1995年度日本オペレーションズ・リサーチ学会
春季研究発表会

メタ戦略とその周辺

01108010　東京商船大学　久保 幹雄　KUBO Mikio

1　はじめに

メタ戦略（メタヒューリスティック，メタ解法，モダンヒューリスティック）とは，組合せ最適化問題を解くためのヒューリスティックを有機的に結合させたものであり，従来の近似解法を超えたパラダイムとして注目を浴びている．

ヒューリスティックという言葉は，しばしば得られる解の精度に理論的な保証のない近似解法という意味で使われる．したがって，メタ戦略は，理論的な保証のないものの寄せ集めであるかのような（すなわちメタメタな）印象を受けるが，ここでは少し違った視点からメタ戦略を捉えてみる．

ここで用いる「メタ」は，Donald Knuth のメタフォント（METAFONT）の精神と似たものである．メタフォントとは「あらゆる文字の形を統一的に扱う」ためのツールであり，幾つかのパラメータを制御することによって，（もちろん全てではないが）様々なフォントを作ることができる．同様に，メタ戦略とは，ヒューリスティックにパラメータを追加し，そこで生まれた自由度を用いて問題を巧く解くテクニックであると考える．

幾つかのメタ戦略は，自然界にアナロジーを持つことを特色とするが，それはメタ戦略と呼ばれるための必要条件ではない．ここで取り上げるメタ戦略としては，Simulated Annealing 法 [4, 6, 18, 22], Tabu Search [9, 10, 11], Genetic Algorithm [12, 15, 21], Quicker [7, 8] および筆者が提案する Life Span Method, Generic Local Search である．Simulated Annealing 法は焼き鈍し現象，Genetic Algorithm は遺伝学，Tabu Search は人間の記憶過程にアナロジーを持つ．Life Span Method は実験的解析のために設計された Tabu Search の変形であり，Generic Local Search は，多くのメタ戦略を含む一般的なフレームワークである．

2　諸定義

以下の組合せ最適化問題を考える．ある空でない有限集合$\mathcal{U}$, $\mathcal{U}$から整数（実数）への写像$c: \mathcal{U} \to Z(R)$, $\mathcal{U}$の部分集合の集まり$\mathcal{F} \subseteq 2^{\mathcal{U}}$が与えられたとき

$$\min\{f(x) = \sum_{u \in x} c(u) : x \in \mathcal{F}\} \tag{1}$$

を与える$x \in \mathcal{F}$を求める．

式 (1) における関数$f(x) = \sum_{u \in x} c(u)$を目的関数または費用関数と呼ぶ．目的関数を最小にする解$x \in \mathcal{F}$を（大域的）最適解と呼び，その集合を$\mathcal{F}^*$と記す．すなわち，$\mathcal{F}^* = \{x \in \mathcal{F} : f(x) \leq f(y), \forall y \in \mathcal{F}\}$である．

実行可能解の集合$\mathcal{F}$を与えたとき，近傍Nは以下の写像と定義される．

$$N : \mathcal{F} \to 2^{\mathcal{F}}$$

すなわち，実行可能解の集合から，そのべき集合（部分集合の集合）への写像が近傍である．

実行可能解$x \in \mathcal{F}$で，$f(x) \leq f(y), \forall y \in N(x)$を満たすものを（近傍$N$に対する）局所的最適解と呼ぶ．本稿で取り上げる多くのメタ戦略は，近傍を基礎として構築される．

3　Local Search

次の関数$improve(x)$を用いることによって，Local Search を擬似コードで記述する．

$$improve(x) = \begin{cases} \text{any } x' \in N(x) & \text{with } f(x') < f(x) \\ & \text{if such an } x' \text{ exists} \\ \emptyset & \text{otherwise} \end{cases}$$

procedure local search
1 $x :=$ some initial feasible solution
2 **while** $improve(x) \neq \emptyset$ **do**
3 　$x := improve(x)$
4 **return** x

4　Simulated Annealing 法

Local Search を拡張したメタ戦略として Simulated Annealing 法がある．この解法は Kirkpatrick ら [18], および Černý [5] が独立に提案したものであると言われているが，その土台は，1953年に Metropolis ら [20] によって築かれていた．

Local Search の場合と同じように近傍Nが与えられているものとする．イテレーションkのときの温度を表すパラメータT_kは，非負の値をとる非増加な数列であり，kが無限大になるにしたがって 0 に収束するものとする．すなわち

$$T_1 \geq T_2 \geq \cdots \geq T_{k-1} \geq T_k \geq T_{k+1} \geq \cdots \geq T_\infty = 0$$

