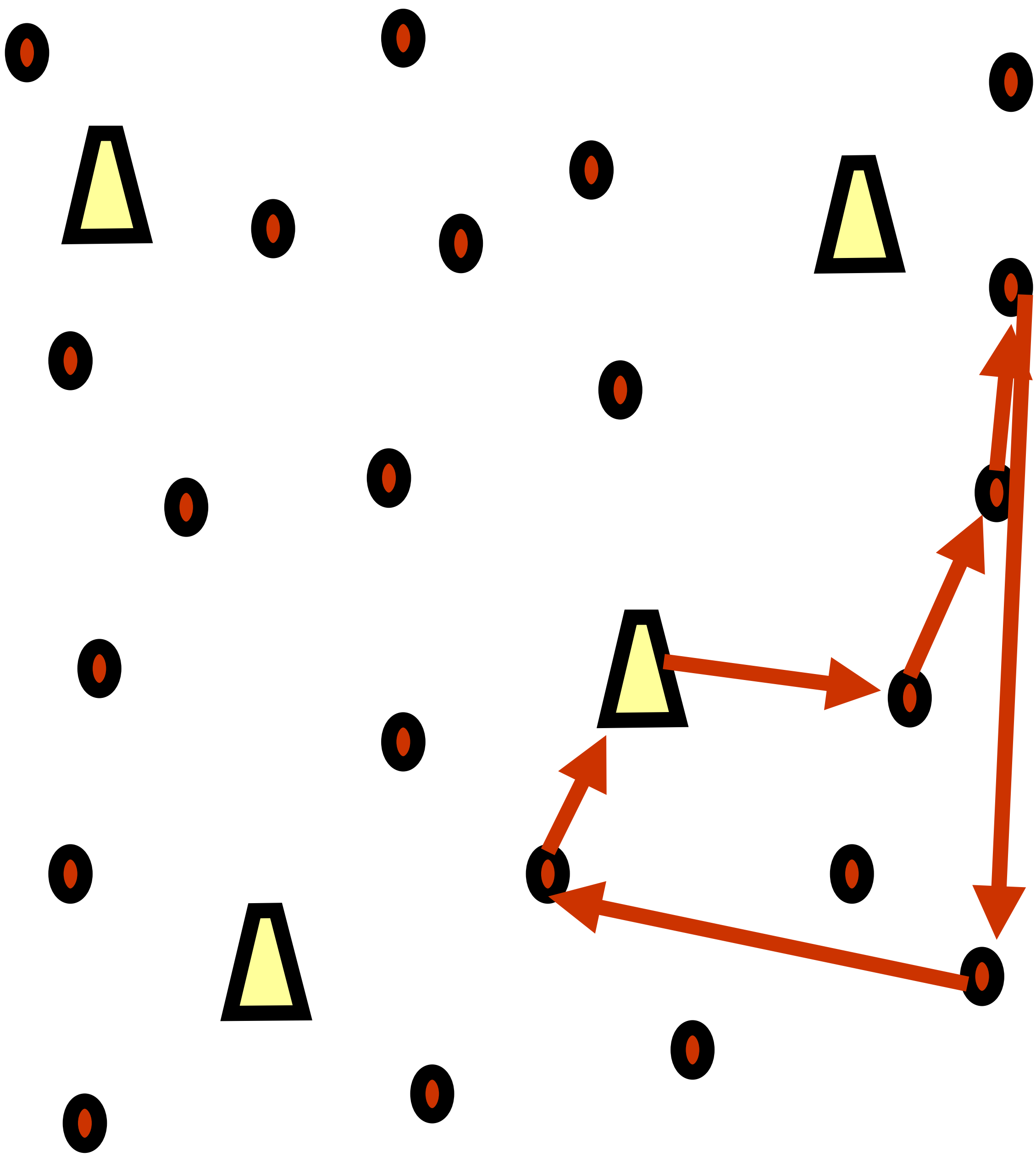
STABILIZED COLUMN GENERATION

Jacques Desrosiers
HEC* & *GERAD

Hatem Ben Amor
François Soumis
Daniel Villeneuve
GERAD

Multiple Depot Vehicle Scheduling Problem



MDVSP

K : set of depots

N : set of scheduled tasks (no time windows)

t_i : time at which service must start for task $i \in N$

s_i : service duration for task $i \in N$

d_{ij} : time to go from node i to node j, for $i, j \in N \cup K$

A : set of arcs; $(i, j) \in A$ if $t_i + s_i + d_{ij} \leq t_j$, for $i, j \in N \cup K$

