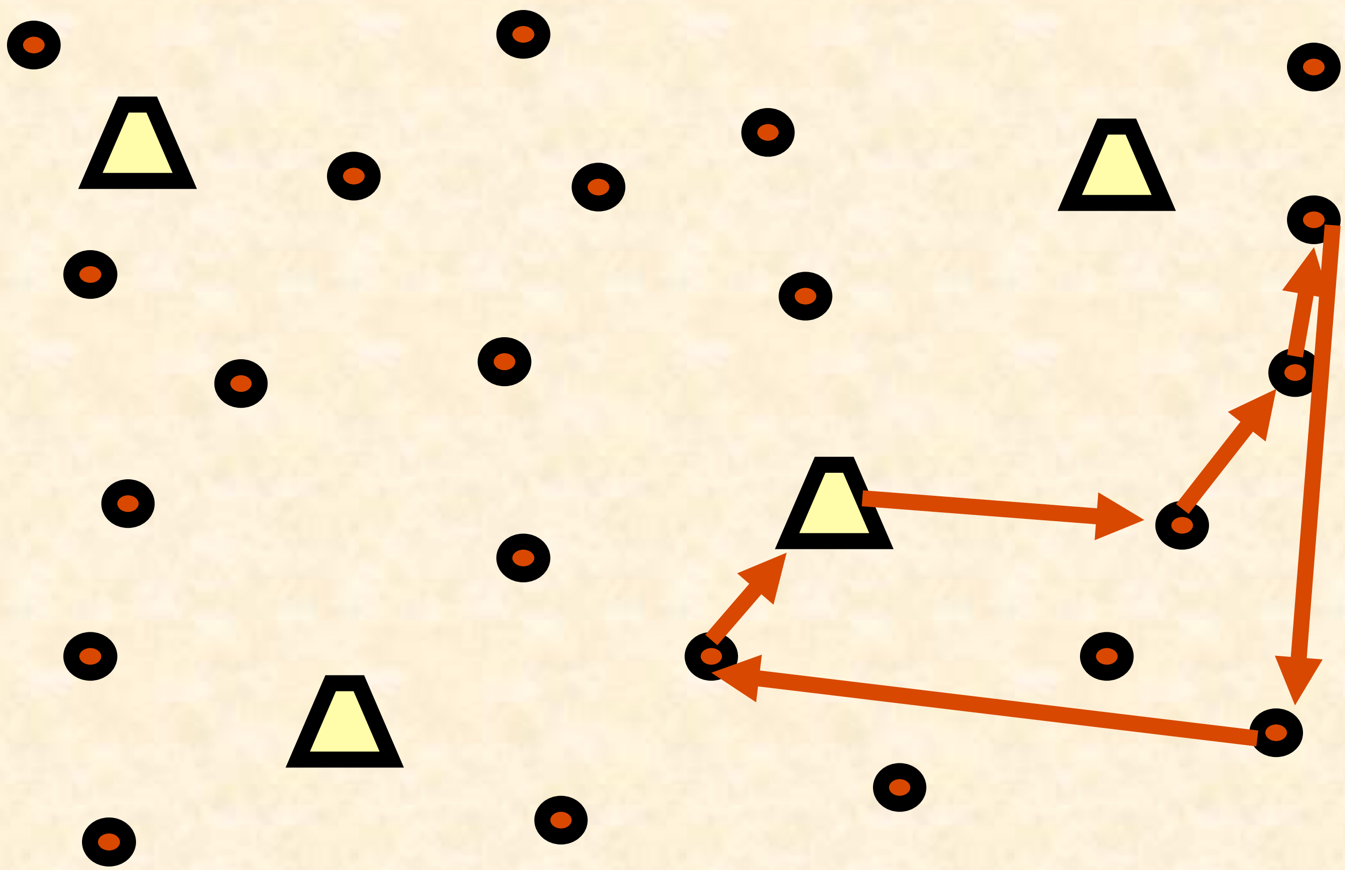
School on Column Generation: *Stabilized Column Generation*

Jacques Desrosiers
HEC Montréal & GERAD, Canada

Guy Desaulniers
Polytechnique Montréal & GERAD, Canada

Marco Lübbecke
RWTH Aachen University, Germany

MULTIPLE DEPOT VEHICLE SCHEDULING PROBLEM (MDVSP)



MDVSP: NOTATION

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 $i \in N$

t_{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 $t_i + s_i + t_{ij} \leq t_j$, 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

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

MDVSP: Set Partitioning Formulation

$$Z_M = \min \sum_{k \in K} \sum_{p \in \Omega^k} c_p^k \lambda_p^k$$

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

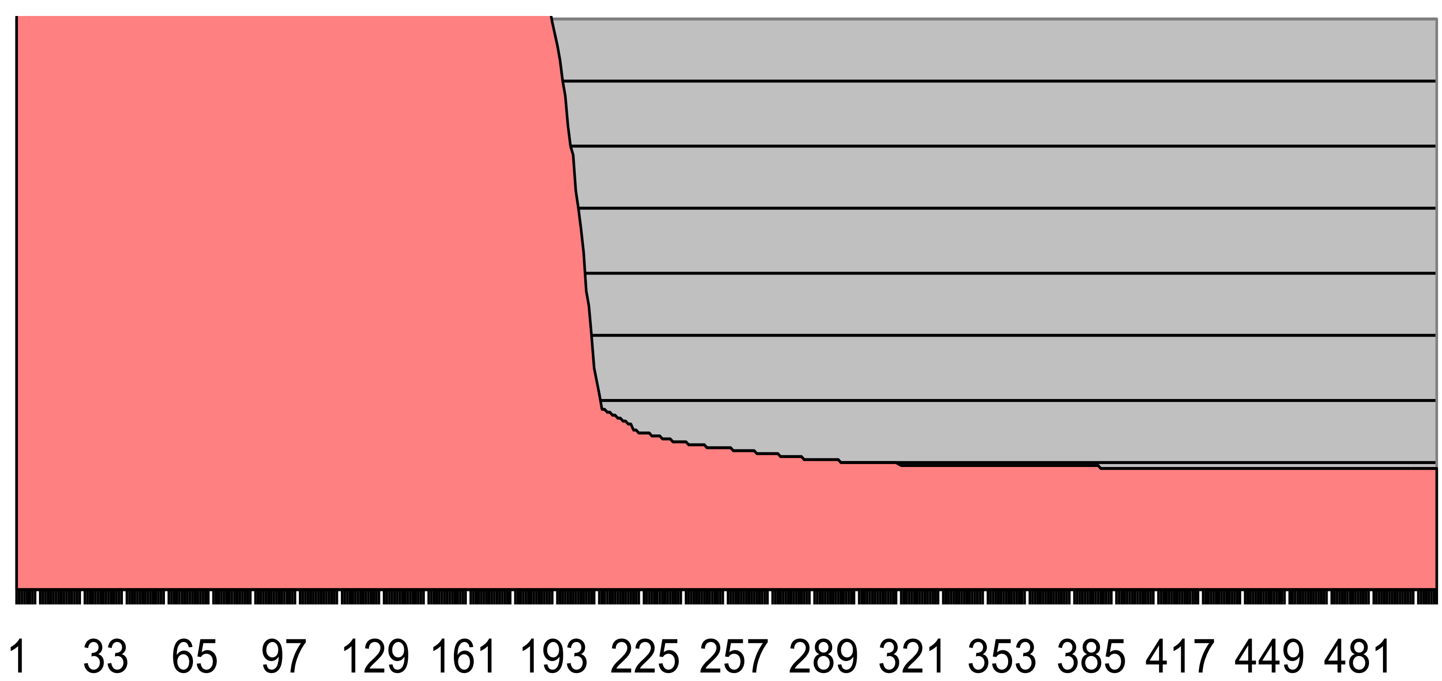
$$\sum_{p \in \Omega^k} \lambda_p^k \le v^k, \qquad k \in K$$

