

5N-5 高次元アルゴリズムによる離散変数を含む系の最適化(4)

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(*)

1. はじめに

我々はある装置において2種類の切り替えが行われる場合の最適化問題について考察してきた。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0-1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、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。0-1整数変数を最適化変数として取りこみ、最適化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以前に報告した[2-4]。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。これらの結果から、高次元アルゴリズムを用いて建物エネルギーシステムの最適化が行える事がわかったが、特に混合性の工夫を行った場合には[3]、計算に要するCPU時間が増大するため、より整数変数が増大した場合に問題によっては適用が困難となる事が予想される。そこで本研究においては、最適化計算の並列処理を試みたので報告する。

2. プログラミングと並列処理

以前に発表した関係式[2-3]を用いて実際にプログラムを作成した。ここでは、ループ変数として月を利用することで並列処理を行った。並列化ライブラリとしてMPIを用い図1に示すような並列処理のプログラムを作成した。なおノード間の通信量を極力少なくするために、月を変数として並列処理を行ったため、最大でも6並列までにとどめた。混合性の工夫を行う場合には、図1から容易にわかるように、ベクトル演算が大量に発生するため、並列化による大幅なパフォーマンス向上が期待できる。実際の計算機環境としては以下の2種類を用いた。

1. 並列計算機 DEC Alpha Server 21164A 612MHz 8CPU 8GB メモリー、MPI ライブラリ DMPI(DEC 製)
2. PC クラスタ Intel Pentium3 450MHz 6CPU、100BaseT イーサネットボード、FreeBSD 4.2Release、MPI ライブラリ LAM Ver. 6.2 β

上記、1.は共有メモリー型の並列計算専用機であり、2.は通常のパーソナルコンピューターを複数台イーサネットを介してクラスタ構成としたものである。

3. 結果と考察

表1. 並列計算によるパフォーマンスの向上(DEC Alpha Server)

CPU数	混合性工夫無し	混合性工夫あり
1CPU	290 秒(1.00)	7357 秒(1.00)
2CPU	230 秒(1.26)	3861 秒(1.95)
6CPU	177 秒(1.63)	1558 秒(4.72)

A Study on Optimization of System Containing Switching Variables Part 4.

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 Kitaotsuka, Toshimaku, Tokyo 170-0004, Japan

```

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
    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 il=1,m
          do jl=1,ih
            vscr(i,j,k)=vscr(i,j,k)+dxhw(il,jl,k)*vdxhw(il,jl,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
    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 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
return
end

```

図1. 並列化プログラムの概要

表1に並列専用機による並列化の効果について示した。混合性の工夫を行った場合には、大規模な行列計算が必要となるため、並列化により大きな効果が得られていることがわかる。一方、混合性の工夫を行わなかった場合には、今回の計算ではコスト関数が非常に簡単であるために、それほど効果が得られていないにせよ、若干ではあるが並列化の効果が得られていることがわかる。なおここには記していないが、PCクラスターではネットワーク I/O 待ちが大きく、専用機よりは得られる効果は小さい事がわかった。

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.
- [3] 寺前、斎藤、柳、植草、情報処理学会第60回全国大会予稿集、2-79.
- [4] 寺前、斎藤、柳、植草、情報処理学会第61回全国大会予稿集、1-199.