Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

AMBROS M. GLEIXNER
DANIEL E. STEFFY
KATI WOLTER

Iterative Refinement for Linear Programming

ZIB Report 15-15 (May 2015)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

Iterative Refinement for Linear Programming

Ambros M. Gleixner* Daniel E. Steffy† Kati Wolter‡

May 18, 2015

Abstract

We describe an iterative refinement procedure for computing extended precision or exact solutions to linear programming problems (LPs). Arbitrarily precise solutions can be computed by solving a sequence of closely related LPs with limited precision arithmetic. The LPs solved share the same constraint matrix as the original problem instance and are transformed only by modification of the objective function, right-hand side, and variable bounds. Exact computation is used to compute and store the exact representation of the transformed problems, while numeric computation is used for solving LPs. At all steps of the algorithm the LP bases encountered in the transformed problems correspond directly to LP bases in the original problem description. We show that this algorithm is effective in practice for computing extended precision solutions and that it leads to a direct improvement of the best known methods for solving LPs exactly over the rational numbers. Our implementation is publically available as an extension of the academic LP solver SoPlex.

1 Introduction

Most fast linear programming solvers available today for solving linear programs (LPs) use floating-point arithmetic, which can lead to numerical errors. Although such implementations are effective at computing approximate solutions for a wide range of instances, there are situations when they give unreliable results, or when extended-precision or exact solutions are desirable. Fast algorithms for exact linear programming are also directly useful as subroutines for solving mixed-integer programming problems exactly Applegate et al. (2007a), Cook et al. (2011).

Some recent articles that have used the exact solution of linear or integer programming instances to establish theoretical results, or to solve instances from numerically demanding applications, include Buchheim et al. (2008), Bulutoglu and Kaziska (2010), Burton and Ozlen (2012), Cohn et al. (2011), de Oliveira Filho and Vallentin (2010), Hales (2005), Held et al. (2012), Hicks and McMurray (2007), Lerman et al. (2012), Chindelevitch et al. (2014).

Computational tools for exact linear and integer programming have not been readily available until recently. Improving their speed and capabilities will expand the range of problems and instance sizes where they can be successfully applied.

Our main contribution is a new algorithm based on iterative refinement that builds extended-precision LP solutions using an approximate LP solver as a subroutine. As a byproduct, this algorithm helps to accelerate the state-of-the-art for solving LPs exactly over the rational numbers. This article is a significant extension of Gleixner et al. (2012). It is organized as follows. In Section 2 we give an overview of previous methods and introduce notation. Section 3 introduces iterative refinement for primal and dual feasible

*Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany, gleixner@zib.de
†Mathematics and Statistics, Oakland University, Rochester, Michigan, USA, steffy@oakland.edu
‡MOSEK ApS, Fruebjergvej 3, Box 16, 2100 Copenhagen, Denmark, support@mosek.com

LPs and discusses convergence and computational efficiency of the algorithm. In Section 4 we address the case of infeasible or unbounded LPs and give an integrated algorithm to handle LPs with a priori unknown status. Section 5 describes our implementation within the academic LP solver SoPlex and investigates the performance of the algorithm over a test set of 1,202 publically available benchmark instances. We give concluding remarks in Section 6.

2 Previous work

We assume that the reader is familiar with the concepts of linear programming and refer to Dantzig (1963), Chvátal (1983), or Schrijver (1986) for details. For clarity of presentation, we assume that a linear program (LP) is given in standard form

$$\min\{c^\mathsf{T} x \mid Ax = b, x \geqslant \ell\} \tag{1}$$

where $c \in \mathbb{Q}^n$ is the objective function vector, $\ell \in \mathbb{Q}^n$ is the vector of lower bounds, $A \in \mathbb{Q}^{m\times n}$ is the constraint matrix, and $b \in \mathbb{Q}^m$ is the right-hand side. Extensions to general form LPs are discussed in Appendix B. Without loss of generality, we assume A has full row rank and $n \geqslant m$. As most of the previous work in the literature for solving LPs accurately or exactly, we assume rational input data. The *dual LP* of (1) reads

$$\max\{b^\mathsf{T} y + \ell^\mathsf{T} z \mid A^\mathsf{T} y + z = c, z \geqslant 0\} \tag{2}$$

where $z = c - A^\mathsf{T} y$ is the vector of *dual slacks*. Because these are uniquely determined by the vector of dual multipliers y, we sometimes speak of a dual solution y when we actually mean a solution (y, z).

Most LP algorithms produce pairs of primal–dual solutions x, y. They are optimal if and only if x is primal feasible, y is dual feasible, and complementary slackness is satisfied, i.e., their *duality gap* is zero, defined as $\gamma(x, y) = (x - \ell)^\mathsf{T}(c - A^\mathsf{T} y)$. They are called basic if there exists subset of column indices $\mathcal{B} \subseteq \{1, \ldots, n\}$, $|\mathcal{B}| = m$, called a *basis* such that the *basis matrix* $A_{\cdot\mathcal{B}}$ is regular and x and y are uniquely determined by the linear systems $A_{\cdot\mathcal{B}} x_\mathcal{B} = b - \sum_{i\notin\mathcal{B}} A_{\cdot i}\ell_i$ and $y^\mathsf{T} A_{\cdot\mathcal{B}} = c_\mathcal{B}^\mathsf{T}$.[1] Geometrically, the primal feasible basic solutions are the vertices of the feasible region.

2.1 Exact methods for linear programming over the rational numbers

There is a trivial method of solving LPs with rational input data exactly, which is to apply a simplex algorithm and perform all computations in (exact) rational arithmetic.[2] Because of the high computational cost of rational arithmetic, this approach becomes prohibitively slow for large instances. More precisely, as Espinoza (2006) demonstrates computationally, the running time of the naïve approach is not so much correlated with the size of an LP, but with the encoding length of the basic solutions traversed by the simplex algorithm. Notable improvements of this approach are Edmonds' Q-pivoting (see Edmonds (1994), Edmonds and Maurras (1997), Azulay and Pique (1998)) and the mixed-precision simplex algorithm of Gärtner (1999).

Recent, more performant research efforts exploit the basis information provided by the simplex algorithm. If a candidate for an optimal basis is identified, then the corresponding primal–dual solution can be computed exactly and checked. If it is primal and dual feasible then it is optimal, because by construction basic solutions are complementary slack. It

[1] Here we use the notation $A_{\mathcal{I},\mathcal{J}}$ for the submatrix of A with rows and columns restricted to index sets $\mathcal{I} \subseteq \{1, \ldots, m\}$ and $\mathcal{J} \subseteq \{1, \ldots, n\}$, similarly for vectors. We abbreviate the set of all columns or rows by '$\cdot$'.

[2] Implementations of this approach can be found, for instance, in the packages for discrete computational geometry cdd+, see Fukuda and Prodon (1996) and lrs by Avis and Fukuda (1992).

has been observed by Dhiflaoui et al. (2003) and Koch (2004) that LP bases returned by floating-point solvers are often optimal. Applegate et al. (2007b) developed an exact rational LP solver, QSopt_ex, that exploits this behavior to achieve fast computation times on average. If an optimal basis is not identified by the double-precision subroutines, more simplex pivots are performed using increased levels of precision until the exact rational solution is identified. A simplified version of this procedure is summarized as Algorithm 1.

Algorithm 1: Incremental precision boosting for a primal and dual feasible LP

in : $\min\{c^\mathsf{T}x \mid Ax = b, x \geqslant \ell\}$ with $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m, c, \ell \in \mathbb{Q}^n$
out : primal–dual solution $x^* \in \mathbb{Q}^n, y^* \in \mathbb{Q}^m$, basis $\mathcal{B} \subseteq \{1,\ldots,n\}$

1 **begin**
2 **for** $p \leftarrow$ *double*, $128, 256, \ldots,$ *rational* **do**
3 get $\bar{A}, \bar{b}, \bar{c}, \bar{\ell} \approx A, b, c, \ell$ in precision p
4 solve $\min\{\bar{c}^\mathsf{T}x \mid \bar{A}x = \bar{b}, x \geqslant \bar{\ell}\}$ in precision p
5 get basis $\mathcal{B}$ returned as optimal
6 solve $A_{\cdot\mathcal{B}}x^*_\mathcal{B} = b - \sum_{i\notin\mathcal{B}} A_{\cdot i}\ell_i$ and ${y^*}^\mathsf{T}A_{\cdot\mathcal{B}} = c^\mathsf{T}_\mathcal{B}$ in rational arithmetic
7 **if** $x^*_\mathcal{B} \geqslant \ell_\mathcal{B}$ *and* $A^\mathsf{T}y^* \leqslant c$ **then**
8 **foreach** $i \notin \mathcal{B}$ **do** $x^*_i \leftarrow \ell_i$
9 return $x^*, y^*, \mathcal{B}$

QSopt_ex is often very effective at finding exact solutions quickly, especially when the double-precision LP subroutines are able to find an optimal LP basis. However, in cases when extended-precision computations are used to identify the optimal basis, or when the rational systems of equations solved to compute the rational solution are difficult, solution times can increase significantly.

2.2 Iterative refinement for linear systems of equations

Iterative refinement is a common technique to improve numerical accuracy when solving linear systems of equations going back to Wilkinson (1963). Given a system of linear equations $Mx = r$, $M \in \mathbb{Q}^{n\times n}$, $r \in \mathbb{Q}^n$, a sequence of increasingly accurate solutions $\{x_1, x_2, \ldots\}$ is constructed by first computing an approximate solution x_1, with $Mx_1 \approx r$. Then for $k \geqslant 2$, a refined solution $x_k \leftarrow x_{k-1} + \hat{x}$ is computed where $\hat{x}$ satisfies $M\hat{x} \approx \hat{r}$ and is a correction of the error $\hat{r} = r - Mx_{k-1}$ observed from the solution at the previous iteration. This procedure can either be applied in *fixed precision* where all operations are performed using the same level of precision, or in *mixed precision* where the computation of the residual errors $\hat{r}$ and the addition of the correction are computed with a higher level of precision than the system solves. See, e.g., Golub and van Loan (1983) for more details.

2.3 Moving from approximate to exact solutions

Iterative refinement can also be used as a subroutine for computing exact solutions to rational systems of linear equations. After a sufficiently accurate solution has been constructed, the exact rational solution vector can be recovered using continued fractions approximations, as computed by the extended Euclidean algorithm. This idea was first described by Ursic and Patarra (1983) and improved upon by Wan (2006), Pan (2011), and Saunders et al. (2011). Cook and Steffy (2011) compared this to other strategies for computing the exact rational solutions in line 6 of Algorithm 1. In many cases it was the fastest method.

The idea of "rounding" approximate solutions to exact rational solutions has been applied in more theoretical contexts as well. Grötschel et al. (1988) give polynomial-time LP algorithms based on the method of Khachiyan (1979), each iteration of which produces a smaller and smaller ellipsoid enclosing an optimal solution. Using this ability to find a tight enclosure around an optimal solution, techniques of lattice basis reduction can be applied

to recover an exact rational solution. For further discussion of these techniques refered to as *Diophantine approximation* see Schrijver (1986), Grötschel et al. (1988), von zur Gathen and Gerhard (2003). We also refer the reader to Yap (1997) for a general discussion on Exact and Robust Computational Geometry—although it does not discuss exact linear programming directly, many of the ideas are of direct relevance.

2.4 Rigorous bounds

Finally, we want to mention a different line of research that has focused on computing rigorous bounds on the objective function value of an LP. This can be particularly useful when solving mixed-integer linear programs exactly, see Jansson (2004), Neumaier and Shcherbina (2004), Althaus and Dumitriu (2009), and Steffy and Wolter (2013) for more details. These approaches employ interval arithmetic, see, e.g., Moore et al. (2009), in place of rational arithmetic in parts or throughout their algorithms.

3 Iterative refinement for linear programming

Iterative refinement is already applied by many floating-point LP solvers in order to improve their numerical robustness when solving linear systems (Maes 2013). We take this idea one step further and show how iterative refinement can be applied to an entire LP.

3.1 The Basic Algorithm

Our method solves a sequence of LPs, each one computing a correction of the previous to build an accurate primal–dual solution. This strategy will simultaneously refine both the primal and dual solutions by adjusting the primal feasible region and the objective function of the LP to be solved. It is based on the following theorem, which formally holds for all positive scaling factors Δ_P, Δ_D. As will become clear soon, we are interested in the case when Δ_P, $\Delta_D \gg 1$.

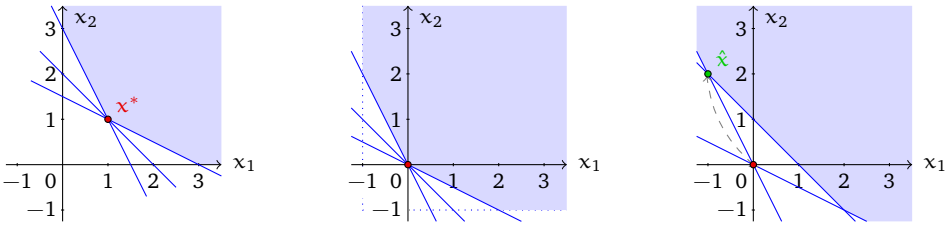
Theorem 3.1. *Suppose we are given an LP (P) in form $\min\{c^\mathsf{T}x \mid Ax = b, x \geqslant \ell\}$, then for $x^* \in \mathbb{R}^n$, $y^* \in \mathbb{R}^m$, and scaling factors Δ_P, $\Delta_D > 0$, consider the transformed problem*

$$\min\{\Delta_D\hat{c}^\mathsf{T}x \mid Ax = \Delta_P\hat{b}, x \geqslant \Delta_P\hat{\ell}\} \tag{$\hat{P}$}$$

where $\hat{c} = c - A^\mathsf{T}y^*$, $\hat{b} = b - Ax^*$, and $\hat{\ell} = \ell - x^*$. Then for any $\hat{x} \in \mathbb{R}^n$, $\hat{y} \in \mathbb{R}^m$ the following hold:

1. *$\hat{x}$ is primal feasible for $\hat{P}$ within an absolute tolerance $\epsilon_P \geqslant 0$ if and only if $x^* + \frac{1}{\Delta_P}\hat{x}$ is primal feasible for P within ϵ_P/Δ_P.*
2. *$\hat{y}$ is dual feasible for $\hat{P}$ within an absolute tolerance $\epsilon_D \geqslant 0$ if and only if $y^* + \frac{1}{\Delta_D}\hat{y}$ is dual feasible for P within ϵ_D/Δ_D.*
3. *$\hat{x}$, $\hat{y}$ violate complementary slackness for $\hat{P}$ by $\epsilon_S \geqslant 0$ if and only if $x^* + \frac{1}{\Delta_P}\hat{x}$, $y^* + \frac{1}{\Delta_D}\hat{y}$ violates complementary slackness for P by $\epsilon_S/\Delta_P\Delta_D$.*
4. *$\hat{x}$, $\hat{y}$ is an optimal primal–dual solution for $\hat{P}$ if and only if $x^* + \frac{1}{\Delta_P}\hat{x}$, $y^* + \frac{1}{\Delta_D}\hat{y}$ is optimal for P.*
5. *$\hat{x}$, $\hat{y}$ is a basic primal–dual solution of $\hat{P}$ associated with basis $\mathcal{B}$ if and only if $x^* + \frac{1}{\Delta_P}\hat{x}$, $y^* + \frac{1}{\Delta_D}\hat{y}$ is a basic primal–dual solution for P associated with basis $\mathcal{B}$.*

Proof. Proof. See Appendix A.1. □



(a) Original LP (b) Shifted LP (c) Shifted and scaled LP

Figure 1: Two-variable example for primal LP refinement

This theorem can be viewed in two complementary ways. From a numerical perspective, $(\hat{P})$ is formed by replacing the right-hand side b, the bounds on the variables ℓ, and the objective function vector c by the corresponding residual errors in an approximate solution x^*, y^*. This is similar to an iterative refinement step for linear systems of equations, but additionally the residual errors are magnified by the scaling factors Δ_P and Δ_D. Points 1 to 3 state the improved accuracy of the corrected solution $x^* + \frac{1}{\Delta_P}\hat{x}, y^* + \frac{1}{\Delta_D}\hat{y}$ if $\hat{x}, \hat{y}$ is an approximate solution to $(\hat{P})$.

Geometrically, $(\hat{P})$ is the result of applying the affine transformation $x \mapsto \Delta_P(x - x^*)$ to the primal and $y \mapsto \Delta_D(y - y^*)$ to the dual solution space of (P). Theorem 3.1 summarizes the straightforward one-to-one correspondence between solutions of the original problem (P) and the transformed problem $(\hat{P})$. Graphically, the primal transformation zooms in on the reference solution x^*—by first shifting the reference solution x^* to the origin, then scaling the problem by a factor of Δ_P. The dual transformation tilts the objective function to become the vector of reduced cost. This is illustrated by the following examples.

Example 3.2 (Primal LP refinement)**.** Consider the LP on two variables

$$\min\{x_1 + x_2 \mid 2x_1 + x_2 \geqslant 3, x_1 + 2x_2 \geqslant 3, x_1 + x_2 \geqslant 2 + 10^{-6}, x_1, x_2 \geqslant 0\}$$

with an approximate solution $x^* = (1, 1)^\mathsf{T}$ as depicted in the Figure 1a. (We use the inequality form without slack variables here for better visualization.) Note that the constraint $x_1 + x_2 \geqslant 2 + 10^{-6}$ is indistinguishable from $x_1 + x_2 \geqslant 2$ and the tiny violation of 10^{-6} is invisible on this scale. Shifting the problem such that the reference solution is centered at the origin gives the shifted LP in Figure 1b. After scaling the primal space by $\Delta_P = 10^6$, we obtain the transformed problem $\min\{x_1 + x_2 \mid 2x_1 + x_2 \geqslant 0, x_1 + 2x_2 \geqslant 0, x_1 + x_2 \geqslant 1, x_1, x_2 \geqslant -10^6\}$ in Figure 1c—$(\hat{P})$ in Theorem 3.1. Here, the infeasibility of the initial solution is apparent. An LP solver might return the solution $\hat{x} = (-1, 2)^\mathsf{T}$ instead, which corresponds to the corrected, exactly feasibly solution $x^* + \frac{1}{\Delta_P}\hat{x} = (1 - 10^{-6}, 1 + 2 \cdot 10^{-6})$ for the original problem.

Example 3.3 (Dual LP refinement)**.** Consider the LP on two variables

$$\min\{x_1 + (1 - 10^{-6})x_2 \mid x_1 + x_2 = 2, x_1, x_2 \geqslant 0\}$$

with an approximate solution $x^* = (2, 0)^\mathsf{T}$ as shown in Figure 2a. Note that any point on the line $x_1 + x_2 = 2$ looks optimal because on this scale the objective function is not distinguishable from $x_1 + x_2$. With dual multiplier $y^* = 1$ the reduced cost vector is $\hat{c} = (0, -10^{-6})^\mathsf{T}$. The solution x^*, y^* is complementary slack, but dual infeasible and slightly suboptimal. After replacing the objective function with the reduced cost vector and scaling it by $\Delta_D = 10^6$, we obtain the transformed LP with objective function $-x_2$ in Figure 2b. Now, the initial solution is seen to be clearly suboptimal and any LP solver should return

(a) Original LP

(b) Scaled reduced cost vector as objective

Figure 2: Two-variable example for dual LP refinement

the solution $\hat{x} = (2, 0)^\mathsf{T}$ instead, which—because we did not transform the primal in this example—is already the corrected, now optimal solution for the original LP. Alternatively, we can view this dual refinement as a primal-refinement step on the one-dimensional dual LP $\max\{2y \mid y \leqslant 1, y \leqslant 1 - 10^{-6}\}$. Shifting by the approximate dual solution $y^* = 1$ and scaling by $\Delta_D = 10^6$ yields the transformed problem $\max\{2y \mid y \leqslant 0, y \leqslant -1\}$. This gives the dual corrector $\hat{y} = -1$ and the corrected dual solution $y^* + \frac{1}{\Delta_D}\hat{y} = 1 - 10^{-6}$.

Applying the refinement step of Theorem 3.1 iteratively gives the scheme outlined in Algorithm 2. First, the LP is solved approximately, producing an initial primal–dual solution x_k, y_k for $k = 1$. Then, the primal and dual residual errors are computed and used to check whether termination tolerances for primal and dual feasibility and complementary slackness has been reached. If not, then the transformed problem $(\hat{P})$ is set up as in Theorem 3.1. The scaling factors are chosen as the inverse of the maximum primal and dual violations in order to normalize the right-hand side, lower bound, and objective function vectors. Additionally, we limit the increase of the scaling factors from round to round by the incremental scaling factor α to ensure that we do not scale by infinity if one of the violations drops to zero. Eventually, the transformed LP is solved approximately to obtain a solution $\hat{x}, \hat{y}$, which is used to refine the accuracy of the candidate solution x_k, y_k. This process is repeated until the required accuracy is reached. All operations are performed in exact rational arithmetic unless otherwise noted.

To our knowledge, such an iterative refinement algorithm for linear programming has not been described in the literature. However, we want to mention the presentation of Saunders and Tenenblat (2006) on warm-starting interior point methods for convex quadratic programs via a "zoom strategy" that also shifts the problem and scales quadratic terms in the objective function. Furthermore, we believe that some aspects of our approach might have been used in software packages, although most likely not with extended precision or rational arithmetic. In particular, we have heard that some interior point solvers may have experimented with the idea of replacing the objective function of an LP by its reduced cost vector and resolving the problem to improve some of its numerical properties – this would correspond to performing a single dual-refinement step (Ladányi 2011).

Remark 3.4. Classical iterative refinement for linear systems does not scale the residual errors on the right-hand side. Instead it exploits the fact that floating-point arithmetic is more accurate close to zero and so solving the linear system in floating-point naturally yields a error that is relative in the right-hand side. This does not hold for linear programming since LP solvers work with fixed, absolute tolerances. If we did not scale the right-hand side, bounds, and reduced-cost values in the objective function of the transformed problem $\hat{P}$, a standard floating-point LP solver would consider the zero solution optimal within its tolerances. This would result in a zero corrector solution $\hat{x}, \hat{y}$ and no increase in precision. Furthermore, we need scaling for the same reason as for the iterative refinement scheme for linear systems of Wan (2006)—because we want to compute solutions that have a higher precision than can be represented in the precision of the floating-point solver.

Algorithm 2: Iterative refinement for a primal and dual feasible LP

in : $\min\{c^\mathsf{T}x \mid Ax = b, x \geqslant \ell\}$ with $A \in \mathbb{Q}^{m\times n}, b \in \mathbb{Q}^m, c, \ell \in \mathbb{Q}^n$, incremental scaling limit $\alpha > 1$, termination tolerances ϵ_P, ϵ_D, $\epsilon_S > 0$

out : primal–dual solution $x^* \in \mathbb{Q}^n, y^* \in \mathbb{Q}^m$

begin
- $\Delta_P^1 \leftarrow 1$, $\Delta_D^1 \leftarrow 1$ `/* initial solve */`
- get $\bar{A}, \bar{b}, \bar{\ell}, \bar{c} \approx A, b, \ell, c$ in precision of the LP solver
- solve $\min\{\bar{c}^\mathsf{T}x \mid \bar{A}x = \bar{b}, x \geqslant \bar{\ell}\}$ approximately
- x^1, y^1 $\leftarrow$ approximate primal–dual solution returned
- **for** $k \leftarrow 1, 2, \ldots$ **do** `/* refinement loop */`
  - $\hat{b} \leftarrow b - Ax_k$ `/* compute violations */`
  - $\hat{\ell} \leftarrow \ell - x_k$
  - $\hat{c} \leftarrow c - A^\mathsf{T}y_k$
  - $\delta_{P,k} \leftarrow \max\{\max_{j=1,\ldots,m} |\hat{b}_j|, \max_{i=1,\ldots,n} \hat{\ell}_i\}$ `/* check termination */`
  - $\delta_{D,k} \leftarrow \max\{0, \max\{-\hat{c}_i \mid i = 1, \ldots, n\}\}$
  - $\delta_{S,k} \leftarrow |\sum_{i=1,\ldots,n} -\hat{\ell}_i\hat{c}_i|$
  - **if** $\delta_{P,k} \leqslant \epsilon_P$ *and* $\delta_{D,k} \leqslant \epsilon_D$ *and* $\delta_{S,k} \leqslant \epsilon_S$ **then**
    - return $x^* \leftarrow x_k, y^* \leftarrow y_k$
  - $\Delta_{P,k+1} \leftarrow 1/\max\{\delta_{P,k}, (\alpha\Delta_{P,k})^{-1}\}$ `/* solve transformed problem */`
  - $\Delta_{D,k+1} \leftarrow 1/\max\{\delta_{D,k}, (\alpha\Delta_{D,k})^{-1}\}$
  - get $\bar{b}, \bar{\ell}, \bar{c} \approx \Delta_{P,k+1}\hat{b}, \Delta_{P,k+1}\hat{\ell}, \Delta_{D,k+1}\hat{c}$ in precision of the LP solver
  - solve $\min\{\bar{c}^\mathsf{T}x \mid \bar{A}x = \bar{b}, x \geqslant \bar{\ell}\}$ approximately
  - $\hat{x}$, $\hat{y}$ $\leftarrow$ approximate primal–dual solution returned
  - $x_{k+1} \leftarrow x_k + \frac{1}{\Delta_{P,k+1}}\hat{x}$ `/* perform correction */`
  - $y_{k+1} \leftarrow y_k + \frac{1}{\Delta_{D,k+1}}\hat{y}$

3.2 Convergence

In the iterative refinement scheme of Algorithm 2, the approximate LP solver is treated as a black-box oracle. This permits an implementation where the LP solver is accessed through an interface and has the advantage that it allows substitution whenever an application benefits from a specific LP algorithm. The following basic assumption suffices to obtain a sequence of increasingly accurate solutions.

Assumption 3.5. *For every $A \in \mathbb{R}^{m\times n}$ there exist constants ϵ, $0 \leqslant \epsilon < 1$, and $\sigma \geqslant 0$ such that for all $c, \ell \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ for which the LP $\min\{c^\mathsf{T}x \mid Ax = b, x \geqslant \ell\}$ is primal and dual feasible, the LP solver returns an approximate primal–dual solution $\bar{x} \in \mathbb{Q}^n$,$\bar{y} \in \mathbb{Q}^m$ that satisfies*

1. $\|A\bar{x} - b\|_\infty \leqslant \epsilon$,
2. $\bar{x} \geqslant \ell - \epsilon\mathbb{1}$,
3. $c - A^\mathsf{T}\bar{y} \geqslant -\epsilon\mathbb{1}$, *and*
4. $|\gamma(\bar{x}, \bar{y})| \leqslant \sigma$

when it is given the LP $\min\{\bar{c}^\mathsf{T}x \mid \bar{A}x = \bar{b}, x \geqslant \bar{\ell}\}$, where $\bar{A} \in \mathbb{Q}^{m\times n}$, $\bar{c}, \bar{\ell} \in \mathbb{Q}^n$, and $\bar{b} \in \mathbb{R}^m$ are A, c, ℓ, and b rounded to the working precision of the LP solver.

This assumption suffices for the following simple convergence result.

Corollary 3.6. *Suppose we are given a primal and dual feasible LP as in* (1) *with constraint matrix A for which Assumption 3.5 holds with constants ϵ and σ. Let $x_k, y_k, \Delta_{P,k}$, and $\Delta_{D,k}$, $k = 1, 2, \ldots$, be the sequences of primal–dual solutions and scaling factors produced by Algorithm 2 with incremental scaling limit $\alpha > 1$, and let $\tilde{\epsilon} \coloneqq \max\{\epsilon, 1/\alpha\}$. Then for all k,*

1. $\Delta_{P,k}, \Delta_{D,k} \geqslant 1/\tilde{\epsilon}^{k-1}$,
2. $\|Ax_k - b\|_\infty \leqslant \tilde{\epsilon}^k$,
3. $x_k - \ell \geqslant -\tilde{\epsilon}^k \mathbb{1}$,
4. $c - A^\mathsf{T} y_k \geqslant -\tilde{\epsilon}^k \mathbb{1}$, *and*
5. $|\gamma(x_k, y_k)| \leqslant \sigma \tilde{\epsilon}^{2(k-1)}$.

Hence, Algorithm 2 terminates after at most $\max\{\log(\epsilon_P), \log(\epsilon_D), \log(\epsilon_S \epsilon/\sigma)/2\}/\log(\tilde{\epsilon})$ *approximate LP solves.*

Proof. Proof. See Appendix A.2. □

Several remarks on the reasonableness of Assumption 3.5 are in order. As Klotz (2014) points out for the CPLEX LP code, state-of-the-art LP solvers typically use an absolute definition for their tolerance requirements. First and foremost, because limited floating-point precision is by construction relative, an LP solver based only on floating-point computation will in general not be able to return solutions within absolute tolerances—certainly not the fast LP solvers we have in mind for practical applications. Otherwise, this would permit the following, much simpler approach. In Theorem 3.1, choose $x^*, y^* = 0$ and scaling factors $\Delta_P = \Delta_D = N$ arbitrarily large and solve $(\hat{P})$ for a solution $\bar{x}, \bar{y}$ as guaranteed by Assumption 3.5. Then $\bar{x}/N, \bar{y}/N$ would violate primal and dual feasibility by at most ϵ/N and complementary slackness by at most σ/N^2. One refinement step would suffice to reach arbitrary precision.

Because of this, the primal and dual scaling factors in Algorithm 2 are limited by the inverse of the maximum primal and dual violations, respectively. As a result, the largest entries in the right-hand side, lower bounds, and objective function vector of the transformed LP that correspond to violations of primal and dual feasibility become at most one in absolute value. For these variables and constraints, a relative tolerance requirement implies the absolute tolerance requirement of Assumption 3.5. However, note that when in x_k, y_k the lower bound is already satisfied for a variable then its lower bound in the transformed LP may have a large absolute value; and if its reduced cost is already nonnegative then the same holds for the transformed objective function coefficient.

Although there is no guarantee that a floating-point solver will produce solutions that are accurate to within an absolute tolerance, with our scaling strategy we expect that a modern LP solver will yield satisfactory results for most LPs in practice. However, in some cases LPs may be so poorly conditioned that the floating-point LP solver produces meaningless results. In such a case performing extended-precision computations within the solver may be necessary. A minor modification could be done to Algorithm 2 to incrementally boost the working precision of the LP solver when needed as is done in Algorithm 1.

Note that if the floating-point solver encounters numerical difficulties, it may not only return a solution with large violations, but even incorrectly conclude infeasibility or unboundedness. Section 4.1 describes a robust implementation of the iterative refinement scheme that, to the extent possible, tries to cope with such violations of Assumption 3.5.

3.3 Arithmetic precision

Like many iterative refinement procedures for linear systems, Algorithm 2 is a mixed-precision procedure. This has the advantage that the most involved part—LP solving—is executed in fast floating-point arithmetic. Expensive rational arithmetic is only used for computing and scaling the new objective function, right-hand side, and bounds of the transformed LP, which amounts to two matrix-vector multiplications and a constant multiple of $n + m$ elementary operations per refinement round. Hence the number of elementary operations performed in rational arithmetic grows linearly with the number of nonzeros of the constraint matrix.

Nevertheless, rational arithmetic remains computationally expensive. Unlike the case of floating-point arithmetic, its cost is not constant per operation, but may increase if the encoding length of the corrected solution grows with increased accuracy. In this respect, one crucial modification to Algorithm 2 is to round the scaling factors $\Delta_{P,k}$ and $\Delta_{D,k}$ to powers of two. This helps to keep the representation of the refined solution simple. It does not affect convergence, since for this only the order of magnitude of the scaling factors matters.

Finally we wish to remark that we use exact rational arithmetic in order to allow the computation of arbitrarily precise solutions. If the goal is merely to reach a certain fixed level of accuracy then it may be possible to replace rational by (sufficiently high) extended-precision arithmetic.

4 Handling infeasibility and unboundedness

For infeasible or unbounded LPs there exists no approximately primal and dual feasible reference solution that can be refined. In this case, our goal is to construct a high-precision certificate of infeasibility or unboundedness.

4.1 Testing feasibility

By Farkas lemma we know that an LP of form (1) is feasible if and only if the following auxiliary LP has optimal objective value of zero

$$\max\{(b - A\ell)^\mathsf{T} y \mid A^\mathsf{T} y \leqslant 0, (b - A\ell)^\mathsf{T} y \leqslant 1\}. \tag{3}$$

The last inequality on the objective function ensures boundedness, and that if the optimal objective value is nonzero, it is equal to one. Feasible solutions to (3) with positive objective value serve as infeasibility certificates to the LP (1) and are often referred to as *Farkas proofs*. Because the zero solution is trivially feasible, this LP is primal and dual feasible and iterative refinement can be applied to compute an arbitrarily accurate Farkas proof.

In order to integrate this most seamlessly into one refinement scheme for LPs with unknown status, we will see later that it is more suitable to consider the dual of (3), which reads

$$\min\{\tau \mid A\xi + (b - A\ell)\tau = (b - A\ell), \xi, \tau \geqslant 0\}.$$

Substituting $1 - \tau$ for τ gives the more natural formulation

$$\max\{\tau \mid A\xi - (b - A\ell)\tau = 0, \xi \geqslant 0, \tau \leqslant 1\}, \tag{4}$$

to which we will refer to as *feasibility LP*. The following lemma summarizes how solving this LP gives either a primal feasible solution or a Farkas proof of infeasibility for (1).

Lemma 4.1. *Suppose we are given an LP in equality form* (1)*, then the following hold.*

1. *The auxiliary LP* (4) *is primal and dual feasible.*
2. *The original LP* (1) *is feasible if and only if the auxiliary LP* (4) *has an optimal objective value of one.*

3. *If the optimal objective value of* (4) *is less than one and* y^* *is an optimal dual solution vector for* (4)*, then* $-y^*$ *is a Farkas proof for the infeasibility of* (1)*.*

4. *If* (ξ^*, τ^*)*,* $\tau^* > 0$*, is an approximate optimal solution of* (4) *that violates primal feasibility by at most* ϵ_P*, then* $x^* = \frac{1}{\tau^*}\xi^* + \ell$ *is a feasible solution for* (1) *within tolerance* ϵ_P/τ^**.*

Proof. Proof. See Appendix A.3. □

As point 4 shows, when applying iterative refinement to the feasibility LP we have to adjust our termination criterion for primal feasibility. When a primal violation of $\delta_{P,k}$ is achieved in Algorithm 2, we may terminate if $\tau_k > 0$ and $\delta_{P,k}/\tau_k \leqslant \epsilon_P$. If $\tau_k \approx 0$, the dual solution gives an approximate Farkas proof. For a discussion of how to test the feasibility of LPs in general form, see Appendix B.3

4.2 Computing a rigorous infeasibility box from an approximate Farkas proof

Because a Farkas proof remains valid when multiplied by an arbitrary positive scalar, absolute tolerance requirements are meaningless. A Farkas proof of infeasibility for an LP in form (1) consists of a vector of dual multipliers $y \in \mathbb{Q}^m$ for the rows and a "reduced cost" vector $z \in \mathbb{Q}^n$ of multipliers for the bound constraints satisfying $z \geqslant 0$ such that the following hold.

$$A^\mathsf{T} y + z = 0 \tag{5}$$

$$b^\mathsf{T} y + \ell^\mathsf{T} z > 0 \tag{6}$$

An approximate Farkas proof may violate both of these conditions. Given an approximate Farkas proof y, z the equation (5) can be enforced by adjusting z to be equal to $-A^\mathsf{T} y$. This, however, may set some components of z to negative values and may also create, or increase, a violation of (6). Even if iterative refinement applied to the feasibility LP produces Farkas proofs with smaller and smaller violations, this does in general not suffice to obtain a reliable certificate of infeasibility.

In the following, we introduce the concept of an *infeasibility box* that does not prove "approximate" infeasibility of the entire LP, but establishes exactly proven infeasibility within restricted bounds. As Neumaier and Shcherbina (2004) note, even if a Farkas proof is invalid in the classical sense of (5) and (6), the vector of dual multipliers y by aggregation gives the valid inequality

$$(y^\mathsf{T} A)x \geqslant b^\mathsf{T} y \tag{7}$$

which we call a *Farkas cut*. If we can show that (7) is violated by all points x with $x \geqslant \ell$, then the LP is proven infeasible. In Neumaier and Shcherbina (2004) it was observed that interval arithmetic can be used to compute a lower bound on the left-hand side of (7), and if this is below the right-hand side value it produces a certificate of infeasibility. However, this approach may fail, even if the approximate Farkas proof is very accurate.

We extend this notion by describing a method that will, given an approximate Farkas proof y, work backwards to determine a domain in which no feasible solution x can exist. We compute the largest value R such that (7) is violated by all points x with $\|x\|_\infty < R$, i.e., that the feasible region (of the one-row relaxation given by the Farkas cut) intersected with the infeasibility box $\{x \mid -R\mathbb{1} < x < R\mathbb{1}\}$ is empty.

If the Farkas cut is written in form $d^\mathsf{T} x \geqslant d_0$, this largest R is computed by the mapping $\rho : \mathbb{Q}^n \times \mathbb{Q} \to \mathbb{Q}_{\geq 0} \cup \{\infty\}$ defined as

$$\rho(d, d_0) \coloneqq \begin{cases} 0 & \text{if } d_0 \leqslant 0, \\ d_0/\|d\|_1 & \text{if } d \neq 0 \text{ and } d_0 > 0, \\ \infty & \text{if } d = 0 \text{ and } d_0 > 0. \end{cases} \tag{8}$$

If $\rho(d, d_0) = 0$, then the infeasibility box is empty; $\rho(d, d_0) = \infty$ implies that the full LP is successfully proven infeasible.

This kind of answer is both mathematically sound and helpful to users of an LP solver in practice. It allows them to conclude that feasible solution vectors must have large entries in absolute value, which might or might not be viable for the application at hand. The dimensions of this infeasibility box may be more comprehensible to an end user than, for example, a relative feasibility of a normalized Farkas proof.

Algorithm 3 describes the complete procedure for computing the infeasibility box. We assume that all arithmetic operations are executed in rational arithmetic. Throughout the algorithm, $d^\mathsf{T} x \geqslant d_0$ is a valid inequality for the LP and $R = \rho(d, d_0)$. First, the algorithm computes the aggregated constraint. Next, the loop starting on line 6 tries to incorporate the bounds on the variables in order to further increase the size of the infeasibility box. At this point, $d = A^\mathsf{T} y^*$, so $-d$ corresponds to the vector z in (5) and (6) of dual multipliers for the bound constraints. Suppose d_0 is positive and d contains more than one nonzero entry. For some $d_i < 0$, adding the inequality $-d_i x_i \geqslant -d_i \ell_i$ to $d^\mathsf{T} x \geqslant d_0$ increases the value of $\rho(d, d_0)$ if and only if

$$\begin{aligned}
\rho(d - d_i e_i, d_0 - d_i \ell_i) > \rho(d, d_0) &\Leftrightarrow \frac{d_0 - d_i \ell_i}{\sum_{q \neq i} |d_q|} > \frac{d_0}{\sum_q |d_q|} \\
&\Leftrightarrow (d_0 - d_i \ell_i) \sum_q |d_q| > d_0 \sum_{q \neq i} |d_q| \\
&\Leftrightarrow d_0 |d_i| - d_i \ell_i \|d\|_1 > 0 \\
&\Leftrightarrow -\ell_i < \rho(d, d_0). \qquad (9)
\end{aligned}$$

This motivates the order in which the variables are considered when attempting to strengthen the Farkas cut in this manner, the variables with larger lower bounds are considered first.

At the end of the first loop, we may have $d_0 \leqslant 0$ and $R = 0$. As long as this holds, including lower bounds will increase d_0, although not necessarily R. Once $-\ell_i$ exceeds R, d_0 cannot be further increased by this procedure and the algorithm is terminated, returning R. Otherwise, we continue to include bound constraints in decreasing order as long as the criteria given by (9) is satisfied, returning R.

Algorithm 3: Infeasibility box computation for an approximate Farkas proof

in : Inequality system $Ax = b, x \geqslant \ell$ with $A \in \mathbb{Q}^{m \times n}$, $b \in \mathbb{Q}^m$, $\ell, \in \mathbb{Q}^n$, approximate Farkas proof $y^* \in \mathbb{Q}^m$

out : $R \geqslant 0$ such that $\|x\|_\infty \geqslant R$ for all feasible solutions x

```
1  begin
2      d_0 ← b^T y*
3      d ← A^T y*
4      R ← ρ(d, d_0)
5      reindex variables such that ℓ_1 ⩾ ℓ_2 ⩾ ... ⩾ ℓ_n
6      for i ← 1, 2, ..., n do                    /* include bound constraints */
7          if −ℓ_i ⩾ R then
8              return R
9          else if d_i < 0 then
10             d_0 ← d_0 − d_i ℓ_i
11             d_i ← 0
12             R ← ρ(d, d_0)
13     return R
```

Remark 4.2 (Infeasibility box arithmetic)**.** The infeasibility box algorithm as described uses rational arithmetic in order to obtain provable results. To save computation, R can be updated efficiently instead of recomputing $\rho(d, d_0)$ from scratch several times. Using the sparsity of the constraint matrix A and of vectors y^*, d is also critical. Still, rational arithmetic may in some cases be too expensive. Alternatively, Algorithm 3 could be implemented using interval arithmetic, which is faster and still yields proven results, if also potentially a smaller infeasibility box.

It remains to be explained how the computation of an infeasibility box interacts with iterative refinement of the feasibility LP. Intuitively, the size of the infeasibility box should grow as the approximate Farkas proof given as input becomes more accurate. This becomes most clear when we look at the simple case of a system of equations without bounds, $Ax = b$, with the feasibility LP $\max\{\tau \mid A\xi - b\tau = 0, \tau \leqslant 1\}$.

Let ξ_k, τ_k, y_k be a sequence of more and more accurate primal–dual solutions produced by Algorithm 2. From the proof of Theorem 3.1 it follows that not only the total violation of complementary slackness γ, but also the individual violation w.r.t. τ goes to zero, i.e.,

$$(1 - b^\mathsf{T} y_k)(1 - \tau_k) \to 0.$$

If τ_k does not converge to one, i.e. $\tau_k < C < 1$ (indicating infeasibility of the system), then $b^\mathsf{T} y_k \to 1$. At the same time, the dual violation $\|0 - A^\mathsf{T} y_k\|_\infty$ goes to zero. If we apply Algorithm 3 to y_k, then after the first loop we will have $d = A^\mathsf{T} y_k$ and $d_0 = b^\mathsf{T} y_k$. Hence, $R = d_0/\|d\|_\infty$ grows towards infinity with increasingly accurate iterates y_k.

This makes it natural to interleave the infeasibility box computation with the iterative refinement of the feasibility LP right after checking termination in line 13 of Algorithm 2. It will typically be called after a normal floating-point solve that claimed infeasibility, hence we assume an approximate Farkas proof as input that is tested right at the beginning with Algorithm 3. If the computed radius of the infeasibility box is below the termination threshold, we continue to construct the feasibility LP by shifting bounds and sides. As first reference point we choose the all zero solution for the primal and the given approximate Farkas proof for the dual solution.

The refinement loop works as in Algorithm 2 except when checking termination (which is skipped for the first artificial solution). If tolerance ϵ_P is not satisfied, this is either because (ξ_k, τ_k) is not yet sufficiently accurate for the feasibility LP or because the optimal value of the feasibility LP is less than one. In the former case we continue with the next refinement step, in the latter case we run Algorithm 3 on y_k. The latter is checked by $\tau_k < 1 - \delta_{S,k+1}$ because careful calculation shows that for y_k the left-hand term in (6), which needs to be positive, equals $1 - \tau_k - \gamma(\xi_k, \tau_k, y_k)$. It is important that we do not use a fixed threshold to compare τ_k against one since the optimal value of the feasibility LP for infeasible LPs may be arbitrarily close to one. For discussion on how the infeasibility box is constructed for general form LPs, see Appendix B.4.

