Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

TOBIAS ACHTERBERG
THORSTEN KOCH
ALEXANDER MARTIN

MIPLIB 2003

http://miplib.zib.de

ZIB-Report 05-28 (May 2005)

MIPLIB 2003

Tobias Achterberg* Thorsten Koch† Alexander Martin‡

May 2005

Abstract

This paper reports on the fourth version of the Mixed Integer Programming Library. Since MIPLIB is to provide a concise set of challenging problems, it became necessary to purge instances that became too easy. We present an overview of the 27 new problems and statistical data for all 60 instances.

Keywords: Mixed Integer Programming, Problem Instances, IP, MIP, MIPLIB

1 Introduction

A popular strategy to lend weight to important and timeless facts is to make quotations that seem to speak of the present, but are stated a long way back in the past. One famous example is from the Greek philosopher Socrates, 470-399 BC:

The children now love luxury; they have bad manners, contempt for authority; they allow disrespect for elders and love chatter in place of exercise. (...) They contradict their parents, chatter before company, gobble up dainties at the table, cross their legs, and tyrannize their teachers.

Here is another not as famous but nevertheless timeless and important quotation:

In response to the needs of researchers for access to challenging mixed integer programs, Bixby et al. [2] created MIPLIB, an electronically available library of both pure and mixed integer programs, most of which arise from real-world applications.

Since its introduction, MIPLIB has become a standard test set for comparing the performance of mixed integer optimization codes. Its availability has provided an important stimulus for researchers in this very active area. As technology has progressed, however, there have been significant improvements in state-of-the-art optimizers and computing machinery. Consequently, several instances have become too easy, and a need has emerged for more difficult instances. Also, it has been observed that certain types of problems are overrepresented in MIPLIB and others underrepresented. These considerations have prompted the present update.

Since mixed integer programming is such an active research area, and the performance of optimizers keeps improving, we anticipate that this update will not be the last. Subsequent updates are planned on a yearly basis. We encourage both researchers and practitioners in integer programming to submit real-world instances for consideration and possible inclusion in MIPLIB.

*Zuse Institute Berlin
†Zuse Institute Berlin, Takustr. 7, Berlin, koch@zib.de, corresponding author
‡Technische Universtität Darmstadt

This quotation is taken from the introduction of Bixby, McZeal, Ceria, and Savelsbergh [3] and serves as an excellent introduction to this note as well. More than seven years have passed since the last update of the MIPLIB. And again improvements in state-of-the-art optimizers as well as in computing machinery have made several instances too easy to be of challenging interest. Therefore we purged the library of those instances and filled the free slots with new interesting candidates.

This paper describes the 2003 MIPLIB update. We provide information on the new instances including information on the coefficient matrices, the types of constraints and variables. The library is available online at http://miplib.zib.de, where we have collected further information such as references to papers using the MIPLIB as test set for their algorithms.

We have divided the instances of the MIPLIB into three classes: *easy* are those instances that can be solved by a commercial MIP solver within an hour. We classify an instance as *hard* if it could be solved at all, even if specialized algorithms have to be used. Finally, all instances without known optimal solution are declared to be *open*. This classification is to be kept up-to-date, and we ask for notification whenever an optimal solution is determined for an *open* instance.

In the discussions preceding this update concerns were expressed about a possible lack of *easy* problems. We therefore added a mirror of the "old" MIPLIB 3.0 to the new website. It is interesting to note that providing challenging problems has indeed an impact on the MIP community: since the first announcement of the new MIPLIB, 11 instances already changed their status due to progress in MIP solver performance. This includes five new instances which have already become *easy*.

Another question is whether to provide "best known" bounds on the *open* instances. We decided not to maintain such bounds for each instance explicitly, as it is impossible to keep them up-to-date. Instead, we installed a bulletin board where everybody may post new results.

2 Problem Instances

Referring to the quote above, the focus of the MIPLIB has always been to provide a concise set of challenging problems. For this reason we did not add new instances only, but also retired some old ones.

2.1 Removed Instances

To decide which instances to remove, we used CPLEX version 8 with the following settings: no preprocessing, no probing, no cut generation, no heuristics, a relative mipgap of zero, and pseudocost branching.

We discarded all instances that were solved within a minute on a fast PC: air03, bell3a, bell5, blend2, dcmulti, dsbmip, egout, enigma, flugpl, gen, gesa3, gesa3-o, gt2, khb05250, l152lav, lseu, misc03, misc06, mitre, mod008, mod010, p0033, p0201, p0282, p0548, qnet1, qnet1-o, rentacar, rgn, stein27, stein45, and vpm1. A more recent test using CPLEX 9.03 on a 3.4 GHz PC with the above settings shows that the total time to solve these instances is 101 seconds where more than half of it is spent on vpm1.