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

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

Tailling off effect of the objective function



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

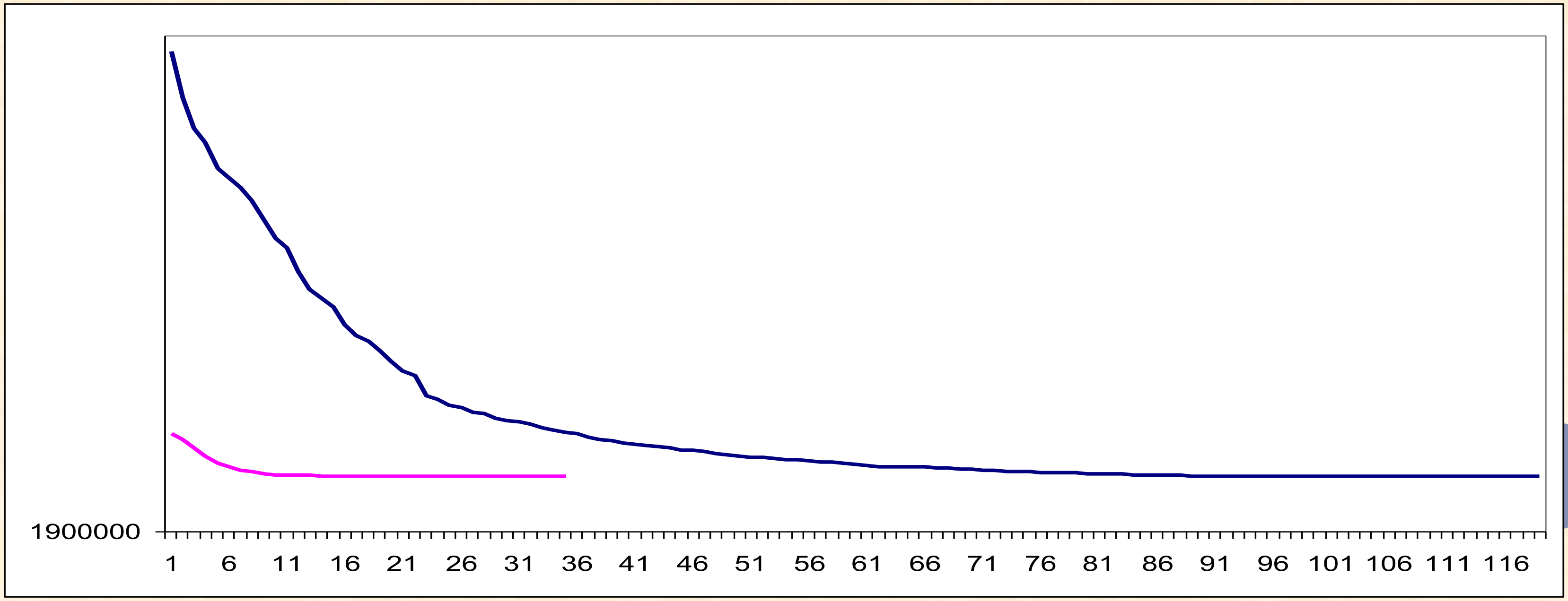
- **Master problem requires more than 75% of total cpu time**

Impact of Perfect Dual Information on CG

Problem R800 (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
0.01	1915589.1	1915602.5	***37.3***	**17**	1145	6288

IMPACT OF PERFECT DUAL INFORMATION ON CG

Problem R800 (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



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:**

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

At most 2m dual constraints (or primal columns) inserted a priori.

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
$\pi_i = \ell_i / L, \quad i = 1,\ldots,m$	12.2 iterations
(Average over 10 problems)	

Perfect dual information

LP Column Generation

MASTER PROBLEM

Columns

Dual Multipliers

COLUMN GENERATOR

Stopping rule?

<u>Optimality</u> of Dual Multipliers

DUAL BOXES

Primal

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

$\longleftrightarrow$

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

Dual

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

$\longleftrightarrow$

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

Relaxed **Primal**

Restricted **Dual**

Surplus & Slack Variables

DEGENERACY & PERTURBATION

Primal

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

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

***Relaxed* Primal**

Alternative

***Perturbed* Primal**

Surplus & Slack Variables

Perturbation & Dual Boxes

Primal

$$\boxed{\begin{array}{r} \min cx \\ Ax = b \\ x \geq 0 \end{array}}$$

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

Relaxed **Primal**

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

Relaxed **Primal**

STABILIZED PRIMAL PROBLEM

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

Primal

P

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

Stabilized **Primal**

SP

STABILIZED PRIMAL & DUAL PROBLEMS

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

Stabilized **Primal SP**

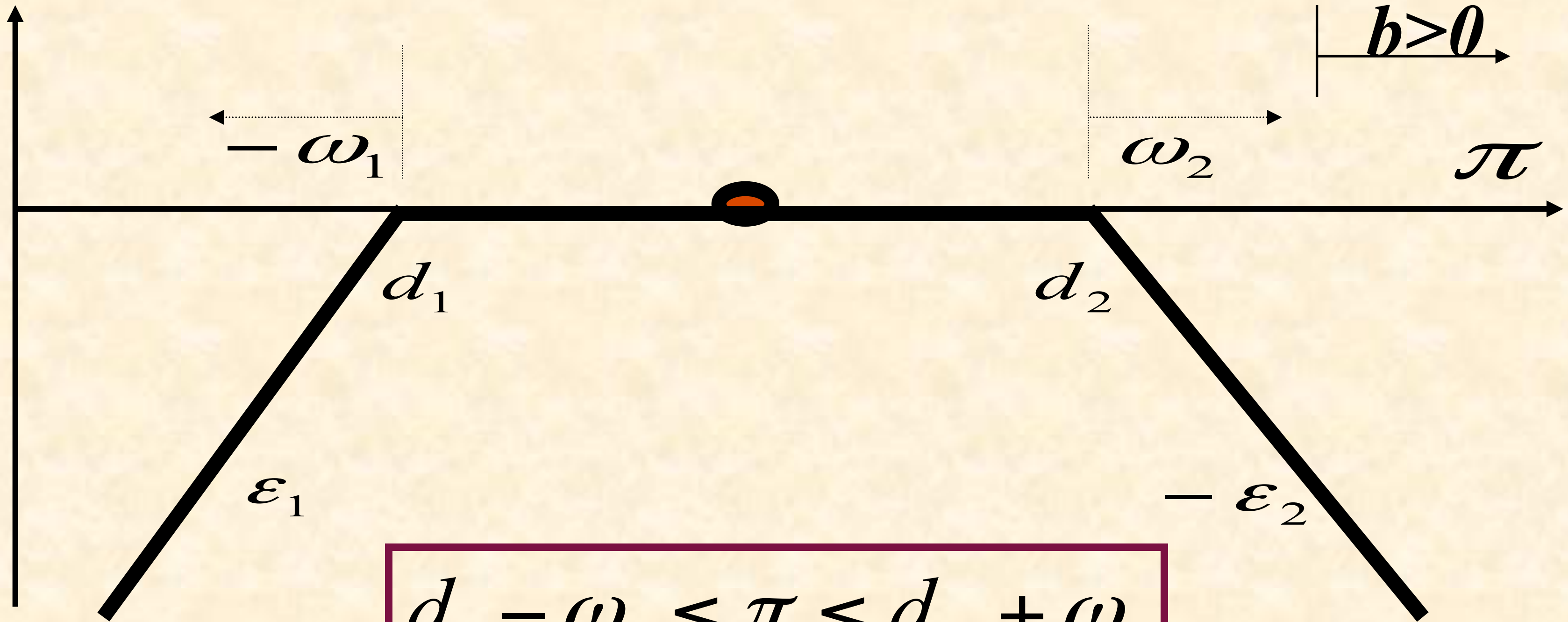
$$
\begin{aligned}
& \max\ b\pi \\
& \pi A \le c
\end{aligned}
$$