を満たすものとする．

procedure simulated annealing
1 $x_1 :=$ some initial feasible solution
2 　**for** $k = 1$ **to** ∞ **do**
3 　　choose a $y \in N(x_k)$ randomly
4 　　$x_{k+1} := y$ w.p. $\exp\{-[f(y) - f(x_k)]^+/T_k\}$
5 　　otherwise, $x_{k+1} := x_k$

上のアルゴリズムで$[a]^+$は$\max\{a, 0\}$を表す．

Simulated Annealing 法の漸近的最適性の必要十分条件については Hajek [13] を参照されたい．そこでは，温度Tを$T_k = d^*/\log(k+1)$（d^*は局所最適解の深さの最大値）とする方法を用いている，このような温度パラメータの制御法は対数冷却 (logarithmic cooling) と呼ばれる．この冷却法は，理論的には優れているが，実用上の価値は（現在のところ）疑問視されている．実用的な実装および実験的解析については Johnson ら [16, 17] を参照されたい．そこでは，幾何学的冷却 (geometric cooling) と呼ばれる方法を推奨している．

5 Tabu Search

ここでは，Glover [9, 10, 11] によって最近提案された Tabu Search について概説する．Glover とは独立に Hansen によってほぼ同じ解法が提案されている [14]. Hansen は彼の解法を Steepest Ascent Mildest Descent 法と呼んでいる．個人的には，この名前の方が Tabu Search よりもアルゴリズムの内容を良く表していると思うが，ここでは慣用にしたがい Tabu Search と呼ぶことにする．

Tabu Search では，近傍から禁断リスト Ξ を除いた中から，最も良い解へ移動する．移動を行うための関数 $move(x)$ を以下のように定義する．

$$move(x) = \begin{cases} x' & \text{if } f(x') \le f(y), \forall y \in N(x) \setminus \Xi \\ \emptyset & \text{if } N(x) \setminus \Xi = \emptyset \end{cases}$$

この関数を用いることによって Tabu Search の概要は以下のように書くことができる．なお，TL (Tabu Length を表す) は禁断リスト Ξ の長さである．

```
procedure tabu search
  t := 0
  x_0 := some initial solution
  Ξ := ∅
  TL := a positive integer
  while stopping-criterion ≠ yes do
      x_{t+1} := move(x_t)
      Ξ := Ξ ∪ {x_t} \ {x_{t-TL}}
      t := t + 1
  return x
```

上のアルゴリズムでは禁断リストに解自身を保持するように書いたが，実際の実装においては，解の変化を表す何らかの印 (属性: attribute) を保存する．より正確に言うと，属性集合 (attribute set) A に対する写像 $\psi : X \times X \to A$ によって，禁断リスト Ξ に保存する「何か」を決める．さらに，Glover の提案する Tabu Search は “The philosophy of tabu search is to collect principles of intelligent problem solving” の言葉通り，たくさんの ad hoc なルールを探索に付加する傾向がある．Tabu Search における種々の工夫については，Glover らによるサーベイ [9, 10, 11] を参照されたい．

6 Life Span Method

筆者らは，当初 Johnson らの Simulated Annealing 法に対する比較実験を模範として，Tabu Search に対しても同様の実験的解析を行うことを目的としていた．しかし，Glover の言う種々のルールの導入は，正当な比較を困難にしているという理由から，最小限のパラメータ (およびルール) を用いた Tabu Search の変形を構築した．ここでは，この Tabu Search の変形 (Life Span Method) について述べる．

まず，属性への写像 $\psi(x, x')$ は，台集合 $\mathcal{U}$ 上での解の対称差 $x \triangle x' = (x' \setminus x) \cup (x \setminus x')$ とすることによって Tabu Search の属性の定義に対する曖昧さを解消する．このとき，禁断リストは Glover が推奨するキュー構造でなく，台集合に対応する配列 LS (LS は，Life Span (寿命) を意味する) に移動禁止の残りイテレーション数を保持する．Life Span Method の呼び名は，Life Span によって禁断リスト (短期メモリ) を管理することから名付けられた．

Tabu Search に対して提案されている種々の工夫は，大きく分けると多様化 (diversification) と集中化 (intensification) を目的としている．多様化は一度探索した解を探索しにくくすることであり，集中化は良好な解の近所を集中的に探索することである．この二つの概念は，良いメタ戦略設計のための鍵である．

