0-1 ナップサック問題に対する メタヒューリスティックスの性能比較

細田　康之　(沼田一道　助教授)

1．はじめに

意思決定や生産計画に現れる最適化問題は、「離散的資源緒の最適配分を求めよ」とか、「仕事を処理する最適順序を求めよ」というように、多くの場合、与えられた制約を満たす最適な組み合わせを探索すること—組合せ最適化問題—に帰着する。

組合せ最適化問題は、探索すべき組み合わせの個数が有限なので、問題の規模が小さい場合は全数探索を行うことも可能であるが、規模が大きくなると探索対象の組合せは、爆発的に増大し、実用的な時間内に厳密な最適解を求めることは困難になる。このため、近似的な解を求める発見的解法（ヒューリスティックス）や汎用的な探索戦略にもとづくメタヒューリスティックスとよばれる解法が研究されている。

本研究では、組合せ最適化問題に属する 0-1 ナップサック問題を例題とし、それに対する 2 つのメタヒューリスティックス、多スタート局所探索法と禁断探索法の性能を比較検討する。

2．ナップサック問題

例えば、n 個の投資案件があり、それぞれの投資額と、投資した場合の期待利益が与えられている。このとき、「総予算の枠内で、期待利益が最大となるような投資の組合せを求めよ。」という問題がナップサック問題である。

2.1 定式化

この投資組合せ問題の例において、投資案件の個数を n,それぞれの投資額を $a_j(j=1,2,\ldots,n)$,期待利益を c_j,,総予算を b とする。各投資案を採用するか、しないかの二者択一とした場合の問題は 0-1 変数 x_j, (x_j,=1 で投資案を採用、x_j,=0 で不採用）を用いて(P)のような 0-1 ナップサック問題に定式化される。同一案を複数採用できる場合は、(P')のような整数ナップサック問題になる。

$$
(\mathrm{P})\quad \left[\begin{array}{ll} \text{最大化} & z=\sum_{j=1}^{n} c_j x_j \\ \text{制約} & \sum_{j=1}^{n} a_j x_j \le b \\ & x_j \in \{0,1\},\quad j=1,2,\ldots n \end{array}\right.
\qquad
(\mathrm{P}')\quad \left[\begin{array}{ll} \text{最大化} & z=\sum_{j=1}^{n} c_j x_j \\ \text{制約} & \sum_{j=1}^{n} a_j x_j \le b \\ & x_j \ge 0,\quad x_j \in Z\ ,\ j=1,2,\ldots n \end{array}\right.
$$

本研究では、0-1 ナップサック問題を例題として、取り上げる。

2.2 厳密解法

ナップサック問題は、n が小さい場合、その厳密な最適解を Dynamic Programming(DP)による解法で求めることができる。0-1 変数 x_j を $x_1,x_2,\ldots x_r$ と r まで考え、容量を $y(y \leqq b)$ としたときの、最適値を $f_r(y)$とすると、$f_b(r)$は次の漸化式を満たす。

$$
f_r(y)=\begin{cases} \max\{f_{r-1}(y), c_r+f_{r-1}(y-a_r)\} & if \quad a_r \le y \\ f_{r-1}(y) & if \quad a_r > y \end{cases} \qquad \text{ただし、} f_o(y)=0, f_r(0)=0
$$

$f_n(b)$を計算することにより厳密解が求まる。本研究の実験でも、小規模の問題については厳密解（値）をこの解法で求めておき、評価の際の参考にする。

2.3 釘付けテスト

0-1ナップサック問題(P)の0-1変数x_jを$0 \le x_j \le 1$で置き換えた連続緩和問題$(\overline{\mathrm{P}})$について考える。この問題の最適値を$z(\overline{\mathrm{P}})$とし、0-1ナップサック問題の最適値を$z(\mathrm{P})$とすると$z(\mathrm{P}) \le z(\overline{\mathrm{P}})$である。また、暫定値(何らかの方法で求めた実行可能解に対応する目的関数値)を$\tilde{z}$とすると、$\tilde{z} \le z(\mathrm{P}) \le z(\overline{\mathrm{P}})$である。この関係を利用して、問題の規模を縮小することが出来る。

[定理]

(P)の変数$x_1,\ldots x_2,\ldots x_n$を対応する(目的関数と制約式の)係数の比$\gamma_j = c_j / a_j$が、$\gamma_1 \ge \gamma_2 \ge \ldots \ge \gamma_n$となるように並べ替え、(番号を付け換え) $\sum_{j=1}^{p-1} a_j \le b < \sum_{j=1}^{p} a_j$ を満たすpを定める。

$$\mu_j^* = \begin{cases} c_j - \gamma_p a_j, & j = 1,2,\ldots p \\ -c_j + \gamma_p a_j, & j = p+1,\ldots n \end{cases}$$

