On solving the ‘progressive party problem’ as a MIP

Erwin Kalvelagen

GAMS Development Corp

erwin@gams.com

INFORMS Annual Meeting 2002, San Jose, 11/17/02-11/20/02

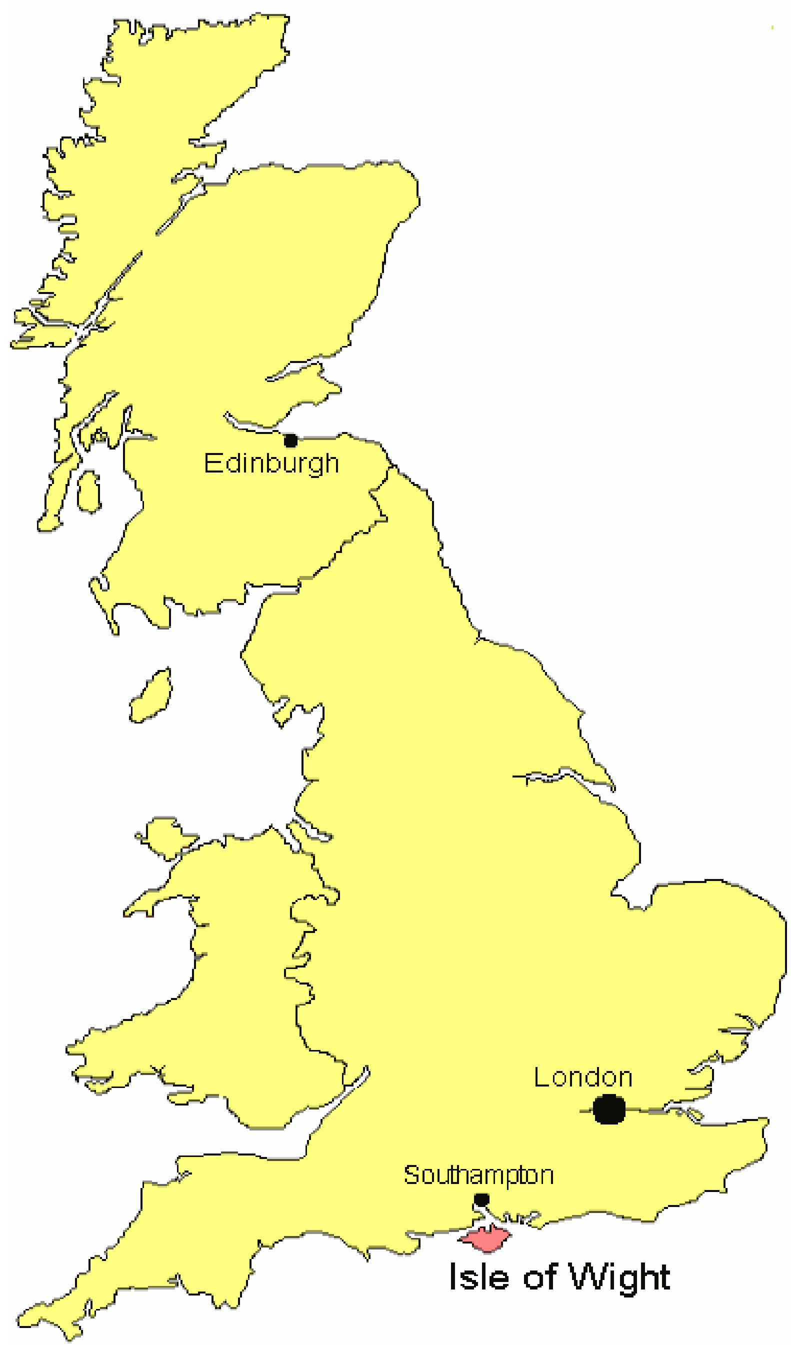
