入門講座
システム技術編- 2

システム最適化

玉置　久　神戸大学工学部電気電子工学科　講師
Hisashi Tamaki

Optimization

1 はじめに

最適化(optimization)は、システムを効率的かつ合理的に設計・計画・運用・制御したいと考えたときに直面する問題であり、その起源は1947年のD. B. Dantzigによる線形計画問題の定義およびその解法としてのシンプレックス法の提案に遡る。その後、この最適化の分野は大きな発展を遂げ、今日ではシステム工学の中心的課題となっている。

本講座では、最適化問題のモデル化(定式化)や解法について概観してみよう。なお、(広い意味での)最適化全般を取り扱うことは難しいので、数理的アプローチによる最適化を中心に話を進める。また、入門講座ということで、基本的な考え方・枠組みを直観的に把握できるように配慮した結果、厳密性が多少犠牲になっている点についてもご容赦願いたい。

2 最適化問題

一般に最適化問題は、決定すべき数・量(決定変数(decision variable)と呼ぶ)をxで表すことにすると、

$$\min_x \quad f(x) \quad (\text{または} \quad \max_x \quad f(x)) \cdots\cdots(1)$$

$$\text{sub. to} \quad x \in F \cdots\cdots(2)$$

$$F \subseteq X \cdots\cdots(3)$$

のように記述される[1]。式(1)の$f(x)$は目的関数(objective function)と呼ばれ、決定変数値xの良さを表す尺度である。また、xは対象とする問題(システム)の性質から定められる基本空間Xの要素であるが(式(3))、実際にはその部分集合であるFの要素しか取ることを許されない(式(2))。この意味で、式(2)を制約条件(constraint)といい、Fを可能領域(feasible region)という。さらに、制約条件(2)を満たすxを可能解と呼び、目的関数(1)の値を最小(あるいは最大)にする可能解x^*を最適解(optimal solution)と呼ぶ。

ここで、あるシステムを最適に運用したい場合を考えてみよう。最初に必要なことは、実際に上のような形式で問題を捉える(定式化する)ことである。まず、

(a)対象(システム)がどのようなものであって、

(b)どのような解が求められているか、

を正確に記述しなければならない。この(a)と(b)によって問題の制約条件および目的関数が明確にされることになる。現実の大規模・複雑なシステムを対象とする場合、この(a)および(b)を明確にすることすら容易でないといった状況も多いが、この部分を疎かにすることはできない。問題が定式化できてはじめて、(c)どのようにして最適解(あるいは準最適解)を求めるか、に進むことができる。この(c)の部分を達成するためには、問題についての十分な理解だけでなく、一般に最適化手法についての深い知識が必要とされることが多い。

さて、まず最適化問題としての定式化例を見ておこう。いずれも現実問題に比べると極めて単純化されたものではあるが、最適化問題の本質を把握するのには十分なものであろう。

例1(生産計画問題):m種類の資源$\mathrm{R}_i(i=1,\cdots,m)$を用いて$n$種類の製品$\mathrm{P}_j(j=1,\cdots,n)$を生産することを考えよう[2]。問題は、各資源の利用可能量に関する制約の下で、価値の総和が最大となるように各製品の生産量を決めることである。なお、製品P_jを1単位生産するのに必要な資源R_iの量a_{ij}、資源R_iの利用可能量b_i、製品P_jを1単位生産した場合の価値c_jは、あらかじめ与えられているものとする。このとき、各製品P_jの生産量x_jを決定変数に選ぶと、問題は以下のように定式化される。

$$\max_x \quad \sum_{j=1}^{n} c_j x_j \cdots\cdots(4)$$

$$\text{sub. to} \quad \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (i=1,\cdots,m) \cdots\cdots(5)$$

$$x_j \geq 0 \quad (j=1,\cdots,n) \cdots\cdots(6)$$

例 2 (化学平衡問題)：混合気体の平衡状態における分子組成を求める問題を考えよう[3]。混合気体はm種類の原子$A_i(i=1,\cdots,m)$からなり、各気体の原子数をb_iとする。化学反応の結果、この気体は混ざりあって平衡状態でn種類の分子$M_j(j=1,\cdots,n)$がそれぞれx_jモルずつできるものとし、1モルの分子M_jに含まれるA_iの原子数をa_{ij}とする。このとき問題は、化学反応前後の質量が保存されるという制約の下で、自由エネルギーを最小化する問題として捉えることができる。すなわち、x_jを決定変数として、以下のような問題に定式化される。