Life Span Method においては，多様化は長期メモリという Tabu Search の基本的な道具を用いて達成される．禁断リストと同様に，台集合上に配列 LTM(Long Term Memory) を定義する．台集合の要素が，解の移動の対称差に含まれるならば LTM を 1 だけ増やす．すなわち，LTM には，台集合の要素が移動に含まれた回数を保持する．長期メモリは，目的関数 $f(x)$ にペナルティ項として加えられることによって，探索の多様化をもたらす．ペナルティ項の重みを α とすると，長期メモリを導入した目的関数は，解 x から x' への移動のときには $f(x') + \alpha \sum_{u \in x \triangle x'} LTM(u)$ と定義される．

また，集中化は再出発 (re-start) によって達成される．再出発は，決められたイテレーションの間，それまでに見つかった最良の解を改善できないときに行われ，禁断リストを空にし (すなわち $LS(u) = 0, \forall u \in \mathcal{U}$ とし)，現在までに発見された最良解に戻って再び探索を行う方法である．(問題によっては，探索を多様化させるため最良解に戻らないこともある．) 終了判定も再出発と同様に，最良解の更新が，決められたイテレーションの間に起きないときに行われる．

Life Span Method においては，移動を制御するための関数 $move(x)$ を以下のように定義する．

$$move(x) = \arg\min\{f(x') + \alpha \sum_{u \in x \triangle x'} LTM(u) : x' \in N(x), LS(u) = 0, \forall u \in x \triangle x'\}$$

すなわち，禁止されていない ($LS(u) = 0$ である) u だけを使って移れる解の中で (ペナルティ付きの) 目的関数値を最小にするものが $move(x)$ である．

```
procedure life span method
  x := some initial solution
  f* := an upper bound on the optimal value
  LS(u) := 0, ∀u ∈ U (Uは台集合)
  LTM(u) := 0, ∀u ∈ U
  while stopping-criterion ≠ yes do
      x' := move(x)
      LS(u) := TL, ∀u ∈ x△x'
      LTM(u) := LTM(u) + 1, ∀u ∈ x△x'
      x := x'
      if f(x) < f* then f* := f(x), x* := x
      LS(u) := LS(u) − 1, ∀u ∈ U, LS(u) > 0
      if re-starting-criterion = yes then
          x := x*, f := f*
          LS(u) := 0, ∀u ∈ U
  return x
```

上のアルゴリズムで，TL は Tabu Search の場合と同じく Tabu Length (禁止するイテレーション数を表すパラメータ) を表す．Life Span Method においては TL に乱数を用いることによって，簡単に探索を多様化することができる．現在までに Life Span Method は，グラフ分割問題，最大安定集合問題 (最大クリーク

問題)，グラフ彩色問題, 二次割当問題, ジョブショップスケジューリング問題および n-Queen 問題に適用され，ある程度の成果をあげている．ちなみに Neural Net 法では n が数百から数千程度の n-Queen 問題を対象にしているが，Life Span Method をベースにした解法では 2 億 Queen を解くことができる．Life Span Method の詳細については [19] を参照されたい．

7 Genetic Algorithm

Genetic Algorithm は複数の実行可能解を保持し，その解を進化のアナロジーを用いて改善していく方法の総称である．古典的な Genetic Algorithm [12] では，解を 0,1 のビット列として表していたが，最近では問題の構造に合わせた任意の方法で解を表現する Genetic Algorithm の変形 (Evolution Algorithm または Evolution Program と呼ばれる) も多く用いられている [21].

いま，第 t イテレーションの実行可能解の集合を $P(t) = \{x_1^t, \cdots, x_p^t\}$ とする．Genetic Algorithm 特有の用語を用いると，$P(t)$ は第 t 世代の集団 (population)，各 x_iは個体と呼ばれる．基本的な Genetic Algorithm では交叉 (crossover)，突然変異 (mutation) と呼ばれる二つの操作を用いて新たな世代の候補 (子孫) を生成し (下のアルゴリズムの関数 *generate*)，自然淘汰 (natural selection) と呼ばれる操作で，次の世代に生き残るものを選択する (下のアルゴリズムの関数 *select*).

procedure genetic algorithm
1 $t := 0$
2 $P(0) :=$ a set of initial feasible solutions
3 **while** stopping-criterion $\neq$ yes **do**
4 $\quad P'(t+1) := generate(P(t))$
5 $\quad P(t+1) := select(P'(t+1))$
6 $\quad t := t+1$
7 **return** the best solution among $P(i), i = 0, \cdots, t$