Dual D

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

Stabilized **Dual SD**

INTERPRETATION IN DUAL SPACE



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

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

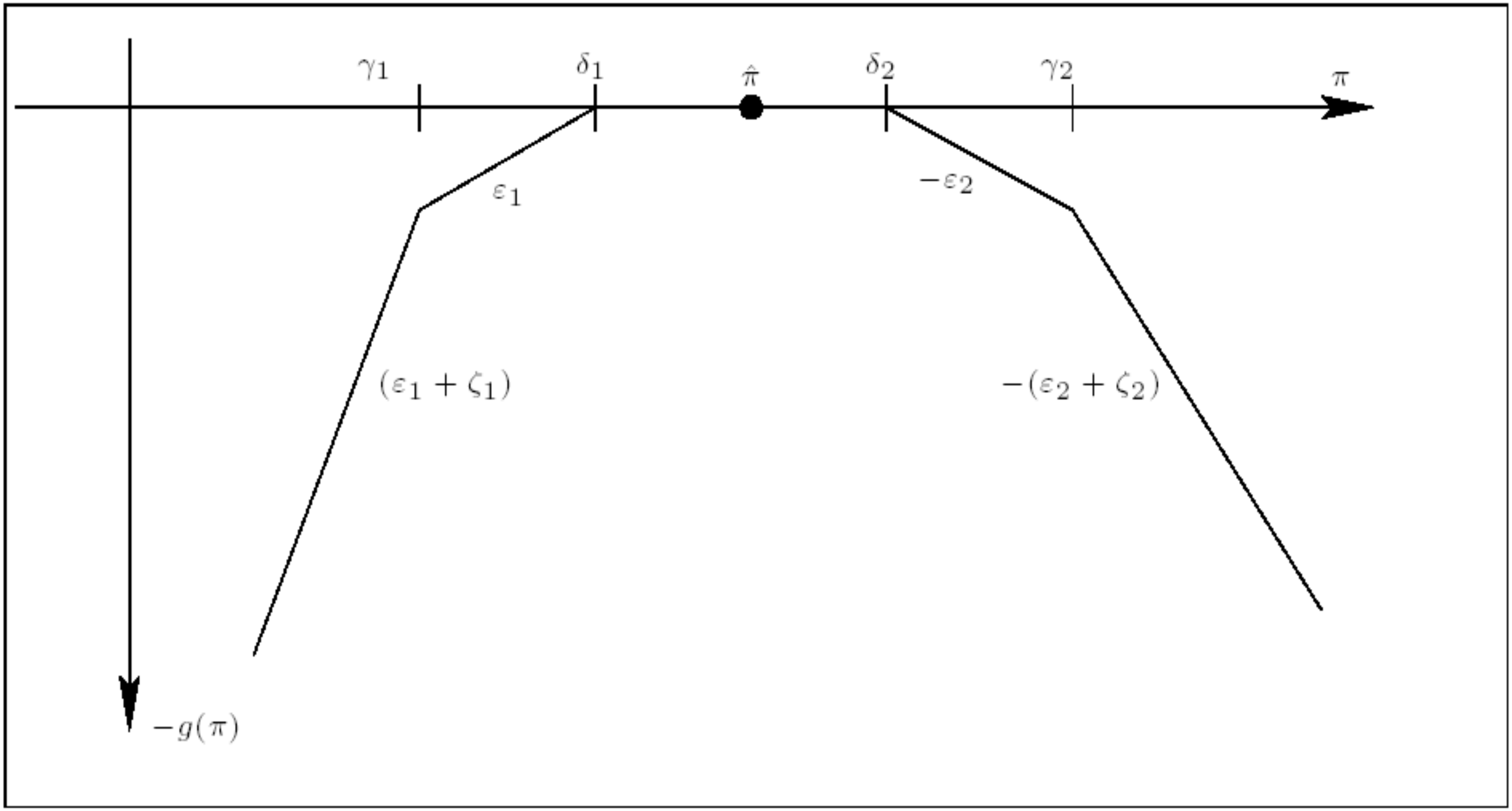


Figure 1: 5-piecewise linear dual penalty function

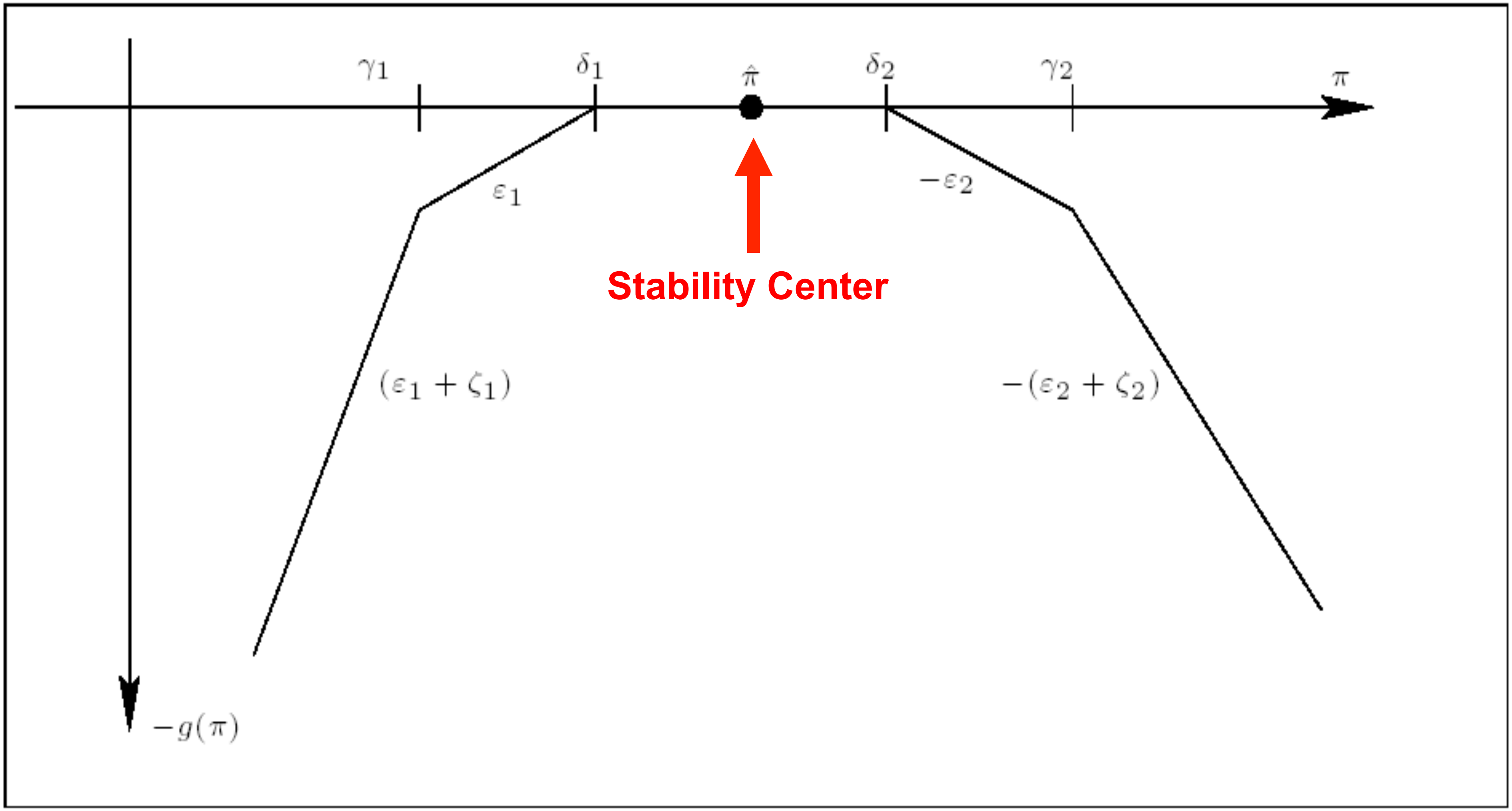


Figure 1: 5-piecewise linear dual penalty function

MDVSP: Solution Strategy

1. Preprocessing
   - Assignment problem: Single depot approximation
     - Lower bound Z_L
     - Optimal number of vehicles v^*