$$\min_{x} \quad \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n} x_j \log x_j - \left(\sum_{j=1}^{n} x_j\right) \log \left(\sum_{j=1}^{n} x_j\right) \quad \cdots\cdots (7)$$

$$\text{sub. to} \quad \sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i=1,\cdots,m) \quad \cdots\cdots (8)$$

$$x_j \geq 0 \quad (j=1,\cdots,n) \quad \cdots\cdots (9)$$

ただし、c_jは温度と圧力によって定まる定数である。

例 3 (ナップサック問題)：n個の荷物を袋に入れる問題を考えよう。各荷物$B_j(j=1,\cdots,n)$の重量a_jとその価値(重要度)c_j、および袋の許容重量bが与えられており、すべての荷物を袋に入れることはできないものとする。問題は、価値の総和ができるだけ大きくなるように、袋に入れる荷物を決めることである。いま、決定変数x_jを導入し、荷物B_jを袋に入れることを$x_j=1$、入れないことを0で表すと、問題は以下のように定式化される。

$$\max_{x} \quad \sum_{j=1}^{n} c_j x_j \quad \cdots\cdots (10)$$

$$\text{sub. to} \quad \sum_{j=1}^{n} a_j x_j \leq b \quad \cdots\cdots (11)$$

$$x_j \in \{0,1\} \quad (j=1,\cdots,n) \quad \cdots\cdots (12)$$

ここで、上で紹介した3つの最適化問題をもう少し詳しく見てみよう。まず例題1で紹介した問題では、決定変数が実数であって目的関数および制約条件がともに線形式で記述されている。このような問題を線形計画問題(linear programming problem)という。これに対して、例題2では目的関数が非線形となっている。このように、目的関数あるいは制約条件に非線形の式が含まれる問題を非線形計画問題(nonlinear programming problem)という。さらに、例題3では決定変数が整数値(離散的な値)に限られるものとなっており、この形式を整数計画問題(integer programming problem)という*1。また、整数計画問題に定式化される、すなわち可能領域が離散的となる問題を組合せ最適化問題(combinatorial optimization problem)という*2。

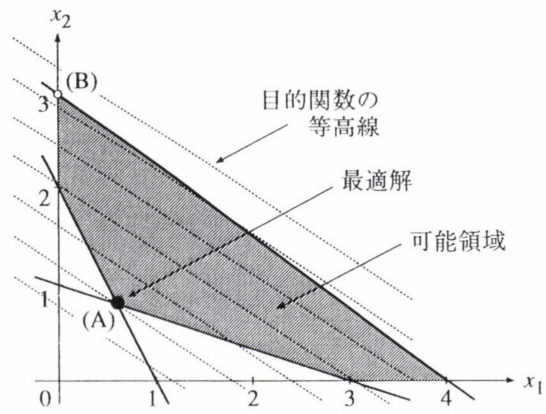


図1　線形計画問題の例

以下では、線形計画問題、非線形計画問題および組合せ最適化問題に対する代表的な解法について、その基本的な考え方を中心に概説しよう[1,2,5]。

3 線形計画法

いま、次の線形計画問題を考える。

$$\min_{x_1,x_2} \quad 2x_1+3x_2 \quad \cdots\cdots (13)$$

$$\text{sub. to} \quad 2x_1+x_2 \geq 2 \quad \cdots\cdots (14)$$

$$x_1+3x_2 \geq 3 \quad \cdots\cdots (15)$$

$$3x_1+4x_2 \leq 12 \quad \cdots\cdots (16)$$

$$x_1,x_2 \geq 0 \quad \cdots\cdots (17)$$

図1に、この問題の可能領域および目的関数の等高線を示す。この図からわかるように、2変数の線形計画問題の可能領域は凸多角形となる。また、目的関数の等高線は平行な直線となるので、最適解は可能領域である凸多角形の境界上に存在することがわかる。したがって、その凸多角形の端点(頂点)のうちの一つが必ず最適解になる*3。さらに、この性質は一般のn変数の問題についても成り立つことがわかっている。すなわち、線形計画問題の可能領域は一般に凸多面体となり、最適解はその凸多面体の端点の中に必ず存在する[2]。