を定義し、$J^* = \left\{ j \middle| \mu_j^* \ge z(\overline{P}) - \tilde{z} \right\}$ としたとき、

(i) $j \in J^*$かつ$j < p$なら、$\tilde{z}$よりも良い目的関数値を与える(P)の実行可能解は、$x_j = 1$ を満たす。

(ii) $j \in J^*$かつ$j > p$なら、$\tilde{z}$よりも良い目的関数値を与える(P)の実行可能解は、$x_j = 0$ を満たす。

この定理を用いることによって、問題の規模を縮小することが可能になる。

本研究では、データをランダムに発生させた 0-1 ナップサック問題 (P) と、(P) を釘付けテストによって縮小した問題について実験を行い、釘付けテストにどの程度の効果があるのかを考える。

3. メタヒューリスティックス

メタヒューリスティックスは、対象とする問題の近似解 x が 1 つ適当な方法で得られたとして、この近似解 x をさらに改良する(より良いものを探す)ためのアルゴリズムの枠組みである。この探索を行う際に基本となるのが近傍の概念である。

0-1 ナップサック問題の解候補の集合を X として、はじめに解候補を 1 つ選び出し、これを $x(x \in X)$ としたときの、この x に簡単な操作を施して、得られる解候補の集合を近傍 $N(x)$ と呼ぶ。(図 1 参照)

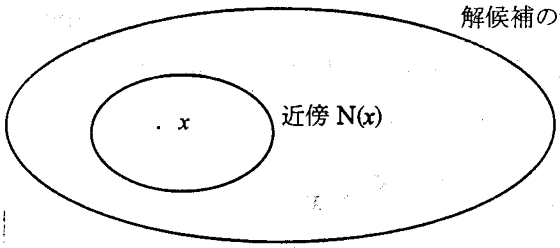

図 1.近傍

本研究では、メタヒューリスティックスとして、多スタート局所探索法とタブー探索法を用いる。

3.1 多スタート局所探索法(Multi-start Local Search(MLS))

多スタート局所探索法は、局所探索（$N(x)$内に x より良い解があればそれに置き換えるという操作を可能なかぎり反復する）を多数の初期解に与える解法である。

step 0 : 最良値 $MLS := -\infty$,探索回数 $m = 0$ とする。

step 1 : $m := m+1$ とする。初期解をランダムに生成し、現在解 x とし、初期値を現在値 Zmax とする。

step 2 : $N(x)$内に x により良い解x'があれば、$x := x'$とし，現在解を更新する。Zmax を更新し、

step 2 を繰り返す。$N(x)$内により良い解がなければ step 3 へ。

step 3： Zmax > MLS ならば、MLS:=Zmax と更新し、$updtf$:=m とする。

step 4： $m - updtf > 5*n$ ならば MLS を出力して終了。さもなければ、step 1 へ。

3.2 タブー探索法(Tabu Search(TS))

タブー探索法は、現在の解を x とし、$N(x)$内の最良の解 x'を求め、それが仮に改悪であっても、$x:=x'$ と更新する。しかし、これをそのまま実行すると、x から $x' \in N(x)$に更新されたのち、$N(x')$内の最良の解を求めると、再び、x に戻ってしまう可能性が高い。そこで、このようなサイクリングをさけるため、タブー集合 T を作り、$N(x)-T$ 内の最良解 x'へ移るという解法である。

step 0： 初期解を求め、暫定解 x^*(現在解 x)とする。探索回数 m:=0,タブー配列 $TAB[j]$:=0(j=1,2,…n) とする。

step 1： m:=m+1 とする。$N(x)-T$ 内の最良解 x'を求め、$f(x') > f(x^*)$ならば、$updtf$:=m として、暫定値 $f(x^*)$:=$f(x')$と更新する。また、$x:=x'$更新し、x から x'へ変更のあった場所(添字)を j^* として、$TAB[j^*]$:=m+ θ (θ はランダムに決定)とする。

step 2： $m - updtf > 5*n$ ならば $f(x^*)$を出力して終了。さもなければ、step 1 へ。

4.ナップサック問題への適用

本研究では、多スタート局所探索法における近傍はn個の0－1変数のうち3個まで反転させた集合、タブー探索法における近傍はn個の0－1変数のうち1個だけを反転させた集合とした。多スタート局所探索法では、実行可能解のみを対象として探索を行う。タブー探索法では、実行可能でない解も探索の対象とし、その場合には超過分 $\sum_{j=1}^{n} a_j x_j - b$ を罰金（ペナルティ）として目的関数から差し引いて評価する。

