に分からない動的環境では、短期間の静的問題の最適化を想定した場合よりも、 b を比較的大きめに設定する必要がある。将来投入されてくるジョブが割り込んでくることを考慮すると待ち時間を長く予測する必要があるからである。 k については、式(4)を見ると、この値が小さいほど予測される納期までの正味の余裕時間を拡大して評価することがわかる。すなわち、 k の値が小さいほど、納期までの正味の余裕時間が小さい、すなわち、より遅れそうなジョブにジョブに相対的に高い優先度を与えることができる。動的環境では、遅れの大きなジョブに高い優先度を与えることは必ずしも有効ではない⁽⁷⁾ため、動的環境では k の値はある程度大き目に設定した方が性能が良くなると考えられる。

このように、ATCルールは二つのパラメータ b と k を調節することによって、動的環境で高い性能を持つ優先規則として利用でき、また逆に短期間の静的問題の最適化に有効な性質を持たせることも可能である。これが本論文でATCルールを採用した理由である。具体的なATCルールの二つのパラメータの設定法を次に示す。

方法(1) 動的環境で最適なパラメータ：対象とする動的ジョブショップをATCルール単独でスケジューリングする場合を想定し、統計的に最

適なパラメータの値を事前に調査しておく。これによって得られた動的環境での最適な値にパラメータを固定して利用する方法。

方法(2) 各期間で最適化される静的問題に対して最適なパラメータ：各期間でのスケジュールの最適化を開始する前に、 $J_{shop(1)}$ を対象としてATCルールを単独で用いたスケジューリングシミュレーションを行い、その期間での問題に対する最適なパラメータを調査する。具体的には、 b と k の値をそれぞれ0.5から5.0の間で0.5刻みに変化させてスケジューリングシミュレーションを行う。その結果 TC_i が最も小さかったパラメータの組合せに固定する。このように、期間ごとに直接的なシミュレーションで最適なパラメータの調査を行い、その期間に対して最適なパラメータを持つATCルールを利用する方法。

5. 数値実験

5.1 一期間を想定した静的問題による数値実験

5.1.1 一期間を想定した静的問題の作成 まず、ローリングスケジューリングの各期間での最適化を想定した少数のジョブからなる静的問題を使って、ATCルールのパラメータの違いによる一期間でのスケジュールの探索性能を調べる。 $M=16$, $N=48$, $r_i=U[1,16]$, $p_{ij}=U[5,100]$, $v_i=U[1,10]$ の問題を10題作成した。ただし、 $U[ab]$ は、 a から b の間の一様乱数で決定されることを意味する。ジョブの納期 d_i は次式で設定した。

$$d_i = r_i + k_i \times \sum_{j=1}^{n_i} p_{ij} \quad (5)$$

ここで、 k_i は納期のきつさを表す係数で $U[1.5,4.0]$ で与えた。 r_i はジョブ J_i のショップへの到着時刻である。初期時刻に32個のジョブがショップに到着していると、残りの16個のジョブは、機械稼働率が約80%になるようにジョブの平均到着率 λ を調整することで到着時刻をランダムに決定した。これらの情報が全て既知としてスケジューリングを行う。

5.1.2 ATCルールのパラメータ設定 方法(1)では、10題の問題全てについて $b=1.5$, $k=2.5$ に固定した。文献(4)の動的ジョブショップでは $b=2$, $k=3$ と設定されているが、本論文では次節で扱う動的問題に対する事前実験で調査した結果、 $b=1.5$, $k=2.5$ が最も性能が良かったのでこの値を採用する。方法(2)では、ATCルール単独での性能を問題ごとに調べ、最も TC が小さいパラメータの組合せを選んだ。問題ごとの b と k の

Table 1 The value of b and k for each static problem

No. of static problem	The values of b and k			
	Method(1)		Method(2)	
	b	k	b	k
1	1.5	2.5	0.5	0.5
2			0.5	1.5
3			0.5	1.0
4			0.5	0.5
5			0.5	0.5
6			0.5	1.0
7			0.5	1.5
8			0.5	0.5
9			0.5	2.0
10			0.5	3.0

値を表1に示す。この表から、43節で述べたように、動的環境を想定した方法(1)に比べて一期間のみの最適化を行う方法(2)の方が、 b , k ともに小さ目の値をとっていることがわかる。