Ω^k : set of paths from and to depot $k \in K$

c_p^k : cost of path $p \in \Omega^k$

$a_{ip}^k = 1$ if path $p \in \Omega^k$ covers task $i \in N$; 0 otherwise

n^k : available number of vehicles at depot $k \in K$

$$\min \sum_{k \in K} \sum_{p \in \Omega^k} c_p^k x_p^k$$

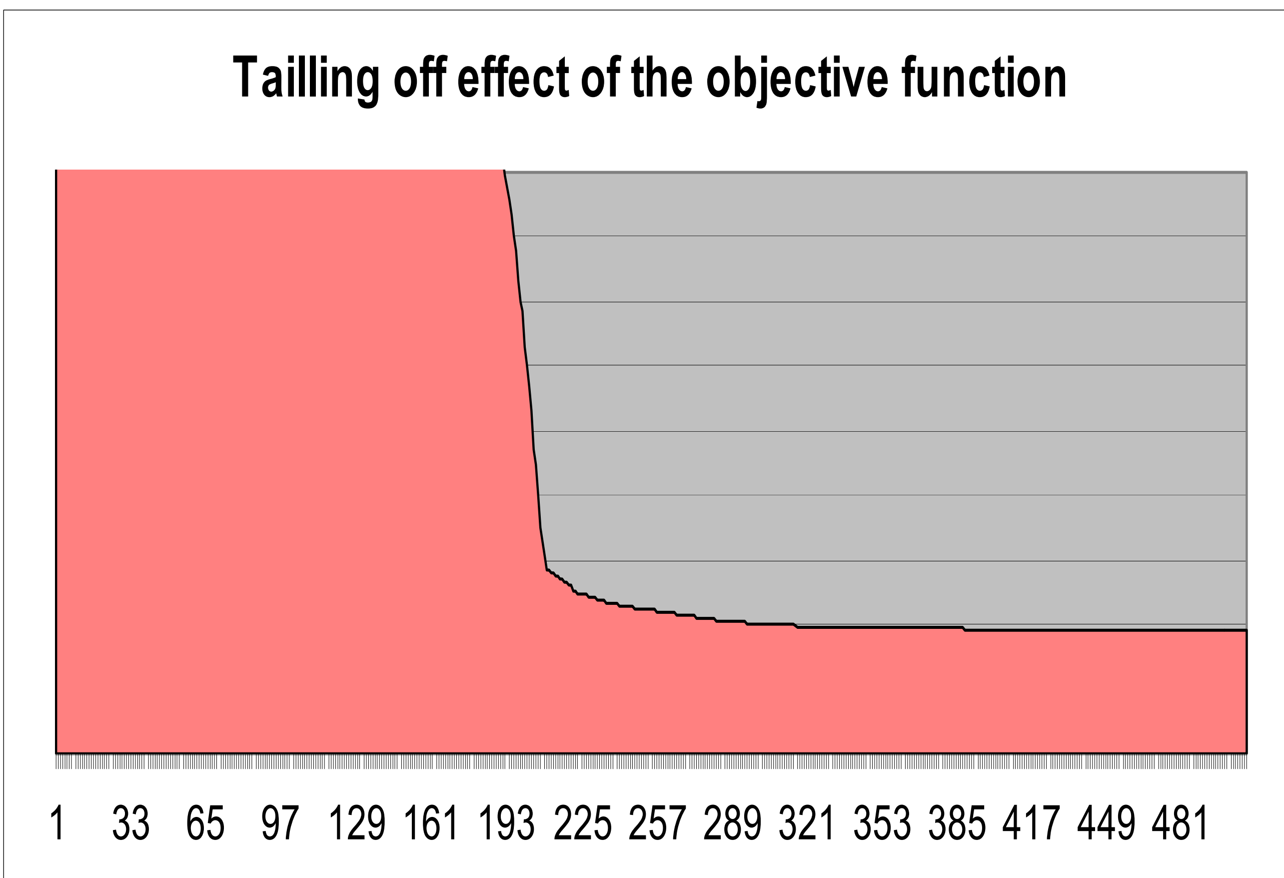
$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip}^k x_p^k = 1, \qquad i \in N$$

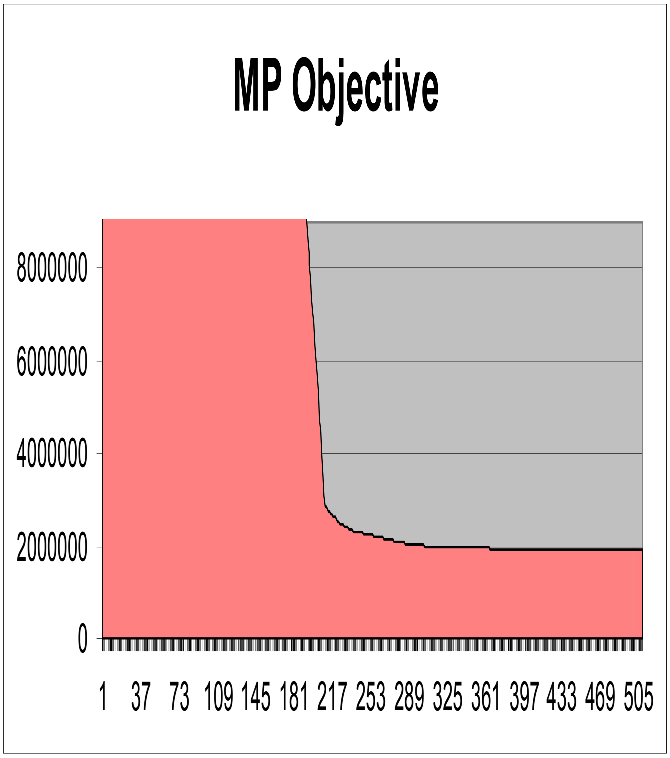
$$\sum_{p \in \Omega^k} x_p^k \leq n^k, \qquad k \in K$$

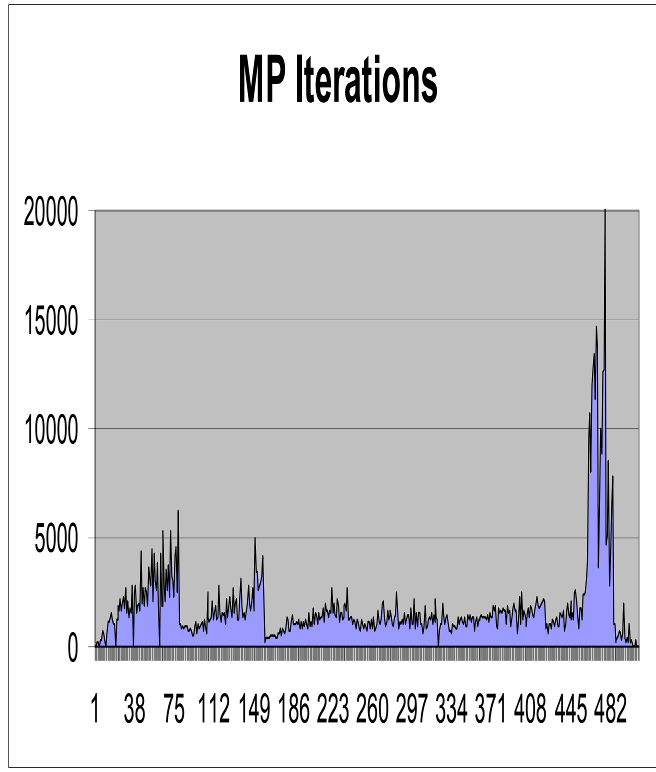
$$x_p^k \text{ binary}, \qquad k \in K, p \in \Omega^k$$