これより、線形計画問題の最適解を見い出すためには、可能領域の端点(有限個)だけを調べればよいことがわかる。しかしながら問題の規模(変数や制約条件の数)が大きくなると、端点の数はたとえ有限個であっても膨大なものとなり、すべての端点を調べ尽くすのではなく、一部の端点だけを調べて最適解を見つけるような効率的な手順(ア

＊1　例題3では、決定変数が0または1の値しか取らないので、特に0-1整数計画問題とも呼ばれる。
＊2　これらはいずれも目的関数が唯一に定まる問題であるが、現実には必ずしも一つだけ、あるいは一つにまとめられるとは限らない。このように複数の目的関数を考慮しなければならない問題は多目的最適化問題(multiobjective optimization problem)と呼ばれ、最適化の一分野を形成している[4]。
＊3　目的関数の等高線が凸多角形の一つの辺と平行になる場合、その辺上の点(端点を含む)がすべて最適解となる。

ルゴリズム)が必要である。その代表的な手法が、冒頭でも触れたG. B. Dantzigによるシンプレックス法(simplex method：単体法ともいう)である。

ここで、端点を調べる(あるいは端点から端点に移動する)という手順を形式的(数学的)に表現してみよう。一般に、線形計画問題はいくつかの線形不等式や等式で表される制約条件および最小化あるいは最大化すべき目的関数によって記述される。しかしながら、このままでは種々の形式の問題表現が可能となるので、以下のような定義を考える。

$$\min_{x} \quad \sum_{j=1}^{n} c_j x_j \quad \cdots\cdots (18)$$

$$\text{sub. to} \quad \sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i=1,\cdots,m) \quad \cdots\cdots (19)$$

$$x_j \geq 0 \quad (j=1,\cdots,n) \quad \cdots\cdots (20)$$

この表現形式を線形計画問題の標準形(standard form)と呼ぶ。任意の線形計画問題を標準形に変換する場合、

(a)最小化すべき目的関数が与えられること

(b)すべての変数に非負条件が課されること

(c)制約条件がすべて等式で与えられること

を満足するようにしなければならない。それぞれ以下に示すような方法で簡単に解決できる。

まず(a)については、例えばもとの問題が目的関数を最大化する形：

$$\max_{x} \quad \sum_{j=1}^{n} c_j x_j \quad \cdots\cdots (21)$$

になっている場合、目的関数全体に-1を掛けることによって、

$$\min_{x} \quad \sum_{j=1}^{n} -c_j x_j \quad \cdots\cdots (22)$$

と変形できる。次に(b)について、ある非負制約のない変数x_jに対して、新たに2つの非負変数x_j'およびx_j''を導入し、

$$x_j = x_j' - x_j'' \quad \cdots\cdots (23)$$

とすればよい。最後の(c)については、不等号で与えられる制約条件、例えば

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \cdots\cdots (24)$$

となる場合、新たな非負変数(スラック変数と呼ばれる)λ_iを導入し、

$$\sum_{j=1}^{n} a_{ij} x_j + \lambda_i = b_i \quad \cdots\cdots (25)$$

のように書き換えることができる。

ここで、式(13)～(17)に示した例題を標準形に変換してみよう。この例の場合、スラック変数λ_1, λ_2およびλ_3を導入し、制約条件(14)～(16)を等式条件に書き換えるだけで、次の標準形が得られる。

$$\min_{x_1, x_2} \quad 2x_1 + 3x_2 \quad \cdots\cdots (26)$$

$$\text{sub. to} \quad 2x_1 + x_2 - \lambda_1 = 2 \quad \cdots\cdots (27)$$

$$x_1 + 3x_2 - \lambda_2 = 3 \quad \cdots\cdots (28)$$

$$3x_1 + 4x_2 + \lambda_3 = 12 \quad \cdots\cdots (29)$$

$$x_1, x_2, \lambda_1, \lambda_2, \lambda_3 \geq 0 \quad \cdots\cdots (30)$$

上の等式制約条件(27)～(29)を満たす変数の組$x=(x_1, x_2, \lambda_1, \lambda_2, \lambda_3)$は無数に存在するが、2つの変数を適当に選んでそれらの値を0とおくと、残りの3つの変数の値は一意に定められる。このようにして定められる解xを基底解(basic solution)といい、基底解の中で$x \geq 0$を満たすものを基底可能解(basic feasible solution)という。さらに、基底解を定める際に値を0とおいた変数を非基底変数(non-basic variable)といい、それ以外の変数を基底変数(basic variable)という。