4.3 Testing unboundedness

A certificate of primal unboundedness consists of a feasible primal solution vector and an unbounded direction of improving objective function value in the recession cone. The former can be computed as described above. For the latter we apply the above feasibility test—assuming the LP is in form (1)—to the system

$$\begin{aligned} Av &= 0 \\ v &\geqslant 0 \\ c^\mathsf{T} v &= -1. \end{aligned} \tag{10}$$

Since the primal violation of an unbounded ray is not scale-invariant, we should normalize the violation of primal feasibility by the objective function decrease, i.e., given an approximate unbounded direction v^* we should use the violation of $v^*/|c^\mathsf{T} v^*|$ for checking

the primal termination tolerance. This guarantees a maximum increase of the primal violation by ϵ_P as we decrease the objective function by one unit along the ray. Discussion on testing unboundedness for general form LPs can be found in Appendix B.5.

4.4 An integrated refinement algorithm

After discussing how iterative refinement is applied to infeasible and unbounded LPs separately, this section will show how these techniques are integrated into one algorithm to handle LPs of an a priori unknown status. A crucial property of such an algorithm is its ability to cope with incorrect answers of the underlying floating-point LP solver, since numerically challenging problems for which floating-point solvers return inconsistent results are one of the prime motivations for applying iterative refinement.

Figure 3 gives a flowchart of the integrated algorithm. We start by applying Algorithm 2 to the given LP. If the floating-point solver returns approximately optimal solutions at each call, the refinement steps will yield increasingly accurate solutions and we return a solution meeting the termination tolerances. If the floating-point solver concludes infeasibility or unboundedness for the initial or one of the transformed LPs, then the refinement loop is interrupted to test this claim. Here we implicitly rely on the fact that the transformed LP is infeasible or unbounded if and only if the original LP is infeasible or unbounded, respectively, which is a consequence of Theorem 3.1.

In the case of floating-point infeasibility, we apply iterative refinement to the feasibility LP (4) described in Section 4.1. Using the primal formulation here keeps the overhead of transforming the original LP low, the dual formulation would result in increased overhead. We only need to add one column for the auxiliary variable τ to the constraint matrix, then modify the objective function and some of the bounds and sides. If this refinement terminates with an approximately feasible solution we reject the floating-point solver's claim of infeasibility and start the iterative refinement algorithm again. Since the attempt to establish infeasibility has failed, we again attempt to compute an optimal solution, but do so with modified settings in the floating-point LP solver such as alternate tolerance levels and hope for success. Otherwise, we terminate and conclude infeasibility.

In the case that the floating-point solver claims unboundedness, we first apply iterative refinement to (10) to compute a primal ray. If this fails we restart the refinement of the original LP with modified floating-point parameters. If it succeeds we continue testing feasibility. If we obtain a primal feasible solution we conclude unboundedness. Otherwise, we return (primal and dual) infeasibility.

Remark 4.3. Some LPs are formulated such that a small perturbation may change the feasibility status of the problem. As a result, there may simultaneously exist solutions that are feasible within a small feasiblity tolerance as well as highly accurate Farkas proofs. LPs near or on this boundary of feasibility and infeasibility are called ill-conditioned, or ill-posed, see Renegar (1994). As our algorithm first attempts to find and refine a feasible solution, it is in a sense biased toward finding approximate feasible solutions if they exist, even if approximate Farkas proofs also exist. In such cases one may at least conclude that a small perturbation of the LP would be feasible.

5 Computational study

As pointed out, Assumption 3.5 on the accuracy of the floating-point solutions will not hold in general. We hope that in practice it is satisfied for LPs constructed during iterative refinement. In the following, we describe computational results for a simplex-based implementation. One of the motivations for basing our experiments on a simplex solver instead of an interior point solver is the fact that the LPs solved are highly similar in that they all share the same constraint matrix and their solution spaces are affine transformations of each other. Hence the solution refined by the last LP solve gives a starting point

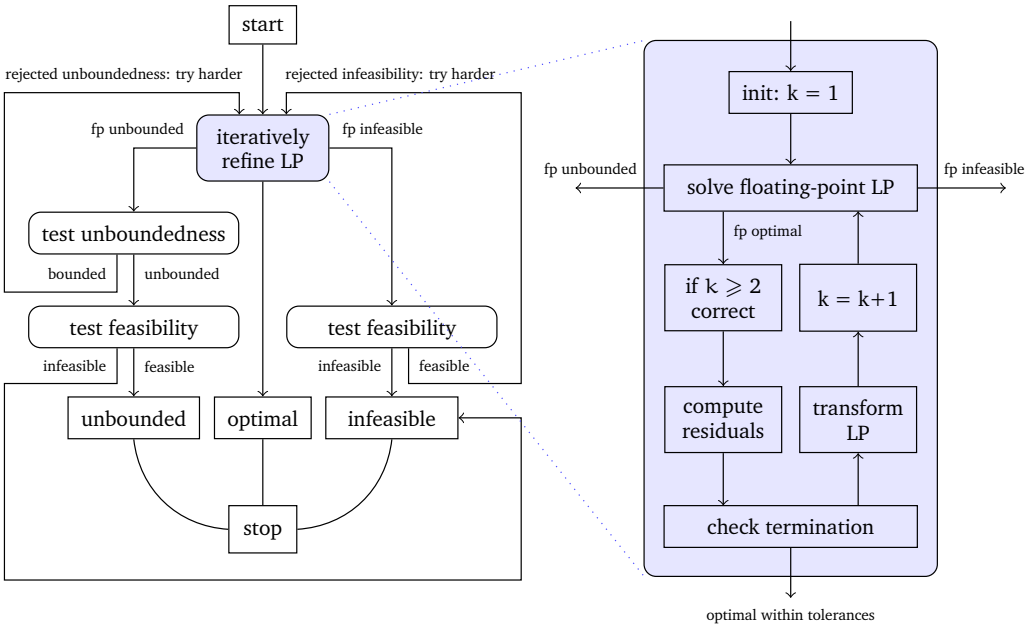


Figure 3: Iterative refinement for LPs of a priori unknown status

for the next LP solve and its basis information carries over as shown by Theorem 3.1. While modern implementations of interior point methods are reported to be faster on average for solving LPs from scratch, the unmatched hot-starting capabilities of the simplex method promise greater gains in performance when a larger number of refinement rounds are applied.

5.1 Implementation

Starting with Version 1.7 released in July 2012, we extended the SoPlex[3] LP solver (Wunderling 1996) with functionality to read, store, check, and process LPs and LP solutions in rational precision. Our implementation is based on GMP[4] and can be linked to EGlib[5] for faster memory allocation.

Using basis information. The basis information is first used for computing the maximum dual violation $\delta_{D,k}$ in line 11 of Algorithm 2. The reduced cost of a variable is considered infeasible if it is positive but the basis status of the column is *not* nonbasic (i.e., not fixed) at the lower bound. The violation of complementary slackness is not tracked and line 12 is skipped since basic solutions are by construction complementary slack.

Second, we hot start the LP solver in line 18 from the previous basis, reusing the factorization of the basis matrix. Note that unless the primal and dual scaling factors are limited by the incremental scaling limit α, the corresponding basic solution is guaranteed to be primal and dual infeasible for the transformed LP. SoPlex allows for arbitrary regular starting bases using a shifting technique, see (Wunderling 1996, Section 1.4).

Third, we have to consider basis information in the correction step. Suppose in the floating-point solution $\hat{x}$ some variable is nonbasic at its lower bound, $\hat{x}_i = \bar{\ell}_i$, say. Then the corrected solution value $(x_{k+1})_i$ may differ from ℓ_i by $(\bar{\ell}_i/\Delta_{D,k+1}) - \hat{\ell}_i$. This is due to the necessary rounding step in line 17. We correct this by setting $\hat{x}_i$ directly to ℓ_i in this

[3]Zuse Institute Berlin. SoPlex—the Sequential object-oriented simPlex. `http://soplex.zib.de/`.
[4]GMP. The GNU Multiple Precision Arithmetic Library. `http://www.gmplib.org/`
[5]Daniel G. Espinoza and Marcos Goycoolea. EGlib. Efficient General Library. `http://www.dii.uchile.cl/~daespino/EGlib_doc/main.html`

case, which has the added advantage of avoiding one multiplication and addition in rational arithmetic. This adjustment of the corrector solution is on a tiny order of magnitude and should not compromise Assumption 3.5—if it does then the floating-point solutions are almost certainly already unreliable.

Robust Floating-Point Solves. For numerically difficult instances, SoPlex may encounter numerical troubles and fail to solve one of the floating-point LPs—running into a singular basis, aborting due to cycling, returning infeasible or unbounded for the auxiliary LPs when testing feasibility or unboundedness, or returning infeasible or unbounded although these results have been rejected earlier in the integrated solving loop displayed in Figure 3. If this happens, we try again with a series of different parameter settings until successful. If this still does not help, we terminate without reaching the desired tolerances.

Updating Residual Vectors. In line 7 of Algorithm 2 for $k \geqslant 2$, we can use $x_k = x_{k-1} + \frac{1}{\Delta_{P,k}}\hat{x}$ in order to compute $b - Ax_k$ by subtracting $\frac{1}{\Delta_{P,k}}A\hat{x}$ from the previous $\hat{b}$. If the corrector solution $\hat{x}$ is sparser than the corrected solution x_k then this update is likely to be faster than recomputing $\hat{b}$ from scratch. The same holds for computing the new objective function in line 9. We already store the differences $\frac{1}{\Delta_{P,k+1}}\hat{x}$ and $\frac{1}{\Delta_{D,k+1}}\hat{y}$ as sparse vectors when correcting the primal and dual solution in lines 20 and 21.

This is a heuristic minimization of the number of arithmetic operations performed in rational arithmetic since we do not take into account a potentially unequal distribution of nonzeros over the constraint matrix. However, it promises a good approximation. Especially when the basis between two subsequent floating-point solves does not change much or at all, which is typical after a few refinement rounds, the primal update vectors will contain only entries for basic variables; as a result, they will typically be much sparser than the corrected solution.

5.2 Setup

The goal of our experiments was threefold. First we wanted to analyze the behavior of the plain iterative refinement procedure for computing high-precision solutions. Does it always converge and how fast does it converge, both in terms of the number of refinements and the time spent? How much time is spent in the refinement phase compared to the initial floating-point solve? How expensive are the rational arithmetic operations performed? Are there differences between numerically easy and difficult instances?

To this end, we performed the plain floating-point solve of standard SoPlex without iterative refinement with a primal and dual tolerance of 10^{-9} and compared it to iterative refinement runs with both primal and dual tolerances of 10^{-50}, and 10^{-250}. We will denote these settings by SoPlex$_9$, SoPlex$_{50}$, and SoPlex$_{250}$, respectively. In all cases, we parsed LP files exactly and compute violations in rational arithmetic.

Second, we wanted to know whether the iterative refinement procedure always converges to an optimal basis and how many refinements are needed? For this, we used the basis verification tool perPlex of Koch (2004).

Third, we wanted to investigate whether iterative refinement helps to improve the performance of exact LP solving. A natural idea to test this is to warm start the QSopt_ex solver—which may today be considered as the state-of-the-art in general-purpose exact LP solving—from the advanced starting basis obtained after several rounds of iterative refinement, at the end of SoPlex$_{50}$, say, and measure the total running time of SoPlex$_{50}$ and QSopt_ex. Comparing this to the plain QSopt_ex performance would be biased, however, since the underlying floating-point simplex implementations in SoPlex and QSopt_ex are significantly different. (QSopt_ex, for instance, does not implement presolving techniques.) As a meaningful point of reference we use the performance of QSopt_ex when warm started from the basis returned by SoPlex$_9$. We measured whether iterative refinement decreases the running time and maximum precision used by QSopt_ex.

At last, we need to comment on the parameter settings used for the floating-point LP solves. While by default, SoPlex uses an absolute feasibility tolerance of 10^{-6}, in our experiments, we have tightened it to 10^{-9}. Generally, using a stricter tolerance will return higher-precision corrector solutions, but the same time too strict of a tolerance can lead to a numerical breakdown, which is why we did not use a value of 10^{-12}. Presolving and scaling were applied when solving LPs from scratch, but not when hot starting from an advanced basis.

Hardware and software. The experiments were conducted on a cluster of 64-bit Intel Xeon X5672 CPUs at 3.2 GHz with 12 MB cache and 48 GB main memory, simultaneously running only one job per node. We used a time limit of two hours per instance for each SoPlex, perPlex, and QSopt_ex run. We used the SoPlex developer version 2.0.0.2 implementing the iterative refinement algorithm with features and parameters as described above. SoPlex was compiled with GCC 4.8.2 and linked to the external libraries GMP 5.1.3, EGlib 2.6.20, and zlib 1.2.8 for reading compressed instance files.

Instances. We compiled a large test set of 1,202 primal and dual feasible LPs from several sources: the Netlib LP test set including the *kennington* folder, Hans Mittelmann's benchmark instances, Csaba Mészáros's LP collection, the LP relaxations of the COR@L mixed-integer programming test set, and the LP relaxations of the mixed-integer programs from MIPLIB, MIPLIB 2, MIPLIB 3, MIPLIB 2003, and MIPLIB 2010. A detailed description of this collection with problem statistics is given in Table 4 of Appendix C.

5.3 Computational results

Let us first look at two individual instances, `momentum3` from MIPLIB 2003 and `world` from Mészáros's test set. These are nontrivial instances with 949,495 nonzeros and 164,470 nonzeros in their constraint matrix, respectively. Table 1 shows how the primal and dual violations progress with time and number of simplex iterations elapsed. Because we are mainly interested in the order of magnitude of the violations, we only report the precision as the rounded negative base 10 logarithm.

Instance `momentum3` shows steady convergence. Each iteration added between seven and 16 orders of magnitude to the precision of the solution, such that after 18 refinement rounds primal and dual violations below 10^{-250} are reached. The initial floating-point solve with 46,184 iterations consumes the largest part of the running time. Only during the first refinement LP one more simplex iteration is performed. After that the basis remains unchanged. The refinement phase only incurs an overhead of 4.2% in running time and is dominated by rational arithmetic. The hot start proves very efficient and the simplex time for solving the refinement LPs is negligible. The time consumed by rational arithmetic grows only slightly as the precision of the solution improves, from 0.3 seconds for the first to 0.5 seconds for the last refinement. This is largely due to positive effect of rounding the scaling factors to powers of two; in an earlier implementation without this feature the last refinements consumed almost two seconds. When refining further, this slowdown became increasingly pronounced.

The observed distribution of running time is typical and similar for the instance `world`. Numerically, however, it seems to be more challenging to SoPlex and the convergence of iterative refinement is slower. Simplex pivots, although comparatively few, are performed up to refinement round twelve until the final basis is reached. In the rounds where pivots are performed, the dual violation does not decrease, meaning that the floating-point corrector solutions returned by SoPlex must exhibit high absolute violation in dual multipliers or reduced costs. This violation of Assumption 3.5 is explained as follows. In the transformed LP, nonbasic variables with (dual) feasible reduced cost may have large objective coefficients, because they do not limit the dual scaling factor. If these are pivoted into the basis their reduced cost should be zero. Solving the linear system for the dual

Table 1: Progress of iterative refinement for instances `momentum3` and `world`

R — number of refinements
iter — number of simplex iterations elapsed
t — time elapsed (in seconds)
t_{rat} — time spent on rational arithmetic (cumulative, in seconds)
δ_P — maximum primal violation (rounded negative $\log_{10}$)
δ_D — maximum dual violation (rounded negative $\log_{10}$)

	`momentum3`					`world`				
R	iter	t	t_{rat}	δ_P	δ_D	iter	t	t_{rat}	δ_P	δ_D
0	46184	190.4	0.3	8	10	70204	131.5	0.1	8	11
1	46185	191.0	0.6	23	17	70256	131.7	0.2	17	12
2	46185	191.3	1.0	37	33	70282	131.9	0.4	22	13
3	46185	191.7	1.3	51	48	70287	132.1	0.5	23	13
4	46185	192.0	1.7	65	65	70289	132.3	0.7	23	14
5	46185	192.4	2.1	80	78	70292	132.4	0.9	24	13
6	46185	192.8	2.4	93	93	70292	132.6	1.0	24	29
7	46185	193.2	2.8	108	107	70315	132.8	1.2	34	13
8	46185	193.6	3.2	121	120	70315	133.0	1.4	34	29
9	46185	194.0	3.6	136	134	70319	133.2	1.6	40	13
10	46185	194.4	4.0	150	149	70319	133.4	1.7	40	29
11	46185	195.1	4.7	165	164	70319	133.6	1.9	40	45
12	46185	195.6	5.1	178	177	70320	133.8	2.1	55	15
13	46185	196.0	5.6	193	191	70320	134.0	2.3	70	30
14	46185	196.4	6.0	207	205	70320	134.2	2.5	85	46
15	46185	196.9	6.4	222	220	70320	134.4	2.7	100	61
16	46185	197.4	6.9	235	234	70320	134.6	2.9	115	77
17	46185	197.9	7.4	250	248	70320	134.8	3.1	130	93
18	46185	198.4	7.9	264	262	70320	135.0	3.3	146	107

solution vector, however, only yields a precision that is relative w.r.t. the (large) objective coefficients.

As we can see in Table 1, after the first refinement round without pivots, round four, the dual violation starts to decrease below 10^{-29}. In the next round, however, new pivots occur, and the maximum dual violation even falls back to 10^{-13}. Only from round 13 on both primal and dual precision improves continuously, in each round reducing the violations by about 15 orders of magnitude. Because of the setbacks during the refinement rounds with simplex pivots, the dual precision lags behind the primal precision. This could be even more pronounced, but because we limit the primal scaling factor by the dual scaling factor in our implementation, the primal precision stalls during some rounds. After 29 refinement rounds (not shown in the table) a maximum violation below 10^{-250}, comparable to the precision on `momentum3` after 18 rounds, is reached.

General Results. We move on to discuss the results over the entire test set. Detailed results for each instance can be found in Table 5 of Appendix C. Out of the 1,202 instances in our test set, SoPlex$_{50}$ and SoPlex$_{250}$ converged successfully to the specified tolerance on 1,195 instances. On three instances,[6] they timed out because the floating-point solver could not solve the first refinement LP within the time limit. For three instances, the initial floating-point solve (equivalent to SoPlex$_9$) incorrectly claimed unboundedness and for one instance it incorrectly claimed infeasibility.[7] In all cases, SoPlex$_{50}$ and SoPlex$_{250}$ rejected these claims successfully using the feasibility and unboundedness tests described

[6] `neos-954925`, `neos-956971`, and `neos-957143`

[7] `de063157`, `l30`, and `stat96v1` were claimed to be unbounded, `neos-1603965` was claimed to be infeasible by SoPlex$_9$

in Section 4.1, but after starting to refine the original LP again, floating-point SoPlex failed to return an approximately optimal solution even when run with different settings. Furthermore, for five of the 1,195 instances[8] the floating-point solver claimed one of the intermediate refined LPs infeasible. SoPlex$_{50}$ and SoPlex$_{250}$ rejected these claims and continued to converge to their target tolerance.

Results for the Floating-Point Solves. For four instances SoPlex$_9$ claimed a wrong status,[7] and for 99 instances it returned a numerical solution that exhibited violations *above* 10^{-9}, though most of those only slightly. However, for the instance `de063155` from Mészáros's "problematic" test set, which features constraint coefficients with absolute values ranging from approximately 10^{-7} to 10^{12} SoPlex$_9$ even returned a completely meaningless solution violating primal and dual feasibility by almost 10^3 and 10^8, respectively. This instance is solved correctly by SoPlex$_{50}$ and SoPlex$_{250}$ in seven and 20 refinements, respectively.

For 1,024 instances perPlex verified that the basis returned by SoPlex$_9$ was indeed optimal. For 95 instances, the SoPlex$_9$ basis was detected as primal infeasible, for 30 instances as dual infeasible, and for six instances as both primal and dual infeasible. On the remaining instances, perPlex hit a time or memory limit, or—as for the instance `neos-619167`—it could not handle free nonbasic variables correctly. Hence, our test set seems to contain a non-negligible number of numerically challenging instances. Although for most instances, we can confirm the conclusion of Dhiflaoui et al. (2003) and Koch (2004) that floating-point LP solvers often succeed in returning optimal bases (on the Netlib test set), we have to relativize this finding for more than 14% of the instances.

Results for Iterative Refinement. As mentioned above, on all 1,195 instances, both SoPlex$_{50}$ and SoPlex$_{250}$ converged to the specified tolerance. On all instances that perPlex could handle, it verified that the final basis returned was primal and dual feasible. For 61 instances, they even terminated with an exactly optimal numeric solution with a zero primal and dual violation.

For the vast majority of instances, the basis after the initial floating-point solve was already the final basis later returned. On 93 and 30 instances, one and two refinement rounds, respectively, were needed to reach the final (optimal) basis. Only for instances `de063155`, `mod2`, and `world`, three, eleven, and twelve rounds, respectively, were performed until the final basis. This happened already when using SoPlex$_{50}$, so the additional rounds performed by SoPlex$_{250}$ essentially only refine the primal and dual solutions to the linear systems defined by the basis matrix. Furthermore, for the five instances[8] for which feasibility was tested, only few refinements with actual pivots were performed.

Average Performance Comparison. In order to compare performance of the three runs on subsets of instances with varying numerical difficulty, we categorized the instances according to the number of refinement rounds performed until the final basis was reached, denoted by R_0. We excluded the five instances[8] where the infeasibility test was triggered, because their run actually amounts to the refinement of two LPs. We excluded simple instances that were solved in under two seconds by each algorithm.

For the resulting 356 instances, Table 2 reports the average number of refinement rounds, simplex iterations, and the average running time over these subsets. Because simplex iterations and running times vary drastically across instances, we computed their averages not as arithmetic means—which would introduce a bias towards large values—but as shifted geometric means. We use shifts of two seconds and 100 simplex iterations. For the number of refinements we report arithmetic averages.

For SoPlex$_{50}$, most notably, the average running time increases by only 3% for the easy subset $R_0 = 0$ and by only 7% for the set $R_0 \geqslant 1$ of instances with at least one

[8] `fome11,12,13`, `rail01`, and `shs1023`

Table 2: Computational comparison of iterative refinement and pure floating-point performance

R_0 — number of refinements to final basis
N — number of instances in this R_0-class
R — number of refinements (arithmetic mean)
iter — number of simplex iterations (shifted geometric mean)
t — total running time (shifted geom. mean in seconds)
Δt — relative running time w.r.t. SoPlex$_9$

		SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
R_0	N	iter	t	R	iter	t	Δt	R	iter	t	Δt
0	326	19291.8	20.0	3.0	19291.8	20.6	1.03	16.4	19291.8	22.7	1.14
1	26	16936.8	20.6	3.9	17918.2	22.3	1.08	17.8	17918.2	25.1	1.22
2	2	79895.4	90.3	4.5	79964.6	92.8	1.03	17.5	79964.6	99.0	1.10
11	1	58340.0	100.4	14.0	58534.0	103.8	1.03	26.0	58534.0	104.2	1.04
12	1	70204.0	131.2	15.0	70320.0	132.1	1.01	29.0	70320.0	137.6	1.05
$\geqslant 1$	30	20531.5	25.7	4.6	21562.4	27.6	1.07	18.5	21562.4	30.7	1.19

refinement round with simplex pivots. This is reflected in the small number of additional simplex iterations that are performed during the refinement rounds. Furthermore, the average number of refinement rounds is often smaller than expected. To achieve a maximum violation of 10^{-250} by floating-point solves with tolerance 10^{-9}, one would have to estimate approximately $250/9 - 1 \approx 26.8$ refinement rounds. By contrast, even the class $R_0 \geqslant 1$ shows only 18.5, indicating that most floating-point solves—in particular the final ones—return higher-precision solutions. This is slightly less pronounced for SoPlex$_{50}$.

The overhead in the running time of SoPlex$_{250}$ beyond SoPlex$_{50}$ stems only from refining the primal and dual solution to the linear systems defined by the basis matrix. We note that the increase in time of SoPlex$_{250}$ to SoPlex$_9$ is still very small on average—14% for $R_0 = 0$ and 19% for $R_0 \geqslant 1$. Incorporating some of the more sophisticated techniques recently developed by Wan (2006), Pan (2011), or Saunders et al. (2011) may close this gap even further.

Accelerating exact LP. Finally, we compared the performance of QSopt_ex when warm started from SoPlex$_9$ and SoPlex$_{50}$, respectively. Of the 1,195 instances for which both SoPlex$_9$ and SoPlex$_{50}$ returned basic solutions, 1,166 instances could be solved by both version. Nine instances could be solved only when warm starting from the advanced basis returned by SoPlex$_{50}$ (within running times between 108 and 2026 seconds) while starting from SoPlex$_9$'s basis QSopt_ex hit the time limit of two hours.[9] 20 instances could not be solved by either version. Detailed results can be found in Table 6 of Appendix C.

Table 3 gives an aggregated comparison over the 307 instances that were solved by both versions, but were nontrivial in the sense that at least one version took more than two seconds. It compares the total number of iterations taken (by SoPlex and QSopt_ex), the total running time, and additionally states the running times of SoPlex and QSopt_ex individually. We again report shifted geometric means, using a shift of two seconds and 100 simplex iterations. (Note that it is hence correct that the values in columns $t_{9/50}$ and t_{ex} do not exactly add up to the t-values.) In order to distinguish numerically easy from difficult instances, we again categorized them by the number of refinement rounds (R_0) needed by SoPlex$_{50}$ in order to reach the final basis.

[9] `fome13` (in 328.6 seconds), `mod2` (in 108.1 seconds), `momentum3` (in 2025.5 seconds), `ofi` (in 432.2 seconds), `sgpf5y6` (in 228.7 seconds), `shs1023` (in 625.4 seconds), `watson_1` (in 379.8 seconds), `watson_2` (in 1538.1 seconds), and `world` (in 136.6 seconds).

Table 3: Computational comparison of QSOPT_EX's performance when warm started from bases returned by SOPLEX$_9$ and SOPLEX$_{50}$

R_0 — number of refinements to final basis
N — number of instances in this R_0-class
iter — number of simplex iterations by SOPLEX+QSOPT_EX (shifted geom. mean)
B — number of precision boosts in QSOPT_EX (total over all LPs in this R_0-class)
$t_{9/50}$ — running time of SOPLEX$_{9/50}$ (shifted geom. mean in seconds)
t_{ex} — running time of QSOPT_EX (shifted geom. mean in seconds)
t — total running time (shifted geom. mean in seconds)
Δt — relative total running time w.r.t. SOPLEX$_9$ +QSOPT_EX

		SOPLEX$_9$+QSOPT_EX					SOPLEX$_{50}$+QSOPT_EX					
R_0	N	iter	B	t_9	t_{ex}	t	iter	B	t_{50}	t_{ex}	t	Δt
0	284	22306.5	2	20.7	3.1	27.2	22306.5	2	21.2	3.1	27.5	1.01
1	22	23812.9	13	18.1	110.2	168.7	16423.4	0	19.4	4.6	31.0	0.18
2	1	51469.0	1	18.8	240.8	259.6	45950.0	0	19.7	2.6	22.2	0.09
$\geqslant 1$	23	24625.4	14	18.1	114.0	171.9	17176.3	0	19.4	4.5	30.6	0.18

On the instances for which iterative refinement did not change the basis, the QSOPT_EX performance is necessarily identical. It never increased the simplex precision beyond 64 bit except for the instance `maros-r7`, for which it started pivoting despite the optimality of the starting basis and performed to precision boosts to 192 bit. The total running time on these instances increases by only 1% on average, corresponding to the small overhead of iterative refinement. On the contrary, for the 23 instances for which iterative refinement affected the final basis the performance gain is drastic: the total running time is reduced to only 18%, i.e., by a factor 5.5 on average. (Note that this does not even include the nine instances[9] on which SOPLEX$_9$+QSOPT_EX timed out.) This improvement cannot be explained by the reduction in simplex iterations alone. Most importantly, warm starting from the basis returned by SOPLEX$_{50}$, no precision boosts were performed and so the expensive pivots performed by QSOPT_EX in extended-precision were reduced to zero.

6 Conclusion

We have presented a new algorithm to solve linear programs to high levels of precision. It extends the idea of iterative refinement for linear systems of equations by Wilkinson (1963) to the domain of optimization problems by simultaneously correcting primal and dual residual errors. Algebraically, it builds up an increasingly accurate solution by solving a sequence of LPs, which differ only in the bounds of the variables, the sides of the constraints, and the objective function coefficients, to fixed precision. Geometrically, it can be viewed as zooming further and further into the area of interest around the refined solution. While it is designed to work with an arbitrary LP oracle, it combines especially well with the hot-starting capabilities of the simplex method. For infeasible and unbounded instances it can be used to compute high-precision certificates via an auxiliary reformulation of the LP. In this context, we have developed an algorithm to convert an approximate Farkas proof into a rigorous infeasibility box centered at the origin that helps users to understand the domains in which feasible solutions can or cannot exist.

For a simplex-based implementation we demonstrated it to be efficient in practice: On a large test set of publicly available benchmark instances, computing solutions up to a precision of 10^{-50} incurred an average slowdown of only 3% on numerically easy and 7% on numerically difficult LPs. In addition we saw that the basis corresponding to the refined solution was always optimal. We exploited this in order to warmstart the exact LP solver QSOPT_EX. As a result, we observed a more than five times speedup on difficult instances

and could solve nine more instances than QSopt_ex alone.

As with classical iterative refinement the algorithm shares the limitation that it breaks down when the LP is too ill-conditioned for the underlying floating-point routine. This could be overcome by increasing the working precision of the underlying floating-point LP solver whenever necessary in a similar fashion as in the exact LP solver QSopt_ex discussed in Section 2.1. As a final remark, we note that some applications may require extended-precision solutions, but not need exact solutions. In such cases iterative refinement can be used to meet this demand without requiring the amount of time taken by an exact LP solver, giving it a competitive advantage.

Acknowledgements. Ambros M. Gleixner was supported by the Research Campus Modal "Mathematical Optimization and Data Analysis Laboratories" funded by the German Ministry of Education and Research. Kati Wolter was supported by the DFG Priority Program 1307 "Algorithm Engineering".

References

E. Althaus and D. Dumitriu. Fast and accurate bounds on linear programs. In J. Vahrenhold, editor, *Proc. 8th International Symposium on Experimental Algorithms*, volume 5526 of *LNCS*, pages 40–50. Springer, June 2009.

D. L. Applegate, W. Cook, S. Dash, and D. G. Espinoza. Exact solutions to linear programming problems. *Oper. Res. Lett.*, 35(6):693–699, 2007a.

D. L. Applegate, W. Cook, S. Dash, and D. G. Espinoza. QSopt_ex, 2007b. `http://www.dii.uchile.cl/~daespino/ESolver_doc/`.

D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8(1):295–313, 1992. `doi:10.1007/BF02293050`.

D.-O. Azulay and J.-F. Pique. Optimized Q-pivot for exact linear solvers. In M. Maher and J.-F. Puget, editors, *Principles and Practice of Constraint Programming CP98*, volume 1520 of *Lecture Notes in Computer Science*, pages 55–71. Springer, 1998. `doi:10.1007/3-540-49481-2_6`.

C. Buchheim, M. Chimani, D. Ebner, C. Gutwenger, M. Jünger, G. Klau, P. Mutzel, and R. Weiskircher. A branch-and-cut approach to the crossing number problem. *Discrete Optimization*, 5(2):373 – 388, 2008.

D. A. Bulutoglu and D. M. Kaziska. Improved WLP and GWP lower bounds based on exact integer programming. *Journal of Statistical Planning and Inference*, 140(5):1154–1161, 2010. `doi:10.1016/j.jspi.2009.10.013`.

B. A. Burton and M. Ozlen. Computing the crosscap number of a knot using integer programming and normal surfaces. *ACM Transactions on Mathematical Software*, 39(1):4:1–4:18, 2012. `doi:10.1145/2382585.2382589`.

L. Chindelevitch, J. Trigg, A. Regev, and B. Berger. An exact arithmetic toolbox for a consistent and reproducible structural analysis of metabolic network models. *Nature Communications*, 5, 2014. `doi:10.1038/ncomms5893`.

V. Chvátal. *Linear Programming*. W. H. Freeman and Company, New York, 1983.

H. Cohn, Y. Jiao, A. Kumar, and S. Torquato. Rigidity of spherical codes. *Geometry & Topology*, 15(4):2235–2273, 2011.

W. Cook and D. E. Steffy. Solving very sparse rational systems of equations. *ACM Trans. on Math. Software*, 37(4), 2011.

W. Cook, T. Koch, D. E. Steffy, and K. Wolter. An exact rational mixed-integer programming solver. In O. Günlük and G. Woeginger, editors, *Integer Programming and Combinatoral Optimization*, volume 6655 of *LNCS*, pages 104–116. Springer Berlin / Heidelberg, 2011.

G. B. Dantzig. *Linear programming and extensions*. Princeton University Press, Princeton, NJ, 1963.

F. M. de Oliveira Filho and F. Vallentin. Fourier analysis, linear programming, and densities of distance avoiding sets in $\mathbb{R}^n$. *J. Eur. Math. Soc.*, 12(6):1417–1428, 2010.

M. Dhiflaoui, S. Funke, C. Kwappik, K. Mehlhorn, M. Seel, E. Schömer, R. Schulte, and D. Weber. Certifying and repairing solutions to large LPs: How good are LP-solvers? In *Proceedings of the 14th annual symposium on Discrete algorithms*, SODA '03, pages 255–256, Philadelphia, PA, USA, 2003. SIAM.

J. Edmonds. Exact pivoting. Talk at ECCO VII, Milan, Italy, February 1994.

J. Edmonds and J.-F. Maurras. Note sur les Q-matrices d'Edmonds. *RAIRO. Recherche Opérationnelle*, 31(2):203–209, 1997. `www.numdam.org/item?id=RO_1997__31_2_203_0`.

D. G. Espinoza. *On Linear Programming, Integer Programming and Cutting Planes*. Ph.D. thesis, Georgia Institute of Technology, 2006.

K. Fukuda and A. Prodon. Double description method revisited. In M. Deza, R. Euler, and I. Manoussakis, editors, *Combinatorics and Computer Science*, volume 1120 of *Lecture Notes in Computer Science*, pages 91–111. Springer, 1996. `doi:10.1007/3-540-61576-8_77`.

B. Gärtner. Exact arithmetic at low cost – a case study in linear programming. *Computational Geometry*, 13(2):121–139, 1999. `doi:10.1016/S0925-7721(99)00012-7`.

A. M. Gleixner, D. E. Steffy, and K. Wolter. Improving the accuracy of linear programming solvers with iterative refinement. In *ISSAC '12. Proceedings of the 37th International Symposium on Symbolic and Algebraic Computation*, pages 187–194. ACM, July 2012. `doi:10.1145/2442829.2442858`.

G. Golub and C. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, USA, 1983.

M. Grötschel, L. Lovasz, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, Berlin / Heidelberg, 1988.

T. C. Hales. A proof of the Kepler conjecture. *Annals of Mathematics*, 162(3):1065–1185, 2005.

S. Held, W. Cook, and E. Sewell. Maximum-weight stable sets and safe lower bounds for graph coloring. *Math. Program. Comp.*, 4(4):363–381, 2012.

I. Hicks and N. McMurray. The branchwidth of graphs and their cycle matroids. *Journal of Combinatorial Theory Series B*, 97(5):681–692, 2007.

C. Jansson. Rigorous lower and upper bounds in linear programming. *SIAM Journal on Optimization*, 14(3):914–935, 2004. `doi:10.1137/S1052623402416839`.

L. G. Khachiyan. A polynomial algorithm in linear programming (in Russian). *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979. English translation: *Soviet Mathematics Doklady*, 20(1):191–194, 1979.

E. Klotz. Identification, assessment and correction of ill-conditioning and numerical instability in linear and integer programs. In A. Newman and J. Leung, editors, *TutORials in Operations Research: Bridging Data and Decisions*, pages 54–108. INFORMS, 2014.

T. Koch. The final NETLIB-LP results. *Operations Research Letters*, 32(2):138–142, 2004.

L. Ladányi. IBM T.J. Watson Research Center, Yorktown Heights, New York, USA. Personal communication, November 26, 2011.

J. A. Lerman, D. R. Hyduke, H. Latif, V. A. Portnoy, N. E. Lewis, J. D. Orth, A. C. Schrimpe-Rutledge, R. D. Smith, J. N. Adkins, K. Zengler, and B. O. Palsson. In silico method for modelling metabolism and gene product expression at genome scale. *Nature Communications*, 3, 2012. `doi:10.1038/ncomms1928`.

C. Maes. Gurobi Optimization, Inc. Personal communication, September 6, 2013.

R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, 2009. `doi:10.1137/1.9780898717716`.

A. Neumaier and O. Shcherbina. Safe bounds in linear and mixed-integer linear programming. *Mathematical Programming*, 99(2):283–296, 2004. `doi:10.1007/s10107-003-0433-3`.

V. Y. Pan. Nearly optimal solution of rational linear systems of equations with symbolic lifting and numerical initialization. *Computers & Mathematics with Applications*, 62(4):1685–1706, 2011.

J. Renegar. Some perturbation theory for linear programming. *Math. Program.*, 65:73–91, 1994. ISSN 0025-5610.

B. D. Saunders, D. H. Wood, and B. S. Youse. Numeric-symbolic exact rational linear system solver. In *Proceedings of the 36th international symposium on Symbolic and algebraic computation*, ISSAC '11, pages 305–312, New York, NY, USA, 2011. ACM.

M. A. Saunders and L. Tenenblat. The zoom strategy for accelerating and warm-starting interior methods. Talk at INFORMS Annual Meeting, Pittsburgh, PA, USA, November 2006. `http://www.stanford.edu/group/SOL/talks/saunders-tenenblat-INFORMS2006.pdf`.

A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, Chichester, UK, 1986.

D. E. Steffy and K. Wolter. Valid linear programming bounds for exact mixed-integer programming. *INFORMS Journal on Computing*, 25(2):271–284, 2013. `doi:10.1287/ijoc.1120.0501`.

S. Ursic and C. Patarra. Exact solution of systems of linear equations with iterative methods. *SIAM Journal on Matrix Analysis and Applications*, 4(1):111–115, 1983.

J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, Cambridge, UK, 2003.

Z. Wan. An algorithm to solve integer linear systems exactly using numerical methods. *J. of Symbolic Computation*, 41(6):621–632, 2006.

J. H. Wilkinson. *Rounding Errors in Algebraic Processes*. Prentice Hall, Englewood Cliffs, NJ, 1963.

R. Wunderling. *Paralleler und objektorientierter Simplex-Algorithmus*. PhD thesis, Technische Universität Berlin, 1996.

C. K. Yap. Robust geometric computation. In J. E. Goodman and J. O'Rourke, editors, *Handbook of discrete and computational geometry*, pages 653–668. CRC Press, Inc., Boca Raton, FL, USA, 1997. ISBN 0-8493-8524-5.

Appendix

A Proofs

A.1 Theorem 3.1

Proof. Proof. For primal feasibility, point 1, we must check that the violation of variable bounds and equality constraints is simply scaled by $1/\Delta_P$:

$$\big(x^* + \frac{1}{\Delta_P}\hat{x}\big) - \ell = \frac{1}{\Delta_P}(\hat{x} - \Delta_P(\ell - x^*)) = \frac{1}{\Delta_P}(\hat{x} - \Delta_P\hat{\ell})$$

and

$$A\big(x^* + \frac{1}{\Delta_P}\hat{x}\big) - b = \frac{1}{\Delta_P}(\Delta_P A x^* + A\hat{x} - \Delta_P b) = \frac{1}{\Delta_P}(A\hat{x} - \Delta_P\hat{b}).$$

For dual feasibility, point 2, we check the dual slacks,

$$c - A^\mathsf{T}\big(y^* + \frac{1}{\Delta_D}\hat{y}\big) = \frac{1}{\Delta_D}(\Delta_D c - \Delta_D A^\mathsf{T} y^* - A^\mathsf{T}\hat{y}) = \frac{1}{\Delta_D}(\Delta_D\hat{c} - A^\mathsf{T}\hat{y}).$$

Using this, point 3 on complementary slackness follows from the definition of the duality gap via

$$\begin{aligned}\gamma_P\big(x^* + \frac{1}{\Delta_P}\hat{x}, y^* + \frac{1}{\Delta_D}\hat{y}\big) &= \big(\big(x^* + \frac{1}{\Delta_P}\hat{x}\big) - \ell\big)^\mathsf{T}\big(c - A^\mathsf{T}\big(y^* + \frac{1}{\Delta_D}\hat{y}\big)\big)\\ &= \frac{1}{\Delta_P\Delta_D}(\hat{x} - \Delta_P\hat{\ell})^\mathsf{T}(\Delta_D\hat{c} - A^\mathsf{T}\hat{y})\\ &= \gamma_{\hat{P}}(\hat{x},\hat{y})/\Delta_P\Delta_D\end{aligned}$$

where γ_P and $\gamma_{\hat{P}}$ represent the duality gaps of the problems P and $\hat{P}$, respectively. And since a solution is optimal if and only if it is primal and dual feasible and complementary slack, these first points entail point 4.

Finally, a solution is basic if there is a regular basis $\mathcal{B}$ such that the nonbasic variables—the variables with index $i \notin \mathcal{B}$—are at their bounds and the basic variables have zero reduced cost. In the following equivalence, the left-hand equations are the conditions for $\hat{x}, \hat{y}$ as solution of $(\hat{P})$, while the right-hand equations are the conditions for $x^* + \frac{1}{\Delta_P}\hat{x}, y^* + \frac{1}{\Delta_D}\hat{y}$ w.r.t. the same basis for (P):

$$\begin{aligned}\hat{x}_i = \Delta_P\hat{\ell}_i \Leftrightarrow x_i^* + \frac{\hat{x}_i}{\Delta_P} = \ell_i \text{ for all } i \notin \mathcal{B},\\ A_\mathcal{B}^\mathsf{T}\hat{y} = \Delta_D\hat{c}_\mathcal{B} \Leftrightarrow A_\mathcal{B}^\mathsf{T}\big(y^* + \frac{1}{\Delta_D}\hat{y}\big) = c_\mathcal{B}.\end{aligned}$$

This proves point 5. □

A.2 Corollary 3.6

Proof. Proof. The result is intuitive. We prove all points together by induction over k. For $k = 1$ they hold trivially. Consider $k+1$, $k \geqslant 1$. Because points 2–4 hold for x_k, y_k, their violations satisfy $\delta_{P,k}, \delta_{D,k} \leqslant \tilde{\epsilon}^k$. Because point 1 holds for k, $\alpha\Delta_{P,k} \geqslant \alpha/\tilde{\epsilon}^{k-1} \geqslant 1/\tilde{\epsilon}^k$; analogously, $\alpha\Delta_{D,k} \geqslant 1/\tilde{\epsilon}^k$. Hence, in lines 15 and 16 of the algorithm we have

$$\Delta_{P,k+1}, \Delta_{D,k+1} \geqslant 1/\tilde{\epsilon}^k \tag{11}$$

proving point 1.

If we let $x^* = x_k$, $y^* = y_k$, $\Delta_P = \Delta_{P,k+1}$, and $\Delta_D = \Delta_{D,k+1}$ in Theorem 3.1, then $(\hat{P})$ is the exact shifted and scaled LP before rounding it to the working precision of the LP solver. $(\hat{P})$ is primal and dual feasible because it is simply an affine transformation of the original LP (P). By Assumption 3.5, the LP solver returns $\hat{x}, \hat{y}$ that are primal and dual feasible for $(\hat{P})$ within absolute tolerance $\epsilon \leqslant \tilde{\epsilon}$. By Theorem 3.1, the corrected solution x_{k+1}, y_{k+1} violates primal and dual feasibility for the original LP by at most $\tilde{\epsilon}/\Delta_{P,k+1}$ and $\tilde{\epsilon}/\Delta_{D,k+1}$, respectively. By (11), this is at most $\tilde{\epsilon}^{k+1}$, proving points 2–4 for $k+1$.

Finally, by Assumption 3.5, the violation of complementary slackness $|\gamma(\hat{x}, \hat{y})|$ in $(\hat{P})$ is at most σ. Using again Theorem 3.1 with $\epsilon_S = \sigma$ we get

$$|\gamma(x_{k+1}, y_{k+1})| \leqslant \sigma/(\Delta_{P,k+1}\Delta_{D,k+1}) \overset{(11)}{\leqslant} \sigma\tilde{\epsilon}^{2k},$$

proving point 5 for $k+1$.

