A Test Set of Real-world Mixed
Integer Programming Problems[1]

Robert E. Bixby

E. Andrew Boyd

Ronni R. Indovina

November, 1991
(revised January 1992)

TR91-38

MIPLIB:

A Test Set of Real-world Mixed Integer Programming Problems

Robert E. Bixby   E. Andrew Boyd   Ronni R. Indovina

1   Introduction

Integer programming problems, both pure and mixed, have traditionally been considered very difficult. Indeed, there are currently no guaranteed polynomial-time algorithms available for general integer programming. Within the last ten years, however, major advances in preprocessing methods and cutting plane techniques have provided more insight into what makes these problems so difficult. These breakthroughs have sparked much interest and have led to new research in the field of integer programming.

Integer programs are problems that consist of maximizing or minimizing a linear function on a linearly defined constraint set with some or all of the variables restricted to take on integer values. If only some of the variables have this restriction, the problem is a mixed integer program. These problems have applications in many diverse areas. Among these applications are: fiber optic network design, modeling the acquisition, manufacturing and distribution of natural gas, machine and job scheduling, vehicle routing, drainage system design, generator scheduling, airline crew and flight scheduling, modeling automobile acquisition and selling. This wide applicability of integer programming presents a real need for efficient methods for solving both pure and mixed integer programming problems. One hindrance to research in this area has been a lack of general availability of real models.

In 1985, David M. Gay introduced a library of linear programming problems [4] that is electronically accessible through NETLIB [3], and, in 1990, Gerhard Reinelt in-

troduced TSPLIB, an electronically accessible library of Traveling Salesman problems [6]. The availability of real models through both of these libraries has sparked much computational research on linear programming and traveling salesman problems. In fact, since the creation of TSPLIB, 13 problems that were previously considered "unsolvable" have been solved to provable optimality. Given that these two libraries have had such an impact on their respective fields, we felt that a similar effort was needed in the area of integer programming. MIPLIB is an electronic library of both pure and mixed integer programs created in an effort to make real models available to researchers working on such problems.

The types of problems included in the MIPLIB set are varied. For every application of integer programming mentioned above, there is at least one model of that type in the library. In addition, the level of difficulty of the models in the library is equally as varied. The **bell** series of problems is a set of relatively small but incredibly difficult models, arising from the design of fiber optic networks, which are characterized by a large number of general integer variables. The only method that has been able to solve these is a new code developed by Cook, Rutherford, Scarf, and Shallcross [1] based on the basis reduction technique of Scarf and Lovasz [5]. The **pxxxx** problems are also difficult problems for a standard branch and bound approach. A preprocessing and cutting plane technique developed by Crowder, Johnson, and Padberg [2] has been very successful in dealing with these models. The **airxx** models are all set partitioning problems that arise in airline crew scheduling. These airline models are generally very difficult, and in particular, no optimal solution has yet been proven for **air05**. The **stein** models, while all very small, are not as harmless as they appear, particularly **stein45**. Another very difficult model is **rentacar**, which models automobile acquisition and selling. The basis matrix suddenly becomes very dense during the solution process, causing basis factorizations to slow down and potential memory difficulties. The problem **misc07** also seems to be very difficult.

2   Index

The following is a two-part index of the models currently contained in the library. The first part contains statistics for each problem, and the second part contains information regarding the origins of each problem. In PART A, the first column, NAME,

contains the name of the model. The next two columns, ROWS and COLS, contain the number of rows in the constraint matrix, not including free rows, and the number of variables in the problem, respectively. The column INT tells the number of variables that are restricted to integer values, and the column 0/1 tells how many of these integer variables are binary. The last two columns report the optimal solution value and the optimal solution value with the integrality restrictions relaxed, respectively. In the column INT SOLUTION however, the solutions reported are followed by qualifiers. The qualifier (opt) indicates that the reported solution is indeed integer optimal, while (not opt) indicates that the reported solution is not integer optimal. The qualifier (opt)* indicates that the solution is reportedly integer optimal, as stated by various sources, but it has not yet been verified by us.

The first column of PART B is, again, NAME. The second column, ORIGINATOR, gives the name of the person or institution with which the problem originated. The next column, FORMULATOR, gives the name of the person or organization responsible for formulating the model, and the last column, DONATOR, gives the name of the person or institution who contributed the problem.

