Set Covering/Partitioning Applications

Jacques Desrosiers

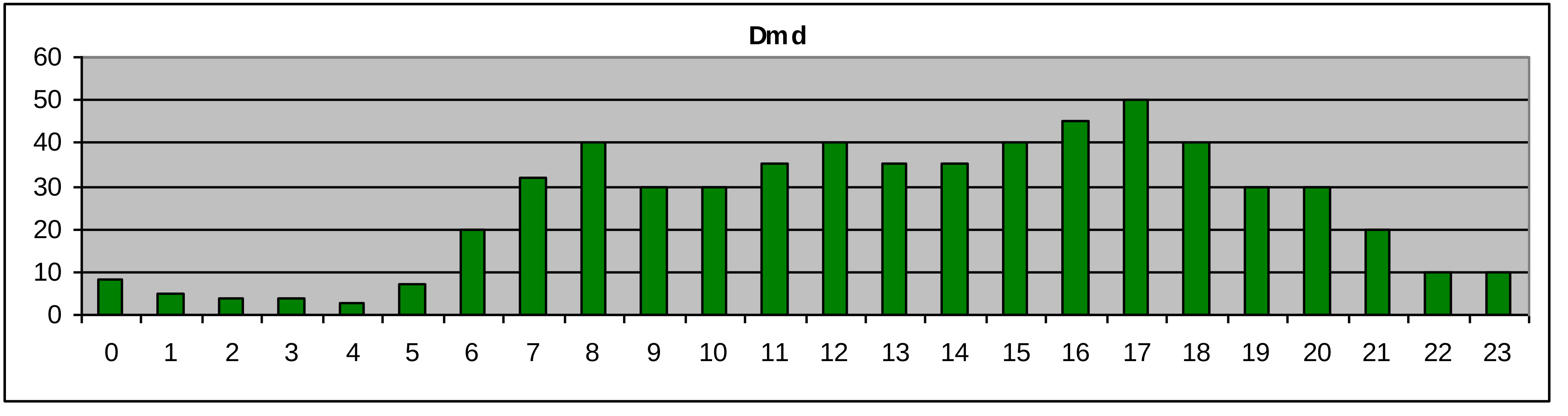
HEC Montréal

& GERAD

Canada

A Simple Scheduling Problem

- **Goal : Find work schedules for a set of people.**
- **Each person works 6 consecutive hours and can start at the beginning of any hour.**
- **Data : demand curve per hour.**



Various Scheduling Problems

- **Minimum number of persons**
- **Minimum cost**

- **Split duties : 6 on 7 hrs**
  - ***3hrs, 1hr lunch, 3hrs***
  - ***Fractional solution***
  - ***Large branch & bound tree (very difficult !)***
  - ***Rounding cut***

Various Scheduling Problems

- **Split duties : 7 on 8 hrs**
  - ***3hrs, 1hr lunch, 4hrs   or***

    4hrs, 1hr lunch, 3hrs
  - ***48 variables***
- **Additional am & pm breaks (15 minutes)**
  - ***More constraints***

Airline Applications

- **Replace "time periods to be covered, each by a certain number of people" by**

- **"flights to be covered, each by a single aircraft"**

  each by a certain number of pilots

  each by a certain number of flight attendants

  …

Rail Applications

- **Replace "time periods to be covered, each by a certain number of people" by**

- **"trains to be covered, each by a certain number of locomotives"**

General Structure

- **Replace "time periods to be covered, each by a certain number of people" by**

- **"tasks to be performed, each by a certain number of vehicles or crews"**

General Structure

- **Each column provides a feasible pattern, represented by a set of 0/1 values, that is, uncovered and covered tasks.**

- **Feasible patterns mathematically given by paths on time space networks.**

PROBLEM STRUCTURE

- **Time-Space Networks**
- **Local Schedule (*Path*) Restrictions**
- **Covering of a Set of Tasks**
- **Schedule Composition**

- **Non Linear Cost Functions**

1. **A GENERIC PROBLEM**

   Difficult to solve but many applications

2. **A MATHEMATICAL FORMULATION**

   A huge size with complex constraints

3. **DANTZIG-WOLFE REFORMULATION**

   Elimination of the non linearity and the complex constraints

Subproblem: Constrained Shortest Path

MIN REDUCED COST

$$\text{MIN} \sum_{\text{PAIRINGS}} \text{MAX}\left(\frac{\text{Pairing Duration}}{3.5}, \sum_{\text{DAYS}} \text{MAX}(4, \text{Credits})\right) - \text{Dual Costs}$$

S.T. - PATH STRUCTURE

- DAY DURATION ≤ 12 HOURS
- WORK TIME / DAY ≤ 8 HOURS
- WORK TIME / PAIRING ≤ MAX
- NIGHT REST ≥ MIN
- ...

10 TO 20 RESOURCE

Resource Constrained Shortest Path Problem on G=(N,A)

P(N, A) :

$$Min \sum_{(i,j)\in A} \bar{c}_{ij} x_{ij} + \sum_{i\in N} \sum_{r\in R} \bar{\lambda}_i T_i^r$$

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ij} = \begin{cases} 1\,(i=o) \\ 0\,(i\neq o,d) \\ -1\,(i=d) \end{cases}$$

$$x_{ij}(f_{ij}(\vec{T}_i) - T_j^r) \leq 0, \quad \forall (i,j)\in A, \forall r\in R$$

$$(\sum_{j:(i,j)\in A} x_{ij})\, a_i^r \leq T_i^r \leq (\sum_{j:(i,j)\in A} x_{ij})\, b_i^r, \quad \forall i\in N, \forall r\in R$$

$$x_{ij} \text{ binary}, \quad \forall (i,j)\in A$$

Linear Master Problem

COSTS

TASKS	39	43	42	40	38	38	…	43	44	34	41	34	33	38	58	41	…		
1	1						…	1			1				1		…	=	**1**
2		1	1			1	…		1	1		1		1		1	…	=	**1**
3	1	1		1			…	1			1		1		1		…	=	**1**
4		1	1		1	1	…		1	1			1	1	1		…	=	**1**
5			1		1		…	1	1			1				1	…	=	**1**
6			1	1		1	…				1			1	1	1	…	=	**1**
7				1			…	1		1		1					…	=	**1**
8	1	1			1		…		1				1		1	1	…	=	**1**
9					1	1	…	1			1	1		1	1		…	=	**1**
10	1	1		1			…		1	1					1		…	=	**1**
	13	17	15	11	13	14	…										…		**25**
							…	12	18	13	12	9	10	14	24	13	…	≤	**40**
	1	1	1	1	1	1	…										…	≤	**2**
							…	1	1	1	1	1	1	1	1	1	…	≤	**4**

RESEARCH TRENDS

- **Constraint Aggregation (F.S.)**
- **Integrated Levels**
- **Primal - Dual Stabilization**
- **Sub-Problem Speed-up**

