ハイパフォーマンス
コンピューティング 84－4
（２０００．１２．８）

# 高次元アルゴリズムによる整数変数を含む系の最適化

寺前裕之、斎藤茂、柳正秀*、植草常雄*
ＡＴＲ環境適応通信研究所、ＮＴＴファシリティーズ(*)

0-1 整数変数を最適化変数として含む系に対し、最適化手法として高次元アルゴリズムを用いた最適方法について報告する。実際の最適化問題としては建物エネルギーシステムにおいて、複数のヒートポンプ(エアーコンディショナー)を用いて冷暖房をまかなう場合を考え、ヒートポンプ容量を評価関数として最小値を求めることを試みた。床面積 10,000m$^2$ の標準的なホテルの冷暖房需要データを用い各月各時刻における冷暖房の各出力を計算することにより、その最小値と最適な運用法ならびにヒートポンプ数と各々の容量が求まることを示した。このホテルのデータでは、2台では最小値にならず、最低3台が必要となることがわかった。

## A Study on Optimization of System Containing Integer Variables

Hiroyuki Teramae, Shigeru Saito, Masahide Yanagi*, Tsuneo Uekusa*
ATR Adaptive Communications Laboratories, 2-2 Hikaridai, Seikacho Soraku-gun, Kyoto 619-0288, Japan
*NTT Power and Building Facilities, 2-13-1 Kitaohtsuka, Toshimaku, Tokyo 170-0004, Japan

This paper describes the optimization method of the systems containing the 0-1 integer variables. The Hamiltonian algorithm, a method based on the classical dynamics, was used to find the minimum of the cost function. The example of the problem is the optimum number and the capacities of the heat pump of the building. We performed the optimization of the heat/cool outputs of at the each month/hours using the standard heat demand pattern of the hotel of 10,000m$^2$ and the total heat pump capacities as the cost function. We showed at least 3 heat pumps were necessary to obtain the minimum heat pump capacities.

## １．はじめに

ある装置で２種類の異なった働きを行える物を使用している場合を考える。さらにこの２種類の出力が同時には取り出せないとする。このような装置を含む系の最適化問題を取り扱う場合には、いずれの出力が得られているかを表す整数変数を導入する必要が生じる。この問題はいわゆる 0-1 混合計画問題であり、分枝限定法、平面切除法、群論的手法などの最適化方法が知られているが、扱う整数変数が増加すると解を得るのが大変困難になることが知られている。このような 0-1 整数変数をも最適化変数として取りこみ、手法として新上らによる高次元アルゴリズム[1]を用いた最適化方法について報告する。例題として建物エネルギーシステムにおいて、ヒートポンプのみを用いて冷暖房をまかなう場合に、その容量の最小値を求めた。ただし、高次元アルゴリズム使用の際には、最適解が求まるかどうかは大きく初期値に依存する。本研究では、その初期値依存性を、運動量の混合（ミキシング）を行うことで改善することが出来たので併せて報告する。

## ２．モデルと最適化方法

以下のような関係式を持つ場合について、

$$\sum_{k=1}^{n} c_{hw}(m,k)\cdot x_{hw}(m,h,k) = d_{hw}(m,h) \qquad (1)$$

$$\sum_{k=1}^{n} (1-c_{hw}(m,k))\cdot x_{hc}(m,h,k) = d_{hc}(m,h) \qquad (2)$$

ここで、$d_{hw}$ および $d_{hc}$ は $m$ 月の時刻 $h$ における2種類の異なった需要データを表す。変数 $c_{hw}$ が切替を表す変数で0または1とし、同じ月は同一の出力モードで運転すると仮定した。2種類の異なった需要を同時刻に満たす必要が生じるため装置は複数台必要で、$n$ は装置の台数。評価関数として、

$$V = \sum_{k=1}^{n} v(k) = \sum_{k=1}^{n} \max_{m,h}[c_{hw}(m,k)\cdot x_{hw}(m,h,k) + (1-c_{hw}(m,k))\cdot x_{hc}(m,h,k)] + a_1 c_{hw}(m,k)^2\cdot(c_{hw}(m,k)-1)^2$$