NAME	ROWS	COLS	INT	0/1	INT SOLUTION	LP SOLUTION
air01	23	771	771	ALL	6796 (opt)	6743.0
air02	50	6774	6774	ALL	7810 (opt)	7640.0
air03	124	10757	10757	ALL	340160 (opt)	338864.25
air04	823	8904	8904	ALL	56138 (opt)*	55535.436
air05	426	7195	7195	ALL	26402 (not opt)	25877.609
air06	825	8627	8627	ALL	49649 (opt)	49616.364
bell3a	123	133	71	39	878430.32 (opt)*	862578.64
bell3b	123	133	71	39	11786160.62 (opt)*	11404143.89
bell4	105	117	64	34	18541484.20 (opt)*	17984775.91
bell5	91	104	58	30	8966406.49 (opt)*	8608417.95
bm23	20	27	27	ALL	34 (opt)	20.57
cracpb1	143	572	572	ALL	22199 (opt)	22199.0
diamond	4	2	2	ALL	integer infeasible	-1.0
dsbmip	1182	1886	192	160	-305.198 (opt)	-305.198
egout	98	141	55	ALL	568.101 (opt)*	149.589
enigma	21	100	100	ALL	0.0 (opt)	0.0
fixnet3	478	878	378	ALL	51973 (opt)*	40717.018
flugpl	18	18	11	0	1201500 (opt)	1167185.73
gen	780	870	150	144	112313 (opt)*	112130.0
khb05250	101	1350	24	ALL	106940226 (opt)	95919464.0
l152lav	97	1989	1989	ALL	4750 (not opt)	4656.36
lp4l	85	1086	1086	ALL	2967 (opt)	2942.5
lseu	28	89	89	ALL	1120 (opt)	834.68
modglob	291	422	98	ALL	20740508 (opt)*	20430947.0
misc01	54	83	82	ALL	563.5 (opt)	57.0
misc02	39	59	58	ALL	1690 (opt)	1010.0
misc03	96	160	159	ALL	3360 (opt)	1910.0
misc04	1725	4897	30	ALL	2666.699 (opt)	2656.42
misc05	300	136	74	ALL	2984.5 (opt)	2930.9
misc06	820	1808	112	ALL	12850.8607 (opt)	12841.69
misc07	212	260	259	ALL	2810 (not opt)	1415.0
mod008	6	319	319	ALL	307 (opt)	290.93
mod010	146	2655	2655	ALL	6548 (opt)	6532.08
mod011	4480	10958	96	ALL	-54558535 (opt)*	-62121982.552
mod013	62	96	48	ALL	280.95 (opt)	256.02
noswot	182	128	100	75	-43 (opt)	-43.0
p0033	16	33	33	ALL	3089 (opt)	2520.57
p0040	23	40	40	ALL	62027 (opt)	61796.55
p0201	133	201	201	ALL	7615 (opt)	6875.0
p0282	241	282	282	ALL	258411 (opt)	176867.50
p0291	252	291	291	ALL	5223.7490 (opt)	1705.13
p0548	176	548	548	ALL	8691 (opt)	315.29
p2756	755	2756	2756	ALL	3124 (opt)*	2688.75
pipex	25	48	48	ALL	788.263 (opt)	773.751
rentacar	6803	9557	55	ALL	30356761 (opt)*	28806137.644
rgn	24	180	100	ALL	82.1999 (opt)	48.7999
sample2	45	67	21	ALL	375 (opt)	247.0
sentoy	30	60	60	ALL	-7772 (opt)	-7839.278
stein15	36	15	15	ALL	9 (opt)	7.0
stein27	118	27	27	ALL	18 (opt)	13.0
stein45	331	45	45	ALL	30 (opt)*	22.0
stein9	13	9	9	ALL	5 (opt)	4.0
vpm1	234	378	168	ALL	20 (opt)*	15.4167

PART B : ORIGINS

NAME	ORIGINATOR	FORMULATOR	DONATOR
air01			Greg Astfalk
air02			Greg Astfalk
air03			Greg Astfalk
air04			Greg Astfalk
air05			Greg Astfalk
air06			Greg Astfalk
bell3a	William Cook	William Cook	William Cook
bell3b	William Cook	William Cook	William Cook
bell4	William Cook	William Cook	William Cook
bell5	William Cook	William Cook	William Cook
bm23	B. Bouvier, G. Messoumain	Harlan Crowder	E. Andrew Boyd
cracpb1	Harlan Crowder	Harlan Crowder	E. Andrew Boyd
diamond	John W. Gregory	John W. Gregory	E. Andrew Boyd
dsbmip			John J. Forrest
egout		Laurence A. Wolsey	Martin W. P. Savelsbergh
enigma	Harlan Crowder	Harlan Crowder	E. Andrew Boyd
fixnet3		Laurence A. Wolsey	Martin W. P. Savelsbergh
flugpl	Harvey M. Wagner	John W. Gregory	E. Andrew Boyd
gen		Laurence A. Wolsey	Martin W. P. Savelsbergh
khb05250		Laurence A. Wolsey	Martin W. P. Savelsbergh
l152lav	Harlan Crowder	Harlan Crowder	John W. Gregory
lp4l	Harlan Crowder	Harlan Crowder	E. Andrew Boyd
lseu	C. E. Lemke, K. Spielberg	Ellis L. Johnson, Uwe H. Suhl	John J. Forrest
modglob		Laurence A. Wolsey	Martin W. P. Savelsbergh
misc01			Greg Astfalk
misc02			Greg Astfalk
misc03			Greg Astfalk
misc04			Greg Astfalk
misc05			Greg Astfalk
misc06			Greg Astfalk
misc07			Greg Astfalk
mod008	IBM France	IBM France	John J. Forrest
mod010	IBM Yorktown Hts	IBM Yorktown Hts	John J. Forrest
mod011	Uwe H. Suhl	Uwe H. Suhl	John J. Forrest
mod013	Laurence A. Wolsey	Laurence A. Wolsey	John J. Forrest
noswot		Linus E. Schrage	John W. Gregory
p0033	CJP set		E. Andrew Boyd
p0040	CJP set		E. Andrew Boyd
p0201	CJP set		E. Andrew Boyd
p0282	CJP set		E. Andrew Boyd
p0291	CJP set		E. Andrew Boyd
p0548	CJP set	Ellis L. Johnson	E. Andrew Boyd
p2756	CJP set	Ellis L. Johnson	E. Andrew Boyd
pipex		Laurence A. Wolsey	Martin W. P. Savelsbergh
rentacar			John J. Forrest
rgn		Laurence A. Wolsey	Martin W. P. Savelsbergh
sample2		Laurence A. Wolsey	Martin W. P. Savelsbergh
sentoy		Senju-Toyoda	Linus E. Schrage
stein15	George L. Nemhauser	John W. Gregory	E. Andrew Boyd
stein27	George L. Nemhauser	John W. Gregory	E. Andrew Boyd
stein45	George L. Nemhauser	John W. Gregory	E. Andrew Boyd
stein9	George L. Nemhauser	John W. Gregory	E. Andrew Boyd
vpm1		Laurence A. Wolsey	Martin W. P. Savelsbergh