School Busing

Dial-A-Ride

Bus Drivers

Airline Crew Scheduling

Vehicle Routing

Crew Rostering

Locomotive Assignment

Fleet Assignment

Preferential Bidding

1981 **Integer Programming Column Generation**

Locomotives & Cars

Buses & Drivers

Aircraft & Crews

Crew Recovery

1997 **2-Level Structures**

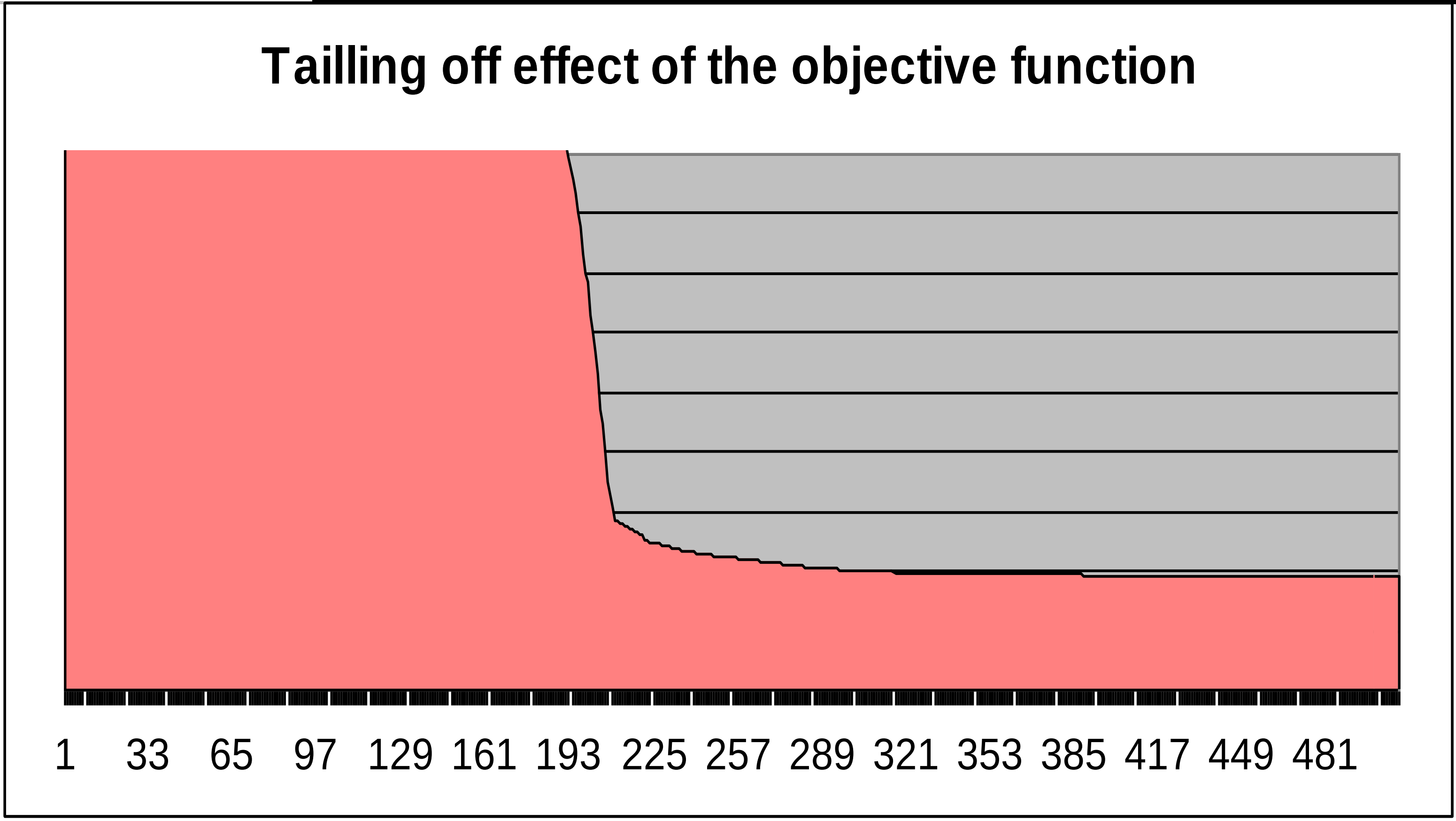
Bombardier Flexjet
Aircraft Fractional Ownership Operations

Flight Scheduling
& Fleet Assignment
& Aircraft Routing
& Crew Scheduling

2000 **A 4-Level Integrated Structure**

Primal Degeneracy in Column Generation

MDVSP R 800 (4)	cpu total	cpu mp	cpu sp	# CG iter	# SP cols	# MP itr
standard CG	**4178.4**	**3149.2**	1029.2	**509**	37579	926161



Primal Degeneracy

MDVSP *R* 800 (4)	cpu total	cpu mp	cpu sp	# CG iter	# SP cols	# MP itr
standard CG	**4178.4**	**3149.2**	1029.2	**509**	37579	926161

- **Master problem requires more than 75% of total cpu time**

Impact of Perfect Dual Information on CG

Problem R 800 (4)	Opt sol	Init sol	*cpu total*	# CG iter	# SP cols	# MP itr
standard	1915589.5	800000000	***4178.4***	**509**	37579	926161
dual boxes						
100		2035590.5	***835.5***	**119**	9368	279155
10		1927590.5	***117.9***	**35**	2789	40599
1		1916790.5	***52.0***	**20**	1430	8744
0.1		1915710.5	***47.5***	**19**	1333	8630

Dual-optimal Inequalities

Valério de Carvalho, Using extra dual cuts to accelerate column generation for the Cutting Stock problem, *Informs Journal on Computing* (2004).

- **Small items (*i=1,…,m*) are ranked:**

$$l_1 > l_2 > l_3 > \ldots \Rightarrow \pi_1 \geq \pi_2 \geq \pi_3 \geq \ldots$$

- **Additionally:**

$$l_i \geq l_j + l_k \Rightarrow \pi_i \geq \pi_j + \pi_k$$

A priori at most 2m dual constraints (or primal columns)

Dual-optimal Inequalities / Primal Columns

Generated cutting patterns			***a priori columns***						
			1				1		
			-1	1				1	
				-1	1				
					-1				
						…			
							-1		
							-1	-1	
								-1	
									…

Total cpu time reduced by 40%.

Triplets (501 items)
each roll is cut into exactly twice without waste

Standard CG	124.2 iterations
$\pi_1 \geq \pi_2 \geq \pi_3 \geq \ldots$	113.3 iterations

(Average over 10 problems)

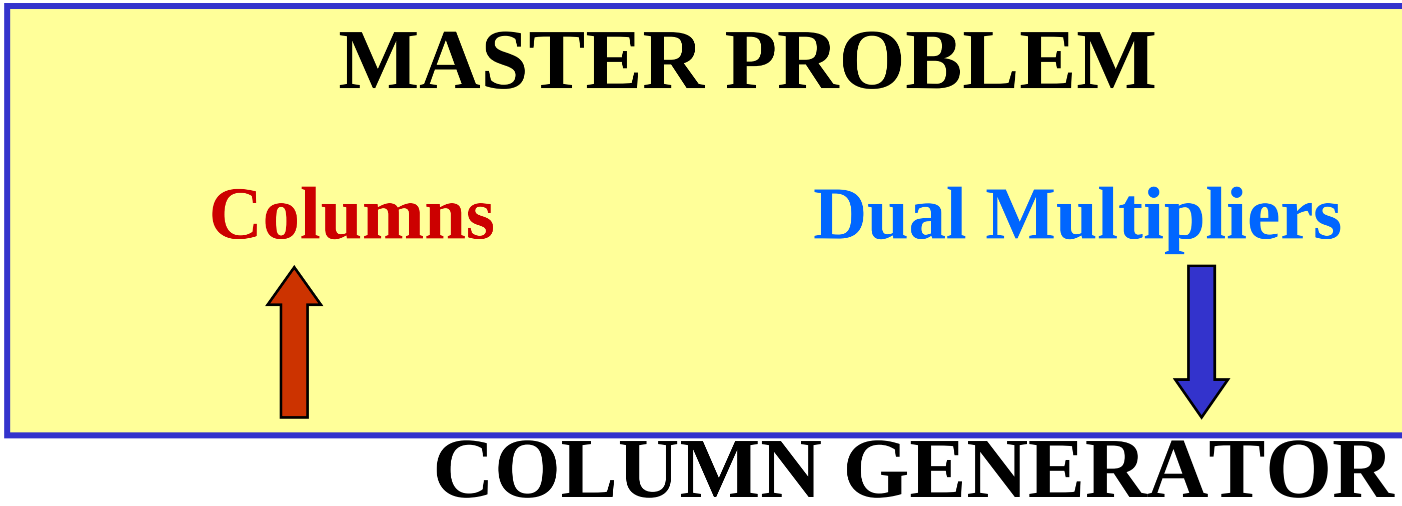
Triplets (501 items)

each roll is cut into exactly twice without waste

Standard CG	124.2 iterations
$\pi_1 \geq \pi_2 \geq \pi_3 \geq \ldots$	113.3 iterations
$\pi_i = \ell_i / L, \quad i = 1, \ldots, m$	12.2 iterations
(Average over 10 problems)	

Perfect dual information → $\pi_i = \ell_i / L$

LP Column Generation



LP Column Generation

MASTER PROBLEM

Columns ↑ **Dual Multipliers** ↓

COLUMN GENERATOR

Stopping rule?