Outline

1. Problem description
2. Literature review
3. New formulation + results
4. Heuristic + results
5. Final remarks

Progressive Party Problem

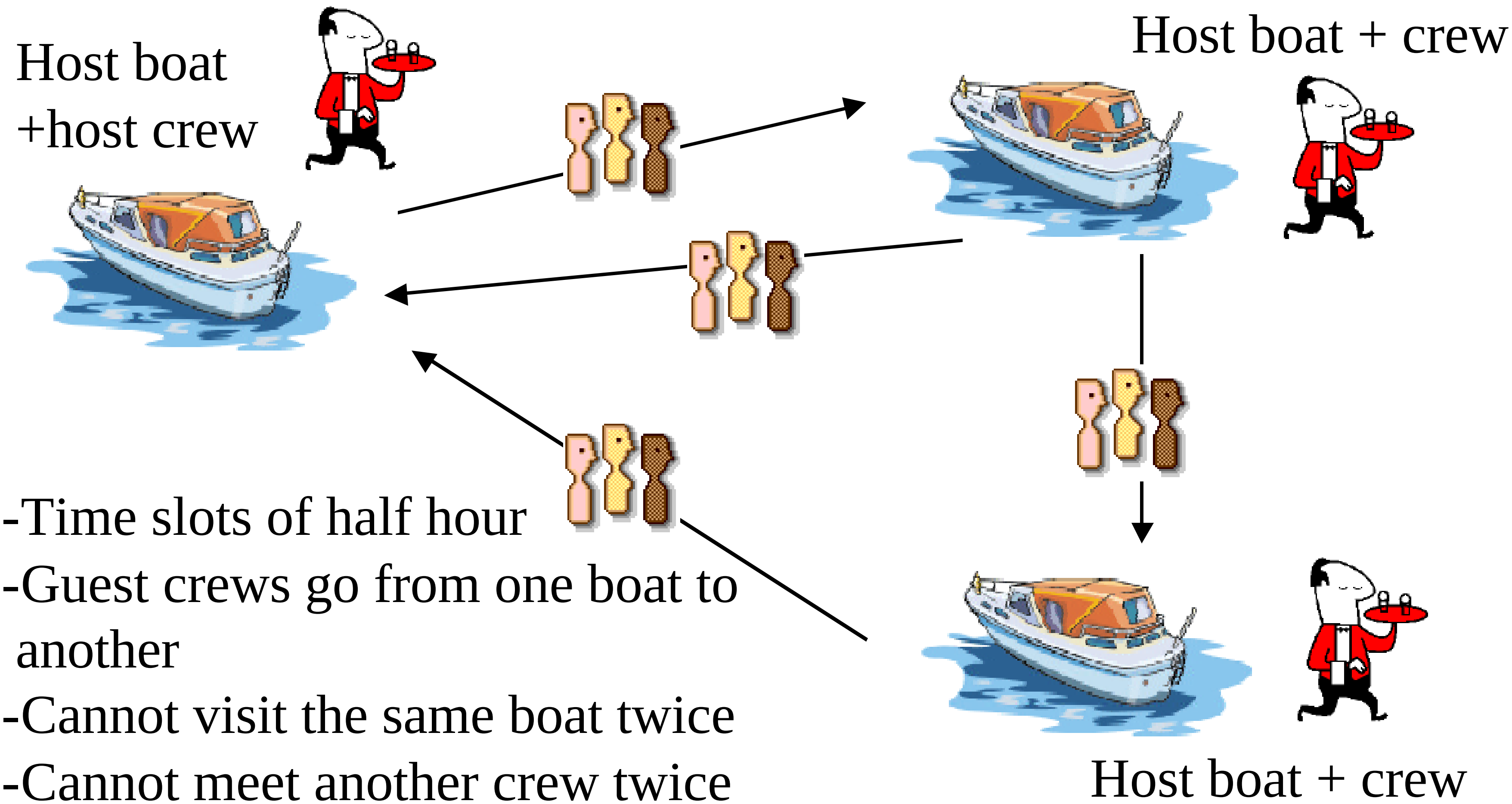
Background:

