4H-06

タスクスケジューリング問題における DF/IHS 法のハッシュテーブルを用いた探索ノード数削減

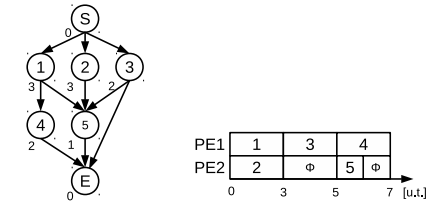
松瀬 弘明 †　　中村 あすか ††　　富永　浩文 ††　　前川 仁孝 †
† 千葉工業大学情報科学部情報工学科 ††　千葉工業大学大学院情報科学研究科情報科学専攻

1　はじめに

DF/IHS(Depth First/Implicit Heuristic Seach) 法は，タスクスケジューリング問題の求解に有効な手法の一つである [1]．本手法は，木探索によって最適解を求める手法であり，問題規模が大きいほど探索ノード数が多くなる．DF/IHS 法は，実行可能なタスクをプロセッサ (PE) に割り当てるすべての組合せを部分問題として生成するため，PE に不必要な待ち状態を割当てる部分問題を生成する．このような部分問題は，探索中ノードと探索済みノードを比較することで削減できる [2]．そこで本研究では，DF/IHS 法の探索ノード数を削減するために，探索済みノード情報をハッシュテーブルに格納し，探索する必要のない部分問題を枝刈りする手法を提案する．

2　タスクスケジューリング問題

タスクスケジューリング問題は，n 個のタスクからなるタスク集合 T を能力の等しい m 台の PE で並列処理するスケジュールのうち，スケジュール長が最も短いスケジュールを求める問題である．図 1 に，タスクグラフとスケジュールの例を示す．図 1(a) のノードはタスク，ノード内の数字はタスク番号，ノード左下の数字は処理時間，矢印は先行制約を表し，タスク S, E は処理時間 0 のダミータスクである．また，図 1(b) のガントチャートは，タスクを 2 台の PE に割り当てた例である．図中の ϕ は，PE が待ち状態であることを示す．本例では，S の割当てが 0[u.t.]，E の割当てが 7[u.t.] である．



(a)タスクグラフの例　(b)スケジュールの例

図 1: タスクグラフとスケジュールの例

3　DF/IHS 法

DF/IHS 法は，実行可能なタスクを PE に割り当てるすべての組合せを部分問題として探索木を作成し，探索木の左側から深さ優先探索する．図 2 に図 1 のタスクグラフを DF/IHS 法で探索する例を示す．図中の根ノードは，割当て済みタスクが S，PE の最小時刻が 0[u.t.] の部分問題であるため，割当て可能なタスクが 1, 2, 3, ϕ であり 6 個の子問題を生成する．本手法は，割当て可能タスクに ϕ を含めて部分問題を列挙するため，図中の点線で示すように割当て済みタスクが同じ部分問題を複数生成する．このような部分問題のうち，スケジュール長が長い部分問題は，最適解が得られないため探索する必要がない．

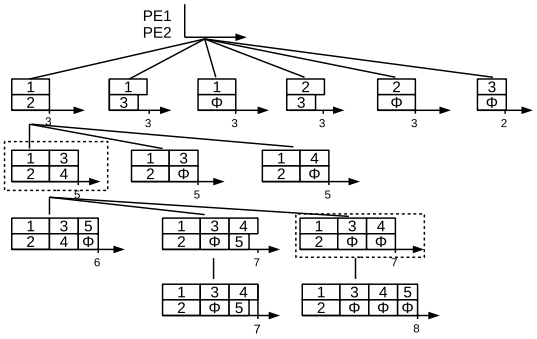


図 2: DF/IHS 法で生成する探索木の例

4　ハッシュテーブルを用いた削減手法

DF/IHS 法は，探索中ノードが探索済みノードの割当て済みタスクに包含されており，以下の条件を一つでも満たすとき，探索中ノードを枝刈りできる [2].

(1) 探索中ノードの PE の最小時刻が探索済みノードの最長時間より大きい

(2) 探索中ノードの各 PE の時刻が探索済みノードより大きく，割当て時刻に実行中タスクが同じ

上記条件を用いてすべての探索済みノードに対して判定を行うと，判定に長い時間を要する．そこで本研究

A Reducing Algorithm of Search Nodes of the DF/IHS Method Using a Hash Table
Hiroaki MATSUSE† and Asuka NAKAMURA†† and Hirobumi TOMINAGA†† and Yoshitaka MAEKAWA†
†Department of Computer Science,Chiba Institute of Technology
††Graduate School of Information and Computer Science, Chiba Institute of Technology

Copyright ©2016 Information Processing Society of Japan.
All Rights Reserved.