LP Column Generation

MASTER PROBLEM

Columns ↑ **Dual Multipliers** ↓

COLUMN GENERATOR

Stopping rule?

Optimality of Dual Multipliers

Dual Boxes

Primal

$$\min cx$$
$$Ax = b$$
$$x \geq 0$$

↔

$$\max \; b\pi$$
$$\pi A \leq c$$

Dual

Dual Boxes

Primal

$$\begin{aligned} &\min cx \\ &Ax = b \\ &x \geq 0 \end{aligned}$$

⟷

Dual

$$\begin{aligned} &\max \ b\pi \\ &\pi A \leq c \end{aligned}$$

$$\begin{aligned} \min \quad & cx - d_1 y_1 + d_2 y_2 \\ & Ax - y_1 + y_2 = b \\ & x \geq 0 \\ & y_1 \geq 0, y_2 \geq 0 \end{aligned}$$

Relaxed **Primal**

⟷

$$\begin{aligned} &\max b\pi \\ &\pi A \leq c \\ &d_1 \leq \pi \leq d_2 \end{aligned}$$

Restricted **Dual**

Surplus & Slack Variables

Degeneracy & Perturbation

Primal

$$\begin{aligned} &\min cx \\ &Ax = b \\ &x \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & cx \\ Ax &= b \pm \varepsilon \\ x &\geq 0 \\ \varepsilon &> 0 \end{aligned}$$

Perturbed **Primal**

Degeneracy & Perturbation

Primal

$$\begin{aligned} \min\ & cx \\ Ax &= b \\ x &\geq 0 \end{aligned}$$

$$\begin{aligned} \min\quad & cx \\ Ax - y_1 + y_2 &= b \\ x &\geq 0 \\ 0 \leq y_1 \leq \varepsilon_1,\ 0 \leq y_2 &\leq \varepsilon_2 \end{aligned}$$

Relaxed **Primal**

Alternative

Perturbed **Primal**

Surplus & Slack Variables

Perturbation & Dual Boxes

Primal

$$\min cx$$
$$Ax = b$$
$$x \geq 0$$

$$\min \quad cx$$
$$Ax - y_1 + y_2 = b$$
$$x \geq 0$$
$$0 \leq y_1 \leq \varepsilon_1, 0 \leq y_2 \leq \varepsilon_2$$

Relaxed **Primal**

Perturbation & Dual Boxes

Primal

$$\begin{aligned} &\min cx \\ &Ax = b \\ &x \geq 0 \end{aligned}$$

$$\begin{aligned} \min \quad & cx - d_1 y_1 + d_2 y_2 \\ & Ax - y_1 + y_2 = b \\ & x \geq 0 \\ & y_1 \geq 0, y_2 \geq 0 \end{aligned}$$

Relaxed **Primal**

Perturbation & Dual Boxes

Primal

$$\min cx$$
$$Ax = b$$
$$x \geq 0$$

$$\min \quad cx$$
$$Ax - y_1 + y_2 = b$$
$$x \geq 0$$
$$0 \leq y_1 \leq \varepsilon_1, 0 \leq y_2 \leq \varepsilon_2$$

Relaxed **Primal**

$$\min \quad cx - d_1 y_1 + d_2 y_2$$
$$Ax - y_1 + y_2 = b$$
$$x \geq 0$$
$$y_1 \geq 0, y_2 \geq 0$$

Relaxed **Primal**

Perturbation & Dual Boxes

Primal

$$\min cx$$
$$Ax = b$$
$$x \geq 0$$

$$\min \quad cx$$
$$Ax - y_1 + y_2 = b$$
$$x \geq 0$$
$$0 \leq y_1 \leq \varepsilon_1, 0 \leq y_2 \leq \varepsilon_2$$

Relaxed **Primal**

$$\min \quad cx - d_1 y_1 + d_2 y_2$$
$$Ax - y_1 + y_2 = b$$
$$x \geq 0$$
$$y_1 \geq 0, y_2 \geq 0$$

Relaxed **Primal**

Stabilized Primal & Dual Problems

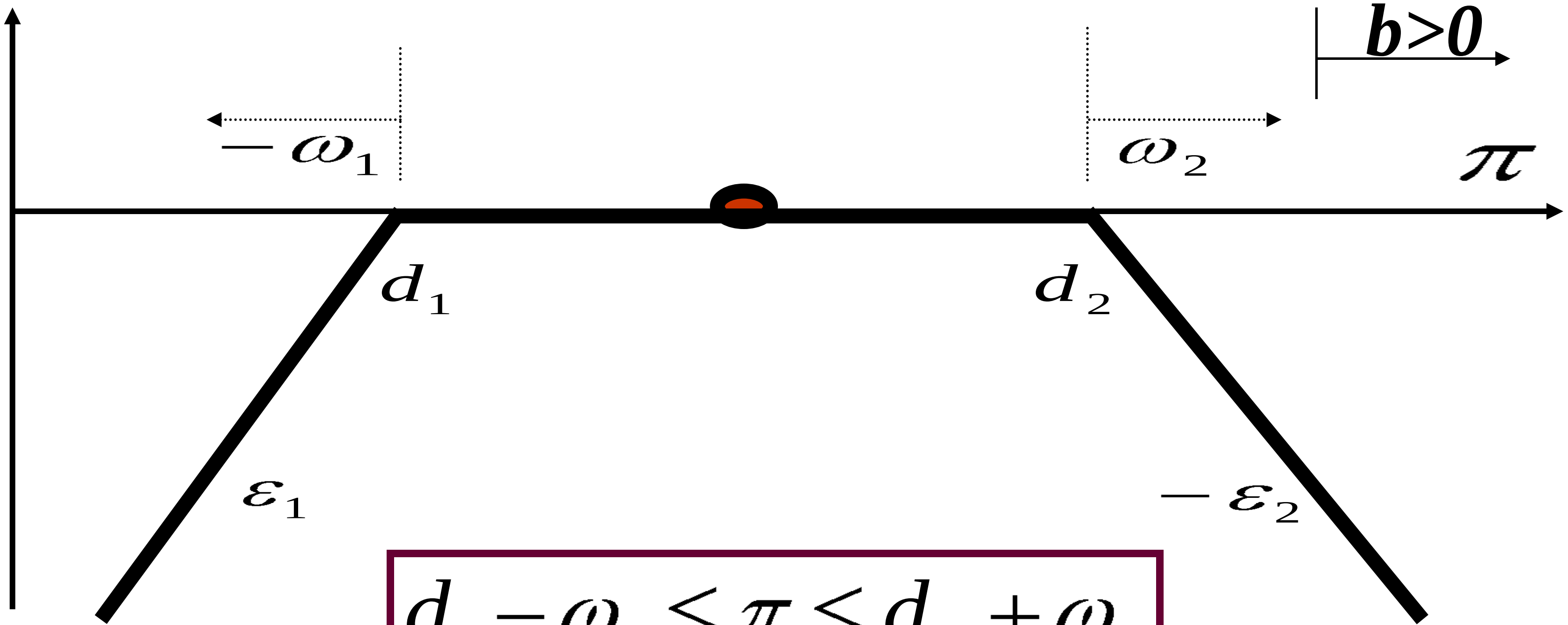
$$
\begin{aligned}
\min \quad & cx - d_1 y_1 + d_2 y_2 \\
& Ax - y_1 + y_2 = b && \pi \\
& y_1 \le \varepsilon_1 && -\omega_1 \le 0 \\
& y_2 \le \varepsilon_2 && -\omega_2 \le 0 \\
& x \ge 0, y_1 \ge 0, y_2 \ge 0
\end{aligned}
$$

