Electronic Mail Distribution of Linear Programming Test Problems*

David M. Gay

Lucent Bell Laboratories
Murray Hill, New Jersey 07974

Netlib is an experimental facility for distributing public-domain numerical software by electronic mail. An article, ‘‘Distribution of Mathematical Software by Electronic Mail’’, about *netlib* by its creators, Jack Dongarra and Eric Grosse, appears in the May, 1987 issue of *Communications of the ACM* (volume 30, number 5, pages 403–407).

To use netlib, you send electronic mail to a *netlib* server, such as *netlib@research.att.com*, asking for what you want; if available, it will be sent back to you by return electronic mail. For example, to see what *netlib* currently offers, you would send the message ‘‘`send index`’’ to *netlib* at one of the following addresses:

Network	*netlib* address
USENET	`research!netlib`
U.S. Internet	`netlib@research.att.com`
U.S. Internet	`netlib@ornl.gov`
European Internet	`netlib@nac.no`
European EARN/BITNET	`netlib%nac.no@norunix.bitnet`
European X.400	`s=netlib; o=nac; prmd=uninett; c=no;`
European EUNET/uucp	`nuug!netlib`
Australian Internet	`netlib@draci.cs.uow.edu.au`

The powers that be have consented to allow distribution of linear programming test problems via *netlib*. To see what problems are currently available, send the message ‘‘`send readme from lp/data`’’ to *netlib*. The return mail that *netlib@research.att.com* would send you (at the time of this writing) is shown below. New problems will appear first on *netlib@research.att.com* and *research!netlib*, then (after an unpredictable length of time) on the other *netlib* servers. (There is no guarantee that *netlib* service will continue to be available; on the other hand, there are presently no plans to withdraw this service.)

The *netlib* linear programming test problems are also available by anonymous *ftp* from *netlib.att.com*; look in the subdirectory `netlib/lp/data`.

Anyone willing to make interesting linear programming test problems publicly available is encouraged to submit them for possible inclusion in *netlib*’s `lp/data` chapter. Contact David Gay electronically (`research!dmg`), by phone at (908) 582–5623, or in writing at the address shown above. You can save me some hassle by making sure the problems you send have a single right-hand side, a single (or no) BOUNDS set, and a single cost (free) row with signs reversed, if necessary, so that the LP is to be minimized. (Otherwise I will arbitrarily delete all but one right-hand side, free row, and BOUNDS set before putting the problems into `lp/data`. The obvious point of this exercise is to make as clear as possible what problem is to be solved.)

The LP test problems currently available from *netlib* are in the form of MPS files or Fortran programs that generate MPS files.

Problems in MPS format can be large. To reduce transmission times, *netlib* sends a compressed

---

* This is an updated version of pp. 10–12 of the December, 1985 issue (No. 13) of the *COAL Newsletter*.

version of MPS files — and makes available C and Fortran 77 programs for expanding the problems back to MPS format. The compression and expansion involve no roundoff, so the problems should not be changed by the process. The compressed form contains only printing ASCII characters, and is broken into lines at most 72 characters long. There is one checksum character per line (sent somewhat later in a line full of checksums), so the expansion program is likely to warn you if a transmission error has occurred. Even with the checksum characters, problems are typically reduced to between 16% and 26% of their original size (where size is in characters of lines stripped of trailing blanks).

The current ‘‘`readme from lp/data`’’ follows:

```
===== LP/DATA README (formerly index) =====

To reduce transmission times, linear programming test problems
are stored in a compressed format; issue the netlib request

    send emps.f from lp/data

to obtain a Fortran 77 Subset program for expanding the test problems
into MPS-standard input form.  The program includes comments giving
test data.  To get a (more efficient and convenient) C version of this
program (without the test data), issue the netlib request

    send emps.c from lp/data

If you are not familiar with MPS files, see Chapter 9 of "Advanced
Linear Programming" by Bruce A. Murtagh, McGraw-Hill, 1981,
or look at the information on MPS files in

    http://www.mcs.anl.gov/home/otc/Guide/faq/

All the material described here is now available by ftp from
netlib.bell-labs.com (login: anonymous; Password: your E-mail address;
cd /netlib/lp/data).  If you can, please use ftp to obtain the larger
problems.  Note that the *.Z files in lp/data must be copied in binary
mode and uncompressed two ways: first with uncompress, then with emps.
If you are using a Unix system and your solver reads standard input,
you can save some disk space by executing, e.g.,
    zcat pilot.Z | emps | solver
On some Unix systems and with solvers that require a named file,
you may also be able to use a named pipe, e.g.,
    /etc/mknod pilot.mps p
    zcat pilot.Z | emps >pilot.mps & solver pilot.mps
    rm pilot.mps

The "Kennington" problems, sixteen problems described in "An Empirical
Evaluation of the KORBX Algorithms for Military Airlift Applications"
by W. J. Carolan, J. E. Hill, J. L. Kennington, S. Niemi, S. J.
Wichmann (Operations Research vol. 38, no. 2 (1990), pp. 240-248),
are available only by ftp: login as above, and cd lp/data/kennington .
More details appear in lp/data/kennington/readme .

People who use EBCDIC systems may wish to issue the netlib request

    send ascii from lp/data
```