5

3 MIPLIB Specifics

Each problem in MIPLIB is stored in a separate file. Each file begins with a header section, followed by the actual problem data, which is stored in MPS format[1]. Each line of the header section has an * in column one, and no lines are skipped between the header and the data. The following is a description of the header section:

*NAME: 1 line. This section contains the name of the problem.

*ROWS: 1 line. This section contains the number of constraints in the problem, not including free rows.

*COLUMNS: 1 line. This section contains the total number of variables in the problem.

*INTEGER: 1 line. This section contains the number of variables that are restricted to integer values.

*NONZERO: 1 line. This section contains the number of nonzeroes in the constraint matrix, not including free rows.

*BEST SOLN: 1 line. This section contains the best known solution to the problem. If the solution given is optimal, then the indicator (opt) will appear behind the solution. It is also possible that either "infeasible" or "integer infeasible" will appear here, which are self-explanatory terms.

*LP SOLN: 1 line. This section contains the optimal solution to the linear relaxation of the problem.

*SOURCE: 3 lines. The first line contains the name of the person (or organization) responsible for originating the problem. The second line contains the name of the person (or organization) responsible for formulating the problem as a MIP. Following this, in parentheses, is the name of the institution with which the person is affiliated. The third line contains the name of the person who donated the problem, along with the name of his or her affiliated institution in

---

[1]For detailed documentation on MPS format, see an IBM MPSX manual.

6

parentheses. If any line is blank, it is because that information is not available. (Each line will have an asterisk in column 1.)

*APPLICATION: 1 line. This section contains a brief description of the application from which the problem arose. It is possible that "unknown" will appear here, which is self-explanatory.

*COMMENTS: 3 lines. This section will contain other miscellaneous information about the problem. Things that may appear here are data pertaining to the number of binary integer variables in the problem, constraint classifications, comments as to the difficulty of the problem, etc. Also, if the optimal solution reported by the contributor could not be verified, the system with which this solution was obtained is reported here. (Each line will have an asterisk in column 1.)

The section labels appear exactly as they do here, and no lines are skipped between sections. All data for the various sections begin in column 16.

4 Accessing MIPLIB

The library is accessible through SOFTLIB, an electronic software distribution system available through Internet. By sending e-mail to softlib@rice.edu with a message body reading "send catalogue," you will receive via electronic mail a list of software codes and libraries currently available for public access through the system, along with specific instructions for accessing each one.

In addition to making this library available for public use, we wish to invite comments and suggestions regarding the library. Contributions of new problems to the library would also be greatly appreciated, as well as any new solutions that have been proven optimal.

To make submissions or comments, please contact Dr. Robert E. Bixby or Dr. E. Andrew Boyd, Department of Mathematical Sciences, Rice University, Houston, TX 77251, or send e-mail to either bixby@rice.edu or boyd@rice.edu.

7

References

[1] William Cook, Thomas Rutherford, Herbert Scarf, and David Shallcross, "An implementation of the generalized basis reduction algorithm for integer programming," Cowles Foundation Discussion Paper No. 990, 1991, Yale University, New Haven, Connecticut.

[2] H. Crowder, E. L. Johnson, and M. Padberg, "Solving large-scale zero-one linear programming problems," *Operations Research* 31 (1983), 803-834.

[3] J. J. Dongarra, and E. Grosse, "Distribution of mathematical software by electronic mail," *Communications of the ACM*, Vol. 30, No. 5, 1987.

[4] David M. Gay, "Electronic mail distribution of linear programming test problems," *COAL* newsletter #13, December 1985, pp. 10-12.

[5] L. Lovász, and H. E. Scarf, "The generalized basis reduction algorithm," Cowles Foundation Discussion Paper No. 946, 1990, Yale University, New Haven, Connecticut.

[6] Gerhard Reinelt, "TSPLIB—A traveling salesman problem library," *ORSA Journal on Computing*, Vol. 30, No. 4, 1991.