HOPDM, version 2.30 Benchmark Results: Solving Linear Programming Problems

Jacek Gondzio

Logilab, HEC Geneva, Department of Management Studies
University of Geneva, 102 Bd Carl-Vogt, 1211 Geneva, Switzerland

`gondzio@com7000.unige.ch` or `gondzio@ibspan.waw.pl`
`http://ecolu-info.unige.ch/~logilab/gondzio/`

March 22, 1998

Abstract

This note presents computational results obtained with HOPDM interior point code applied to solve large scale linear programming problems.

Keywords: Large Scale LP, Benchmarks.

Table 1: Factorizations: normal equations vs. augmented system (Pentium Pro, 200 MHz).

Name	Dense	Normal equations				Augmented system			
	Col	$nz(AA^T)$	$nz(L)$	it	time	$nz(H)$	$nz(L)$	it	time
scsd8-2r-27	56	12435	30508	11	6.75	11794	25841	13	4.63
scsd8-2r-54	110	39435	110532	11	24.10	23458	45617	13	8.76
scsd8-2r-108	218	137175	419316	12	155.04	46786	89789	15	20.12
scsd8-2r-216	434	507615	1631712	13	1195.52	93442	178133	17	45.46
scsd8-2r-432	866	1948335	6435988	10	9426.95	186754	354821	16	91.53

Table 1 compares the efficiency of the normal equations approach with that of the augmented system. We ran HOPDM with two different factorization methods on one specific stochastic optimization problem `scsd8-2r` with a number of scenarios varying from 27 through 432. Stochastic linear problems are known to contain dense columns, which clearly favorizes augmented system factorization. Data collected in Table 1 is self-explanatory: for two factorization techniques we report the number of off-diagonal nonzero entries in the matrices to be factorized and in their triangular factors as well as the solution statistics (iterations and CPU time) to reach 8-digit optimal solution.

The computations were done on a PC with a 200MHz Pentium Pro processor and 128 MB of RAM run under Linux operating system. HOPDM was compiled with the FORTRAN GNU compiler `fort77`; options `-O2 -m486` were used. All solution times were measured with the Linux `time` command. Solution times are given in seconds and include reading the MPS input file and writing MPS output report.

Table 2: Factorizations: normal equations vs. augmented system (Power 2, 58 MHz).

Name	Dense	Normal equations				Augmented system			
	Col	$nz(AA^T)$	$nz(L)$	it	time	$nz(H)$	$nz(L)$	it	time
dbir1	223	1200381	2674564	14	28m 14s	1053925	3654470	15	51m 31s
dbir2	233	1300060	2711493	19	37m 22s	1135032	4040297	19	62m 27s
qap8	16	13952	193032	9	24s	7296	195546	9	25s
qap12	24	74592	2135388	13	26m 11s	38304	1987717	14	16m 33s
qap15	30	186075	8191638	16	180m 58s	94950	7332416	15	121m 52s

In the majority of cases (80-90 % of real-life LP's), normal-equations factorization is sparser and, in consequence, cheaper to compute. Actually, this is the default factorization approach in HOPDM. There exist, however, rare problems which favorize the augmented system factorization. Examples are given in Table 2.

Table 2 compares the efficiency of the normal equations approach with that of the augmented system. We ran HOPDM with two different factorization methods on a number of difficult linear problems. `dbir*` problems contain dense columns but are solved faster with the normal equations approach. Quadratic Assignement Problems (`qap*`) have very "short" columns but are solved faster with the augmented system approach. Data collected in Table 2 is self-explanatory: for two factorization techniques we report the number of off-diagonal nonzero entries in the matrices to be factorized and in their triangular factors as well as the solution statistics (iterations and CPU time) to reach 8-digit optimal solution.

The computations were done on an IBM RS6000 Power 2 workstation with a 58 MHz processor and 128 MB of RAM running under AIX 4.2 operating system. HOPDM was compiled with the FORTRAN compiler `xlf`; options `-O -qarch=pwr` were used. Solution times are given in seconds (they do not include reading the MPS input file and writing MPS output report).

Table 3: Solving stochastic optimization problems (Pentium Pro, 200 MHz).

Name	m	n	nz	Dense	$nz(H)$	$nz(L)$	it	time
aircraft	3754	7517	24034	751	19667	19759	10	8.12
deter3	7647	21777	44619	73	44474	84913	25	36.60
deter7	6375	18153	37191	61	37070	70789	24	29.46
fxm2.16	3900	5602	31412	36	28596	78439	25	18.50
fxm3.16	41340	64162	372452	36	341076	725261	36	316.69
fxm4.6	22400	30732	250142	23	238096	560766	29	161.02
pltexpA2.16	1726	4540	11154	18	8289	20949	18	5.84
pltexpA3.16	28350	74172	181650	18	135697	363991	42	212.07
pltexpA4.6	26894	70364	172326	8	130315	1185774	34	609.95
sc205.2r.800	17613	17614	48031	802	53638	100881	14	33.34
sc205.2r.1600	35213	35214	96031	1602	107238	201681	23	112.42
scagr7.2r.432	16431	17300	70477	1297	42372	75678	26	39.52
scagr7.2r.864	32847	34580	140893	2593	84708	151278	28	91.13
scfxm1.2r.128	19036	28914	107572	260	90228	277101	42	126.39
scfxm1.2r.256	37980	57714	214452	516	179828	551533	42	260.16
scrs8.2r.256	7196	9765	30035	257	8421	13044	13	9.06
scrs8.2r.512	14364	19493	59987	513	17269	26864	15	20.94
scsd8.2r.216	4330	30310	125480	434	93442	178133	17	45.03
scsd8.2r.432	8650	60550	250760	866	186754	354821	16	90.24
sctap1.2r.216	12990	20784	91728	650	81272	143210	21	48.63
sctap1.2r.480	28830	46128	203664	1442	180536	317714	25	132.56

In Table 3 we report results of the solution od several large stochastic optimization linear programs. The column Dense shows the number of nonzeros in the densest column.

The computations were done on a PC with a 200MHz Pentium Pro processor and 128 MB of RAM run under Linux operating system. HOPDM was compiled with the FORTRAN GNU compiler `fort77`; options `-O2 -m486` were used. All solution times were measured with the Linux `time` command. Solution times are given in seconds and include reading the MPS input file and writing MPS output report.