Stabilized Primal SP

$$
\begin{aligned}
\max \; & b\pi - \omega_1 \varepsilon_1 - \omega_2 \varepsilon_2 \\
& \pi A \le c \\
& d_1 - \omega_1 \le \pi \le d_2 + \omega_2 \\
& \omega_1 \ge 0, \omega_2 \ge 0
\end{aligned}
$$

Stabilized Dual SD

Interpretation in Dual Space



$$d_1 - \omega_1 \le \pi \le d_2 + \omega_2$$

$$\omega_1 \ge 0, \omega_2 \ge 0$$

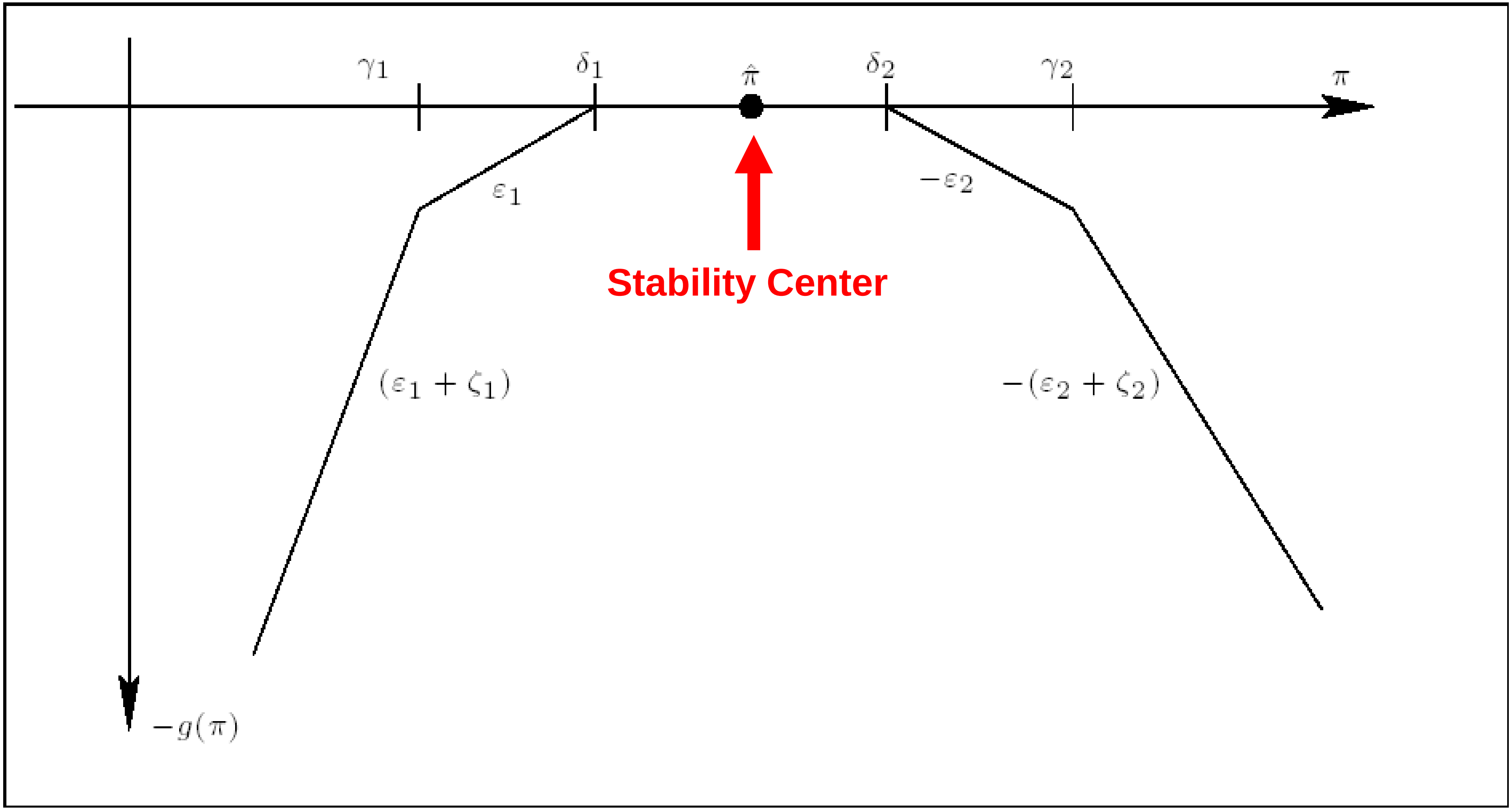


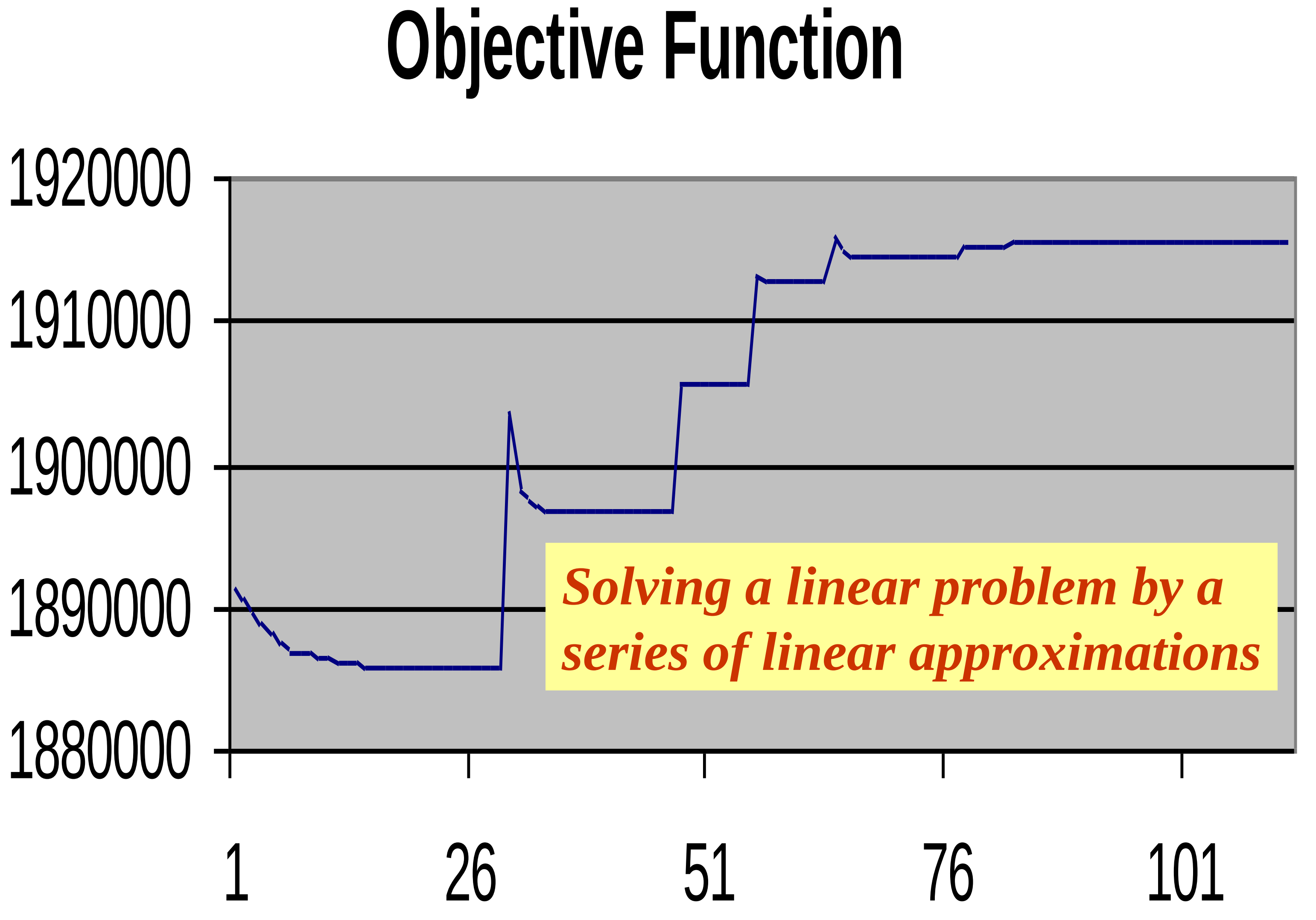
Figure 1: 5-piecewise linear dual penalty function