   - Transportation problem to derive a primal integer solution
     - Upper bound Z_U
   - Arc elimination based reduced cost greater than $Z_U - Z_L$
   - Estimation of dual multipliers for all tasks

2. Stabilization procedure

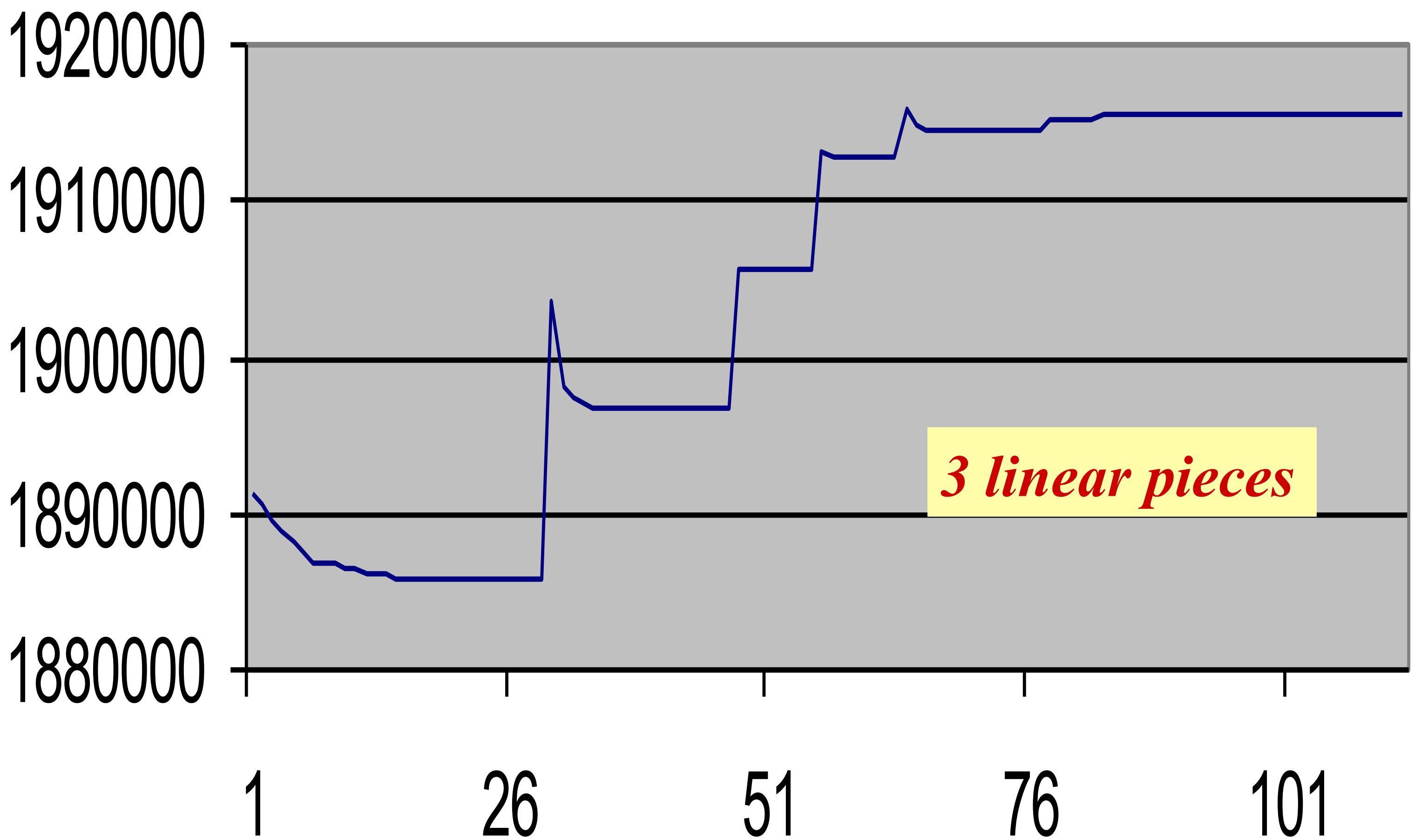
STABILIZATION PROCEDURE FOR PROBLEM P

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

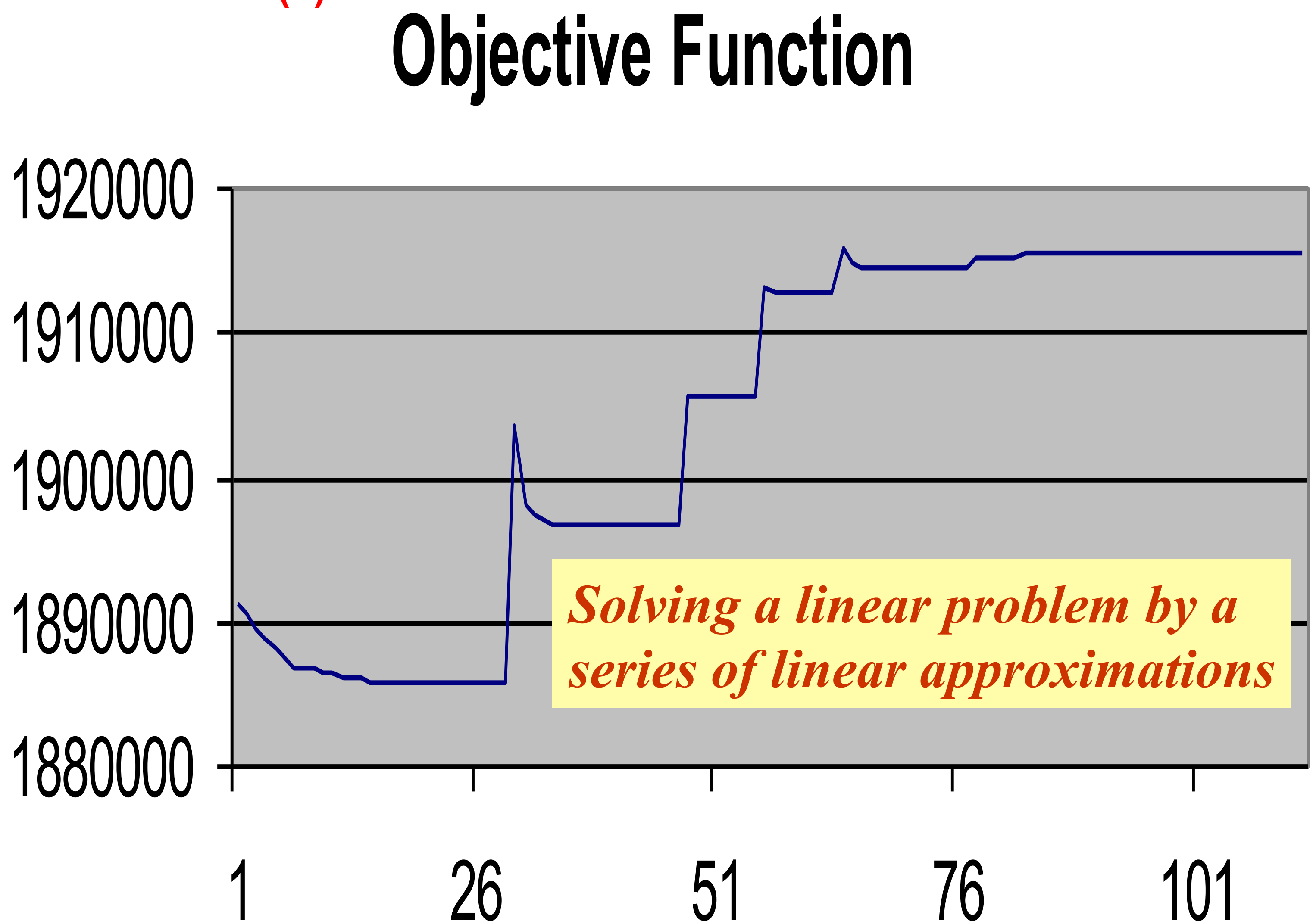
- Solve stabilized problems SP & SD until P is feasible
  - Otherwise update problems SP & SD
    - Stability center, trust region and penalties