Name	Description	Ref.
a1c1s1	Lot sizing instance	[22]
aflow30a, aflow40b	Arborescence flow problems on a graph with 30 and 40 nodes and edge densities 0.5 and 0.9	[19]
atlanta-ip, msc98-ip	Min-cost network dimensioning problems with a finite set of link capacities for each bidirected link, unsplittable shortest path routing, path restoration for single node failures, and routing path length restrictions	[4]
disctom	Instance originating from discrete tomography	[11]
ds	Set partitioning problem from public transport service planning	[6]
glass4	Nesting instance	[17]
liu	Floorplan and placement problem in the physical design of VLSI circuits	
manna81	Stable multi sets on a graph. A maximum clique instance was complemented, and bounds on nodes and edges were set to form a stable multi sets instance	[15]
momentum1, momentum2, momentum3	Snapshot based UMTS planning problems. They have a very wide dynamic range in the matrix coefficients and tend to be numerically instable	[8]
mzzv11, mzzv42z	Railway slot allocation problems	[20]
net12	Network design instance	[1]
nsrand-ipx, roll3000	Rolling stock and line planning instances	
opt1217	This MIP represents one instance of time in the swiftly changing wireless environment of an orthogonal frequency division multiplexing system	[12]
protfold	Protein folding instance	[7]
rd-rplus-21	From reactor design of a metathesis process occuring in chemical process dynamics investigations in steady-state, the problem is a linearized relaxation of the originally nonlinear model for a destillation column with external reactor	[9]
sp97ar	Railway line planning instance	[10, 21]
stp3d	Steiner tree packing instance in a 3 dimensional grid-graph. The LP relaxation is highly degenerate	[14]
t1717	Vehicle scheduling set partitioning problem. Quite dense “chaining” problem	[5]
timetab1, timetab2	Public transport scheduling problems	[16]
tr12-30	Lot sizing instance	[22]

Table 1: Descriptions and references for new MIPLIB instances

2.2 New Instances

One goal of the MIPLIB is to collect challenging real world instances rather than artificially constructed ones. Of course, this requirement seems to be paradox because models that are likely to produce challenging instances are usually not used in practice. This means that many of the instances in the MIPLIB are derived from models that have not been implemented in a production environment.

However, it is not obvious whether such instances result from models which are poor in principle. It may also be the case that they include features which current MIP solvers are just not able to cope with. Therefore, it is reasonable to include such instances in the MIPLIB. They help to direct research to areas which are currently not well understood.

In addition, it is desirable to foster improvements on models that can be solved quite well with current techniques in order to extend the applicability of MIP solvers to very large instances. A few of those big instances have been included in the MIPLIB 2003 as well. Table 1 gives an overview of all 27 new instances.

2.3 Numerical difficulties

To get an idea how difficult the instances are with respect to numerics, we made the following experiment: Using different simplex based LP solvers [23, 24, 25, 26, 27, 28, 29] we computed "optimal" simplex bases of the LP relaxations of the MIPLIB instances. Of course, the resulting bases are only optimal with respect to the tolerances set in the LP solvers. We then used PERPLEX [13] to test the primal and dual feasibility of the computed bases using rational arithmetic. Note that the LP solvers are not able to represent all numbers given in the ASCII coded MPS files exactly and therefore work on a slightly perturbed problem, since they employ double precision floating point arithmetic."

	Clp		CPLEX		GLPSOL		lp_solve		QSopt		SoPlex		XPress	
Instance	primal	dual	primal	dual	primal	dual	primal	dual	primal	dual	primal	dual	primal	dual
arki001	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$
atlanta-ip	$\surd$	$\otimes$	$\surd$	$\surd$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$
dano3mip	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$
harp2	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\otimes$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
momentum1	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\otimes$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
momentum2	$\otimes$	$\otimes$	$\otimes$	$\otimes$	$\surd$	$\surd$	$\surd$	$\otimes$	$\otimes$	$\otimes$	$\surd$	$\otimes$	$\otimes$	$\otimes$
momentum3	$\otimes$	$\surd$	$\otimes$	$\otimes$	$\otimes$	$\surd$	—		$\otimes$	$\surd$	$\otimes$	$\otimes$	$\otimes$	$\otimes$
msc98-ip	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$	—		$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$
noswot	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	—		$\surd$	$\surd$	$\surd$	$\surd$	$\otimes$	$\surd$
qiu	$\otimes$	$\surd$	$\otimes$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$	$\otimes$	$\surd$
roll3000	$\otimes$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$
stp3d	$\surd$	$\surd$	$\surd$	$\surd$	$\otimes$	$\surd$	—		$\surd$	$\surd$	$\surd$	$\surd$	$\surd$	$\surd$

Table 2: Results of solving the root relaxations

The LP solvers were run with default settings where possible. The results are shown in Table 2. The first column lists those instances that were not solved to optimality by all LP solvers. The following columns show for each solver whether the computed basis was indeed *primal* and *dual* feasible: the symbol $\surd$ means *feasible* and $\otimes$ *infeasible*. If an instance could not be solved we write —. Instances arki001, momentum3, and msc98-ip are the only ones that none of the LP solvers was able to solve to optimality with default settings. Experiments with CPLEX, SOPLEX, and XPRESS revealed that the solvers are usually able to find an optimal LP solution when instance specific parameter settings are used.

