B-025

評価関数の精度による階層的挟み撃ち探索の性能評価
Effect of Hierarchical Pincers Attack Search by Accuracy of Evaluation Functions

中村 あすか† 富永 浩文† 前川 仁孝†
Asuka Nakamura Hirobumi Tominaga Yoshitaka Maekawa

1. はじめに

多くの組合せ最適化問題の最適解を求める有効な手法のひとつとして，分枝限定法が知られている [1]．分枝限定法は，問題の規模が大きくなると求解に時間がかかる．分枝限定法の求解時間を削減するために，複数のプロセッサで並列に探索する並列分枝限定法が研究されている [2][3]．多くの並列分枝限定法は，評価の良いノードから順にプロセッサに割り当てる [2][3]．このため，評価関数の精度が悪いほど，並列化における高速化率が高くなる．組合せ最適化問題のひとつである実行時間最小マルチプロセッサスケジューリング問題では，階層的挟み撃ち探索による並列分枝限定法が有効であることが示されている [4]．分枝限定法は，問題ごとに固有の評価関数を与える．階層的挟み撃ち探索が組合せ最適化問題全般に対して有効であることを示すためには，本手法が評価値の精度に関係なく高速に求解可能であることを示す必要がある．そこで，本稿では，組合せ最適化問題の問題全般に対する階層的挟み撃ち探索の有効性を示すために，分枝限定法の評価関数の精度が求解時間に与える影響を評価する．評価では，評価関数の精度を明確にするために擬似的に探索木を生成し，階層的挟み撃ち探索による求解過程をシミュレーションし，評価を行う．

2. 階層的挟み撃ち探索

分枝限定法における階層的挟み撃ち探索は，共有メモリ環境上において，探索木上の最適解の位置に関係なく高速に求解を行うために，実行時間最小マルチプロセッサスケジューリング問題の求解において提案された並列探索手法である [4]．図 1 に，階層的挟み撃ち探索における各プロセッサの振舞を示す．本手法は，評価の良いノードが探索木の左側に来るようにソートされた探索木を探索する．リーダプロセッサは，評価の良いノードの多い探索木左側から深さ優先探索を行い，待ち状態のスレーブプロセッサに探索経路上のノードを割り当てる．スレーブプロセッサは，割り当てられたノードを根とする探索部分木に対し，右側から深さ優先探索を行う．左右から探索するプロセッサの探索領域が重複すると，スレーブプロセッサは，割り当てられた領域の探索を終了し，リーダプロセッサに新しい探索領域の割当を要求する．リーダプロセッサは，スレーブプロセッサに広い探索領域を割り当てるため，再割当の回数を抑えて動的に負荷分散を行うことができる．

階層的挟み撃ち探索では，マスタープロセッサが探索木左側から探索し，スレーブプロセッサが探索木右側から探索する．このため，探索木における最適解の位置に関係なく高速に求解することができる．

†千葉工業大学情報工学科
Department of Computer Science, Chiba Institute of Technology

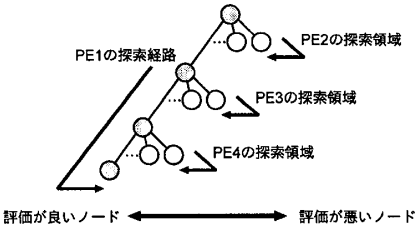

図 1: 4 プロセッサによる階層的挟み撃ち探索の振舞

3. 階層的挟み撃ち探索と高速化率

分枝限定法は，限定操作の効率を上げることことで，高い高速化率を得ることができる．このため，精度の良い評価関数を用いたり，なるべく早い段階で最適解を探索する必要がある．多くの並列分枝限定法では，なるべく早い段階で最適解を探索するために，評価の良いノードから順にプロセッサに割り当てる [3]．並列分枝限定法は，多くのプロセッサが同時に探索を行うので，評価関数の精度が良い問題では逐次同様に早い段階で最適解を探索することができる．また，評価関数の精度が悪い問題では，逐次探索よりも早い段階で最適解を探索するので，高い高速化率が得られることが知られている [3]．

階層的挟み撃ち探索は，左右から複数のプロセッサが挟み撃つように探索を行うので，最適解を探索するタイミングは，評価関数の精度から影響を受けにくいと考えられる．また，評価関数の精度が悪い問題では，限定操作の効率が落ちる．このため，階層的挟み撃ち探索は，評価関数の精度が悪い問題の求解において高い高速化率を得られにくいと考えられる．

4. 求解対象とする問題の生成方法

本稿では，評価関数の精度を明確にするために，目的関数が n 個の変数からなるベクトル X から定まる組合せ最適化問題を想定して，擬似的に探索木を生成する．探索木は，今井らの実験で用いた手法 [2][5] を基に，4.1 節のように探索木の形状を決定し，4.2 節のように各ノードの最適解と下界値を定める．

4.1 探索木の形状

今井らの実験で生成した探索木は二分木であるが，より一般的な探索木を想定するために，本稿では各ノードにおける分枝数を 2 に限定しない．あるノード P_i の子ノードを $P_{i0}, P_{i1}, P_{i2} \cdots P_{i(n-i-1)}$ とおく．分枝操作では X の要素を 1 つ固定することで分枝を行うと仮定し，ノード P_i の分枝数を $(n-i)$ とする．このため，生成する探索木の最大深さは，$(n-1)$ となる．ここで，生成する探索木の深さ i のノードを P_i とおき，根ノードを P_0 とおく．また，ノード P_i の深さを $depth(P_i)$ とおく．