を最小とする最適化問題として取り扱った。4次関数は最適化終了時に $c_{hw}$ を0または1にするためのペナルティ項として加えた。$a_1$ は適当な定数。さらに式(1)(2)を満たすようにペナルティ項を加えて、ポテンシャルエネルギー $V$ を持つ質量1の質点の運動を追跡して最小値を求める。

$$V_2 = a_2 \sum_{m,h}[\sum_{k=1}^{n} c_{hw}(m,k)x_{hw}(m,h,k) - d_w(m,h)]^2$$

$$V_3 = a_3 \sum_{m,h}[\sum_{k=1}^{n} (1-c_{hw}(m,k))x_{hc}(m,h,k) - d_c(m,h)]^2$$

$$V_4 = a_4 \Big| \sum_{m,h}[\sum_{k=1}^{n} c_{hw}(m,k)x_{hw}(m,h,k) - d_w(m,h)] \Big|$$

$$V_5 = a_4 \Big| \sum_{m,h}[\sum_{k=1}^{n} (1-c_{hw}(m,k))x_{hc}(m,h,k) - d_c(m,h)] \Big|$$

また $c_{hw}(m,k)$ に関してはある時刻において2種類の同時需要がある場合には、最低1台は各々の需要に対して必要となるため、$1 \le \sum_{k=1}^{n} c_{hw}(m,k) \le n-1$ を満たす必要があり、そこで $V_6 = a_5(c_{hw}(m,k)-1)$ $(c_{hw}(m,k) \le 1)$ および $V_6 = a_5(c_{hw}(m,k)-(n-1))$ $(c_{hw}(m,k) \ge n-1)$ を評価関数に加えた

この運動を追跡する際に適当な初期座標と初期運動量を与える必要があるが、この初期値に対する依存性が非常に大きい。そこで最適化空間の高次元化のために便宜的に加えた運動量の変形を行う。このことにより運動に均一性を持たせ、初期値の依存性を減じることが出来る。

適当な正の規格直交行列 $b_{ij}$ を用いて運動量の混合（ミキシング）を行う。

$$\ddot{x}_i = \partial H / \partial p_i = \sum_j b_{ij} p_j$$

$$p_i = -\partial H / \partial x_i = f_i$$

$$\ddot{x}_i = \sum_j b_{ij} f_j$$

以上により、ミキシングを行うと各変数にかかる力を適当な線形結合を取り直すことと等価になることがわかる。

ミキシングのための行列 **B** を生成する手順を以下に示す。実対称行列 **A** を乱数等によって作り対角化。その固有ベクトルを **C**、固有値を対角要素とする行列を $\varepsilon'$ とすると、

$$\mathbf{CAC}^T = \varepsilon'$$

この固有ベクトル **C** を利用して、ミキシングのための行列 **B** を作る。行列が positive definite となるように、固有値を 1 近傍に値を分散させて選ぶ。$\delta_j$ をミキシング定数と定義する。

$$\varepsilon_j = \mathbf{1} + \delta_j \quad (\delta_j > -\mathbf{1})$$

以上から求める行列 **B** の満たすべき固有方程式は、次式のようになる。

$$\mathbf{BC} = \mathbf{C}\varepsilon$$

ここで **C** は定義より直交行列であるので、$\mathbf{CC}^{\mathrm{T}} = \mathbf{1}$、右から $\mathbf{C}^{\mathrm{T}}$ を乗じることにより、