Assuming slowest convergence gives $\tilde{\epsilon}^k \leqslant \epsilon_P$, $\tilde{\epsilon}^k \leqslant \epsilon_D$, and $\sigma\tilde{\epsilon}^{2k} \leqslant \epsilon_S$, in the termination conditions in line 13 of the Algorithm 2, which is equivalent to the stated bound on the number of refinement rounds. □

A.3 Lemma 4.1

Proof. Proof. 1. The zero vectors are feasible in the primal and the dual of (4). 2. A primal solution x^* for (1) gives the primal solution $(x^* - \ell, 1)$ for (4), and $(\xi^*, 1)$ can be mapped back as $\xi^* + \ell$. 3. The dual of (4) is $\min\{z \mid A^\mathsf{T} y \leqslant 0, (b - A\ell)^\mathsf{T} y + z = 1, z \geqslant 0\}$. If its optimal value is less than one then for an optimal dual solution y^*, $(b - A\ell)^\mathsf{T} y^* = 1 - z^* > 0$, but $A^\mathsf{T} y^* \leqslant 0$. 4. $\|A(\frac{1}{\tau^*}\xi^* + \ell) - b\|_\infty = \|A\xi^* - (b - A\ell)\tau^*\|_\infty/\tau^*$ and $\xi^* \geqslant -\epsilon_P \Rightarrow \frac{1}{\tau^*}\xi^* + \ell \geqslant \ell - \epsilon_P/\tau^*$. □

B Iterative refinement for LPs in general form

B.1 Variables with general bounds

For clarity of presentation we have so far only considered lower bounds on the variables. Variables with additional upper bounds, i.e., $\ell \leqslant x \leqslant u$, $\ell, u \in \mathbb{R}^n$, require only few modifications in Algorithm 2. For the primal refinement we need to compute $\hat{u} \leftarrow u - x_k$ in line 8 and consider it in the calculation of the primal violation line 10,

$$\delta_{P,k} \leftarrow \max\{\max_{j=1,\dots,m} |\hat{b}_j|, \max_{i=1,\dots,n} \hat{\ell}_i, \max_{i=1,\dots,n} -\hat{u}_i\}.$$

In line 17, we also need to compute and round the transformed upper bound vector

$$\bar{u} \leftarrow \Delta_{P,k+1}\hat{u}.$$

The dual-refinement is only affected by the calculation of the dual scaling factor. With upper bounds, the dual LP contains separate dual slack variables for the lower and the upper bound of each variable and the dual LP (2) reads

$$\max\{b^\mathsf{T} y + \ell^\mathsf{T} z^\ell - u^\mathsf{T} z^u \mid A^\mathsf{T} y + z^\ell - z^u = c, z^\ell, z^u \geqslant 0\}. \tag{12}$$

The dual slack, or reduced cost, vector $z = c - A^\mathsf{T} y$ is split into $z = z^\ell - z^u$, where z^ℓ is associated with the lower bounds and z^u with the upper bounds. The duality gap, our measure for the violation of complementary slackness, becomes

$$\gamma(x, y) \coloneqq (x - \ell)^\mathsf{T} z^\ell + (u - x)^\mathsf{T} z^u.$$

If all variables have finite lower and upper bounds then each dual solution y gives a feasible solution to (12) if we let z^ℓ be the positive and z^u be the negative part of z. However, this ignores its connection to the primal solution. Suppose some variable x_i with domain $[0, U]$ is tight at its lower bound, $x_i = 0$, but its reduced reduced cost $z_i = -10^{-9}$ is slightly negative. Setting $z_i^\ell = 0$ and $z_i^u = 10^{-9}$ would guarantee dual feasibility, but the variable would contribute $U \cdot 10^{-9}$ to the duality gap.

A more suitable definition of the dual violation in line 11 is obtained if we distribute the reduced cost value not according to its sign, but associate it with the bound which is closest to the primal solution value via

$$\begin{aligned}\delta_{D,k} \leftarrow \max\{0, \max\{-\hat{c}_i \mid i = 1, \dots, n, (x_k)_i \leqslant (\ell_i + u_i)/2\},\\ \max\{\hat{c}_i \mid i = 1, \dots, n, (x_k)_i > (\ell_i + u_i)/2\}\}\end{aligned}$$

where $\hat{c}$ is the reduced cost vector computed as before in line 9. The according violation of complementary slackness is calculated as

$$\delta_{S,k} \leftarrow \Big| \sum_{i:(x_k)_i \leqslant (\ell_i+u_i)/2} -\hat{\ell}_i \hat{c}_i + \sum_{i:(x_k)_i > (\ell_i+u_i)/2} \hat{u}_i \hat{c}_i \Big|$$

in line 12.

This definition is also meaningful if one of the bounds is $\pm\infty$. If the variable is free we include the absolute value of its reduced cost in the dual violation and exclude it from the violation of complementary slackness. This has the same effect as if splitting the variable into its positive and negative part.

Remark B.1. If the floating-point solver returns basic solutions, one alternative is to directly use the basis information in the definition of dual feasibility. This is described in more detail in Section 5.1 together with our extension of the LP solver SoPlex, which implements the revised simplex algorithm.

B.2 Inequality constraints

For clarity of presentation, we have so far only considered equality constraints. In the case of an LP with ranged rows

$$\min\{c^\mathsf{T}x \mid L \leqslant Ax \leqslant U, \ell \leqslant x \leqslant u\} \tag{13}$$

with some $L_i \neq U_i$, the primal refinement only requires a small adjustment of Algorithm 2. We refer to (13) as the *general form* of an LP. In line 7, both the left-hand and right-hand side vector has to be shifted via $\hat{L} \leftarrow L - Ax_k$ and $\hat{U} \leftarrow U - Ax_k$, and the computation of the primal violation in line 10 must include $\max_{j=1,\dots,m} \hat{L}_j$ and $\max_{j=1,\dots,m} -\hat{U}_j$.

The dual-refinement step, however, does not allow for an equally straightforward generalization. The dual LP (2) now reads

$$\begin{aligned}\max\{L^\mathsf{T}y^L - U^\mathsf{T}y^U + \ell^\mathsf{T}z^\ell - u^\mathsf{T}z^u \mid A^\mathsf{T}(y^L - y^U) + z^\ell - z^u = c,\\ y^L, y^U, z^\ell, z^u \geqslant 0\},\end{aligned}$$

i.e., the dual vector is split into $y = y^L - y^U$, where y^L is associated with the left-hand sides and y^U with the right-hand sides. If entries in L and U are $\pm\infty$, the corresponding dual variables are left out. The duality gap becomes

$$\gamma(x, y) \coloneqq (x - \ell)^\mathsf{T}z^\ell + (u - x)^\mathsf{T}z^u + (Ax - L)^\mathsf{T}y^L + (U - Ax)^\mathsf{T}y^U. \tag{14}$$

Similar to the reduced costs for variables with lower and upper bounds in Section B.1, the dual multipliers should not be distributed to y^L and y^U according to their sign, but according to the row activity of the primal solution, i.e., the dual violation in line 11 should be computed as

$$\begin{aligned}\delta_{D,k} \leftarrow \max\{0, &\max\{-\hat{c}_i \mid i = 1,\dots,n, (x_k)_i \leqslant (\ell_i + u_i)/2\},\\ &\max\{\hat{c}_i \mid i = 1,\dots,n, (x_k)_i > (\ell_i + u_i)/2\},\\ &\max\{-(y_k)_j \mid j = 1,\dots,m, A_{j\cdot}x_k \leqslant (U_j + L_j)/2\},\\ &\max\{(y_k)_j \mid j = 1,\dots,m, A_{j\cdot}x_k > (U_j + L_j)/2\}\},\end{aligned}$$

and the violation of complementary slackness in line 12 as

$$\begin{aligned}\delta_{S,k} \leftarrow \Big| &\sum_{i:(x_k)_i \leqslant (\ell_i+u_i)/2} -\hat{\ell}_i\hat{c}_i + \sum_{i:(x_k)_i > (\ell_i+u_i)/2} \hat{u}_i\hat{c}_i\\ &+ \sum_{j:A_{j\cdot}x \leqslant (U_j+L_j)/2} -\hat{L}_j(y_k)_j + \sum_{j:A_{j\cdot}x > (U_j+L_j)/2} \hat{U}_j(y_k)_j\Big|.\end{aligned}$$

Where $A_{j\cdot}$ denotes the j-th row of A. Some adjustments must be made to Algorithm 2 in order for it to work properly on such general form LPs. In the case of equality constraints $Ax = b$, replacing the objective function vector by the reduced cost vector $c - A^\mathsf{T}y_k$ is an equivalent transformation because it amounts to subtracting the constant offset $(A^\mathsf{T}y_k)^\mathsf{T}x = (Ax)^\mathsf{T}y_k = b^\mathsf{T}y_k$ from the objective function value of any primal solution x. This argument falls short if the row activity Ax may vary between L and U. In order to compensate for this, the activity of each row must be considered in the objective function with its dual multiplier as objective coefficient.

The naïve solution is to call Algorithm 2 for the reformulated LP with slack variables

$$\min\{c^\mathsf{T}x + 0^\mathsf{T}s \mid Ax - s = 0, \ell \leqslant x \leqslant u, L \leqslant s \leqslant U\}, \tag{15}$$

where for notational simplicity we write slacks also for equality constraints. While dual feasibility is identical for (15) and (13) this does not hold for primal feasibility. If (x, s) is primal feasible for (15) with maximum violation of ϵ_P then x is primal feasible for (13) with maximum violation of $2\epsilon_P$ because the violation of $L \leqslant s \leqslant U$ and $Ax - s = 0$ may

add up. Additionally, because for an approximate primal solution the constraints do not hold exactly, i.e., only $Ax \approx s$ we should check complementary slackness in terms of (14) to meet the termination conditions in the original LP.

We obtain a slightly more involved implementation if we note that it suffices to introduce slack variables in the floating-point solver and work with the original LP in the main algorithm. The first approximate LP solve in line 4 can be performed without slack variables. During the refinement loop in line 18 we then solve

$$\min\{\bar{c}^\mathsf{T} x + \bar{y}^\mathsf{T} s \mid \bar{A}x - s = 0, \bar{\ell} \leqslant x \leqslant \bar{u}, \bar{L} \leqslant s \leqslant \bar{U}\} \tag{16}$$

where $\bar{y} \approx \Delta_{D,k+1} y_k$, $\bar{L} \approx \Delta_{P,k+1}\hat{L}$ and $\bar{U} \approx \Delta_{P,k+1}\hat{U}$. The approximate solution values of the slack variables are ignored. As before, only $\hat{x}, \hat{y}$ are used for the subsequent correction.

This approach can be beneficial for the performance of the algorithm if the floating-point solver already uses slack variables internally and provides access to set their objective coefficients to a value different than zero, or if it has another more efficient way of solving the slack formulation, this becomes possible. We exploit this in the implementation described in Section 5.1. We conclude this section with a discussion of an alternative, overly simplistic strategy for handling inequality constraints that we have experimented with.

Remark B.2 (Naïve treatment of inequality constraints)**.** The slack formulation (16) could be simplified by substituting and removing the slack variables exactly as a modern LP solver would do in its presolving phase. This would result in

$$\min\{(\bar{c} + \bar{A}^\mathsf{T}\bar{y})^\mathsf{T} x \mid \bar{L} \leqslant \bar{A}x \leqslant \bar{U}, \bar{\ell} \leqslant x \leqslant \bar{u}\},$$

where the objective function vector $\bar{c} + \bar{A}^\mathsf{T}\bar{y} \approx \Delta_{D,k+1} c$. From this observation one could think of using a refinement step where the objective function is not replaced by the scaled reduced cost vector, but by $\Delta_{D,k+1}(c - \sum_{j:L_j=U_j} y_j A_{j.}^\mathsf{T})$, i.e., we only subtract the row vectors corresponding to equality constraints.

The same objective function is obtained when forcing the dual multipliers of all non-equality constraints to zero after the correction step and so in theory Theorem 3.1 could be applied. In practice, however, this algorithm fails to converge because in general Assumption 3.5 does not hold. If ranged rows with large dual multipliers are ignored in the new objective function, it may contain coefficients of large absolute value and a floating-point solver will in general not be able to satisfy an absolute dual feasibility tolerance below one.

If they are considered, i.e., if we do replace the objective function by the scaled reduced cost vector, then as explained above we would need to fix the non-equality constraints with a nonzero dual multiplier to one of their sides. Within the simplex-based implementation described in Section 5 we had also experimented with such a scheme that fixes each ranged row with nonzero dual multiplier in the transformed problem, either to its left-hand side if finite and the multiplier is positive or to its right-hand side if finite and the multiplier is negative, and forces infeasible multipliers to zero. This ensures that replacing the objective function by the (scaled) reduced cost vector is an equivalent affine transformation for this heuristically restricted LP.

If the dual multiplier changes sign after the correction step, the fixing was relaxed and the multiplier reset to zero. If the transformed LP becomes infeasible, this is performed for all fixed rows.

For cases when the initial LP solve returned a basis that was almost optimal, the approach worked well. For more complicated cases, however, this strategy seemed to increase the number of LP solves, sometimes drastically, because too many unfixings were necessary.

B.3 Feasibility LP in general form

For an LP given in general form (13), the slightly more technical formulation of the feasibility LP (4) becomes

$$\max\{\tau \mid L - At - w \leqslant A\xi - w\tau \leqslant U - At - w, \ell - t \leqslant \xi \leqslant u - t, \tau \leqslant 1\}$$

with shift vectors $t = t(\ell, u) \in \mathbb{Q}^n$,

$$t(\ell,u)_i \coloneqq \begin{cases} \ell_i & \text{if } \ell_i > 0, \\ u_i & \text{if } u_i < 0, \\ 0 & \text{otherwise,} \end{cases}$$

and $w = w(L, U, \ell, u) \in \mathbb{Q}^m$,

$$w(L,U,\ell,u)_j \coloneqq \begin{cases} L_j - A_j.t(\ell,u) & \text{if } L_j - A_j.t(\ell,u) > 0, \\ U_j - A_j.t(\ell,u) & \text{if } U_j - A_j.t(\ell,u) < 0, \\ 0 & \text{otherwise,} \end{cases}$$

to ensure that the zero solution is primal feasible. In order to minimize the overhead in transforming the original LP into the this form, we do not shift variables that already contain zero within their bounds and we do not homogenize the bounds, since then these would become additional constraints involving the auxiliary variable τ.

This generalized feasibility LP cannot be derived simply by a twofold dualization as was the case for the standard form (4), but it can be verified that Points 1 to 3 of Lemma 4.1 still hold. Point 4, however, becomes slightly more technical. If (ξ^*, τ^*), $\tau^* \approx 1$, is an approximate optimal solution of the feasibility LP that violates primal feasibility by at most ϵ_P, then $x^* = \frac{1}{\tau^*}\xi^* + t$ is a feasible solution for the original LP within tolerance $M\frac{\max\{1-\tau^*,0\}}{\tau^*} + \epsilon_P/\tau^*$, where M is the maximum of $\|\ell - t\|_\infty$, $\|u - t\|_\infty$, $\|L - At - w\|_\infty$, and $\|U - At - w\|_\infty$. The first term goes to zero as τ^* goes to one. If τ is bounded away from one, we will not converge to a feasible solution and have to apply infeasibility detection.

While for (4) the only possible optimal values are zero and one, this is not the case for this general form of the feasibility LP. Consider, for instance, the system $\{x = 1+\epsilon, 0 \leqslant x \leqslant 1\}$, which is infeasible for $\epsilon > 0$. The corresponding feasibility LP is $\max\{\tau \mid \xi - (1+\epsilon)\tau = 0, 0 \leqslant x \leqslant 1\}$. Its optimal solution is $(\xi, \tau) = (1, 1/(1+\epsilon))$ and its objective value $1/(1+\epsilon)$ is arbitrarily close to one as ϵ approaches zero.

B.4 Infeasibility box in general form

Similar to the case in Section 4.2, an exact Farkas proof for an LP of general form (13) consists of a vector of dual multipliers $y = y^L - y^U \in \mathbb{Q}^m$ for the rows and a "reduced cost" vector $z = z^\ell - z^u \in \mathbb{Q}^n$ of multipliers for the bound constraints, all $y^L, y^U, z^\ell, z^u \geqslant 0$, such that

$$A^\mathsf{T} y + z = 0 \tag{17}$$

and

$$L^\mathsf{T} y^L - U^\mathsf{T} y^U + \ell^\mathsf{T} z^\ell - u^\mathsf{T} z^u > 0 \tag{18}$$

hold. An approximate Farkas proof may violate both these conditions. For instance, if some y_j is only slightly positive although constraint j has no left-hand side, i.e., $L_j = -\infty$, then the left-hand side of (18) is already at $-\infty$.

This may be corrected by setting such entries in y to zero. Also, the first equation can be enforced by adjusting z to be equal to $-A^\mathsf{T} y$. This, however, may increase the violation of (18) or create it in the first place. In particular, if some $\ell_i = -\infty$ or $u_i = \infty$, there is no guarantee that $z_i \leqslant 0$ and $z_i \geqslant 0$, respectively. Even if iterative refinement applied to the feasibility LP produces Farkas proofs with smaller and smaller violations, this does in general not suffice to obtain a reliable certificate of infeasibility.

As described in Section 4.2, we may consider the idea of an infeasibility box, a box around the origin within which no feasible solution exists. For a general form LP, the Farkas cut (see (7)) will take the form

$$(y^\mathsf{T} A)x \geqslant L^\mathsf{T} y^L - U^\mathsf{T} y^U. \tag{19}$$

Its right-hand side is finite if L, U are finite or inconsistent entries in y have been adjusted to zero if necessary. As before, if we can show that (19) is violated by all points x with $\ell \leqslant x \leqslant u$, then the LP is proven infeasible. The approach of Neumaier and Shcherbina (2004) to apply interval arithmetic to compute a lower bound on the left-hand side of (19) is especially likely to produce a certificate of infeasibility if reasonable upper and lower bounds are known on each component of x. However, as mentioned before, if some entries of ℓ or u are not finite, this approach may fail, even if the approximate Farkas proof is very accurate. Just as in Section 4.2 the radius of an infeasibility box R can be computed using the formula for $\rho(d, d_0)$ given by (8).

In order to compute and strengthen R some small modifications to Algorithm 3 should be made. First, we note that if all entries of ℓ and u are finite we may halt the algorithm and conclude proven infeasibility whenever R exceeds $max(\|\ell\|_\infty, \|u\|_\infty)$. With this possibility in mind, the aggregation and computation in lines 2–4 of Algorithm 3 is done row by row and if the intermediate computed value of R exceeds $max(\|\ell\|_\infty, \|u\|_\infty)$, the algorithm is terminated immediately, returing proven infeasibility.

The second change to the algorithm is that the loop starting on line 6 should include both upper and lower bounds. Similar to the observation made in (9) we have that for any index with $d_i > 0$, we may add $-d_i x_i \geqslant -d_i u_i$ to strengthen the Farkas cut under the following condition

$$\rho(d - d_i e_i, d_0 - d_i u_i) > \rho(d, d_0) \Leftrightarrow u_i < \rho(d, d_0). \tag{20}$$

The loop starting on line 6 will then iterate through both upper and lower bounds, considering them individually and independently, as sorted such that the values ℓ_i and the negations of the corresponding upper bounds u_j appear in nonincreasing order. Similar to the check on line 7 we may terminate the algorithm when all the remaining lower bounds satisfy $-\ell_i \geqslant R$ and remaining upper bounds satisfy $u_i \geqslant R$. Moreover, an additional check can be added to this loop terminating with proven infeasibility if R is ever increased to exceed $max(\|\ell\|_\infty, \|u\|_\infty)$.

Remark B.3 (Infeasibility box with slack variables)**.** Instead of skipping ranged rows with infeasible dual multipliers in lines 2–4, we could (implicitly or explicitly) introduce slack variables. Then Algorithm 3 would effectively compute a radius R such that $\|x\|_\infty \geqslant R$ *and* $\|Ax\|_\infty \geqslant R$ must hold for any feasible solution. We decided against this in our implementation because a certificate on the bounds of the variables alone is easier to interpret by the user.

B.5 Unboundedness certificate in general form

As previously discussed, a feasible primal solution together with an unbounded direction of improving objective function value in the recession cone can serve as a certificate for unboundness of an LP. For an LP in the general form given by (13) a primal feasible solution can be constructed by solving the LP described in B.3 and a cost improving vector in the recession cone can be found by applying the feasibility test to the following system

$$\begin{aligned} A_{j\cdot} v &\leqslant 0 \text{ for all } j \text{ with } U_j < \infty, \\ A_{j\cdot} v &\geqslant 0 \text{ for all } j \text{ with } L_j > -\infty, \\ v &\leqslant 0 \text{ for all } i \text{ with } u_i < \infty, \\ v &\geqslant 0 \text{ for all } i \text{ with } \ell_i > -\infty, \\ c^\mathsf{T} v &= -1. \end{aligned} \tag{21}$$

Note that this is analogous to (10) in Section 4.3.

C Experimental Data and Results

This appendix comprises detailed, instance-wise data from our computational results. LP statistics are found in Table 4 and computational results in Tables 5 and 6.

C.1 LP Test Suite

To compile our test suite, we collected a large set of publicly available instances from the following sources:

- the NETLIB LP test set including the "kennington" folder,[10]
- Hans Mittelmann's benchmark instances,[11]
- Csaba Mészáros's LP collection,[12]
- the LP relaxations of the COR@L mixed-integer linear programming test set,[13] and
- the LP relaxations of the mixed-integer linear programs from the five versions of the MIPLIB.[14]

Some instances appeared in several collections. We removed all obvious duplicates and selected the 1,242 primal and dual feasible linear programs. Furthermore, we had to replace blank characters in the column and row names of some MPS files because they could not be parsed by the solver QSOPT_EX used in the second experiment.

We had to exclude seven large-scale instances instances[15] having between 4,366,648 and 183,263,061 nonzeros in the constraint matrix for which the SOPLEX LP solver used in our experiments hit the memory limit of 48 GB when parsing the instance using the exact rational data type. Furthermore, we removed the 33 instances[16] which even standard, floating-point SOPLEX could not solve within a time limit of two hours.

This left us with altogether 1,202 instances. The number of columns ranges from 3 to 2,277,736, the number of rows from 1 to 656,900, and the constraint matrices contain between 6 and 27,678,735 nonzero entries. Table 4 gives basic statistics for the LP test set used.

[10] University of Tennessee Knoxville and Oak Ridge National Laboratory. NETLIB LP Library. `http://www.netlib.org/lp/`, accessed September 2014.

[11] Hans Mittelmann. LP Test Set. `http://plato.asu.edu/ftp/lptestset/`, accessed July 16, 2014.

[12] Csaba Mészáros. LP Test Set. `http://www.sztaki.hu/~meszaros/public_ftp/lptestset/`, accessed July 16, 2014.

[13] Computational Optimization Research At Lehigh. MIP Instances. `http://coral.ie.lehigh.edu/data-sets/mixed-integer-instances/`, accessed June 6, 2011.

[14] Zuse Institute Berlin. MIPLIB—Mixed Integer Problem Library. `http://miplib.zib.de/`, accessed July 16, 2014.

[15] `cont1_l`, `hawaiiv10-130`, `netlarge1,6`, `pb-simp-nonunif`, and `zib01,02`

[16] `L1_d10_40`, `Linf_520c`, `bley_xl1`, `cdma`, `cont11`, `cont11_l`, `datt256`, `dbic1`, `degme`, `in`, `karted`, `mining`, `nb10tb`, `neos3`, `netlarge3`, `ns1687037`, `ns1688926`, `ns1853823`, `ns1854840`, `nug15`, `nug20`, `nug30`, `rail02`, `rail03`, `rmine21`, `rmine25`, `sing161`, `spal_004`, `splan1`, `stat96v2`, `stat96v3`, `tp-6`, and `ts-palko`

Table 4: Statistics on 1,202 LP test instances. Column "sparsity" reports the number of columns times the number of rows divided by the number of nonzeros in the constraint matrix. The last two columns refer to the absolute values of the nonzeros in the constraint matrix.

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
10teams	230	2025	12150	38	1	1
16_n14	16384	262144	524288	8.2e+03	1	1
22433	198	429	3408	28	1	1.1e+03
23588	137	368	3701	14	1	1.1e+03
25fv47	821	1571	10400	1.4e+02	0.0002	2.4e+02
30_70_45_095_100	12526	10976	46640	3.1e+03	1	1
30n20b8	576	18380	109706	1.2e+02	1	2.2e+02
50v-10	233	2013	2745	2.3e+02	1	2.2e+02
80bau3b	2262	9799	21002	1.1e+03	0.00022	1e+02
Test3	50680	72215	617906	6.3e+03	0.0001	1.8e+07
a1c1s1	3312	3648	10178	1.7e+03	1	4.8e+02
aa01	823	8904	72965	1e+02	1	1
aa03	825	8627	70806	1e+02	1	1
aa3	825	8627	70806	1e+02	1	1
aa4	426	7195	52121	61	1	1
aa5	801	8308	65953	1.1e+02	1	1
aa6	646	7292	51728	92	1	1
acc-tight4	3285	1620	17073	3.3e+02	1	1
acc-tight5	3052	1339	16134	2.5e+02	1	1
acc-tight6	3047	1335	16108	2.5e+02	1	1
adlittle	56	97	383	19	0.0012	64
afiro	27	32	83	14	0.11	2.4
aflow30a	479	842	2091	2.4e+02	1	1e+02
aflow40b	1442	2728	6783	7.2e+02	1	1e+02
agg	488	163	2410	35	2e-05	4.2e+02
agg2	516	302	4284	37	2e-05	4.2e+02
agg3	516	302	4300	37	2e-05	4.2e+02
air02	50	6774	61555	5.6	1	1
air03	124	10757	91028	16	1	1
air04	823	8904	72965	1e+02	1	1
air05	426	7195	52121	61	1	1
air06	825	8627	70806	1e+02	1	1
aircraft	3754	7517	20267	1.9e+03	1	81
aligninq	340	1831	15734	42	1	3.0e+02
app1-2	53467	26871	199175	7.6e+03	1e-05	1
arki001	1048	1388	20439	75	0.0002	2.3e+07
ash608gpia-3col	24748	3651	74244	1.2e+03	1	1
atlanta-ip	21732	48738	257532	4.3e+03	0.028	64
atm20-100	4380	6480	58878	4.9e+02	0.1	1.3e+04
b2c1s1	3904	3872	11408	2e+03	0.2	4.8e+02
bab1	60680	61152	854392	4.7e+03	0.05	4
bab3	23069	393800	3301838	2.9e+03	0.09	8
bab5	4964	21600	155520	7.1e+02	0.09	8
bal8x12	116	192	384	58	1	35
bandm	305	472	2494	61	0.001	2e+02
bas1lp	5411	4461	582411	42	1	14
baxter	27441	15128	95971	4.6e+03	0.001	3.2e+05
bc	1913	1751	276842	12	1.1e-13	10
bc1	1913	1751	276842	12	1.1e-13	10
beaconfd	173	262	3375	14	0.0012	5e+02
beasleyC3	1750	2500	5000	8.7e+02	1	82
bell3a	123	133	347	61	8.3e-05	1.3e+03

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
bell5	91	104	266	46	8.3e-05	1.3e+03
berlin_5_8_0	1532	1083	4507	3.8e+02	1	2.4e+02
bg512142	1307	792	3953	3.3e+02	1	5.6e+03
biella1	1203	7328	71489	1.3e+02	1	1e+08
bienst1	576	505	2184	1.4e+02	1	81
bienst2	576	505	2184	1.4e+02	1	81
binkar10_1	1026	2298	4496	1e+03	1	29
bk4x3	19	24	48	9.5	1	40
blend	74	83	491	15	0.003	66
blend2	274	353	1409	91	1	7.2e+03
blp-ar98	1128	16021	200601	94	1	1e+03
blp-ic97	923	9845	118149	77	1	9.6e+02
bnatt350	4923	3150	19061	8.2e+02	0.12	1
bnatt400	5614	3600	21698	9.4e+02	0.12	1
bnl1	643	1175	5121	1.6e+02	0.0011	78
bnl2	2324	3489	13999	5.8e+02	0.0006	78
boeing1	351	384	3485	39	0.011	3.1e+03
boeing2	166	143	1196	21	0.01	3e+03
bore3d	233	315	1429	58	0.0001	1.4e+03
brandy	220	249	2148	28	0.0008	2e+02
buildingenergy	277594	154978	788969	5.6e+04	0.05	64
cap6000	2176	6000	48243	2.7e+02	1	9.9e+04
capri	271	353	1767	54	9e-05	2.2e+02
car4	16384	33052	63724	1.6e+04	0.00098	1
cari	400	1200	152800	3.1	7.8e-05	1
cep1	1521	3248	6712	7.6e+02	0.01	1
ch	3700	5062	20873	9.2e+02	4.3e-05	2.6e+02
circ10-3	42620	2700	307320	3.8e+02	1	16
co-100	2187	48417	1995817	53	1	2e+04
co5	5774	7993	53661	9.6e+02	1e-05	2.7e+03
co9	10789	14851	101578	1.8e+03	1e-05	2.7e+03
complex	1023	1408	46463	32	1	1
cont1	160792	40398	399990	1.8e+04	0.005	4
cont4	160792	40398	398398	1.8e+04	0.005	4
core2536-691	2539	15293	177739	2.3e+02	0.1	1e+02
core4872-1529	4875	24656	218762	6.1e+02	0.1	1e+02
cov1075	637	120	14280	5.4	1	1
cq5	5048	7530	47353	8.4e+02	1.6e-05	1e+03
cq9	9278	13778	88897	1.5e+03	1.6e-05	1e+03
cr42	905	1513	6614	2.3e+02	0.012	40
cre-a	3516	4067	14987	1.2e+03	0.6	71
cre-b	9648	72447	256095	3.2e+03	0.6	71
cre-c	3068	3678	13244	1e+03	0.5	71
cre-d	8926	69980	242646	3e+03	0.5	71
crew1	135	6469	46950	19	1	1
csched007	351	1758	6379	1.2e+02	1	1.9e+02
csched008	351	1536	5687	1.2e+02	1	1.8e+02
csched010	351	1758	6376	1.2e+02	1	1.8e+02
cycle	1903	2857	20720	2.7e+02	1e-05	9.1e+02
czprob	929	3523	10669	3.1e+02	0.0016	1.4e+02
d10200	947	2000	57637	34	1	1.4e+02
d20200	1502	4000	189389	32	1	2.8e+02
d2q06c	2171	5167	32417	3.6e+02	0.0002	2.3e+03
d6cube	415	6184	37704	69	1	3.6e+02
dano3_3	3202	13873	79655	6.4e+02	0.5	1e+03
dano3_4	3202	13873	79655	6.4e+02	0.5	1e+03
dano3_5	3202	13873	79655	6.4e+02	0.5	1e+03

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
dano3mip	3202	13873	79655	6.4e+02	0.5	1e+03
danoint	664	521	3232	1.1e+02	0.5	66
dbir1	18804	27355	1058605	4.9e+02	1	1.5e+05
dbir2	18906	27355	1139637	4.6e+02	1	1.1e+05
dc1c	1649	10039	121158	1.4e+02	1	1e+07
dc1l	1653	37297	448754	1.4e+02	1	1e+02
dcmulti	290	548	1315	1.4e+02	1	6e+02
de063155	852	1488	4553	2.8e+02	2.1e-07	8.4e+11
de063157	936	1488	4699	3.1e+02	1.3e-09	2.3e+18
de080285	936	1488	4662	3.1e+02	1.6e-17	9.7e+02
degen2	444	534	3978	63	1	1
degen3	1503	1818	24646	1.2e+02	1	1
delf000	3128	5464	12606	1.6e+03	1e-06	1.9e+03
delf001	3098	5462	13214	1.5e+03	1e-06	1.9e+03
delf002	3135	5460	13287	1.6e+03	1e-06	1.9e+03
delf003	3065	5460	13269	1.5e+03	1e-06	1.9e+03
delf004	3142	5464	13546	1.6e+03	1e-06	1.9e+03
delf005	3103	5464	13494	1.6e+03	1e-06	1.9e+03
delf006	3147	5469	13604	1.6e+03	1e-06	1.9e+03
delf007	3137	5471	13758	1.6e+03	1e-06	1.9e+03
delf008	3148	5472	13821	1.6e+03	1e-06	1.9e+03
delf009	3135	5472	13750	1.6e+03	1e-06	1.9e+03
delf010	3147	5472	13802	1.6e+03	1e-06	1.9e+03
delf011	3134	5471	13777	1.6e+03	1e-06	1.9e+03
delf012	3151	5471	13793	1.6e+03	1e-06	1.9e+03
delf013	3116	5472	13809	1.6e+03	1e-06	1.9e+03
delf014	3170	5472	13866	1.6e+03	1e-06	1.9e+03
delf015	3161	5471	13793	1.6e+03	1e-06	1.9e+03
delf017	3176	5471	13732	1.6e+03	1e-06	1.9e+03
delf018	3196	5471	13774	1.6e+03	1e-06	1.9e+03
delf019	3185	5471	13762	1.6e+03	1e-06	1.9e+03
delf020	3213	5472	14070	1.6e+03	1e-06	1.9e+03
delf021	3208	5471	14068	1.6e+03	1e-06	1.9e+03
delf022	3214	5472	14060	1.6e+03	1e-06	1.9e+03
delf023	3214	5472	14098	1.6e+03	1e-06	1.9e+03
delf024	3207	5466	14456	1.6e+03	1e-06	1.9e+03
delf025	3197	5464	14447	1.6e+03	1e-06	1.9e+03
delf026	3190	5462	14220	1.6e+03	1e-06	1.9e+03
delf027	3187	5457	14200	1.6e+03	1e-06	1.9e+03
delf028	3177	5452	14402	1.6e+03	1e-06	1.9e+03
delf029	3179	5454	14402	1.6e+03	1e-06	1.9e+03
delf030	3199	5469	14262	1.6e+03	1e-06	1.9e+03
delf031	3176	5455	14205	1.6e+03	1e-06	1.9e+03
delf032	3196	5467	14251	1.6e+03	1e-06	1.9e+03
delf033	3173	5456	14205	1.6e+03	1e-06	1.9e+03
delf034	3175	5455	14208	1.6e+03	1e-06	1.9e+03
delf035	3193	5468	14284	1.6e+03	1e-06	1.9e+03
delf036	3170	5459	14202	1.6e+03	1e-06	1.9e+03
deter0	1923	5468	11173	9.6e+02	0.72	1.4
deter1	5527	15737	32187	2.8e+03	0.55	1.5
deter2	6095	17313	35731	3e+03	0.62	1.5
deter3	7647	21777	44547	3.8e+03	0.55	1.5
deter4	3235	9133	19231	1.6e+03	0.61	1.5
deter5	5103	14529	29715	2.6e+03	0.41	1.4
deter6	4255	12113	24771	2.1e+03	0.6	1.5
deter7	6375	18153	37131	3.2e+03	0.58	1.4
deter8	3831	10905	22299	1.9e+03	0.67	1.4

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
df2177	630	9728	21706	3.2e+02	1	1
dfl001	6071	12230	35632	3e+03	0.083	2
dfn-gwin-UUM	158	938	2632	79	1	6.2e+02
dg012142	6310	2080	14795	9e+02	1	7.2e+03
disctom	399	10000	30000	1.3e+02	1	1
disp3	2182	1856	6407	7.3e+02	0.016	4.5
dolom1	1803	11612	190413	1.1e+02	1	1e+08
ds	656	67732	1024059	44	1	1
ds-big	1042	174997	4623442	40	1	1
dsbmip	1182	1886	7366	3.9e+02	0.062	3.6e+04
e18	24617	14231	132095	2.7e+03	1	2e+03
e226	223	282	2578	25	0.00026	1.5e+03
egout	98	141	282	49	1	1.2e+02
eil33-2	32	4516	44243	3.6	1	1
eilA101-2	100	65832	959373	7.1	1	1
eilB101	100	2818	24120	12	1	1
enigma	21	100	289	10	1	9e+05
enlight13	169	338	962	84	1	2
enlight14	196	392	1120	98	1	2
enlight15	225	450	1290	1.1e+02	1	2
enlight16	256	512	1472	1.3e+02	1	2
enlight9	81	162	450	40	1	2
etamacro	400	688	2409	1.3e+02	0.019	2e+03
ex10	69608	17680	1162000	1.1e+03	1	1
ex1010-pi	1468	25200	102114	3.7e+02	1	1
ex3sta1	17443	8156	59419	2.5e+03	0.29	1e+02
ex9	40962	10404	517112	8.4e+02	1	1
f2000	10500	4000	29500	1.5e+03	1	1
farm	7	12	36	2.3	1	2.5e+02
fast0507	507	63009	409349	84	1	1
fffff800	524	854	6227	75	0.008	1.1e+05
fiball	3707	34219	104792	1.2e+03	0.6	94
fiber	363	1298	2944	1.8e+02	1	2.2e+02
finnis	497	614	2310	1.7e+02	0.00046	32
fit1d	24	1026	13404	1.8	0.01	1.9e+03
fit1p	627	1677	9868	1.3e+02	0.01	1.9e+03
fit2d	25	10500	129018	2.1	0.05	2.6e+03
fit2p	3000	13525	50284	1e+03	0.05	2.6e+03
fixnet6	478	878	1756	2.4e+02	1	5e+02
flugpl	18	18	46	9	0.9	1.5e+02
fome11	12142	24460	71264	6.1e+03	0.083	2
fome12	24284	48920	142528	1.2e+04	0.083	2
fome13	48568	97840	285056	2.4e+04	0.083	2
fome20	33874	105728	230200	1.7e+04	1	1
fome21	67748	211456	460400	3.4e+04	1	1
forplan	161	421	4563	16	0.0074	2.8e+03
fxm2-16	3900	5602	31239	7.8e+02	0.0005	1.3e+02
fxm2-6	1520	2172	12139	3e+02	0.0005	1.3e+02
fxm3_16	41340	64162	370839	8.3e+03	0.0005	1.3e+02
fxm3_6	6200	9492	54589	1.2e+03	0.0005	1.3e+02
fxm4_6	22400	30732	248989	2.8e+03	0.0005	1.3e+02
g200x740i	940	1480	2960	4.7e+02	1	2e+02
gams10a	114	61	297	28	0.38	1
gams30a	354	181	937	71	0.14	1
ganges	1309	1681	6912	3.3e+02	0.0014	1
ge	10099	11098	39554	3.4e+03	4.5e-06	1.4e+04
gen	780	870	2592	3.9e+02	1	6.7e+02

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
gen1	769	2560	63085	32	2e-07	1
gen2	1121	3264	81855	45	2e-09	1
gen4	1537	4297	107102	64	5.1e-08	1
ger50_17_trans	499	22414	172035	71	1	3.2e+04
germanrr	10779	10813	175547	6.7e+02	1	8.2e+05
germany50-DBM	2526	8189	24479	1.3e+03	1	40
gesa2	1392	1224	5064	3.5e+02	0.069	1.2e+02
gesa2-o	1248	1224	3672	4.2e+02	0.069	1.2e+02
gesa2_o	1248	1224	3672	4.2e+02	0.069	1.2e+02
gesa3	1368	1152	4944	3.4e+02	0.069	1.2e+02
gesa3_o	1224	1152	3624	4.1e+02	0.069	1.2e+02
gfrd-pnc	616	1092	2377	3.1e+02	1	1.1e+03
glass4	396	322	1815	79	1	8.4e+06
gmu-35-40	424	1205	4843	1.1e+02	0.8	2.6e+03
gmu-35-50	435	1919	8643	1.1e+02	0.8	2.6e+03
gmut-75-50	2565	68865	571475	3.2e+02	0.95	7.7e+03
gmut-77-40	2554	24338	159902	4.3e+02	0.95	6.1e+03
go19	441	441	1885	1.1e+02	1	1
gr4x6	34	48	96	17	1	35
greenbea	2392	5405	30877	4.8e+02	6e-05	1e+02
greenbeb	2392	5405	30877	4.8e+02	6e-05	1e+02
grow15	300	645	5620	38	6e-06	1
grow22	440	946	8252	55	6e-06	1
grow7	140	301	2612	18	6e-06	1
gt2	29	188	376	14	1	2.5e+03
hanoi5	16399	3862	39718	1.6e+03	1	1
haprp	1048	1828	3628	1e+03	1	1.8e+04
harp2	112	2993	5840	1.1e+02	1	4.2e+09
i_n13	8192	741455	1482910	4.1e+03	1	1
ic97_potential	1046	728	3138	2.6e+02	1	60
iiasa	669	2970	6648	3.3e+02	0.51	8.8e+03
iis-100-0-cov	3831	100	22986	17	1	1
iis-bupa-cov	4803	345	38392	43	1	1
iis-pima-cov	7201	768	71941	77	1	1
israel	174	142	2269	12	0.001	1.6e+03
ivu06-big	1177	2277736	23125770	1.2e+02	1	1
ivu52	2116	157591	2179476	1.6e+02	0.0027	9
janos-us-DDM	760	2184	6384	3.8e+02	1	64
jendrec1	2109	4228	89608	1e+02	0.00018	1.5e+02
k16x240	256	480	960	1.3e+02	1	1e+03
kb2	43	41	286	7.2	0.17	1.1e+02
ken-07	2426	3602	8404	1.2e+03	1	1
ken-11	14694	21349	49058	7.3e+03	1	1
ken-13	28632	42659	97246	1.4e+04	1	1
ken-18	105127	154699	358171	5.3e+04	1	1
kent	31300	16620	184710	2.8e+03	0.03	1.4e+03
khb05250	101	1350	2700	50	1	5e+03
kl02	71	36699	212536	14	1	1
kleemin3	3	3	6	1.5	1	2e+02
kleemin4	4	4	10	2	1	2e+03
kleemin5	5	5	15	1.7	1	2e+04
kleemin6	6	6	21	2	1	2e+05
kleemin7	7	7	28	1.8	1	2e+06
kleemin8	8	8	36	2	1	2e+07
l152lav	97	1989	9922	24	1	43
l30	2701	15380	51169	9e+02	0.017	1.8
l9	244	1401	4577	81	0.056	1.8

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
large000	4239	6833	16573	2.1e+03	1e-06	1.9e+03
large001	4162	6834	17225	2.1e+03	1e-06	1.9e+03
large002	4249	6835	18330	2.1e+03	1e-06	1.9e+03
large003	4200	6835	18016	2.1e+03	1e-06	1.9e+03
large004	4250	6836	17739	2.1e+03	1e-06	1.9e+03
large005	4237	6837	17575	2.1e+03	1e-06	1.9e+03
large006	4249	6837	17887	2.1e+03	1e-06	1.9e+03
large007	4236	6836	17856	2.1e+03	1e-06	1.9e+03
large008	4248	6837	17898	2.1e+03	1e-06	1.9e+03
large009	4237	6837	17878	2.1e+03	1e-06	1.9e+03
large010	4247	6837	17887	2.1e+03	1e-06	1.9e+03
large011	4236	6837	17878	2.1e+03	1e-06	1.9e+03
large012	4253	6838	17919	2.1e+03	1e-06	1.9e+03
large013	4248	6838	17941	2.1e+03	1e-06	1.9e+03
large014	4271	6838	17979	2.1e+03	1e-06	1.9e+03
large015	4265	6838	17957	2.1e+03	1e-06	1.9e+03
large016	4287	6838	18029	2.1e+03	1e-06	1.9e+03
large017	4277	6837	17983	2.1e+03	1e-06	1.9e+03
large018	4297	6837	17791	2.1e+03	1e-06	1.9e+03
large019	4300	6836	17786	2.2e+03	1e-06	1.9e+03
large020	4315	6837	18136	2.2e+03	1e-06	1.9e+03
large021	4311	6838	18157	2.2e+03	1e-06	1.9e+03
large022	4312	6834	18104	2.2e+03	1e-06	1.9e+03
large023	4302	6835	18123	2.2e+03	1e-06	1.9e+03
large024	4292	6831	18599	2.1e+03	1e-06	1.9e+03
large025	4297	6832	18743	2.1e+03	1e-06	1.9e+03
large026	4284	6824	18631	2.1e+03	1e-06	1.9e+03
large027	4275	6821	18562	2.1e+03	1e-06	1.9e+03
large028	4302	6833	18886	2.2e+03	1e-06	1.9e+03
large029	4301	6832	18952	2.2e+03	1e-06	1.9e+03
large030	4285	6823	18843	2.1e+03	1e-06	1.9e+03
large031	4294	6826	18867	2.1e+03	1e-06	1.9e+03
large032	4292	6827	18850	2.1e+03	1e-06	1.9e+03
large033	4273	6817	18791	2.1e+03	1e-06	1.9e+03
large034	4294	6831	18855	2.1e+03	1e-06	1.9e+03
large035	4293	6829	18881	2.1e+03	1e-06	1.9e+03
large036	4282	6822	18840	2.1e+03	1e-06	1.9e+03
lectsched-1	50108	28718	310792	5e+03	1	1.3e+03
lectsched-1-obj	50108	28718	310792	5e+03	1	1.3e+03
lectsched-2	30738	17656	186520	3.1e+03	1	1.3e+03
lectsched-3	45262	25776	279967	4.5e+03	1	1.3e+03
lectsched-4-obj	14163	7901	82428	1.4e+03	1	1.3e+03
leo1	593	6731	131218	31	1	9e+07
leo2	593	11100	219959	31	1	1.7e+08
liu	2178	1156	10626	2.4e+02	1	8.4e+03
lo10	46341	406225	812450	2.3e+04	1	1
long15	32769	753687	1507374	1.6e+04	1	1
lotfi	153	308	1078	51	0.019	1e+03
lotsize	1920	2985	6565	9.6e+02	1	2e+04
lp22	2958	13434	65560	7.4e+02	1	1
lpl1	39951	125000	381259	1.3e+04	1	5.4e+02
lpl2	3294	10755	32106	1.6e+03	1	5e+02
lpl3	10828	33538	100377	5.4e+03	1	5e+02
lrn	8491	7253	46123	1.4e+03	0.00098	6.2e+07
lrsa120	14521	3839	39956	1.5e+03	1	2
lseu	28	89	309	9.3	1	5.2e+02
m100n500k4r1	100	500	2000	25	1	1