4.2 各ノードの最適解と下界値

探索木の下界値の精度を明かにするために，求解対象とする問題の最適解を 1.0 とする．ノード P_i の部分問題における最適解を $cost(P_i)$ とおき，各ノードの部分問題における最適解を式 (1) で定める．

$$\begin{cases} cost(P_0) &= 1.0 \\ cost(P_{i0}) &= cost(P_i) \\ cost(P_{ij}) &= cost(P_i) + \alpha \quad (j \neq 0) \end{cases} \tag{1}$$

ただし，α は，パラメータ λ による式 (2) の確率密度関数 $f(\alpha)$ で与えられる確率変数である．

$$f(\alpha) = \begin{cases} \lambda e^{-\alpha\lambda} & (\alpha \geq 0) \\ 0 & (\alpha < 0) \end{cases} \tag{2}$$

ノード P_i の下界値を $lb(P_i)$ とおき，式 (3) のように定める．

$$\begin{cases} lb(P_0) &= 0.0 \\ lb(P_{ij}) &= cost(P_{ij}) \quad (i = n-1) \\ lb(P_{ij}) &= \beta \quad\quad\quad (i \neq n-1) \end{cases} \tag{3}$$

ただし，β は，区間 $[\, lb(P_i), lb(P_i) + 2\Delta_{ij} \,]$ 内の一様乱数の実数値とし，Δ_{ij} は，式 (4) で定める．

$$\Delta_{ij} = \frac{cost(P_{ij}) - lb(P_i)}{n - depth(P_i)} \tag{4}$$

本手法を用いると，λ の値が大きいほど，$cost(P_i)$ と $cost(P_{i0})$ の差 α が小さくなり，多くのノードが $lb(P_i) \leq cost(P_0)$ を満たす探索木が生成できる．また，λ の値が大きいほど，下界 $lb(P_i)$ の精度が良くなる．

5. 評価

評価関数の精度が階層的挟み撃ち探索に与える影響を明かにするために，上記の条件で生成した探索木の求解過程をシミュレーションし，階層的挟み撃ち探索と評価関数の精度について評価する．求解過程のシミュレーションでは，各ノードにおける計算にかかる時間が一定であり，再割当に必要な処理にかかる時間は無視できると仮定し，4 章の方法で生成した探索木に対して求解時間を測定する．図 2 に，$n = 15, \lambda = 0.1$ および $n = 15, \lambda = 0.5$ における，階層的挟み撃ち探索のプロセッサ数ごとの求解時間を示す．図 2 より，$n = 15$ における階層的挟み撃ち探索の求解時間に対する λ の値の影響は少ない．このため，評価関数の精度が階層的挟み撃ち探索の求解時間に与える影響は少ないといえる．また，$\lambda = 0.1$ の求解では，プロセッサ数 3 による求解において最も求解時間が短くなった．この理由として，最適解を早い段階に探索できたことが挙げられる．$\lambda = 0.1$ における最適解発見までにかかった時間の平均は 695997[*u.t.*] であり，プロセッサ数 3 による求解において最適解を発見するまでにかかった時間の平均は 118454[*u.t.*] であった．プロセッサ数 3 による求解では，早い段階で最適解が発見されていたため，分枝限定法における限定操作の効率が上がり，高速に求解できたと考えられる．

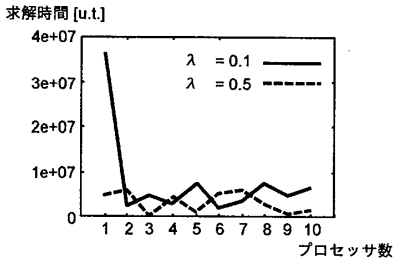

図 2: $n = 15$ の問題の階層的挟み撃ち探索の求解時間

これらのことから，階層的挟み撃ち探索の求解時間は，評価関数の精度よりも最適解を発見するタイミングに影響を受けていると考えられる．探索木上でノードの位置情報を示す指標として，セレクションポインタ (SP) 値が使われる [4]．SP 値は，(深さ，兄弟番号) によって，ノードの位置情報を表す．$\lambda = 0.1$，プロセッサ数 3 の条件で求解を行った問題において，最適解が存在するノードの SP 値は $\{(0,11),(1,9),(2,11),(3,10),\cdots\}$ であったので，最適解を持つノードは探索木の右側に存在することが分かる．このため，この問題に対して評価関数の良いノードだけを優先して探索する手法を用いる場合，求解時間が長くなる可能性がある．

6. おわりに

本稿では，評価関数の精度の違いによる階層的挟み撃ち探索の振る舞いについて評価を行った．評価の結果，階層的挟み撃ち探索が評価関数の精度の悪い問題においても高速に求解可能であることを確認できた．また，本稿の評価により，階層的挟み撃ち探索は，実行時間最小マルチプロセッサスケジューリング問題以外の組合せ最適化問題に対する求解にも有効であること，および，評価関数の精度よりも，最適解を発見するまでにかかる時間が求解時間に与える影響の方が大きいことを確認できた．

参考文献

[1] 茨木俊秀：組合せ最適化 –分枝限定法を中心として–，産業図書 (1983).

[2] 今井正治，吉田雄二，福村晃夫：分枝限定アルゴリズムの並列化とその評価，電子情報通信学会論文誌 D， Vol. J62-D, No. 6, pp. 403–410 (1979).

[3] Li, G. J. and Wah, B. W.: Coping with anomalies in parallel branch-and-bound algorithms, *IEEE TRANS.Comput.*, Vol. C-35, No. 6, pp. 568–573 (1986).

[4] 笠原博徳，伊藤敦，田中久充，伊藤敬介：実行時間最小マルチプロセッサスケジューリング問題に対する並列最適化アルゴリズム，電子情報通信学会論文誌 D，Vol. J74-D1, No. 11, pp. 755–764 (1991).

[5] 今野浩史，鈴木久敏：整数計画法と組合せ最適化，日科技連出版社 (1982).