to get a listing of the distinct character codes that appear in the compressed LP data files -- for the uncompression routines to work, these distinct ASCII characters must be translated into distinct EBCDIC characters.

The column and nonzero counts in the PROBLEM SUMMARY TABLE below exclude slack and surplus columns and the right-hand side vector, but include the cost row. We have omitted other free rows and all but the first right-hand side vector, as noted below. The byte count is for the compressed file; it includes a newline character at the end of each line. These files start with a blank initial line intended to prevent mail programs from discarding any of the data. The BR column indicates whether a problem has bounds or ranges: B stands for "has bounds", R for "has ranges". The BOUND-TYPE TABLE below shows the bound types present in those problems that have bounds.

The problems below are sorted (according to the ASCII collating sequence) on their names. Unless problem characteristics suggest a more rational order, we suggest using this order for reporting results.

PROBLEM SUMMARY TABLE

Name	Rows	Cols	Nonzeros	Bytes	BR	Optimal Value
25FV47	822	1571	11127	70477		5.5018458883E+03
80BAU3B	2263	9799	29063	298952	B	9.8723216072E+05
ADLITTLE	57	97	465	3690		2.2549496316E+05
AFIRO	28	32	88	794		-4.6475314286E+02
AGG	489	163	2541	21865		-3.5991767287E+07
AGG2	517	302	4515	32552		-2.0239252356E+07
AGG3	517	302	4531	32570		1.0312115935E+07
BANDM	306	472	2659	19460		-1.5862801845E+02
BEACONFD	174	262	3476	17475		3.3592485807E+04
BLEND	75	83	521	3227		-3.0812149846E+01
BNL1	644	1175	6129	42473		1.9776292856E+03
BNL2	2325	3489	16124	127145		1.8112365404E+03
BOEING1	351	384	3865	25315	BR	-3.3521356751E+02
BOEING2	167	143	1339	8761	BR	-3.1501872802E+02
BORE3D	234	315	1525	13160	B	1.3730803942E+03
BRANDY	221	249	2150	14028		1.5185098965E+03
CAPRI	272	353	1786	15267	B	2.6900129138E+03
CYCLE	1904	2857	21322	166648	B	-5.2263930249E+00
CZPROB	930	3523	14173	92202	B	2.1851966989E+06
D2Q06C	2172	5167	35674	258038		1.2278423615E+05
D6CUBE	416	6184	43888	167633	B	3.1549166667E+02
DEGEN2	445	534	4449	24657		-1.4351780000E+03
DEGEN3	1504	1818	26230	130252		-9.8729400000E+02
DFL001	6072	12230	41873	353192	B	1.12664E+07 **
E226	224	282	2767	17749		-1.8751929066E+01
ETAMACRO	401	688	2489	21915	B	-7.5571521774E+02
FFFFF800	525	854	6235	39637		5.5567961165E+05
FINNIS	498	614	2714	23847	B	1.7279096547E+05
FIT1D	25	1026	14430	51734	B	-9.1463780924E+03