単純な Genetic Algorithm の非効率性は，集中化のメカニズムが入っていないこと，および集団に含まれる解が同じ解に収束してしまうことに起因する．集中化は Local Search を組み込むことによって，ある程度は達成できる．また，複数の解を持つだけでは探索の多様化を行うためには不十分であり，多様化を促す何らかの工夫が必要となる．以下で述べる Generic Local Search および Fox の Quicker [7, 8] は，集中化，多様化および Tabu Search の記憶の概念を組み込んだ Genetic Algorithm の一般化と考えられる．

8 Generic Local Search

ここでは, Local Search を基礎とする多くのメタ戦略を含む一般的な枠組み (Generic Local Search) を導入する．

まず，実行可能解の集合 $\mathcal{F}$を含む集合を，擬似実行可能解の集合と呼び $\mathcal{F}'$と書く．すなわち，$\mathcal{F} \subseteq \mathcal{F}' \subseteq 2^{\mathcal{U}}$なる $\mathcal{F}'$を考える．実行不可能解も探索することによって，局所最適解にトラップされる危険性を回避しようというのが，擬似実行可能解を導入する理由であるが，一方では最終的な解が実行可能解でない可能性も出てくる．実行可能性を保持する方法しては種々の方法が提案されているが，どの方法を選択するかは問題の構造を吟味して決める必要がある．

さらに $\mathcal{F}'$の部分集合の集まりを考え，それを $\mathcal{X}$ と書く．すなわち，$\mathcal{X} = 2^{\mathcal{F}'}$である．このような $\mathcal{X}$の要素を一般化解 (generic solution) と呼ぶことにする．$\mathcal{X}$の要素は Genetic Algorithm における集団や Tabu Search における連続した移動によって生成された解の列に対応する．すなわち，現在の解を x_kとしたとき，それ以前に通過した $q-1$ 個の解とを合わせた q 組 $(x_{k-q+1}, x_{k-q+2}, \cdots, x_k)$ を解とみなす訳である．Tabu Search のこのような解釈は Fox [7, 8] による．また，通常の Local Search や Simulated Annealing 法は $|\mathcal{X}| = 1$ の特別な場合と考えられる．

Generic Local Search は，一般化解上で探索を行う．探索を制御するための何らかの情報を保持した集合を A で表す．A は，例えば Tabu Search の属性集合 (attribute set) に対応する．ここでは，A 上に以下の写像を導入する．

$$\Lambda : A \to R(Z)$$

すなわち，ある属性 $a \in A$ に対して実数または整数 $\Lambda(a)$ を対応させる．$\Lambda(a)$ を属性 a の状態 (state) と呼ぶ．例えば，Simulated Annealing 法においては，温度 Tはイテレーション数 k から非負実数への写像であり，Life Span Method では，寿命 LSは台集合 $\mathcal{U}$から非負整数への写像となる．また，目的関数 fも解集合という一つの属性から整数 (実数) 全体への写像であり，擬似乱数もイテレーション数 kから $[0,1]$ への写像と考える．

ここでは，属性の全ての状態の集合を一般化属性 (generic attribute) と呼ぶことにし，$\mathcal{A}$ と書くものとする．

一般化解の間の移動を制御する写像 Φを導入する．

$$\Phi : \mathcal{X} \times \mathcal{A} \to \mathcal{X}$$

また，一般化属性も以下の写像によって制御される．

$$\Psi : \mathcal{X} \times \mathcal{X} \times \mathcal{A} \to \mathcal{A}$$

Generic Local Search は，一般化解と一般化属性の組に対する (拡張した意味での) 近傍探索法と考えられる．以下にその概要を示す．

procedure generic local search
1 $x :=$ some initial generic solution ($\in \mathcal{X}$)
2 $\alpha :=$ some initial generic attribute state ($\in \mathcal{A}$)
3 **while** stopping-criterion $\neq$ yes **do**
4 $\quad x' := \Phi(x, \alpha)$
5 $\quad \alpha := \Psi(x, x', \alpha)$
6 $\quad x := x'$
7 **return** x

Fox の Quicker は，$\mathcal{A} = \emptyset$ の特殊型に対して Hajek の漸近的最適性を満たす Simulated Annealing 法を適用したものと考えられる．