2.4 Problem Instance Statistics

Table 4 lists the number of rows and columns for each instance. The columns are subdivided into binary (BIN), general integer (INT), and continuous variables (CON). Additionally, the objective values of an optimal MIP solution (MIP SOLN) and the LP relaxation (LP SOLN) are listed. Instances where no optimal MIP solution is known, i. e., *open* instances, are labeled with a question mark '?'.

A classification of the constraints is given in Table 3. In the table, x denotes a binary, y a general integer, and z a continuous variable. Letters a and b indicate integer numbers, letters p, q, r, s indicate real numbers. All constraints have been normalized by allowing to complement binary variables and to scale rows with some non zero value.

For every instance, each column of Table 5 shows the number of constraints for each type depicted in Table 3. If a constraint falls into more than one category, it is counted only once in the left most column of Table 5.

Column	Constraint type	Normalized representation
PAR	Set partitioning	$\sum x_i = 1$
PAC	Set packing	$\sum x_i \le 1$
COV	Set covering	$\sum x_i \ge 1$
CAR	Cardinality constraint	$\sum x_i = b$
BIN	Bin packing	$\sum a_i x_i + a_k x_k \le a_k$
IVK	Invariant knapsack	$\sum x_i \le b$
KNA	Knapsack	$\sum a_i x_i \le b$
IKN	Integer knapsack	$\sum a_i y_i \le b$
VLB	Variable lower bound	$p_k x_k - z_k \le 0$ (or $p_k x_k - y_k \le 0$)
VUB	Variable upper bound	$p_k x_k - z_k \ge 0$ (or $p_k x_k - y_k \ge 0$)
M01	Mixed binary constraint	$\sum p_i x_i + \sum r_i z_i \le t$ (or $= t$)
GEN	General constraint	$\sum p_i x_i + \sum q_i y_i + \sum r_i z_i \le t$ (or $= t$)

Table 3: Constraint types

Table 6 contains information regarding the origins of each problem instance. The column *originator* gives the name of the person (institution) whom (which) the problem originated from. The *formulator* denotes the person or institution who (which) created the MIP model. The *donator* contributed the instance to the MIPLIB.

2.5 Coefficient Matrices

It is interesting to analyze the structure of the coefficient matrices (see [18]). In addition to the aspect ratio of the number of columns and the number of rows, the instances differ in the density and the distribution of the non zero matrix elements. Figures 1, 2, 3, and 4 visualize the structure of the matrices. Instances markshare1, markshare2, and pk1 have been omitted because they are to small to be interesting in this regard. Dots in the pictures represent non zero elements. Note that the structure shown in the pictures is intensified, i. e., the instances are usually sparser than they appear here. This is especially true for stp3d where the ratio of zero to non zero elements of the matrix is about 50,000 to 1.

3 Submission and further development

We would like to continue to collect interesting instances sent to us and to make them available in the so-called *contributed* area. However, we point out that the goal of the MIPLIB is not to become an unlimited collection of MIP instances. Instead, we aim to provide a preselected set of interesting instances which deal as an acknowledged benchmark set for the community.

Regarding the next major update we see two directions of particular importance: Up to now, there are no genuine infeasible instances in the MIPLIB. We would like to supplement the library with a collection of infeasible instances. One problem in this area is that most infeasible instances are either easily solved, e. g., by preprocessing, or are very hard since the whole B&C-tree has to be explored. Nevertheless there are several applications which require a proof of infeasibility.

The second direction is still subject to discussion. We think it is more important to increase the information about each instance than to increase the number of instances. We would like to include the models with the instances, for example, in a modeling language or even by providing a problem description including the original data sets. This information is hardly available for the currently included instances. We encourage every reader knowing something about an instance to share this knowledge with us.

The MIPLIB 2003 including the bulletin board and the contributed area can be found at http://miplib.zib.de. There is also a mailing list at mailto:miplib@zib.de for announcements and questions.