基底解に関する重要な性質を紹介しておこう。それは"xが基底解であることと、可能領域の端点に対応することとは等価である"というものである(証明略)。例えば、$\lambda_1 = \lambda_2 = 0$とおいて定められる基底可能解

$$x = (0.6, 0.8, 0, 0, 7) \quad \cdots\cdots (31)$$

は図1中の端点(A)に、また$x_1 = \lambda_3 = 0$とおいた基底可能解

$$x = (0, 3, 1, 7, 0) \quad \cdots\cdots (32)$$

は端点(B)にそれぞれ対応していることがわかる。

これ以上の詳しい説明は省略させていただくが、シンプレックス法の数値計算においては、ある初期基底可能解からスタートして最適性の条件が満たされるまで、目的関数値が減少するように基底変数を入れ替える(すなわち基底可能解を更新する)という手続きが繰り返される。さらに、線形計画問題に関しては、

(a)双対問題(dual problem)とシンプレックス乗数(simplex multiplier)

(b)計算の効率化のための改訂シンプレックス法(revised simplex method)

なども重要である。また、最近の話題としては、

(c)KhachiyanやKarmarkarなどによって提案された方法に端を発した多項式時間アルゴリズムの研究

が挙げられる。これらについては参考文献[1,2,5]を参照されたい。

4 非線形計画法

前節では、目的関数および制約条件のいずれもが線形であるような特別な場合について、典型的な解法の基本的考え方を紹介した。ここでは、目的関数と制約条件のいずれかあるいは双方が線形ではない非線形計画問題を考える。

ところが、この非線形計画問題となると取り扱いが途端に難しくなるので、ここでは、制約のない非線形最適化問題：

$$\min_{x} \quad f(x) \quad \cdots\cdots(33)$$

$$x \in R^n \quad \cdots\cdots(34)$$

を取り上げ、解法の基本的考え方を示そう[1,3]。

この問題では、もちろん大域的最小点(global optimum)、すなわち任意の$x \in R^n$に対して$f(x^*) \leq f(x)$となる解$x^* \in R^n$を求めることが要求されているのであるが、$f(x)$の凸性が保証されているような特殊な場合を除いて一般には難しい。そこで、以下では局所的最小点(local optimum)、すなわちx^*の近傍にあるすべての$x \in R^n$に対して$f(x^*) \leq f(x)$となる解$x^* \in R^n$を求める方法を考えることにし、局所的最小点を単に最適解と呼ぼう。

線形計画問題の場合と同様、制約のない非線形最適化問題についても最適解を閉じた形の計算で求めることは一般には不可能である。そこで、以下の枠組みに基づいた繰り返し計算によって、最適解に収束させる手順が必要となる。

0° 適当なxの初期点$x^{(0)}$を与え、繰り返し回数$k=0$とする。

1° $x^{(k)}$の最適性を判定し、最適であれば3°へ。最適でなければ2°へ。

2° 目的関数fの値を減少させる新しい点$x^{(k+1)}$を生成する。$k=k+1$として1°へ戻る。

3° 計算終了。$x^{(k)}$が最適解である。

この際、

$$f(x^{(0)}) > f(x^{(1)}) > \cdots > f(x^{(k)}) > \cdots \quad \cdots\cdots(35)$$

となり、かつこの数列が最小値に収束するように手順を定めることが肝要である。この点に留意し、上の手順の2°で$x^{(k)}$から$x^{(k+1)}$を定める更新式として、探索方向(ベクトル)$p^{(k)}$および歩み幅(step size)$\alpha^{(k)}$を用い、

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} p^{(k)} \quad \cdots\cdots(36)$$