9 おわりに

以上，Local Search を基礎とした種々のメタ戦略を概説した．

メタ戦略を題材とした論文の多くは「提案法が最も優れた解法である」という結論を急ぎがちであるが，あるメタ戦略が唯一の勝利者になることは，Johnsonらが Simulated Annealing 法に対する実験的解析の後に到達した結論「Simulated Annealing 法は便利な道具ではあるが，決して万能薬ではない」と同様に，まずあり得ないと考えられる．

実際には，(特定の問題を対象にしても) 最良の解法は複数存在し，解くべき問題のサイズ，使用可能な計算機の速度およびメモリの量，許容される計算時間などの諸要因に応じて使い分ける必要がある．我々にできることは，どのような場合に，どのアルゴリズムが最良の (または妥当な) 選択になるのかを，系統的な実験的解析を通して明らかにしておき，道具箱の中にしまっておくことである．

多くのメタ戦略に共通する特性として，大量の計算機資源 (時間および容量) を消費する代わりに比較的良い解を算出することがあげられる．今後は，計算機資源の廉価化および速度の向上に伴い，ある程度の計算時間を要してもさらに良好な解を探索しようとする要望が増大するので，メタ戦略へのニーズはますます高まるものと考えられる．

また，Local Search を基礎とするメタ戦略は，Local Search の特性である実装のし易さ，付加条件に対する頑強性を引き継いでおり，実際問題に対する馬車馬的アルゴリズムとなり得ると考えられる．メタ戦略の実質的な歴史は浅く，研究は始まったばかりであり，未だ多くの研究課題を残している．

なお，ここで述べられなかったメタ戦略および文献表については，[3] を参照されたい．また，OR 誌 [1, 2] にもサーベイを書いているので，そちらも参照されたい．

謝辞

本研究は一部文部省科学研究費の援助を受けている．

参考文献

[1] 久保幹雄. 巡回セールスマン問題への招待 (I),(II),(III). オペレーションズリサーチ, 39 (1,2,3):25–31,91–96,156–162, 1994.

[2] 高山裕士, 久保幹雄, 森戸晋. スケジューリング問題に対する Tabu Search. オペレーションズリサーチ, 40 (1): 1995.

[3] 久保 幹雄. メタヒューリスティックス. 室田 一雄 編, 離散構造とアルゴリズム IV, 第 4 章. 近代科学社, 1995.

[4] E. H. L. Aarts and J. H. M. Korst. *Simulated Annealing and Boltzmann Machines.* John Wiley & Sons, Chichester, U.K., 1989.

[5] V. Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimmization and Applications*, 45:41–51, 1985.

[6] N. E. Collins, R. W. Eglese, and B. L. Golden. Simulated annealing: An annotated bibliography. *American Journal of Mathematical and Management Sciences*, 8:205–307, 1988.

[7] B. L. Fox. Uniting probabilistic methods for optimization. In *Proc. the* 1992 *Winter Simulation Conference*, pages 500–505, 1992.

[8] B. L. Fox. Integrating and accelerating tabu search, simulated annealing, and genetic algorithms. *Annals of Operations Research*, 41:47–67, 1993.

[9] F. Glover. Tabu search I. *ORSA Journal on Computing*, 1:190–206, 1989.

[10] F. Glover. Tabu search II. *ORSA Journal on Computing*, 2:4–32, 1989.

[11] F. Glover, E. Taillard, and D. de Werra. A users' guide to tabu search. *Annals Operations Research*, 41:3–28, 1993.

[12] D. E. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning.* Addison-Wesley, Reading, MA, 1989.

[13] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13:311–329, 1988.

[14] P. Hansen. The steepest ascent mildest descent heuristic for combinatorial programming. In *The Congress on Numerical Methods in Combinatorial Optimization*, Capri, March 1986.

[15] J. H. Holland. *Adaptation in Natural and Artificial Systems.* Michigan Press, 1975.

[16] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation, part I, graph partitioning. *Operations Research*, 37:865–892, 1989.

[17] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation, part II, graph coloring and number partitioning. *Operations Research*, 39:378–406, 1991.

[18] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[19] M. Kubo. *The Life Span Method – A New Variant of Local Search –.* The Institute of Statistical Mathematics Cooperative Research Report 53, pp. 85–104, February, 1994.

[20] N. Metropolis, A. Rosenblut, M. Rosenblut, A. Teller, and E. Teller. Equation of state calculation by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[21] Z. Michalewicz. *Genetic Algorithm + Data Structure = Evolution Programs.* Springer-Verlag, 1992.

[22] P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Practice.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 1987.