$$\mathbf{B} = \mathbf{C}\varepsilon\mathbf{C}^T$$

が求めるミキシングのための行列になる。$\delta$ の値を増減する事でミキシングの度合いをコントロールできる。

## 3．プログラミングと並列処理

以上のような関係式を用いて実際にプログラムを作成した。ヒートポンプ数が増加してくると計算量がかなり増大してくることが見込まれたため、ループ変数としての月を利用することで並列処理を行った。ここでは並列化ライブラリとして MPI を用い以下に示すような並列処理のプログラムを作成した。

```
      implicit real*8(a-h,o-z)
      parameter (m = 12, ih = 24)
      dimension xhw(m,ih,maxhp),xhc(m,ih,maxhp),cw(m,maxhp)
      dimension xhmax(m)
      include 'mpif.h'

      call mpi_init(ierr)
      icomm=mpi_comm_world
      call mpi_comm_rank(icomm,me,ierr)
      call mpi_comm_size(icomm,nproc,ierr)

      calculate the initial geometry and velocity
      if(imix.eq.1) then
              calculate mixing matrix
      endif

      do loop=1,nloop
      calculate xhmax=max(cw*xhw+(1-cw)*xhc)
      call mpi_allreduce(xhmax,xhmax,nhp,mpi_double_precision,
     1 mpi_max,icomm,ierr)
      do k=1,nhp
        do i=1,m
          if(mod(i,nproc).eq.me) then
            do j=1,ih
              calculate first derivative of cw, xhw, xhc
            enddo
          endif
        enddo
        enddo
        if(imix.eq.1) then
c         calculate mixing of derivatives
          call mpi_allreduce(dcw,dcw,m*nhp,mpi_double_precision,
     1 mpi_sum,icomm,ierr)
          call domix(vdcw,dcw,vscr,m,nhp)
          call mpi_allreduce(dxhw,dxhw,inum,mpi_double_precision,
     1 mpi_sum,icomm,ierr)
          call mpi_allreduce(dxhc,dxhc,inum,mpi_double_precision,
     1 mpi_sum,icomm,ierr)
          call domix2(vdxhw,dxhw,vscr1,m,ih,nhp)
          call domix2(vdxhc,dxhc,vscr1,m,ih,nhp)
        endif
      do k=1,nhp
        do i=1,m
          if(mod(i,nproc).eq.me) then
            do j=1,ih
              calculate next geometries and total energy
            enddo
          endif
        enddo
        enddo
      enddo
      call mpi_finalize(ierr)
      stop
      endsubroutine domix2(vdxhw,dxhw,vscr,m,ih,nhp)
      include 'mpif.h'
      integer status
      common/pproc/nproc,me,icomm,status(mpi_status_size)
      dimension
     1   vdxhw(m,ih,m,ih,nhp),dxhw(m,ih,nhp),vscr(m,ih,nhp)
      do k=1,nhp
      do i=1,m
      if(mod(i,nproc).eq.me) then
      do j=1,ih
      vscr(i,j,k)=0.0d0
        do i1=1,m
        do j1=1,ih
          vscr(i,j,k)=vscr(i,j,k)+dxhw(i1,j1,k)*vdxhw(i1,j1,i,j,k)
        enddo
```

```
      enddo
    enddo
  endif
  enddo
  enddo
  do k=1,nhp
  do i=1,m
  if(mod(i,nproc).eq.me) then
  do j=1,ih
  dxhw(i,j,k)=vscr(i,j,k)
  enddo
  enddo
  endif
  enddo
  return
  end
  subroutine domix(vdcw,dcw,vscr,m,nhp)
  implicit real*8(a-h,o-z)
  include 'mpif.h'
  integer status
  common/pproc/nproc,me,icomm,status(mpi_status_size)
  dimension vdcw(m,m,nhp),dcw(m,nhp),vscr(m)
  do k=1,nhp
  do i=1,m
  if(mod(i,nproc).eq.me) then
      vscr(i)=0.0d0
          do 310 j=1,m
              vscr(i)=vscr(i)+dcw(j,k)*vdcw(j,i,k)
            enddo
        endif
      enddo
    do i=1,m
        if(mod(i,nproc).eq.me) then
            dcw(i,k)=vscr(i)
          endif
    enddo
    enddo
    enddo
  return
  end
```

## 4．結果と考察

冷暖房需要データとして延べ床面積 10,000m² の標準的なホテルのデータを使用して、計算された最適値を表 1 に示す。またここで用いた標準的なホテルの負荷データは図 1 に示した。このホテルの負荷データでは、6-10 月は冷房需要のみであるが、他の月は冷暖房需要が同時に発生するという特徴的な負荷パターンとなっている。