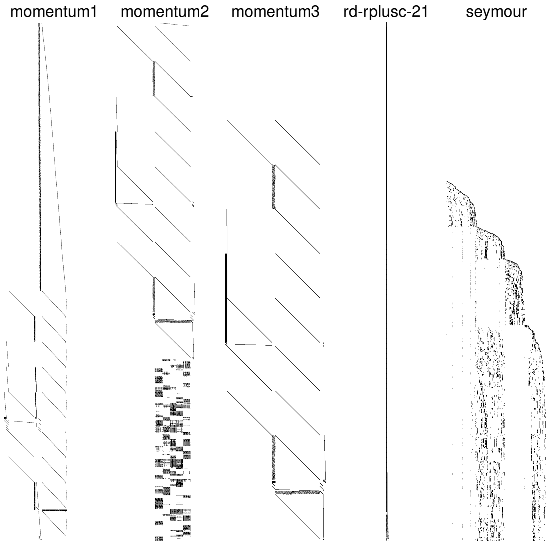

Figure 1: Coefficient matrix structure of instances with many more rows than columns

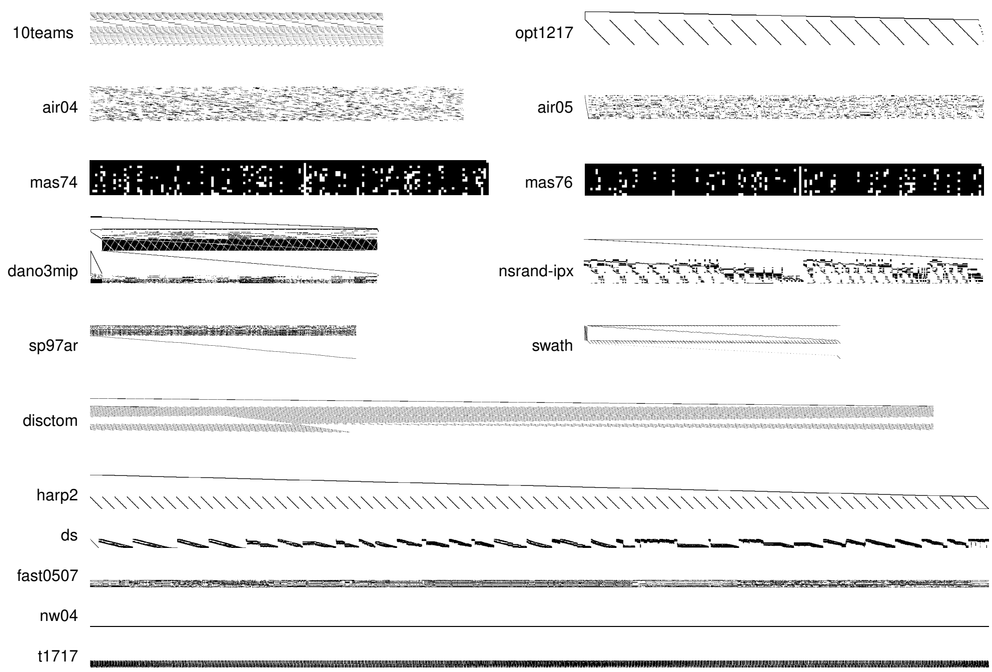

Figure 2: Coefficient matrix structure of instances with many more columns than rows

