2000年度日本オペレーションズ・リサーチ学会
秋季研究発表会

リソース制約型スケジューリング問題解法スケジューラの開発と評価

01507203　愛知女子短期大学　＊堀尾　正典　HORIO Masanori
01204223　南山大学　鈴木　敦夫　SUZUKI Atsuo

1．はじめに

リソース制約型スケジューリング問題（ＲＣＰＳＰ）は、割り付けるタスク（ジョブとも言う）に、機械や人手などリソース資源の消費制限が付随する問題である。この問題では、タスク間の順序制約の他に、同一時刻で実施されるタスクのリソース消費量が供給量の最大量を超えてはいけないと言う制約が加わる。このため、一般に問題はより複雑となり（ＮＰハード）厳密解を得るのは難しくなるが、その反面、対象となる問題領域も広く、各種ショップ問題やナーススケジューリング問題、時間割作成問題など様々な問題に適用が期待されている[3]。

我々は、汎用的なスケジューラの開発を目的として、このＲＣＰＳＰに着目しモデル化とシステム化を試み、その有用性の検証のため時間割作成問題へ適用を行った[1]。今回はそのシステムに順序制約の充足機能を加え、スケジューリング・ベンチマーク問題[2]を用いて性能評価を行い、実現したアルゴリズムの有効性、今後の検討課題を明確にした。

2．アルゴリズム

2.1 基本構造

本システムでは、一つのタスクを割り付けるたびに幾つかの候補時刻を列挙し次のタスクを割りつけていく、解の列挙と探索を基本としている。有効な時刻が得られない場合は、バックトラックによる解探索を行う。基本ステップを次に示す。

＜基本ステップ＞

①全タスクの中から 2.3 節の基準に従い、一つのタスクＡを選び出す。
②与えらた状況下でＡの実施可能な時間帯を 2.2 節に従いすべて計算し、そこから幾つかの候補時刻を選出する。
③候補時刻が計算できない場合、2.4 節のルールに従いバックトラックを実施し、過去に割り付けた他タスクの別の候補時刻を選出する。
④候補時刻が計算できた時、その一つにＡを割り当てる。Ａによって消費されたリソース条件を計算し新環境を算出する。
⑤全タスク処理なら終了へ。そうでなければ①へ。

2.2 制約充足時間帯の算出

与えらた状況下でのタスクの実施可能な時間帯すべてを、以下の手法を用いて計算する。その上でこれら時間帯より、候補とする時刻をあらためて選別する。

＜基本ステップ＞

①リソース制約にはパッキングアルゴリズムを用いる[1]。
②順序制約には、先行・後続タスクの割り付け状況から開始可能時間帯を求める。その際、自分の最遅開始時刻を上界に用いる。
③上二つの時間帯のＡＮＤを計算する。

2.3 タスク処理順位計算

今回の研究では、特にタスクの処理順位（どのタスクから割り付けていくか）が結果に大きな影響を与えることが分かった。よいタスクから割付作業を行えばほとんどバックトラックすることなしに、最短時刻でのスケジューリングに到達するケースもある。そこで本システムでは、よいタスクの評価を定量的に行うため、割り付けにくさを示す値、固定度を定義し用いる（現状は、タスクの各消費パターン面積の供給パターン面積に対する占有率の合計を固定度にしている）。

＜基本ステップ＞

①固定度を各タスクに対して求める。
②この値をタスクの処理時間とみなし疑似的なクリティカル・パス（疑似ＣＰ）を計算する。
③最初に処理するタスクを、このＣＰ上のタスクを主にして選び出す。スケジューリングの最初に処理されるこれらこれらタスク群をフェーズ０（疑似ＣＰ上）のタスクと呼んでいる。
④フェーズ内タスクを固定度でソートして、一番大きなものを一つ取り出し割り付けを行う。
⑤フェーズ内タスクがすべて無くなれば、先行タスクが割り付け済みのタスクを主として、次フェーズを形成し④へ。すべてのタスクが処理できれば終了へ。（図１参照）