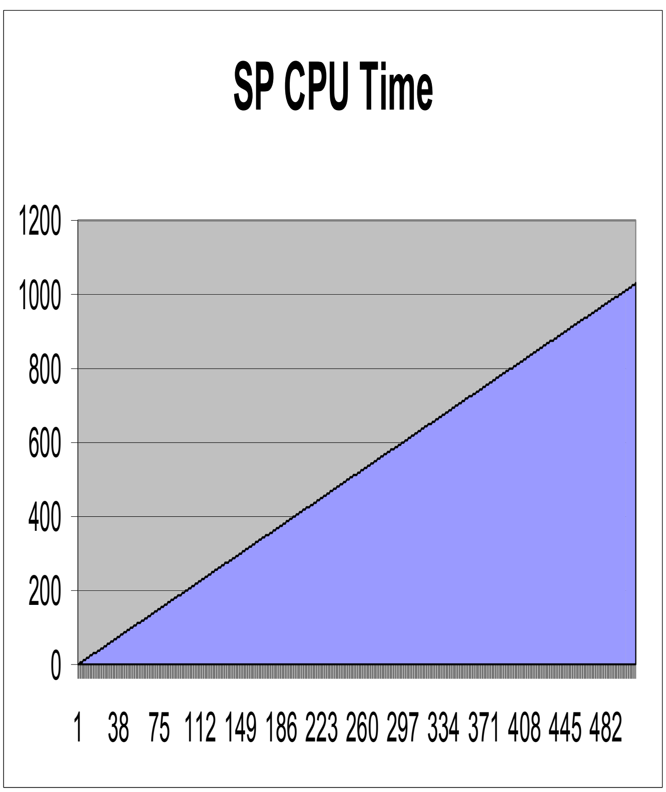
Motivation

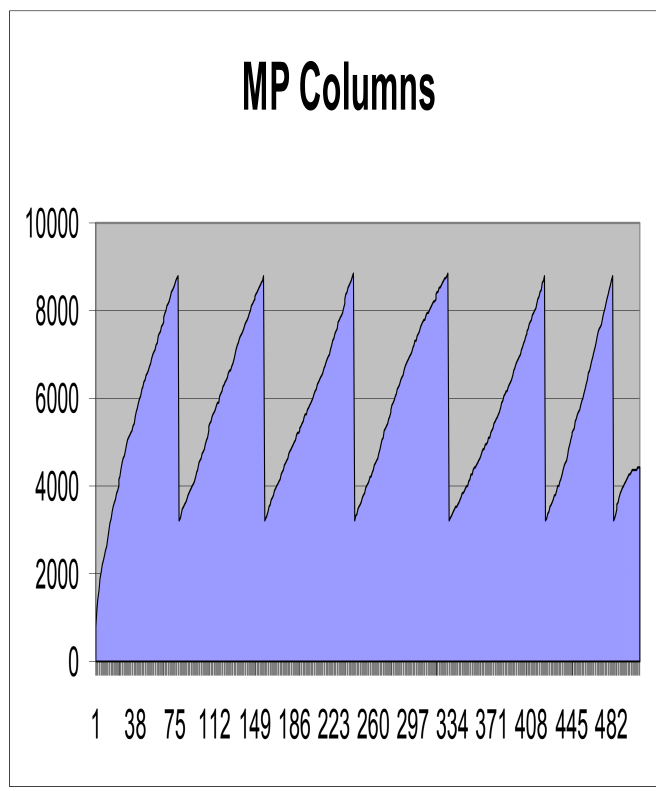
MDVSP R 800 (4)	cpu tot	cpu mp	cpu sp	# CG iter	# SP cols	# MP itr
standard	**4178.4**	3149.2	1029.2	**509**	37579	926161




MP Objective
8000000
6000000
4000000
2000000
0
1 37 73 109 145 181 217 253 289 325 361 397 433 469 505

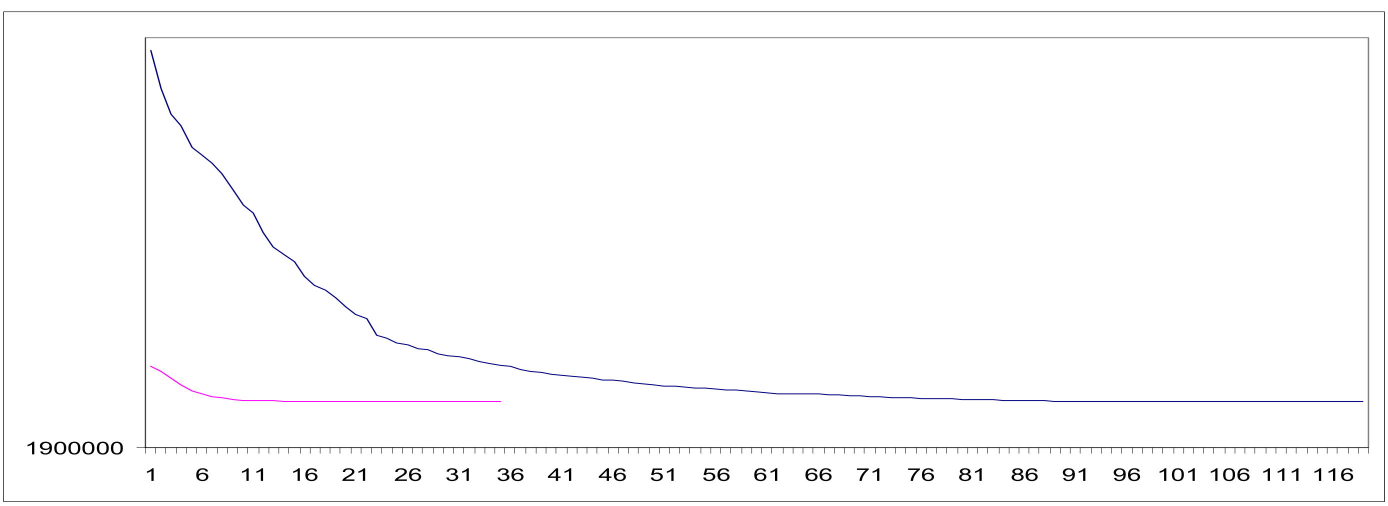

MP Iterations
20000
15000
10000
5000
0
1 38 75 112 149 186 223 260 297 334 371 408 445 482


SP CPU Time
1200
1000
800
600
400
200
0
1 38 75 112 149 186 223 260 297 334 371 408 445 482


MP Columns
10000
8000
6000
4000
2000
0
1 38 75 112 149 186 223 260 297 334 371 408 445 482

Using Optimal Dual Values

Problem R 800 (4)	Opt sol	Init sol	cpu tot	# CG iter	# SP cols	# MP itr
standard	1915589.5	800000000	***4178.4***	**509**	37579	926161
delta 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
0.01	1915589.1	1915602.5	***37.3***	**17**	1145	6288
delta boxes	Final gap(%)	Initial gap (%)	***(results in % vs standard set***			
100		6.26	***20.0***	23.4	24.9	30.1
10		0.63	***2.8***	6.9	7.4	4.4
1		0.063	***1.2***	3.9	3.8	0.9
0.1		0.0063	***1.1***	3.7	3.5	0.9
0.01	-0.00002088	0.0007	***0.9***	3.3	3.0	0.7