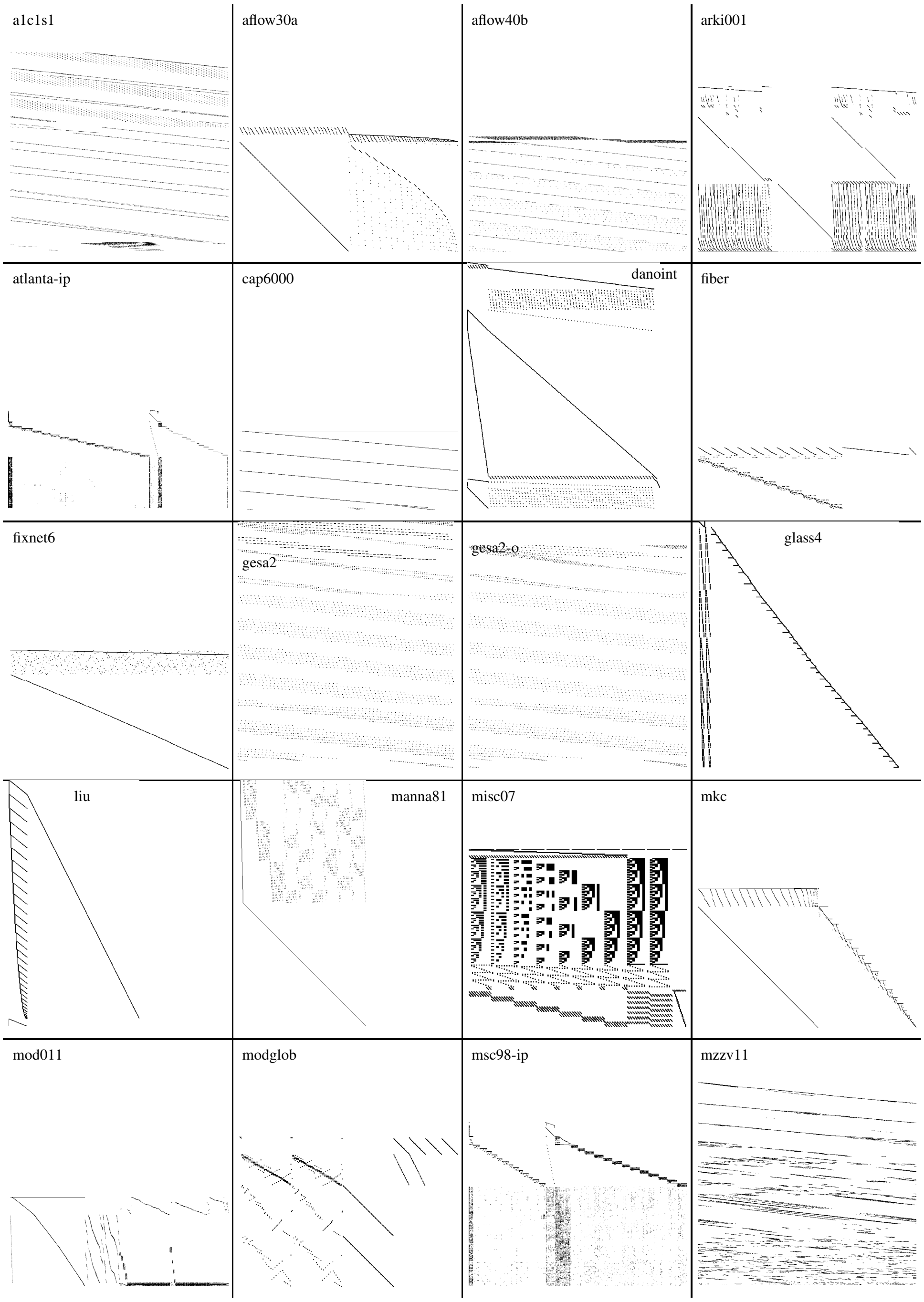

Figure 3: Coefficient matrix structure of instances

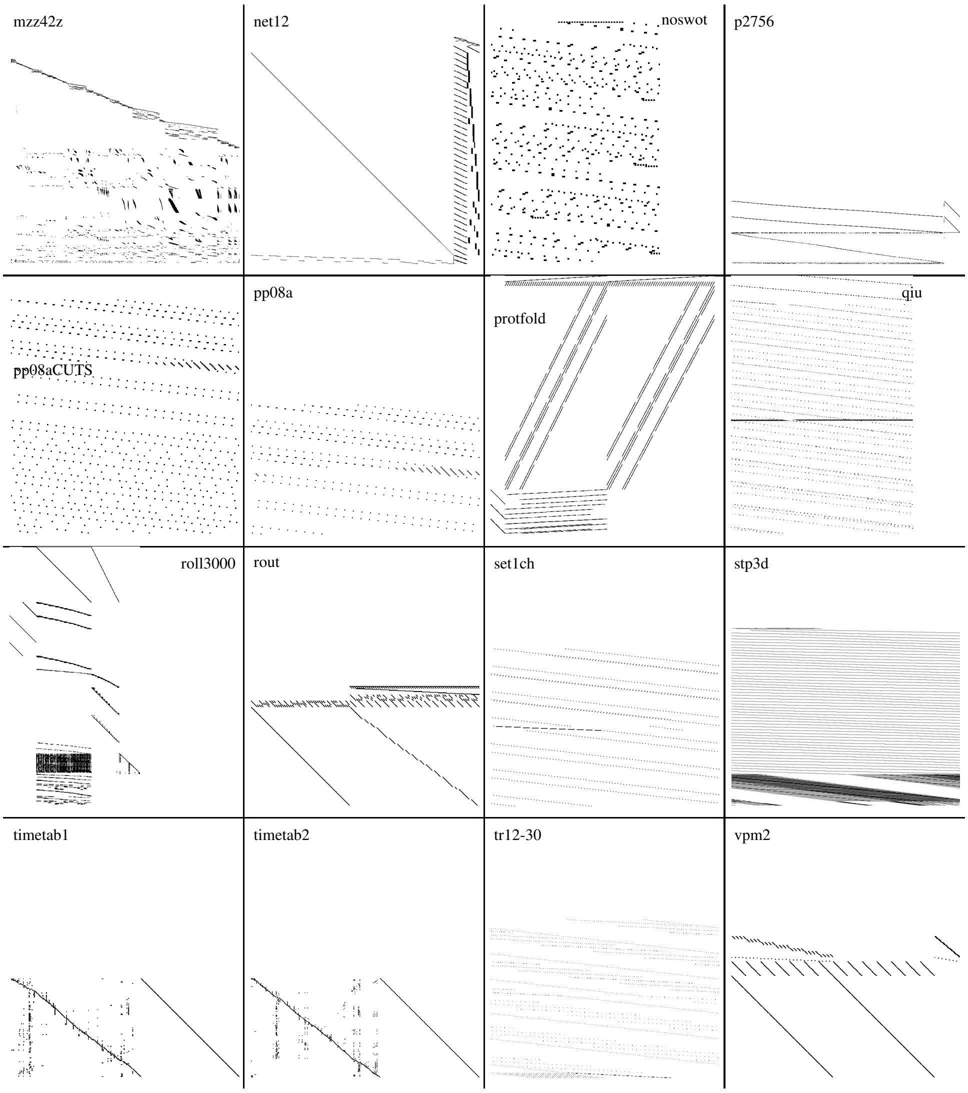

Figure 4: Coefficient matrix structure of instances

NAME	ROWS	COLS	BIN	INT	CON	MIP SOLN	LP SOLN	status
10teams	230	2025	1800		225	924	917	easy
a1c1s1	3312	3648	192		3456	?	997.529583	open
aflow30a	479	842	421		421	1158	983.167425	easy
aflow40b	1442	2728	1364		1364	1168	1005.66482	hard
air04	823	8904	8904			56137	55535.4364	easy
air05	426	7195	7195			26374	25877.6093	easy
arki001	1048	1388	415	123	850	?	7579599.81	open
atlanta-ip	21732	48738	46667	106	1965	?	81.2432017	open
cap6000	2176	6000	6000			-2451377	-2451537.33	easy
dano3mip	3202	13873	552		13321	?	576.23162	open
danoint	664	521	56		465	65.66666667	62.6372804	hard
disctom	399	10000	10000			-5000	-5000	easy
ds	656	67732	67732			?	57.2345653	open
fast0507	507	63009	63009			174	172.145567	hard
fiber	363	1298	1254		44	405935.18	156082.518	easy
fixnet6	478	878	378		500	3983	1200.884	easy
gesa2	1392	1224	240	168	816	25779856.4	25476489.7	easy
gesa2-o	1248	1224	384	336	504	25779856.4	25476489.7	easy
glass4	396	322	302		20	?	800002400	open
harp2	112	2993	2993			-73899798.84	-74353341.5	hard
liu	2178	1156	1089		67	?	346	open
manna81	6480	3321	18	3303		-13164	-13297	easy
markshare1	6	62	50		12	1	0	hard
markshare2	7	74	60		14	1	0	hard
mas74	13	151	150		1	11801.1857	10482.7953	easy
mas76	12	151	150		1	40005.0541	38893.9036	easy
misc07	212	260	259		1	2810	1415	easy
mkc	3411	5325	5323		2	-563.846	-611.85	hard