5.1.3 納期遅れコストの性能比較 GAの個体数を100、世代交代数を200とし、計算時間の制約 Δt は考慮せずに数値実験を行った。10題の問題に対して初期集団の遺伝子を一様乱数でランダムに発生させて実験を行った。一つの問題につき乱数の種を3回変えて実験した。世代交代に伴う最良個体の TC の変化を図4に示す。ただし縦軸の値は、初期集団の初期値の変更3回×10題の計30回の実験に対する平均値である。

図4から、問題ごとにパラメータを調整する方法(2)の方が当然ではあるが性能が良いことがわかる。初期世代から方法(2)の方が圧倒的に性能が良い。世代交代を進めると方法(1)は急速に TC を小さくできるが、方法(2)との差はある一定値を保った状態で落ち着いている。この結果は、方法(1)、方法(2)ともにスケジューリング性能(適応度)は納期遅れコストのみで評価され、それを最適化するように探索を進めるが、その探索はあくまでも融合する優先規則にある程度拘束された状態での探索であることを意味する。

5.1.4 納期余裕の比較 次に、二つの方法によるスケジュールの差をさらに詳しく調べるため、式(6)で表される DA_i の大きさを調べた。

$$DA_i = \sum_{i \in J_{shop(1)}} \max(0, d_i - C_i) / |J_{shop(1)}| \quad (6)$$

DA_i は、納期に遅れなかったジョブについて、完了時刻から納期までどれくらい余裕があるかを表す指標で、本論文ではこれを納期余裕と呼ぶ。この節では一期間を想定したスケジューリングを考えているので、 $t=1$ とする。また、 $|J_{shop(1)}|=48$ である。

DA_1 の世代交代に伴う変化を調べた結果を図5に示している。図5の縦軸の値は、初期集団の初期値の変更3回×10題の計30回の実験に対する平均値である。これから、方法(1)に比べ方法(2)の DA_1 の値は小さいことがわかる。方法(2)では、納期に遅れていないショップ内のジョブの余裕時間を短縮することによって、TCを小さくしていると考えられる。これに対して方法(1)では、動的環境において適切なパラメータに固定しているため、将来投入されるであろう未知のジョブが割り込めるように、敢えてTCが大きくなっても DA_1 を大きくするような性質を持ったスケジュールが探索されていると考えられる。

さらにこの数値実験において、スケジュールを前半部分と(時刻500以前)と後半部分(時刻500以降)に分割して考え、それぞれの部分で完了したジョブについての DA_1 の変化を調べた。図6と図7に、世代交代に伴う DA_1 の変化を示す。これも計30回の実験に対する平均値で調べた。スケジューリングの前半部と後半部で方法(1)と方法(2)の DA_1 の大小関係が逆転していることがわかる。方法(1)はスケジュール前半の DA_1 が小さい反面、後半の DA_1 が大きい。これは、動的環境で優れた性能を持つATCのパラメータを用いた場合のスケジュールは、将来投入されるであろう未知のジョブが割り込む余裕時間をスケジュール後半部分に比較的多く持つ性質があることを意味する。

なお、これら図5から図7の結果から、納期余裕に関するこのような性質は、優先規則単独のスケジュールに相当する初期世代から見られ、GAと優先規則とを融合して探索を進める過程においても、この性質が保持されながら探索が進行していることがわかる。

5.2 ローリングスケジューリングの数値実験

5.2.1 動的問題の作成 次に、動的環境でのローリングスケジューリングにおける数値実験を行う。ジョブ数1000個の問題を10題作成した。ジョブの納期係数 k_i は $U[2.5, 4.0]$ で与えた。初期時刻に16個のジョブがショップに到着しているとし、機械稼働率が定常状態で約80%になるようにジョブの平均到着率 λ を調整した。その他の条件は、5.1節の条件と同じとした。

5.2.2 納期遅れコストの性能比較 本論文では、 $\Delta T = 250$, $\Delta F = 500$ とし、計算時間制限 Δ はシミュレーション上では1単位時間とした。現実の計算時間は、世代交代数に上限を設けることで制限した。GAの個体数は100とした。

