

分枝限定法

巡回セールスマンやナップサック問題のような組み合わせ最適化問題を解くことを考える。解の組み合わせは有限なので、すべての組み合わせを列挙して評価すれば最適解が得られるが、組み合わせの数は膨大であり効率が悪い。分枝限定法はこのような解の列挙による最適解の探索において、解の集合の階層的な分割を行って、無駄な部分集合の探索を省略していく方法である。

具体的な手順は以下のようなものである。例として、解に対する評価値の最小化問題を考える。まず、根として、解全体の集合を割り当てる。次に、解全体を部分集合に分け、それぞれを子とする。さらに、その子を再帰的に部分集合に分けて、階層構造を構成する（分枝操作）。葉が個別の解に相当するので、すべての葉を評価して最小となるものを見つければ最適解が得られるが、効率が悪い。そこで、各中間ノード（解の集合）に対して、評価値の上限値と下限値を何らかの方法で計算する。ここで、あるノードAの下限値が、他のノードBの上限値より高い場合、ノードAの解集合の中には最適解は存在しないことが保証されるので、ノードAはこれ以上探索する必要がなくなる（限定操作）。分枝限定法の利用においては、分割の方法や評価値の計算方法を問題に応じて適切に設定する必要がある。

巡回セールスマン問題における分枝限定法の利用例を説明する。巡回セールスマン問題では、コスト付きの無向グラフにおいて、すべての頂点を一度ずつ訪問して元に戻ってくる巡回路（辺の集合）を求める。分枝操作においては、特定の辺を巡回路に含むか含まないか、で解集合の分割を行う。評価値の例として、使える辺のうちからコストの小さい辺を順に選んだものを下限値として計算する。上限値は中間ノードでは計算せず、葉に達したとき（巡回路が確定したとき）に確定値として計算する。そうしておくことで、解の探索の過程において、中間ノードの下限値がすでに計算済みの葉のコスト（確定値）より大きい場合には、その中間ノードをそれ以上探索しなくてよいことがわかり、探索の手間を大きく削減できる。

以下の例で説明する。まず、根(制約なし)の下限値は、コストの低い辺として ab,ce,ac,de,cd を選んで 14 となる。根の左の子は ab を使う場合として下限値は変わらず 14、根の右の子は ab を使わない場合として下限値は ce,ac,de,cd,bd を選んで 18 となる。同様にして探索を続けると、ab,ac,bd を含むとした時点で巡回路が一意に決まり(ab,bd,de,ec,ca)そのコストは 15 である。この情報を使うことで、下限値が 15 より大きい中間ノードを刈ることができ、探索の手間を大きく削減することができる。