mod011	4480	10958	96		10862	-54558535	-62121982.6	easy
modglob	291	422	98		324	20740508.1	20430947.6	easy
momentum1	42680	5174	2349		2825	?	72793.3456	open
momentum2	24237	3732	1808	1	1923	?	7225.44069	open
momentum3	56822	13532	6598	1	6933	?	91952.3923	open
msc98-ip	15850	21143	20237	53	853	?	19520966.2	open
mzzv11	9499	10240	9989	251		-21718	-22945.2407	easy
mzzv42z	10460	11717	11482	235		-20540	-21622.9985	easy
net12	14021	14115	1603		12512	214	17.2494792	hard
noswot	182	128	75	25	28	-41	-43	hard
nsrand-ipx	735	6621	6620		1	51200	48880	hard
nw04	36	87482	87482			16862	16310.6667	easy
opt1217	64	769	768		1	-16	-20.0213904	easy
p2756	755	2756	2756			3124	2688.75	easy
pk1	45	86	55		31	11	1.4738988e-9	easy
pp08aCUTS	246	240	64		176	7350	5480.60616	easy
pp08a	136	240	64		176	7350	2748.34524	easy
protfold	2112	1835	1835			?	-41.9574468	open
qiu	1192	840	48		792	-132.873137	-931.638845	easy
rd-rplusc-21	125899	622	457		165	?	100	open
roll3000	2295	1166	246	492	428	?	11097.1277	open
rout	291	556	300	15	241	1077.56	981.864286	easy
set1ch	492	712	240		472	54537.75	32007.7299	easy
seymour	4944	1372	1372			423	403.846474	hard
sp97ar	1761	14101	14101			?	652560397	open
stp3d	159488	204880	204880			?	481.877787	open
swath	884	6805	6724		81	?	334.496858	open
t1717	551	73885	73885			?	134531.021	open
timtab1	171	397	64	107	226	764772	28694	hard
timtab2	294	675	113	181	381	?	83592	open
tr12-30	750	1080	360		720	130596	14210.4266	hard
vpm2	234	378	168		210	13.75	9.8892646	easy