Fig.4 Average tardiness cost of the static problems

Fig.5 Average due-date allowance of the static problems

Fig.6 Average due-date allowance of the early part of the static problems

Fig.7 Average due-date allowance of the late part of the static problems

図8に世代交代数の上限を7通りに変化させた場合の実験結果を示す。ただし、世代交代数0回はATCルール単独でスケジューリングした場合を意味する。縦軸は、10題の問題を方法(1)と方法(2)によってスケジューリングしたときのTC(式(1))を示す。ただし、これは定常状態とみなされるスケジューリング中盤の $N=500$ についてのTCを10題での問題に対する結果で平均した値である。この結果から、世代交代数の上限の設定が全ての場合において、方法(1)が優れていることがわかる。方法(1)の方が、ATCルール単独で15.6%、世代交代数の制限が100回の場合で39.0%性能が良かった。

5.2.3 納期余裕の比較 次に納期余裕の比較を行う。納期余裕は、スケジュール後半部における量がより重要と考えられるが、前節の実験で、スケジュールの後半(図7)と全体(図5)での納期余裕 DA_i の傾向がほとんど同じであったため、本節では、全体での DA_i のみに着目する。図9には DA_i の値を示す。ただし、

$$DA = \sum_{i=1}^L DA_i / L \quad (7)$$

である。 L は性能評価に用いる定常状態と判断される期間の総数で、問題ごとに異なる。図9の DA_i は、10題の問題に対する平均値を示している。これより、世代交代数の上限設定が全ての場合において、方法(1)の納期余裕が大きく、方法(2)の納期余裕は小さいことがわかる。方法(2)に比べ方法(1)の納期余裕は、世代交代数0回(ATCルール単独)で39.6%、世代交代数の制限100回で35.5%大きい。

図10には、典型的な傾向が見られた一つの問題に対して100回の世代交代をさせた場合の TC_i の変化を示す。同様に、図11には DA_i の変化を示す。これより、二つの方法では、時間の進行と共に DA_i の差が大きくなり、 TC_i はその後半部に大きな差がついている様子がわかる。

5.2.4 方法(1)での各期間の問題を用いた性能比較 次に、ここで用いた動的問題に対する各期間における二つの方法の探索性能を確認する。図8から図11に示した実験では、各問題の各ジョブの到着時刻は同じであるが、一つの期間に対して得られるスケジュールは二つの方法で異なり、それが次の期間でのスケジュール問題に引き継がれるので、最適化する問題自身が時間の進行と共にずれてしまう。そこで、二つの方法による一期間での探索性能の差を同一問題で比較して確認するため、方法(1)で探索された各期間でのスケジューリング問題をファイルに保存し、それを方法(2)で解いた場合の性能を調べ、両者を比較した。

方法(1)と方法(2)について、ATCルール単独でスケジュールした場合と世代交代数100回まで探索した場合における TC_i と DA_i の値を調べた。ATCルール単独でのスケジュールの性能は、初期世代の中の遺伝子の値が全て0.5に対応した個体の適応度から得られる。まず、図10と図11に示した問題と同じ問題に対し、方法(1)の場合の定常状態と見なされた各期間での初期状態をファイルに保存し、それを方法(2)で最適化した場合の結果を、方法(1)による結果と合せて図12~図14に示す。図12は初期世代(ATCルール単独に相当)での TC_i で、図13は世代交代100回後の TC_i である。これを見ると、各期間を分離して比較するこの実験で

Fig.8 Average tardiness cost of the dynamic problems

Fig.9 Average due-date allowance of the dynamic problems

は、5.1節で既に確認したように、その期間の問題に対して直接パラメータを最適化する方法(2)の方が当然ではあるが性能が良い。全ての期間で方法(2)の方が TC_i を小さくできている。また、図12と図13を見ると、40期目付近と95期目付近で、 TC_i に差が生じている。図14には、世代交代数100回後の納期余裕 DA_i の変化を示している。図12と図13で TC_i に差が生じた期間で DA_i の差が確認できる。すなわち、方法(2)は TC_i の値を小さくできているが、反面 DA_i の値も小さくなってしまい、将来投入されてくるジョブが割り込む余裕が小さくなると考えられる。一期間でのこのような差が累積していくことで、動的環境では図10、図11の後半のような大きな差がつく結果になってしまう。

次に図12、図13に示したような TC_i の値を評価対象の期間で平均し、さらにそれを10題の問題で平均した値で比較した。その結果、方法(2)の TC_i の平均値は、方法(1)よりも初期世代で47.9%、100回の世代交代後でも19.8%小さかった。このように各期間での初期状態をそろえた評価の結果、一期間のみに対する最適化には方法(2)が圧倒的に優れていることが再確認された。しかし、ローリングスケジューリングを続けた結果の納期遅れコスト TC の評価では逆転し、図8に示したとおり方法(2)に比べて方法(1)の方が世代交代数の上限100回の場合で39.0%も TC を小さくできる。

Fig.10 Average tardiness cost in the case that the number of generations is limited to 100.