Stabilization Procedure for Problem P

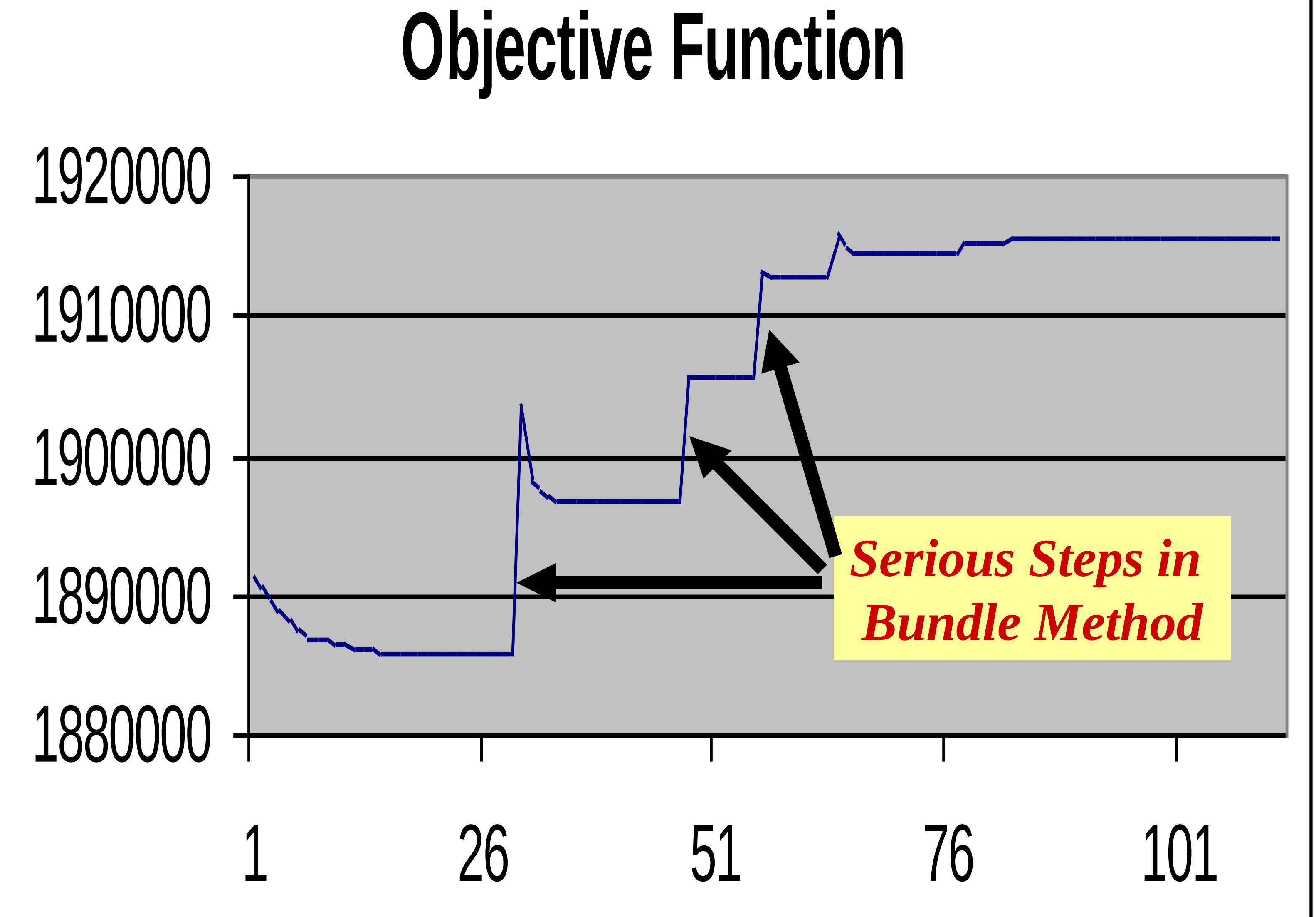
- **Initialize approximation problems SP & SD**
  - ***Stability center***
  - ***Trust region without penalties***
  - ***Penalties outside the trust region (3 to 5 pieces)***

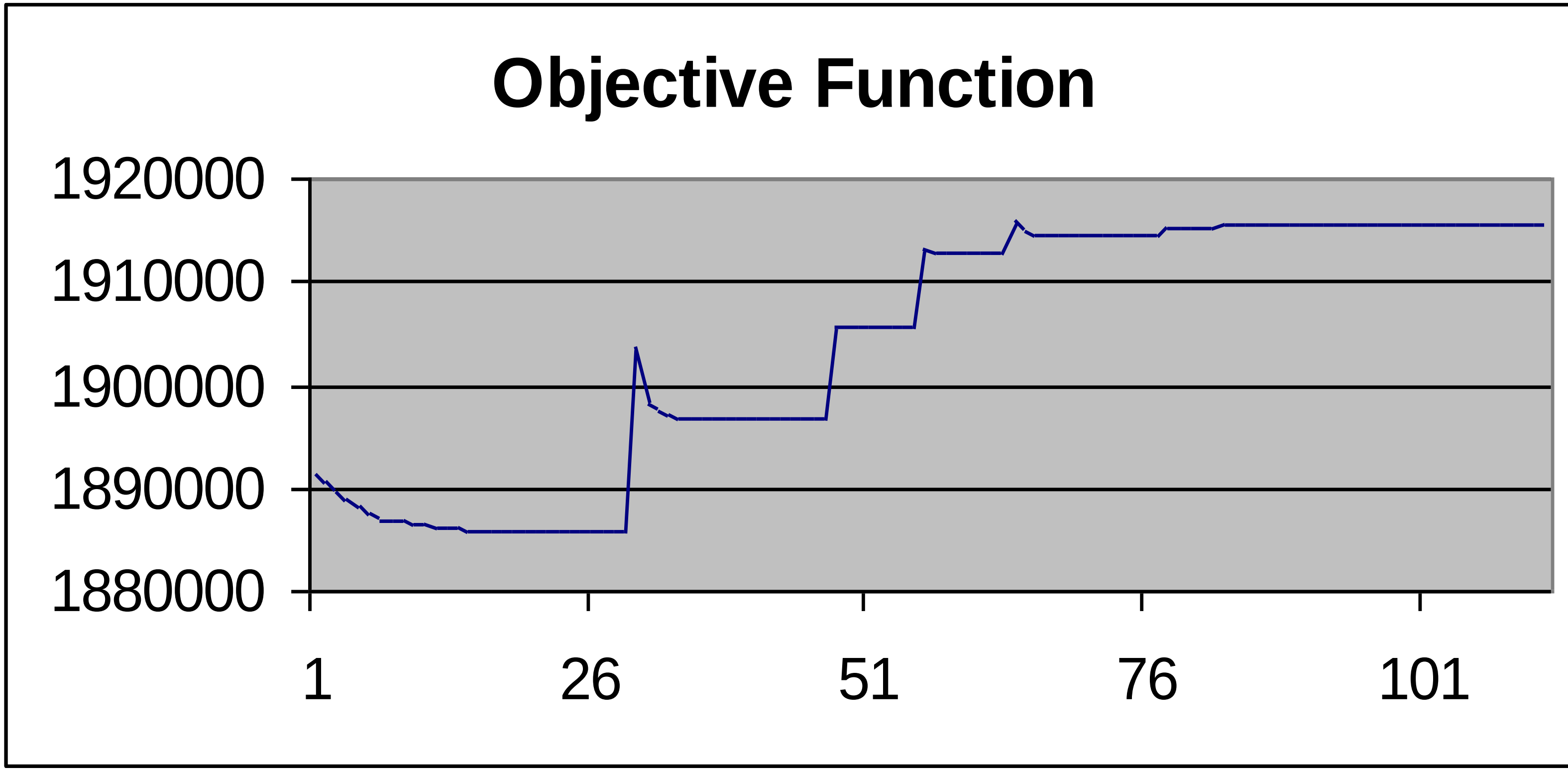
- **Solve stabilized problems SP & SD until P is feasible**
  - ***Otherwise update problems SP & SD***
    - Stability center, trust region and penalties

