Octave-function OrtPro: Use and Features

Viktor Stovba, Petro Stetsyuk, Volodymyr Zhydkov
vik.stovba@gmail.com

Department of Non-smooth Optimization Methods
V.M. Glushkov Institute of Cybernetics, Kyiv, Ukraine

V.M. Glushkov Institute of Cybernetics, Seminar, room 403, 14:00, August 25, 2020

Outline

Outline

OrtPro function: purpose

OrtPro finds a solution of system

$$Ax = b, \quad x_{low} \leq x \leq x_{up}, \quad n \gg m \approx 1000$$

$$A \in \mathbb{R}^{m \times n}, \quad x \in \mathbb{R}^n, \quad b \in \mathbb{R}^m$$

or reporting that system is inconsistent.

1. Stetsyuk, P.I. On linear equation system solving with two-sided constraints on variables. Algebra and linear optimization: International scientific conference (Ekaterinburg, Russia, May 14-19, 2012).

OrtPro function: quadratic problem

OrtPro is designed for quadratic programming problem

$$f^\star = f(x^\star) = \min_{x \in \mathbb{R}^n} \{ f(x) = (x - x_0)^T \mathrm{diag}(\omega)(x - x_0) \} \quad (1)$$

$$\text{s.t.} \quad Ax = b, \;\; x_{low} \leq x \leq x_{up} \quad (2)$$

$$\omega \in \mathbb{R}^n \, (\omega > 0), \;\; A \in \mathbb{R}^{m \times n}, \;\; b \in \mathbb{R}^m$$

Remark: objective function $f(x)$ can be replaced by generalized quadratic separable function $F(x) = \sum_{i=1}^{n} (d_i x_i^2 + c_i x_i), d_i > 0$

OrtPro function: algorithm

OrtPro implements dual algorithm [1]

- finds maximum point u^* of the concave differentiable function

$$\psi(u) = \min_{x_{low} \le x \le x_{up}} \{(x-a)^T \mathrm{diag}(\omega)(x-a) + u^T Ax - u^T b\},\ u \in \mathbb{R}^m \quad (3)$$

- calculates x^* – solution of the problem (1)-(2)

$$x^* = x(u^*) = min(max(x^{low}, x^0 - 0.5 diag(\omega)^{-1} A^T u^*), x^{up}) \quad (4)$$

- reports if constraints (2) are inconsistent using condition

$$\psi(u) > x_{up}^T diag(\omega) x_{up}, x_{up} = max(abs(x^{up} - x^0), abs(x^0 - x^{low})) \quad (5)$$

OrtPro function: $\psi(u)$ maximization

OrtPro uses octave-function **ralgb5** – Shor's r-algorithm with

- constant space dilation coefficient
- adaptive step regulation towards normed antisubgradient

Program **ralgb5** requires $5m^2$ multiplications on every iteration, not accounting $\nabla\psi(u)$ calculation

Computational complexity for finding ψ^* with accuracy ε is

$$O\left(m\log\frac{1}{\varepsilon}\right) * (5m^2 + 4nm)$$

Outline

OrtPro function: input/output

Input parameters

- $\omega, x_0, x_{low}, x_{up}, A, b$
- r-algorithm parameters
  - α – space dilation coefficient
  - h_0, q_1 – parameters for adaptive step regulation
  - $\varepsilon_u, \varepsilon_g, maxitn$ – stop parameters

Output parameters

- x_r – approximation to x^*
- f_r – $f(x_r)$
- ist – stop code
  - $2, 3$ – solution is optimal
  - $4, 5$ – solution is not found
  - 6 – constraints are inconsistent

OrtPro function: Octave code

```
1  function [xr,fr,ist] = OrtPro(w,x0,A,b,xlow,xup,alpha,h0,q1,epsu,epsg,maxitn);
2  itn=0; hs=h0; m=length(b); B=eye(m); u=zeros(m,1); xs=max(abs(xlow-x0),abs(xup-
       x0));
3  fup = 1.1*sum(xs.*xs.*w); nfg = 1; xr = min(max(xlow,x0),xup);
4  g0 = A*xr-b; fr=sum((xr-x0).*(xr-x0).*w)+g0'*u; dgr=dg0=norm(g0);
5  printf("itn%4d f%14.6e fr%14.6e dg0%11.3e",itn,fr,fr,dg0);
6  printf("dgr%11.3e ls%2d nfg%4d \n",dgr,0,nfg);
7  if (norm(g0) < epsu) ist = 2; return; endif
8  for (itn = 1:maxitn)
9      du = B * (g1 = B' * g0)/norm(g1);
10     d=1; ls=0; ddu=0;
11     while (d > 0)
12         u += hs * du; ddu += hs * norm(du); nfg++;
13          xs = min(max(xlow,x0-0.5*A'*u./w),xup); g1 = A*xs - b;
14         dg1 = norm(g1); f = sum((xs-x0).*(xs-x0).*w) + g1'*u;
15         if (dg1 < dgr) fr = f; xr = xs; dgr = dg1; endif
16         if (f > fup) ist = 6; return; endif
17         if (dg1 < epsg) ist = 2; return; endif
18         ls++; (mod(ls,3) == 0) && (hs *= 1.1);
19         if (ls > 500) ist = 5; return; endif
20         d = du' * g1;
21     endwhile
22     (ls == 1) && (hs *= q1);
23     printf("itn%4d f%14.6e fr%14.6e dg1%11.3e",itn,f,fr,dg1);
24     printf("dgr%11.3e ls%2d nfg%4d \n",dgr,ls,nfg);
25     if (ddu < epsu) ist = 3; return; endif
26     xi = (dg = B' * (g1 - g0))/norm(dg);
27     B += (1/alpha - 1) * B * xi * xi';
28     g0 = g1;
29 endfor
30 ist = 4;
31 endfunction
```