Optimal Dual Values

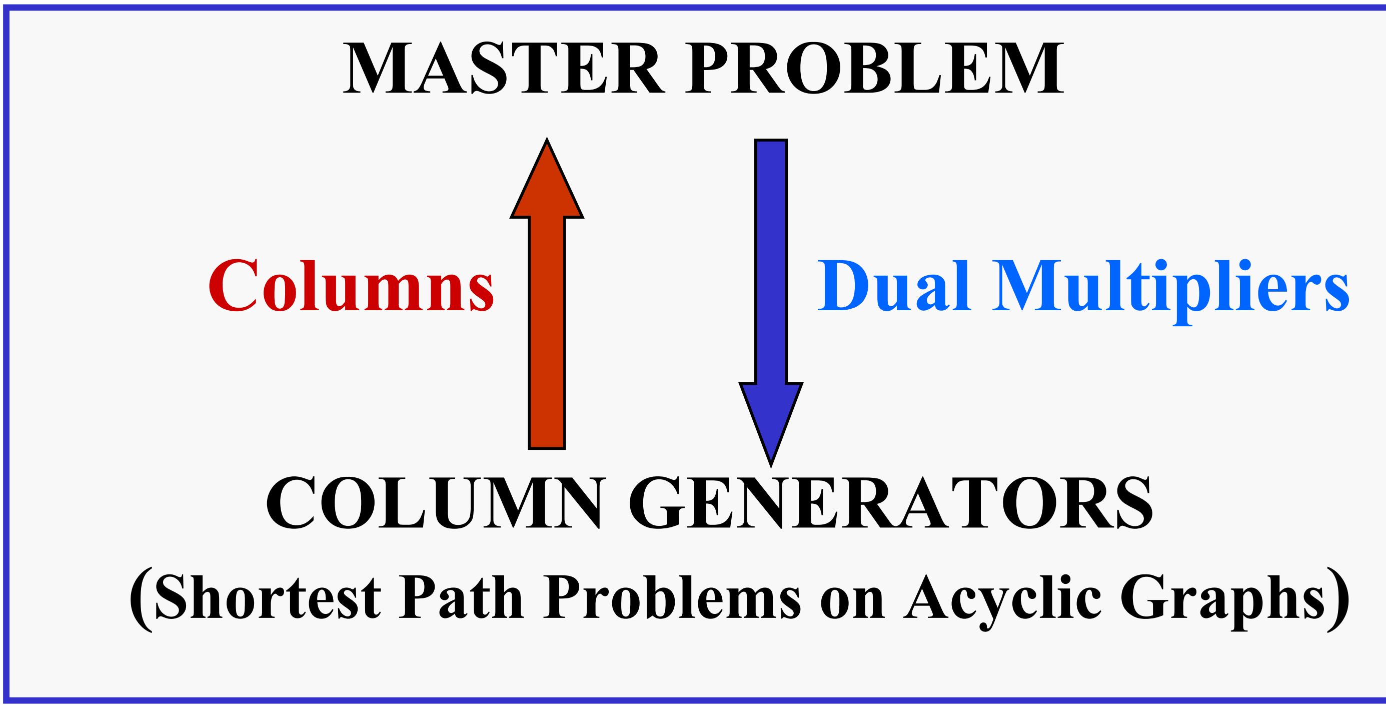
- **Useful in the context of Lagrangian Relaxation to recover primal feasibility**

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

Crossover: CPLEX vs STABILIZATION (cpu times in seconds)

	Problem 1	Problem 2	Problem 3	Problem 4	Problem 5
Constraints	12,354	12,313	13,344	13,453	13,269
Variables	126,329	129,349	151,665	156,841	148,040
CPLEX Primopt	> 10000	> 10000	> 10000	> 10000	> 10000
CPLEX Baropt	1,056	1,011	1,342	1,382	1,171
CROSSOVER					
CPLEX Promopt	13	3,427	> 10000	97	6,659
CPLEX Dualopt	50	697	1,876	686	778
Stabilazation 10^{-1}	189	291	671	631	417
Stabilazation 10^{-2}	100	121	554	557	380
Stabilazation 10^{-3}	91	102	407	396	353
Stabilazation 10^{-4}	87	92	350	347	285

LP Column Generation



***Optimality Conditions*:**

primal feasibility
dual feasibility
complementary slackness

Some References

1960 Gillmore & Gomory	Cutting Stock Problem
1989 Agarwal, Mathur & Salkin	VRP
1963 Marquardt	Trust region
1975 Marsten	Box Step
2000 Kallehauge & Madsen	VRPTW
1992 Vial & Goffin	ACCPM
1999 du Merle et al.	Stabilized CG
2000 Valério de Carvalho	Cutting Stock Problem

Impact of Dual Bounds on the Primal Formulation

Primal (P)

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

Dual (D)