Table 4: Problem sizes and optimal solution values

NAME	PAR	PAC	COV	CAR	BIN	IVK	KNA	IKN	VLB	VUB	M01	GEN
10teams	80	40									110	
a1c1s1										960	2352	
aflow30a	10			19						421	29	
aflow40b	15			24						1364	39	
air04	823											
air05	426											
arki001		2				6	13	40			558	429
atlanta-ip	1752	178	1884	5306		342	774	446		1848	9202	
cap6000	123	2046				5	2					
dano3mip				48					30	606	2518	
danoint				16						392	256	
disctom	6			393								
ds	656											
fast0507		3	504									
fiber	90			229							44	
fixnet6										378	100	
gesa2-o		48	144					168	144	288	48	408
gesa2		48						24	144	288	192	696
glass4	36										360	
harp2	73				9		30					
liu											2178	
manna81								6480				
markshare1											6	
markshare2											7	
mas74						1					12	
mas76						1					11	
misc07	7	3	127	27	2	45					1	
mkc		3361			24	24					2	
mod011										64	4416	
modglob										196	95	
momentum1	100	24415				116			8522	2445	7082	
momentum2	100	2023	8336		60	13		1	6562	1763	5379	
momentum3	450	6215			201	3		3	24123	6282	19545	
msc98-ip	1110	773	4360	3168		851	473	218		828	4069	
mzzv11	1073	5804	28	2208		5	2	288				91
mzzv42z	2157	5684		2262				274				83
net12	16	401	2	536		10					13056	
noswot								32		20		130
nsrand-ipx		331					403				1	
nw04	36											
opt1217	48										16	
p2756		352			16		387					
pk1											45	
pp08a										64	72	
pp08aCUTS										64	182	
protfold	35	49	1906	2		120						
qiu										528	664	
rd-rplusc-21	44										125855	
roll3000		405	119		88			1020			254	409
rout		14						6		230		41
set1ch										240	252	
seymour		285	4659									
sp97ar		1214	129				418					
stp3d	82	131403		28003								
swath	423			80							381	
t1717	551											
timtab1											64	107
timtab2											113	181
tr12-30										360	390	
vpm2										168	66	

Table 5: Constraint classification

NAME	ORIGINATOR	FORMULATOR	DONATOR
10teams	Dash Associates		Daniel
a1c1s1	Vyve, Pochet	Vyve, Pochet	Lodi
aflow30a	Achterberg	Achterberg	Achterberg
aflow40b	Achterberg	Achterberg	Achterberg
air04			Astfalk
air05			Astfalk
arki001	Avesta-Sheffield	Holmberg	Drud
atlanta-ip	Bienstock	Bley	Bley
cap6000	Hoffman, Padberg	Telecom. Corp.	Hoffman
dano3mip	Bell Com. Research		Bienstock
danoint	CTR U Columbia		Bienstock
disctom	de Vries	de Vries	de Vries
ds	Löbel	Löbel	Löbel
fast0507	Italian Railway Company	Guida	Ceria
fiber	US West	Lee	Savelsbergh
fixnet6		Van Roy	Savelsbergh
gesa2-o	Spanish Electricity	GESA	Ceria
gesa2	Spanish Electricity	Wolsey	Ceria
glass4	Luzzi	Luzzi	Lodi
harp2			Savelsbergh
liu	U Washington EE	Liu	Klotz
manna81	Koster, Zymolka	Koster, Zymolka	Koster, Zymolka
markshare1	Cornuejols, Dawande	Cornuejols, Dawande	Dawande
markshare2	Cornuejols, Dawande	Cornuejols, Dawande	Dawande
mas74			Eckstein
mas76			Eckstein
misc07			Astfalk
mkc	Kalagnanam, Dawande	Kalagnanam, Dawande	Kalagnanam, Dawande
mod011	Suhl	Suhl	Forrest
modglob	Smeers	Wolsey	Savelsbergh
momentum1	IST-2000-28088	WP4 Team	Koch
momentum2	IST-2000-28088	WP4 Team	Koch
momentum3	IST-2000-28088	WP4 Team	Koch
msc98-ip	E-Plus, Wessäly	Bley	Bley
mzzv11	Lukac	Lukac	Lukac
mzzv42z	Lukac	Lukac	Lukac
net12	Belotti	Belotti	Lodi
noswot		Schrage	Gregory
nsrand-ipx	Kroon	Kroon	Lodi
nw04	Northwest Airlines		Hoffman
opt1217	TNG TU Berlin	Gross	Geerdes
p2756	CJP set	Johnson	Boyd
pk1		Keskinocak	Ceria
pp08a			Savelsbergh
pp08aCUTS			Savelsbergh
protfold	Fügenschuh	Fügenschuh	Fügenschuh
qiu	Chiu	Chiu	Eckstein
rd-rplusc-21		Haus, Michaels, Weismantel	Haus
roll3000	Kroon	Kroon	Lodi
rout	Graves	Abeledo	Ceria
set1ch		Wolsey	Savelsbergh
seymour			Seymour
sp97ar	Goessens, v. Hoessel, Kroon	Goessens, v. Hoessel, Kroon	Lodi
stp3d	ZIB	Jørgensen, Koch	Koch
swath	Panton	Panton	Panton
t1717	Telebus Berlin	Borndörfer	Borndörfer
timtab1		Liebchen, Möhring	Liebchen, Möhring
timtab2		Liebchen, Möhring	Liebchen, Möhring
tr12-30	Vyve, Pochet	Vyve, Pochet	Lodi
vpm2		Wolsey	Savelsbergh

Table 6: Problem origins

References

[1] P. Belotti and F. Malucelli. A lagrangian relaxation approach for the design of networks with shared protection. In *Proceedings of INOC'2003*, pages 72–77, Oct. 2003.