OrtPro function: matrix-vector operations

OrtPro has 8 matrix-vector operations:

- For r-algorithm: $B' * g_0, B * g_1$ (line 9), $B' * (g_1 - g_0)$ (line 26), $(B * xi) * xi'$ (line 27)
- To calculate $\nabla\psi(u)$: $A * xr$ (line 4), $A' * u, A * xs$ (line 13)

They can be accelerated by means of:

- **BLAS** (Basic Linear Algebra Subprograms)
- **Intel(R) MKL** (Math Kernel Library)
- accelerating computation using **Huawei hardware**
- parallel computation using **Huawei cloud**

OrtPro function: application

OrtPro can be used for

solving of the problem (1)-(2), where $n \gg m$, $m \approx 1000$

Remark: OrtPro implementation uses **dense** matrix A, but it can use **sparse** matrices as well in case of $A * x$ and $A' * u$ operations are implemented

Outline

OrtPro function: test 1, test 2

Octave code

```
1  x0 = zeros(n,1); w = ones(n,1);
2  A = ones(m,1)*[1:n]/n + [1:m]'*ones(1,n)/m; A(1:m,1:m) += eps*diag([1:m]);
3  x00 = ones(n,1) + [1:n]'/n; xup = 1.1*x00; xlow = 0.9*x00; b = A * x00;
```

test 1: eps = 1, matrix A has full rank
test 2: eps = 10^{-7}, matrix A is close to singular
(if eps = 0 matrix A has rank 2 for any m)

Experiments for test 1 and test 2 are described in [2].

2. Stetsyuk P.I., Zhydkov V.A., Fesyuk A.V. Octave function OrtPro: use and features. System analysis and information technologies (SAIT 2014): 16th International scientific conference (Kyiv, May 26–30).

OrtPro function: code for Octave 3.0.0

```
1  nmtest = [1 5000  50; 2 5000  50; 1 50000  50; 2 50000  50];
2  fout = fopen("OrtPro-2test.out","w");
3  fprintf(fout, "x0=1+i/n; epsu = 1.d-7 epsg = 1.d-6 \n");
4  for itest = 1:rows(nmtest)
5      ntest = nmtest(itest,1); n = nmtest(itest,2); m = nmtest(itest,3);
6      x0 = zeros(n,1); w = ones(n,1);
7      if (ntest == 1) eps = 1.0; endif
8      if (ntest == 2) eps = 0.000001; endif
9      A = ones(m,1)*[1:n]/n + [1:m]'*ones(1,n)/m; A(1:m,1:m) += eps*diag([1:m]);
10     x00 = ones(n,1) + [1:n]'/n; xup = 1.1*x00; xlow = 0.9*x00; b = A * x00;
11     alpha = 1.5; h0 = 1.0; q1 = 0.9; epsu = 1.e-6, epsg = 1.e-7, maxitn = 1000,
12     ntest, n, m, tstart = time();
13     [xr,fr,ist] = OrtPro(w,x0,A,b,xlow,xup,alpha,h0,q1,epsu,epsg,maxitn);
14     fprintf(fout,"\n ntest n m ist time %2d %8d %8d %2d %8d",
15                          ntest,n,m,ist,round(time()-tstart));
16     fprintf(fout,"\n fr max(Axr-b) max((Axr-b)/b) %16.8e %12.4e %12.4e \n",
17                    fr, max(abs(A*xr-b)),max(abs(A*xr-b)./b));
18 endfor
19 fclose(fout);
```

OrtPro function: results for Octave 3.0.0

No	n	m	f_r	$max(\|Ax - b\|/b)$	ist	time (sec)
1	5000	50	11658.5744	3.7095e-9	3	7
2	5000	50	11668.1667	1.6077e-12	2	15
1	50000	50	116658.583	1.2241e-9	3	65
2	50000	50	116668.167	1.6518e-13	2	105

The following computer was used:

Pentium E5200 (2.5 GHz), 4 Gb RAM, Windows Server

OrtPro function: results for NEOS solvers

Matrix A was generated in the following way:

AMPL: `let a[i,j] := 5*Uniform(0,1); let x00[j] := 3*Uniform(0,1);`
Octave: `rand("seed", 2020); A = 5*rand(m,n); x00 = 3*rand(n,1);`

n	m	time (sec)		
		Gurobi	CPLEX	Octave 3.0.0
10000	400	13	27	20
10000	500	23	31	26
10000	700	44	57	38

* _solve_time (August 16, 2020)
* time() in Octave interpreter mode

Conclusion 1: quadratic program

The approach used in dual algorithm construction can be applied to problems of type (1)-(2) with strictly convex separable functions, in particular for strictly convex separable quadratic function, i.e. for

Quadratic Program (n,m), $n \gg m \approx 1000$, $d > 0$

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{n} (d_i x_i^2 + c_i x_i), \tag{1Q}$$

$$\sum_{i=1}^{n} a_{ij} x_i = b_j, \quad j = \overline{1, m}, \tag{2Q}$$

$$x_i^{low} \leq x_i \leq x_i^{up}, \quad i = \overline{1, n}. \tag{3Q}$$

Conclusion 2: block constraints

Dual algorithm can be significantly accelerated using block structure of constraint matrix [3],[4, p. 26-29].

3. Stetsyuk P.I., Nurminski E.A., Solomon D.I. Transportation problem and orthogonal projection on linear manifolds. Transport systems and logistics, Chisinau: Vth International scientific conference (December 11-13, 2013).

4. Mikhalevich V.S., Trubin V.A., Shor N.Z. Optimization problems of production-transportation planning. Models, methods, algorithms. Moscow: Nauka, 1986 (in Russian).

Questions?

THANK YOU
FOR YOUR ATTENTION