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

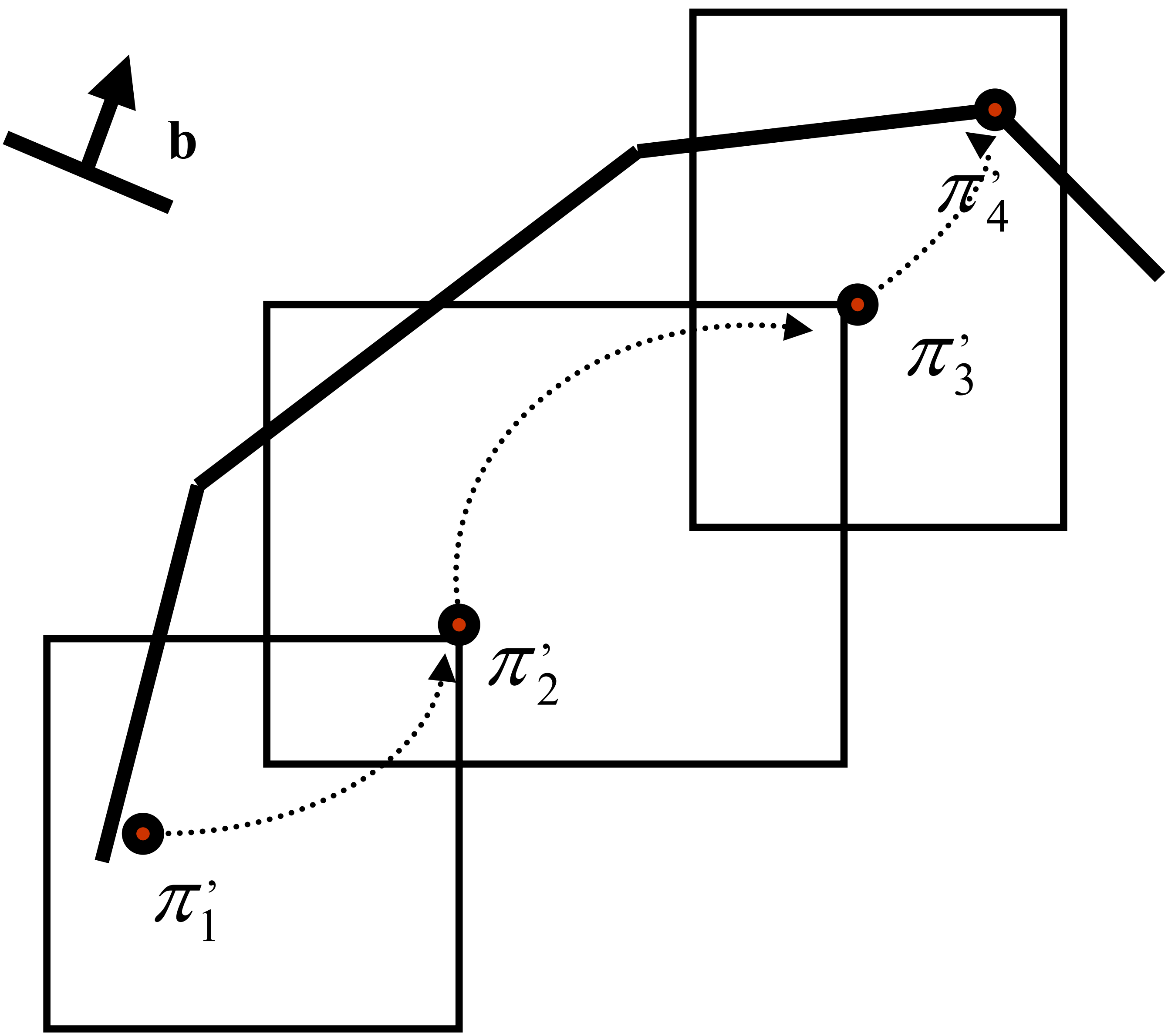
***Restricted* Dual (D^d)**

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

***A Relaxed* Primal (P^d)**

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

Trust Regions / Box Step



Dual Cuts
for the Cutting Stock problem

Valério de Carvalho (2000)

"Using extra dual cuts to accelerate column generation"

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:

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

Using a priori at most 2m dual constraints

(or primal columns)

…

Dual Cuts / Primal Columns

Valério de Carvalho (2000)

Application to the Cutting Stock Problem

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

reduces the CPU time by 40%.

Triplets (501 items)

each roll of length L
is cut into exactly three orders without any waste

Standard Column Generation	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
Both strategies	11.2 iterations

Average over 10 problems

Degeneracy & Perturbation

Primal (P)

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

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

***Perturbed* Primal**

(P_e)

A relaxed primal

Stabilized Column Generation

$$\min\ cx \quad Ax = b \quad x \ge 0$$

$$\max\ b\pi \quad \pi A \le c$$

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

Perturbed Primal

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

Restricted Dual

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

Stabilized Primal (P_ε^d)

Stabilized Primal & Dual

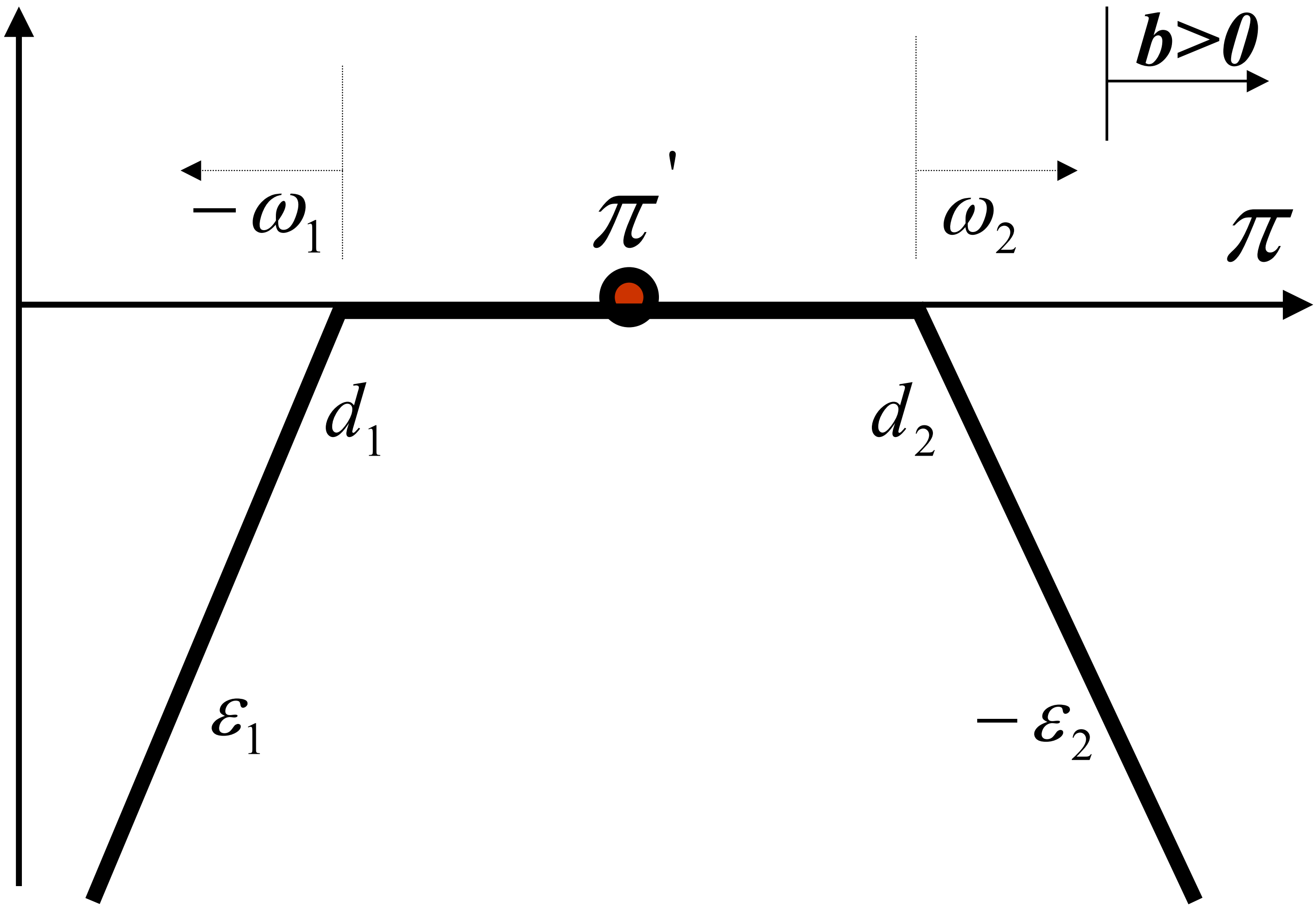
$$\begin{aligned}
\min \quad & cx - d_1 y_1 + d_2 y_2 & \\
& Ax - y_1 + y_2 = b & \pi \\
& y_1 \le \varepsilon_1 & -\omega_1 \le 0 \\
& y_2 \le \varepsilon_2 & -\omega_2 \le 0 \\
& x \ge 0,\ y_1 \ge 0,\ y_2 \ge 0 &
\end{aligned}$$