2.4 バックトラックの処理

また、本システムでは、良解を得るためにバッ

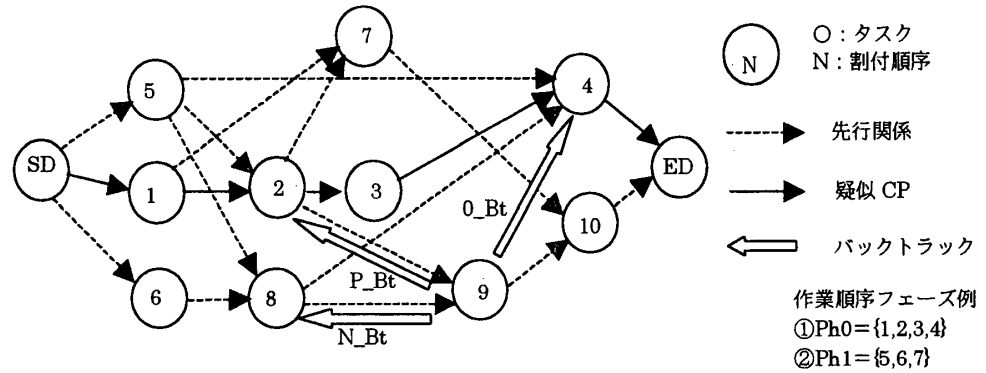

図 1 .処理順序とバックトラックの決定

クトラック(以下ＢＴと略す)手法を用いている。ＢＴを効率的に実現するためには、当然のことながら、よい実現ルール（いつどこに戻るのか）が必須となる。今回は効果的なＢＴ実現のため、戻りポイントが３種類あるマルチモードＢＴを考案した。(図１参照)

＜基本ルール＞

①Normal_ＢＴ（通常時）

今の自分に一番近い候補時刻解へ戻る。通常時発生する。

②Prec_ＢＴ（順序制約のみで失敗した時）

先行タスクの割り付け時刻がネックになっている場合、そのタスクまで戻って次候補時刻を探索する。

③Ph0_BT（一定回割り付けに失敗した時）

失敗タスクをフェーズ０のタスク列中に加え、スケジューリングに再挑戦する（難しそうなタスクなので割付処理順位を上げてやり直す)。

３．ＲＣＰＳＰベンチ問題への適用と評価

本システムをスケジューリング・ベンチマーク問題である PSPLIB[2]に適用した。幾つかのテスト事例とその具体的な実行結果については当日発表する。現状の結果は次の通りである。

表 1 .現状得られている実行結果

問題ＮＯ	知られている最短スケジューリング期間	本システムが計算した最短スケジュール結果	本システムが処理に要した時間
J301_1SM	43	43	2.91sec
J301_2SM	47	47	2.09sec
J601_4SM	91(*)	91	5.38sec
J605_3SM	81(*)	95	21.32sec
J609_1SM	88(*)	102	25.10sec

注１：(*) は近似解であることを示す。

注２：使用計算機は、PentiumIII660Mhz の PC で実行。

４．まとめと今後の課題

J605、J609 は一つのタスクに複数のリソース制約が付随するやや複雑な問題となる。このレベルでは、リソース制約も厳しくなり、順序制約との整合性違反が多くなり、頻繁にＢＴを繰り返す結果となった。今回の研究で明らかとなった主な課題を次に示す。

①タスク選別方法の改良

先述したが、タスクを割り付けていく順番がスケジューリングの結果に大きく影響を与えている。今後、フェーズ０のタスク群のよい選別方法を各種パラメータを変更して計測していく必要がある。

②ＢＴルールの追加

マルチモードＢＴは今回のシステムに良好な結果を与えている。しかし、より複雑な問題に対応するため、リソース制約と順序制約の整合性違反の場合など多様なケースについて、より効率的なＢＴルールを追加しなければならない。

③候補時刻の選別方法

より適切な候補時刻の抽出方法。

これらのためにも、今後多くの問題を実施し得られた結果を随時フィードバックしていきたい。

参考文献

[1]堀尾正典、鈴木敦夫：リソース制約型スケジューリング問題のモデル化と時間割作成システムへの適用，日本オペレーションズ・リサーチ学会 2000 年度春季研究発表会アブストラクト集，1-C-2，pp.44-45.

[2]R. Kolisch and A. Sprecher: PSPLIB-A project scheduling library, *European Journal of Operational Research*, Vol96, 1997, pp.205-216.

[3]野々部宏司、茨木俊秀：汎用スケジューラーRCPSP によるアプローチー，オペレーションズ・リサーチ，第４５巻第３号，2000，pp.10-16.