Bembridge Yacht Rally, Island of Wight



Parties are an important part of such events

Progressive Party



-Time slots of half hour

-Guest crews go from one boat to another

-Cannot visit the same boat twice

-Cannot meet another crew twice

Data

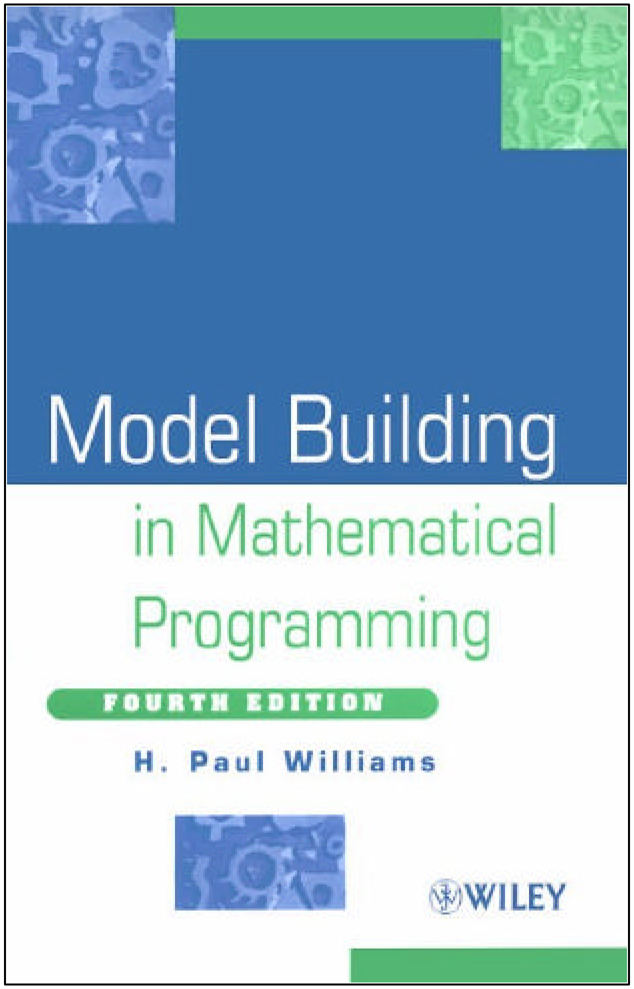
boat number	capacity	crew size	boat number	capacity	crew size
1	6	2	22	8	5
2	8	2	23	7	4
3	12	2	24	7	4
4	12	2	25	7	2
5	12	4	26	7	2
6	12	4	27	7	4
7	12	4	28	7	5
8	10	1	29	6	2
9	10	2	30	6	4
10	10	2	31	6	2
11	10	2	32	6	2
12	10	3	33	6	2
13	8	4	34	6	2
14	8	2	35	6	2
15	8	3	36	6	2
16	12	6	37	6	4
17	8	2	38	6	5
18	8	2	39	9	7
19	8	4	40	0	2
20	8	2	41	0	3
21	8	4	42	0	4

Notes:

- Capacity is including Hosts: max number of guest = capacity minus host crew size
- Some boats cannot serve as host boat (these are “crews” of children of the organizers)
- First three boats are designated host boats
- T = 6 visits for each guest crew

History

The problem was originally stated and solved heuristically by Peter Hubbard (*Southampton University*), organizer of the yacht rally. The problem was suggested to H. Paul Williams who formulated it as a MIP. Sally Brailsford tried to solve the problem using commercial MIP codes. Barbara Smith used constraint programming techniques.



First publications

B.M. Smith, S.C. Brailsford, H.P. Williams and P.M.Hubbard, "The Progressive Party Problem: Integer Linear Programming and Constraint Programming Compared", 36-52, in Principles and Practice of Constraint Programming: CP95, ed. Montanari and Rossi, Lecture Notes in Computer Science, 976, Springer (1995).

B.M. Smith, S.C. Brailsford, H.P. Williams and P.M.Hubbard, "Organising a social event: a difficult problem in combinatorial optimization", Computers and OR, 23, pp 845-856 (1996).

B.M. Smith, S.C. Brailsford, H.P. Williams and P.M.Hubbard, "The Progressive Party Problem: Integer Linear Programming and Constraint Programming Compared", Constraints, Vol 1 pp 119-38 (1996).

First Model

Let:

$$x_{i,j,t} = \begin{cases} 1 \text{ if crew } j \text{ visits boat } i \text{ at time slot } t \\ 0 \text{ otherwise} \end{cases}$$

$$h_i = \begin{cases} 1 \text{ if boat } i \text{ is a host boat} \\ 0 \text{ otherwise} \end{cases}$$

Minimize number of host boats

$$\min \sum_i h_i$$

Parties on host boats

$$x_{i,j,t} \leq h_i \qquad i \neq j$$

Capacity constraint of host boats

$$\sum_{j|j\neq i} w_j x_{i,j,t} \leq g_i \quad \forall i,t$$

Crews are host or guest (no idling)

$$h_j + \sum_{i|i\neq j} x_{i,j,t} = 1 \quad \forall j,t$$

Crews can visit a boat only once

$$\sum_t x_{i,j,t} \leq 1 \qquad i \neq j$$

Crews cannot meet twice

$$\sum_{(i,t)|i \neq j, i \neq j'} x_{i,j,t} x_{i,j',t} \leq 1 \quad \forall j < j' \qquad \text{(nonlinear)}$$

Linearize into:

(a) $$x_{i,j,t} + x_{i,j',t} + x_{i',j,t'} + x_{i',j',t'} \leq 3$$

$$\forall (i, i', j, j', t, t') \mid \begin{array}{c} i \neq j, i \neq j', i' \neq j, i' \neq j' \\ i \neq i', j < j', t \neq t' \end{array}$$

There are $O(n^4 t^2)$ of these equations, too many to be workable. A test program showed the actual number is: 40294800

Crews cannot meet twice (2)