とする方法が一般的に用いられる。

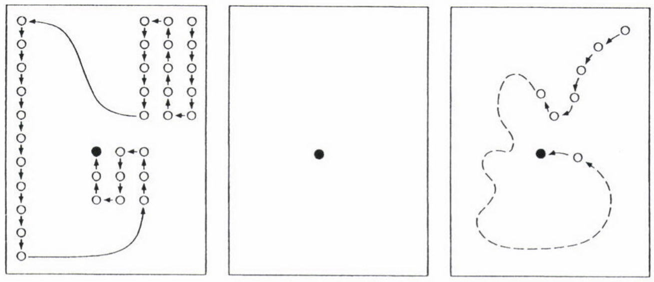
さらに、探索方向$p^{(k)}$の決め方として最も簡便なものは、目的関数の勾配：

$$\nabla f(x^{(k)}) = \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n} \right)^{\mathrm{T}}_{x=x^{(k)}} \quad \cdots\cdots(37)$$

のみを用いる最急降下法(steepest descent method)で、この場合、

$$p^{(k)} = -\nabla f(x^{(k)}) \quad \cdots\cdots(38)$$

のように探索方向が決められる。この最急降下法以外にも、目的関数のHesse行列(２階微分)までをも用いるNewton法、Newton法におけるHesse行列を近似的に求めることによって計算の手間を減らした準Newton法、連続する探索方向が互いに一次独立となるようにして収束性を改善した



(a) 列挙的手法　(b) 発見的手法　(c) 探索的手法

図2　組合せ最適化手法の基本的考え方
枠が可能領域、丸が一つの可能解をそれぞれ表す。また、黒丸は最終的に選ばれる解を表す。

共役方向法(conjugate direction method)あるいは共役勾配法(conjugate gradient method)などが代表的である[3]。歩み幅$\alpha^{(k)}$については、いずれの探索方向を用いる場合においても、一次元探索：

$$\alpha^{(k)} = \arg \min_{\alpha > 0} f(x^{(k)} + \alpha p^{(k)}) \quad \cdots\cdots(39)$$

によって$\alpha^{(k)}$を定めるといった方法が一般的である。

以上、本節では制約なしの非線形最適化問題に対する解法を概観した。非線形最適化問題に関する理論等を含めて、より詳細については参考文献[3,5]を参照されたい。

5 組合せ最適化

これまで扱ってきた線形計画問題および非線形最適化問題はいずれも連続変数を対象とするものであり、問題の可能領域は実数集合であった。一方、ここで取り上げる組合せ最適化問題に対しては、実数集合を対象とした連続性や微分の概念に基づく方法を直接利用することはできない。したがって、組合せ最適化問題の解法は連続変数の最適化手法と本質的に異なるものとなり、一般に解を調べ上げるというアプローチとならざるを得ない。しかし、可能領域Fの要素(可能解)の総数は有限ではあっても膨大な数にのぼることが多く、実際に列挙・探索する範囲をいかに限定するかが重要となる*4。

ここで、最適解あるいは準最適解を探索するという観点から組合せ最適化手法を捉えると、その基本的な考え方は、図２に示すように、

(a) 列挙的手法：全ての解を列挙して(またはこれと等価な手続きによって)厳密な最適解を見い出す

(b) 発見的手法：(準)最適解を生成するためのアルゴリズムまたはルールを利用して、ただ1つの解を求める

*4　線形計画問題に対するシンプレックス法は基本的に列挙法であるが、連続性に基づく効率的な基底変数の入れ替え(端点の移動)と最適性の判定(計算の終了)が可能となっている。

表1　ナップサック問題の例

荷物番号 j	1	2	3	4	5	6	7	8	許容量 b
重量 a_j	3	6	5	4	8	5	3	4	25
価値 c_j	7	12	9	7	13	8	4	5	

(c)探索的手法：列挙的手法と発見的手法の中間に位置する方法で、解空間の部分空間内を探索することによって、(準)最適解を探し出す

の３種類に分類される。それぞれの考え方に基づく代表的な手法としては、

(a)分枝限定法(branch and bound method)、分枝カット法(branch and cut method)、動的計画法(dynamic programming method)など

(b)欲張り法(greedy method)、ヒューリスティック・ルール(heuristic rule)など*5

(c)ランダム探索法(randam search method)、メタ・ヒューリスティクス(meta-heuristics)など

が挙げられよう。厳密な最適解が求められることを保証しているという点において(a)に属する手法は重要であるが、一般に膨大な計算時間を必要とする点において非現実的である(ただし、計算機の能力の飛躍的な向上に伴って最近再び見直されつつあるが…)。そこで、以下では実用的という点から(b)と(c)に焦点を当て、欲張り法の構成例およびメタヒューリスティクス手法の概略を紹介する。