Problem R800 (4)



Problem R800 (4)





***Problem R* 800 (4)**	**Opt sol**	**Init sol**	***cpu tot***	**cpu mp**	**cpu sp**	**# CG iter**	**# SP cols**	**# MP itr**
standard	1915589.5	800000000	***4178.4***	3149.2	1029.2	**509**	37579	926161
delta = 100		2035590.5	***835.5***	609.1	226.4	**119**	9368	279155
network pi		2014429.8	***1097.1***	518.5	578.6	**301**	10105	324959
network pi +	network sol	1891386.0	***439.2***	216.2	223.0	**112**	4749	153420
		% reduction	***89.5***	**93.1**	**78.3**	**78.0**	**87.4**	**83.4**
			9.5 times faster					

Perfect Dual Information: Applications

- **Useful in the context of Lagrangian Relaxation to recover primal feasibility**

- **Useful to perform *Crossover* from an interior point solution to an extreme point solution**

Crossover: Large Crew Rostering Instances

	Constraints	Variables
pb1	12 351	126 326
pb2	12 310	129 046
pb3	13 190	146 013
pb4	13 433	151 654
pb5	13 550	162 914
pb6	13 451	156 839
pb7	13 254	148 025
pb8	13 424	154 205
pb9	13 598	163 707
pb10	13 310	155 313

- CPLEX7.5 **primal simplex algorithm fails** to solve any of these 10 problems in less than **18000 seconds** on a Entreprise 10000 solaris2.7 400MHZ machine (64 CPU, RAM=64G).

- The **dual simplex** needed less than 5000 seconds for two problems and failed to solve one within 18,000 seconds. The seven others require between 8,000 and 13,000 seconds.

- The problems are rather better solved by combining the CPLEX **Barrier algorithm with a primal or dual crossover.**

- For the interior point algorithm, we used values **10^{-8} and 10^{-10}** for the optimality parameter; in both cases, all problems were **solved in less than 1900 seconds.**

Crossover: CPLEX vs Box Methods (sec.)

Barrier 10^{-8}

Crossover	CrPrimal	CrDual	*BoxCrPrimal*
pb1	188	72	**9**
pb2	20	1183	**13**
pb3	327	435	**26**
pb4	8166	2568	**35**
pb5	59	2645	**45**
pb6	***	1797	**86**
pb7	270	1092	**22**
pb8	1036	1876	**20**
pb9	78	2811	**43**
pb10	37	3011	**30**
Avg	1834.7	1749	**32.9**
StdD	3350.1	1027.6	**22.1**
Min	20	72	**9**
Max	***	3011	**86**

Crossover: CPLEX vs Box Methods (sec.)

Barrier 10^{-10}

Crossover	CrPrimal	CrDual	*BoxCrPrimal*
pb1	101	15	**7**
pb2	***	27	**7**
pb3	605	69	**43**
pb4	232	2121	**16**
pb5	89	2819	**25**
pb6	72	2328	**22**
pb7	192	1025	**13**
pb8	1405	1407	**15**
pb9	7190	1957	**23**
pb10	4159	2832	**15**
Avg	2123.5	1460	**18.6**
StdD	2944.5	1127.2	**10.5**
Min	72	15	**7**
Max	***	2832	**43**

Variable Fixing in VR&CS Applications
(*basic ideas*)

- Consider the following IP

$$Z_{IP}^{*} = \min \quad c^{T} x$$

$$Ax \geq b \qquad [\pi]$$

$$Dx = d \qquad [\alpha]$$

$$x \in Z_{+}^{n}$$

$$X = \{x : Dx = d, x \in Z_{+}^{n}\}$$

Dantzig-Wolfe Reformulation

$$X = \{x : Dx = d, x \in Z_+^n\} = \{Qy : 1^T y = 1; y \geq 0\}$$

$$\begin{aligned} Z_{IP}^* = \min \quad & (c^T Q) y \\ & (AQ) y \geq b & [\pi] \\ & 1^T y = 1 & [\mu] \\ & Qy - x = 0 \\ & y \geq 0, \quad x \in Z_+^n \end{aligned}$$

Dantzig-Wolfe Reformulation with m identical pricing problems

$$X = \{x : Dx = d, x \in Z_+^n\} = \{Qy : 1^T y = 1; y \geq 0\}$$

$$
\begin{aligned}
Z_{IP}^* = \min \quad & (c^T Q) y \\
& (AQ) y \geq b && [\pi] \\
& 1^T y \leq m && [\mu = 0] \\
& Qy - x = 0 \\
& y \geq 0, \quad x \in Z_+^n
\end{aligned}
$$

Pricing Problem on Network N: A Classical Shortest s-t Path

$$Z^*_{PP(\pi,\mu)} = \min \quad (c^T - \pi^T A)x$$

$$I^N x = e_s - e_t \qquad [\alpha]$$

$$x \in Z^n_+$$

Reduced Cost of Arc (i,j)

$$\bar{c}_{ij} = c_{ij} - (\pi^T A)_{ij} - (\alpha^T D)_{ij}$$
$$= c_{ij} - (\pi^T A)_{ij} - \alpha_i + \alpha_j$$

If the reduced cost of arc (i,j) is larger than the *gap* (=UB-LB), then set x_{ij}=0.

If Network N does not contains negative cycles

$$\alpha_i = -\vec{l}_i \qquad \text{for all nodes } i$$

where $\vec{l}_i$ is the shortest path from s to i.

$$\bar{c}_{ij} = c_{ij} - (\pi^T A)_{ij} + \vec{l}_i - \vec{l}_j$$

Alternative Dual Solution for the Pricing Problem

$\alpha_i = \overleftarrow{l}_i$ for all nodes i

where $\overleftarrow{l}_i$ is the (backward) shortest path from t to i.

$$\bar{c}_{ij} = c_{ij} - (\pi^T A)_{ij} - \overleftarrow{l}_i + \overleftarrow{l}_j$$

Optimal Dual Solutions for the Pricing Problem

$$-\overrightarrow{l}_i \le \alpha_i \le \overleftarrow{l}_i \qquad \text{for all nodes } i$$

Maximum Reduced Cost for Arc (i,j)

$$\bar{r}_{ij} = \max_{\alpha} \{ c_{ij} - (\pi^T A)_{ij} - \alpha_i + \alpha_j \}$$

$$= c_{ij} - (\pi^T A)_{ij} + \overrightarrow{l}_i + \overleftarrow{l}_j$$

This value is always reachable on Acyclic Networks

Interpretation

$$\bar{r}_{ij} = \vec{l}_i + [c_{ij} - (\pi^T A)_{ij}] + \overleftarrow{l}_j$$

It is the reduced cost of the shortest *s*-*t*-path, a column, passing through arc (*i*, *j*) in network *N* with the modified arc cost.