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
macrophage	3164	2260	9492	7.9e+02	1	1
manna81	6480	3321	12960	2.2e+03	1	1
map06	328818	164547	549920	1.1e+05	1	1.1e+07
map10	328818	164547	549920	1.1e+05	1	1.1e+07
map14	328818	164547	549920	1.1e+05	1	1.1e+07
map18	328818	164547	549920	1.1e+05	1	1.1e+07
map20	328818	164547	549920	1.1e+05	1	1.1e+07
markshare1	6	62	312	1.2	1	99
markshare2	7	74	434	1.4	1	99
markshare_5_0	5	45	203	1.2	1	1e+02
maros	846	1443	9614	1.4e+02	0.0001	1.7e+04
maros-r7	3136	9408	144848	2.1e+02	0.0017	1
mas74	13	151	1706	1.2	1	9.9e+03
mas76	12	151	1640	1.2	1	9.9e+03
maxgasflow	7160	7437	19717	3.6e+03	1	1e+04
mc11	1920	3040	6080	9.6e+02	1	2.1e+02
mcf2	664	521	3232	1.1e+02	0.5	66
mcsched	2107	1747	8088	5.3e+02	1	1e+03
methanosarcina	14604	7930	43812	2.9e+03	1	1
mik-250-1-100-1	151	251	5351	7.2	1	2e+03
mine-166-5	8429	830	19412	3.7e+02	1	2.5e+04
mine-90-10	6270	900	15407	3.7e+02	1	4.7e+04
misc03	96	160	2053	8	1	9.6e+02
misc06	820	1808	5859	2.7e+02	0.12	2.6e+02
misc07	212	260	8619	6.4	1	7e+02
mitre	2054	10724	39704	6.8e+02	1	1.1e+03
mkc	3411	5325	17038	1.1e+03	1	38
mkc1	3411	5325	17038	1.1e+03	1	38
mod008	6	319	1243	2	0.5	7.6
mod010	146	2655	11203	36	1	20
mod011	4480	10958	22254	2.2e+03	0.14	4.2e+04
mod2	34774	31728	165129	7e+03	0.0019	5.6e+03
model1	362	798	3028	1.2e+02	0.001	3
model10	4400	15447	149000	4.9e+02	2.2e-05	1.9e+04
model11	7056	18288	55859	2.4e+03	2e-05	2e+02
model2	379	1212	7498	63	0.0001	4e+02
model3	1609	3840	23236	2.7e+02	1e-05	6.9e+03
model4	1337	4549	45340	1.5e+02	2.2e-05	3.1e+03
model5	1888	11360	89483	2.7e+02	1e-05	1.7e+02
model6	2096	5001	27340	4.2e+02	8e-05	1.1e+03
model7	3358	8007	49452	5.6e+02	1e-05	7.3e+03
model8	2896	6464	25277	9.7e+02	0.001	3
model9	2879	10257	55274	5.8e+02	0.0001	1e+03
modglob	291	422	968	1.5e+02	0.9	1.7e+04
modszk1	687	1620	3168	6.9e+02	0.00074	1.2
momentum1	42680	5174	103198	2.2e+03	2.5e-10	10
momentum2	24237	3732	349695	2.6e+02	3.3e-21	20
momentum3	56822	13532	949495	8.1e+02	2.1e-19	20
msc98-ip	15850	21143	92918	4e+03	0.0078	2.3e+06
mspp16	561657	29280	27678735	5.9e+02	1	1.5e+03
multi	61	102	961	6.8	0.0001	10
mzzv11	9499	10240	134603	7.3e+02	1	2e+03
mzzv42z	10460	11717	151261	8.7e+02	1	1.1e+03
n15-3	29494	153140	611000	9.8e+03	1	2.6e+02
n3-3	2425	9028	35380	8.1e+02	1	2.6e+02
n3700	5150	10000	20000	2.6e+03	1	3.8e+03
n3701	5150	10000	20000	2.6e+03	1	2.1e+03

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
n3702	5150	10000	20000	2.6e+03	1	2.5e+03
n3703	5150	10000	20000	2.6e+03	1	2.2e+03
n3704	5150	10000	20000	2.6e+03	1	2.7e+03
n3705	5150	10000	20000	2.6e+03	1	3.1e+03
n3706	5150	10000	20000	2.6e+03	1	2.1e+03
n3707	5150	10000	20000	2.6e+03	1	2.3e+03
n3708	5150	10000	20000	2.6e+03	1	2.9e+03
n3709	5150	10000	20000	2.6e+03	1	2.5e+03
n370a	5150	10000	20000	2.6e+03	1	2.2e+03
n370b	5150	10000	20000	2.6e+03	1	2.4e+03
n370c	5150	10000	20000	2.6e+03	1	2.3e+03
n370d	5150	10000	20000	2.6e+03	1	2.3e+03
n370e	5150	10000	20000	2.6e+03	1	2.5e+03
n3div36	4484	22120	340740	3e+02	1	24
n3seq24	6044	119856	3232340	2.3e+02	1	24
n4-3	1236	3596	14036	4.1e+02	1	2.6e+02
n9-3	2364	7644	30072	7.9e+02	1	2.6e+02
nag	5840	2884	26499	6.5e+02	1	1e+04
nemsafm	334	2252	2730	3.3e+02	0.99	1
nemscem	651	1570	3698	3.3e+02	0.18	1
nemsemm1	3945	71413	1050047	2.8e+02	1e-05	4.5e+03
nemsemm2	6943	42133	175267	1.7e+03	1e-07	5e+03
nemspmm1	2372	8622	55586	4e+02	0.001	8.8e+03
nemspmm2	2301	8413	67904	2.9e+02	0.00097	8.8e+03
nemswrld	7138	27174	190907	1e+03	0.00055	1.6e+02
neos	479119	36786	1047675	1.7e+04	0.5	1
neos-1053234	2596	5621	14920	1.3e+03	0.0099	9.8e+05
neos-1053591	1263	1386	3543	6.3e+02	1e-05	1e+05
neos-1056905	900	463	3510	1.3e+02	1	1e+02
neos-1058477	1529	2805	9376	5.1e+02	0.0099	1.2e+06
neos-1061020	10618	14010	114508	1.3e+03	1	1
neos-1062641	1677	1748	4544	8.4e+02	0.0028	1e+05
neos-1067731	3423	8779	30998	1.1e+03	1	1
neos-1096528	550339	1520	2171928	3.9e+02	1	2
neos-1109824	28979	1520	89528	5e+02	1	1
neos-1112782	2115	4140	8145	2.1e+03	1	2.9e+07
neos-1112787	1680	3280	6440	1.7e+03	1	2.5e+07
neos-1120495	21739	1140	67146	3.7e+02	1	1
neos-1121679	6	62	312	1.2	1	99
neos-1122047	57791	5100	163640	1.8e+03	1	1e+04
neos-1126860	36709	2565	105219	9e+02	1	1e+04
neos-1140050	3795	40320	808080	1.9e+02	5.6e-07	2.9e+04
neos-1151496	982	1549	27817	58	1	1
neos-1171448	13206	4914	131859	5.1e+02	1	1.2e+03
neos-1171692	4239	1638	42945	1.6e+02	1	3.2e+03
neos-1171737	4179	2340	58620	1.7e+02	1	1.6e+03
neos-1173026	893	1314	6933	1.8e+02	0.0099	2.7e+06
neos-1200887	633	234	6084	24	1	6.4e+03
neos-1208069	1150	2322	27242	1e+02	0.2	1
neos-1208135	1040	2322	24034	1e+02	0.25	1
neos-1211578	356	260	1540	71	1	8.2e+03
neos-1215259	1236	1601	38435	52	0.2	1
neos-1215891	6068	5035	44590	7.6e+02	0.25	7
neos-1223462	5890	5495	47040	7.4e+02	0.33	7
neos-1224597	3276	3395	25090	4.7e+02	0.5	7
neos-1225589	675	1300	2525	6.8e+02	1	1.8e+06
neos-1228986	356	260	1540	71	1	1.2e+04

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
neos-1281048	522	739	8808	47	1	2
neos-1311124	1643	1092	7140	2.7e+02	1	4.1e+03
neos-1324574	5904	5256	20880	2e+03	1	1
neos-1330346	4248	2664	13032	1.1e+03	1	1
neos-1330635	2717	1736	8260	6.8e+02	1	1e+06
neos-1337307	5687	2840	30799	5.7e+02	1	1.6e+04
neos-1337489	356	260	1540	71	1	8.2e+03
neos-1346382	796	520	3400	1.3e+02	1	8.2e+03
neos-1354092	3135	13702	187187	2.4e+02	1	1
neos-1367061	102750	36600	260250	1.5e+04	1	8e+03
neos-1396125	1494	1161	5511	3.7e+02	1	2.5e+02
neos-1407044	6908	16604	206633	5.8e+02	1	1
neos-1413153	2500	2451	9653	8.3e+02	1	4.2e+02
neos-1415183	2809	2757	10868	9.4e+02	1	5e+02
neos-1417043	3284	573315	1146630	1.6e+03	1	1
neos-1420205	383	231	1050	96	1	1e+06
neos-1420546	12671	26055	67959	6.3e+03	0.00072	7.9
neos-1420790	2310	4926	12720	1.2e+03	0.00067	8
neos-1423785	25721	21506	64082	1.3e+04	1	4.1
neos-1425699	89	105	430	22	1	2e+05
neos-1426635	796	520	3400	1.3e+02	1	8.2e+03
neos-1426662	1914	832	8048	2.1e+02	1	1e+03
neos-1427181	1786	832	7792	2e+02	1	2e+03
neos-1427261	2226	1040	9740	2.5e+02	1	2e+03
neos-1429185	1346	624	5844	1.5e+02	1	2e+03
neos-1429212	58726	416040	1855220	1.5e+04	0.067	27
neos-1429461	1096	520	4780	1.2e+02	1	3.1e+03
neos-1430701	668	312	2868	74	1	4.1e+03
neos-1430811	73661	519704	2474280	1.8e+04	0.01	27
neos-1436709	1417	676	6214	1.6e+02	1	3.1e+03
neos-1436713	2666	1248	11688	3e+02	1	2e+03
neos-1437164	187	2256	9016	62	1	1e+04
neos-1439395	775	364	3346	86	1	8.2e+03
neos-1440225	330	1285	14168	30	1	1
neos-1440447	561	260	2390	62	1	6.1e+03
neos-1440457	1952	936	8604	2.2e+02	1	3.1e+03
neos-1440460	989	468	4302	1.1e+02	1	6.1e+03
neos-1441553	316	960	11138	29	1	1e+04
neos-1442119	1524	728	6692	1.7e+02	1	4.1e+03
neos-1442657	1310	624	5736	1.5e+02	1	4.1e+03
neos-1445532	1924	14406	27736	1.9e+03	1	1.3e+02
neos-1445738	2145	20631	40256	2.1e+03	1	44
neos-1445743	2148	20344	39685	2.1e+03	1	43
neos-1445755	2139	20516	40020	2.1e+03	1	47
neos-1445765	2147	20617	40230	2.1e+03	1	48
neos-1451294	1238	1626	21036	1e+02	1	1
neos-1456979	6770	4605	36440	9.7e+02	1	5e+02
neos-1460246	306	285	2303	38	1	10
neos-1460265	1656	1728	11902	2.8e+02	1	10
neos-1460543	2012	1700	15121	2.5e+02	1	10
neos-1460641	1532	1641	11697	2.2e+02	1	10
neos-1461051	4370	528	14220	1.7e+02	1	2
neos-1464762	1632	1721	12313	2.3e+02	1	9.2
neos-1467067	1084	1196	4692	3.6e+02	1	3
neos-1467371	1628	1693	12084	2.3e+02	1	9.2
neos-1467467	1644	1693	12116	2.3e+02	1	9.2
neos-1480121	363	222	1060	91	1	1e+03

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
neos-1489999	1046	534	2186	2.6e+02	0.015	1
neos-1516309	489	4500	30400	82	1	5
neos-1582420	10180	10100	24814	5.1e+03	0.1	3
neos-1593097	798	18460	113308	1.3e+02	1	7.3e+02
neos-1595230	1750	490	3885	2.5e+02	1	1
neos-1597104	109833	714	331373	2.4e+02	1	1
neos-1599274	1237	4500	46800	1.2e+02	1	5
neos-1601936	3131	4446	72500	2e+02	1	1
neos-1603512	555	730	13541	31	1	3
neos-1603518	880	1272	25716	44	1	3
neos-1603965	28984	15003	86947	5.8e+03	0.05	1e+10
neos-1605061	3474	4111	93483	1.6e+02	1	1e+04
neos-1605075	3467	4173	91377	1.7e+02	1	1
neos-1616732	1999	200	3998	1.1e+02	1	1
neos-1620770	9296	792	19292	3.9e+02	1	1
neos-1620807	1340	231	2860	1.1e+02	1	1
neos-1622252	9695	828	20125	4e+02	1	1
neos-430149	990	395	2895	1.4e+02	0.004	2.5e+02
neos-476283	10015	11915	3945693	30	0.0002	1e+04
neos-480878	1321	534	44370	16	0.0002	1e+04
neos-494568	2215	6889	115463	1.4e+02	1	1e+02
neos-495307	3	9423	27831	1.5	1	1.8e+03
neos-498623	2047	9861	148434	1.4e+02	1	1e+02
neos-501453	40	165	535	13	0.62	40
neos-501474	265	206	2228	26	0.62	40
neos-503737	500	2850	16850	1e+02	1	1
neos-504674	1344	844	3450	3.4e+02	0.1	78
neos-504815	1067	674	2736	2.7e+02	0.1	63
neos-506422	6811	2527	31815	5.7e+02	1	40
neos-506428	129925	42981	343466	1.9e+04	4.5e-08	33
neos-512201	1337	838	3418	3.3e+02	0.1	78
neos-520729	31178	91149	322203	1e+04	1	5e+03
neos-522351	1705	1524	5436	5.7e+02	0.1	5e+04
neos-525149	144120	3640	1519200	3.5e+02	1	2
neos-530627	113	103	324	38	1	1.3e+02
neos-538867	1170	792	3888	2.9e+02	1	1
neos-538916	1314	864	4272	3.3e+02	1	1
neos-544324	732	10080	1757280	4.2	1	1
neos-547911	693	3528	615048	4	1	1
neos-548047	3970	2020	26405	3.1e+02	1	1
neos-548251	2386	1922	5791	8e+02	1	31
neos-551991	3332	1730	31631	1.9e+02	1	1
neos-555001	3474	3855	16649	8.7e+02	0.2	40
neos-555298	2755	4827	20145	6.9e+02	0.2	40
neos-555343	3326	3815	16967	8.3e+02	0.2	40
neos-555424	2676	3815	15667	6.7e+02	0.2	40
neos-555694	1948	4139	39543	2.2e+02	0.01	1e+02
neos-555771	1978	4170	40349	2.2e+02	0.01	1e+02
neos-555884	4331	3815	19067	1.1e+03	0.2	40
neos-555927	1403	1945	7965	3.5e+02	0.2	40
neos-565672	318334	190589	809816	8e+04	2.9e-07	44
neos-565815	15413	1276	124071	1.6e+02	1	1
neos-570431	931	511	12041	40	1	1
neos-574665	3790	740	16792	1.7e+02	0.93	1.8e+05
neos-578379	21703	17010	101560	4.3e+03	1	1
neos-582605	1240	1265	3735	6.2e+02	1	2
neos-583731	1491	1350	5220	5e+02	0.5	1

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
neos-584146	936	811	3035	3.1e+02	0.5	2
neos-584851	661	445	1709	2.2e+02	1	1
neos-584866	9009	3674	21338	1.8e+03	1	1
neos-585192	2628	2597	72396	97	0.81	1e+06
neos-585467	2166	2116	50058	94	0.89	1e+06
neos-593853	1606	2400	6000	8e+02	1	1.1e+07
neos-595904	2452	4508	22364	6.1e+02	1	1e+03
neos-595905	704	1200	5788	1.8e+02	1	1e+03
neos-595925	956	1276	5960	2.4e+02	1	1e+03
neos-598183	992	1696	8388	2.5e+02	1	1e+03
neos-603073	992	1696	8388	2.5e+02	1	1e+03
neos-611135	5277	6400	769300	44	0.25	2e+02
neos-611838	1876	9954	37027	6.3e+02	1	3e+04
neos-612125	1795	9554	35791	6e+02	1	3e+04
neos-612143	1842	9832	36643	6.1e+02	1	3e+04
neos-612162	1859	9893	36835	6.2e+02	1	3e+04
neos-619167	6800	3452	20020	1.4e+03	0.053	1e+06
neos-631164	406	1282	3156	2e+02	1	1.6e+05
neos-631517	351	1090	2743	1.8e+02	1	1.6e+05
neos-631694	3996	3725	18523	1e+03	1	4
neos-631709	46496	45150	225148	1.2e+04	1	4
neos-631710	169576	167056	834166	4.2e+04	1	4
neos-631784	23996	22725	113023	6e+03	1	4
neos-632335	24864	12719	73025	5e+03	1	1
neos-633273	21781	11154	63910	4.4e+03	1	1
neos-641591	1085	18235	200055	1.1e+02	1	5
neos-655508	13573	13572	40484	6.8e+03	1	1
neos-662469	1085	18235	200055	1.1e+02	1	5
neos-686190	3664	3660	18085	9.2e+02	1	69
neos-691058	2667	3006	30837	2.7e+02	1	7
neos-691073	2667	1935	29766	1.8e+02	1	7
neos-693347	3192	1576	113472	44	1	1
neos-702280	1600	7199	2421882	4.8	1	1
neos-709469	469	224	4432	25	1	10
neos-717614	891	3049	10477	3e+02	0.0024	1.6e+04
neos-738098	25849	9093	101360	2.3e+03	1	12
neos-775946	6602	4710	107876	3e+02	0.01	1e+02
neos-777800	479	6400	32000	96	1	1
neos-780889	73910	182700	497210	3.7e+04	1	3
neos-785899	1653	1320	17180	1.3e+02	1	1
neos-785912	1714	1380	16610	1.4e+02	1	1
neos-785914	1590	1260	15290	1.3e+02	1	1
neos-787933	1897	236376	298320	1.9e+03	1	1.3e+02
neos-791021	3694	9448	29708	1.2e+03	1	12
neos-796608	286	311	778	1.4e+02	1	4
neos-799711	59218	41998	147164	2e+04	0.00058	1e+08
neos-799838	5976	20844	57888	3e+03	1	50
neos-801834	3300	3220	55200	1.9e+02	1	1
neos-803219	901	640	3020	2.3e+02	0.0079	1e+02
neos-803220	891	630	2980	2.2e+02	0.0079	1e+02
neos-806323	1541	1060	5650	3.1e+02	0.0079	6.9e+02
neos-807454	1622	1638	35272	77	1	1
neos-807456	840	1635	4905	2.8e+02	1	1
neos-807639	1541	1030	5520	3.1e+02	0.0079	6e+02
neos-807705	1541	1030	5520	3.1e+02	0.0079	6e+02
neos-808072	1713	1702	38054	78	1	1
neos-808214	640	1308	22530	38	1	1

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
neos-810286	2675	2915	69952	1.2e+02	1	1
neos-810326	1749	1702	38810	80	1	1
neos-820146	830	600	3225	1.7e+02	1	1
neos-820157	1015	1200	4875	2.5e+02	1	1
neos-820879	361	9522	72356	52	1	8e+03
neos-824661	18804	45390	138890	6.3e+03	1	1e+03
neos-824695	9576	23970	72590	3.2e+03	1	1e+03
neos-825075	328	800	5480	55	1	1
neos-826224	17266	41820	127840	5.8e+03	1	1e+03
neos-826250	5250	12250	37520	1.7e+03	1	1e+03
neos-826650	2414	5912	20440	8e+02	1	1e+03
neos-826694	6904	16410	59268	2.3e+03	1	1e+03
neos-826812	6844	15864	53808	2.3e+03	1	1e+03
neos-826841	2354	5516	18460	7.8e+02	1	1e+03
neos-827015	7688	79347	166239	3.8e+03	1	1
neos-827175	14187	32504	110790	4.7e+03	1	1e+03
neos-829552	5153	40971	86952	2.6e+03	1	1
neos-830439	1375	1468	4804	4.6e+02	1	1e+02
neos-831188	2185	4612	11256	1.1e+03	1	1
neos-839838	12751	7700	47800	2.1e+03	1	1e+08
neos-839859	3251	1975	12025	5.4e+02	1	1e+08
neos-839894	33201	16325	98825	5.5e+03	1	1e+06
neos-841664	3135	2925	10920	1e+03	1	1e+04
neos-847051	4731	5417	19372	1.6e+03	0.00011	1e+06
neos-847302	609	737	9566	51	1	1
neos-848150	731	949	12300	61	1	1
neos-848198	924	10164	29106	4.6e+02	1	1e+03
neos-848589	1484	550539	1101078	7.4e+02	1	1e+06
neos-848845	1050	1737	19470	95	1	1
neos-849702	1041	1737	19308	95	1	1
neos-850681	2067	2594	37113	1.5e+02	1	6.7e+02
neos-856059	17827	450	35654	2.3e+02	1	1
neos-859770	2065	2504	880736	5.9	1	1
neos-860244	675	3105	413305	5.1	1	1
neos-860300	850	1385	384329	3.1	1	21
neos-862348	5801	3835	81027	2.8e+02	0.01	1e+02
neos-863472	523	588	5440	58	0.43	45
neos-872648	93291	175219	350438	4.7e+04	1	1
neos-873061	93360	175288	350576	4.7e+04	1	1
neos-876808	85808	87268	682376	1.2e+04	1	22
neos-880324	348	261	1484	70	1	1e+03
neos-881765	278	712	7208	28	1	1
neos-885086	11574	4860	248310	2.3e+02	1	1.6e+03
neos-885524	65	91670	258309	32	3	1.3e+03
neos-886822	1089	1057	4128	3.6e+02	1	1.5e+04
neos-892255	2137	1800	10005	4.3e+02	1	25
neos-905856	403	686	6601	45	1	1
neos-906865	1634	1184	5728	4.1e+02	1	1
neos-911880	83	888	2568	42	1	1.4e+02
neos-911970	107	888	3408	36	1	1.4e+02
neos-912015	617	686	14742	29	1	1
neos-912023	623	686	14728	30	1	1
neos-913984	1076	76000	152000	5.4e+02	1	6
neos-914441	15129	15007	59658	5e+03	1	1.8e+08
neos-916173	1413	1084	72701	21	0.015	1e+03
neos-916792	1909	1474	134442	21	0.017	1e+03
neos-930752	6549	9674	27864	3.3e+03	1	1

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
neos-931517	5529	7920	29565	1.8e+03	1	2
neos-931538	5964	7920	33480	1.5e+03	1	2
neos-932721	18085	22266	107908	4.5e+03	1	1
neos-932816	30823	21007	484926	1.3e+03	1	1
neos-933364	1006	1728	6768	3.4e+02	1	1.4e+02
neos-933550	2288	3032	13776	5.7e+02	1	1
neos-933562	3200	3032	28800	3.6e+02	1	1
neos-933638	13658	32417	187173	2.7e+03	1	1
neos-933815	947	1728	5088	4.7e+02	1	1.4e+02
neos-933966	12047	31762	180618	2.4e+03	1	1
neos-934184	1006	1728	6768	3.4e+02	1	1.4e+02
neos-934278	11495	23123	125577	2.3e+03	1	1
neos-934441	11691	23362	127383	2.3e+03	1	1
neos-934531	47078	1082	136119	3.8e+02	1	1e+05
neos-935234	9568	10309	55271	1.9e+03	1	1
neos-935348	7859	10301	40476	2.6e+03	1	1
neos-935496	2890	2820	27984	3.2e+02	1	1
neos-935627	7859	10301	40476	2.6e+03	1	1
neos-935674	2890	3108	28560	3.2e+02	1	1
neos-935769	6741	9799	36447	2.2e+03	1	1
neos-936660	7311	10019	39546	2.4e+03	1	1
neos-937446	8176	11341	44697	2.7e+03	1	1
neos-937511	8158	11332	44237	2.7e+03	1	1
neos-937815	9251	11646	48013	2.3e+03	1	1
neos-941262	6703	9480	35659	2.2e+03	1	1
neos-941313	13189	167910	484080	6.6e+03	1	1.3e+02
neos-941698	844	946	13002	65	1	1
neos-941717	1092	1350	20214	78	1	1
neos-941782	968	1094	17086	65	1	1
neos-942323	754	732	10884	54	1	1
neos-942830	803	882	13290	54	1	1
neos-942886	359	464	7109	24	1	1
neos-948126	7271	9551	38219	1.8e+03	1	1
neos-948268	4773	7550	26410	1.6e+03	1	1
neos-948346	1570	57855	540443	1.7e+02	1	1e+04
neos-950242	34224	5760	104160	1.9e+03	1	1
neos-952987	354	31329	90384	1.8e+02	1	1.0e+03
neos-953928	12498	23305	169861	1.8e+03	1	53
neos-954925	2989	84718	844983	3.3e+02	1	53
neos-955215	723	1302	3822	3.6e+02	1	1.1e+02
neos-955800	6516	1848	19536	6.5e+02	1	1
neos-956971	2527	57756	483560	3.2e+02	1	53
neos-957143	2767	57756	497676	3.5e+02	1	53
neos-957270	3282	5929	417968	47	1	43
neos-957323	3757	57756	499656	4.7e+02	1	53
neos-957389	5115	6036	355372	88	1	30
neos-960392	4744	59376	189503	1.6e+03	1	53
neos-983171	6711	8965	36691	1.7e+03	1	1
neos-984165	6962	8883	36742	1.7e+03	1	1
neos1	131581	1892	468009	5.3e+02	1	1
neos13	20852	1827	253842	1.5e+02	1.5e-05	2.3e+02
neos15	552	792	1766	2.8e+02	1	1e+03
neos16	1018	377	2801	1.5e+02	1	7
neos18	11402	3312	24614	1.6e+03	1	1
neos2	132568	1560	552519	3.7e+02	1	1
neos6	1036	8786	251946	37	1	18
neos788725	433	352	4912	33	1	5

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
neos808444	18329	19846	120512	3.1e+03	1	79
neos858960	132	160	2770	7.8	1	1
nesm	662	2923	13288	1.7e+02	0.001	33
net12	14021	14115	80384	2.8e+03	1	12
netdiversion	119589	129180	615282	3e+04	1	1
netlarge2	40000	1160000	2320000	2e+04	1	1
newdano	576	505	2184	1.4e+02	1	81
nl	7039	9718	41428	1.8e+03	0.0004	2.2e+02
nobel-eu-DBE	879	3771	11313	2.9e+02	1	7.7e+04
noswot	182	128	735	36	0.25	21
npmv07	76342	220686	859614	2.5e+04	0.003	7.1e+05
ns1111636	13895	360822	568444	1.4e+04	1	1.9e+02
ns1116954	131991	12648	410582	4.1e+03	1	10
ns1208400	4289	2883	81746	1.5e+02	0.2	1
ns1456591	1997	8399	199862	87	1	3e+04
ns1606230	3503	4173	92133	1.6e+02	1	1
ns1631475	24496	22696	116733	4.9e+03	1	4.8e+03
ns1644855	40698	30200	2110696	5.9e+02	1	20
ns1663818	172017	124626	20433649	1.1e+03	1	1e+03
ns1685374	44121	10000	220859	2e+03	0.01	1
ns1686196	4055	2738	68529	1.6e+02	1	1e+03
ns1688347	4191	2685	66908	1.7e+02	1	1e+03
ns1696083	11063	7982	384129	2.3e+02	1	1e+03
ns1702808	1474	804	5856	2.1e+02	1	1e+04
ns1745726	4687	3208	90278	1.7e+02	1	1e+03
ns1758913	624166	17956	1283444	8.8e+03	1	2.7e+03
ns1766074	182	100	666	30	1	10
ns1769397	5527	3772	117383	1.8e+02	1	1e+03
ns1778858	10666	4720	32673	1.8e+03	1	7.3e+05
ns1830653	2932	1629	100933	48	1	2.9e+02
ns1856153	35407	11998	105882	4.4e+03	0.5	1e+03
ns1904248	149437	38458	378770	1.7e+04	6.1e-17	98
ns1905797	51884	18192	239700	4e+03	0.2	50
ns1905800	8289	3228	38100	7.5e+02	0.75	50
ns1952667	41	13264	335643	1.6	1	1.9e+02
ns2017839	54510	55224	317840	1.1e+04	0.33	3.5e+07
ns2081729	1190	661	5680	1.5e+02	0.5	1e+02
ns2118727	163354	167440	646864	5.4e+04	0.042	8.8
ns2122603	24754	19300	77044	8.3e+03	0.042	1e+08
ns2124243	139280	156083	429032	7e+04	0.5	1
ns2137859	206726	103361	923682	2.6e+04	1	2e+03
ns4-pr3	2210	8601	25986	7.4e+02	1	60
ns4-pr9	2220	7350	22176	7.4e+02	1	35
ns894236	8218	9666	41067	2.1e+03	1	1
ns894244	12129	21856	90864	3e+03	1	1
ns894786	16794	27278	113575	4.2e+03	1	1
ns894788	2279	3463	14381	5.7e+02	1	1
ns903616	18052	21582	91641	4.5e+03	1	1
ns930473	23240	11328	121764	2.3e+03	1	1.2e+05
nsa	1297	388	4204	1.3e+02	1	1
nsct1	22901	14981	656259	5.3e+02	1	1.4e+05
nsct2	23003	14981	675156	5.1e+02	1	1.4e+05
nsic1	451	463	2853	75	1	5e+05
nsic2	465	463	3015	78	1	5e+05
nsir1	4407	5717	138955	1.8e+02	1	2e+05
nsir2	4453	5717	150599	1.7e+02	1	2e+05
nsr8k	6284	38356	371608	7e+02	1	1e+08

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
nsrand-ipx	735	6621	223261	22	1	1.8e+04
nu120-pr3	2210	8601	25986	7.4e+02	1	1.2e+02
nu60-pr9	2220	7350	22176	7.4e+02	1	60
nug05	210	225	1050	52	1	1
nug06	372	486	2232	93	1	1
nug07	602	931	4214	1.5e+02	1	1
nug08	912	1632	7296	2.3e+02	1	1
nug08-3rd	19728	20448	139008	3.3e+03	1	1
nug12	3192	8856	38304	8e+02	1	1
nw04	36	87482	636666	5.1	1	1
nw14	73	123409	904910	10	1	1
ofi	422587	420434	1778754	1.1e+05	3e-22	1e+11
opm2-z10-s2	160633	6250	371243	2.7e+03	1	4.1e+03
opm2-z11-s8	223082	8019	510283	3.5e+03	1	4.1e+03
opm2-z12-s14	319508	10800	725376	4.8e+03	1	4.1e+03
opm2-z12-s7	319508	10800	725385	4.8e+03	1	4.1e+03
opm2-z7-s2	31798	2023	79762	8.2e+02	1	4.1e+03
opt1217	64	769	1542	32	1	8
orna1	882	882	3108	2.9e+02	1.5	1.4e+04
orna2	882	882	3108	2.9e+02	1.5	1.4e+04
orna3	882	882	3108	2.9e+02	1.5	1.4e+04
orna4	882	882	3108	2.9e+02	1.5	1.4e+04
orna7	882	882	3108	2.9e+02	1.5	1.4e+04
orswq2	80	80	264	27	0.023	42
osa-07	1118	23949	143694	1.9e+02	0.29	13
osa-14	2337	52460	314760	3.9e+02	0.29	13
osa-30	4350	100024	600138	8.7e+02	0.16	13
osa-60	10280	232966	1397793	2.1e+03	0.35	13
p0033	16	33	98	8	1	4e+02
p0040	23	40	110	12	1	2.2e+03
p010	10090	19000	117910	1.7e+03	0.1	3
p0201	133	201	1923	15	1	64
p0282	241	282	1966	40	1	2e+02
p0291	252	291	2031	42	0.4	52
p05	5090	9500	58955	8.5e+02	0.1	3
p0548	176	548	1711	59	1	1e+04
p100x588b	688	1176	2352	3.4e+02	1	9e+02
p19	284	586	5305	32	0.0001	5.4
p2756	755	2756	8937	2.5e+02	1	1e+04
p2m2p1m1p0n100	1	100	100	1	6.6e+03	1.4e+04
p6000	2095	5872	17731	7e+02	1	9.9e+04
p6b	5852	462	11704	2.3e+02	1	1
p80x400b	480	800	1600	2.4e+02	1	8e+02
pcb1000	1565	2428	20071	2e+02	1	2
pcb3000	3960	6810	56557	5e+02	1	2
pds-02	2953	7535	16390	1.5e+03	1	1
pds-06	9881	28655	62524	4.9e+03	1	1
pds-10	16558	48763	106436	8.3e+03	1	1
pds-100	156243	505360	1086785	7.8e+04	1	1
pds-20	33874	105728	230200	1.7e+04	1	1
pds-30	49944	154998	337144	2.5e+04	1	1
pds-40	66844	212859	462128	3.3e+04	1	1
pds-50	83060	270095	585114	4.2e+04	1	1
pds-60	99431	329643	712779	5e+04	1	1
pds-70	114944	382311	825771	5.7e+04	1	1
pds-80	129181	426278	919524	6.5e+04	1	1
pds-90	142823	466671	1005359	7.1e+04	1	1

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
perold	625	1376	6018	1.6e+02	5.3e-05	2.4e+04
pf2177	9728	900	21706	4.1e+02	1	1
pg	125	2700	5200	1.2e+02	1	1.6e+03
pg5_34	225	2600	7700	1.1e+02	1	1.5e+03
pgp2	4034	9220	18440	2e+03	1	16
pigeon-10	931	490	8150	58	1	1e+03
pigeon-11	1123	572	9889	66	1	1e+03
pigeon-12	1333	660	11796	78	1	1e+03
pigeon-13	1561	754	13871	87	1	1e+03
pigeon-19	3307	1444	29849	1.7e+02	1	1e+03
pilot	1441	3652	43167	1.3e+02	1e-06	1.5e+02
pilot-ja	940	1988	14698	1.3e+02	2e-06	5.9e+06
pilot-we	722	2789	9126	2.4e+02	0.00014	4.8e+04
pilot4	410	1000	5141	82	3.7e-05	2.8e+04
pilot87	2030	4883	73152	1.4e+02	1e-06	1e+03
pilotnov	975	2172	13057	1.6e+02	2e-06	5.9e+06
pk1	45	86	915	4.5	1	55
pldd000b	3069	3267	8980	1.5e+03	1e-06	1.1e+02
pldd001b	3069	3267	8981	1.5e+03	1e-06	1.1e+02
pldd002b	3069	3267	8982	1.5e+03	1e-06	1.1e+02
pldd003b	3069	3267	8983	1.5e+03	1e-06	1.1e+02
pldd004b	3069	3267	8984	1.5e+03	1e-06	1.1e+02
pldd005b	3069	3267	8985	1.5e+03	1e-06	1.1e+02
pldd006b	3069	3267	8986	1.5e+03	1e-06	1.1e+02
pldd007b	3069	3267	8987	1.5e+03	1e-06	1.1e+02
pldd008b	3069	3267	9047	1.5e+03	1e-06	1.1e+02
pldd009b	3069	3267	9050	1.5e+03	1e-06	1.1e+02
pldd010b	3069	3267	9053	1.5e+03	1e-06	1.1e+02
pldd011b	3069	3267	9055	1.5e+03	1e-06	1.1e+02
pldd012b	3069	3267	9057	1.5e+03	1e-06	1.1e+02
pltexpa2-16	1726	4540	9233	8.6e+02	1	1e+03
pltexpa2-6	686	1820	3703	3.4e+02	1	1e+03
pltexpa3_16	28350	74172	150801	1.4e+04	1	1e+03
pltexpa3_6	4430	11612	23611	2.2e+03	1	1e+03
pltexpa4_6	26894	70364	143059	1.3e+04	1	1e+03
pp08a	136	240	480	68	1	5e+02
pp08aCUTS	246	240	839	82	1	5e+02
primagaz	1554	10836	21665	1.6e+03	1	1
problem	12	46	86	12	1	1
probportfolio	302	320	6620	15	0.8	1.5
prod1	208	250	5350	9.9	6.2e-05	7
prod2	211	301	10501	6.2	6.2e-05	10
progas	1650	1425	8422	3.3e+02	3.9e-05	1e+04
protfold	2112	1835	23491	1.8e+02	1	1
pw-myciel4	8164	1059	17779	5.1e+02	1	1
qap10	1820	4150	18200	4.6e+02	1	1
qiu	1192	840	3432	3e+02	0.26	22
qiulp	1192	840	3432	3e+02	0.26	22
qnet1	503	1541	4622	2.5e+02	1	4.1e+03
qnet1_o	456	1541	4214	2.3e+02	1	4.1e+03
queens-30	960	900	93440	9.3	1	7
r05	5190	9500	103955	5.2e+02	0.1	3
r80x800	880	1600	3200	4.4e+02	1	1e+03
rail01	46843	117527	392086	1.6e+04	1	2.3e+02
rail2586	2586	920683	8008776	3.2e+02	1	1
rail4284	4284	1092610	11279748	4.3e+02	1	1
rail507	509	63019	468878	73	1	1

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
rail516	516	47311	314896	86	1	1
rail582	582	55515	401708	83	1	1
ramos3	2187	2187	32805	1.5e+02	1	1
ran10x10a	120	200	400	60	1	16
ran10x10b	120	200	400	60	1	19
ran10x10c	120	200	400	60	1	17
ran10x12	142	240	480	71	1	30
ran10x26	296	520	1040	1.5e+02	1	56
ran12x12	168	288	576	84	1	26
ran12x21	285	504	1008	1.4e+02	1	50
ran13x13	195	338	676	98	1	28
ran14x18	284	504	1008	1.4e+02	1	38
ran14x18-disj-8	447	504	10277	22	3.2e-09	38
ran14x18.disj-8	447	504	10277	22	3.2e-09	38
ran14x18_1	284	504	1008	1.4e+02	1	38
ran16x16	288	512	1024	1.4e+02	1	54
ran17x17	323	578	1156	1.6e+02	1	35
ran4x64	324	512	1024	1.6e+02	1	44
ran6x43	307	516	1032	1.5e+02	1	49
ran8x32	296	512	1024	1.5e+02	1	44
rat1	3136	9408	88267	3.5e+02	0.00023	1
rat5	3136	9408	137413	2.2e+02	0.0012	1
rat7a	3136	9408	268908	1.1e+02	0.0004	1
rd-rplusc-21	125899	622	852384	92	0.2	1e+07
reblock166	17024	1660	39442	7.4e+02	1	2.5e+04
reblock354	19906	3540	52901	1.4e+03	1	7.2e+02
reblock420	62800	4200	138670	1.9e+03	1	9.1e+03
reblock67	2523	670	7495	2.3e+02	1	6.5e+03
recipe	91	180	663	30	0.12	1.4e+02
refine	29	33	124	9.7	0.5	66
rentacar	6803	9557	41842	1.7e+03	0.01	1e+05
rgn	24	180	460	12	1	4.6
rlfddd	4050	57471	260577	1e+03	1	1
rlfdual	8052	66918	273979	2e+03	1	1
rlfprim	58866	8052	265927	1.8e+03	1	1
rlp1	68	461	836	68	1	14
rmatr100-p10	7260	7359	21877	3.6e+03	1	1
rmatr100-p5	8685	8784	26152	4.3e+03	1	1
rmatr200-p10	35055	35254	105362	1.8e+04	1	1
rmatr200-p20	29406	29605	88415	1.5e+04	1	1
rmatr200-p5	37617	37816	113048	1.9e+04	1	1
rmine10	65274	8439	162264	3.4e+03	1	1e+02
rmine14	268535	32205	660346	1.3e+04	1	1e+02
rmine6	7078	1096	18084	4.4e+02	1	1e+02
rocII-4-11	21738	9234	243106	8.4e+02	0.38	12
rocII-7-11	37215	16101	423661	1.4e+03	0.38	12
rocII-9-11	47533	20679	544031	1.8e+03	0.38	12
rococoB10-011000	1667	4456	16517	5.6e+02	1	1.6e+03
rococoC10-001000	1293	3117	11751	4.3e+02	1	3.4e+04
rococoC11-011100	2367	6491	30472	5.9e+02	1	4.1e+03
rococoC12-111000	10776	8619	48920	2.2e+03	1	3.2e+04
roll3000	2295	1166	29386	92	0.25	3.2e+02
rosen1	520	1024	23274	24	1	9
rosen10	2056	4096	62136	1.4e+02	1	9
rosen2	1032	2048	46504	47	1	9
rosen7	264	512	7770	18	1	9
rosen8	520	1024	15538	35	1	9