Stabilized Primal (P_ε^d)

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

Stabilized Dual (D_ε^d)

Dual Space



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

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

Propositions

Within a column generation scheme, problem P_ε^d is optimal when $\bar{c}_p \geq 0$, for all columns $p \in \Omega$

Since P_ε^d is a relaxation of P

$$cx - d_1 y_1 + d_2 y_2 \;=\; b\pi - \varepsilon_1 \omega_1 - \varepsilon_2 \omega_2$$

is a lower bound on $v(P)$.

$$b\pi$$

is a better lower bound on $v(P)$.

Propositions

$$\text{Problem } P \equiv P_\varepsilon^d \quad \text{if}$$

$$\varepsilon = 0, \text{ or } \quad \exists\, \pi^* \mapsto d_1 < \pi^* < d_2$$

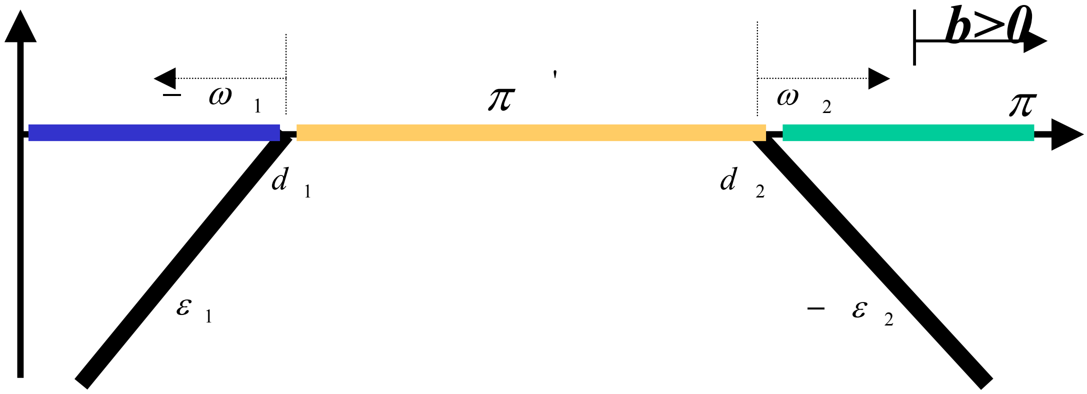
Within a column generation scheme, problem P is optimal while solving P_ε^d if