Fig.11 Average due-date allowance in the case that the number of generations is limited to 100.

6. 結 言

本論文では、動的ジョブショップスケジューリング問題を GA と優先規則を融合した探索法によってローリングベースでスケジューリングする方法において、融合する優先規則が最適化に与える影響を調べた。

数値実験によって、融合する優先規則は最適化の性能に大きな影響を与え、その性質の違いによって動的環境でのスケジューリング性能は大きく変わることがわかった。すなわち、高性能な動的スケジューリングを行うためには、動的環境で有効な優先規則と融合して最適化を行う方が良く、将来の短期間のみの最適化を意図した優先規則と融合して最適化を行うと動的環境でのスケジューリング性能は大きく劣る。

さらに、このような差が生じる原因は、動的環境で有効な優先規則と融合して最適化したスケジュールには、将来投入されてくるであろうジョブが割り込めるような余裕時間が多く残されていることにある。このような性質は、動的環境で有効な優先規則が単独で適用される場合にも持つ性質である。提案手法は、動的環境においてこのような有効な性質をもつ優先規則にガイドされながら最適化を行うこと

Fig.12 Comparison of average tardiness costs scheduled using the ATC rules for the problems solved by the method (1) in the case that the number of generations is limited to 100.

Fig.13 Comparison of average tardiness costs when the number of generations is 100 for the problems solved by the method (1) in the case that the number of generations is limited to 100.

Fig.14 Comparison of average due-date allowances when the number of generations is 100 for the problems solved by the method (1) in the case that the number of generations is limited to 100.

で、最適化がなじみにくい動的環境においてより効果的に最適解探索手法を利用していると考えられる。

なお、このような動的環境においては、将来が予測できる長さやスケジューリング周期が及ぼす影響が大きいと考えられる。本論文では ΔF や ΔT の長さを固定して数値実験を行ったが、これらがスケジューリングの性能に及ぼす影響を調べ、それらを適切に設定することも重要な課題であり、今後検討する必要がある。

最後に、本研究は科学研究費（基盤研究(C) No.15560098）の援助により行われました。記して謝意を表します。

参考文献

- (1) D.Sun, L.Lin : A dynamic job shop scheduling framework: a backward approach, *International Journal of Production Research*, Vol.32, No.4, pp.967-985,(1994)
- (2) I.M.Ovacik, R.Uzsoy : Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times, *International Journal of Production Research*, Vol.33, No.11, pp.3173-3192,(1995)
- (3) E.Kutanoglu, I. Sabuncuoglu : Experimental investigation of iterative simulation-based scheduling in a dynamic and stochastic job shop, *Journal of Manufacturing Systems*, Vol.20, No.4, pp.264-279, (2001)
- (4) A.P.J.Vepsalainen, T.E.Morton : Priority rules for job shop with weighted tardiness costs, *Management science*, Vol.33, No.8, pp.1035-1047, (1987)
- (5) E.J. Anderson and J.C. Nyirenda : Two new rules to minimize tardiness in a job shop, *International Journal of Production Research*, Vol.28, No.12, pp.2277-2292 (1990)
- (6) 江口, 大場, 豊岡: 階層型ニューラルネットワークを用いた動的ジョブショップスケジューリング (第 1 報), 日本機械学会論文集 C 編, Vol.68, No.672, pp.2494-2501 (2002)
- (7) 江口, 大場, 豊岡: 階層型ニューラルネットワークを用いた動的ジョブショップスケジューリング (第 2 報), 日本機械学会論文集 C 編, Vol.68, No.673, pp.2817-2824 (2002)
- (8) 江口, 大場, 小崎: 遺伝的アルゴリズムと優先規則を融合した動的スケジューリング, 日本機械学会論文集 C 編, Vol.71, No.703, pp.1047-1053 (2005)