(b) Introduce new binary variables:

$$y_{i,j,j',t} = \begin{cases} 1 & \text{if crews } j \text{ and } j' \text{ visit boat } i \text{ at period } t \\ 0 & \text{otherwise} \end{cases}$$

then the binary multiplication can be written as:

$$y_{i,j,j',t} \le \frac{x_{i,j,t} + x_{i,j',t}}{2}$$

or

$$y_{i,j,j',t} \le x_{i,j,t}$$
$$y_{i,j,j',t} \le x_{i,j',t}$$

$$y_{i,j,j',t} \ge x_{i,j,t} + x_{i,j',t} - 1 \qquad i \ne j, i \ne j', j < j'$$

$$\sum_{(j',t)|j<j'} y_{i,j,j',t} \le 1$$

Crews cannot meet twice (3)

(c) Introduce binary variables

$$m_{j,j',t} = \begin{cases} 1 & \text{if crews } j, j' \text{ meet at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$m_{j,j',t} \geq x_{i,j,t} + x_{i,j',t} - 1 \quad \forall j < j', i, t$$

$$\sum_t m_{j,j',t} \leq 1 \quad \forall j < j'$$

We leave *m* unrestricted if crews *j*,*j'* don't meet

J. P. Walser. *Solving linear pseudo-boolean constraint problems with local search*. In Proceedings of the Fourteenth National Conference on Artificial Intelligence, 1997.

Alternative formulations

J. N. Hooker and M. H. Osorio, Mixed Logical-Linear Programming, Discrete Applied Mathematics, 96-97, pp. 395-442, 1999.

- Uses explicitly $x_{i,i,t}$ by adding $x_{i,i,t} \geq h_i$

we prefer

$$x_{i,j,t} \text{ only if } i \neq j$$

- Introduction of general integer variables

$$v_{j,t} = \text{boat visited by crew } j \text{ at time } t$$

which leads to some difficult big-M constraints

Results

- Smith e.a.: no results with XPRESS-MP, no results with parallel OSL on 7 RS/6000 machines after 189 hours. Some nodes took more than 2 hours. Constraint programming implementation using Ilog solver: 27 minutes on Sparc IPX
- Walser reports 7 minutes with Oz on Sparc 20 but mentions problems with other instances
- Hooker e.a. reports only smaller instances using Cplex/MIP

Complete model

$$\min \ z = \sum h_i$$

$$x_{i,j,t} \le h_i$$

$$\sum_j w_j x_{i,j,t} \le g_i$$

$$h_j + \sum_i x_{i,j,t} = 1$$

$$\sum_t x_{i,j,t} \le 1$$

$$m_{j,j',t} \ge x_{i,j,t} + x_{i,j',t} - 1$$

$$\sum_t m_{j,j',t} \le 1$$

GAMS Implementation

- GAMS implementation was straightforward
- Most complicated equation:

```
*
* guest crews can meet only once
* with aid of extra binary variables
*

meet.lo(lti(j,jj), t) = 0;
meet.up(lti(j,jj), t) = 1;
link(i,lti(j,jj),t)$(nd(i,j) and nd(i,jj))..
    meet(j,jj,t) =g= x(i,j,t) + x(i,jj,t) - 1;

meetonce(lti(j,jj)).. sum(t, meet(j,jj,t)) =l= 1;
```

Refinements

- Relax *m* to continuous variables (automatically integer)
- Fix h_i=1 for i=1,2,3
- Fix h_i=*0* for i=40,41,42
- Tighten

$$\sum_j w_j x_{i,j,t} \leq g_i h_i$$

$$\sum_t x_{i,j,t} \leq h_i$$

Refinements (2)

- Fix z=13
  - We can show that z=12 provides not enough capacity for all crews in period t=1.
- Priorities (branching order):
  - first deal with h then worry about x
  - First handle large crews, then smaller ones
- Cplex options:
  - Mipemphasis 1 (integer feasibility rather than optimality)
  - Primal simplex

GAMS/Cplex 7.0 results
AMD 1.2 Ghz PC/Linux/Win

Number of equations	220060
Number of variables	15541
Number of binary variables	10368
Nonzero elements	678997
Model Compilation time	0 s.
Model Generation time	4 s.
Cplex total solution time (lnx/win)	9054/1375 s.
Cplex lp solution time	165 s.
Cplex total iterations (lnx/win)	777860/107541
Cplex nodes (lnx/win)	507/197

Time staged heuristic

- Find solution for $t=1$
- Fix $h(i)$ and $x(i,j,1)$
- Find solution for $t=2$
- Fix $x(i,j,2)$
- Etc.
- Note: don't fix m as they are partly left undefined in each solve