FIT1P	628	1677	10894	65116	B	9.1463780924E+03
FIT2D	26	10500	138018	482330	B	-6.8464293294E+04
FIT2P	3001	13525	60784	439794	B	6.8464293232E+04
FORPLAN	162	421	4916	25100	BR	-6.6421873953E+02
GANGES	1310	1681	7021	60191	B	-1.0958636356E+05
GFRD-PNC	617	1092	3467	24476	B	6.9022359995E+06
GREENBEA	2393	5405	31499	235711	B	-7.2462405908E+07
GREENBEB	2393	5405	31499	235739	B	-4.3021476065E+06
GROW15	301	645	5665	35041	B	-1.0687094129E+08
GROW22	441	946	8318	50789	B	-1.6083433648E+08
GROW7	141	301	2633	17043	B	-4.7787811815E+07
ISRAEL	175	142	2358	12109		-8.9664482186E+05
KB2	44	41	291	2526	B	-1.7499001299E+03
LOTFI	154	308	1086	6718		-2.5264706062E+01
MAROS	847	1443	10006	65906	B	-5.8063743701E+04
MAROS-R7	3137	9408	151120	4812587		1.4971851665E+06
MODSZK1	688	1620	4158	40908	B	3.2061972906E+02
NESM	663	2923	13988	117828	BR	1.4076073035E+07
PEROLD	626	1376	6026	47486	B	-9.3807580773E+03
PILOT	1442	3652	43220	278593	B	-5.5740430007E+02
PILOT.JA	941	1988	14706	97258	B	-6.1131344111E+03
PILOT.WE	723	2789	9218	79972	B	-2.7201027439E+06
PILOT4	411	1000	5145	40936	B	-2.5811392641E+03
PILOT87	2031	4883	73804	514192	B	3.0171072827E+02
PILOTNOV	976	2172	13129	89779	B	-4.4972761882E+03
QAP8	913	1632	8304	(see NOTES)		2.0350000000E+02
QAP12	3193	8856	44244	(see NOTES)		5.2289435056E+02
QAP15	6331	22275	110700	(see NOTES)		1.0409940410E+03
RECIPE	92	180	752	6210	B	-2.6661600000E+02
SC105	106	103	281	3307		-5.2202061212E+01
SC205	206	203	552	6380		-5.2202061212E+01
SC50A	51	48	131	1615		-6.4575077059E+01
SC50B	51	48	119	1567		-7.0000000000E+01
SCAGR25	472	500	2029	17406		-1.4753433061E+07
SCAGR7	130	140	553	4953		-2.3313892548E+06
SCFXM1	331	457	2612	19078		1.8416759028E+04
SCFXM2	661	914	5229	37079		3.6660261565E+04
SCFXM3	991	1371	7846	53828		5.4901254550E+04
SCORPION	389	358	1708	12186		1.8781248227E+03
SCRS8	491	1169	4029	36760		9.0429998619E+02
SCSD1	78	760	3148	17852		8.6666666743E+00
SCSD6	148	1350	5666	32161		5.0500000078E+01
SCSD8	398	2750	11334	65888		9.0499999993E+02
SCTAP1	301	480	2052	14970		1.4122500000E+03
SCTAP2	1091	1880	8124	57479		1.7248071429E+03
SCTAP3	1481	2480	10734	78688		1.4240000000E+03
SEBA	516	1028	4874	38627	BR	1.5711600000E+04
SHARE1B	118	225	1182	8380		-7.6589318579E+04
SHARE2B	97	79	730	4795		-4.1573224074E+02
SHELL	537	1775	4900	38049	B	1.2088253460E+09
SHIP04L	403	2118	8450	57203		1.7933245380E+06
SHIP04S	403	1458	5810	41257		1.7987147004E+06
SHIP08L	779	4283	17085	117083		1.9090552114E+06
SHIP08S	779	2387	9501	70093		1.9200982105E+06

SHIP12L	1152	5427	21597	146753		1.4701879193E+06
SHIP12S	1152	2763	10941	82527		1.4892361344E+06
SIERRA	1228	2036	9252	76627	B	1.5394362184E+07
STAIR	357	467	3857	27405	B	-2.5126695119E+02
STANDATA	360	1075	3038	26135	B	1.2576995000E+03
STANDGUB	362	1184	3147	27836	B	(see NOTES)
STANDMPS	468	1075	3686	29839	B	1.4060175000E+03
STOCFOR1	118	111	474	4247		-4.1131976219E+04
STOCFOR2	2158	2031	9492	79845		-3.9024408538E+04
STOCFOR3	16676	15695	74004	(see NOTES)		-3.9976661576E+04
TRUSS	1001	8806	36642	(see NOTES)		4.5881584719E+05
TUFF	334	587	4523	29439	B	2.9214776509E-01
VTP.BASE	199	203	914	8175	B	1.2983146246E+05
WOOD1P	245	2594	70216	328905		1.4429024116E+00
WOODW	1099	8405	37478	240063		1.3044763331E+00