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
rout	291	556	2431	73	0.22	1.4e+02
route	20894	23923	187686	3e+03	1	1.5e+03
roy	162	149	411	81	1	30
rvb-sub	225	33765	984143	7.8	0.00035	1
satellites1-25	5996	9013	59023	1e+03	0.36	2.3e+05
satellites2-60	20916	35378	283668	2.6e+03	0.36	2.4e+05
satellites2-60-fs	16516	35378	125048	5.5e+03	0.36	2.4e+05
satellites3-40	44804	81681	698176	5.6e+03	0.36	2.4e+05
satellites3-40-fs	35553	81681	291161	1.2e+04	0.36	2.4e+05
sc105	105	103	280	52	0.1	2
sc205	205	203	551	1e+02	0.1	2
sc205-2r-100	2213	2214	6030	1.1e+03	1	2
sc205-2r-16	365	366	990	1.8e+02	1	2
sc205-2r-1600	35213	35214	96030	1.8e+04	1	2
sc205-2r-200	4413	4414	12030	2.2e+03	1	2
sc205-2r-27	607	608	1650	3e+02	1	2
sc205-2r-32	717	718	1950	3.6e+02	1	2
sc205-2r-4	101	102	270	50	1	2
sc205-2r-400	8813	8814	24030	4.4e+03	1	2
sc205-2r-50	1113	1114	3030	5.6e+02	1	2
sc205-2r-64	1421	1422	3870	7.1e+02	1	2
sc205-2r-8	189	190	510	94	1	2
sc205-2r-800	17613	17614	48030	8.8e+03	1	2
sc50a	50	48	130	25	0.1	2
sc50b	50	48	118	25	0.3	3
scagr25	471	500	1554	1.6e+02	0.2	9.3
scagr7	129	140	420	43	0.2	9.3
scagr7-2b-16	623	660	2058	2.1e+02	0.2	9.3
scagr7-2b-4	167	180	546	56	0.2	9.3
scagr7-2b-64	9743	10260	32298	3.2e+03	0.2	9.3
scagr7-2c-16	623	660	2058	2.1e+02	0.2	9.3
scagr7-2c-4	167	180	546	56	0.2	9.3
scagr7-2c-64	2447	2580	8106	8.2e+02	0.2	9.3
scagr7-2r-108	4119	4340	13542	1.4e+03	0.2	9.3
scagr7-2r-16	623	660	2058	2.1e+02	0.2	9.3
scagr7-2r-216	8223	8660	27042	2.7e+03	0.2	9.3
scagr7-2r-27	1041	1100	3444	3.5e+02	0.2	9.3
scagr7-2r-32	1231	1300	4074	4.1e+02	0.2	9.3
scagr7-2r-4	167	180	546	56	0.2	9.3
scagr7-2r-432	16431	17300	54042	5.5e+03	0.2	9.3
scagr7-2r-54	2067	2180	6846	6.9e+02	0.2	9.3
scagr7-2r-64	2447	2580	8106	8.2e+02	0.2	9.3
scagr7-2r-8	319	340	1050	1.1e+02	0.2	9.3
scagr7-2r-864	32847	34580	108042	1.1e+04	0.2	9.3
scfxm1	330	457	2589	66	0.0005	1.3e+02
scfxm1-2b-16	2460	3714	13959	8.2e+02	0.001	62
scfxm1-2b-4	684	1014	3999	2.3e+02	0.001	62
scfxm1-2b-64	19036	28914	106919	6.3e+03	0.001	62
scfxm1-2c-4	684	1014	3999	2.3e+02	0.001	62
scfxm1-2r-128	19036	28914	106919	6.3e+03	0.001	62
scfxm1-2r-16	2460	3714	13959	8.2e+02	0.001	62
scfxm1-2r-256	37980	57714	213159	1.3e+04	0.001	62
scfxm1-2r-27	4088	6189	23089	1.4e+03	0.001	62
scfxm1-2r-32	4828	7314	27239	1.6e+03	0.001	62
scfxm1-2r-4	684	1014	3999	2.3e+02	0.001	62
scfxm1-2r-64	9564	14514	53799	3.2e+03	0.001	62
scfxm1-2r-8	1276	1914	7319	4.3e+02	0.001	62

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
scfxm1-2r-96	14300	21714	80359	4.8e+03	0.001	62
scfxm2	660	914	5183	1.3e+02	0.0005	1.3e+02
scfxm3	990	1371	7777	2e+02	0.0005	1.3e+02
scorpion	388	358	1426	1.3e+02	0.01	1
scrs8	490	1169	3182	2.4e+02	0.001	3.9e+02
scrs8-2b-16	476	645	1633	2.4e+02	0.001	36
scrs8-2b-4	140	189	457	70	0.001	36
scrs8-2b-64	1820	2469	6337	9.1e+02	0.001	36
scrs8-2c-16	476	645	1633	2.4e+02	0.001	36
scrs8-2c-32	924	1253	3201	4.6e+02	0.001	36
scrs8-2c-4	140	189	457	70	0.001	36
scrs8-2c-64	1820	2469	6337	9.1e+02	0.001	36
scrs8-2c-8	252	341	849	1.3e+02	0.001	36
scrs8-2r-128	3612	4901	12609	1.8e+03	0.001	36
scrs8-2r-16	476	645	1633	2.4e+02	0.001	36
scrs8-2r-256	7196	9765	25153	3.6e+03	0.001	36
scrs8-2r-27	784	1063	2711	3.9e+02	0.001	36
scrs8-2r-32	924	1253	3201	4.6e+02	0.001	36
scrs8-2r-4	140	189	457	70	0.001	36
scrs8-2r-512	14364	19493	50241	7.2e+03	0.001	36
scrs8-2r-64	1820	2469	6337	9.1e+02	0.001	36
scrs8-2r-64b	1820	2469	6337	9.1e+02	0.001	36
scrs8-2r-8	252	341	849	1.3e+02	0.001	36
scsd1	77	760	2388	26	0.24	1
scsd6	147	1350	4316	49	0.24	1
scsd8	397	2750	8584	1.3e+02	0.24	1
scsd8-2b-16	330	2310	7170	1.1e+02	0.24	1
scsd8-2b-4	90	630	1890	30	0.24	1
scsd8-2b-64	5130	35910	112770	1.7e+03	0.24	1
scsd8-2c-16	330	2310	7170	1.1e+02	0.24	1
scsd8-2c-4	90	630	1890	30	0.24	1
scsd8-2c-64	5130	35910	112770	1.7e+03	0.24	1
scsd8-2r-108	2170	15190	47650	7.2e+02	0.24	1
scsd8-2r-16	330	2310	7170	1.1e+02	0.24	1
scsd8-2r-216	4330	30310	95170	1.4e+03	0.24	1
scsd8-2r-27	550	3850	12010	1.8e+02	0.24	1
scsd8-2r-32	650	4550	14210	2.2e+02	0.24	1
scsd8-2r-4	90	630	1890	30	0.24	1
scsd8-2r-432	8650	60550	190210	2.9e+03	0.24	1
scsd8-2r-54	1090	7630	23890	3.6e+02	0.24	1
scsd8-2r-64	1290	9030	28290	4.3e+02	0.24	1
scsd8-2r-8	170	1190	3650	57	0.24	1
scsd8-2r-8b	170	1190	3650	57	0.24	1
sct1	12154	22886	105571	3e+03	2.1e-06	6e+04
sct32	5440	9767	109654	4.9e+02	0.0012	1.5e+05
sct5	13304	37265	147037	4.4e+03	0.0024	1.9e+05
sctap1	300	480	1692	1e+02	1	80
sctap1-2b-16	990	1584	5740	3.3e+02	1	80
sctap1-2b-4	270	432	1516	90	1	80
sctap1-2b-64	15390	24624	90220	5.1e+03	1	80
sctap1-2c-16	990	1584	5740	3.3e+02	1	80
sctap1-2c-4	270	432	1516	90	1	80
sctap1-2c-64	3390	5424	19820	1.1e+03	1	80
sctap1-2r-108	6510	10416	38124	2.2e+03	1	80
sctap1-2r-16	990	1584	5740	3.3e+02	1	80
sctap1-2r-216	12990	20784	76140	4.3e+03	1	80
sctap1-2r-27	1650	2640	9612	5.5e+02	1	80

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
sctap1-2r-32	1950	3120	11372	6.5e+02	1	80
sctap1-2r-4	270	432	1516	90	1	80
sctap1-2r-480	28830	46128	169068	9.6e+03	1	80
sctap1-2r-54	3270	5232	19116	1.1e+03	1	80
sctap1-2r-64	3870	6192	22636	1.3e+03	1	80
sctap1-2r-8	510	816	2924	1.7e+02	1	80
sctap1-2r-8b	510	816	2924	1.7e+02	1	80
sctap2	1090	1880	6714	3.6e+02	1	80
sctap3	1480	2480	8874	4.9e+02	1	80
seba	515	1028	4352	1.3e+02	1	1.6e+02
self	960	7364	1148845	6.2	2.3e-07	1
set1ch	492	712	1412	4.9e+02	1	1.1e+03
set3-10	3747	4019	13747	1.2e+03	0.022	1.9e+04
set3-15	3747	4019	13747	1.2e+03	0.022	1.8e+04
set3-20	3747	4019	13747	1.2e+03	0.022	1.8e+04
seymour	4944	1372	33549	2.1e+02	1	1
seymour-disj-10	5108	1209	64704	96	1e-08	9
seymour.disj-10	5108	1209	64704	96	1e-08	9
seymourl	4944	1372	33549	2.1e+02	1	1
sgpf5y6	246077	308634	828070	1.2e+05	1	1
share1b	117	225	1151	23	0.1	1.3e+03
share2b	96	79	694	12	0.01	1e+02
shell	536	1775	3556	2.7e+02	1	1
ship04l	402	2118	6332	2e+02	0.014	4.7
ship04s	402	1458	4352	2e+02	0.014	4.7
ship08l	778	4283	12802	3.9e+02	0.011	5
ship08s	778	2387	7114	3.9e+02	0.011	5
ship12l	1151	5427	16170	5.8e+02	0.0062	1.6
ship12s	1151	2763	8178	5.8e+02	0.0062	1.6
shipsched	45554	13594	121571	5.7e+03	1	7.3e+04
shs1023	133944	444625	1044725	6.7e+04	9.2e-06	42
siena1	2220	13741	258915	1.2e+02	1	1e+08
sierra	1227	2036	7302	4.1e+02	1	1e+05
sing2	28891	31630	149712	7.2e+03	0.36	4e+02
sing245	143161	235146	652817	7.2e+04	0.044	4.3e+02
sing359	437116	713762	1975605	2.2e+05	0.044	4.3e+02
slptsk	2861	3347	72465	1.4e+02	0.0044	16
small000	709	1140	2749	3.5e+02	6e-06	1e+03
small001	687	1140	2871	3.4e+02	6e-06	1e+03
small002	713	1140	2946	3.6e+02	6e-06	1e+03
small003	711	1140	2945	3.6e+02	6e-06	1e+03
small004	717	1140	2983	3.6e+02	1e-06	1e+03
small005	717	1140	3017	3.6e+02	1e-06	1e+03
small006	710	1138	3024	3.6e+02	6e-06	1e+03
small007	711	1137	3079	3.6e+02	6e-06	1e+03
small008	712	1134	3042	3.6e+02	6e-06	1e+03
small009	710	1135	3030	3.6e+02	6e-06	1e+03
small010	711	1138	3027	3.6e+02	6e-06	1e+03
small011	705	1133	3005	3.5e+02	6e-06	1e+03
small012	706	1134	3014	3.5e+02	6e-06	1e+03
small013	701	1131	2989	3.5e+02	2e-06	1e+03
small014	687	1130	2927	3.4e+02	6e-06	1e+03
small015	683	1130	2967	3.4e+02	6e-06	1e+03
small016	677	1130	2937	3.4e+02	1e-06	1e+03
south31	18425	35421	111498	6.1e+03	0.021	1.1e+04
sp97ar	1761	14101	290968	88	1	21
sp97ic	2086	1662	66632	52	1	2.2e+02

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
sp98ar	4680	5478	231756	1.1e+02	1	8.4e+02
sp98ic	2311	2508	138053	42	1	4.5e+02
sp98ir	1531	1680	71704	36	1	5.7e+02
square15	32762	753526	1507052	1.6e+04	1	1
stair	356	467	3856	44	1e-05	9.9
standata	359	1075	3031	1.8e+02	1	3e+02
standmps	467	1075	3679	1.6e+02	1	3e+02
stat96v1	5995	197472	588798	3e+03	0.065	2
stat96v4	3173	62212	490472	4.5e+02	0.00091	20
stat96v5	2307	75779	233921	7.7e+02	1.6e-06	20
stein27	118	27	378	8.4	1	1
stein45	331	45	1034	15	1	1
stocfor1	117	111	447	29	0.063	3.4e+02
stocfor2	2157	2031	8343	5.4e+02	0.2	3.4e+02
stocfor3	16675	15695	64875	4.2e+03	0.063	3.4e+02
stockholm	57346	20644	171076	7.2e+03	1	1e+03
stormG2_1000	528185	1259121	3341696	2.6e+05	1	71
stormg2-125	66185	157496	418321	3.3e+04	1	71
stormg2-27	14441	34114	90903	7.2e+03	1	71
stormg2-8	4409	10193	27424	2.2e+03	1	71
stormg2_1000	528185	1259121	3341696	2.6e+05	1	71
stp3d	159488	204880	662128	5.3e+04	1	1
sts405	27270	405	81810	1.4e+02	1	1
sts729	88452	729	265356	2.4e+02	1	1
swath	884	6805	34965	1.8e+02	1	1.1e+03
sws	14310	12465	93015	2e+03	0.03	1
t0331-4l	664	46915	430982	74	1	1
t1717	551	73885	325689	1.4e+02	1	1
t1722	338	36630	133096	1.1e+02	1	1
tanglegram1	68342	34759	205026	1.4e+04	1	1
tanglegram2	8980	4714	26940	1.8e+03	1	1
testbig	17613	31223	61639	1.8e+04	1	2
timtab1	171	397	829	86	1	60
timtab2	294	675	1482	1.5e+02	1	60
toll-like	4408	2883	13224	1.1e+03	1	1
tr12-30	750	1080	2508	3.8e+02	1	1.5e+03
transportmoment	9616	9685	29541	3.2e+03	1	1e+04
triptim1	15706	30055	515436	9.2e+02	0.0001	6.3e+03
triptim2	14427	27326	521898	7.6e+02	0.0001	1e+03
triptim3	14939	28440	524124	8.3e+02	0.0001	4.1e+03
truss	1000	8806	27836	3.3e+02	0.45	1
tuff	333	587	4520	48	1e-05	1e+04
tw-myciel4	8146	760	27961	2.3e+02	1	1
uc-case11	51438	34134	202042	1e+04	1	1.1e+03
uc-case3	52003	37749	273618	7.4e+03	0.044	3.7e+02
uct-subprob	1973	2256	10147	4.9e+02	1	1
ulevimin	6590	44605	162206	2.2e+03	4.4e-05	2.1e+07
umts	4465	2947	23016	6.4e+02	1	1e+09
unitcal_7	48939	25755	127595	1.2e+04	1	1e+03
us04	163	28016	297538	16	1	1
usAbbrv-8-25_70	3291	2312	9628	8.2e+02	1	36
van	27331	12481	487296	7e+02	1	1.6e+02
vpm1	234	378	749	2.3e+02	1	6e+02
vpm2	234	378	917	1.2e+02	0.025	6e+02
vpphard	47280	51471	372305	6.8e+03	1	1
vpphard2	198450	199999	648340	6.6e+04	1	1
vtp-base	198	203	908	49	0.13	4e+03

Table 4 continued

Instance	columns	rows	nonzeros	sparsity	min. abs.	max. abs.
wachplan	1553	3361	89361	60	1	1
watson_1	201155	383927	1052028	1e+05	0.018	8.8
watson_2	352013	671861	1841028	1.8e+05	0.015	8.3
wide15	32769	753687	1507374	1.6e+04	1	1
wnq-n100-mw99-14	656900	10000	1333400	4.9e+03	1	1
wood1p	244	2594	70215	9	3e-05	1e+03
woodw	1098	8405	37474	2.7e+02	0.01	1e+03
world	34506	32734	164470	6.9e+03	0.0028	5.6e+03
zed	116	43	567	8.9	0.05	3.5e+03
zib54-UUE	1809	5150	15288	9e+02	1	2e+03

C.2 Computational Results

The results of the first experiment comparing the performance of iterative refinements to a precision of 10^{-50} (SoPlex$_{50}$) and 10^{-250} (SoPlex$_{250}$) with the performance of the floating-point solver to 10^{-9} (SoPlex$_9$) are listed in Table 5. Aggregated results can be found in Table 2 in the main paper. The results of the second experiment comparing QSopt_ex warm started from bases returned by SoPlex$_9$ and SoPlex$_{50}$ are given in Table 6. Aggregated results can be found in Table 3 in the main paper.

Table 5: Detailed results comparing iterative refinement to floating-point performance. Entries corresponding to unsolved instances are printed in italics. See Table 2 for aggregated results.

iter — number of simplex iterations
R — number of refinements
R_0 — number of refinements to final basis
t — total running time (in seconds)

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
10teams	1611	0.2	1611	3	0	0.3	1611	17	0	0.3
16_n14	329933	376.0	329933	0	0	371.8	329933	0	0	377.1
22433	1041	0.1	1041	3	0	0.1	1041	17	0	0.1
23588	548	0.0	548	3	0	0.0	548	17	0	0.1
25fv47	4359	0.8	4359	3	0	0.7	4359	17	0	0.8
30_70_45_095_100	16103	7.4	16103	3	0	7.6	16103	17	0	8.3
30n20b8	269	0.3	269	3	0	0.3	269	18	0	0.5
50v-10	220	0.0	220	3	0	0.0	220	15	0	0.1
80bau3b	7797	1.0	7797	3	0	1.1	7797	16	0	1.3
Test3	6948	3.3	6948	3	0	3.7	6948	16	0	5.0
a1c1s1	1742	0.1	1742	3	0	0.1	1742	15	0	0.1
aa01	11898	3.5	11898	3	0	3.6	11898	17	0	3.9
aa03	7584	2.0	7584	3	0	1.9	7584	16	0	2.5
aa3	7584	2.1	7584	3	0	2.0	7584	16	0	2.2
aa4	4486	1.1	4486	3	0	1.2	4486	17	0	1.2
aa5	9706	2.4	9706	3	0	2.6	9706	16	0	2.8
aa6	4870	1.1	4870	3	0	1.2	4870	16	0	1.4
acc-tight4	11142	2.6	11142	3	0	2.2	11142	17	0	2.3
acc-tight5	10469	2.0	10469	3	0	2.2	10469	17	0	2.2
acc-tight6	10672	2.2	10672	3	0	2.3	10672	17	0	2.0
adlittle	88	0.0	88	3	0	0.0	88	16	0	0.0
afiro	16	0.0	16	3	0	0.0	16	15	0	0.0
aflow30a	396	0.0	396	3	0	0.0	396	15	0	0.0
aflow40b	1826	0.2	1826	3	0	0.1	1826	15	0	0.3
agg	80	0.0	80	3	0	0.0	80	16	0	0.1
agg2	152	0.0	152	3	0	0.0	152	16	0	0.1
agg3	165	0.0	165	3	0	0.0	165	16	0	0.1
air02	95	0.1	95	3	0	0.1	95	15	0	0.2
air03	626	0.3	626	2	0	0.4	626	2	0	0.4
air04	11898	3.4	11898	3	0	3.4	11898	17	0	4.0
air05	4486	1.2	4486	3	0	1.3	4486	17	0	1.4
air06	7584	2.0	7584	3	0	2.0	7584	16	0	2.2
aircraft	1912	0.4	1912	3	0	0.2	1912	16	0	0.6
aligninq	1492	0.1	1492	3	0	0.2	1492	16	0	0.3
app1-2	14617	7.2	14622	4	1	8.0	14622	18	1	10.0
arki001	1570	0.2	1571	4	1	0.3	1571	18	1	0.2
ash608gpia-3col	3123	0.2	3123	3	0	0.3	3123	15	0	0.4
atlanta-ip	38285	34.8	38286	3	1	35.2	38286	18	1	37.3
atm20-100	4586	0.4	4699	4	1	0.5	4699	17	1	0.8

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
b2c1s1	2621	0.1	2621	3	0	0.1	2621	16	0	0.2
bab1	5711	4.0	5711	3	0	4.7	5711	16	0	7.6
bab3	697791	5263.4	697791	3	0	5266.5	697791	16	0	5292.6
bab5	29788	10.3	29788	3	0	10.5	29788	16	0	11.3
bal8x12	102	0.0	102	3	0	0.0	102	15	0	0.0
bandm	474	0.0	474	3	0	0.0	474	17	0	0.2
bas1lp	2582	1.0	2582	3	0	1.1	2582	16	0	1.4
baxter	12347	2.6	12347	3	0	2.8	12347	16	0	3.6
bc	3759	1.3	3759	3	0	1.4	3759	18	0	1.6
bc1	3759	1.3	3759	3	0	1.4	3759	18	0	1.6
beaconfd	88	0.0	88	3	0	0.0	88	15	0	0.0
beasleyC3	1140	0.1	1140	3	0	0.1	1140	15	0	0.3
bell3a	81	0.0	81	3	0	0.0	81	16	0	0.0
bell5	66	0.0	66	3	0	0.0	66	16	0	0.0
berlin_5_8_0	1017	0.0	1017	3	0	0.1	1017	15	0	0.1
bg512142	2238	0.3	2238	3	0	0.1	2238	17	0	0.4
biella1	16158	4.2	16158	3	0	4.2	16158	18	0	4.7
bienst1	455	0.0	455	3	0	0.0	455	15	0	0.0
bienst2	455	0.0	455	3	0	0.0	455	15	0	0.0
binkar10_1	1267	0.1	1267	3	0	0.1	1267	16	0	0.1
bk4x3	16	0.0	16	3	0	0.0	16	15	0	0.0
blend	97	0.0	97	3	0	0.0	97	16	0	0.0
blend2	175	0.0	175	3	0	0.0	175	17	0	0.0
blp-ar98	368	0.2	368	3	0	0.5	368	16	0	1.4
blp-ic97	446	0.3	446	3	0	0.4	446	16	0	1.3
bnatt350	685	0.1	685	3	0	0.1	685	15	0	0.2
bnatt400	831	0.1	831	3	0	0.1	831	15	0	0.2
bnl1	1474	0.1	1474	3	0	0.2	1474	17	0	0.5
bnl2	2718	0.4	2718	3	0	0.5	2718	16	0	0.4
boeing1	458	0.0	458	3	0	0.0	458	16	0	0.1
boeing2	145	0.0	145	3	0	0.0	145	16	0	0.0
bore3d	100	0.0	100	3	0	0.0	100	16	0	0.0
brandy	482	0.0	482	3	0	0.0	482	16	0	0.1
buildingenergy	144595	827.1	144595	4	0	835.0	144595	17	0	837.0
cap6000	815	0.2	815	3	0	0.3	815	15	0	0.5
capri	338	0.0	338	3	0	0.0	338	16	0	0.0
car4	10442	1.9	10442	4	0	2.0	10442	21	0	2.6
cari	681	0.1	681	3	0	0.3	681	18	0	0.5
cep1	1399	0.1	1399	3	0	0.1	1399	16	0	0.4
ch	10747	1.4	10747	3	0	1.5	10747	16	0	1.7
circ10-3	10910	10.2	10910	3	0	10.4	10910	17	0	11.3
co-100	783	1.2	783	3	0	2.3	783	16	0	6.8
co5	12067	3.2	12067	3	0	3.5	12067	17	0	4.0
co9	19820	11.6	19820	3	0	11.7	19820	17	0	13.2
complex	9592	2.4	9646	4	1	2.6	9646	20	1	2.8
cont1	40707	985.6	40707	3	0	1226.7	40707	18	0	1234.3
cont4	40802	2698.5	40802	3	0	2839.8	40802	18	0	2842.6
core2536-691	41182	16.0	41182	3	0	16.8	41182	16	0	16.7
core4872-1529	69516	69.1	69516	4	0	70.6	69516	19	0	70.9
cov1075	3270	0.5	3270	3	0	0.4	3270	18	0	0.6
cq5	12350	3.1	12350	3	0	3.1	12350	17	0	3.7
cq9	17826	7.1	17826	3	0	7.2	17826	17	0	8.2
cr42	999	0.1	999	3	0	0.1	999	16	0	0.4
cre-a	3555	0.3	3555	3	0	0.3	3555	16	0	0.6
cre-b	12712	5.0	12712	3	0	5.4	12712	16	0	6.6

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
cre-c	2842	0.3	2842	3	0	0.3	2842	16	0	0.5
cre-d	9271	3.5	9271	3	0	3.9	9271	16	0	5.0
crew1	1849	0.6	1849	3	0	0.4	1849	16	0	0.8
csched007	5522	0.7	5522	3	0	0.4	5522	17	0	0.6
csched008	3102	0.4	3102	3	0	0.4	3102	17	0	0.2
csched010	6832	0.8	6832	3	0	0.8	6832	17	0	0.7
cycle	920	0.1	920	3	0	0.1	920	17	0	0.1
czprob	1572	0.2	1572	3	0	0.1	1572	16	0	0.4
d10200	2852	0.6	2852	4	0	0.5	2852	19	0	0.8
d20200	9723	1.8	9723	4	0	1.9	9723	22	0	2.5
d2q06c	13920	3.9	13920	4	0	3.7	13920	18	0	4.2
d6cube	1179	0.4	1179	3	0	0.4	1179	16	0	0.4
dano3_3	46251	28.8	46251	3	0	29.0	46251	18	0	29.4
dano3_4	46251	28.9	46251	3	0	28.7	46251	18	0	29.5
dano3_5	46251	30.5	46251	3	0	29.0	46251	18	0	29.3
dano3mip	46251	28.8	46251	3	0	29.1	46251	18	0	29.4
danoint	2763	0.3	2763	3	0	0.3	2763	17	0	0.2
dbir1	14306	11.2	14306	3	0	11.4	14306	16	0	12.2
dbir2	11641	4.9	11641	3	0	5.3	11641	16	0	6.7
dc1c	19795	6.2	19795	4	0	6.2	19795	19	0	6.9
dc1l	24825	21.1	24825	3	0	21.4	24825	18	0	23.3
dcmulti	479	0.0	479	3	0	0.0	479	16	0	0.0
de063155	2546	0.3	2623	7	3	0.4	2623	20	3	0.4
de063157	31021	1.8	*145532883*	*5*	*0*	*7200.0*	*145916183*	*20*	*0*	*7200.0*
de080285	888	0.1	888	3	0	0.1	888	16	0	0.2
degen2	1325	0.1	1325	3	0	0.1	1325	16	0	0.2
degen3	5832	0.9	5832	3	0	0.8	5832	16	0	0.9
delf000	1675	0.4	1675	3	0	0.5	1675	16	0	0.8
delf001	1703	0.4	1703	3	0	0.5	1703	16	0	0.7
delf002	2009	0.4	2009	3	0	0.2	2009	16	0	0.8
delf003	3118	0.6	3180	3	1	0.8	3180	16	1	0.7
delf004	2624	0.6	2704	3	1	0.3	2704	16	1	0.6
delf005	3226	0.7	3294	3	1	0.8	3294	16	1	1.1
delf006	3066	0.6	3125	3	1	0.3	3125	16	1	0.8
delf007	2757	0.5	2826	3	1	0.3	2826	16	1	0.8
delf008	3393	0.7	3493	3	2	0.6	3493	17	2	0.8
delf009	3307	0.6	3383	3	1	0.8	3383	16	1	0.7
delf010	3132	0.7	3209	3	2	0.4	3209	16	2	0.9
delf011	3030	0.6	3104	3	2	0.4	3104	16	2	0.7
delf012	2890	0.6	2963	3	1	0.5	2963	16	1	0.6
delf013	3103	0.6	3184	3	1	0.5	3184	16	1	1.0
delf014	4257	0.8	4322	3	1	0.9	4322	16	1	0.8
delf015	3176	0.7	3242	3	1	0.4	3242	16	1	0.9
delf017	3075	0.5	3131	3	1	0.4	3131	17	1	0.8
delf018	3281	0.5	3335	3	1	0.4	3335	16	1	0.6
delf019	3160	0.6	3160	3	0	0.6	3160	16	0	0.7
delf020	3701	0.6	3784	3	2	0.6	3784	17	2	0.9
delf021	3247	0.5	3331	3	1	0.3	3331	16	1	0.7
delf022	3673	0.6	3757	3	2	0.6	3757	17	2	0.9
delf023	4603	0.8	4747	3	2	0.6	4747	17	2	0.8
delf024	3973	0.8	4150	3	1	0.7	4150	16	1	0.8
delf025	4032	0.7	4152	3	2	0.8	4152	16	2	1.0
delf026	3378	0.8	3495	3	1	0.5	3495	16	1	1.0
delf027	3402	0.8	3519	3	2	0.5	3519	16	2	0.7
delf028	3269	0.6	3425	3	2	0.4	3425	17	2	1.1

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
delf029	2949	0.6	3085	3	1	0.5	3085	16	1	0.7
delf030	3002	0.6	3135	3	1	0.6	3135	16	1	0.9
delf031	2835	0.6	2969	3	1	0.5	2969	16	1	0.7
delf032	2958	0.5	3091	3	1	0.6	3091	16	1	0.7
delf033	2210	0.4	2344	3	1	0.3	2344	16	1	0.5
delf034	2699	0.6	2832	3	1	0.4	2832	16	1	0.7
delf035	2407	0.5	2545	3	2	0.4	2545	16	2	0.6
delf036	2522	0.5	2655	3	2	0.4	2655	16	2	1.0
deter0	3725	0.2	3725	3	0	0.2	3725	16	0	0.5
deter1	9022	0.6	9022	3	0	0.8	9022	16	0	1.2
deter2	11017	0.6	11017	3	0	0.8	11017	16	0	1.3
deter3	12348	0.8	12348	3	0	1.0	12348	16	0	1.6
deter4	5633	0.3	5633	3	0	0.2	5633	16	0	0.6
deter5	9277	0.6	9277	3	0	0.8	9277	16	0	1.3
deter6	7482	0.4	7482	3	0	0.3	7482	16	0	1.0
deter7	11149	0.8	11149	3	0	0.8	11149	16	0	1.6
deter8	6977	0.4	6977	3	0	0.3	6977	16	0	0.9
df2177	1391	0.7	1391	3	0	0.7	1391	18	0	1.0
dfl001	30478	18.5	30478	3	0	18.6	30478	18	0	18.9
dfn-gwin-UUM	373	0.0	373	3	0	0.0	373	15	0	0.0
dg012142	11646	2.0	11646	3	0	1.9	11646	18	0	2.0
disctom	13965	1.7	13965	3	0	1.7	13965	17	0	1.8
disp3	490	0.0	490	3	0	0.0	490	16	0	0.1
dolom1	19382	7.3	19382	4	0	7.5	19382	19	0	8.3
ds	13089	17.8	13089	3	0	19.0	13089	18	0	25.0
ds-big	44732	604.9	44732	4	0	610.6	44732	21	0	638.7
dsbmip	2179	0.1	2179	3	0	0.2	2179	17	0	0.3
e18	12395	4.6	12395	3	0	4.8	12395	16	0	5.3
e226	402	0.0	402	3	0	0.0	402	16	0	0.1
egout	96	0.0	96	3	0	0.0	96	15	0	0.0
eil33-2	308	0.1	308	3	0	0.2	308	16	0	0.7
eilA101-2	5338	20.6	5338	3	0	21.9	5338	17	0	28.7
eilB101	1459	0.3	1459	3	0	0.4	1459	16	0	0.6
enigma	44	0.0	44	3	0	0.0	44	15	0	0.0
enlight13	0	0.0	0	0	0	0.0	0	0	0	0.0
enlight14	0	0.0	0	0	0	0.0	0	0	0	0.0
enlight15	0	0.0	0	0	0	0.0	0	0	0	0.0
enlight16	0	0.0	0	0	0	0.0	0	0	0	0.0
enlight9	0	0.0	0	0	0	0.0	0	0	0	0.0
etamacro	766	0.0	771	4	1	0.1	771	16	1	0.1
ex10	115687	1791.6	115687	3	0	1799.0	115687	18	0	1797.9
ex1010-pi	19385	10.0	19385	3	0	10.1	19385	18	0	10.8
ex3sta1	7713	4.8	7713	4	0	5.0	7713	19	0	6.2
ex9	57559	349.2	57559	3	0	348.8	57559	18	0	350.3
f2000	40611	60.6	40611	3	0	60.2	40611	19	0	61.4
farm	0	0.0	0	0	0	0.0	0	0	0	0.0
fast0507	13213	11.4	13213	3	0	11.9	13213	17	0	13.5
fffff800	855	0.1	855	3	0	0.0	855	16	0	0.1
fiball	2818	0.4	2818	3	0	0.7	2818	16	0	1.5
fiber	278	0.0	278	3	0	0.0	278	16	0	0.1
finnis	524	0.0	524	3	0	0.0	524	16	0	0.1
fit1d	1006	0.1	1006	3	0	0.0	1006	16	0	0.2
fit1p	3573	0.4	3573	3	0	0.4	3573	18	0	0.5
fit2d	10781	1.9	10781	3	0	2.1	10781	16	0	2.8
fit2p	16070	3.8	16070	3	0	4.0	16070	16	0	4.3

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
fixnet6	184	0.0	184	3	0	0.0	184	15	0	0.0
flugpl	11	0.0	11	3	0	0.0	11	16	0	0.0
fome11	45759	38.5	46151	8	1	39.9	46151	38	1	41.0
fome12	93828	108.4	95445	8	1	115.1	95445	38	1	118.8
fome13	175861	309.8	177979	8	1	321.1	177979	39	1	327.6
fome20	37294	13.1	37294	3	0	13.3	37294	15	0	14.4
fome21	81051	77.4	81051	3	0	76.9	81051	16	0	78.9
forplan	369	0.0	369	3	0	0.0	369	18	0	0.1
fxm2-16	6817	0.6	6817	3	0	0.8	6817	17	0	1.0
fxm2-6	2277	0.1	2277	3	0	0.1	2277	17	0	0.4
fxm3_16	48534	22.2	48534	3	0	22.7	48534	17	0	27.1
fxm3_6	9986	0.8	9986	3	0	1.1	9986	17	0	1.5
fxm4_6	25805	4.9	25805	3	0	5.7	25805	17	0	8.5
g200x740i	721	0.0	721	3	0	0.0	721	15	0	0.1
gams10a	38	0.0	44	4	1	0.0	44	16	1	0.0
gams30a	146	0.0	184	4	1	0.0	184	17	1	0.0
ganges	1291	0.1	1291	3	0	0.1	1291	18	0	0.3
ge	11171	2.4	11172	4	1	2.5	11172	19	1	3.3
gen	384	0.0	384	3	0	0.0	384	16	0	0.0
gen1	11914	12.1	12282	4	1	12.4	12282	20	1	13.2
gen2	12239	25.4	12239	4	0	26.4	12239	19	0	27.9
gen4	14397	56.3	14823	4	1	57.2	14823	20	1	59.1
ger50_17_trans	4819	1.3	4819	3	0	1.6	4819	15	0	2.6
germanrr	8775	2.7	8775	3	0	2.8	8775	16	0	3.6
germany50-DBM	8399	1.2	8399	3	0	0.9	8399	15	0	0.9
gesa2	1118	0.0	1118	3	0	0.0	1118	16	0	0.1
gesa2-o	653	0.0	653	3	0	0.0	653	16	0	0.1
gesa2_o	653	0.0	653	3	0	0.0	653	16	0	0.1
gesa3	974	0.1	974	3	0	0.1	974	16	0	0.2
gesa3_o	530	0.0	530	3	0	0.0	530	16	0	0.1
gfrd-pnc	664	0.0	664	3	0	0.1	664	16	0	0.1
glass4	73	0.0	73	3	0	0.0	73	15	0	0.0
gmu-35-40	316	0.0	316	3	0	0.0	316	16	0	0.1
gmu-35-50	359	0.0	359	3	0	0.1	359	16	0	0.1
gmut-75-50	6042	8.1	6042	3	0	8.5	6042	16	0	10.7
gmut-77-40	4047	1.7	4047	3	0	2.0	4047	16	0	2.6
go19	2606	0.2	2606	3	0	0.1	2606	17	0	0.3
gr4x6	36	0.0	36	3	0	0.0	36	15	0	0.0
greenbea	18382	4.7	18382	4	0	4.8	18382	23	0	5.0
greenbeb	11957	2.5	11957	4	0	2.5	11957	21	0	2.7
grow15	2102	0.2	2102	3	0	0.1	2102	16	0	0.2
grow22	3334	0.4	3334	3	0	0.2	3334	17	0	0.6
grow7	1071	0.1	1071	3	0	0.0	1071	16	0	0.1
gt2	19	0.0	19	3	0	0.0	19	15	0	0.0
hanoi5	7389	1.6	7389	3	0	1.7	7389	15	0	1.6
haprp	1303	0.0	1303	3	0	0.0	1303	16	0	0.1
harp2	323	0.0	323	3	0	0.0	323	16	0	0.1
i_n13	948541	2586.7	948541	1	0	2571.1	948541	1	0	2573.6
ic97_potential	339	0.0	339	3	0	0.0	339	15	0	0.0
iiasa	1562	0.1	1562	3	0	0.1	1562	16	0	0.2
iis-100-0-cov	1279	0.5	1279	3	0	0.5	1279	18	0	0.7
iis-bupa-cov	4196	1.1	4196	3	0	1.1	4196	17	0	1.1
iis-pima-cov	3532	1.2	3532	3	0	1.1	3532	16	0	1.5
israel	149	0.0	149	3	0	0.0	149	16	0	0.1
ivu06-big	37753	3149.6	37753	3	0	3170.1	37753	19	0	3335.5

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
ivu52	18983	107.6	18983	3	0	110.3	18983	17	0	124.0
janos-us-DDM	1042	0.0	1042	0	0	0.0	1042	0	0	0.0
jendrec1	11116	2.4	11116	3	0	2.8	11116	17	0	4.3
k16x240	39	0.0	39	3	0	0.0	39	15	0	0.0
kb2	58	0.0	58	3	0	0.0	58	17	0	0.0
ken-07	2777	0.1	2777	3	0	0.2	2777	16	0	0.5
ken-11	17739	2.7	17739	3	0	3.0	17739	16	0	3.8
ken-13	41646	20.1	41646	3	0	20.5	41646	16	0	22.0
ken-18	192333	500.8	192333	3	0	501.6	192333	16	0	505.9
kent	1680	0.3	1680	3	0	0.5	1680	16	0	1.2
khb05250	119	0.0	119	3	0	0.0	119	15	0	0.0
kl02	263	0.2	263	3	0	0.4	263	15	0	0.6
kleemin3	0	0.0	0	0	0	0.0	0	0	0	0.0
kleemin4	0	0.0	0	0	0	0.0	0	0	0	0.0
kleemin5	0	0.0	0	0	0	0.0	0	0	0	0.0
kleemin6	0	0.0	0	0	0	0.0	0	0	0	0.0
kleemin7	0	0.0	0	0	0	0.0	0	0	0	0.0
kleemin8	0	0.0	0	0	0	0.0	0	0	0	0.0
l152lav	700	0.1	700	3	0	0.1	700	16	0	0.2
l30	1379080	3042.6	*3304695*	*6*	*0*	*7200.0*	*3304140*	*34*	*0*	*7200.0*
l9	746	0.1	746	3	0	0.1	746	17	0	0.3
large000	3748	0.6	3748	3	0	0.7	3748	16	0	1.2
large001	7844	1.4	7844	3	0	1.4	7844	17	0	1.6
large002	3866	0.8	4065	3	2	0.6	4065	16	2	1.5
large003	4179	0.8	4254	3	1	0.7	4254	17	1	1.2
large004	4647	1.0	4681	4	2	0.8	4681	17	2	1.4
large005	4501	0.8	4567	3	1	0.8	4567	16	1	0.9
large006	4864	0.8	4942	3	2	0.7	4942	17	2	1.2
large007	5010	0.8	5094	3	1	0.9	5094	17	1	1.3
large008	5203	0.8	5291	3	1	0.9	5291	16	1	1.2
large009	4988	0.9	5074	3	1	1.1	5074	16	1	1.2
large010	4639	0.7	4725	3	1	0.8	4725	16	1	1.2
large011	5135	0.9	5221	3	1	0.9	5221	16	1	1.0
large012	4924	0.8	5009	3	1	0.9	5009	16	1	1.0
large013	4975	0.8	5062	3	1	0.7	5062	16	1	1.2
large014	5082	0.9	5148	3	1	0.9	5148	16	1	1.1
large015	4318	0.7	4380	3	1	0.6	4380	16	1	0.9
large016	4571	0.6	4633	3	1	0.8	4633	16	1	0.9
large017	3980	0.8	3980	3	0	0.8	3980	16	0	0.9
large018	4459	0.8	4459	3	0	0.8	4459	16	0	1.0
large019	4909	0.8	4909	3	0	0.8	4909	16	0	1.1
large020	6984	1.1	7059	3	2	0.9	7059	17	2	1.5
large021	6201	0.9	6288	3	2	0.9	6288	16	2	1.5
large022	6907	0.9	6993	3	2	0.8	6993	16	2	1.4
large023	4224	0.9	4398	3	2	0.7	4398	17	2	1.1
large024	5788	1.0	5988	4	2	1.2	5988	17	2	1.6
large025	4811	0.8	5000	3	2	1.0	5000	16	2	1.2
large026	4196	0.8	4367	3	2	0.7	4367	17	2	1.1
large027	4172	0.8	4353	3	2	0.8	4353	16	2	1.0
large028	4691	1.1	4906	3	1	0.8	4906	16	1	1.4
large029	4158	0.8	4372	4	2	0.6	4372	17	2	1.2
large030	3732	0.6	3930	3	2	0.7	3930	16	2	1.2
large031	3729	0.6	3931	3	1	0.6	3931	16	1	1.2
large032	4851	0.9	5052	3	1	0.9	5052	16	1	1.1
large033	3675	0.7	3877	3	2	0.5	3877	16	2	1.2

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
large034	4009	0.7	4201	3	2	0.7	4201	16	2	0.9
large035	3450	0.8	3655	3	1	0.7	3655	17	1	1.1
large036	3111	0.6	3314	3	2	0.6	3314	17	2	1.0
lectsched-1	7	1.1	7	3	0	1.3	7	15	0	1.9
lectsched-1-obj	963	1.2	963	3	0	1.5	963	15	0	2.1
lectsched-2	3	0.5	3	3	0	0.7	3	15	0	1.1
lectsched-3	7	0.9	7	3	0	1.1	7	15	0	1.6
lectsched-4-obj	174	0.3	174	3	0	0.2	174	15	0	0.7
leo1	862	0.3	862	3	0	0.5	862	16	0	0.9
leo2	1637	0.6	1637	4	0	0.9	1637	17	0	1.8
liu	543	0.1	543	3	0	0.1	543	15	0	0.1
lo10	953870	5941.4	953870	0	0	5931.8	953870	0	0	5940.8
long15	229488	2753.1	229488	0	0	2738.1	229488	0	0	2736.6
lotfi	226	0.0	226	3	0	0.0	226	15	0	0.0
lotsize	1460	0.0	1460	3	0	0.1	1460	15	0	0.2
lp22	38451	35.6	38451	3	0	35.8	38451	18	0	36.3
lpl1	36759	59.4	36759	3	0	60.0	36759	17	0	64.2
lpl2	1465	0.2	1465	3	0	0.1	1465	15	0	0.4
lpl3	5040	1.1	5040	3	0	1.2	5040	15	0	1.4
lrn	11450	2.6	11452	4	1	2.5	11452	17	1	3.0
lrsa120	9787	2.5	9789	3	1	2.4	9789	17	1	2.8
lseu	25	0.0	25	3	0	0.0	25	15	0	0.0
m100n500k4r1	174	0.0	174	3	0	0.0	174	17	0	0.0
macrophage	706	0.0	706	0	0	0.0	706	0	0	0.0
manna81	3018	0.1	3018	0	0	0.1	3018	0	0	0.1
map06	23840	39.4	23840	3	0	40.6	23840	16	0	44.4
map10	23747	40.6	23747	3	0	41.5	23747	16	0	45.2
map14	23178	38.3	23178	3	0	39.4	23178	16	0	42.9
map18	20964	33.6	20964	3	0	34.7	20964	16	0	37.8
map20	19686	33.1	19686	3	0	30.9	19686	15	0	33.8
markshare1	35	0.0	35	3	0	0.0	35	16	0	0.0
markshare2	43	0.0	43	3	0	0.0	43	16	0	0.0
markshare_5_0	24	0.0	24	3	0	0.0	24	16	0	0.0
maros	1255	0.1	1255	4	0	0.2	1255	19	0	0.4
maros-r7	7953	2.9	7953	3	0	3.5	7953	16	0	4.6
mas74	224	0.0	224	3	0	0.0	224	17	0	0.0
mas76	132	0.0	132	3	0	0.0	132	16	0	0.0
maxgasflow	6737	0.6	6737	3	0	0.8	6737	16	0	1.1
mc11	1239	0.1	1239	3	0	0.1	1239	15	0	0.3
mcf2	2763	0.3	2763	3	0	0.1	2763	17	0	0.4
mcsched	2546	0.3	2546	3	0	0.3	2546	17	0	0.5
methanosarcina	655	0.1	655	0	0	0.2	655	0	0	0.2
mik-250-1-100-1	100	0.0	100	3	0	0.0	100	15	0	0.1
mine-166-5	1642	0.5	1642	3	0	0.5	1642	16	0	0.8
mine-90-10	1948	0.5	1948	3	0	0.4	1948	16	0	0.4
misc03	45	0.0	45	0	0	0.0	45	0	0	0.0
misc06	816	0.1	816	3	0	0.1	816	16	0	0.2
misc07	157	0.0	157	3	0	0.0	157	15	0	0.1
mitre	2451	0.3	2451	3	0	0.4	2451	17	0	0.7
mkc	538	0.1	538	3	0	0.1	538	16	0	0.1
mkc1	538	0.1	538	3	0	0.0	538	16	0	0.1
mod008	27	0.0	27	3	0	0.0	27	15	0	0.0
mod010	1062	0.1	1062	3	0	0.2	1062	15	0	0.4
mod011	6153	0.9	6153	3	0	0.8	6153	17	0	1.0
mod2	58340	100.4	58534	14	11	103.8	58534	26	11	104.2