Results

time stage	rows	cols	nz	disc var	obj	gen time	sol time
1	38830	2620	114130	1758	13	0.73	1.62
2	73270	3445	146371	1722	13	1.17	1.72
3	107710	4306	181672	1722	13	1.58	2.35
4	142150	5167	216973	1722	13	2.1	3
5	176590	6028	252274	1722	13	2.52	3.84
6	211030	6889	287575	1722	13	2.96	4.39
(7)	245470	7750	322876	1722	13	3.42	5.5

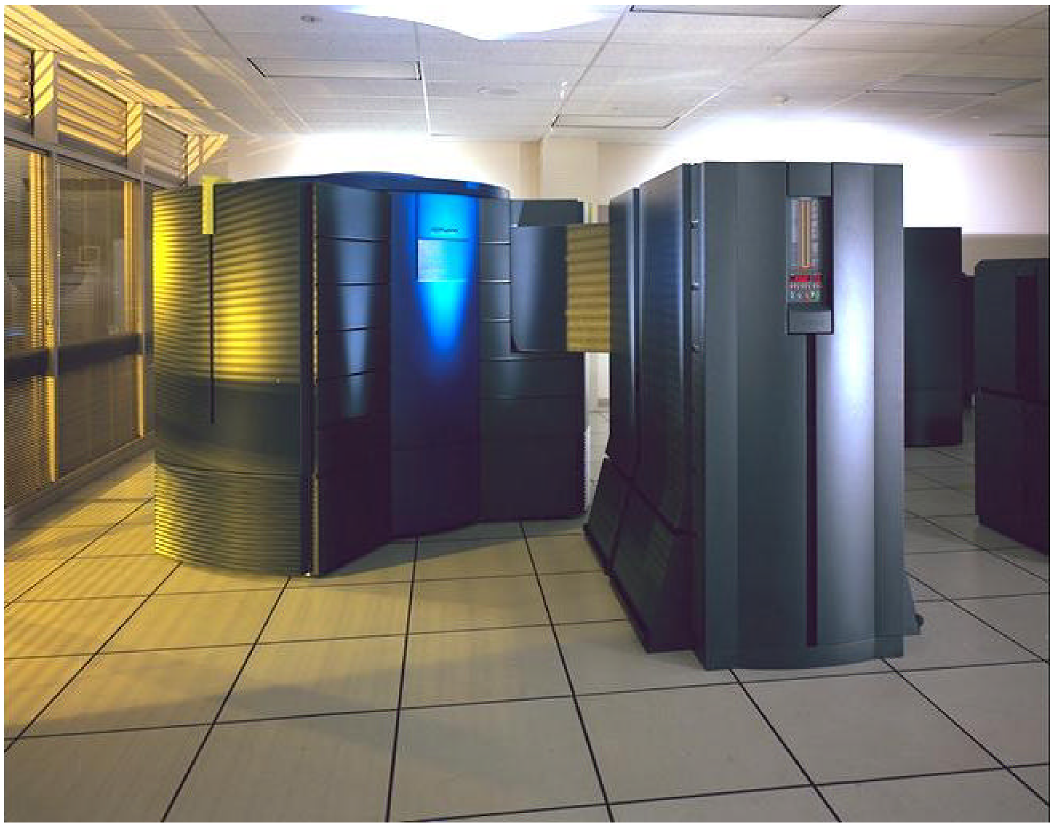
Even a seventh period could be added without an extra host boat

GAMS Implementation

```
loop(t,

*
* add new member to dynamic set
*
   td(t) = yes;

   solve m using mip minimizing nh;
   abort$(m.modelstat <> 1) "model became infeasible";

*
* fix variables for this time stage
*
   h.fx(i)$(ord(t)=1) = h.l(i);
   x.fx(i,j,t) = x.l(i,j,t);
);
```

Discussion

- Hardware progress



LINPACK:

Cray C90 (16 procs, 4.2 ns): 479 Mflop/s

AMD 1.2 Ghz: 558 Mflop/s

Discussion (2)

- Solver progress. E.g. presolver:

	before	*after*
rows	220060	125085
cols	15541	11928
nz	678997	396392

R. E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, R. Wunderling, MIP: Theory and Practice Closing the Gap, System Modelling and Optimization: Methods, Theory and Applications, Kluwer, The Netherlands, M. J. D. Powell and S. Scholtes, editors, pp. 19-49, 2000.

Discussion (3)

- Use of a modeling system
  - Invites doing experiments (minutes between idea and running a new reformulation)
  - Concise model representation that can be understood in full (a model is a system of *simultaneous* equations; no step-wise refinement).