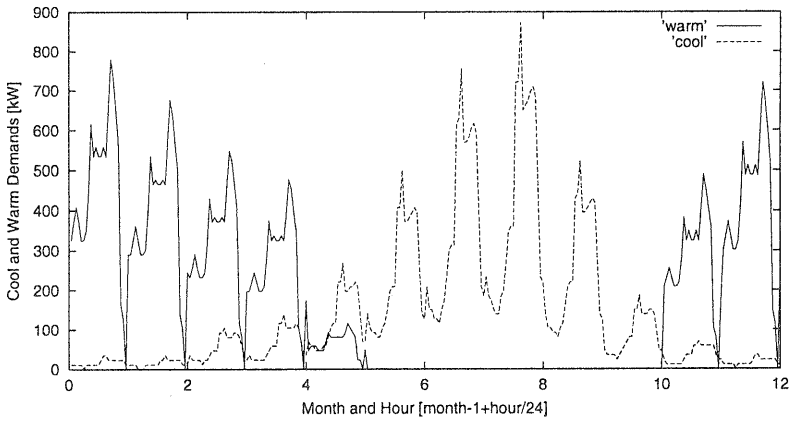


図 1．冷暖房の需要データ

高次元アルゴリズムにより計算された、2 分割時の各ヒートポンプの出力状況を図 2 に示した。ここで重要なのは、ヒートポンプ 1 の他の月は全て冷房が割り当てられているが、5 月のみ暖房が割り当てられていることである。この冷暖房の切り替えを行うことにより、より大きな容量が必要となるのを回避している。実際この切り替えを考慮せず、ヒートポンプ 1 が常に冷房に割り当てられたとすると、ヒートポンプ容量は 1.0533 に対して 1.3 となる。なお最適化の初期値としてはこのような切り替えを意識していないものを用いた。ただし 2 台のヒートポンプでは切り替えによってこれ以上のヒートポンプ容量の節約は行えないが、より大きい分割数では、理論的最小値(標準的ホテル負荷データでは 8 月の冷房最大負荷値)を取ることができ、この最小値は 1 となる。ここで用いた負荷パターンでは 3 分割以上でこの最小値をとることがわかる。なお理論最小値とは、冷暖房需要の和の最大値であり、これを下回ると熱需要をまかなえなくなる。