Problem R800 (4)

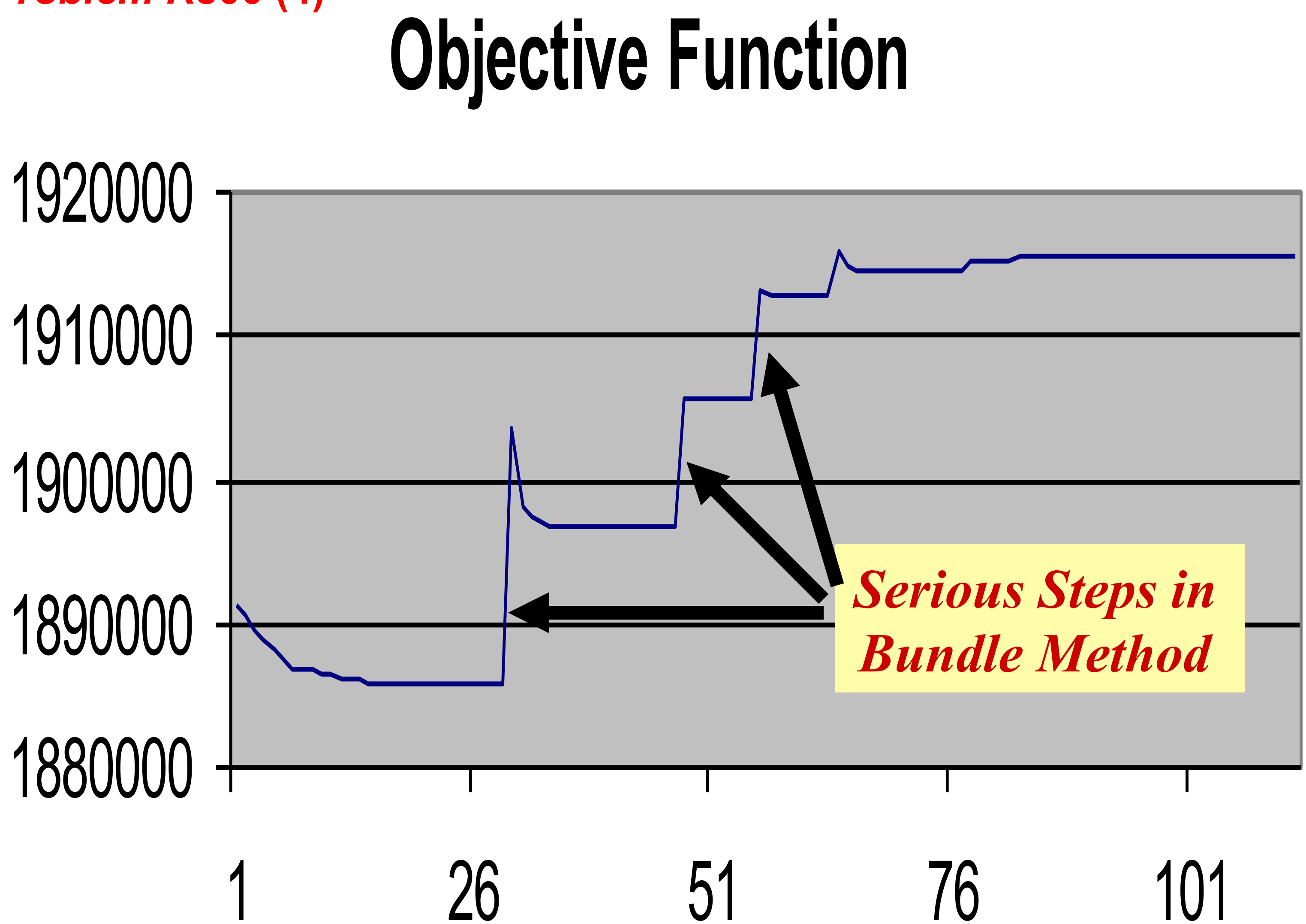
Objective Function