[2] R. Bixby, E. Boyd, and R. Indovina. MIPLIB: A test set of mixed integer programming problems. *SIAM News*, 25:16, 1992.

[3] R. Bixby, S. Ceria, C. McZeal, and M. Savelsbergh. An updated mixed integer programming library: MIPLIB 3.0. *Optima*, 58:12–15, June 1998.

[4] A. Bley and T. Koch. Integer programming approaches to access and backbone IP-network planning. Technical Report 02-41, Zuse Institute Berlin, 2002.

[5] R. Borndörfer. *Aspects of Set Packing, Partitioning, and Covering*. PhD thesis, Technische Universität Berlin, 1998.

[6] R. Borndörfer, M. Grötschel, and A. Löbel. Duty scheduling in public transit. In W. Jäger and H.-J. Krebs, editors, *Mathematics - Key Technology for the Future*, pages 441–459. Springer, 2003.

[7] A. Dittel, A. Fügenschuh, and A. Martin. Protein folding and integer programming. Technical Report to appear, Technische Universität Darmstadt, 2005.

[8] A. Eisenblätter, A. Fügenschuh, E. Fledderus, H.-F. Geerdes, B. Heideck, D. Junglas, T. Koch, T. Kürner, and A. Martin. Mathematical methods for automatic optimization of UMTS radio networks. Technical Report D4.3, IST-2000-28088 MOMENTUM, 2003.

[9] J. Gangadwala, A. Kienle, U.-U. Haus, D. Michaels, and R. Weismantel. Optimal process design for the synthesis of 2,3-dimethylbutene-1. In *European Symposium on Computer-Aided Process Engineering - 15*, Computer-aided chemical engineering, Barcelona, May 29–June 1 2005, 2005. Elsevier. To appear.

[10] J.-W. H. M. Goossens, S. van Hoesel, and L. G. Kroon. A branch-and-cut approach for solving railway line-planning problems. *Transportation Science*, 38(3):379–393, 2004.

[11] P. Gritzmann and S. de Vries. Reconstructing crystalline structures from few images under high resolution transmission electron microscopy. In W. Jäger and H.-J. Krebs, editors, *Mathematics - Key Technology for the Future*, pages 441–459. Springer, 2003.

[12] J. Gross and H. Karl. Comparison of different fairness approaches in OFDM-FDMA systems. Technical Report TKN-04-003, Technische Universität Berlin, Telecommunication Networks Group, Mar. 2004.

[13] T. Koch. The final NETLIB-LP results. *Operations Research Letters*, 32:138–142, 2004.

[14] T. Koch. *Rapid Mathematical Programming*. PhD thesis, Technische Universität Berlin, 2004.

[15] A. Koster and A. Zymolka. Polyhedral investigations on stable multi-sets. Technical Report 03-10, Zuse Institute Berlin, 2003.

[16] C. Liebchen and R. Möhring. Information on the MIPLIB's timetab-instances. Technical Report 2003/49, Technische Universität Berlin, Dept. of Mathematics, Dec. 2003.

[17] I. Luzzi. *Exact and Heuristic Methods for Nesting Problems*. PhD thesis, University of Padova, 2002.

[18] A. Martin. Integer programs with block structure. Habilitations-Schrift, Technische Universität Berlin, 1998.

[19] T. Pfender. Arboreszenz-Flüsse in Graphen: polyedrische Untersuchungen. Master's thesis, Technische Universität Berlin, 2000.

[20] A. Reuter. Kombinatorische Auktionen und ihre Anwendungen im Schienenverkehr. Master's thesis, Technische Universität Berlin, 2005.

[21] S. van Hoesel, J. Goessens, and L. Kroon. A branch-and-cut approach to line planning problems. *Working Paper, Erasmus University*, 2001.

[22] M. V. Vyve and Y. Pochet. A general heuristic for production planning problems. *CORE Discussion Paper*, 56, 2001.

[23] COIN-OR Clp version 1.01.01d. http://www.coin-or.org.

[24] ILOG CPLEX version 8.0/9.03/9.1. http://www.ilog.com/products/cplex.

[25] GNU linear programming toolkit glpsol version 4.7. http://www.gnu.org/software/glpk.

[26] lp_solve 5.1. http://groups.yahoo.com/group/lp_solve.

[27] QSopt version 1.01 (build 031219). http://www.isye.gatech.edu/~wcook/qsopt.

[28] SoPlex version 1.2.2. http://www.zib.de/Optimization/Software/Soplex.

[29] Dash XPress-MP Optimizer version 15.30.12. http://www.dashoptimization.com.