階層型山登り法：制約充足問題の高速な解法

3N－3

湯上　伸弘　　太田　唯子　　原　裕貴

富士通研究所

1　はじめに

Minton等が提案した最小矛盾戦略に基づく山登り法（MCHC, Min-Conflicts Hill Climbing）[1] は、制約充足問題に対する非常に有効な手法のひとつである。MCHCは、ある種の制約充足問題を、問題の規模が大きい場合でも、高速に解くことができる。MCHCでは、制約充足問題を制約違反の数を最小化する最適化問題に置き換え、その最適化問題を山登り法を用いて解く。しかし、山登り法による最小化では、変換された最適化問題が多数の局所最適解を持つような場合には、制約違反数が0であるような最適解に到達できる確率は非常に小さい。本稿では、MCHCの持つこの問題点を克服するために、局所最適解から脱出するための手続きEFLOPと、EFLOPを用いて効率的に制約充足解を探索するための手法である階層型山登り法（HHC）を提案し、その効果を検証する。

2　階層型山登り法

図1は、階層型山登り法（HHC）の動作を概念的に表した図である。階層型山登り法では、2つのレベルで山登り法を行うことにより制約違反数の最小化を行う。上位レベルにおける山登り法は、各局所最適解間を遷移するようなオペレータを用いて行われる。下位レベルの山登り法は、このオペレータを実現するために用いられる。このオペレータは、後に説明する手続きEFLOPを用いて局所最適解から脱出し、MCHCを用いて、EFLOPから出力された割り当ての近くの局所最適解へと到達することにより実現される。この結果得られた局所最適解における制約違反の数が、以前の局所最適解における制約違反数よりも少ないか等しいならば、新しい局所最適解を受理する。そうでない場合には元の局所最適解に再びEFLOPとMCHCを適用して別の局所最適解を生成する。これを、制約違反数が0になるまで、すなわち制約充足解が得られるまで続行することにより、制約充足問題を解く。

階層型山登り法における最大のポイントは、いかにして局所最適解から脱出するかである。本稿では、そのための手続きEFLOP（Escaping From Local Optima by Propagation）を提案する。図2にEFLOPのアルゴリズムを示す。EFLOPは、まずはじめに満足されていない制約に含まれる変数を選択し、その値を変更する。その結果別の制約が満足されなくなったなら、その制約に含まれる別の変数の値を変更することにより、その制約を満足させる。この際に、変数の新しい値としては、それまでにEFLOP内で変更された変数の変更後の値と矛盾しないような値を選択する。この条件を満足する、変数とその値の組が複数存在する場合には、値の変更によって新たに生じる制約違反の数を最小とするような変数と値を選択し、値の変更を行う。このようにEFLOPでは、制約を通して変数値の変更を伝播させることにより、相互に依存している複数の変数の値を、それらの間の無矛盾性を保ったまま同時に変更する。

3　評価

ランダムに生成された制約充足問題（制約は全て

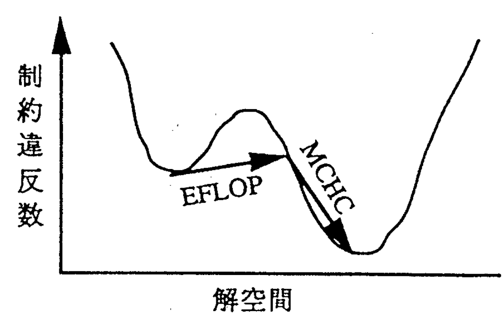

図1　階層型山登り法

HHC: An Efficient Method for Constraint Satisfaction
Nobuhiro Yugami, Yuiko Ohta, Hirotaka Hara
FUJITSU LABORATORIES LTD.

```
procedure EFLOP
  select a variable v  in conflict randomly;
  change v 's value randomly;
  V := {v};
  while possible
    select a constraint c  that satisfies
    following conditions;
      (c1) c  is satisfied before EFLOP is
           called;
      (c2) c  is violated now;
      (c3) there is a variable v  and its
           value a such that;
        (c3-1) v  is not an element of V;
        (c3-2) v=a  makes c  satisfied;
        (c3-3) v=a  is consistent with new
               values of variables in V;
    change v 's value to a ;
    add v  to V;
  end_of_while;
```