$$y_1 = y_2 = 0$$

and

$$\bar{c}_p \geq 0, \text{ for all columns } p \in \Omega$$

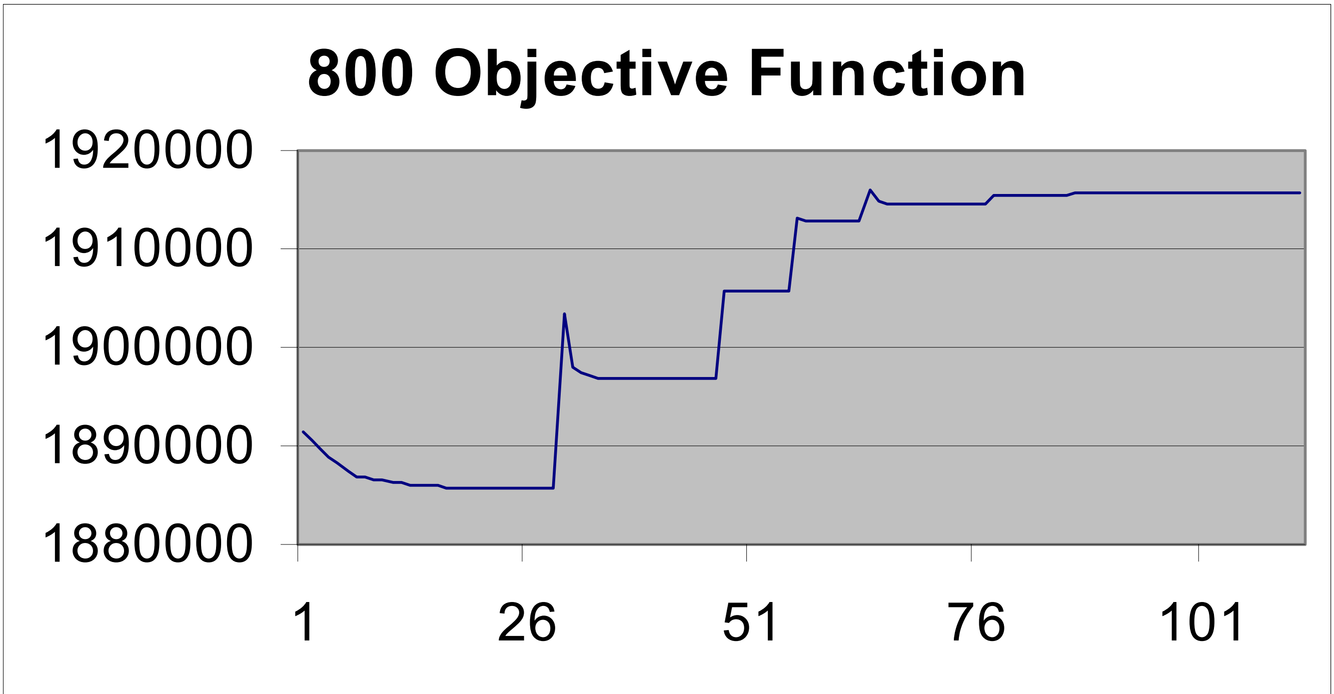
Parameter Adjustment, by component



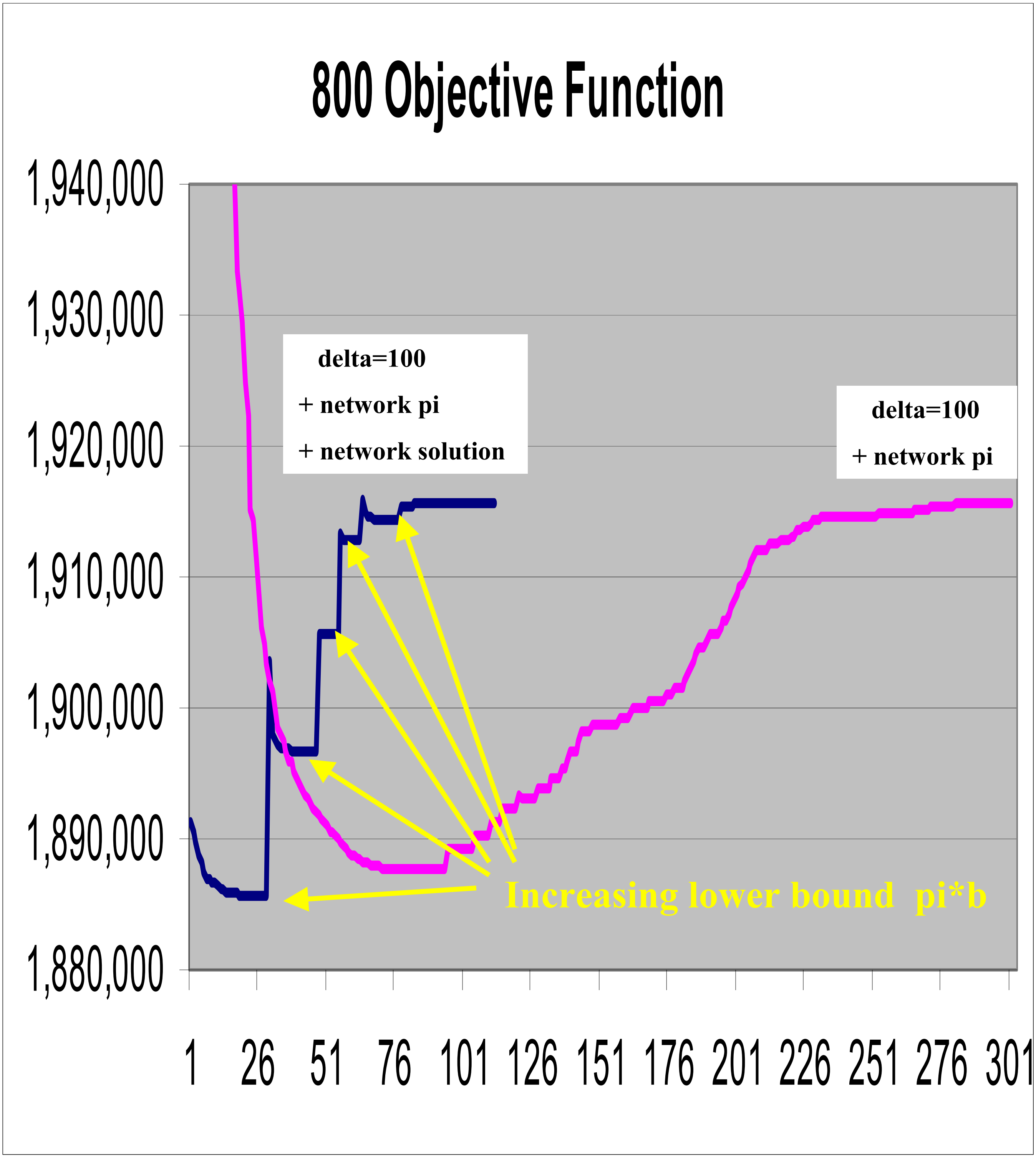
If the dual value is too small,
re-center and enlarge the interval.

If dual value is within the interval,
re-center and reduce the interval.

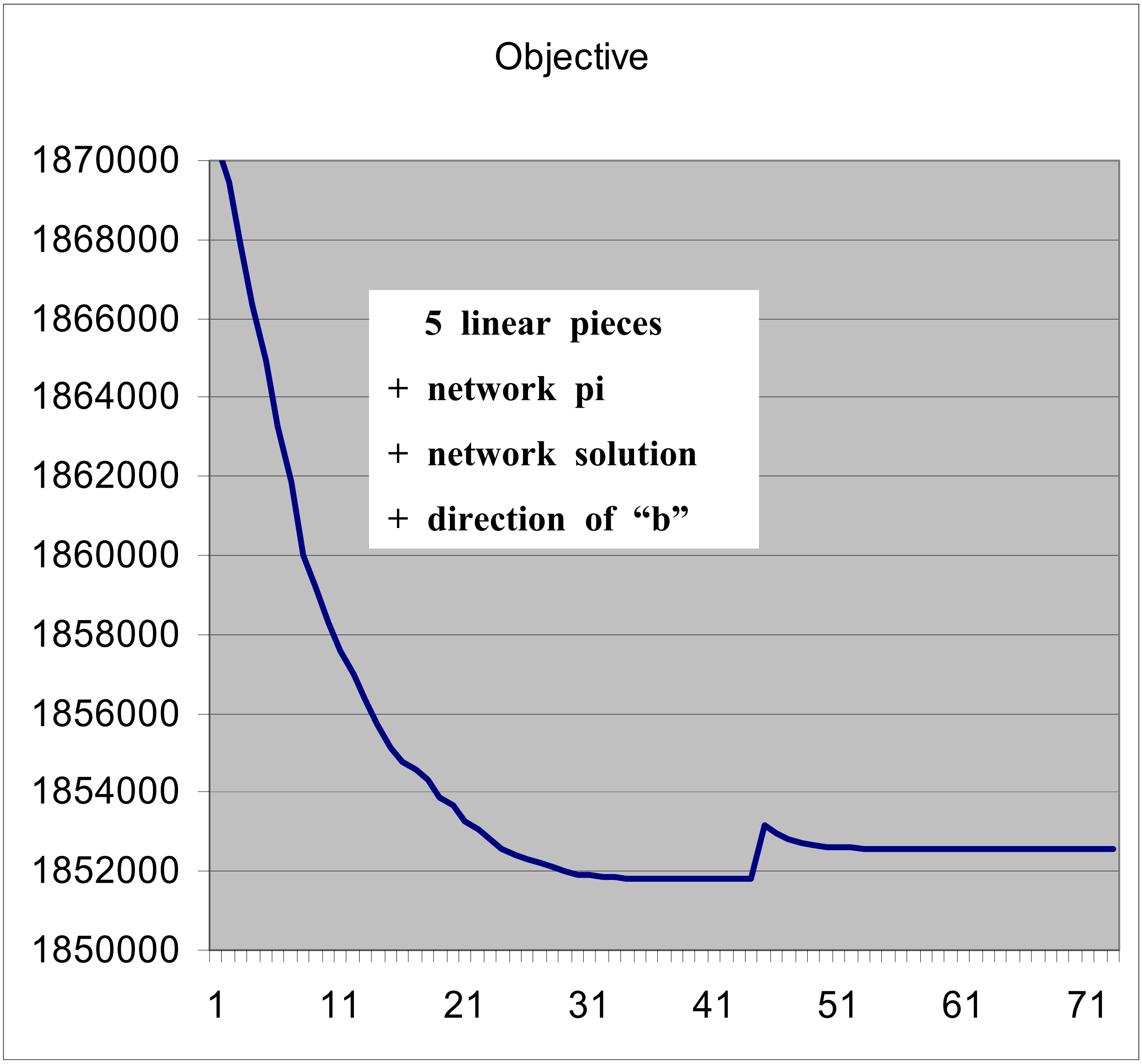
If the dual value is too large,
re-center and enlarge the interval.



