

JAVA を用いたメタヒューリスティクス汎用クラスの開発

01404650 法政大学工学部 西岡靖之 NISHIOKA Yasuyuki
 法政大学工学部 * 久保琢磨 KUBO Takuma

1 はじめに

組合せ最適化問題の多くは NP 困難である。このような問題に対し、多大な時間をかけて最適解を求めることより、現実的な時間である程度精度の高い解を求めるメタヒューリスティクスが注目されている。しかし、実際にメタヒューリスティクスを利用して問題を解く汎用ソルバーは存在せず、個々にプログラミングするには高度なプログラミング技術を要するため、現実の問題で手軽に使われるまでには至っていない。

そこで本研究では JAVA を用いてメタヒューリスティクス汎用クラスの開発を行い、それによって様々な組合せ最適化問題に対し、メタヒューリスティクスを容易に実装できるような環境を提供することを目的とする。

2 メタヒューリスティクス汎用クラス

本研究では、メタヒューリスティクスがもつ探索の共通的な構造をクラス化した後に、焼きなまし法 (Simulated Annealing)、遺伝的アルゴリズム (Genetic Algorithm)、タブー探索 (Tabu Search) をとりあげ、個々の特徴を表現するクラスの設計をする。ユーザーは個別の問題に対して、まず探索手法を選択した後に、その探索クラスのサブクラスとして独自の要素を追加的にプログラミングすることになる。それぞれのクラスの属性やメソッドの内容、およびクラス間の関係を図 1 に示す。

2.1 Search クラス

メタヒューリスティクスの枠組の共通部分を取り出したクラスが Search クラスである。また、解を表すクラスとして Solution クラスも存在する。探索の実行を行うメソッド exec のソースコード

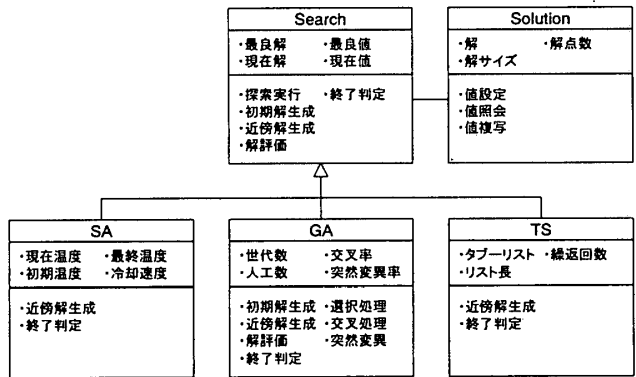


図 1. 主なクラスの関係図

リスト① 探索の共通部分

```

protected void exec()
{
    current.copy(init()); //初期解生成

    while(!end()){ //終了判定
        neighbor.copy(next(current)); //近傍解を求める

        if(check(neighbor)){ //近傍解の制約チェック
            neighborEval = eval(neighbor); //近傍解の評価
            if(bestEval > neighborEval){ //解を評価する
                bestEval=neighborEval; //最良値を更新
                best.copy(neighbor); //最良解を更新
            }
        }

        current.copy(neighbor); //現在解を更新
        currentEval = neighborEval; //現在解の評価を設定
    }
}
    
```

をリスト①に示す。

2.2 メタヒューリスティクス用サブクラス

焼きなまし法、遺伝的アルゴリズム、タブー探索のためのクラスは、Search クラスのサブクラスとなる。これらのクラスは、Search クラスの属性やメソッドを継承し、必要に応じて修正または追加することにより、個々の探索の特徴を表現して

いる。以下のリスト②に、焼きなまし法の近傍解の生成処理をサンプルとして示す。

②. 焼きなまし法の近傍解の生成処理

```
protected Solution next(Solution cur)
{
    for(int r=1;r<=loop;r++) {
        neighbor.copy(cur);
        neighbor.copy(generate(cur)); //近傍解の生成
        neighborEval=eval(neighbor); //近傍解評価の設定

        if(neighborEval<=currentEval || //解を受受するか判定
            Math.exp((currentEval-neighborEval)/temperature) >
            Math.random())
        {
            current.copy(neighbor);
            currentEval=neighborEval;
        }
    }
    temperature=temperature*rate; //温度を下げる処理
    return cur;
}
```

同様に、図1で示したような個々のメタヒューリスティクス用の属性やメソッドを定義している。たとえば、遺伝的アルゴリズム用クラスでは探索処理において選択や交叉、突然変異などが定義されており、タブー探索用クラスには、タブーリストや近傍解生成などが定義されている。

2. 3 Problem クラス

このクラスは、個々の問題に応じて、ユーザーが独自にプログラムするためのクラスである。ただし、対象とする問題に対して、メタヒューリスティクスのアルゴリズムを最初から記述しなくてもよいように、前節の各サブクラスを継承し、必要な部分のみ追加するだけでプログラムが完成できるように工夫されている。

3 実行例

本研究で開発したメタヒューリスティクス汎用クラスを利用して作成したプログラムと実行例を示す。ここでは、TSP を焼きなまし法とタブー探索を用いて解いた。図2はその実行結果である。ここで注目すべき点は、TSP 用に独自に作成した Problem クラスは、その属性およびメソッドをまったく変更することなく、継承クラスの指定を変

えるだけで、探索のためのメタヒューリスティクスを切り替えることができたという点である。TSP のための探索部分のプログラムのサイズは39行であった。

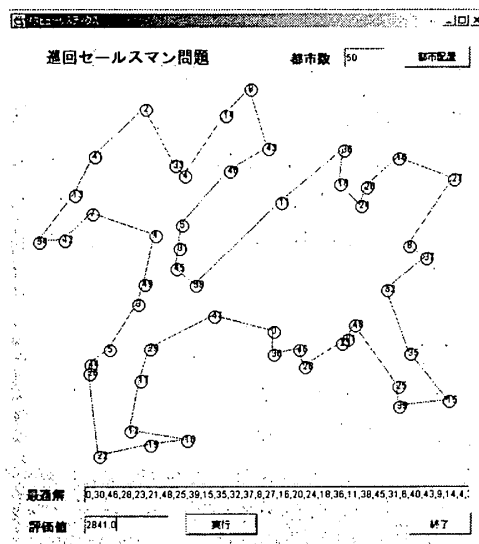


図2. TSP 実行結果画面

4 おわりに

様々な組合せ問題に対し、比較的簡単なプログラミングによって容易にメタヒューリスティクスを適用することができるメタヒューリスティクス汎用クラスの開発を行った。また、簡単なサンプル問題を用いて実際にSAを用いたTSPの実装が、数十行程度で実装できることを示した。ここで紹介したプログラムはインターネットから自由にダウンロードして利用することができるので、ぜひともご利用いただき、忌憚のないご意見を賜れば幸いである。

参考文献

- [1] 柳浦睦憲/茨木俊秀: 組合せ最適化 - メタ戦略を中心として, 朝倉書店, 2001.
- [2] 柳浦睦憲/茨木俊秀: 『組合せ最適化問題に対するメタ戦略について』 電子情報通信学会論文誌 D-I Vol.J83-D-I No.1(2000年1月)p3-25
- [3] 生産スケジューリングライブラリー, <http://www.img.k.hosei.ac.jp/pslib/>(2003)