まず、１節で紹介したナップサック問題を例題として取り上げ、欲張り法の構成例を示そう。なお、各荷物の重量(a_j)と価値(c_j)、袋の許容重量(b)がそれぞれ表１のように与えられている問題を考える。さて、欲張り法とは目的関数への貢献度を局所的に評価し、この評価値に基づいて可能解を直接構成していく方法である[7]。ナップサック問題に適用する場合、以下のような手順が考えられる。

0° n個の荷物をc_j/a_jの非増加順に並べ、この順に番号を付け直す。また、カウンタj=1、袋に残されている許容量$s=b$および入れられた荷物(番号)の集合$A=\phi$と初期化しておく。

1° $a_j \leq s$ならば、荷物B_jを袋に入れることに決め、$A=AU\{j\}$および$s=s-a_j$とする。

2° $j=n$または$s=0$ならば3°へ。そうでなければ、$j=j+1$として1°へ戻る。

3°計算終了。Aの要素が、袋に入れるべき荷物の番号を表している。

表１に示した例題(もともとc_j/a_jの非増加順に荷物の番号が付けられている)にこの手順を適用すると、荷物B_1, B_2, B_3, B_4の順に袋に入れられ、この段階で袋の残り許容量が$s=7$となる。したがって、次の荷物B_5を入れることができない。その次の荷物B_6を調べると$a_6=5$なのでこれを入れることができ、残り許容量が$s=2$となる。このような手順を繰り返していくと、結局、

$$x=(1,1,1,1,0,1,0,0) \cdots\cdots (40)$$

すなわち荷物B_1, B_2, B_3, B_4およびB_6を選んで入れるという解が得られる。このときの目的関数値は43となる。

なお、この問題の最適解は、例えば分枝限定法などを適用することによって(この例題の場合は手で全列挙することもできるが…)

$$x=(1,1,1,0,1,0,1,0) \cdots\cdots (41)$$

と求められ、目的関数の最適値は45であることがわかる。この例からもわかるように、欲張り法などの近似解法では比較的良い解が求められるが、最適な解が求められる保証はない。

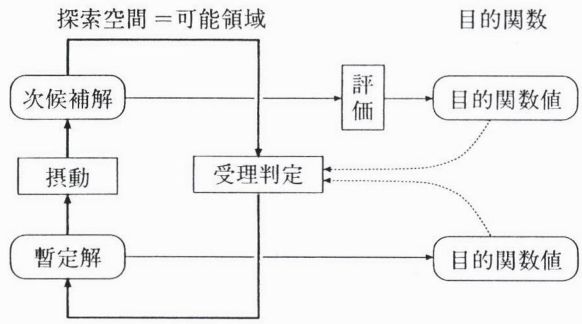


図3　メタヒューリスティクスによる探索の枠組み
メタヒューリスティクスの具体的なアルゴリズム(LS,SA,TS,GAなど)によって、外側のループが特徴づけられる。

次に、探索的手法の中でも最近特に注目されつつあるメタ・ヒューリスティクスを取り上げよう。大雑把に言うと、探索的手法は計算時間および求められる解の良さの両面において、列挙的手法と発見的手法の中間に位置するものである。すなわち、ある限られた計算時間の範囲内で、ある程度良い解を求めることが期待できる方法である。

図３に、メタヒューリスティクスによる探索の枠組みを示す。この枠組みでは、

(a)現時点での候補解(暫定解)に摂動を加えることによって次候補解を生成する手続き、および、

(b)新たに生成された次候補解の目的関数値を暫定解のものと比較することによって、暫定解を更新するか否かを決める手続き(受理判定)

*5　特定の問題に対する多項式時間アルゴリズム(例えば、２機械フローショップ・スケジューリング問題に対するJohnsonアルゴリズム[6])もこの範疇に入る。

によって、探索の過程が特徴付けられる。このとき、(i)良い解を見つけるためには、できるだけ広い領域を探索すること(diversification, exploration)、および(ii)計算時間を短縮させるためには、探索履歴等を利用してできるだけ探索領域を狭めること(intensification, exploitation)、といった相反する要求をうまくバランスさせ、解の精度向上と計算時間短縮の高バランス化が実現するように(a)および(b)の手続きを定めることが肝要である。