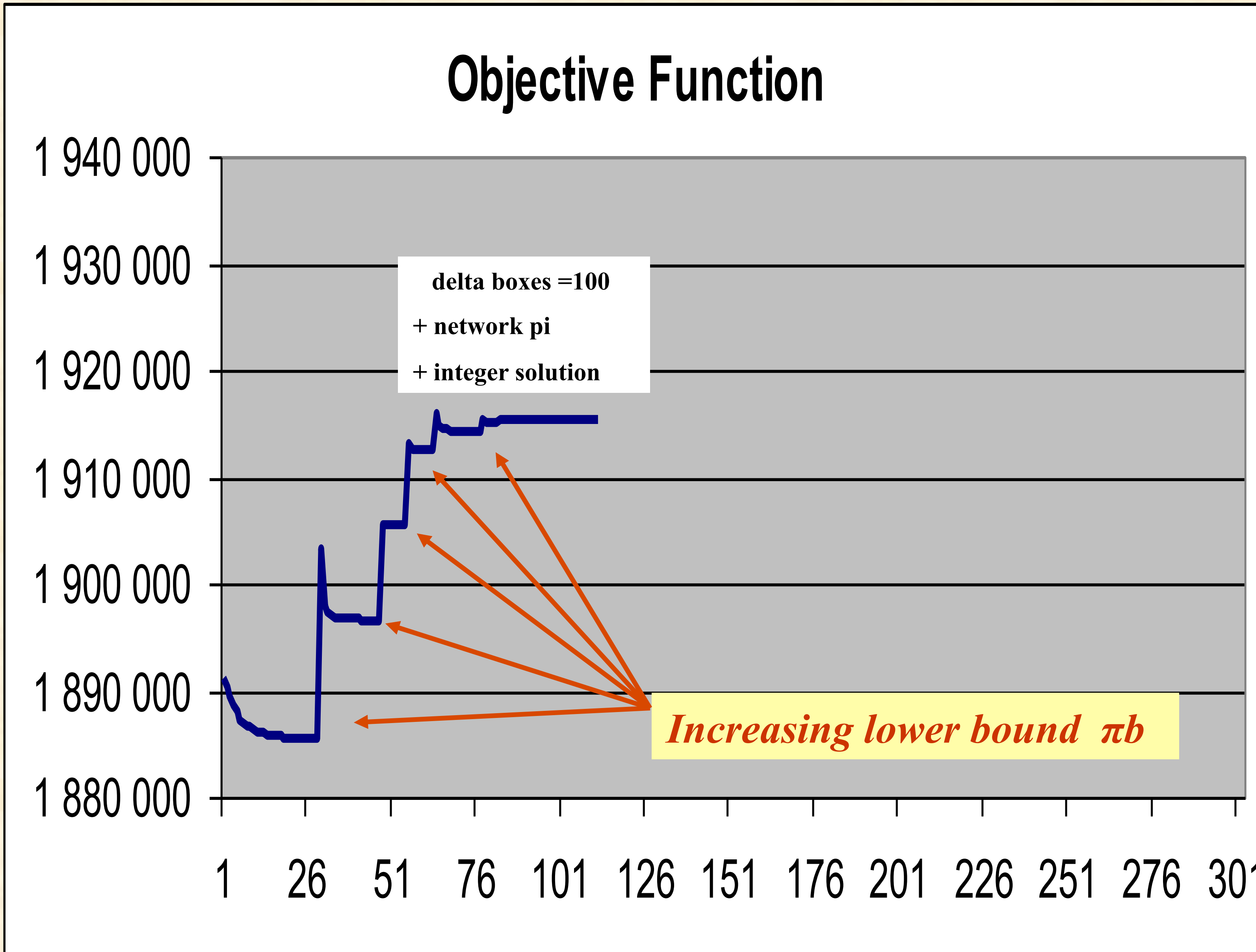


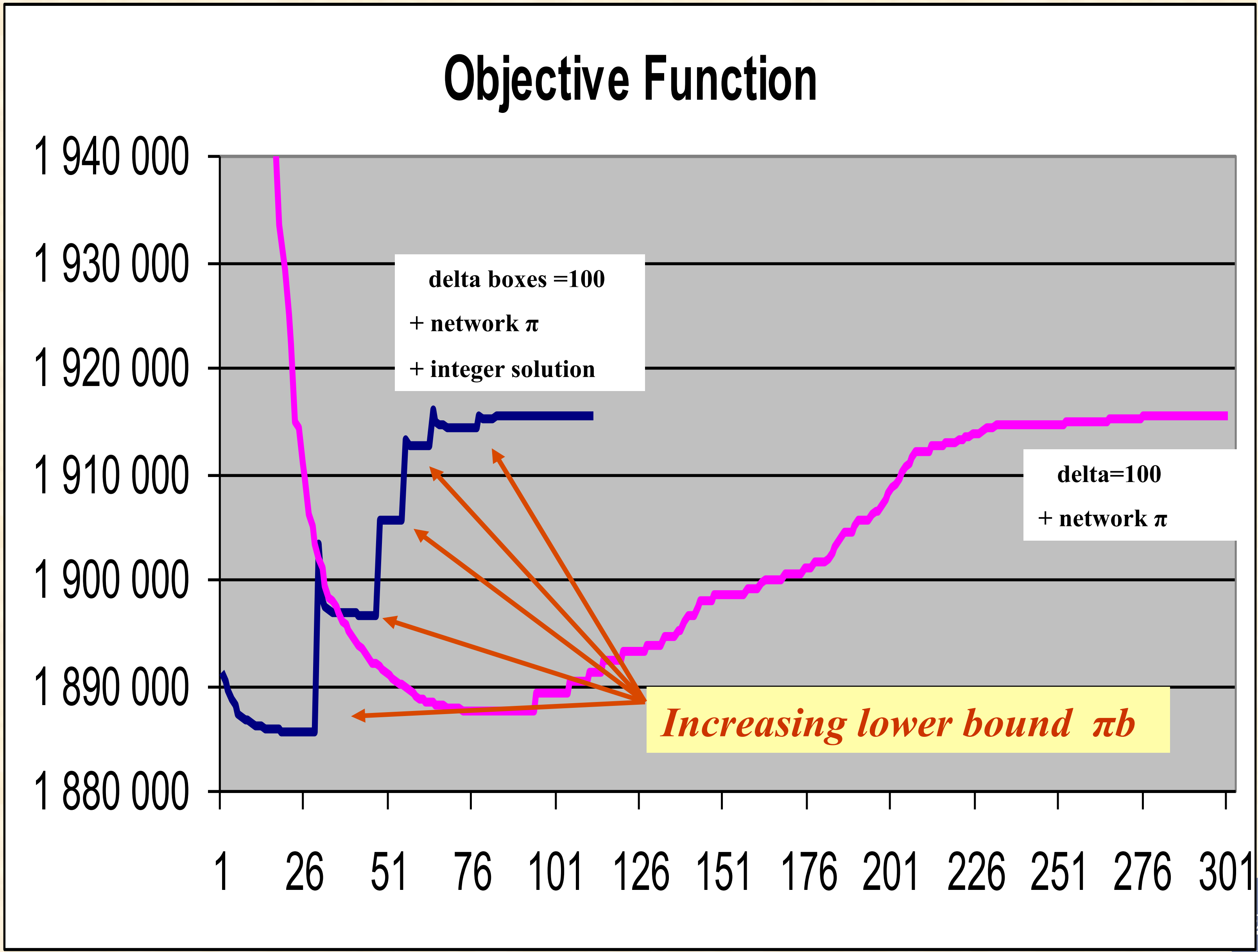
Problem R800 (4)