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
model1	180	0.0	182	3	1	0.0	182	16	1	0.0
model10	44687	22.8	44687	3	0	23.2	44687	17	0	24.5
model11	5273	1.0	5273	3	0	1.0	5273	17	0	1.2
model2	3466	0.4	3466	3	0	0.2	3466	18	0	0.5
model3	9208	1.3	9208	3	0	1.3	9208	18	0	1.7
model4	15098	3.0	15098	3	0	2.8	15098	18	0	3.1
model5	19877	3.8	19877	3	0	4.1	19877	17	0	4.7
model6	14770	3.3	14770	4	0	3.2	14770	19	0	3.5
model7	16070	4.0	16070	3	0	4.0	16070	18	0	4.6
model8	2522	0.3	2538	3	1	0.4	2538	16	1	0.7
model9	12397	2.3	12397	4	0	2.5	12397	20	0	3.2
modglob	359	0.0	359	3	0	0.0	359	15	0	0.0
modszk1	653	0.0	653	3	0	0.0	653	17	0	0.1
momentum1	3305	0.8	3306	4	1	1.1	3306	17	1	1.9
momentum2	45882	18.8	45950	5	2	19.7	45950	18	2	21.9
momentum3	46184	189.5	46185	4	1	192.0	46185	18	1	198.4
msc98-ip	9496	1.3	9549	4	1	1.7	9549	16	1	2.2
mspp16	52	19.3	52	3	0	20.0	52	15	0	25.1
multi	59	0.0	59	3	0	0.0	59	16	0	0.0
mzzv11	37921	26.4	37921	3	0	26.2	37921	18	0	26.6
mzzv42z	34373	17.6	34373	3	0	17.2	34373	17	0	17.5
n15-3	43662	114.6	43662	3	0	115.3	43662	16	0	118.7
n3-3	2965	0.6	2965	3	0	0.5	2965	15	0	0.9
n3700	8698	1.6	8698	3	0	1.6	8698	15	0	1.6
n3701	8106	1.3	8106	3	0	1.3	8106	15	0	1.5
n3702	7987	1.2	7987	3	0	1.1	7987	15	0	1.5
n3703	7397	1.1	7397	3	0	1.0	7397	15	0	1.4
n3704	7325	1.1	7325	3	0	0.9	7325	15	0	1.4
n3705	6974	1.3	6974	3	0	1.2	6974	16	0	1.4
n3706	7305	1.1	7305	3	0	1.1	7305	15	0	1.4
n3707	8681	1.2	8681	3	0	1.4	8681	15	0	1.6
n3708	7133	1.6	7133	3	0	1.2	7133	15	0	1.7
n3709	8030	1.5	8030	3	0	1.4	8030	15	0	1.8
n370a	8608	1.2	8608	3	0	1.3	8608	15	0	1.3
n370b	8553	1.4	8553	3	0	1.2	8553	15	0	1.6
n370c	7273	1.1	7273	3	0	0.9	7273	15	0	1.4
n370d	7273	1.1	7273	3	0	0.9	7273	15	0	1.4
n370e	7852	1.2	7852	3	0	1.0	7852	15	0	1.3
n3div36	306	0.4	306	3	0	0.7	306	16	0	1.9
n3seq24	4646	14.1	4720	4	1	15.9	4720	18	1	20.4
n4-3	1341	0.2	1341	3	0	0.2	1341	15	0	0.1
n9-3	3531	0.6	3531	3	0	0.5	3531	16	0	0.9
nag	2476	0.2	2476	3	0	0.1	2476	15	0	0.3
nemsafm	755	0.0	755	3	0	0.0	755	16	0	0.1
nemscem	932	0.1	932	3	0	0.1	932	17	0	0.2
nemsemm1	10608	3.6	10608	3	0	5.4	10608	16	0	12.7
nemsemm2	10445	1.5	10445	3	0	1.9	10445	16	0	3.8
nemspmm1	15049	4.3	15049	4	0	4.5	15049	21	0	5.4
nemspmm2	15056	4.2	15058	4	1	4.4	15058	19	1	5.4
nemswrld	35922	32.6	35922	4	0	33.7	35922	20	0	35.3
neos	89942	497.3	90253	4	1	497.8	90253	17	1	500.9
neos-1053234	257	0.1	257	3	0	0.0	257	16	0	0.1
neos-1053591	821	0.0	821	3	0	0.0	821	16	0	0.1
neos-1056905	140	0.0	140	3	0	0.0	140	15	0	0.0
neos-1058477	48	0.0	48	3	0	0.0	48	15	0	0.1

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
neos-1061020	16447	7.3	16447	3	0	7.5	16447	17	0	8.1
neos-1062641	873	0.0	902	3	1	0.0	902	16	1	0.1
neos-1067731	12132	2.4	12132	3	0	2.5	12132	16	0	2.8
neos-1096528	115	47.5	115	3	0	54.0	115	15	0	56.4
neos-1109824	115	0.5	115	3	0	0.6	115	15	0	0.5
neos-1112782	622	0.1	622	3	0	0.0	622	16	0	0.1
neos-1112787	557	0.1	557	3	0	0.0	557	16	0	0.1
neos-1120495	96	0.2	96	0	0	0.3	96	0	0	0.2
neos-1121679	35	0.0	35	3	0	0.0	35	16	0	0.0
neos-1122047	5499	1.2	5499	3	0	1.5	5499	16	0	3.0
neos-1126860	5216	0.9	5216	3	0	1.0	5216	16	0	1.9
neos-1140050	13327	30.1	13327	5	0	32.6	13327	29	0	38.6
neos-1151496	4930	0.8	4930	3	0	0.8	4930	17	0	0.6
neos-1171448	4075	0.6	4075	1	0	0.5	4075	1	0	0.5
neos-1171692	1290	0.1	1290	3	0	0.1	1290	16	0	0.1
neos-1171737	2060	0.3	2060	3	0	0.2	2060	15	0	0.3
neos-1173026	33	0.0	33	3	0	0.0	33	15	0	0.0
neos-1200887	294	0.0	294	3	0	0.0	294	15	0	0.0
neos-1208069	3008	0.6	3008	3	0	0.6	3008	16	0	0.6
neos-1208135	2364	0.4	2364	3	0	0.4	2364	17	0	0.2
neos-1211578	187	0.0	187	3	0	0.0	187	15	0	0.0
neos-1215259	6319	1.0	6319	3	0	1.0	6319	16	0	1.2
neos-1215891	3541	0.9	3541	3	0	0.7	3541	17	0	1.1
neos-1223462	13116	3.7	13116	3	0	3.6	13116	16	0	4.0
neos-1224597	10673	1.8	10673	3	0	1.5	10673	15	0	1.7
neos-1225589	438	0.0	438	3	0	0.0	438	15	0	0.1
neos-1228986	197	0.0	197	3	0	0.0	197	15	0	0.0
neos-1281048	2276	0.2	2276	3	0	0.3	2276	15	0	0.3
neos-1311124	639	0.0	639	3	0	0.0	639	15	0	0.1
neos-1324574	6564	0.8	6564	3	0	0.9	6564	16	0	0.9
neos-1330346	1983	0.3	1983	3	0	0.1	1983	16	0	0.2
neos-1330635	141	0.0	141	3	0	0.0	141	16	0	0.1
neos-1337307	10964	2.1	10964	3	0	2.2	10964	16	0	2.5
neos-1337489	187	0.0	187	3	0	0.0	187	15	0	0.0
neos-1346382	350	0.0	350	3	0	0.0	350	15	0	0.0
neos-1354092	12518	7.0	12518	3	0	7.1	12518	18	0	8.2
neos-1367061	18400	23.5	18400	4	0	24.1	18400	18	0	25.9
neos-1396125	4834	0.6	4834	3	0	0.7	4834	16	0	0.7
neos-1407044	27160	31.7	27160	3	0	32.1	27160	19	0	33.5
neos-1413153	1027	0.2	1027	3	0	0.2	1027	16	0	0.3
neos-1415183	1763	0.3	1763	3	0	0.4	1763	16	0	0.2
neos-1417043	13642	57.3	13642	0	0	57.2	13642	0	0	57.2
neos-1420205	944	0.0	944	3	0	0.0	944	15	0	0.0
neos-1420546	102727	92.3	102727	3	0	92.2	102727	17	0	93.4
neos-1420790	12930	2.5	12930	3	0	2.4	12930	17	0	2.3
neos-1423785	19268	3.1	19268	3	0	3.6	19268	15	0	4.2
neos-1425699	26	0.0	26	3	0	0.0	26	15	0	0.0
neos-1426635	350	0.0	350	3	0	0.0	350	15	0	0.0
neos-1426662	665	0.0	665	3	0	0.0	665	16	0	0.0
neos-1427181	591	0.0	591	0	0	0.0	591	0	0	0.0
neos-1427261	883	0.1	883	3	0	0.1	883	15	0	0.1
neos-1429185	466	0.0	466	3	0	0.0	466	15	0	0.1
neos-1429212	28835	220.5	28835	3	0	218.8	28835	16	0	223.0
neos-1429461	399	0.0	399	3	0	0.0	399	16	0	0.1
neos-1430701	257	0.0	257	3	0	0.0	257	15	0	0.0

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
neos-1430811	32707	340.4	32707	3	0	338.3	32707	17	0	343.2
neos-1436709	505	0.0	505	3	0	0.0	505	15	0	0.1
neos-1436713	835	0.0	835	3	0	0.1	835	15	0	0.1
neos-1437164	296	0.0	296	3	0	0.1	296	16	0	0.2
neos-1439395	312	0.0	312	3	0	0.0	312	16	0	0.1
neos-1440225	486	0.1	486	3	0	0.1	486	17	0	0.1
neos-1440447	239	0.0	239	3	0	0.0	239	15	0	0.0
neos-1440457	820	0.0	820	3	0	0.1	820	15	0	0.1
neos-1440460	359	0.0	359	3	0	0.0	359	16	0	0.1
neos-1441553	288	0.0	288	3	0	0.1	288	16	0	0.1
neos-1442119	607	0.0	607	3	0	0.1	607	15	0	0.1
neos-1442657	498	0.0	498	3	0	0.0	498	15	0	0.1
neos-1445532	14103	1.7	14103	3	0	1.8	14103	16	0	1.9
neos-1445738	15164	2.8	15164	3	0	2.9	15164	16	0	3.0
neos-1445743	16103	3.2	16103	3	0	3.1	16103	16	0	3.4
neos-1445755	15634	3.1	15634	3	0	3.2	15634	16	0	3.4
neos-1445765	15960	3.3	15960	3	0	3.4	15960	16	0	4.0
neos-1451294	10087	1.8	10087	3	0	1.6	10087	18	0	1.7
neos-1456979	957	0.4	957	3	0	0.2	957	15	0	0.5
neos-1460246	276	0.0	276	3	0	0.0	276	16	0	0.1
neos-1460265	808	0.1	808	3	0	0.1	808	16	0	0.1
neos-1460543	9199	1.3	9199	3	0	1.2	9199	18	0	1.6
neos-1460641	10168	1.1	10168	3	0	1.1	10168	16	0	1.1
neos-1461051	418	0.0	418	3	0	0.1	418	15	0	0.1
neos-1464762	10563	1.2	10563	3	0	1.2	10563	16	0	1.0
neos-1467067	673	0.0	673	3	0	0.0	673	15	0	0.1
neos-1467371	8604	1.1	8604	3	0	0.9	8604	16	0	1.0
neos-1467467	8764	0.9	8764	3	0	0.9	8764	16	0	1.0
neos-1480121	86	0.0	86	3	0	0.0	86	15	0	0.0
neos-1489999	835	0.1	835	3	0	0.1	835	16	0	0.1
neos-1516309	134	0.0	134	3	0	0.1	134	15	0	0.1
neos-1582420	2433	0.5	2433	3	0	0.7	2433	16	0	0.7
neos-1593097	921	0.6	921	3	0	0.7	921	17	0	1.3
neos-1595230	518	0.0	518	3	0	0.0	518	16	0	0.1
neos-1597104	150	0.5	150	3	0	0.7	150	15	0	1.7
neos-1599274	357	0.1	357	3	0	0.1	357	15	0	0.3
neos-1601936	14758	5.3	14758	3	0	5.4	14758	18	0	5.7
neos-1603512	814	0.1	814	3	0	0.1	814	15	0	0.2
neos-1603518	2362	0.4	2362	3	0	0.4	2362	16	0	0.6
neos-1603965	30552	4.3	*1109741*	*4*	*4*	*1033.8*	*1109741*	*16*	*16*	*1032.2*
neos-1605061	23399	11.5	23399	3	0	11.5	23399	19	0	11.7
neos-1605075	16980	8.8	16980	3	0	8.8	16980	18	0	9.1
neos-1616732	314	0.0	314	0	0	0.0	314	0	0	0.0
neos-1620770	772	0.1	772	0	0	0.1	772	0	0	0.0
neos-1620807	222	0.0	222	0	0	0.0	222	0	0	0.0
neos-1622252	807	0.1	807	0	0	0.1	807	0	0	0.1
neos-430149	240	0.0	240	3	0	0.0	240	15	0	0.1
neos-476283	5536	3.8	5536	3	0	4.3	5536	17	0	5.8
neos-480878	466	0.1	466	3	0	0.1	466	16	0	0.2
neos-494568	553	0.2	553	3	0	0.2	553	15	0	0.4
neos-495307	963	0.3	963	3	0	0.2	963	15	0	0.5
neos-498623	1034	0.3	1037	3	1	0.6	1037	16	1	0.7
neos-501453	37	0.0	37	3	0	0.0	37	16	0	0.0
neos-501474	288	0.0	288	3	0	0.0	288	16	0	0.0
neos-503737	4883	0.8	4883	3	0	0.8	4883	18	0	0.9

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
neos-504674	548	0.0	548	3	0	0.0	548	15	0	0.1
neos-504815	468	0.0	468	3	0	0.0	468	15	0	0.0
neos-506422	98	0.1	98	3	0	0.1	98	15	0	0.3
neos-506428	2092	1.0	2092	0	0	1.1	2092	0	0	1.0
neos-512201	543	0.0	543	3	0	0.0	543	15	0	0.0
neos-520729	32688	19.9	32688	3	0	20.2	32688	16	0	20.8
neos-522351	205	0.0	205	3	0	0.0	205	15	0	0.1
neos-525149	1124	0.7	1124	0	0	0.8	1124	0	0	0.8
neos-530627	68	0.0	68	3	0	0.0	68	15	0	0.0
neos-538867	169	0.0	171	3	1	0.0	171	15	1	0.1
neos-538916	168	0.0	168	3	0	0.0	168	15	0	0.1
neos-544324	2629	1.2	2629	3	0	1.5	2629	19	0	3.3
neos-547911	1895	0.3	1895	3	0	0.5	1895	17	0	0.9
neos-548047	11782	1.6	11782	3	0	1.8	11782	16	0	1.6
neos-548251	2037	0.1	2037	3	0	0.1	2037	15	0	0.1
neos-551991	7378	1.1	7378	3	0	1.1	7378	16	0	1.1
neos-555001	10198	1.0	10198	3	0	0.6	10198	16	0	0.9
neos-555298	3995	0.2	3995	3	0	0.1	3995	16	0	0.2
neos-555343	11043	1.1	11043	3	0	0.9	11043	16	0	1.3
neos-555424	5895	0.4	5895	3	0	0.2	5895	16	0	0.3
neos-555694	565	0.1	565	3	0	0.1	565	16	0	0.3
neos-555771	575	0.1	575	3	0	0.1	575	16	0	0.3
neos-555884	7183	0.6	7183	3	0	0.4	7183	16	0	0.8
neos-555927	2202	0.1	2202	3	0	0.1	2202	16	0	0.1
neos-565672	87785	259.8	87785	3	0	260.9	87785	16	0	264.9
neos-565815	6971	2.3	6971	3	0	2.4	6971	17	0	2.9
neos-570431	2355	0.2	2355	3	0	0.1	2355	16	0	0.3
neos-574665	506	0.1	506	3	0	0.1	506	16	0	0.3
neos-578379	14389	13.2	14389	3	0	13.4	14389	18	0	13.8
neos-582605	1067	0.1	1067	3	0	0.1	1067	15	0	0.1
neos-583731	602	0.0	602	0	0	0.0	602	0	0	0.0
neos-584146	603	0.1	603	3	0	0.1	603	15	0	0.1
neos-584851	612	0.0	612	3	0	0.1	612	16	0	0.1
neos-584866	11751	2.6	11751	3	0	2.8	11751	16	0	2.8
neos-585192	2614	0.5	2614	3	0	0.7	2614	17	0	0.7
neos-585467	1912	0.4	1912	3	0	0.2	1912	16	0	0.4
neos-593853	615	0.0	615	3	0	0.1	615	16	0	0.1
neos-595904	1276	0.2	1276	3	0	0.3	1276	16	0	0.3
neos-595905	390	0.0	390	3	0	0.1	390	16	0	0.1
neos-595925	521	0.0	521	3	0	0.1	521	16	0	0.1
neos-598183	1148	0.1	1148	3	0	0.1	1148	16	0	0.2
neos-603073	455	0.0	455	3	0	0.0	455	16	0	0.1
neos-611135	11914	4.7	11914	3	0	4.7	11914	16	0	5.1
neos-611838	2046	0.4	2046	3	0	0.2	2046	16	0	0.5
neos-612125	2002	0.4	2002	3	0	0.3	2002	16	0	0.7
neos-612143	2100	0.4	2100	3	0	0.3	2100	16	0	0.5
neos-612162	1828	0.3	1828	3	0	0.3	1828	16	0	0.7
neos-619167	6697	1.3	6697	3	0	1.1	6697	16	0	1.3
neos-631164	954	0.1	954	3	0	0.1	954	16	0	0.1
neos-631517	765	0.0	765	3	0	0.1	765	16	0	0.0
neos-631694	24940	5.2	24940	3	0	5.2	24940	16	0	5.2
neos-631709	70640	213.1	70640	3	0	213.0	70640	16	0	216.1
neos-631710	66455	923.9	66455	3	0	920.8	66455	16	0	922.3
neos-631784	26868	29.2	26868	3	0	29.5	26868	16	0	30.8
neos-632335	5592	1.0	5592	3	0	1.3	5592	16	0	1.7

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
neos-633273	5070	1.1	5070	3	0	1.1	5070	16	0	1.6
neos-641591	14066	6.7	14066	3	0	7.1	14066	16	0	7.6
neos-655508	119	0.1	119	0	0	0.1	119	0	0	0.0
neos-662469	14066	7.2	14066	3	0	7.0	14066	16	0	7.7
neos-686190	834	0.2	834	3	0	0.1	834	16	0	0.2
neos-691058	7206	1.3	7206	3	0	1.2	7206	17	0	1.4
neos-691073	6943	1.4	6943	3	0	1.3	6943	17	0	1.3
neos-693347	9600	2.1	9600	3	0	2.2	9600	17	0	2.5
neos-702280	12807	14.6	12807	3	0	15.3	12807	18	0	17.8
neos-709469	429	0.0	429	3	0	0.0	429	16	0	0.0
neos-717614	1049	0.0	1049	3	0	0.1	1049	16	0	0.2
neos-738098	25294	31.9	25294	3	0	31.8	25294	17	0	32.0
neos-775946	1720	0.6	1720	3	0	0.5	1720	16	0	0.6
neos-777800	1633	0.5	1633	3	0	0.3	1633	17	0	0.6
neos-780889	57556	184.5	57556	0	0	185.1	57556	0	0	185.1
neos-785899	411	0.1	411	3	0	0.1	411	16	0	0.1
neos-785912	1338	0.3	1338	3	0	0.3	1338	17	0	0.4
neos-785914	232	0.0	232	3	0	0.0	232	15	0	0.1
neos-787933	14008	31.7	14008	3	0	32.3	14008	15	0	47.2
neos-791021	10768	2.5	10768	3	0	2.4	10768	16	0	2.8
neos-796608	379	0.0	379	0	0	0.0	379	0	0	0.0
neos-799711	19400	3.0	19400	3	0	3.3	19400	16	0	4.6
neos-799838	11429	3.8	11429	3	0	3.8	11429	15	0	4.0
neos-801834	1782	0.3	1782	3	0	0.2	1782	16	0	0.6
neos-803219	1309	0.1	1309	3	0	0.1	1309	16	0	0.2
neos-803220	888	0.1	888	3	0	0.1	888	16	0	0.1
neos-806323	1246	0.1	1246	3	0	0.1	1246	16	0	0.2
neos-807454	12164	2.2	12164	3	0	2.3	12164	16	0	2.2
neos-807456	11272	1.6	11272	3	0	1.6	11272	17	0	1.3
neos-807639	1305	0.1	1305	3	0	0.1	1305	16	0	0.1
neos-807705	1259	0.1	1259	3	0	0.1	1259	17	0	0.3
neos-808072	8164	1.4	8164	3	0	1.4	8164	16	0	1.3
neos-808214	1236	0.2	1236	3	0	0.1	1236	17	0	0.2
neos-810286	13517	3.5	13517	3	0	3.5	13517	18	0	3.7
neos-810326	8031	1.2	8031	3	0	1.2	8031	17	0	1.3
neos-820146	249	0.0	249	3	0	0.0	249	15	0	0.0
neos-820157	569	0.0	569	3	0	0.0	569	16	0	0.1
neos-820879	1509	0.5	1509	3	0	0.7	1509	16	0	0.9
neos-824661	9912	5.2	9912	3	0	5.6	9912	16	0	5.6
neos-824695	4726	1.8	4726	3	0	1.8	4726	16	0	1.9
neos-825075	1062	0.1	1062	3	0	0.0	1062	17	0	0.1
neos-826224	5396	1.2	5396	3	0	1.4	5396	15	0	1.4
neos-826250	5747	1.3	5747	3	0	1.0	5747	16	0	1.1
neos-826650	8372	1.7	8372	3	0	1.9	8372	17	0	1.6
neos-826694	10754	2.6	10754	3	0	2.7	10754	16	0	2.8
neos-826812	10450	2.4	10450	3	0	2.4	10450	16	0	2.5
neos-826841	3738	0.8	3738	3	0	0.7	3738	15	0	0.8
neos-827015	18897	12.2	18897	3	0	12.8	18897	16	0	14.7
neos-827175	11439	3.2	11439	3	0	3.3	11439	16	0	3.8
neos-829552	13050	4.4	13050	3	0	4.8	13050	16	0	5.7
neos-830439	179	0.0	179	3	0	0.0	179	15	0	0.1
neos-831188	8374	1.1	8374	3	0	1.2	8374	16	0	1.2
neos-839838	6451	1.1	6451	3	0	1.4	6451	16	0	1.6
neos-839859	2312	0.4	2312	3	0	0.2	2312	16	0	0.6
neos-839894	21287	20.6	21287	3	0	20.8	21287	17	0	21.8

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
neos-841664	5975	0.7	5975	3	0	0.6	5975	16	0	0.9
neos-847051	2383	0.2	2383	3	0	0.3	2383	16	0	0.4
neos-847302	2146	0.3	2146	3	0	0.4	2146	17	0	0.2
neos-848150	1409	0.2	1409	3	0	0.2	1409	17	0	0.3
neos-848198	11363	2.1	11363	3	0	2.3	11363	15	0	2.6
neos-848589	1401	1.0	1401	3	0	1.9	1401	16	0	4.8
neos-848845	7295	1.4	7295	3	0	1.4	7295	17	0	1.5
neos-849702	5626	1.0	5626	3	0	1.0	5626	17	0	1.1
neos-850681	25783	5.9	25783	3	0	5.8	25783	17	0	6.2
neos-856059	580	0.1	580	0	0	0.1	580	0	0	0.1
neos-859770	294	0.3	294	3	0	0.3	294	16	0	0.6
neos-860244	90	0.1	90	3	0	0.1	90	15	0	0.3
neos-860300	349	0.2	349	3	0	0.3	349	16	0	0.8
neos-862348	1026	0.2	1026	3	0	0.2	1026	16	0	0.4
neos-863472	142	0.0	142	3	0	0.0	142	15	0	0.0
neos-872648	38712	176.3	38712	3	0	175.4	38712	15	0	180.9
neos-873061	30803	140.9	30803	3	0	142.5	30803	15	0	146.5
neos-876808	92322	98.6	92322	3	0	98.4	92322	16	0	100.0
neos-880324	232	0.0	232	3	0	0.0	232	16	0	0.0
neos-881765	424	0.0	424	3	0	0.1	424	16	0	0.0
neos-885086	3951	0.6	3951	3	0	0.9	3951	16	0	1.1
neos-885524	205	5.4	205	3	0	5.6	205	16	0	7.1
neos-886822	1834	0.3	1834	3	0	0.2	1834	18	0	0.4
neos-892255	1082	0.2	1082	3	0	0.2	1082	16	0	0.3
neos-905856	1228	0.1	1228	3	0	0.2	1228	16	0	0.2
neos-906865	792	0.1	792	3	0	0.1	792	16	0	0.1
neos-911880	319	0.0	319	3	0	0.0	319	16	0	0.1
neos-911970	258	0.0	258	3	0	0.0	258	16	0	0.1
neos-912015	1001	0.1	1001	3	0	0.2	1001	17	0	0.2
neos-912023	1095	0.1	1095	3	0	0.2	1095	16	0	0.3
neos-913984	4692	2.5	4692	3	0	2.8	4692	16	0	3.0
neos-914441	7787	1.9	7787	3	0	1.6	7787	16	0	2.0
neos-916173	862	0.2	862	3	0	0.3	862	17	0	1.2
neos-916792	1165	0.2	1165	3	0	0.6	1165	19	0	2.0
neos-930752	19648	6.0	19648	3	0	6.1	19648	17	0	6.1
neos-931517	10863	2.1	10863	3	0	2.2	10863	15	0	2.1
neos-931538	11171	2.4	11171	3	0	2.2	11171	16	0	2.6
neos-932721	30030	11.8	30030	3	0	11.7	30030	17	0	12.4
neos-932816	14154	4.1	14154	3	0	4.2	14154	17	0	5.0
neos-933364	1547	0.1	1547	3	0	0.1	1547	15	0	0.1
neos-933550	2301	0.5	2301	3	0	0.4	2301	16	0	0.6
neos-933562	4067	1.2	4067	3	0	1.1	4067	17	0	1.2
neos-933638	27131	18.3	27131	3	0	18.0	27131	17	0	18.8
neos-933815	1147	0.1	1147	3	0	0.0	1147	15	0	0.1
neos-933966	21146	11.1	21146	3	0	11.4	21146	17	0	12.2
neos-934184	1547	0.1	1547	3	0	0.1	1547	15	0	0.1
neos-934278	26638	16.5	26638	3	0	16.6	26638	17	0	17.2
neos-934441	29117	18.7	29117	3	0	18.6	29117	17	0	19.1
neos-934531	549	0.3	549	3	0	0.5	549	15	0	0.9
neos-935234	27636	16.9	27636	3	0	17.0	27636	17	0	17.4
neos-935348	28978	17.0	28978	3	0	16.8	28978	17	0	17.0
neos-935496	4000	1.1	4000	3	0	0.8	4000	17	0	1.3
neos-935627	30839	18.9	30839	3	0	18.6	30839	17	0	20.0
neos-935674	4015	1.1	4015	3	0	1.0	4015	17	0	1.1
neos-935769	24339	12.2	24339	3	0	12.5	24339	17	0	12.5

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
neos-936660	26498	14.6	26498	3	0	14.7	26498	17	0	15.1
neos-937446	26033	15.3	26033	3	0	15.2	26033	16	0	15.5
neos-937511	24828	13.7	24828	3	0	13.5	24828	16	0	14.0
neos-937815	33788	23.8	33788	3	0	23.8	33788	17	0	24.2
neos-941262	26859	15.1	26859	3	0	15.1	26859	18	0	15.6
neos-941313	46102	110.4	46102	3	0	112.8	46102	18	0	114.9
neos-941698	682	0.1	682	3	0	0.1	682	16	0	0.1
neos-941717	4805	0.7	4805	3	0	0.8	4805	17	0	0.7
neos-941782	1858	0.3	1858	3	0	0.2	1858	17	0	0.3
neos-942323	468	0.1	468	3	0	0.0	468	16	0	0.1
neos-942830	1444	0.2	1444	3	0	0.1	1444	16	0	0.2
neos-942886	336	0.0	336	3	0	0.1	336	16	0	0.1
neos-948126	30978	19.4	30978	3	0	19.5	30978	17	0	20.0
neos-948268	11614	3.8	11614	3	0	3.6	11614	17	0	3.9
neos-948346	4225	4.1	4225	3	0	4.4	4225	17	0	6.2
neos-950242	1897	0.4	1897	0	0	0.6	1897	0	0	0.5
neos-952987	531	0.3	531	3	0	0.6	531	16	0	1.3
neos-953928	1858	1.1	5020	4	1	2.8	5020	17	1	3.5
neos-954925	7761	14.9	*2098551*	*1*	*1*	*7200.0*	*2114187*	*1*	*1*	*7200.0*
neos-955215	850	0.0	850	3	0	0.0	850	15	0	0.1
neos-955800	882	0.1	882	0	0	0.0	882	0	0	0.0
neos-956971	6109	6.8	*4764547*	*1*	*1*	*7200.0*	*4744541*	*1*	*1*	*7200.0*
neos-957143	4374	4.2	*7409607*	*1*	*1*	*7200.0*	*7358729*	*1*	*1*	*7200.0*
neos-957270	515	0.2	515	3	0	0.3	515	15	0	0.4
neos-957323	3688	3.8	3688	3	0	4.3	3688	16	0	5.5
neos-957389	1299	0.3	1299	3	0	0.5	1299	16	0	0.6
neos-960392	12434	9.2	12434	3	0	9.8	12434	17	0	10.7
neos-983171	27536	16.1	27536	3	0	16.2	27536	19	0	16.7
neos-984165	30160	19.0	30160	3	0	19.1	30160	18	0	19.3
neos1	4509	12.5	4509	3	0	13.2	4509	16	0	15.1
neos13	2438	0.8	2438	3	0	0.9	2438	16	0	1.7
neos15	508	0.0	508	3	0	0.0	508	15	0	0.1
neos16	475	0.0	475	3	0	0.0	475	15	0	0.1
neos18	293	0.1	293	3	0	0.2	293	15	0	0.4
neos2	8434	34.4	8434	3	0	34.8	8434	16	0	37.6
neos6	6681	2.1	6681	3	0	2.1	6681	17	0	2.8
neos788725	603	0.1	603	3	0	0.0	603	17	0	0.1
neos808444	8106	4.1	8106	3	0	4.0	8106	16	0	4.3
neos858960	2	0.0	2	0	0	0.0	2	0	0	0.0
nesm	6164	0.6	6164	3	0	0.4	6164	18	0	0.8
net12	11945	2.9	11945	3	0	2.9	11945	16	0	3.4
netdiversion	26331	27.9	26331	3	0	27.8	26331	15	0	29.4
netlarge2	111407	1542.8	111407	0	0	1528.1	111407	0	0	1525.0
newdano	455	0.0	455	3	0	0.0	455	15	0	0.0
nl	14132	3.9	14132	3	0	3.9	14132	18	0	4.8
nobel-eu-DBE	2927	0.3	2927	3	0	0.1	2927	15	0	0.3
noswot	134	0.0	135	3	1	0.0	135	16	1	0.0
npmv07	168166	45.2	168166	4	0	47.6	168166	17	0	55.3
ns1111636	27016	78.9	27016	3	0	80.2	27016	16	0	99.7
ns1116954	15636	24.2	15636	3	0	24.4	15636	16	0	25.3
ns1208400	9117	1.7	9117	3	0	1.8	9117	17	0	1.9
ns1456591	2343	0.5	2343	3	0	0.6	2343	16	0	0.8
ns1606230	15157	6.2	15157	3	0	6.1	15157	17	0	6.1
ns1631475	101171	253.3	101171	3	0	256.6	101171	17	0	259.3
ns1644855	47488	109.7	47488	3	0	110.5	47488	18	0	112.8

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
ns1663818	1723	24.9	1723	3	0	25.8	1723	17	0	29.1
ns1685374	105867	779.2	105867	3	0	774.7	105867	18	0	774.3
ns1686196	157	0.1	157	3	0	0.1	157	16	0	0.2
ns1688347	178	0.1	178	3	0	0.1	178	16	0	0.2
ns1696083	498	0.3	498	3	0	0.3	498	16	0	0.6
ns1702808	58	0.0	58	3	0	0.0	58	15	0	0.0
ns1745726	182	0.1	182	3	0	0.1	182	15	0	0.3
ns1758913	27608	335.6	27608	3	0	335.8	27608	17	0	336.2
ns1766074	27	0.0	27	3	0	0.0	27	15	0	0.0
ns1769397	206	0.2	206	3	0	0.2	206	15	0	0.3
ns1778858	14685	5.4	14685	3	0	5.4	14685	17	0	5.9
ns1830653	2753	0.5	2753	3	0	0.7	2753	16	0	0.8
ns1856153	3231	0.7	3231	3	0	0.7	3231	15	0	1.0
ns1904248	42350	55.0	42365	4	1	55.9	42365	16	1	57.4
ns1905797	5721	2.5	5721	3	0	2.8	5721	16	0	3.8
ns1905800	2057	0.5	2057	3	0	0.4	2057	16	0	0.8
ns1952667	81	0.2	81	3	0	0.2	81	18	0	0.3
ns2017839	60283	118.1	60283	4	0	117.7	60283	17	0	120.4
ns2081729	115	0.0	115	3	0	0.0	115	15	0	0.0
ns2118727	23580	111.8	23580	3	0	112.7	23580	16	0	117.4
ns2122603	13837	3.5	13837	3	0	3.8	13837	16	0	4.0
ns2124243	94850	312.1	94850	3	0	313.0	94850	16	0	314.2
ns2137859	2864	5.2	2864	3	0	6.2	2864	16	0	10.6
ns4-pr3	10658	1.8	10658	3	0	1.8	10658	15	0	2.1
ns4-pr9	9965	1.5	9965	3	0	1.3	9965	15	0	1.6
ns894236	36452	16.2	36452	3	0	15.4	36452	17	0	15.7
ns894244	29128	21.5	29128	3	0	21.7	29128	17	0	22.1
ns894786	18747	14.5	18747	3	0	14.8	18747	16	0	15.6
ns894788	10998	1.5	10998	3	0	1.4	10998	16	0	1.7
ns903616	30158	22.4	30158	3	0	22.4	30158	16	0	23.1
ns930473	10736	8.2	10736	3	0	8.2	10736	17	0	8.6
nsa	1162	0.1	1162	3	0	0.1	1162	16	0	0.2
nsct1	4841	1.2	4841	3	0	1.4	4841	16	0	1.9
nsct2	6810	1.3	6810	3	0	1.5	6810	16	0	2.1
nsic1	254	0.0	254	3	0	0.0	254	16	0	0.0
nsic2	268	0.0	268	3	0	0.0	268	16	0	0.0
nsir1	3762	0.6	3762	3	0	1.0	3762	16	0	0.8
nsir2	3288	0.5	3288	3	0	0.7	3288	16	0	0.9
nsr8k	102631	402.4	102631	4	0	397.4	102631	22	0	398.9
nsrand-ipx	152	0.1	152	3	0	0.4	152	16	0	0.6
nu120-pr3	6187	1.1	6187	3	0	1.1	6187	16	0	1.1
nu60-pr9	5736	1.1	5736	3	0	1.2	5736	15	0	1.0
nug05	161	0.0	161	3	0	0.0	161	16	0	0.0
nug06	1117	0.1	1117	3	0	0.1	1117	17	0	0.1
nug07	7569	1.0	7569	3	0	0.8	7569	19	0	0.9
nug08	13541	2.0	13541	3	0	1.7	13541	19	0	2.0
nug08-3rd	43975	1840.1	43975	4	0	1869.3	43975	22	0	1861.6
nug12	99946	160.0	99946	4	0	159.6	99946	22	0	160.2
nw04	82	0.3	82	3	0	1.0	82	15	0	3.0
nw14	166	0.7	166	3	0	1.9	166	15	0	4.9
ofi	139069	407.6	139104	4	2	412.4	139104	17	2	424.2
opm2-z10-s2	21918	111.0	21918	3	0	110.7	21918	17	0	113.7
opm2-z11-s8	24349	162.2	24349	3	0	163.7	24349	16	0	167.4
opm2-z12-s14	28647	279.0	28647	3	0	283.9	28647	16	0	298.7
opm2-z12-s7	30797	313.1	30797	3	0	315.2	30797	16	0	310.4

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
opm2-z7-s2	10304	7.3	10304	3	0	7.4	10304	16	0	8.0
opt1217	143	0.0	143	3	0	0.0	143	15	0	0.0
orna1	1327	0.2	1327	4	0	0.2	1327	19	0	0.2
orna2	1539	0.2	1539	4	0	0.2	1539	18	0	0.5
orna3	1754	0.2	1754	4	0	0.3	1754	19	0	0.6
orna4	2708	0.2	2708	4	0	0.3	2708	18	0	0.6
orna7	2280	0.3	2280	4	0	0.3	2280	19	0	0.6
orswq2	148	0.0	148	3	0	0.0	148	16	0	0.0
osa-07	966	0.5	966	3	0	0.5	966	15	0	0.8
osa-14	2104	1.2	2104	3	0	1.4	2104	15	0	2.2
osa-30	5628	5.9	5628	3	0	6.3	5628	15	0	7.9
osa-60	13364	38.1	13364	3	0	39.1	13364	16	0	43.1
p0033	19	0.0	19	3	0	0.0	19	15	0	0.0
p0040	13	0.0	23	4	1	0.0	23	16	1	0.0
p010	13700	1.8	13700	3	0	2.2	13700	16	0	2.7
p0201	50	0.0	51	1	1	0.0	51	1	1	0.0
p0282	114	0.0	114	3	0	0.0	114	16	0	0.0
p0291	27	0.0	27	3	0	0.0	27	15	0	0.0
p05	7515	0.9	7515	3	0	0.8	7515	16	0	1.3
p0548	84	0.0	84	3	0	0.0	84	16	0	0.0
p100x588b	235	0.0	235	3	0	0.0	235	15	0	0.0
p19	315	0.0	315	3	0	0.0	315	17	0	0.1
p2756	73	0.0	73	3	0	0.0	73	15	0	0.1
p2m2p1m1p0n100	10	0.0	10	3	0	0.0	10	15	0	0.0
p6000	728	0.1	728	3	0	0.2	728	16	0	0.5
p6b	548	0.1	548	0	0	0.1	548	0	0	0.0
p80x400b	152	0.0	152	3	0	0.0	152	15	0	0.0
pcb1000	2684	0.3	2684	3	0	0.4	2684	16	0	0.6
pcb3000	7719	0.9	7719	3	0	1.0	7719	16	0	1.3
pds-02	2713	0.1	2713	0	0	0.1	2713	0	0	0.1
pds-06	10699	1.1	10699	2	0	1.1	10699	2	0	1.4
pds-10	15362	1.9	15362	0	0	1.9	15362	0	0	2.4
pds-100	661386	4676.7	661386	3	0	4662.1	661386	15	0	4667.9
pds-20	37294	13.1	37294	3	0	13.4	37294	15	0	14.2
pds-30	65338	58.4	65338	0	0	57.5	65338	0	0	58.3
pds-40	101787	159.1	101787	3	0	155.7	101787	15	0	157.2
pds-50	137692	289.9	137692	3	0	293.2	137692	16	0	302.2
pds-60	188842	563.5	188842	3	0	563.8	188842	15	0	567.6
pds-70	222389	759.3	222389	0	0	761.6	222389	0	0	760.2
pds-80	240224	823.9	240224	3	0	821.3	240224	16	0	824.8
pds-90	512992	3214.8	512992	3	0	3210.0	512992	16	0	3220.0
perold	5155	0.8	5155	4	0	0.6	5155	18	0	0.8
pf2177	9474	3.4	9474	3	0	3.3	9474	18	0	3.5
pg	2164	0.0	2164	3	0	0.1	2164	16	0	0.2
pg5_34	3400	0.1	3400	3	0	0.1	3400	16	0	0.3
pgp2	3713	0.1	3713	3	0	0.2	3713	16	0	0.5
pigeon-10	245	0.0	245	3	0	0.0	245	15	0	0.0
pigeon-11	237	0.0	237	3	0	0.0	237	15	0	0.0
pigeon-12	305	0.0	305	3	0	0.0	305	15	0	0.1
pigeon-13	288	0.0	288	3	0	0.0	288	15	0	0.1
pigeon-19	608	0.0	608	3	0	0.1	608	15	0	0.2
pilot	10227	3.6	10227	3	0	3.6	10227	19	0	4.4
pilot-ja	11126	1.7	11126	4	0	1.6	11126	18	0	1.6
pilot-we	5730	0.8	5730	4	0	0.8	5730	19	0	0.7
pilot4	1483	0.2	1483	4	0	0.2	1483	20	0	0.5

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
pilot87	15871	9.5	15871	4	0	9.5	15871	19	0	10.5
pilotnov	7134	0.9	7134	4	0	0.7	7134	21	0	0.8
pk1	101	0.0	101	3	0	0.0	101	16	0	0.0
pldd000b	1667	0.2	1667	3	0	0.1	1667	16	0	0.3
pldd001b	1662	0.2	1662	3	0	0.2	1662	16	0	0.6
pldd002b	1649	0.2	1649	3	0	0.3	1649	16	0	0.6
pldd003b	1657	0.2	1657	3	0	0.3	1657	16	0	0.3
pldd004b	1654	0.2	1654	3	0	0.2	1654	16	0	0.6
pldd005b	1655	0.2	1655	3	0	0.2	1655	16	0	0.6
pldd006b	1693	0.2	1693	3	0	0.3	1693	16	0	0.5
pldd007b	1714	0.2	1714	3	0	0.3	1714	16	0	0.6
pldd008b	1716	0.2	1716	3	0	0.2	1716	16	0	0.2
pldd009b	2258	0.3	2258	3	0	0.4	2258	16	0	0.7
pldd010b	2462	0.4	2462	3	0	0.5	2462	16	0	0.8
pldd011b	2413	0.4	2413	3	0	0.5	2413	16	0	0.4
pldd012b	2447	0.4	2447	3	0	0.5	2447	16	0	0.4
pltexpa2-16	1094	0.1	1094	3	0	0.0	1094	15	0	0.1
pltexpa2-6	411	0.0	411	3	0	0.0	411	15	0	0.1
pltexpa3_16	15975	4.1	15975	3	0	4.4	15975	15	0	5.9
pltexpa3_6	2741	0.3	2741	3	0	0.5	2741	15	0	0.5
pltexpa4_6	14762	3.5	14762	3	0	4.0	14762	15	0	5.2
pp08a	145	0.0	145	3	0	0.0	145	15	0	0.0
pp08aCUTS	224	0.0	224	3	0	0.0	224	16	0	0.0
primagaz	1802	0.4	1802	3	0	0.4	1802	16	0	0.7
problem	9	0.0	9	0	0	0.0	9	0	0	0.0
probportfolio	126	0.0	126	3	0	0.0	126	15	0	0.0
prod1	193	0.0	193	3	0	0.0	193	16	0	0.1
prod2	344	0.0	344	3	0	0.1	344	16	0	0.1
progas	3998	0.8	3998	3	0	0.8	3998	18	0	0.9
protfold	13331	2.8	13331	3	0	2.7	13331	17	0	2.6
pw-myciel4	2154	0.6	2154	3	0	0.7	2154	16	0	0.5
qap10	58606	30.2	58606	4	0	30.0	58606	22	0	30.2
qiu	1604	0.2	1604	3	0	0.2	1604	16	0	0.3
qiulp	1604	0.2	1604	3	0	0.1	1604	16	0	0.2
qnet1	789	0.1	789	3	0	0.1	789	16	0	0.2
qnet1_o	406	0.0	406	3	0	0.0	406	16	0	0.1
queens-30	13322	10.8	13322	3	0	13.5	13322	17	0	14.6
r05	7499	0.7	7499	3	0	0.9	7499	16	0	1.3
r80x800	188	0.0	188	3	0	0.0	188	15	0	0.1
rail01	349407	4435.1	349603	7	1	4461.4	349603	37	1	4453.6
rail2586	30254	675.5	30254	3	0	679.5	30254	18	0	717.2
rail4284	54982	1670.1	54982	3	0	1678.7	54982	18	0	1708.1
rail507	13033	11.2	13033	3	0	11.6	13033	18	0	13.5
rail516	6992	5.6	6992	3	0	5.9	6992	15	0	7.0
rail582	10608	9.2	10608	3	0	9.5	10608	16	0	10.8
ramos3	19446	58.7	19446	4	0	60.3	19446	21	0	60.5
ran10x10a	133	0.0	133	3	0	0.0	133	15	0	0.0
ran10x10b	149	0.0	149	3	0	0.0	149	15	0	0.0
ran10x10c	161	0.0	161	3	0	0.0	161	15	0	0.0
ran10x12	162	0.0	162	3	0	0.0	162	15	0	0.0
ran10x26	423	0.0	423	3	0	0.0	423	15	0	0.1
ran12x12	186	0.0	186	3	0	0.0	186	15	0	0.0
ran12x21	380	0.0	380	3	0	0.0	380	15	0	0.1
ran13x13	236	0.0	236	3	0	0.0	236	15	0	0.0
ran14x18	397	0.0	397	3	0	0.0	397	15	0	0.0