BOUND-TYPE TABLE

80BAU3B	UP	LO	FX		
BOEING1	UP	LO			
BOEING2	UP	LO			
BORE3D	UP	LO	FX		
CAPRI	UP		FX	FR	
CYCLE	UP			FR	
CZPROB			FX		
DFL001	UP				
D6CUBE		LO			
ETAMACRO	UP	LO	FX		
FINNIS	UP	LO	FX		
FIT1D	UP				
FIT1P	UP				
FIT2D	UP				
FIT2P	UP				
FORPLAN	UP		FX		
GANGES	UP	LO			
GFRD-PNC	UP	LO			
GREENBEA	UP	LO	FX		
GREENBEB	UP	LO	FX	FR	
GROW15	UP				
GROW22	UP				
GROW7	UP				
KB2	UP				
MODSZK1				FR	
NESM	UP	LO	FX		
PEROLD	UP	LO	FX	FR	
PILOT	UP	LO	FX		
PILOT.JA	UP	LO	FX	FR	
PILOT.WE	UP	LO	FX	FR	
PILOT4	UP		FX	FR	PL
PILOTNOV	UP		FX		
RECIPE	UP	LO	FX		
SEBA	UP	LO			
SHELL	UP	LO	FX		

SIERRA	UP			
STAIR	UP		FX	FR
STANDATA	UP		FX	
STANDGUB	UP		FX	
STANDMPS	UP		FX	
TUFF	UP	LO	FX	FR
VTP.BASE	UP	LO	FX	FR

Several problems have an empty RHS section: BORE3D, CYCLE, GREENBEA, GREENBEB, KB2, RECIPE, and TUFF.

HEARTY THANKS go to the people who supplied the above problems. Michael Saunders provided 13 problems from the Systems Optimization Laboratory at Stanford University: ADLITTLE, AFIRO, BANDM, BEACONFD, BRANDY, CAPRI, E226, ETAMACRO, ISRAEL, PILOT, SHARE1B, SHARE2B, STAIR. Four problems are from a tape that John Reid sent me (David Gay) several years ago: 25FV47, CZPROB, FFFFF800, SHELL. Linus Schrage sent GANGES and SEBA. Bob Fourer supplied 44 problems: 80BAU3B, BORE3D, FIT1D, FIT1P, FIT2D, FIT2P, FORPLAN, GFRD-PNC, GREENBEA, GREENBEB, GROW15, GROW22, GROW7, NESM, PILOT.JA, PILOT.WE, PILOT4, PILOTNOV, RECIPE, SC205, SCAGR25, SCAGR7, SCFXM1, SCFXM2, SCFXM3, SCORPION, SCRS8, SCSD1, SCSD6, SCSD8, SCTAP1, SCTAP2, SCTAP3, SHIP04L, SHIP04S, SHIP08L, SHIP08S, SHIP12L, SHIP12S, SIERRA, STANDATA, STANDGUB, STANDMPS, VTP.BASE. Mauricio Resende provided AGG, AGG2, and AGG3, which were formulated by R. C. Leachman. Gus Gassmann contributed STOCFOR1, STOCFOR2, and STOCFOR3. Nick Gould supplied BLEND, BOEING1, BOEING2, FINNIS, PEROLD, SC105, SC50A, and SC50B from the Harwell collection of LP test problems. Vahid Lotfi submitted LOTFI. With the permission of Ketron, John Tomlin provided BNL1, BNL2, CYCLE, D2Q06C, DEGEN2, DEGEN3, KB2, TUFF, WOOD1P, and WOODW. At the request of Olvi Mangasarian, Rudy Setiono supplied the generator and description (both written by Michael Ferris) and data for TRUSS. Istvan Maros provided MAROS, MAROS-R7, and MODSZK1. Irv Lustig supplied PILOT87, which he obtained from John Stone. Marc Meketon submitted DFL001. Robert Hughes supplied D6CUBE. Problems QAP8, QAP12, and QAP15 are from a generator by Terri Johnson (communicated by a combination of Bob Bixby, Matt Saltzman, and Terri Johnson).

Thanks also go to Irv Lustig for helpful comments on this index file.

NOTES: we have omitted extra right-hand side vectors from BEACONFD, BRANDY, FFFFF800, ISRAEL; extra bound sets from GREENBEA, GREENBEB, GROW15, GROW22, GROW7, RECIPE; extra free rows from 80BAU3B, BOEING1, BORE3D, E226, FFFFF800, FINNIS, FORPLAN, GANGES, GREENBEA, GREENBEB, MAROS, PILOT, PILOT87, RECIPE, SCTAP1, SCTAP2, SCTAP3, SHARE2B, SHIP04L, SHIP04S, SHIP08L, SHIP08S, SHIP12L, SHIP12S; and explicit zeros from GROW15, GROW22, GROW7, NESM, SCORPION, SCRS8, SEBA, SIERRA, STAIR. We also negated the cost coefficients in BOEING1, BOEING2, DEGEN2, DEGEN3, ETAMACRO, FIT1D, FIT2D, GANGES, GROW15, GROW22, GROW7, LOTFI, MAROS, PILOT, PILOT.JA, PILOT.WE, PILOTNOV, SC105, SC50A, SC50B, STAIR. In their original form, these problems are usually maximized. In their modified form, all problems are to be minimized. (PILOT4 appeared to be a minimization problem already).