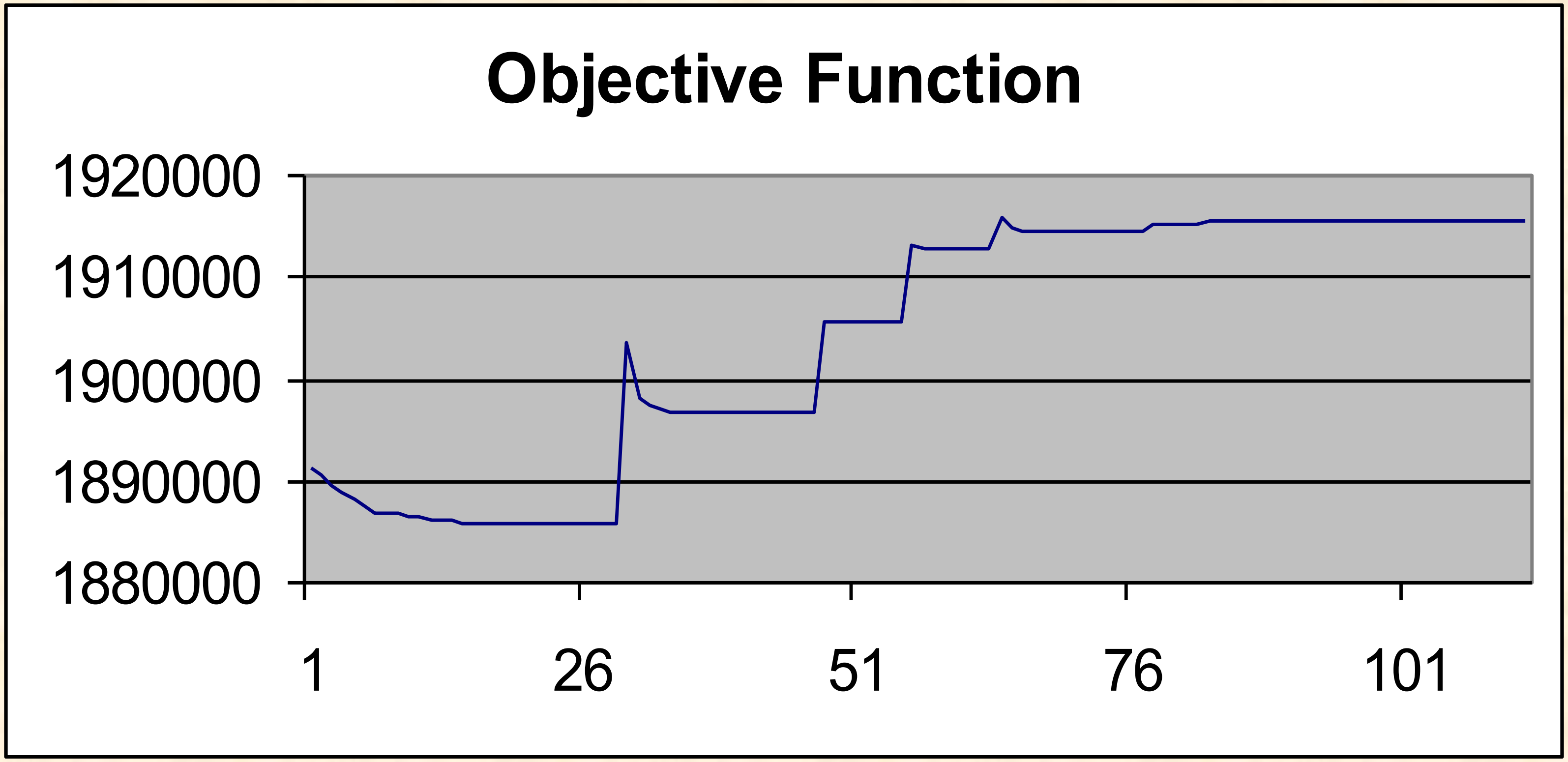


Problem R800 (4)

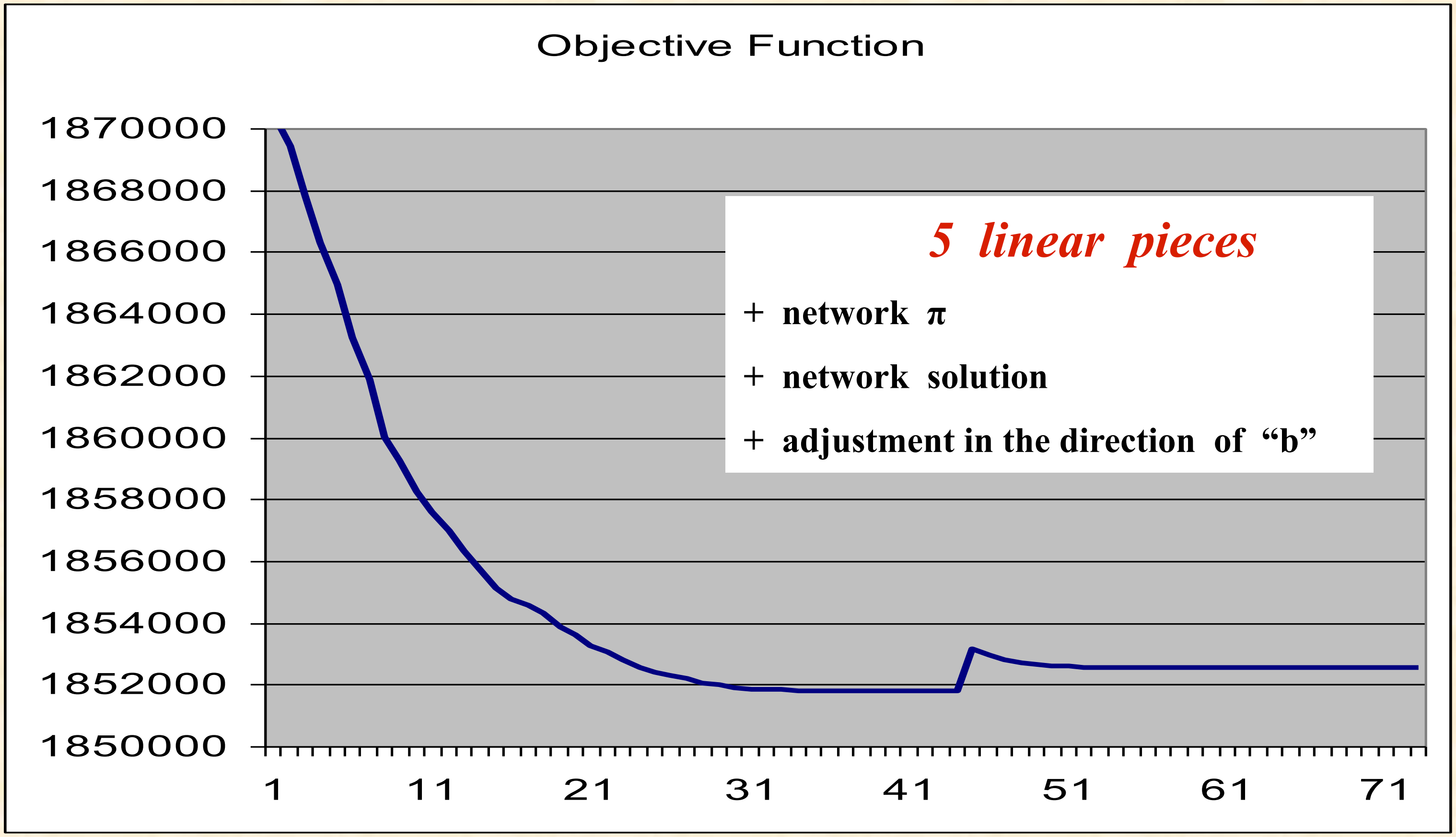



Objective Function
1 940 000
1 930 000
1 920 000
1 910 000
1 900 000
1 890 000
1 880 000
1
26
51
76
101
126
151
176
201
226
251
276
301
delta boxes =100
+ network pi
+ integer solution
Increasing lower bound πb


Objective Function
1 940 000
1 930 000
1 920 000
1 910 000
1 900 000
1 890 000
1 880 000
1
26
51
76
101
126
151
176
201
226
251
276
301
delta boxes =100
+ network π
+ integer solution
delta=100
+ network π
Increasing lower bound πb



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					



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***					

PERFECT DUAL INFORMATION: APPLICATIONS

- Useful in the context of Lagrangean Relaxation to recover primal feasibility (fractional)

- **Useful to perform *Crossover* from an interior point solution to an extreme point solution**
- H. Ben Amor, J. Desrosiers, and F. Soumis

  Recovering an optimal LP basis from an optimal dual solution,

  Operations Research Letters (2006)

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 method based on problem simplifications followed by a primal or dual simplex algorithm, as proposed in Bixby and Saltzman (2002).

- 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**		
pb1	188	72		
pb2	20	1183		
pb3	327	435		
pb4	8166	2568		
pb5	59	2645		
pb6	***	1797		
pb7	270	1092		
pb8	1036	1876		
pb9	78	2811		
pb10	37	3011		
Avg	1834.7	1749		
StdD	3350.1	1027.6		
Min	20	72		
Max	***	3011		

Crossover: CPLEX vs Box Methods (sec.)