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
ran14x18-disj-8	1915	0.2	1916	4	1	0.3	1916	19	1	0.6
ran14x18.disj-8	1915	0.2	1916	4	1	0.3	1916	19	1	0.6
ran14x18_1	434	0.0	434	3	0	0.0	434	15	0	0.1
ran16x16	379	0.0	379	3	0	0.0	379	15	0	0.0
ran17x17	258	0.0	258	3	0	0.0	258	15	0	0.0
ran4x64	515	0.0	515	3	0	0.0	515	15	0	0.0
ran6x43	447	0.0	447	3	0	0.0	447	15	0	0.1
ran8x32	402	0.0	402	3	0	0.0	402	15	0	0.1
rat1	2870	0.8	2870	3	0	0.7	2870	16	0	1.4
rat5	3024	1.1	3024	3	0	1.4	3024	17	0	2.2
rat7a	11319	12.3	11319	3	0	13.7	11319	17	0	15.0
rd-rplusc-21	138	6.3	139	3	1	7.7	139	15	1	12.6
reblock166	3359	1.4	3359	3	0	1.3	3359	16	0	1.6
reblock354	12854	7.9	12854	3	0	8.2	12854	17	0	8.7
reblock420	9480	11.5	9480	3	0	11.6	9480	17	0	12.9
reblock67	972	0.1	972	3	0	0.1	972	16	0	0.4
recipe	40	0.0	40	3	0	0.0	40	15	0	0.0
refine	21	0.0	21	3	0	0.0	21	15	0	0.0
rentacar	6483	1.2	6483	3	0	1.1	6483	17	0	1.5
rgn	85	0.0	85	3	0	0.0	85	15	0	0.0
rlfddd	825	0.2	825	1	0	0.3	825	1	0	0.4
rlfdual	8652	1.7	8652	0	0	1.7	8652	0	0	1.8
rlfprim	5532	1.2	5532	0	0	1.2	5532	0	0	1.2
rlp1	248	0.0	248	3	0	0.0	248	16	0	0.0
rmatr100-p10	6260	1.6	6260	3	0	1.5	6260	16	0	1.7
rmatr100-p5	10885	3.1	10885	3	0	3.4	10885	16	0	3.1
rmatr200-p10	13646	11.7	13646	3	0	11.8	13646	17	0	12.6
rmatr200-p20	11166	7.7	11166	3	0	8.2	11166	17	0	8.1
rmatr200-p5	17650	18.8	17650	3	0	18.8	17650	17	0	19.4
rmine10	13871	20.9	13871	3	0	21.6	13871	17	0	22.7
rmine14	92846	1720.4	92846	3	0	1714.0	92846	17	0	1735.3
rmine6	2058	0.7	2058	3	0	0.6	2058	16	0	0.7
rocII-4-11	751	0.3	751	3	0	0.5	751	16	0	1.2
rocII-7-11	1075	0.5	1075	3	0	0.8	1075	16	0	2.1
rocII-9-11	1523	0.8	1523	3	0	1.1	1523	16	0	2.7
rococoB10-011000	4263	0.8	4263	3	0	0.7	4263	16	0	0.9
rococoC10-001000	2344	0.3	2344	3	0	0.2	2344	16	0	0.5
rococoC11-011100	10112	1.6	10112	3	0	1.8	10112	16	0	1.6
rococoC12-111000	11920	3.4	11920	3	0	3.2	11920	16	0	3.8
roll3000	2627	0.3	2627	3	0	0.2	2627	17	0	0.6
rosen1	898	0.2	898	3	0	0.1	898	17	0	0.3
rosen10	2401	1.0	2401	3	0	0.9	2401	17	0	1.2
rosen2	1627	0.6	1627	3	0	0.6	1627	17	0	1.0
rosen7	332	0.0	332	3	0	0.0	332	16	0	0.1
rosen8	662	0.1	662	3	0	0.2	662	16	0	0.2
rout	293	0.0	293	3	0	0.0	293	16	0	0.0
route	2239	1.1	2239	3	0	1.2	2239	16	0	2.0
roy	119	0.0	119	3	0	0.0	119	15	0	0.0
rvb-sub	512	0.8	512	3	0	1.4	512	16	0	4.1
satellites1-25	9023	4.7	9023	3	0	4.9	9023	19	0	5.2
satellites2-60	79705	373.8	79705	3	0	376.6	79705	19	0	381.0
satellites2-60-fs	65144	270.6	65144	4	0	271.8	65144	19	0	271.9
satellites3-40	190485	2657.9	190485	4	0	2654.6	190485	21	0	2666.7
satellites3-40-fs	199929	2345.5	199931	4	1	2342.0	199931	21	1	2355.8
sc105	99	0.0	99	3	0	0.0	99	16	0	0.0

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
sc205	217	0.0	217	3	0	0.0	217	16	0	0.0
sc205-2r-100	1109	0.1	1109	3	0	0.2	1109	15	0	0.2
sc205-2r-16	171	0.0	171	3	0	0.0	171	15	0	0.0
sc205-2r-1600	9189	4.7	9189	3	0	4.9	9189	15	0	5.2
sc205-2r-200	2195	0.4	2195	3	0	0.5	2195	15	0	0.5
sc205-2r-27	327	0.0	327	3	0	0.0	327	15	0	0.0
sc205-2r-32	331	0.0	331	3	0	0.0	331	15	0	0.1
sc205-2r-4	55	0.0	55	3	0	0.0	55	15	0	0.0
sc205-2r-400	4393	1.1	4393	3	0	1.3	4393	15	0	1.3
sc205-2r-50	621	0.1	621	3	0	0.0	621	16	0	0.1
sc205-2r-64	651	0.1	651	3	0	0.0	651	15	0	0.1
sc205-2r-8	100	0.0	100	3	0	0.0	100	15	0	0.0
sc205-2r-800	8729	3.4	8729	3	0	3.4	8729	15	0	4.0
sc50a	45	0.0	45	3	0	0.0	45	16	0	0.0
sc50b	49	0.0	49	3	0	0.0	49	16	0	0.0
scagr25	784	0.0	784	3	0	0.1	784	16	0	0.1
scagr7	178	0.0	178	3	0	0.0	178	16	0	0.0
scagr7-2b-16	717	0.0	717	3	0	0.1	717	16	0	0.1
scagr7-2b-4	189	0.0	189	3	0	0.0	189	16	0	0.0
scagr7-2b-64	11614	1.2	11614	3	0	1.6	11614	16	0	2.2
scagr7-2c-16	684	0.0	684	3	0	0.0	684	16	0	0.1
scagr7-2c-4	186	0.0	186	3	0	0.0	186	16	0	0.0
scagr7-2c-64	2727	0.2	2727	3	0	0.2	2727	16	0	0.4
scagr7-2r-108	4538	0.3	4538	3	0	0.4	4538	16	0	0.7
scagr7-2r-16	707	0.0	707	3	0	0.0	707	16	0	0.1
scagr7-2r-216	9082	0.8	9082	3	0	1.0	9082	16	0	1.7
scagr7-2r-27	1144	0.0	1144	3	0	0.0	1144	16	0	0.1
scagr7-2r-32	1335	0.1	1335	3	0	0.0	1335	16	0	0.1
scagr7-2r-4	185	0.0	185	3	0	0.0	185	16	0	0.0
scagr7-2r-432	16562	2.5	16562	3	0	2.8	16562	16	0	4.0
scagr7-2r-54	2299	0.1	2299	3	0	0.1	2299	16	0	0.5
scagr7-2r-64	2763	0.2	2763	3	0	0.2	2763	16	0	0.5
scagr7-2r-8	357	0.0	357	3	0	0.0	357	16	0	0.0
scagr7-2r-864	32972	12.0	32972	3	0	12.6	32972	16	0	15.2
scfxm1	447	0.0	447	3	0	0.0	447	17	0	0.1
scfxm1-2b-16	4522	0.4	4522	3	0	0.5	4522	18	0	0.9
scfxm1-2b-4	1135	0.1	1135	3	0	0.1	1135	17	0	0.1
scfxm1-2b-64	26349	8.8	26349	3	0	9.2	26349	19	0	11.4
scfxm1-2c-4	1071	0.1	1071	3	0	0.1	1071	16	0	0.1
scfxm1-2r-128	23788	7.7	23788	3	0	8.0	23788	19	0	10.1
scfxm1-2r-16	4672	0.5	4672	3	0	0.5	4672	18	0	0.6
scfxm1-2r-256	47280	30.8	47280	4	0	31.7	47280	20	0	36.2
scfxm1-2r-27	8158	0.8	8158	3	0	0.6	8158	17	0	1.2
scfxm1-2r-32	8924	1.0	8924	3	0	0.8	8924	19	0	1.5
scfxm1-2r-4	1203	0.1	1203	3	0	0.0	1203	17	0	0.1
scfxm1-2r-64	13133	1.8	13133	3	0	2.1	13133	18	0	3.1
scfxm1-2r-8	2252	0.2	2252	3	0	0.1	2252	18	0	0.2
scfxm1-2r-96	16931	3.7	16931	3	0	3.9	16931	19	0	5.5
scfxm2	1119	0.1	1119	3	0	0.1	1119	19	0	0.1
scfxm3	1992	0.1	1992	3	0	0.1	1992	18	0	0.4
scorpion	245	0.0	245	3	0	0.0	245	17	0	0.1
scrs8	608	0.0	608	3	0	0.0	608	16	0	0.1
scrs8-2b-16	88	0.0	88	3	0	0.0	88	16	0	0.1
scrs8-2b-4	22	0.0	22	3	0	0.0	22	15	0	0.0
scrs8-2b-64	296	0.0	296	3	0	0.0	296	16	0	0.1

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
scrs8-2c-16	91	0.0	91	3	0	0.0	91	16	0	0.0
scrs8-2c-32	179	0.0	179	3	0	0.0	179	16	0	0.1
scrs8-2c-4	22	0.0	22	3	0	0.0	22	15	0	0.0
scrs8-2c-64	358	0.0	358	3	0	0.1	358	16	0	0.2
scrs8-2c-8	45	0.0	45	3	0	0.0	45	15	0	0.0
scrs8-2r-128	543	0.1	543	3	0	0.1	543	16	0	0.3
scrs8-2r-16	96	0.0	96	3	0	0.0	96	16	0	0.1
scrs8-2r-256	1119	0.2	1119	3	0	0.3	1119	16	0	0.6
scrs8-2r-27	103	0.0	103	3	0	0.0	103	15	0	0.0
scrs8-2r-32	192	0.0	192	3	0	0.0	192	16	0	0.1
scrs8-2r-4	24	0.0	24	3	0	0.0	24	15	0	0.0
scrs8-2r-512	2532	0.4	2532	3	0	0.6	2532	16	0	1.1
scrs8-2r-64	384	0.0	384	3	0	0.0	384	16	0	0.1
scrs8-2r-64b	271	0.0	271	3	0	0.0	271	16	0	0.1
scrs8-2r-8	41	0.0	41	3	0	0.0	41	15	0	0.0
scsd1	97	0.0	97	3	0	0.0	97	15	0	0.1
scsd6	423	0.0	423	3	0	0.1	423	16	0	0.1
scsd8	1837	0.2	1837	3	0	0.3	1837	16	0	0.5
scsd8-2b-16	295	0.0	295	3	0	0.0	295	15	0	0.1
scsd8-2b-4	47	0.0	47	1	0	0.0	47	1	0	0.0
scsd8-2b-64	2056	0.1	2056	3	0	0.4	2056	16	0	1.0
scsd8-2c-16	198	0.0	198	3	0	0.0	198	15	0	0.0
scsd8-2c-4	47	0.0	47	1	0	0.0	47	1	0	0.0
scsd8-2c-64	2056	0.2	2056	3	0	0.4	2056	16	0	1.1
scsd8-2r-108	1114	0.1	1114	3	0	0.1	1114	16	0	0.7
scsd8-2r-16	231	0.0	231	3	0	0.0	231	15	0	0.0
scsd8-2r-216	2314	0.1	2314	3	0	0.3	2314	16	0	0.9
scsd8-2r-27	286	0.0	286	3	0	0.0	286	15	0	0.1
scsd8-2r-32	429	0.0	429	3	0	0.1	429	15	0	0.1
scsd8-2r-4	47	0.0	47	1	0	0.0	47	1	0	0.0
scsd8-2r-432	4530	0.2	4530	3	0	0.5	4530	16	0	1.8
scsd8-2r-54	600	0.1	600	3	0	0.1	600	15	0	0.2
scsd8-2r-64	1188	0.1	1188	3	0	0.1	1188	16	0	0.4
scsd8-2r-8	97	0.0	97	3	0	0.0	97	15	0	0.0
scsd8-2r-8b	97	0.0	97	3	0	0.0	97	15	0	0.0
sct1	15178	7.4	15178	4	0	7.6	15178	19	0	8.5
sct32	14051	3.8	14051	4	0	3.7	14051	21	0	4.7
sct5	6217	3.1	6217	3	0	3.5	6217	18	0	4.2
sctap1	262	0.0	262	3	0	0.0	262	16	0	0.0
sctap1-2b-16	323	0.0	323	3	0	0.0	323	15	0	0.1
sctap1-2b-4	83	0.0	83	3	0	0.0	83	15	0	0.0
sctap1-2b-64	4773	0.6	4773	3	0	0.7	4773	16	0	1.7
sctap1-2c-16	329	0.0	329	3	0	0.0	329	16	0	0.1
sctap1-2c-4	85	0.0	85	3	0	0.0	85	16	0	0.0
sctap1-2c-64	1146	0.1	1146	3	0	0.1	1146	15	0	0.3
sctap1-2r-108	2109	0.2	2109	3	0	0.4	2109	15	0	0.8
sctap1-2r-16	282	0.0	282	3	0	0.0	282	15	0	0.0
sctap1-2r-216	4248	0.5	4248	3	0	0.8	4248	16	0	1.4
sctap1-2r-27	536	0.0	536	3	0	0.0	536	15	0	0.1
sctap1-2r-32	559	0.0	559	3	0	0.0	559	15	0	0.1
sctap1-2r-4	84	0.0	84	3	0	0.0	84	15	0	0.0
sctap1-2r-480	9373	1.5	9373	3	0	1.9	9373	16	0	3.4
sctap1-2r-54	1069	0.1	1069	3	0	0.1	1069	15	0	0.2
sctap1-2r-64	1123	0.1	1123	3	0	0.2	1123	15	0	0.3
sctap1-2r-8	147	0.0	147	3	0	0.0	147	15	0	0.0

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
sctap1-2r-8b	161	0.0	161	3	0	0.0	161	15	0	0.0
sctap2	505	0.1	505	3	0	0.1	505	16	0	0.1
sctap3	604	0.1	604	3	0	0.1	604	15	0	0.1
seba	6	0.0	6	3	0	0.0	6	15	0	0.0
self	13495	48.6	13495	3	0	49.7	13495	18	0	51.8
set1ch	513	0.0	513	3	0	0.0	513	15	0	0.0
set3-10	2279	0.1	2279	3	0	0.2	2279	16	0	0.6
set3-15	2269	0.2	2269	3	0	0.2	2269	16	0	0.6
set3-20	2344	0.1	2344	3	0	0.2	2344	17	0	0.3
seymour	3858	1.0	3858	3	0	0.9	3858	16	0	0.9
seymour-disj-10	5700	1.8	5702	4	1	2.1	5702	18	1	2.6
seymour.disj-10	5700	1.7	5702	4	1	2.1	5702	18	1	2.7
seymourl	3858	0.9	3858	3	0	1.1	3858	16	0	0.9
sgpf5y6	153350	167.8	197667	3	1	221.6	197667	16	1	232.0
share1b	217	0.0	217	3	0	0.0	217	17	0	0.0
share2b	159	0.0	159	3	0	0.0	159	17	0	0.0
shell	595	0.0	595	1	0	0.0	595	1	0	0.0
ship04l	473	0.0	473	3	0	0.0	473	16	0	0.1
ship04s	383	0.0	383	3	0	0.0	383	16	0	0.1
ship08l	810	0.1	810	3	0	0.1	810	16	0	0.3
ship08s	513	0.0	513	3	0	0.1	513	16	0	0.2
ship12l	1085	0.1	1085	3	0	0.2	1085	16	0	0.5
ship12s	648	0.0	648	3	0	0.1	648	16	0	0.3
shipsched	2174	0.6	2174	3	0	0.9	2174	15	0	1.1
shs1023	174484	604.0	177407	7	5	618.0	177407	35	19	633.0
siena1	28441	19.0	28441	4	0	19.7	28441	24	0	20.7
sierra	640	0.0	640	3	0	0.0	640	16	0	0.2
sing2	37541	40.3	37541	3	0	40.9	37541	16	0	42.0
sing245	218115	1608.1	218115	3	0	1611.8	218115	17	0	1622.6
sing359	352087	5830.9	352087	3	0	5799.4	352087	18	0	5800.1
slptsk	4856	1.2	4856	3	0	1.8	4856	17	0	2.6
small000	557	0.0	557	3	0	0.0	557	16	0	0.1
small001	725	0.0	737	3	1	0.0	737	16	1	0.2
small002	834	0.0	850	3	1	0.1	850	17	1	0.2
small003	681	0.0	693	3	1	0.1	693	16	1	0.2
small004	465	0.0	476	3	1	0.1	476	16	1	0.2
small005	601	0.0	621	4	1	0.1	621	17	1	0.2
small006	518	0.0	540	3	1	0.1	540	16	1	0.2
small007	517	0.0	547	3	1	0.1	547	16	1	0.2
small008	489	0.0	511	3	1	0.1	511	16	1	0.2
small009	416	0.0	435	3	1	0.1	435	16	1	0.1
small010	328	0.0	342	3	1	0.1	342	16	1	0.1
small011	337	0.0	348	3	1	0.0	348	16	1	0.1
small012	287	0.0	287	3	0	0.1	287	16	0	0.1
small013	289	0.0	289	3	0	0.1	289	16	0	0.1
small014	341	0.0	341	3	0	0.0	341	16	0	0.1
small015	338	0.0	338	3	0	0.1	338	16	0	0.1
small016	338	0.0	338	3	0	0.0	338	16	0	0.1
south31	23654	14.9	23654	3	0	15.2	23654	16	0	16.6
sp97ar	6984	2.9	6986	4	1	3.3	6986	17	1	4.5
sp97ic	3102	0.8	3102	4	0	0.7	3102	18	0	0.8
sp98ar	10342	2.9	10349	4	1	3.1	10349	18	1	3.9
sp98ic	3276	0.7	3276	4	0	0.8	3276	17	0	1.5
sp98ir	2967	0.7	2970	4	1	0.5	2970	19	1	1.1
square15	208626	2424.8	208626	0	0	2424.6	208626	0	0	2438.8

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
stair	658	0.1	658	3	0	0.1	658	18	0	0.2
standata	50	0.0	50	3	0	0.0	50	15	0	0.0
standmps	189	0.0	189	3	0	0.0	189	16	0	0.1
stat96v1	275141	2283.3	*875464*	*4*	*4*	*6516.9*	*875464*	*16*	*16*	*6522.2*
stat96v4	144049	395.8	144049	4	0	402.2	144049	20	0	403.8
stat96v5	11838	18.4	13119	5	1	21.2	13119	20	1	25.3
stein27	32	0.0	32	3	0	0.0	32	16	0	0.0
stein45	59	0.0	59	3	0	0.0	59	16	0	0.0
stocfor1	108	0.0	108	3	0	0.0	108	16	0	0.0
stocfor2	1994	0.3	1994	3	0	0.2	1994	16	0	0.7
stocfor3	16167	6.2	16167	3	0	6.5	16167	16	0	7.6
stockholm	25891	20.3	26971	4	1	21.7	26971	17	1	23.4
stormG2_1000	732642	4648.3	732642	4	0	4679.4	732642	17	0	4728.4
stormg2-125	92103	45.6	92103	3	0	47.4	92103	16	0	50.6
stormg2-27	20754	2.4	20754	3	0	2.7	20754	16	0	3.1
stormg2-8	6863	0.6	6863	3	0	0.4	6863	16	0	0.5
stormg2_1000	732642	4660.6	732642	4	0	4685.5	732642	17	0	4750.8
stp3d	148139	1238.6	148139	3	0	1242.3	148139	18	0	1247.0
sts405	434	0.5	434	3	0	0.7	434	16	0	1.5
sts729	907	1.3	907	3	0	1.8	907	18	0	4.5
swath	127	0.1	127	3	0	0.1	127	16	0	0.2
sws	1150	0.3	1150	3	0	0.3	1150	16	0	0.9
t0331-4l	15487	14.7	15487	4	0	15.4	15487	19	0	18.6
t1717	12220	7.0	12220	4	0	7.6	12220	19	0	10.7
t1722	10612	3.3	10612	3	0	3.6	10612	17	0	4.7
tanglegram1	478	0.3	478	0	0	0.3	478	0	0	0.3
tanglegram2	236	0.1	236	0	0	0.0	236	0	0	0.1
testbig	8010	4.4	8010	3	0	4.5	8010	15	0	5.0
timtab1	20	0.0	20	3	0	0.0	20	15	0	0.0
timtab2	36	0.0	36	3	0	0.0	36	15	0	0.0
toll-like	717	0.1	717	0	0	0.1	717	0	0	0.0
tr12-30	696	0.0	696	3	0	0.0	696	15	0	0.1
transportmoment	9617	1.7	9617	4	0	1.9	9617	20	0	2.5
triptim1	68167	93.3	68167	3	0	92.8	68167	17	0	94.3
triptim2	204335	572.2	204335	3	0	567.7	204335	18	0	579.6
triptim3	89514	156.7	89514	3	0	161.5	89514	18	0	162.3
truss	21892	4.7	21892	3	0	4.6	21892	16	0	5.1
tuff	212	0.0	212	3	0	0.0	212	17	0	0.1
tw-myciel4	11588	2.6	11588	3	0	2.3	11588	15	0	2.4
uc-case11	39603	75.9	39603	3	0	75.3	39603	17	0	79.8
uc-case3	34156	65.4	34156	3	0	66.7	34156	17	0	66.7
uct-subprob	2988	0.5	2988	3	0	0.5	2988	16	0	0.7
ulevimin	125858	108.8	125858	3	0	108.6	125858	17	0	109.4
umts	5537	0.9	5549	5	2	0.7	5549	17	2	1.2
unitcal_7	21824	8.9	21824	3	0	9.1	21824	16	0	10.1
us04	338	0.2	338	3	0	0.5	338	16	0	1.2
usAbbrv-8-25_70	2434	0.1	2434	3	0	0.1	2434	16	0	0.1
van	11014	5.2	11014	3	0	6.0	11014	19	0	10.7
vpm1	130	0.0	130	3	0	0.0	130	15	0	0.0
vpm2	192	0.0	192	3	0	0.0	192	15	0	0.0
vpphard	11075	11.3	11075	3	0	11.5	11075	17	0	12.4
vpphard2	8024	32.4	8024	3	0	33.1	8024	16	0	39.6
vtp-base	75	0.0	75	3	0	0.0	75	16	0	0.0
wachplan	2033	0.5	2033	3	0	0.4	2033	16	0	0.4
watson_1	188103	366.3	188496	4	1	369.2	188496	17	1	386.2

Table 5 continued

	SoPlex$_9$		SoPlex$_{50}$				SoPlex$_{250}$			
Instance	iter	t	iter	R	R_0	t	iter	R	R_0	t
watson_2	333044	1519.4	333083	4	1	1518.6	333083	17	1	1585.2
wide15	229488	2733.8	229488	0	0	2734.6	229488	0	0	2731.0
wnq-n100-mw99-14	764	5.0	764	3	0	5.8	764	15	0	8.7
wood1p	142	0.1	142	3	0	0.2	142	16	0	0.9
woodw	1832	0.6	1832	3	0	0.6	1832	16	0	1.0
world	70204	131.2	70320	15	12	132.1	70320	29	12	137.6
zed	31	0.0	31	3	0	0.0	31	16	0	0.0
zib54-UUE	1855	0.1	1855	3	0	0.1	1855	16	0	0.4

Table 6: Detailed results comparing QSopt_ex's performance warm started from bases returned by SoPlex$_9$ and SoPlex$_{50}$. Entries corresponding to time or memory outs (all due to QSopt_ex) are printed in italics. See Table 3 for aggregated results.

iter — number of simplex iterations by SoPlex and QSopt_ex
p_{ex} — max. precision used by QSopt_ex
$t_{9/50}$ — running time of SoPlex$_{9/50}$
t_{ex} — running time of QSopt_ex
t — total running time
R_0 — number of refinements to final basis

	SoPlex$_9$+QSopt_ex						SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
10teams	1611	64	0.2	0.1	0.3	0	1611	64	0.3	0.1	0.4
16_n14	329933	64	376.0	1.6	377.6	0	329933	64	371.8	1.9	373.7
22433	1041	64	0.1	0.1	0.1	0	1041	64	0.1	0.0	0.1
23588	548	64	0.0	0.0	0.1	0	548	64	0.0	0.0	0.0
25fv47	4359	64	0.8	0.9	1.6	0	4359	64	0.7	0.7	1.3
30_70_45_095_100	16103	64	7.4	0.5	7.9	0	16103	64	7.6	0.3	7.9
30n20b8	269	64	0.3	0.2	0.4	0	269	64	0.3	0.1	0.4
50v-10	220	64	0.0	0.0	0.0	0	220	64	0.0	0.0	0.1
80bau3b	7797	64	1.0	0.2	1.2	0	7797	64	1.1	0.2	1.3
Test3	6948	64	3.3	1.9	5.2	0	6948	64	3.7	1.9	5.6
a1c1s1	1742	64	0.1	0.0	0.1	0	1742	64	0.1	0.0	0.1
aa01	11898	64	3.5	0.4	3.8	0	11898	64	3.6	0.4	4.0
aa03	7584	64	2.0	0.3	2.3	0	7584	64	1.9	0.3	2.1
aa3	7584	64	2.1	0.3	2.4	0	7584	64	2.0	0.1	2.1
aa4	4486	64	1.1	0.2	1.4	0	4486	64	1.2	0.2	1.4
aa5	9706	64	2.4	0.3	2.7	0	9706	64	2.6	0.2	2.9
aa6	4870	64	1.1	0.1	1.2	0	4870	64	1.2	0.1	1.4
acc-tight4	11142	64	2.6	0.5	3.1	0	11142	64	2.2	0.5	2.7
acc-tight5	10469	64	2.0	0.4	2.5	0	10469	64	2.2	0.2	2.4
acc-tight6	10672	64	2.2	0.4	2.6	0	10672	64	2.3	0.4	2.6
adlittle	88	64	0.0	0.0	0.0	0	88	64	0.0	0.0	0.0
afiro	16	64	0.0	0.0	0.0	0	16	64	0.0	0.0	0.0
aflow30a	396	64	0.0	0.0	0.1	0	396	64	0.0	0.0	0.0
aflow40b	1826	64	0.2	0.1	0.2	0	1826	64	0.1	0.1	0.2
agg	80	64	0.0	0.0	0.0	0	80	64	0.0	0.0	0.0
agg2	152	64	0.0	0.0	0.1	0	152	64	0.0	0.0	0.0
agg3	165	64	0.0	0.0	0.1	0	165	64	0.0	0.0	0.1
air02	95	64	0.1	0.1	0.2	0	95	64	0.1	0.1	0.2
air03	626	64	0.3	0.1	0.5	0	626	64	0.4	0.1	0.6
air04	11898	64	3.4	0.2	3.6	0	11898	64	3.4	0.3	3.8
air05	4486	64	1.2	0.2	1.4	0	4486	64	1.3	0.2	1.5
air06	7584	64	2.0	0.3	2.3	0	7584	64	2.0	0.3	2.2
aircraft	1912	64	0.4	0.2	0.6	0	1912	64	0.2	0.2	0.4
aligninq	1492	64	0.1	0.1	0.2	0	1492	64	0.2	0.1	0.3
app1-2	14617	64	7.2	2.1	9.4	1	14622	64	8.0	2.1	10.1
arki001	1570	128	0.2	2.1	2.3	1	1571	64	0.3	0.4	0.7
ash608gpia-3col	3123	64	0.2	0.3	0.6	0	3123	64	0.3	0.5	0.8
atlanta-ip	38285	128	34.8	3035.9	3070.7	1	38286	64	35.2	2.1	37.3
atm20-100	4586	128	0.4	20.2	20.6	1	4699	64	0.5	0.1	0.6
b2c1s1	2621	64	0.1	0.1	0.2	0	2621	64	0.1	0.1	0.2
bab1	5711	64	4.0	3.4	7.4	0	5711	64	4.7	3.3	8.0
bab3	697791	64	5263.4	14.6	5278.0	0	697791	64	5266.5	14.5	5281.0
bab5	29788	64	10.3	0.5	10.8	0	29788	64	10.5	0.4	11.0
bal8x12	102	64	0.0	0.0	0.0	0	102	64	0.0	0.0	0.0
bandm	474	64	0.0	0.1	0.1	0	474	64	0.0	0.1	0.1
bas1lp	2582	64	1.0	1.3	2.3	0	2582	64	1.1	1.3	2.4

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
baxter	12347	64	2.6	0.6	3.2	0	12347	64	2.8	0.3	3.2
bc	3759	64	1.3	1.6	2.8	0	3759	64	1.4	1.6	3.0
bc1	3759	64	1.3	1.6	2.9	0	3759	64	1.4	1.6	3.0
beaconfd	88	64	0.0	0.0	0.0	0	88	64	0.0	0.0	0.0
beasleyC3	1140	64	0.1	0.0	0.1	0	1140	64	0.1	0.0	0.1
bell3a	81	64	0.0	0.0	0.0	0	81	64	0.0	0.0	0.0
bell5	66	64	0.0	0.0	0.0	0	66	64	0.0	0.0	0.0
berlin_5_8_0	1017	64	0.0	0.1	0.1	0	1017	64	0.1	0.0	0.1
bg512142	2238	64	0.3	0.2	0.4	0	2238	64	0.1	0.2	0.3
biella1	16158	64	4.2	0.4	4.6	0	16158	64	4.2	0.3	4.5
bienst1	455	64	0.0	0.0	0.1	0	455	64	0.0	0.0	0.0
bienst2	455	64	0.0	0.0	0.1	0	455	64	0.0	0.0	0.0
binkar10_1	1267	64	0.1	0.1	0.1	0	1267	64	0.1	0.0	0.1
bk4x3	16	64	0.0	0.0	0.0	0	16	64	0.0	0.0	0.0
blend	97	64	0.0	0.0	0.0	0	97	64	0.0	0.0	0.0
blend2	175	64	0.0	0.0	0.0	0	175	64	0.0	0.0	0.0
blp-ar98	368	64	0.2	0.3	0.4	0	368	64	0.5	0.3	0.8
blp-ic97	446	64	0.3	0.4	0.7	0	446	64	0.4	0.4	0.8
bnatt350	685	64	0.1	0.1	0.2	0	685	64	0.1	0.1	0.1
bnatt400	831	64	0.1	0.1	0.2	0	831	64	0.1	0.1	0.2
bnl1	1474	64	0.1	0.1	0.2	0	1474	64	0.2	0.1	0.2
bnl2	2718	64	0.4	0.1	0.6	0	2718	64	0.5	0.1	0.6
boeing1	458	64	0.0	0.0	0.1	0	458	64	0.0	0.0	0.1
boeing2	145	64	0.0	0.0	0.0	0	145	64	0.0	0.0	0.0
bore3d	100	64	0.0	0.0	0.0	0	100	64	0.0	0.0	0.1
brandy	482	64	0.0	0.1	0.1	0	482	64	0.0	0.1	0.1
buildingenergy	144595	64	827.1	18.9	846.0	0	144595	64	835.0	19.2	854.2
cap6000	815	64	0.2	0.2	0.4	0	815	64	0.3	0.2	0.5
capri	338	64	0.0	0.0	0.1	0	338	64	0.0	0.0	0.1
car4	10442	64	1.9	5.6	7.5	0	10442	64	2.0	5.5	7.5
cari	681	64	0.1	0.4	0.5	0	681	64	0.3	0.6	0.9
cep1	1399	64	0.1	0.1	0.1	0	1399	64	0.1	0.1	0.2
ch	10747	64	1.4	0.2	1.7	0	10747	64	1.5	0.2	1.7
circ10-3	10910	64	10.2	0.8	11.1	0	10910	64	10.4	0.8	11.2
co-100	783	64	1.2	3.7	4.9	0	783	64	2.3	3.7	5.9
co5	12067	64	3.2	1.2	4.4	0	12067	64	3.5	1.2	4.7
co9	19820	64	11.6	5.1	16.7	0	19820	64	11.7	5.3	17.0
complex	9592	64	2.4	2.0	4.3	1	9646	64	2.6	1.8	4.5
cont1	*40707*	*64*	*985.6*	*7200.0*	*8185.6*	*0*	*40707*	*64*	*1226.7*	*7200.0*	*8426.7*
cont4	40802	128	2698.5	7200.0	9898.5	0	40802	128	2839.8	7200.0	10039.8
core2536-691	41182	64	16.0	0.3	16.3	0	41182	64	16.8	0.3	17.1
core4872-1529	69516	64	69.1	3.7	72.8	0	69516	64	70.6	3.6	74.2
cov1075	3270	64	0.5	0.8	1.3	0	3270	64	0.4	0.6	1.0
cq5	12350	64	3.1	0.7	3.8	0	12350	64	3.1	0.8	4.0
cq9	17826	64	7.1	6.3	13.5	0	17826	64	7.2	6.3	13.5
cr42	999	64	0.1	0.1	0.1	0	999	64	0.1	0.0	0.1
cre-a	3555	64	0.3	0.1	0.4	0	3555	64	0.3	0.1	0.5
cre-b	12712	64	5.0	0.7	5.7	0	12712	64	5.4	0.7	6.1
cre-c	2842	64	0.3	0.1	0.4	0	2842	64	0.3	0.1	0.4
cre-d	9271	64	3.5	0.6	4.2	0	9271	64	3.9	0.6	4.5
crew1	1849	64	0.6	0.1	0.7	0	1849	64	0.4	0.0	0.4
csched007	5522	64	0.7	0.1	0.7	0	5522	64	0.4	0.0	0.4
csched008	3102	64	0.4	0.0	0.5	0	3102	64	0.4	0.0	0.4
csched010	6832	64	0.8	0.0	0.8	0	6832	64	0.8	0.0	0.8
cycle	920	64	0.1	0.1	0.2	0	920	64	0.1	0.1	0.1

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
czprob	1572	64	0.2	0.1	0.2	0	1572	64	0.1	0.1	0.2
d10200	2852	64	0.6	0.6	1.1	0	2852	64	0.5	0.6	1.1
d20200	9723	64	1.8	0.9	2.7	0	9723	64	1.9	1.0	2.8
d2q06c	13920	64	3.9	28.9	32.8	0	13920	64	3.7	28.8	32.5
d6cube	1179	64	0.4	0.2	0.6	0	1179	64	0.4	0.1	0.5
dano3_3	46251	64	28.8	7.8	36.6	0	46251	64	29.0	7.8	36.9
dano3_4	46251	64	28.9	7.9	36.9	0	46251	64	28.7	8.0	36.7
dano3_5	46251	64	30.5	8.1	38.6	0	46251	64	29.0	8.4	37.4
dano3mip	46251	64	28.8	8.2	36.9	0	46251	64	29.1	7.8	36.9
danoint	2763	64	0.3	0.1	0.4	0	2763	64	0.3	0.0	0.3
dbir1	14306	64	11.2	1.9	13.1	0	14306	64	11.4	1.9	13.3
dbir2	11641	64	4.9	2.2	7.1	0	11641	64	5.3	2.5	7.7
dc1c	19795	64	6.2	0.7	6.9	0	19795	64	6.2	0.5	6.7
dc1l	24825	64	21.1	1.1	22.2	0	24825	64	21.4	1.1	22.5
dcmulti	479	64	0.0	0.0	0.0	0	479	64	0.0	0.0	0.0
de063155	2546	64	0.3	0.6	0.9	3	2623	64	0.4	0.3	0.7
de080285	888	64	0.1	0.3	0.4	0	888	64	0.1	0.3	0.4
degen2	1325	64	0.1	0.1	0.2	0	1325	64	0.1	0.1	0.2
degen3	5832	64	0.9	0.2	1.1	0	5832	64	0.8	0.2	1.1
delf000	1675	64	0.4	0.6	1.0	0	1675	64	0.5	0.6	1.1
delf001	1703	64	0.4	0.6	1.0	0	1703	64	0.5	0.5	1.0
delf002	2009	64	0.4	0.7	1.1	0	2009	64	0.2	0.3	0.5
delf003	3118	128	0.6	7.9	8.6	1	3180	64	0.8	0.9	1.6
delf004	2624	128	0.6	6.7	7.3	1	2704	64	0.3	0.9	1.2
delf005	3226	128	0.7	7.5	8.3	1	3294	64	0.8	0.9	1.7
delf006	3066	128	0.6	8.8	9.3	1	3125	64	0.3	1.6	1.9
delf007	2757	128	0.5	9.5	10.0	1	2826	64	0.3	1.1	1.5
delf008	3393	128	0.7	9.8	10.5	2	3493	64	0.6	1.5	2.0
delf009	3307	128	0.6	11.2	11.8	1	3383	64	0.8	1.7	2.6
delf010	3132	128	0.7	8.6	9.3	2	3209	64	0.4	1.5	1.9
delf011	3030	128	0.6	8.5	9.1	2	3104	64	0.4	0.8	1.2
delf012	2890	128	0.6	8.9	9.4	1	2963	64	0.5	1.1	1.6
delf013	3103	128	0.6	9.4	10.0	1	3184	64	0.5	1.4	1.9
delf014	4257	128	0.8	7.7	8.5	1	4322	64	0.9	0.9	1.8
delf015	3176	128	0.7	9.9	10.7	1	3242	64	0.4	0.9	1.3
delf017	3075	128	0.5	9.8	10.3	1	3131	64	0.4	1.0	1.4
delf018	3281	128	0.5	10.9	11.4	1	3335	64	0.4	0.6	1.0
delf019	3160	64	0.6	0.7	1.3	0	3160	64	0.6	0.5	1.1
delf020	3701	128	0.6	10.7	11.3	2	3784	64	0.6	0.5	1.1
delf021	3247	128	0.5	10.3	10.9	1	3331	64	0.3	0.7	1.1
delf022	3673	128	0.6	11.3	11.9	2	3757	64	0.6	0.5	1.1
delf023	4603	128	0.8	11.4	12.1	2	4747	64	0.6	0.8	1.4
delf024	3973	128	0.8	10.1	10.8	1	4150	64	0.7	1.5	2.1
delf025	4032	128	0.7	8.9	9.6	2	4152	64	0.8	1.0	1.8
delf026	3378	128	0.8	9.8	10.6	1	3495	64	0.5	0.8	1.3
delf027	3402	128	0.8	9.2	9.9	2	3519	64	0.5	0.6	1.2
delf028	3269	128	0.6	9.1	9.6	2	3425	64	0.4	0.7	1.1
delf029	2949	128	0.6	10.3	10.9	1	3085	64	0.5	0.9	1.3
delf030	3002	128	0.6	11.0	11.6	1	3135	64	0.6	0.9	1.6
delf031	2835	128	0.6	8.8	9.4	1	2969	64	0.5	0.8	1.3
delf032	2958	128	0.5	12.3	12.8	1	3091	64	0.6	0.8	1.4
delf033	2210	128	0.4	8.7	9.1	1	2344	64	0.3	1.0	1.3
delf034	2699	128	0.6	9.2	9.7	1	2832	64	0.4	0.8	1.2
delf035	2407	128	0.5	11.3	11.8	2	2545	64	0.4	0.7	1.1
delf036	2522	128	0.5	9.7	10.2	2	2655	64	0.4	0.7	1.1

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
deter0	3725	64	0.2	0.1	0.3	0	3725	64	0.2	0.1	0.3
deter1	9022	64	0.6	0.3	0.9	0	9022	64	0.8	0.1	0.9
deter2	11017	64	0.6	0.3	0.9	0	11017	64	0.8	0.3	1.2
deter3	12348	64	0.8	0.4	1.2	0	12348	64	1.0	0.2	1.3
deter4	5633	64	0.3	0.2	0.4	0	5633	64	0.2	0.1	0.3
deter5	9277	64	0.6	0.3	0.8	0	9277	64	0.8	0.3	1.1
deter6	7482	64	0.4	0.2	0.6	0	7482	64	0.3	0.1	0.5
deter7	11149	64	0.8	0.4	1.1	0	11149	64	0.8	0.3	1.1
deter8	6977	64	0.4	0.2	0.6	0	6977	64	0.3	0.2	0.5
df2177	1391	64	0.7	0.5	1.1	0	1391	64	0.7	0.5	1.1
dfl001	30478	64	18.5	1.4	19.9	0	30478	64	18.6	1.5	20.1
dfn-gwin-UUM	373	64	0.0	0.0	0.0	0	373	64	0.0	0.0	0.0
dg012142	11646	64	2.0	0.2	2.2	0	11646	64	1.9	0.1	2.0
disctom	13965	64	1.7	0.2	1.9	0	13965	64	1.7	0.2	1.9
disp3	490	64	0.0	0.1	0.1	0	490	64	0.0	0.1	0.1
dolom1	19382	64	7.3	0.5	7.9	0	19382	64	7.5	0.5	8.0
ds	13089	64	17.8	3.1	20.9	0	13089	64	19.0	3.1	22.1
ds-big	44732	64	604.9	25.3	630.2	0	44732	64	610.6	25.3	635.9
dsbmip	2179	64	0.1	0.1	0.2	0	2179	64	0.2	0.0	0.2
e18	12395	64	4.6	0.6	5.2	0	12395	64	4.8	0.8	5.6
e226	402	64	0.0	0.1	0.1	0	402	64	0.0	0.0	0.0
egout	96	64	0.0	0.0	0.0	0	96	64	0.0	0.0	0.0
eil33-2	308	64	0.1	0.2	0.3	0	308	64	0.2	0.1	0.4
eilA101-2	5338	64	20.6	1.8	22.4	0	5338	64	21.9	1.8	23.7
eilB101	1459	64	0.3	0.1	0.4	0	1459	64	0.4	0.1	0.4
enigma	44	64	0.0	0.0	0.0	0	44	64	0.0	0.0	0.0
enlight13	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
enlight14	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
enlight15	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
enlight16	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
enlight9	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
etamacro	766	128	0.0	0.5	0.6	1	771	64	0.1	0.0	0.1
ex10	115687	64	1791.6	41.6	1833.3	0	115687	64	1799.0	42.7	1841.8
ex1010-pi	19385	64	10.0	2.3	12.3	0	19385	64	10.1	2.3	12.4
ex3sta1	7713	64	4.8	83.6	88.4	0	7713	64	5.0	84.0	89.0
ex9	57559	64	349.2	12.4	361.6	0	57559	64	348.8	12.4	361.2
f2000	40611	64	60.6	46.9	107.5	0	40611	64	60.2	46.8	107.0
farm	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
fast0507	13213	64	11.4	0.7	12.1	0	13213	64	11.9	0.7	12.6
fffff800	855	64	0.1	0.1	0.1	0	855	64	0.0	0.0	0.1
fiball	2818	64	0.4	0.4	0.8	0	2818	64	0.7	0.6	1.2
fiber	278	64	0.0	0.0	0.1	0	278	64	0.0	0.0	0.0
finnis	524	64	0.0	0.0	0.1	0	524	64	0.0	0.0	0.0
fit1d	1006	64	0.1	0.1	0.1	0	1006	64	0.0	0.0	0.1
fit1p	3573	64	0.4	0.1	0.4	0	3573	64	0.4	0.1	0.5
fit2d	10781	64	1.9	0.5	2.4	0	10781	64	2.1	0.4	2.5
fit2p	16070	64	3.8	0.5	4.3	0	16070	64	4.0	0.2	4.3
fixnet6	184	64	0.0	0.0	0.0	0	184	64	0.0	0.0	0.0
flugpl	11	64	0.0	0.0	0.0	0	11	64	0.0	0.0	0.0
fome11	45759	128	38.5	2932.2	2970.7	1	46151	64	39.9	1.7	41.6
fome12	93828	128	108.4	5503.8	5612.2	1	95445	64	115.1	3.5	118.6
fome13	175861	128	309.8	7200.0	7509.8	1	177979	64	321.1	7.5	328.6
fome20	37294	64	13.1	1.0	14.1	0	37294	64	13.3	1.0	14.3
fome21	81051	64	77.4	2.3	79.7	0	81051	64	76.9	2.3	79.2
forplan	369	64	0.0	0.1	0.1	0	369	64	0.0	0.0	0.1