では，探索済みノードの情報をチェイン法を用いたハッシュテーブルに格納し，割当て情報を包含するノードに対してのみ判定する．図 3 にハッシュテーブルを用いて枝刈りする例を示す．本手法は，探索中ノードが下界値で枝刈りされない場合，割り当て済みタスク情報を図 3 のようにビット配列に格納しハッシュキーとする．探索中ノードのハッシュキーがハッシュテーブル内にあれば，ハッシュキーに連結するリストの情報を用いて枝刈り判定する．本例のように探索中ノードが枝刈りできない場合は，ハッシュテーブルにタスク情報を追加する．

5 評価

ハッシュテーブルを用いたノード数削減手法の有効性を評価するために，DF/IHS 法とハッシュテーブルを用いる手法の探索ノード数を測定し削減率を算出する．本評価は，ハッシュテーブルの最大バケット数を 10 万とし，標準タスクグラフセット [3] のタスク数が 50 の問題 100 問に対し，PE 数を 2, 4, 8, 16 と変化させた合計 400 問を求解する．また削減率は，式 (1) で求める．

$$\text{削減率} = (1 - \frac{\text{提案手法の探索ノード数}}{\text{DF/IHS 法の探索ノード数}}) \times 100 \quad (1)$$

図 4 に，DF/IHS 法に対する提案手法の探索ノード数の削減率を示す．図 4 より，探索ノード数が減少した問題は 70 問，変化がない問題が 318 問，増加した問題が 12 問であることが確認できた．

探索ノード数が減少した問題は，探索ノード数が多い問題ほど高い削減率が得られる傾向があることが分かる．これは，下界値で枝刈りできなかった多くの部分問題に対してハッシュテーブルを用いた枝刈りが有効に働いたためである．

探索ノード数に変化がなかった問題は，ハッシュテーブルを用いた枝刈り判定が起こってないと考えられる．これは，DF/IHS 法の下界値の精度が良く，多くの部分問題を下界値が枝刈りしたためである．

探索ノード数が増加した問題は，下界を用いた枝刈りの精度が低下した．これは，ハッシュテーブルを用いることで精度の高い暫定解が得られるノードを枝刈りしたためである．図 5 に精度の高い暫定解が得られるノードを枝刈りして探索ノード数が増加する例を示す．本例は，各ノードの下界値 lb が $lb_A > lb_C > lb_B > lb_D$ の関係にある．提案手法は本例のノード B を探索しないため図中の灰色ノードを探索し探索ノード数が増加する．提案手法を用いることで本例のように探索ノード数が増加した問題は，図 4 より DF/IHS 法の探索ノード数が少ないため提案手法で探索ノード数が増加しても短い時間で求解できる．

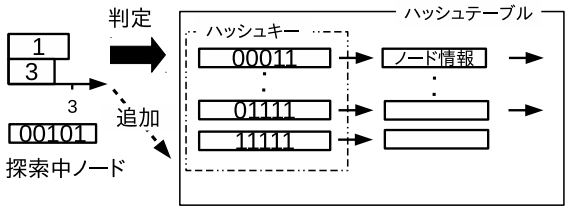


図 3: ハッシュテーブルを用いた枝刈りの例

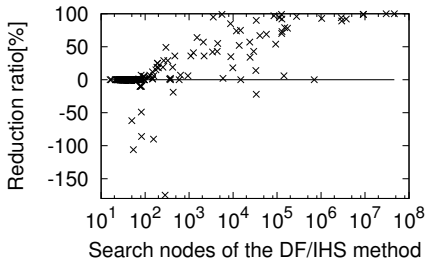


図 4: 提案手法の削減率

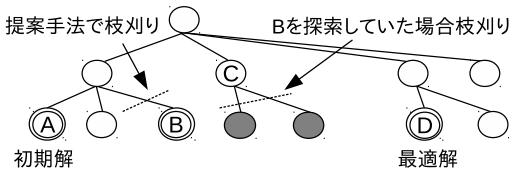


図 5: 精度の高い暫定解を枝刈りする例

6 おわりに

本研究では，DF/IHS 法の探索ノード数を削減するために，ハッシュテーブルを用いて枝刈りする手法を提案した．評価の結果，ハッシュテーブルを用いた削減手法は，DF/IHS 法に対し探索ノード数を最大約 99.8%，平均約 6.5%削減することが確認できた．また，ハッシュテーブルを用いた手法は，DF/IHS 法で探索ノード数が多い問題ほど高い削減率が得られることが確認できた．

参考文献

[1] Kasahara, H. and Narita, S.:Practical Multiprocessor Scheduling Algorithms for Efficient Parallel Processing, IEEE Trans. Comput., Vol.33, No.11, pp.1023-1029(1984).

[2] 中村あすか，前川 仁孝:タスクスケジューリング問題の厳密解における探索ノード数削減アルゴリズム，情報処理学会論文誌，Vol.7，No.1，pp.1-9(2014).

[3] Standard Task GraphSet, available from (http://www.kasahara.elec.waseda.ac.jp/schedule/) (accessed 2016-01-06).

Copyright ©2016 Information Processing Society of Japan.
All Rights Reserved.