要するに$g(x) = \sum_{j=1}^{n} c_j x_j + \alpha * \min\left(0, b - \sum_{j=1}^{n} a_j x_j\right)$として、$g$を探索の評価関数とする。$\alpha$ は、$1 \le \alpha \le \sum_{j=1}^{n} c_j$の範囲で最近実行可能解が、見付かったか否かで増減させている。また、暫定値を更新する実行可能解が見付かった場合はタブーに抵触してもその解へ移動するものとした。

5. 計算機実験の結果と比較

実験に使用するデータを作成し、前節の解法で解く。0-1 ナップサック問題(P)を（1）メタヒューリスティックスのみ、（2）釘付けテスト、メタヒューリスティックスを併用、の２つの方法で解き、実験を行い、得られる解の精度を比較する。

5.1 データの作成

データは、問題のサイズ n、a_j の上限と下限、c_j の上限と下限を入力し、ランダムに発生させて作成した。また、b は a_j の和に対しての割合（%）で与えた。

表 1．実験で用いた数値

ファイル名	サイズ n	b (%)	a (下限)	a (上限)	c (下限)	c (上限)
sample 1-10	30	75	5	40	15	120
sample 11-20	30	75	10	20	30	60
sample 21-30	100	75	5	40	15	120
sample 31-40	100	75	10	20	30	60
sample 41-50	30	75	5	40	a + 10	
sample 51-60	100	75	5	40	a + 10	

sample 1-10 では、a, c のばらつきを大きくし、sample 11-20 では、a, c のばらつきを小さくし、その違いによる影響を調べた。同様の比較を、sample 21-30, sample 31-40 では、n=100 にして、データを発生させた。また、sample 41-50, sample 51-60 では、a と c に相関関係を持たせ、n=30, n=100 において、それぞれデータを発生させた。

ここで、n=30 では、DP を用いて、厳密解を求めておいた(所要時間は、2～3 分)。n=100 では、厳密解を求めていない。

5.2 実験結果

計算機による実験結果を以下の表に示す。

表 2. 数値実験結果

	メタヒューリスティックスのみ				釘付けテスト、メタヒューリスティックス併用				
	MLS		TS			MLS		TS	
ファイル名	MLS/厳密	計算時間	TS/厳密	計算時間	残変数	MLS/厳密	計算時間	TS/厳密	計算時間
sample 1-10	0.999942	393.6	1.000000	366.1	10.6	0.999890	35.8	1.000000	56.4
sample 11-20	0.998758	412.3	1.000000	399.3	21.2	0.999817	203.6	1.000000	216.1
sample 21-30	0.999164	62478.6	0.998841	4439.3	19.4	0.999164	450.9	0.999181	236.6
sample 31-40	0.997604	59387.3	0.998330	4724.4	73.9	0.999461	32002.0	0.998357	2124.5
sample 41-50	0.997447	466.7	0.999725	343.3	28.3	1.000000	327.7	0.999874	286.4
sample 51-60	0.995274	87907.9	0.997682	3336.1	98.3	0.997682	43923.3	0.997682	3224.7
全体	0.998032	35174.4	0.999096	2268.08	41.95	0.999336	12823.9	0.999182	1024.12

n=100 のときは、厳密値ではなく、連続緩和値を用いた。計算時間の単位は、msec。使用した計算機は、EPSON pentium(r) II 450MHz。使用言語は、Delphi3.1 である。

6. 考察

n=30 のときは、MLS, TS ともに、ほぼ厳密値を得ることができた。n=100 においてもほぼ同程度の結果を得たが、a, c のばらつきが小さかったり、a, c に相関関係がある場合など解きにくい問題においては、TS のほうが良い結果を得た。MLS は、問題の性質によって精度にばらつきがあった。計算時間は、MLS よりも TS のほうが短かった。

釘付けテストを行ってからメタヒューリスティックスで解いたときのほうが釘付けテストを行わずに直接、メタヒューリスティックスで解いたときよりも良い結果が得られた。これは、問題のサイズを小さくすると探索の精度が上がる(良い解を見つけやすくなる)からであろう。特に、MLS で釘付けテストによる改善が得られた。また、計算時間も大きく短縮することができた。

また、a, c に相関関係がある場合、釘付けテストによって問題の規模を縮小することが、ほとんどできなかった。

7. おわりに

本研究では、0-1 ナップサック問題に対して MLS と TS を適用した。どちらの場合も厳密値(連続緩和値)に近い値を得ることができ、この 2 つのメタヒューリスティックスの有意性が確認された。MLS と TS では、近傍の大きさに違いがあるので、公平性に問題があるが、MLS の性能の高さは意外だった。また、釘付けテストの効果も確認できた。

【参考文献】

[1] 今野　浩、鈴木　久敏：「整数計画法と組合せ最適化」，日科技連(1982)

[2] 茨木　俊秀：「メタヒューリスティックス：その意義と可能性」,第 5 回 RAMP セミナー資料(1995)