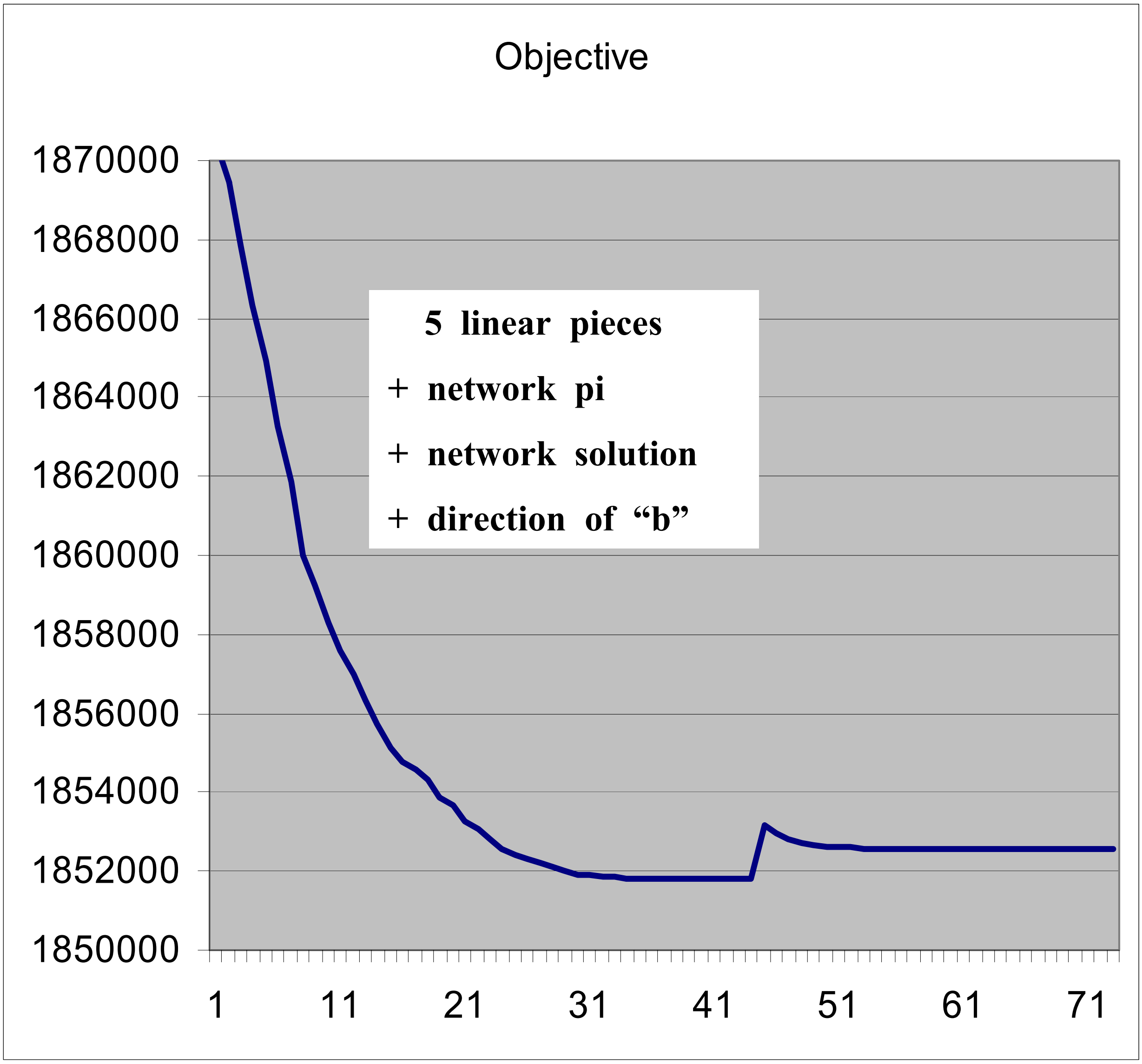
Problem *R800 (4)*	**Opt sol**	**Init sol**	***cpu tot***	**cpu mp**	**cpu sp**	**# CG iter**	**# SP cols**	**# MP itr**
standard	1915589.5	800000000	***4178.4***	3149.2	1029.2	**509**	37579	926161
delta = 100		2035590.5	***835.5***	609.1	226.4	**119**	9368	279155
network pi		2014429.8	***1097.1***	518.5	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					



800 Objective Function
1,940,000
1,930,000
1,920,000
1,910,000
1,900,000
1,890,000
1,880,000
delta=100
+ network pi
+ network solution
delta=100
+ network pi
Increasing lower bound pi*b
1
26
51
76
101
126
151
176
201
226
251
276
301



Problem P 800 (4)	Opt sol	Init sol	*cpu tot*	cpu mp	cpu sp	# CG iter	# SP cols	# MP itr
standard	1852571.5	800000000	***3562***	886	2676	**422**	33280	672726
		1870487.8	***268***	111	157	**73**	4616	79860
		% reduction	***92.5***	***87.5***	***94.1***	***82.7***	***86.1***	***88.1***
		13.3	***times faster***					



Opt sol	Init sol	*cpu tot*	cpu mp	cpu sp	# CG iter	# SP cols	# MP itr
1852571.5	800000000	***3562***	886	2676	**422**	33280	672726
	1870487.8	***241***	110	131	**63**		
	% reduction	***93.2***	***87.6***	***95.1***	***85.1***		
	14.8	***times faster***					