Problem 25FV47 is sometimes called BP or BP1, and FFFFF800 is sometimes called POWELL. Problems GREENBEA and GREENBEB differ only in their BOUNDS sections. The names shown above come mostly from the original NAME line; the optimal values are from MINOS version 5.3 (of Sept. 1988) running on a VAX with default options (except, as described below, for DFL001 and the QAP problems). [Earlier versions of this index file gave values from earlier versions of MINOS. Prior to 29 April 1987, this index file gave the optimal value from maximizing rather than minimizing PILOTNOV.]

Note that MINOS control parameters, such as SCALE, PARTIAL PRICE, FEASIBILITY TOLERANCE, OPTIMALITY TOLERANCE, and CRASH OPTION may affect the optimal value that MINOS reports (as may the version of MINOS, the computer, and even the compiler used).

This directory does not provide compressed MPS files for the QAP problems. Instead, source for Terri Johnson’s generator and input data for producing MPS files for QAP8, QAP12, and QAP15 appear in directory lp/generators/qap.

For discussion of some of the above test problems, including sparsity graphs and MINOS performance with and without scaling and partial pricing, see "An Analysis of an Available Set of Linear Programming Test Problems" by Irvin J. Lustig [Tech. Report SOL 87-11, Systems Optimization Laboratory, Dept. of Operations Research, Stanford Univ., Stanford, CA 94305-4022; a shorter version appears in Comput. Opns. Res. vol. 16, no. 2, pp. 173-184, 1989]. Be warned that the reproduction process may have dropped isolated nonzeros from graphs of the larger problems.

Bob Bixby reports that the CPLEX solver (running on a Sparc station) finds slightly different optimal values for some of the problems. On a MIPS processor, MINOS version 5.3 (with crash and scaling of December 1989) also finds different optimal values for some of the problems. The following table shows the values that differ from those shown above. (Whether CPLEX finds different values on the recently added problems remains to be seen.)

Problem	CPLEX(Sparc)	MINOS(MIPS)
25FV47		5.5018467791E+03
80BAU3B	9.8722419241E+05	9.8722952818E+05
BNL1	1.9776295615E+03	1.9776293385E+03
D2Q06C		1.2278423521E+05
DFL001	1.1266396047E+07	**
ETAMACRO	-7.5571523337E+02	-7.5571522100E+02
FIT2D		-6.8464293232E+04
FFFFF800	5.5567956482E+05	5.5567958085E+05
FORPLAN	-6.6421896127E+02	
GANGES	-1.0958573613E+05	-1.0958577038E+05
GREENBEA	-7.2555248130E+07	
GREENBEB	-4.3022602612E+06	-4.3021537702E+06
NESM	1.4076036488E+07	1.4076065292E+07
PEROLD	-9.3807552782E+03	-9.3807553661E+03

PILOT	-5.5748972928E+02	-5.5741215293E+02
PILOT.JA	-6.1131364656E+03	-6.1131349867E+03
PILOT.WE	-2.7201075328E+06	-2.7201042967E+06
PILOT4	-2.5811392589E+03	-2.5811392624E+03
PILOT87		3.0171074161E+02
SCAGR7	-2.3313898243E+06	-2.3313897524E+06
SCRS8	9.0429695380E+02	9.0429695380E+02
SCSD6	5.0500000077E+01	
SIERRA		1.5394364186E+07
STOCFOR3	-3.9976785944E+04	-3.9976776417E+04

The above CPLEX and MINOS results were both obtained using double-precision IEEE (binary) arithmetic, i.e., arithmetic of precision similar to the VAX double precision with which the MINOS 5.3 results in the PROBLEM SUMMARY TABLE were computed.

The old problem GUB was the same as CZPROB (except for the NAME line) and hence is withdrawn.

STANDGUB includes GUB markers; with these lines removed (lines in the expanded MPS file that contain primes, i.e., that mention the rows MINOS does not understand the GUB markers, so we cannot report an optimal value from MINOS for STANDGUB. STANDMPS amounts to STANDGUB with the GUB constraints as explicit constraints.