ここで、まずメタヒューリスティクスの基本形であると考えられる局所探索法(local search method：反復改善法(iterative improvement method) ともいう) を紹介しよう[8]。いま、候補解(暫定解)xに摂動を加えて得られる解の集合をxの“近傍”(neighborhood)と呼び$N(x)$と記す[*6]。局所探索法は、$N(x)$内にxより良い解があれば,それを改めて暫定解とする操作を、可能な限り繰り返すという方法である。

0° 初期解x^*を求め、これを暫定解とする。

1° $N(x^*)$内にx^*より良い解yがあれば、暫定解を$x^*=y$のように更新して1°へ。なければ、現時点での暫定解x^*を最終解として終了。

局所探索法を実現するためには、近傍$N(x)$の定義とともに、$N(x)$をどう探索し暫定解の更新をどのように行うか、などの細部を定める必要がある。いずれにしても、局所探索法によって最終的に選ばれた解x^*は、近傍$N(x^*)$により良い解が存在しないという意味で局所的最適解(local optimal solution)であり、未探索の領域にさらに良い解が残されている可能性がある。この可能性を少しでも低減させるため、

(a)初期解をいろいろと変えてみる

(b)探索に確率的動作を導入する

(c)暫定解より悪い解への遷移を許す

(d)複数の候補解を保持する

などの工夫が考えられている。これら(a)～(d)に着目した方法についての詳細は参考文献[7,8]を参照していただくことにして、代表的な手法とその要点だけを紹介しておこう。

(a)多スタート局所探索法(multi-start local search)：局所探索法の初期点(通常はランダムに生成される)を多数選んで、より大域的な最適解を得ようとする方法。

(b)シミュレーテッド・アニーリング (simulated annealing)：受理判定の手続きに確率的要素を取り入れた方法で、改悪であっても確率的に暫定解が更新される。さらに、“焼き鈍し(annealing)”における温度制御にならって、受理確率を温度パラメータtで制御する。

(c)タブー・サーチ(tabu search)：近傍$N(x)$内での最良の解yを求め、これが改悪であっても、現時点での解を$x=y$のように更新することを基本とする。しかしながら、この操作をそのまま実行すると循環(cycling)が生じやすい。そこで、タブーリストTを用意しておき(例えば、過去何回かの繰り返しで訪れた解のリスト)、T内への遷移を禁止することで循環を抑制している。

(d)遺伝アルゴリズム(genetic algorithm)：自然界における生物・生体の集団遺伝に基づく進化の過程を模倣した最適値探索法であり、複数個の候補解(集合)を保持して探索を進めるという点に特徴がある。摂動に対応する手続きとしては、交叉(crossover)や突然変異(mutation)などの遺伝的操作(genetic operation)が用いられる。

6 まとめ

本講座では、数理的なアプローチによる最適化問題を対象として、その典型例である線形計画問題、非線形計画問題および組合せ最適化問題のモデル化・定式化および代表的な解法を概観した。冒頭でも述べた通り、基本的な考え方や枠組みを中心に紹介しているので、わかり難い点が少なくないとも思われるが、適宜参考文献を参照していただくとしてご容赦いただきたい。本講座をきっかけとして最適化について多少なりとも興味を持たれれば幸いである。

参考文献

1) 西川禕一，三宮信夫，茨木俊秀：最適化，岩波書店，(1982)
2) 福島雅夫：数理計画入門，朝倉書店，(1996)
3) 今野浩，山下浩：非線形計画法，日科技連，(1978)
4) 中山弘隆，谷野哲三：多目的計画法の理論と応用，計測自動制御学会，(1994)
5) 伊理正夫，今野浩，刀根薫監訳：最適化ハンドブック，朝倉書店，(1995)
6) 今野浩，鈴木久敏：整数計画法と組合せ最適化，日科技連，(1982)
7) 茨木俊秀：離散最適化法とアルゴリズム(岩波講座応用数学［方法8］)，岩波書店，(1993)
8) 茨木俊秀：計測と制御，34 (1995) 5，340.

(1997年11月19日受付)

＊6 近傍の具体的な構成(すなわち摂動の具体的な手続き)に関する一般的・汎用的とも言える手法はなく、問題例に応じて種々の摂動手続きが構成されている。