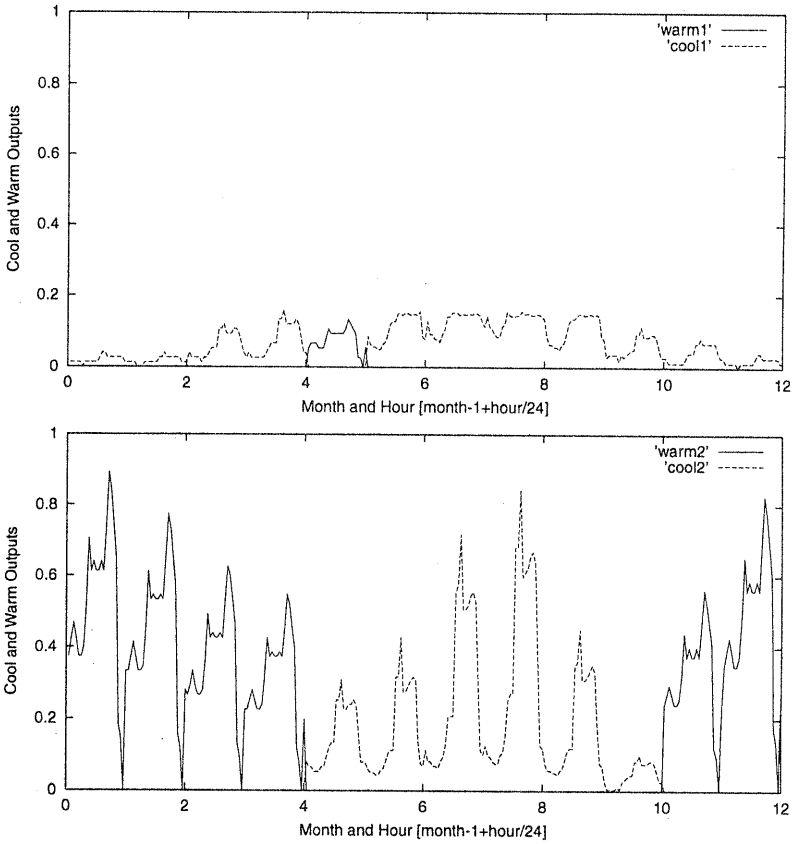


図 2. 2 分割時の各ヒートポンプの出力状況

実際の高次元アルゴリズムによる計算値は数値的な誤差を含むため理論値と完全には合致しないが、表 1 に示したように 10 分割までについて最適化を行い、n=2 の時に 1.0523、n=3-10 の時 0.9995 または 0.9996 となっており、ほぼ一致しているとみなすことができよう。以上から高次元アルゴリズムによる最適化がうまく働いて最小値が求まっていることがわかる。

表 1 高次元アルゴリズムによるヒートポンプ最小値

| 台数 n | 理論最小値 | 最小値 |
|---|---|---|
| 2 | 1.0533 | 1.0523 |
| 3 | 1.0 | 0.9995 |
| 4 | 1.0 | 0.9996 |
| 5 | 1.0 | 0.9995 |
| 6 | 1.0 | 0.9996 |
| 7 | 1.0 | 0.9996 |
| 8 | 1.0 | 0.9996 |
| 9 | 1.0 | 0.9996 |
| 10 | 1.0 | 0.9995 |

なお計算に必要となるパラメーター値$a_1 - a_7$については表 2 にまとめた。計算方法のセクションで述

べたように、最適化開始時と終了時で値が異なるパラメーターは別に示した。

表2. 評価関数中のペナルティー項のパラメーター値

| パラメーター | 最適化開始時 | 最適化終了時 |
|---|---|---|
| $a_1$ | 0.2 | 5 |
| $a_2$ | 5 | 2000 |
| $a_3$ | 5 | 2000 |
| $a_4$ | 0.05 | 0.2 |
| $a_5$ | 0.05 | 0.2 |
| $a_6$ | 0.5 | --- |
| $a_7$ | 0.5 | --- |

また、ミキシング定数の最大値は0.05として各々の$\delta_j$はこの値を越えないように乱数により生成した。

3台分割時にミキシングを行った場合と行わなかった場合について、表3に示した。ミキシング定数が0の場合（ミキシングしない場合）には理論最適値より大きい値が計算されるが、ミキシング定数を0.05とすることで、理論最適値が得られている事がわかる。高次元アルゴリズムを用いた他の最適化においても、同様な手法により正しい最適解が求まる事がわかっている。適切なミキシングを行うことによって局所解近傍で一種のカオス的な運動が導入され、局所解から抜け出すきっかけが与えられていると考えられる。

表3. ミキシングによる最適値の変化。理論最適値は1。

| ミキシング定数$\delta$ | 最適値 |
|---|---|
| 0.0 | 1.1299 |
| 0.05 | 0.9934 |

以上のように本論文では、評価関数にペナルティー項を加え、最適化方法として高次元アルゴリズムを使用する事により0-1整数変数を含めた系の最適化を試みた。実際の最適化は建物エネルギーシステムにおいて、ヒートポンプ(エアーコンディショナー)のみを用いて冷暖房をまかなう場合を考え、ヒートポンプ容量を評価関数として最小値を求めることを試みた。床面積10,000m$^2$の標準的なホテルの冷暖房需要データを用いて、その最小値と最適な運用法ならびに最適な分割数が求まることを示した。このホテルの需要データではヒートポンプが2台ではどのような運用法でも最小値をとることはできず、最低3台以上が必要とされることがわかった。

今後はヒートポンプ以外の熱源機器を用いた系であるとか、他の評価関数を用いた場合への適用が考えられ、現在検討中である。また本論文の結果を遺伝的アルゴリズムやシュミレーティッドアニーリングなどの他の近似解法から得られる結果との比較は大変興味深いと思われる。

**文献**

[1] Shinjo, K. and Sasada, T., *Phys. Rev.*, **E54**, 4686 (1996).

[2]寺前、斎藤、柳、植草、情報処理学会第59回全国大会予稿集、1-167.