STOCFOR1,2,3 are stochastic forestry problems from Gus Gassmann. To quote Gus, "All of them are seven-period descriptions of a forestry problem with a random occurrence of forest fires, and the size varies according to the number of realizations you use in each period." STOCFOR1 "is the deterministic version, STOCFOR2 has 2 realizations each in periods 2 to 7, and the monster STOCFOR3 has 4,4,4,2,2, and 2 realizations, respectively." The compressed form of STOCFOR3 would be 652846 bytes long, so requesting STOCFOR3 will instead get you a bundle of about 174 kilobytes that includes source for Gus’s program, the data files for generating STOCFOR3 and a summary of "A Standard Input Format for Multistage Stochastic Linear Programs" by J.R. Birge, M.A.H. Dempster, H.I. Gassmann, E.A. Gunn, A.J. King, and S.W. Wallace [COAL Newsletter No. 17 (Dec. 1987), pp. 1-19]. Data files are also included for generating versions of STOCFOR1,2 that have more decimal places than the versions in lp/data.

Concerning the problems he supplied, Nick Gould says that BLEND "is is a variant of the [oil refinery] problem in Murtagh’s book (the coefficients are different) which I understand John Reid obtained from the people at NPL (Gill and Murray?); they were also the original sources for the SC problems"; BOEING1 and BOEING2 "have to do with flap settings on aircraft for economical operations"; PEROLD "is another Pilot model (Pilot1)"; and FINNIS "is from Mike Finnis at Harwell, a model for the selection of alternative fuel types."

BOEING1 and BOEING2 were originally mixed-integer programming problems. The COLUMNS section of BOEING1 had

```
    INTBEG    ’MARKER’                 ’INTORG’
```

between the coefficients for columns GRDTIMN6 and N1001AC1, and that

BOEING2 had such a line between columns GRDTIMN4 and N1003AC1. Both had

```
    INTFIN    ’MARKER’                 ’INTEND’
```

just before the start of the ROWS section. These ’MARKER’ lines have been removed. These problems also had a few rows defined as linear combinations of other rows. These rows are now given explicitly, since the compression/expansion programs do not understand D lines in the ROWS section.

LOTFI, says Vahid Lotfi, "involves audit staff scheduling. This problem is semi real world and we have used it in a study, the results of which are to appear in Decision Sciences (Fall 1990). The detailed description of the problem is also in the paper. The problem is actually an MOLP with seven objectives, the first is maximization and the other six are minimization. The version that I am sending has the aggregated objective (i.e., z1-z2-z3-z4-z5-z6-z7)."

On the problems supplied by John Tomlin, MINOS 5.3 reports that about 10% to 57% of its steps are degenerate:

Name	Steps	Degen	Percent
BNL1	1614	169	10.47
BNL2	4914	906	18.44
CYCLE	3156	1485	47.05
D2Q06C	42417	4223	9.96
DEGEN2	1075	610	56.74
DEGEN3	6283	3299	52.51
KB2	82	29	35.37
TUFF	745	345	46.31
WOOD1P	1059	471	44.48
WOODW	4147	1604	38.68

Concerning PILOT87, Irv Lustig says, "PILOT87 is considered (by John Stone, at least) to be harder than PILOT because of the bad scaling in the numerics."

Requesting TRUSS will get you a bundle of Fortran source and data for generating an MPS file for TRUSS, a problem of minimizing the weight of a certain structure. The bundle also includes a description of the problem.

DFL001, says Marc Meketon, "is a ’real-world’ airline schedule planning (fleet assignment) problem. This LP was preprocessed by a modified version of the KORBX(r) System preprocessor. The problem reduced in size (rows, columns, non-zeros) significantly. The row and columns were randomly sorted and renamed, and a fixed adjustment to the objective function was eliminated. The name of the problem is derived from the initials of the person who created it."

Of D6CUBE, Robert Hughes says, "Mike Anderson and I are working on the problem of finding the minimum cardinality of triangulations of the 6-dimensional cube. The optimal objective value of the problem I sent you provides a lower bound for the cardinalities of all triangulations which contain a certain simplex of volume 8/6! and which contains the centroid of the 6-cube in its interior. The linear programming problem is not easily described."

Concerning the problems he submitted, Istvan Maros says that MAROS is an industrial production/allocation model about which "the customer does not want to reveal the exact meaning". MAROS-R7 is "an interesting real-life LP problem which appeared hard to some solvers." It "is an image restoration problem done via a goal programming approach. It is structured, namely, its first section is a band matrix with the dominating number of nonzeros, while the second section is also a band matrix with bandwidth equals 2 and coefficients +1, -1. The problem is a representative of a family of problems in which the number of rows and the bandwidth of the first section can vary. This one is a medium size problem from the family. MAROS-R7 became available in cooperation with Roni Levkovitz and Carison Tong." MODSZK1 is a "real-life problem" that is "very degenerate" and on which a dual simplex algorithm "may require up to 10 times" fewer iterations than a primal simplex algorithm. It "is a multi-sector economic planning model (a kind of an input/output model in economy)" and "is an old problem of mine and it is not easy to recall more."