図2　EFLOPのアルゴリズム

binary）を用いて、HHCの評価を行った。比較対象はMCHCである。HHC, MCHCともに、たとえ解が存在する場合でも局所最適解に捕まり解を発見できない可能性があるので、ある程度探索を行っても解を発見できない場合には、別の初期解を生成し、そこから探索を開始するようにした。初期解の生成は全てランダムに生成した。図3は、２０変数（変数の取り得る値は１０個）４０制約の問題にいて、制約の強さ（制約に含まれる２つの変数の値の組のなかで、制約を満足しない組の割合）を変化させたときに、生成された問題が制約充足解をもつ確率、および解が存在する問題をHHCおよびMCHCで解くのに要した時間（１００個の充足可能な問題の平均）である。図から分かるようにHHCは、全ての制約の強さについてMCHCよりも短時間で問題を解くことができた。求解時間は、MCHC, HHCともに、制約の強さが、生成された問題が充足可能である確率が0.5になるような値（0.64）の時に最大となっている。これは、このような場合に制約充足解の数に対して局所最適解の数が極めて多くなるためである。HHCとMCHCとの差も、制約の強さがこの値の時に最大になっている。すなわち、HHCは問題が難しい場合ほど有効であることが分かる。図4は、このようなSにおける平均求時間を、変数の数が、２０、４０、６０、８０の場合について調べた結果である。制約の数は、全ての場合について変数の数の２倍とした。図4から分かるようにHHCは全ての場合についてMCHCよりも短時間で解を求めることができ、かつ、問題の規模が大きいほど、HHCとMCHCとの差も大きく、８０変数の問題に対しては、HHCはMCHCよりも５０倍以上高速であった。

参考文献

[1] S. Minton et al., Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems, Artificial Intelligence, Vol.58 (1992)

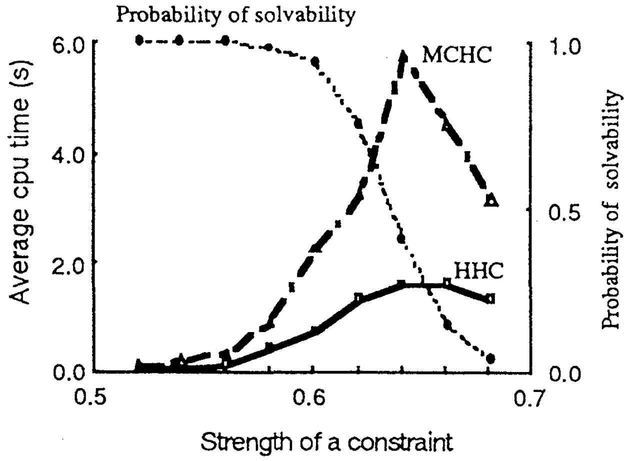

図3　２０変数４０制約問題

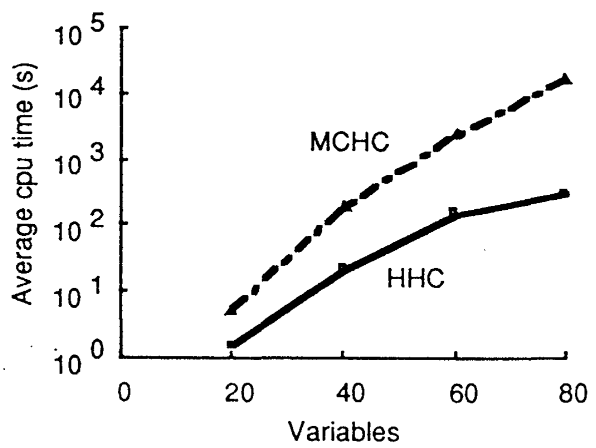

図4　HHCとMCHCの比較