Extensions

- **For shortest paths with resource constraints, in most applications, the underlying graph is acyclic.**

- **Otherwise, similar results can be obtained on the state space graph … which is acyclic.**

- **Computational experiments on the VRPTW show cpu times reduced by a factor of 3.**

Additional Set Partitioning Applications

- **MBA Teams**
- **A Secret Ballot Problem**

MBA Teams

- 26 persons to form teams of 4 or 5 persons.

- 4 x 4 and 2 x 5

- A *Transportation Prob. Constraint Structure*

MBA Teams

- **26 persons to form teams of 4 or 5 persons.**
- **4 teams of 4 and 2 teams of 5**
- ***A Transportation Prob. Constraint Structure***

MBA Teams

- **26 persons to form teams of 4 or 5 persons.**
- **4 teams of 4 and 2 teams of 5**
- **A *Transportation Prob. Constraint Structure***

MBA Teams

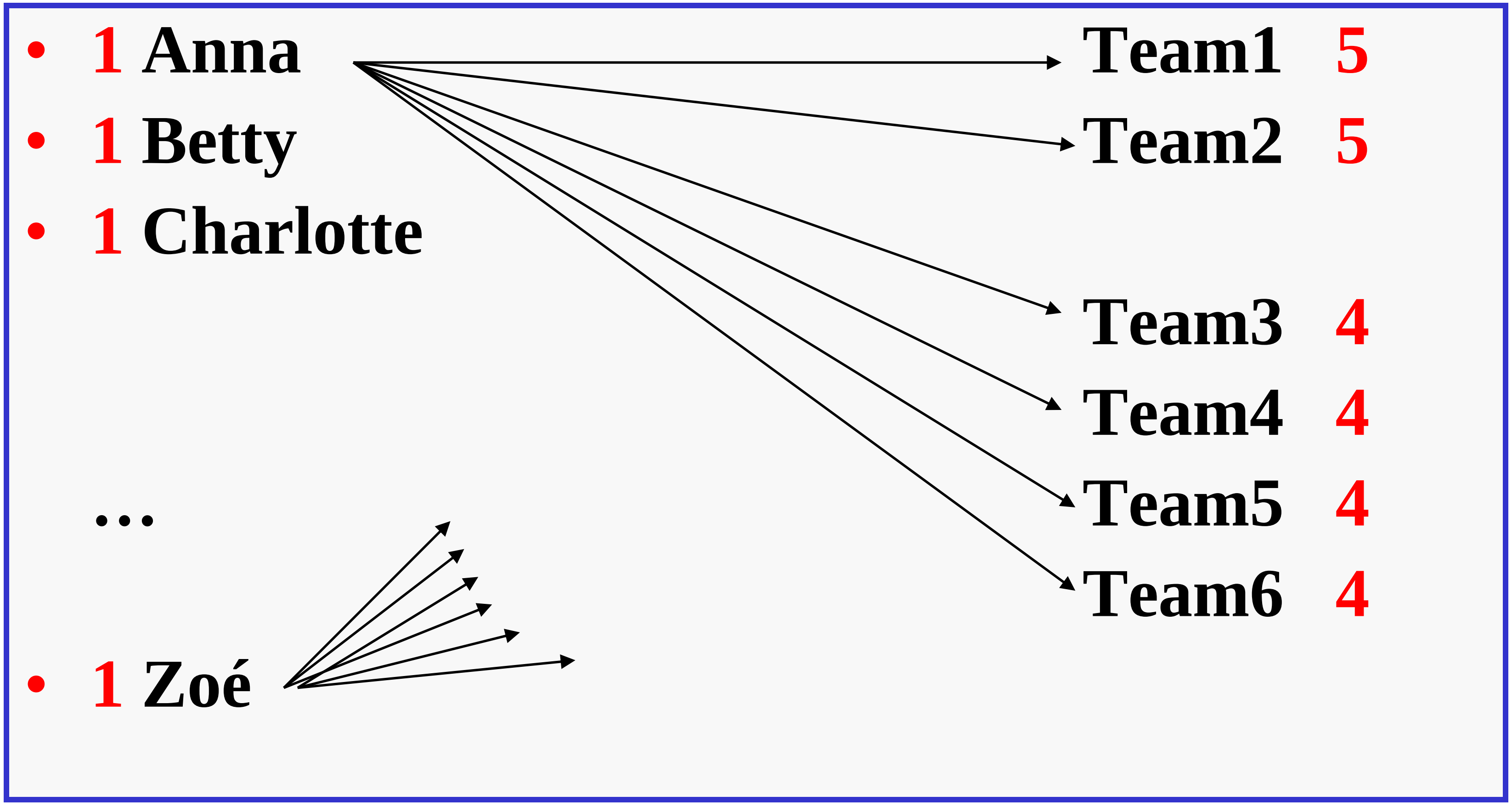
- 1 **Anna**
- 1 **Betty**
- 1 **Charlotte**

…

- 1 **Zoé**

Team1 5
Team2 5
Team3 4
Team4 4
Team5 4
Team6 4

Transportation Constraints



Transportation Constraints

$$X_{ij} = \begin{cases} 1 \text{ if } i \text{ in team } j \\ 0 \text{ otherwise} \end{cases}$$

$$\sum_{j=1}^{6} X_{ij} = 1 \quad i = 1..26$$

$$\sum_{i=1}^{26} X_{ij} = 5 \quad j = 1..2$$

$$\sum_{i=1}^{26} X_{ij} = 4 \quad j = 3..6$$

MBA Teams: Objective function

$$\min \sum_{team} \text{dist}(\text{team}, \text{target})$$

MBA Teams: Objective function

$$\min \sum_{team} \mathrm{dist}(\mathrm{team}, \mathrm{target})$$

- **Target vector**
  - ***Average (proportion) of attributes within the class group***

MBA Teams: Objective function

$$\min \sum_{team} \text{dist}(\text{team}, \text{target})$$

- **Target vector**
  - ***Average (proportion) of attributes within the class group***
- **Team vector**
  - ***Average (proportion) of attributes within the team***

MBA Teams: Objective function

$$\min \sum_{team} \text{dist}(\text{team}, \text{target})$$

- **Target vector**
  - ***Average (proportion) of attributes within the class group***
- **Team vector**
  - ***Average (proportion) of attributes within the team***

- **Attributes**
  - ***Male/Female***
  - ***Scientist***
  - ***Contry***
  - ***IQ***
  - ***etc.***

Euclidian distance:
a non linear function

$$\text{target vector} = \frac{\sum_{i=1}^{26} (\text{attribute vector})_i}{26}$$

Euclidian distance:
a non linear function

$$\text{target vector} = \frac{\sum_{i=1}^{26} (\text{attribute vector})_i}{26}$$

$$(\text{team vector})_j = \frac{\sum_{i=1}^{26} (\text{attribute vector})_i X_{ij}}{n_j}$$

$$n_j = \begin{cases} 5 & j = 1..2 \\ 4 & j = 3..6 \end{cases}$$

Euclidian distance:
a non linear function

$$\text{target vector} = \frac{\sum_{i=1}^{26} (\text{attribute vector})_i}{26}$$

$$(\text{team vector})_j = \frac{\sum_{i=1}^{26} (\text{attribute vector})_i X_{ij}}{n_j}$$

$$n_j = \begin{cases} 5 & j = 1..2 \\ 4 & j = 3..6 \end{cases}$$

$$\min \sum_{j=1}^{6} \| \text{team}_j - \text{target} \|^2$$

MBA Teams

- **Some integrality difficulties in solving this *quadratic* transportation problem.**
- **Assume 70% males**