Barrier 10^{-8}

Crossover	CrPrimal	CrDual	BoxPrimal	BoxDual
pb1	188	72	80	132
pb2	20	1183	98	208
pb3	327	435	189	171
pb4	8166	2568	249	321
pb5	59	2645	313	322
pb6	***	1797	242	229
pb7	270	1092	162	148
pb8	1036	1876	205	235
pb9	78	2811	205	265
pb10	37	3011	290	426
Avg	1834.7	1749	203.3	245.7
StdD	3350.1	1027.6	75.5	90.8
Min	20	72	80	132
Max	***	3011	313	426

Crossover: CPLEX vs Box Methods (sec.)

Barrier 10^{-8}

Crossover	CrPrimal	CrDual	BoxPrimal	BoxDual	*BoxCrPrimal*
pb1	188	72	80	132	**9**
pb2	20	1183	98	208	**13**
pb3	327	435	189	171	**26**
pb4	8166	2568	249	321	**35**
pb5	**59**	2645	313	322	**45**
pb6	***	1797	242	229	**86**
pb7	270	1092	162	148	**22**
pb8	1036	1876	205	235	**20**
pb9	78	2811	205	265	**43**
pb10	**37**	3011	290	426	**30**
Avg	1834.7	1749	203.3	245.7	**32.9**
StdD	3350.1	1027.6	75.5	90.8	**22.1**
Min	20	72	80	132	**9**
Max	***	3011	313	426	**86**

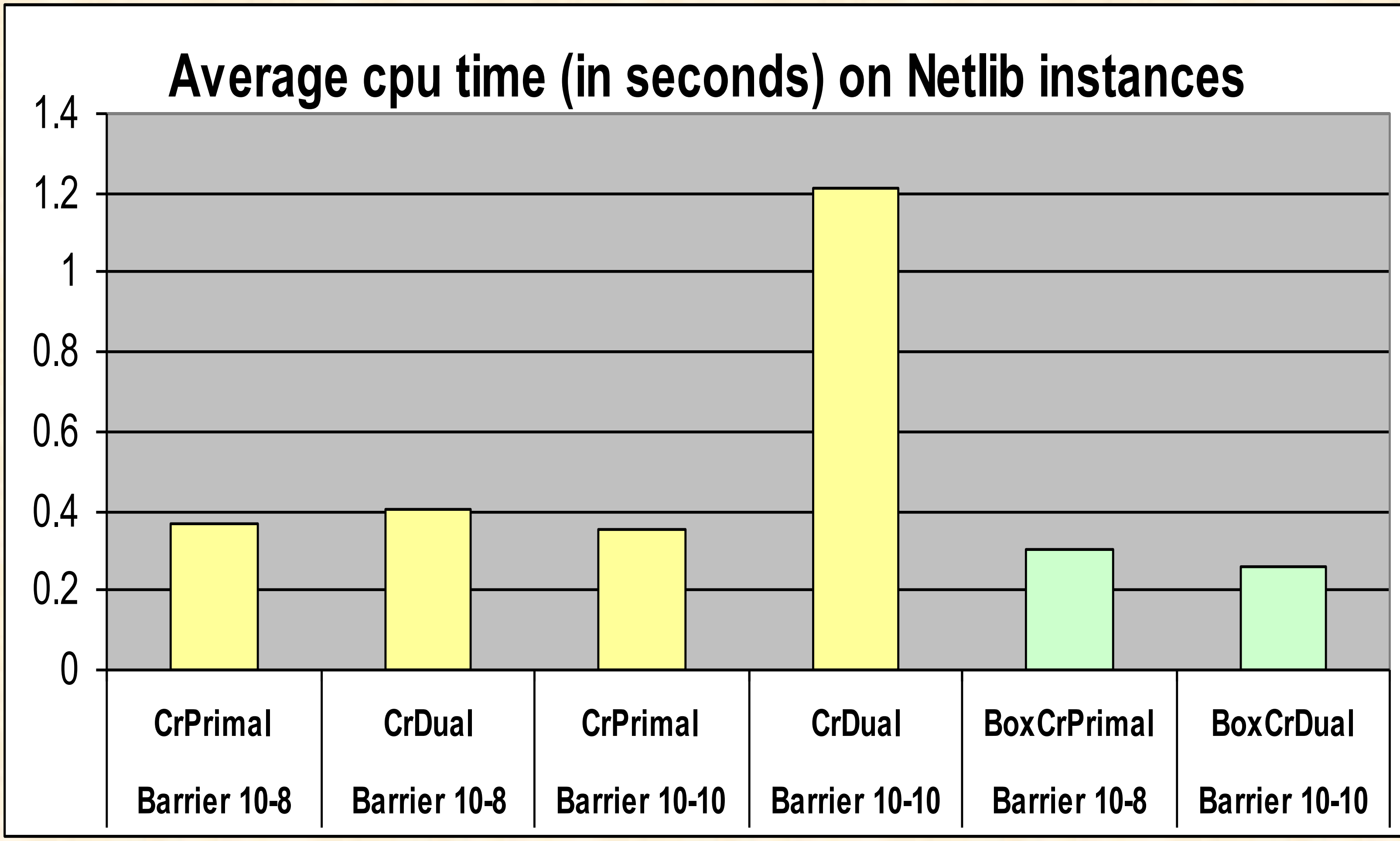
Crossover: CPLEX vs Box Methods (sec.)

Barrier 10^{-10}

Crossover	CrPrimal	CrDual	BoxPrimal	BoxDFual	*BoxCrPrimal*
pb1	101	15	79	129	**7**
pb2	***	27	85	218	**7**
pb3	605	69	146	124	**43**
pb4	232	2121	192	162	**16**
pb5	89	2819	231	234	**25**
pb6	72	2328	201	175	**22**
pb7	192	1025	147	152	**13**
pb8	1405	1407	168	198	**15**
pb9	7190	1957	238	308	**23**
pb10	4159	2832	218	261	**15**
Avg	2123.5	1460	170.5	196.1	**18.6**
StdD	2944.5	1127.2	56.5	59.5	**10.5**
Min	72	15	79	124	**7**
Max	***	2832	238	308	**43**

Efficient… and so simple!

Crossover: CPLEX vs Box Methods (sec.)



Crossover: CPLEX vs Box Methods (sec.)

Crossover	Barrier 10^{-8} CrPrimal	Barrier 10^{-8} CrDual	Barrier 10^{-10} CrPrimal	Barrier 10^{-10} CrDual	Barrier 10^{-8} *BoxCrPrimal*	Barrier 10^{-10} *BoxCrDual*
Netlib 500 (59)	0.05	0.14	0.05	0.14	0.04	0.05
Netlib 1000 (14)	0.12	0.13	0.12	0.14	0.14	0.13
Netlib 2000 (9)	0.24	0.36	0.24	0.26	0.31	0.27
Netlib 3000 (1)	0.10	0.10	0.10	0.10	0.10	0.10
Netlib 4000 (1)	6.40	2.00	5.10	84.90	2.60	0.80
‹ennington 5000 (9)	0.41	0.43	0.50	0.40	0.36	0.34
ɛnnington 10000 (4)	1.70	1.75	1.63	1.73	1.20	1.20
ɛnnington 30000 (5)	2.58	2.78	2.54	2.62	2.52	1.94
Avg	0.367	0.402	0.357	1.212	0.305	0.257
StdD	0.865	0.656	0.77	8.433	0.612	0.455