** On an IEEE-arithmetic machine (an SGI 4D/380S), I (dmg) succeeded in getting MINOS 5.3 to report optimal objective values, 1.1261702419E+07 and 1.1249281428E+07, for DFL001 only by starting with LOAD files derived from the solution obtained on the same machine by Bob Vanderbei’s ALPO (an interior-point code); starting from one of the resulting "optimal" bases, MINOS ran 23914 iterations on a VAX before reporting an optimal value of 1.1253287141E+07. When started from the same LOAD file used on the SGI machine, MINOS on the VAX reported an optimal value of 1.1255107696E+07. Changing the FEASIBILITY TOLERANCE to 1.E-10 (from its default of 1.E-6) led MINOS on the SGI machine to report "optimal" values of 1.1266408461E+07 and 1.1266402835E+07. This clearly is a problem where the FEASIBILITY TOLERANCE, initial basis, and floating-point arithmetic strongly affect the "optimal" solution that MINOS reports. On the SGI machine, ALPO with SPLIT 3 found

```
 primal:  obj value =  1.126639607e+07      FEASIBLE   ( 2.79e-09 )
 dual:    obj value =  1.126639604e+07      FEASIBLE   ( 1.39e-16 )
```

Bob Bixby reports the following about his experience solving DFL001 with CPLEX:

> First, the value for the objective function that I get running defaults is 1.1266396047e+07, with the following residuals:

```
  Max. unscaled (scaled) bound        infeas.: 4.61853e-14 (2.30926e-14)
  Max. unscaled (scaled) reduced-cost infeas.: 6.40748e-08 (6.40748e-08)
  Max. unscaled (scaled) Ax-b          resid.: 4.28546e-14 (4.28546e-14)
  Max. unscaled (scaled) c_B-B’pi      resid.: 8.00937e-08 (8.00937e-08)
```

> The L_infinity condition number of the (scaled) optimal basis is 213737. I got exactly the same objective value solving the problem in several different ways. I played a bit trying to get a better reduced-cost infeasibility, but that seems hopeless (if not pointless) given the c-Bpi residuals.
>
> Just as an aside, this problem exhibits very interesting behavior when solved using a simplex method. I ran reduced-cost pricing on it in phase I, with the result that it took 465810 iterations to get

feasible. Running the default CPLEX pricing scheme, the entire problem solved in 94337 iterations (33059 in phase I) on a Sparcstation. Steepest-edge pricing (and a different scaling) took 25803 iterations. This is a nasty problem.

Notes from Michael Saunders describing experience with MINOS on the problems he provided are available via the netlib request

```
send minos from lp/data
```

Sources for the problems from Bob Fourer:

BORE3D, RECIPE, SHIP04L, SHIP04S, SHIP08L, SHIP08S, SHIP12L, SHIP12S, STANDATA, STANDGUB, STANDMPS, VTP.BASE: consulting.

80BAU3B: W. Kurator and Harvey Greenberg, Energy Information Administration (Greenberg is now at the Univ. of Colorado - Denver).

GREENBEA, GREENBEB: a large refinery model; see the book "A Model-Management Framework for Mathematical Programming" by Kenneth H. Palmer et al. (John Wiley & Sons, New York, 1984).

GROW15, GROW22, GROW7: R. Fourer, "Solving Staircase Linear Programs by the Simplex Method, 2: Pricing", Math. Prog. 25 (1983), pp. 251-292.

PILOT.JA, PILOT.WE, PILOT4, PILOTNOV: SOL, Stanford University.

GFRD-PNC, SIERRA: R. Helgason, J. Kennington, and P. Wong, "An Application of Network Programming for National Forest Planning", Technical Report OR 81006, Dept. of Operations Research, Southern Methodist University.

SC205, SCAGR25, SCAGR7, SCFXM1, SCFXM2, SCFXM3, SCORPION, SCRS8, SCSD1, SCSD6, SCSD8, SCTAP1, SCTAP2, SCTAP3: J.K. Ho and E. Loute, "A Set of Staircase Linear Programming Test Problems", Math. Prog. 20 (1981), pp. 245-250.

NESM: Gerald Brown, Naval Postgraduate School.

FORPLAN: John Mulvey, Princeton.

FIT1D, FIT1P, FIT2D, FIT2P: Bob Fourer himself.