=> 2.8 in team of 4, 3.5 in team of 5

2 and 3 acceptable 2 and 3 acceptable

Solution procedure :

- ***complete enumeration of all acceptable team patterns; cost easily computed a priori.***

MBA Teams

- **Some integrality difficulties in solving this *quadratic* transportation problem.**
- **Moreover, assume 70% males**

=> 2.8 in team of 4, 3.5 in team of 5

2 and 3 acceptable 2 and 3 acceptable

Solution procedure :

– ***complete enumeration of all acceptable team patterns; cost easily computed a priori.***

MBA Teams

- **Some integrality difficulties in solving this *quadratic* transportation problem.**
- **Moreover, assume 70% males**

=> 2.8 in team of 4,   3.5 in team of 5

2 and 3 acceptable   2 and 3 acceptable

Solution procedure :

– ***complete enumeration of all acceptable team patterns; cost easily computed a priori.***

MBA Teams

- **Some integrality difficulties in solving this *quadratic* transportation problem.**
- **Moreover, assume 70% males**

=> 2.8 in team of 4, 3.5 in team of 5

2 and 3 acceptable 3 and 4 acceptable

Solution procedure :

- ***complete enumeration of all acceptable team patterns; cost easily computed a priori.***

MBA Teams

- **Some integrality difficulties in solving this *quadratic* transportation problem.**
- **Moreover, assume 70% males**

=> 2.8 in team of 4, 3.5 in team of 5

2 and 3 acceptable 3 and 4 acceptable

Solution procedure

– ***complete enumeration of all acceptable team patterns; cost easily computed a priori.***

MBA Teams

- **Some integrality difficulties in solving this *quadratic* transportation problem.**
- **Moreover, assume 70% males**

=> 2.8 in team of 4, 3.5 in team of 5

2 and 3 acceptable 3 and 4 acceptable

Solution procedure

- ***complete enumeration of all acceptable team patterns; cost easily computed a priori.***

A Secret Ballot Problem

	p_i	a	b	c	d	e	f
1	**14.3%**						
2	**13.2%**						
3	**12.4%**						
4	**8.4%**						
5	**7.8%**						
6	**6.2%**						
7	**5.7%**						
8	**5.5%**						
9	**4.5%**						
10	**4.2%**						
11	**3.6%**						
12	**3.1%**						
13	**2.7%**						
14	**2.4%**						
15	**1.5%**						
16	**1.4%**						
17	**1.3%**						
18	**1.1%**						
19	**0.4%**						
20	**0.3%**						
v_j		**35.9%**	**11.1%**	**17.4%**	**17.3%**	**13.8%**	**4.5%**

Can you decode the vote?

	p_i	a	b	c	d	e	f
1	**14.3%**						
2	**13.2%**						
3	**12.4%**						
4	**8.4%**						
5	**7.8%**						
6	**6.2%**						
7	**5.7%**						
8	**5.5%**						
9	**4.5%**						
10	**4.2%**						
11	**3.6%**						
12	**3.1%**						
13	**2.7%**						
14	**2.4%**						
15	**1.5%**						
16	**1.4%**						
17	**1.3%**						
18	**1.1%**						
19	**0.4%**						
20	**0.3%**						
	v_j	**35.9%**	**11.1%**	**17.4%**	**17.3%**	**13.8%**	**4.5%**

Generalized Assignment Formulation

$$X_{ij} = \begin{cases} 1 & \text{if } i \text{ voted for } j \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{j} X_{ij} = 1 \qquad i = 1..20$$

$$\sum_{i=1}^{20} p_i X_{ij} = v_j \qquad j \in \{a, b, \ldots, f\}$$

Generalized Assignment Formulation

$$X_{ij} = \begin{cases} 1 & i \text{ voted for } j \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{j} X_{ij} = 1 \qquad\qquad i = 1..20$$

$$\sum_{i=1}^{20} p_i X_{ij} = v_j \qquad j \in \{a, b, \ldots, f\}$$

- **No objective function**
- **No integrality gap to consider**

Generalized Assignment Formulation

$$X_{ij} = \begin{cases} 1 & i \text{ voted for } j \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{j} X_{ij} = 1 \qquad i = 1..20$$

$$\sum_{i=1}^{20} p_i X_{ij} = v_j \qquad j \in \{a, b, \ldots, f\}$$

- **No objective function**
- **No integrality gap to consider**

Solution procedure

- **An infinite number of *fractional* solutions**
- **4.5% = 0.1*14.3% + 0.23257576*13.2% etc.**

Solution procedure

- **An infinite number of *fractional* solutions**
- **4.5% = 0.1*14.3% + 0.23257576*13.2% etc.**
- **A very limited number of *integer* solutions**
  - *4.5% : 10 combinations*
  - *35.9% : 12 combinations*
  - *…*

Complete enumeration

Solution procedure

- **An infinite number of *fractional* solutions**
- **4.5% = 0.1*14.3% + 0.23257576*13.2% etc.**
- **A very limited number of *integer* solutions**
  - ***4.5% : 10 combinations only***
  - ***35.9% : 12 combinations***
  - **…**

Complete enumeration

Solution procedure

- **An infinite number of *fractional* solutions**
- **4.5% = 0.1*14.3% + 0.23257576*13.2% etc.**
- **A very limited number of *integer* solutions**
  - ***4.5% : 10 combinations***
  - ***35.9% : 12 combinations***
  - **…**

Complete enumeration

Solution procedure

- **An infinite number of *fractional* solutions**
- **4.5% = 0.1*14.3% + 0.23257576*13.2% etc.**
- **A very limited number of *integer* solutions**
  - ***4.5% : 10 combinations***
  - ***35.9% : 12 combinations***

Full enumeration and Set Partitioning Formulation

Set Partitioning Formulation

$$Y_k = \begin{cases} 1 & \text{if voting pattern } k \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

$$a_{ik} = \begin{cases} 1 & \text{if } i \text{ votes in pattern } k \\ 0 & \text{otherwise} \end{cases}$$

$$b_{kj} = \begin{cases} 1 & \text{if pattern } k \text{ sums to } v_j \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_k a_{ik} Y_k = 1 \qquad i = 1..20$$

$$\sum_k b_{kj} Y_k = 1 \qquad j \in \{a, b, \ldots, f\}$$

- **No objective function**
- **No integrality gap to consider**

		a	b	c	d	e	f	
1	**14.3%**	**1**						1
2	**13.2%**	**1**						1
3	**12.4%**				**1**			1
4	**8.4%**			**1**				1
5	**7.8%**					**1**		1
6	**6.2%**		**1**					1
7	**5.7%**	**1**						1
8	**5.5%**			**1**				1
9	**4.5%**				**1**			1
10	**4.2%**					**1**		1
11	**3.6%**		**1**					1
12	**3.1%**						**1**	1
13	**2.7%**	**1**						1
14	**2.4%**			**1**				1
15	**1.5%**					**1**		1
16	**1.4%**						**1**	1
17	**1.3%**		**1**					1
18	**1.1%**			**1**				1
19	**0.4%**				**1**			1
20	**0.3%**					**1**		1
		0.0%	**100.0%**	**0.0%**	**0.0%**	**100.0%**	**0.0%**	

Conclusion

- **The presented Set Partitioning / Set Covering formulations can be derived from network-based formulations by applying the appropriate Dantzig-Wolfe decomposition process.**
- **This allows to benefit from the well structured patterns by using delayed or a priori column generation procedures and at the same time to get rid of non linear functions that appear in the objective or the constraints.**

Conclusion

- **The presented Set Partitioning / Set Covering formulations can be derived from network-based formulations by applying the appropriate Dantzig-Wolfe decomposition process.**
- **This allows to benefit from the well structured patterns by using delayed or a priori column generation procedures and at the same time to get rid of non linear functions that appear in the objective or in the constraints.**

Thanks for your attention