Concerning FIT1D, FIT1P, FIT2D, FIT2P, Bob Fourer says

> The pairs FIT1P/FIT1D and FIT2P/FIT2D are primal and dual versions of the same two problems [except that we have negated the cost coefficients of the dual problems so all are minimization problems]. They originate from a model for fitting linear inequalities to data, by minimization of a sum of piecewise-linear penalties. The FIT1 problems are based on 627 data points and 2-3 pieces per primal pl penalty term. The FIT2 problems are based on 3000 data points (from a different sample altogether) and 4-5 pieces per pl term.

To get C source for the compression program, issue the netlib request

```
send mpc.src from lp/data
```

Contributions are welcome, either problems in MPS format or source code for problem generators. Send questions, comments, contributions to

David M. Gay
Bell Laboratories, Lucent Technologies

600 Mountain Avenue, room 2C-463
Murray Hill, NJ 07974-2070
U.S.A.
phone (908) 582-5623; FAX (908) 582-5857
E-mail dmg@research.bell-labs.com

Cross reference: Eberhard Kranich’s extensive bibliography on interior-point methods is available from netlib. For details, ask netlib to

```
send index from bib
```

Change log...

1 June 1987: mpc.src added.

6 May 1988: GREENBEA, GREENBEB, AGG, AGG2, AGG3 added.

25 June 1988: STOCFOR1,2 added

16 Jan. 1989: STOCFOR3 added; bound and range information added to index file; MINOS 5.3 optimal values inserted.

23 Jan. 1989: correction to bound-handling portion of STOCFOR3 source code. This does not affect STOCFOR3 itself, but is relevant to other uses of this Fortran code.

6 April 1989: BLEND BOEING1 BOEING2 FINNIS PEROLD SC105 SC50A SC50B added.

27 June 1989: CYCLE KB2 LOTFI TUFF WOOD1P WOODW added.

30 Oct. 1989: BNL1 BNL2 D2Q06C DEGEN2 DEGEN3 added.

30 Nov. 1989: options -s and -S added to emps.c so you can request several problems at once and split them into files named by the problem name (in upper case with -S or in lower case with -s). For use with these new options, the NAME line of several problems has now been modified so that the first word after "NAME" gives the name specified above for the problem. Now all compressed MPS files have this property. The problems whose NAME line was thus modified are BLEND, BOEING1, FINNIS, FORPLAN, PEROLD, PILOT, PILOTNOV, STANDGUB, STANDMPS, STOCFOR1, and STOCFOR2.

22 Jan. 1990: all material described here made available by anonymous ftp from research.att.com .

31 Jan. 1990: FIT1D, FIT1P, FIT2D, FIT2P added.

8 Feb. 1990: emps.c, emps.f modified to quietly ignore extra lines at the end of a compressed MPS file (e.g., those that mailers add).

15 Feb. 1990: added table of optimal values reported by Bob Bixby.

26 Feb. 1990: TRUSS added.

30 Apr. 1990: ascii (table of ASCII codes) added; MINOS(MIPS) optimal values added to this index file.

15 June 1990: MAROS and PILOT87 added.

11 Oct. 1990: DFL001 added.

9 Jan. 1991: Bixby’s remarks about DFL001 added to index.

6 June 1991: emps.c and emps.f adjusted to pass "mystery lines" through, for possible use in conveying other problem information (in connection with mpc -m). [For years emps.c has had this ability; today’s change fixes a bug with mystery lines just before ENDATA.]

4 Sept. 1991: "Kennington" problems made available by ftp from netlib.

21 Oct. 1991: minor cleanups...

1. BOEING1: remove duplicate upper bounds for columns N1019AC3 and N1019AC4.
2. PILOT: remove 8 duplicate right-hand side values for row BTRB01.

3. PILOT87: remove lower bound of 49.5 on U[OG]ST0[12], which are subsequently fixed at 99 (UOST[12]) or 65.4.

2 May 1992: emps.c ANSIfied (with #ifdef KR_headers lines for old-style C compilers); new option -b changes blanks within names to underscores (and changes blank RHS names to RHS, etc.) -- for awk scripts and other programs that assume no blanks in names.

4 Feb. 1993: STOCFOR3 updated. STOCFOR3 and the other problems you can generate with the data in the stocfor3 bundle are the same numerically as before (but with different row and column labels). The update (courtesy of Gus Gassmann) fixes some bugs in other uses of the generator and expands your options in using the generator. The previous version is now stocfor3.old.

26 March 1993: D6CUBE added.

17 Jan. 1994: MAROS-R7 and MODSZK1 added.

12 April 1996: QAP8, QAP12, QAP15 added to result table; directory lp/generators/qap added for generating these problems.