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
fxm2-16	6817	64	0.6	0.3	0.8	0	6817	64	0.8	0.1	1.0
fxm2-6	2277	64	0.1	0.1	0.3	0	2277	64	0.1	0.1	0.2
fxm3_16	48534	64	22.2	7200.0	7222.2	0	48534	64	22.7	7200.0	7222.7
fxm3_6	9986	64	0.8	7200.0	7200.8	0	9986	64	1.1	7200.0	7201.1
fxm4_6	25805	64	4.9	7200.0	7204.9	0	25805	64	5.7	7200.0	7205.7
g200x740i	721	64	0.0	0.0	0.0	0	721	64	0.0	0.0	0.0
gams10a	38	64	0.0	0.0	0.0	1	44	64	0.0	0.0	0.0
gams30a	146	128	0.0	0.1	0.1	1	184	64	0.0	0.0	0.0
ganges	1291	64	0.1	0.1	0.1	0	1291	64	0.1	0.1	0.2
ge	11171	128	2.4	94.9	97.3	1	11172	64	2.5	1.2	3.7
gen	384	64	0.0	0.0	0.1	0	384	64	0.0	0.0	0.0
gen1	11914	128	12.1	2143.3	2155.4	1	12282	64	12.4	297.3	309.7
gen2	12239	64	25.4	7200.0	7225.4	0	12239	64	26.4	7200.0	7226.4
gen4	14397	64	56.3	7200.0	7256.3	1	14823	64	57.2	7200.0	7257.2
ger50_17_trans	4819	64	1.3	0.3	1.6	0	4819	64	1.6	0.3	1.9
germanrr	8775	64	2.7	0.4	3.1	0	8775	64	2.8	0.4	3.2
germany50-DBM	8399	64	1.2	0.1	1.2	0	8399	64	0.9	0.0	1.0
gesa2	1118	64	0.0	0.1	0.1	0	1118	64	0.0	0.0	0.1
gesa2-o	653	64	0.0	0.0	0.1	0	653	64	0.0	0.0	0.1
gesa2_o	653	64	0.0	0.1	0.1	0	653	64	0.0	0.0	0.1
gesa3	974	64	0.1	0.1	0.1	0	974	64	0.1	0.1	0.1
gesa3_o	530	64	0.0	0.1	0.1	0	530	64	0.0	0.0	0.1
gfrd-pnc	664	64	0.0	0.0	0.1	0	664	64	0.1	0.0	0.1
glass4	73	64	0.0	0.0	0.0	0	73	64	0.0	0.0	0.0
gmu-35-40	316	64	0.0	0.0	0.1	0	316	64	0.0	0.0	0.1
gmu-35-50	359	64	0.0	0.0	0.1	0	359	64	0.1	0.0	0.1
gmut-75-50	6042	64	8.1	1.8	9.9	0	6042	64	8.5	1.8	10.3
gmut-77-40	4047	64	1.7	0.6	2.2	0	4047	64	2.0	0.4	2.4
go19	2606	64	0.2	0.6	0.8	0	2606	64	0.1	0.3	0.4
gr4x6	36	64	0.0	0.0	0.0	0	36	64	0.0	0.0	0.0
greenbea	18382	64	4.7	0.8	5.5	0	18382	64	4.8	0.6	5.5
greenbeb	11957	64	2.5	1.9	4.4	0	11957	64	2.5	1.8	4.3
grow15	2102	64	0.2	0.8	1.0	0	2102	64	0.1	0.7	0.8
grow22	3334	64	0.4	3.2	3.5	0	3334	64	0.2	3.2	3.4
grow7	1071	64	0.1	0.6	0.7	0	1071	64	0.0	0.3	0.3
gt2	19	64	0.0	0.0	0.0	0	19	64	0.0	0.0	0.0
hanoi5	7389	64	1.6	0.3	1.9	0	7389	64	1.7	0.3	2.0
haprp	1303	64	0.0	0.1	0.1	0	1303	64	0.0	0.1	0.1
harp2	323	64	0.0	0.1	0.1	0	323	64	0.0	0.1	0.1
i_n13	948541	64	2586.7	6.8	2593.5	0	948541	64	2571.1	6.8	2577.8
ic97_potential	339	64	0.0	0.0	0.0	0	339	64	0.0	0.0	0.0
iiasa	1562	64	0.1	0.1	0.1	0	1562	64	0.1	0.1	0.1
iis-100-0-cov	1279	64	0.5	0.2	0.7	0	1279	64	0.5	0.2	0.7
iis-bupa-cov	4196	64	1.1	0.3	1.5	0	4196	64	1.1	0.3	1.5
iis-pima-cov	3532	64	1.2	0.5	1.7	0	3532	64	1.1	0.5	1.6
israel	149	64	0.0	0.0	0.0	0	149	64	0.0	0.0	0.0
ivu06-big	*37753*	*64*	*3149.6*	*7200.0*	*10349.6*	*0*	*37753*	*64*	*3170.1*	*7200.0*	*10370.1*
ivu52	18983	64	107.6	8.4	116.0	0	18983	64	110.3	8.4	118.7
janos-us-DDM	1042	64	0.0	0.0	0.1	0	1042	64	0.0	0.0	0.1
jendrec1	11116	64	2.4	1.9	4.3	0	11116	64	2.8	1.9	4.7
k16x240	39	64	0.0	0.0	0.0	0	39	64	0.0	0.0	0.0
kb2	58	64	0.0	0.0	0.0	0	58	64	0.0	0.0	0.0
ken-07	2777	64	0.1	0.1	0.3	0	2777	64	0.2	0.1	0.2
ken-11	17739	64	2.7	0.5	3.3	0	17739	64	3.0	0.6	3.7
ken-13	41646	64	20.1	1.0	21.0	0	41646	64	20.5	1.4	21.9

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
ken-18	192333	64	500.8	5.7	506.5	0	192333	64	501.6	5.5	507.1
kent	1680	64	0.3	0.5	0.9	0	1680	64	0.5	0.5	1.0
khb05250	119	64	0.0	0.0	0.0	0	119	64	0.0	0.0	0.0
kl02	263	64	0.2	0.4	0.5	0	263	64	0.4	0.4	0.8
kleemin3	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
kleemin4	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
kleemin5	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
kleemin6	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
kleemin7	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
kleemin8	0	64	0.0	0.0	0.0	0	0	64	0.0	0.0	0.0
l152lav	700	64	0.1	0.0	0.1	0	700	64	0.1	0.0	0.1
l9	746	64	0.1	0.3	0.4	0	746	64	0.1	0.3	0.5
large000	3748	64	0.6	0.8	1.5	0	3748	64	0.7	0.9	1.6
large001	7844	64	1.4	1.3	2.7	0	7844	64	1.4	1.1	2.6
large002	3866	128	0.8	17.6	18.4	2	4065	64	0.6	1.4	2.1
large003	4179	128	0.8	15.1	15.9	1	4254	64	0.7	1.3	2.0
large004	4647	128	1.0	13.4	14.4	2	4681	64	0.8	2.2	3.0
large005	4501	128	0.8	14.2	15.0	1	4567	64	0.8	1.2	1.9
large006	4864	128	0.8	15.2	16.0	2	4942	64	0.7	1.3	2.0
large007	5010	128	0.8	15.1	16.0	1	5094	64	0.9	1.7	2.6
large008	5203	128	0.8	15.0	15.9	1	5291	64	0.9	1.8	2.7
large009	4988	128	0.9	16.4	17.3	1	5074	64	1.1	2.3	3.4
large010	4639	128	0.7	14.7	15.4	1	4725	64	0.8	1.4	2.3
large011	5135	128	0.9	16.4	17.4	1	5221	64	0.9	1.5	2.4
large012	4924	128	0.8	17.5	18.3	1	5009	64	0.9	1.5	2.4
large013	4975	128	0.8	16.2	17.0	1	5062	64	0.7	0.9	1.6
large014	5082	128	0.9	14.8	15.6	1	5148	64	0.9	1.2	2.1
large015	4318	128	0.7	15.0	15.8	1	4380	64	0.6	1.1	1.7
large016	4571	128	0.6	13.4	14.0	1	4633	64	0.8	1.1	1.9
large017	3980	64	0.8	1.2	2.0	0	3980	64	0.8	1.1	1.9
large018	4459	64	0.8	1.0	1.8	0	4459	64	0.8	0.7	1.6
large019	4909	64	0.8	0.9	1.7	0	4909	64	0.8	0.6	1.4
large020	6984	128	1.1	16.1	17.2	2	7059	64	0.9	1.1	2.0
large021	6201	128	0.9	20.4	21.3	2	6288	64	0.9	1.0	2.0
large022	6907	128	0.9	20.0	20.9	2	6993	64	0.8	0.7	1.5
large023	4224	128	0.9	18.0	19.0	2	4398	64	0.7	1.3	2.0
large024	5788	128	1.0	17.3	18.3	2	5988	64	1.2	1.3	2.5
large025	4811	128	0.8	19.1	19.9	2	5000	64	1.0	2.2	3.2
large026	4196	128	0.8	18.4	19.3	2	4367	64	0.7	1.6	2.2
large027	4172	128	0.8	16.4	17.3	2	4353	64	0.8	0.9	1.7
large028	4691	128	1.1	19.1	20.1	1	4906	64	0.8	1.1	1.9
large029	4158	128	0.8	19.5	20.3	2	4372	64	0.6	1.7	2.4
large030	3732	128	0.6	18.7	19.3	2	3930	64	0.7	1.1	1.8
large031	3729	128	0.6	18.7	19.3	1	3931	64	0.6	2.0	2.6
large032	4851	128	0.9	21.6	22.4	1	5052	64	0.9	3.1	4.0
large033	3675	128	0.7	16.5	17.2	2	3877	64	0.5	1.2	1.7
large034	4009	128	0.7	21.1	21.8	2	4201	64	0.7	2.7	3.4
large035	3450	128	0.8	22.1	22.9	1	3655	64	0.7	3.0	3.6
large036	3111	128	0.6	21.5	22.1	2	3314	64	0.6	2.4	3.0
lectsched-1	7	64	1.1	0.7	1.8	0	7	64	1.3	0.7	2.0
lectsched-1-obj	963	64	1.2	0.8	2.0	0	963	64	1.5	0.8	2.2
lectsched-2	3	64	0.5	0.3	0.8	0	3	64	0.7	0.4	1.0
lectsched-3	7	64	0.9	0.9	1.8	0	7	64	1.1	0.6	1.7
lectsched-4-obj	174	64	0.3	0.3	0.6	0	174	64	0.2	0.1	0.4
leo1	862	64	0.3	0.2	0.5	0	862	64	0.5	0.2	0.7

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
leo2	1637	64	0.6	0.3	0.9	0	1637	64	0.9	0.5	1.4
liu	543	64	0.1	0.0	0.1	0	543	64	0.1	0.0	0.1
lo10	953870	64	5941.4	3.8	5945.2	0	953870	64	5931.8	3.8	5935.7
long15	229488	64	2753.1	8.6	2761.7	0	229488	64	2738.1	8.5	2746.6
lotfi	226	64	0.0	0.0	0.0	0	226	64	0.0	0.0	0.0
lotsize	1460	64	0.0	0.0	0.1	0	1460	64	0.1	0.0	0.1
lp22	38451	64	35.6	5.4	41.0	0	38451	64	35.8	5.3	41.1
lpl1	36759	64	59.4	1.7	61.1	0	36759	64	60.0	1.4	61.4
lpl2	1465	64	0.2	0.1	0.3	0	1465	64	0.1	0.1	0.2
lpl3	5040	64	1.1	0.3	1.3	0	5040	64	1.2	0.2	1.4
lrn	11450	128	2.6	167.8	170.4	1	11452	64	2.5	0.3	2.9
lrsa120	9787	64	2.5	1.6	4.1	1	9789	64	2.4	0.2	2.7
lseu	25	64	0.0	0.0	0.0	0	25	64	0.0	0.0	0.0
m100n500k4r1	174	64	0.0	0.0	0.1	0	174	64	0.0	0.0	0.0
macrophage	706	64	0.0	0.0	0.1	0	706	64	0.0	0.0	0.1
manna81	3018	64	0.1	0.1	0.2	0	3018	64	0.1	0.1	0.2
map06	23840	64	39.4	5.9	45.3	0	23840	64	40.6	5.7	46.3
map10	23747	64	40.6	5.5	46.1	0	23747	64	41.5	5.6	47.1
map14	23178	64	38.3	5.5	43.9	0	23178	64	39.4	5.6	44.9
map18	20964	64	33.6	5.5	39.1	0	20964	64	34.7	5.5	40.2
map20	19686	64	33.1	5.7	38.8	0	19686	64	30.9	5.9	36.8
markshare1	35	64	0.0	0.0	0.0	0	35	64	0.0	0.0	0.0
markshare2	43	64	0.0	0.0	0.0	0	43	64	0.0	0.0	0.0
markshare_5_0	24	64	0.0	0.0	0.0	0	24	64	0.0	0.0	0.0
maros	1255	64	0.1	0.2	0.3	0	1255	64	0.2	0.1	0.3
maros-r7	7953	192	2.9	787.1	790.0	0	7953	192	3.5	785.2	788.7
mas74	224	64	0.0	0.0	0.0	0	224	64	0.0	0.0	0.0
mas76	132	64	0.0	0.0	0.0	0	132	64	0.0	0.0	0.0
maxgasflow	6737	64	0.6	0.2	0.8	0	6737	64	0.8	0.1	0.9
mc11	1239	64	0.1	0.0	0.1	0	1239	64	0.1	0.0	0.1
mcf2	2763	64	0.3	0.1	0.4	0	2763	64	0.1	0.0	0.2
mcsched	2546	64	0.3	0.1	0.4	0	2546	64	0.3	0.1	0.4
methanosarcina	655	64	0.1	0.2	0.3	0	655	64	0.2	0.2	0.3
mik-250-1-100-1	100	64	0.0	0.0	0.0	0	100	64	0.0	0.0	0.0
mine-166-5	1642	64	0.5	0.2	0.8	0	1642	64	0.5	0.2	0.8
mine-90-10	1948	64	0.5	0.2	0.7	0	1948	64	0.4	0.2	0.6
misc03	45	64	0.0	0.0	0.0	0	45	64	0.0	0.0	0.0
misc06	816	64	0.1	0.1	0.1	0	816	64	0.1	0.1	0.1
misc07	157	64	0.0	0.0	0.1	0	157	64	0.0	0.0	0.1
mitre	2451	64	0.3	0.2	0.5	0	2451	64	0.4	0.2	0.6
mkc	538	64	0.1	0.1	0.1	0	538	64	0.1	0.1	0.2
mkc1	538	64	0.1	0.1	0.2	0	538	64	0.0	0.1	0.1
mod008	27	64	0.0	0.0	0.0	0	27	64	0.0	0.0	0.0
mod010	1062	64	0.1	0.0	0.2	0	1062	64	0.2	0.0	0.2
mod011	6153	64	0.9	0.2	1.1	0	6153	64	0.8	0.2	1.0
mod2	58340	128	100.4	7200.0	7300.4	11	58534	64	103.8	4.3	108.1
model1	180	128	0.0	0.2	0.2	1	182	64	0.0	0.0	0.0
model10	44687	64	22.8	35.0	57.8	0	44687	64	23.2	34.9	58.1
model11	5273	64	1.0	0.5	1.5	0	5273	64	1.0	0.3	1.3
model2	3466	64	0.4	0.2	0.6	0	3466	64	0.2	0.2	0.4
model3	9208	64	1.3	1.5	2.8	0	9208	64	1.3	1.5	2.8
model4	15098	64	3.0	1.8	4.8	0	15098	64	2.8	1.8	4.6
model5	19877	64	3.8	1.5	5.3	0	19877	64	4.1	1.5	5.6
model6	14770	64	3.3	12.1	15.4	0	14770	64	3.2	12.1	15.3
model7	16070	64	4.0	8.1	12.1	0	16070	64	4.0	7.8	11.8

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
model8	2522	64	0.3	7200.0	7200.4	1	2538	64	0.4	7200.0	7200.4
model9	12397	64	2.3	1.8	4.0	0	12397	64	2.5	1.8	4.2
modglob	359	64	0.0	0.0	0.0	0	359	64	0.0	0.0	0.0
modszk1	653	64	0.0	0.1	0.1	0	653	64	0.0	0.0	0.1
momentum1	3305	64	0.8	4.1	4.9	1	3306	64	1.1	1.5	2.6
momentum2	45882	128	18.8	240.8	259.6	2	45950	64	19.7	2.6	22.2
momentum3	46184	128	189.5	7200.0	7389.5	1	46185	64	192.0	1833.5	2025.5
msc98-ip	9496	128	1.3	862.3	863.6	1	9549	64	1.7	0.6	2.3
mspp16	*52*	*64*	*19.3*	*7200.0*	*7219.3*	*0*	*52*	*64*	*20.0*	*7200.0*	*7220.0*
multi	59	64	0.0	0.0	0.0	0	59	64	0.0	0.0	0.0
mzzv11	37921	64	26.4	0.5	26.8	0	37921	64	26.2	0.3	26.6
mzzv42z	34373	64	17.6	0.4	18.1	0	34373	64	17.2	0.5	17.7
n15-3	43662	64	114.6	3.4	118.0	0	43662	64	115.3	3.4	118.7
n3-3	2965	64	0.6	0.2	0.8	0	2965	64	0.5	0.1	0.6
n3700	8698	64	1.6	0.2	1.9	0	8698	64	1.6	0.2	1.8
n3701	8106	64	1.3	0.2	1.5	0	8106	64	1.3	0.2	1.5
n3702	7987	64	1.2	0.2	1.5	0	7987	64	1.1	0.2	1.3
n3703	7397	64	1.1	0.2	1.3	0	7397	64	1.0	0.2	1.2
n3704	7325	64	1.1	0.2	1.2	0	7325	64	0.9	0.2	1.2
n3705	6974	64	1.3	0.2	1.5	0	6974	64	1.2	0.2	1.4
n3706	7305	64	1.1	0.2	1.3	0	7305	64	1.1	0.2	1.4
n3707	8681	64	1.2	0.2	1.4	0	8681	64	1.4	0.2	1.6
n3708	7133	64	1.6	0.2	1.8	0	7133	64	1.2	0.2	1.4
n3709	8030	64	1.5	0.2	1.7	0	8030	64	1.4	0.2	1.7
n370a	8608	64	1.2	0.2	1.5	0	8608	64	1.3	0.2	1.5
n370b	8553	64	1.4	0.2	1.6	0	8553	64	1.2	0.2	1.4
n370c	7273	64	1.1	0.2	1.3	0	7273	64	0.9	0.2	1.2
n370d	7273	64	1.1	0.2	1.3	0	7273	64	0.9	0.2	1.2
n370e	7852	64	1.2	0.2	1.4	0	7852	64	1.0	0.2	1.2
n3div36	306	64	0.4	0.4	0.8	0	306	64	0.7	0.4	1.1
n3seq24	4646	64	14.1	10.5	24.6	1	4720	64	15.9	5.7	21.7
n4-3	1341	64	0.2	0.0	0.2	0	1341	64	0.2	0.0	0.2
n9-3	3531	64	0.6	0.1	0.6	0	3531	64	0.5	0.1	0.6
nag	2476	64	0.2	0.1	0.2	0	2476	64	0.1	0.1	0.2
nemsafm	755	64	0.0	0.0	0.1	0	755	64	0.0	0.0	0.1
nemscem	932	64	0.1	0.1	0.1	0	932	64	0.1	0.1	0.1
nemsemm1	10608	64	3.6	2.6	6.2	0	10608	64	5.4	2.7	8.1
nemsemm2	10445	64	1.5	0.7	2.2	0	10445	64	1.9	0.6	2.6
nemspmm1	15049	64	4.3	1.7	6.0	0	15049	64	4.5	1.8	6.3
nemspmm2	15056	128	4.2	223.0	227.2	1	15058	64	4.4	8.6	13.0
nemswrld	35922	64	32.6	160.3	192.9	0	35922	64	33.7	160.3	194.0
neos	89942	64	497.3	22.5	519.8	1	90253	64	497.8	13.1	511.0
neos-1053234	257	64	0.1	0.1	0.2	0	257	64	0.0	0.1	0.1
neos-1053591	821	64	0.0	0.0	0.1	0	821	64	0.0	0.0	0.0
neos-1056905	140	64	0.0	0.0	0.0	0	140	64	0.0	0.0	0.0
neos-1058477	48	64	0.0	0.1	0.1	0	48	64	0.0	0.0	0.1
neos-1061020	16447	64	7.3	0.4	7.7	0	16447	64	7.5	0.6	8.0
neos-1062641	873	128	0.0	0.4	0.5	1	902	64	0.0	0.0	0.0
neos-1067731	12132	64	2.4	0.2	2.7	0	12132	64	2.5	0.2	2.8
neos-1096528	115	64	47.5	20.5	68.0	0	115	64	54.0	20.3	74.4
neos-1109824	115	64	0.5	0.3	0.7	0	115	64	0.6	0.1	0.7
neos-1112782	622	64	0.1	0.1	0.1	0	622	64	0.0	0.0	0.1
neos-1112787	557	64	0.1	0.1	0.1	0	557	64	0.0	0.0	0.1
neos-1120495	96	64	0.2	0.2	0.4	0	96	64	0.3	0.2	0.5
neos-1121679	35	64	0.0	0.0	0.0	0	35	64	0.0	0.0	0.0

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
neos-1122047	5499	64	1.2	1.0	2.2	0	5499	64	1.5	0.8	2.3
neos-1126860	5216	64	0.9	0.5	1.4	0	5216	64	1.0	0.5	1.5
neos-1140050	13327	64	30.1	1115.7	1145.8	0	13327	64	32.6	1113.4	1146.0
neos-1151496	4930	64	0.8	0.1	0.9	0	4930	64	0.8	0.1	0.9
neos-1171448	4075	64	0.6	0.5	1.1	0	4075	64	0.5	0.3	0.8
neos-1171692	1290	64	0.1	0.1	0.2	0	1290	64	0.1	0.1	0.2
neos-1171737	2060	64	0.3	0.1	0.4	0	2060	64	0.2	0.1	0.3
neos-1173026	33	64	0.0	0.0	0.0	0	33	64	0.0	0.0	0.1
neos-1200887	294	64	0.0	0.1	0.1	0	294	64	0.0	0.0	0.1
neos-1208069	3008	64	0.6	0.1	0.7	0	3008	64	0.6	0.1	0.7
neos-1208135	2364	64	0.4	0.1	0.5	0	2364	64	0.4	0.1	0.6
neos-1211578	187	64	0.0	0.0	0.0	0	187	64	0.0	0.0	0.0
neos-1215259	6319	64	1.0	0.1	1.1	0	6319	64	1.0	0.1	1.1
neos-1215891	3541	64	0.9	0.2	1.1	0	3541	64	0.7	0.2	1.0
neos-1223462	13116	64	3.7	0.3	4.0	0	13116	64	3.6	0.2	3.9
neos-1224597	10673	64	1.8	0.1	1.9	0	10673	64	1.5	0.1	1.6
neos-1225589	438	64	0.0	0.0	0.1	0	438	64	0.0	0.0	0.1
neos-1228986	197	64	0.0	0.0	0.0	0	197	64	0.0	0.0	0.0
neos-1281048	2276	64	0.2	0.0	0.3	0	2276	64	0.3	0.1	0.3
neos-1311124	639	64	0.0	0.0	0.1	0	639	64	0.0	0.0	0.1
neos-1324574	6564	64	0.8	0.2	1.0	0	6564	64	0.9	0.2	1.1
neos-1330346	1983	64	0.3	0.1	0.4	0	1983	64	0.1	0.1	0.2
neos-1330635	141	64	0.0	0.1	0.1	0	141	64	0.0	0.1	0.1
neos-1337307	10964	64	2.1	0.3	2.3	0	10964	64	2.2	0.3	2.5
neos-1337489	187	64	0.0	0.0	0.0	0	187	64	0.0	0.0	0.0
neos-1346382	350	64	0.0	0.0	0.0	0	350	64	0.0	0.0	0.0
neos-1354092	12518	64	7.0	28.2	35.2	0	12518	64	7.1	28.2	35.4
neos-1367061	18400	64	23.5	1.4	24.9	0	18400	64	24.1	1.4	25.5
neos-1396125	4834	64	0.6	0.1	0.7	0	4834	64	0.7	0.0	0.8
neos-1407044	27160	64	31.7	71.6	103.3	0	27160	64	32.1	67.7	99.8
neos-1413153	1027	64	0.2	0.1	0.3	0	1027	64	0.2	0.0	0.2
neos-1415183	1763	64	0.3	0.1	0.4	0	1763	64	0.4	0.1	0.4
neos-1417043	13642	64	57.3	3.9	61.2	0	13642	64	57.2	3.9	61.1
neos-1420205	944	128	0.0	0.5	0.5	0	944	64	0.0	0.0	0.0
neos-1420546	102727	64	92.3	2.1	94.4	0	102727	64	92.2	2.1	94.3
neos-1420790	12930	64	2.5	0.4	2.9	0	12930	64	2.4	0.3	2.6
neos-1423785	19268	64	3.1	0.6	3.8	0	19268	64	3.6	0.5	4.0
neos-1425699	26	64	0.0	0.0	0.0	0	26	64	0.0	0.0	0.0
neos-1426635	350	64	0.0	0.0	0.0	0	350	64	0.0	0.0	0.0
neos-1426662	665	64	0.0	0.0	0.1	0	665	64	0.0	0.0	0.0
neos-1427181	591	64	0.0	0.0	0.0	0	591	64	0.0	0.0	0.0
neos-1427261	883	64	0.1	0.0	0.1	0	883	64	0.1	0.0	0.1
neos-1429185	466	64	0.0	0.0	0.0	0	466	64	0.0	0.0	0.0
neos-1429212	28835	64	220.5	10.5	231.0	0	28835	64	218.8	10.3	229.1
neos-1429461	399	64	0.0	0.0	0.0	0	399	64	0.0	0.0	0.0
neos-1430701	257	64	0.0	0.0	0.0	0	257	64	0.0	0.0	0.0
neos-1430811	32707	64	340.4	17.1	357.6	0	32707	64	338.3	17.3	355.6
neos-1436709	505	64	0.0	0.0	0.0	0	505	64	0.0	0.0	0.0
neos-1436713	835	64	0.0	0.0	0.1	0	835	64	0.1	0.0	0.1
neos-1437164	296	64	0.0	0.0	0.1	0	296	64	0.1	0.0	0.1
neos-1439395	312	64	0.0	0.0	0.0	0	312	64	0.0	0.0	0.1
neos-1440225	486	64	0.1	0.1	0.2	0	486	64	0.1	0.1	0.2
neos-1440447	239	64	0.0	0.0	0.0	0	239	64	0.0	0.0	0.0
neos-1440457	820	64	0.0	0.0	0.1	0	820	64	0.1	0.0	0.1
neos-1440460	359	64	0.0	0.0	0.0	0	359	64	0.0	0.0	0.0

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
neos-1441553	288	64	0.0	0.1	0.1	0	288	64	0.1	0.0	0.1
neos-1442119	607	64	0.0	0.0	0.1	0	607	64	0.1	0.0	0.1
neos-1442657	498	64	0.0	0.0	0.0	0	498	64	0.0	0.0	0.0
neos-1445532	14103	64	1.7	0.2	1.9	0	14103	64	1.8	0.2	2.0
neos-1445738	15164	64	2.8	0.3	3.1	0	15164	64	2.9	0.2	3.1
neos-1445743	16103	64	3.2	0.2	3.4	0	16103	64	3.1	0.2	3.3
neos-1445755	15634	64	3.1	0.2	3.4	0	15634	64	3.2	0.2	3.4
neos-1445765	15960	64	3.3	0.2	3.6	0	15960	64	3.4	0.2	3.6
neos-1451294	10087	64	1.8	0.6	2.4	0	10087	64	1.6	0.6	2.2
neos-1456979	957	64	0.4	0.1	0.4	0	957	64	0.2	0.1	0.3
neos-1460246	276	64	0.0	0.0	0.0	0	276	64	0.0	0.0	0.1
neos-1460265	808	64	0.1	0.1	0.1	0	808	64	0.1	0.0	0.1
neos-1460543	9199	64	1.3	0.1	1.5	0	9199	64	1.2	0.1	1.4
neos-1460641	10168	64	1.1	0.1	1.2	0	10168	64	1.1	0.0	1.1
neos-1461051	418	64	0.0	0.0	0.1	0	418	64	0.1	0.0	0.1
neos-1464762	10563	64	1.2	0.1	1.3	0	10563	64	1.2	0.1	1.3
neos-1467067	673	64	0.0	0.0	0.0	0	673	64	0.0	0.0	0.0
neos-1467371	8604	64	1.1	0.1	1.2	0	8604	64	0.9	0.1	1.0
neos-1467467	8764	64	0.9	0.1	1.0	0	8764	64	0.9	0.1	1.0
neos-1480121	86	64	0.0	0.0	0.0	0	86	64	0.0	0.0	0.0
neos-1489999	835	64	0.1	0.0	0.1	0	835	64	0.1	0.0	0.1
neos-1516309	134	64	0.0	0.0	0.1	0	134	64	0.1	0.1	0.1
neos-1582420	2433	64	0.5	0.2	0.8	0	2433	64	0.7	0.2	0.9
neos-1593097	921	64	0.6	0.2	0.8	0	921	64	0.7	0.2	1.0
neos-1595230	518	64	0.0	0.0	0.1	0	518	64	0.0	0.0	0.0
neos-1597104	150	64	0.5	1.3	1.8	0	150	64	0.7	1.3	2.0
neos-1599274	357	64	0.1	0.1	0.2	0	357	64	0.1	0.1	0.1
neos-1601936	14758	64	5.3	0.5	5.9	0	14758	64	5.4	0.7	6.1
neos-1603512	814	64	0.1	0.0	0.1	0	814	64	0.1	0.0	0.1
neos-1603518	2362	64	0.4	0.1	0.5	0	2362	64	0.4	0.1	0.5
neos-1605061	23399	64	11.5	0.5	12.0	0	23399	64	11.5	0.8	12.2
neos-1605075	16980	64	8.8	0.8	9.6	0	16980	64	8.8	0.7	9.4
neos-1616732	314	64	0.0	0.0	0.0	0	314	64	0.0	0.0	0.0
neos-1620770	772	64	0.1	0.1	0.1	0	772	64	0.1	0.0	0.1
neos-1620807	222	64	0.0	0.0	0.0	0	222	64	0.0	0.0	0.0
neos-1622252	807	64	0.1	0.1	0.1	0	807	64	0.1	0.0	0.1
neos-430149	240	64	0.0	0.0	0.0	0	240	64	0.0	0.0	0.0
neos-476283	5536	64	3.8	15.9	19.7	0	5536	64	4.3	16.1	20.4
neos-480878	466	64	0.1	0.2	0.2	0	466	64	0.1	0.1	0.2
neos-494568	553	64	0.2	0.1	0.3	0	553	64	0.2	0.2	0.4
neos-495307	963	64	0.3	0.1	0.3	0	963	64	0.2	0.1	0.2
neos-498623	1034	64	0.3	0.3	0.7	1	1037	64	0.6	0.2	0.8
neos-501453	37	64	0.0	0.0	0.0	0	37	64	0.0	0.0	0.0
neos-501474	288	64	0.0	0.0	0.0	0	288	64	0.0	0.0	0.0
neos-503737	4883	64	0.8	0.1	0.9	0	4883	64	0.8	0.1	0.9
neos-504674	548	64	0.0	0.0	0.1	0	548	64	0.0	0.0	0.1
neos-504815	468	64	0.0	0.0	0.1	0	468	64	0.0	0.0	0.0
neos-506422	98	64	0.1	0.1	0.1	0	98	64	0.1	0.0	0.1
neos-506428	2092	64	1.0	1.7	2.7	0	2092	64	1.1	1.7	2.8
neos-512201	543	64	0.0	0.1	0.1	0	543	64	0.0	0.0	0.1
neos-520729	32688	64	19.9	0.5	20.4	0	32688	64	20.2	0.7	20.9
neos-522351	205	64	0.0	0.0	0.0	0	205	64	0.0	0.0	0.0
neos-525149	1124	64	0.7	5.7	6.5	0	1124	64	0.8	5.9	6.7
neos-530627	68	64	0.0	0.0	0.0	0	68	64	0.0	0.0	0.0
neos-538867	169	64	0.0	0.0	0.1	1	171	64	0.0	0.0	0.1

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
neos-538916	168	64	0.0	0.0	0.0	0	168	64	0.0	0.0	0.0
neos-544324	2629	64	1.2	2.9	4.0	0	2629	64	1.5	2.8	4.3
neos-547911	1895	64	0.3	1.0	1.3	0	1895	64	0.5	1.0	1.4
neos-548047	11782	64	1.6	0.2	1.7	0	11782	64	1.8	0.2	2.0
neos-548251	2037	64	0.1	0.0	0.1	0	2037	64	0.1	0.0	0.1
neos-551991	7378	64	1.1	0.2	1.4	0	7378	64	1.1	0.2	1.3
neos-555001	10198	64	1.0	0.1	1.1	0	10198	64	0.6	0.1	0.7
neos-555298	3995	64	0.2	0.1	0.4	0	3995	64	0.1	0.1	0.3
neos-555343	11043	64	1.1	0.1	1.2	0	11043	64	0.9	0.1	0.9
neos-555424	5895	64	0.4	0.1	0.5	0	5895	64	0.2	0.1	0.3
neos-555694	565	64	0.1	0.1	0.2	0	565	64	0.1	0.0	0.1
neos-555771	575	64	0.1	0.1	0.2	0	575	64	0.1	0.0	0.1
neos-555884	7183	64	0.6	0.1	0.8	0	7183	64	0.4	0.1	0.6
neos-555927	2202	64	0.1	0.1	0.2	0	2202	64	0.1	0.1	0.1
neos-565672	87785	64	259.8	7.5	267.2	0	87785	64	260.9	7.2	268.2
neos-565815	6971	64	2.3	0.4	2.8	0	6971	64	2.4	0.5	3.0
neos-570431	2355	64	0.2	0.1	0.3	0	2355	64	0.1	0.0	0.1
neos-574665	506	64	0.1	0.1	0.2	0	506	64	0.1	0.1	0.2
neos-578379	*14389*	*64*	*13.2*	*7200.0*	*7213.2*	*0*	*14389*	*64*	*13.4*	*7200.0*	*7213.4*
neos-582605	1067	64	0.1	0.0	0.1	0	1067	64	0.1	0.0	0.1
neos-583731	602	64	0.0	0.0	0.0	0	602	64	0.0	0.0	0.0
neos-584146	603	64	0.1	0.0	0.1	0	603	64	0.1	0.0	0.1
neos-584851	612	64	0.0	0.0	0.1	0	612	64	0.1	0.0	0.1
neos-584866	11751	64	2.6	0.2	2.8	0	11751	64	2.8	0.1	2.9
neos-585192	2614	64	0.5	0.3	0.8	0	2614	64	0.7	0.3	1.0
neos-585467	1912	64	0.4	0.2	0.6	0	1912	64	0.2	0.2	0.4
neos-593853	615	64	0.0	0.1	0.1	0	615	64	0.1	0.0	0.1
neos-595904	1276	64	0.2	0.1	0.4	0	1276	64	0.3	0.1	0.4
neos-595905	390	64	0.0	0.0	0.1	0	390	64	0.1	0.0	0.1
neos-595925	521	64	0.0	0.1	0.1	0	521	64	0.1	0.0	0.1
neos-598183	1148	64	0.1	0.1	0.2	0	1148	64	0.1	0.1	0.2
neos-603073	455	64	0.0	0.0	0.0	0	455	64	0.0	0.0	0.1
neos-611135	11914	64	4.7	1.6	6.2	0	11914	64	4.7	1.6	6.2
neos-611838	2046	64	0.4	0.2	0.6	0	2046	64	0.2	0.2	0.4
neos-612125	2002	64	0.4	0.2	0.6	0	2002	64	0.3	0.2	0.5
neos-612143	2100	64	0.4	0.2	0.6	0	2100	64	0.3	0.2	0.5
neos-612162	1828	64	0.3	0.2	0.5	0	1828	64	0.3	0.2	0.5
neos-619167	6697	64	1.3	0.2	1.5	0	6697	64	1.1	0.2	1.3
neos-631164	954	64	0.1	0.0	0.1	0	954	64	0.1	0.0	0.1
neos-631517	765	64	0.0	0.0	0.1	0	765	64	0.1	0.0	0.1
neos-631694	24940	64	5.2	0.1	5.3	0	24940	64	5.2	0.1	5.3
neos-631709	70640	64	213.1	0.8	213.9	0	70640	64	213.0	0.8	213.9
neos-631710	66455	64	923.9	4.6	928.5	0	66455	64	920.8	4.6	925.4
neos-631784	26868	64	29.2	0.2	29.4	0	26868	64	29.5	0.3	29.8
neos-632335	5592	64	1.0	0.5	1.6	0	5592	64	1.3	0.3	1.6
neos-633273	5070	64	1.1	0.5	1.6	0	5070	64	1.1	0.5	1.6
neos-641591	14066	64	6.7	0.3	7.1	0	14066	64	7.1	0.3	7.4
neos-655508	119	64	0.1	0.3	0.4	0	119	64	0.1	0.2	0.3
neos-662469	14066	64	7.2	0.4	7.6	0	14066	64	7.0	0.3	7.3
neos-686190	834	64	0.2	0.1	0.3	0	834	64	0.1	0.1	0.2
neos-691058	7206	64	1.3	0.2	1.5	0	7206	64	1.2	0.1	1.3
neos-691073	6943	64	1.4	0.1	1.6	0	6943	64	1.3	0.1	1.3
neos-693347	9600	64	2.1	0.3	2.3	0	9600	64	2.2	0.3	2.5
neos-702280	12807	64	14.6	15.1	29.7	0	12807	64	15.3	15.0	30.3
neos-709469	429	64	0.0	0.0	0.1	0	429	64	0.0	0.0	0.0

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
neos-717614	1049	64	0.0	0.1	0.1	0	1049	64	0.1	0.0	0.1
neos-738098	25294	64	31.9	0.8	32.6	0	25294	64	31.8	0.7	32.6
neos-775946	1720	64	0.6	0.3	0.9	0	1720	64	0.5	0.3	0.9
neos-777800	1633	64	0.5	0.2	0.6	0	1633	64	0.3	0.1	0.4
neos-780889	57556	64	184.5	2.9	187.4	0	57556	64	185.1	3.0	188.1
neos-785899	411	64	0.1	0.1	0.1	0	411	64	0.1	0.1	0.2
neos-785912	1338	64	0.3	0.1	0.4	0	1338	64	0.3	0.1	0.4
neos-785914	232	64	0.0	0.0	0.1	0	232	64	0.0	0.0	0.1
neos-787933	14008	64	31.7	0.6	32.3	0	14008	64	32.3	0.6	32.9
neos-791021	10768	64	2.5	0.2	2.7	0	10768	64	2.4	0.2	2.7
neos-796608	379	64	0.0	0.0	0.0	0	379	64	0.0	0.0	0.0
neos-799711	19400	64	3.0	0.9	3.9	0	19400	64	3.3	0.9	4.2
neos-799838	11429	64	3.8	0.3	4.2	0	11429	64	3.8	0.2	4.0
neos-801834	1782	64	0.3	0.2	0.5	0	1782	64	0.2	0.1	0.3
neos-803219	1309	64	0.1	0.1	0.2	0	1309	64	0.1	0.0	0.1
neos-803220	888	64	0.1	0.0	0.1	0	888	64	0.1	0.0	0.1
neos-806323	1246	64	0.1	0.1	0.1	0	1246	64	0.1	0.1	0.2
neos-807454	12164	64	2.2	0.1	2.3	0	12164	64	2.3	0.1	2.4
neos-807456	11272	64	1.6	0.4	2.0	0	11272	64	1.6	0.4	2.1
neos-807639	1305	64	0.1	0.1	0.1	0	1305	64	0.1	0.0	0.1
neos-807705	1259	64	0.1	0.1	0.2	0	1259	64	0.1	0.1	0.2
neos-808072	8164	64	1.4	0.1	1.5	0	8164	64	1.4	0.1	1.5
neos-808214	1236	64	0.2	0.1	0.3	0	1236	64	0.1	0.1	0.2
neos-810286	13517	64	3.5	0.3	3.8	0	13517	64	3.5	0.3	3.8
neos-810326	8031	64	1.2	0.1	1.4	0	8031	64	1.2	0.1	1.3
neos-820146	249	64	0.0	0.0	0.0	0	249	64	0.0	0.0	0.0
neos-820157	569	64	0.0	0.0	0.1	0	569	64	0.0	0.0	0.0
neos-820879	1509	64	0.5	0.1	0.7	0	1509	64	0.7	0.1	0.8
neos-824661	9912	64	5.2	0.4	5.6	0	9912	64	5.6	0.4	6.0
neos-824695	4726	64	1.8	0.3	2.1	0	4726	64	1.8	0.1	1.9
neos-825075	1062	64	0.1	0.0	0.1	0	1062	64	0.0	0.0	0.1
neos-826224	5396	64	1.2	0.3	1.5	0	5396	64	1.4	0.4	1.8
neos-826250	5747	64	1.3	0.1	1.4	0	5747	64	1.0	0.1	1.1
neos-826650	8372	64	1.7	0.2	1.9	0	8372	64	1.9	0.1	1.9
neos-826694	10754	64	2.6	0.4	3.0	0	10754	64	2.7	0.3	3.1
neos-826812	10450	64	2.4	0.1	2.5	0	10450	64	2.4	0.1	2.6
neos-826841	3738	64	0.8	0.1	0.9	0	3738	64	0.7	0.1	0.7
neos-827015	18897	64	12.2	0.6	12.8	0	18897	64	12.8	0.6	13.3
neos-827175	11439	64	3.2	0.5	3.7	0	11439	64	3.3	0.4	3.7
neos-829552	13050	64	4.4	0.5	4.8	0	13050	64	4.8	0.4	5.3
neos-830439	179	64	0.0	0.0	0.0	0	179	64	0.0	0.0	0.0
neos-831188	8374	64	1.1	0.1	1.2	0	8374	64	1.2	0.1	1.2
neos-839838	6451	64	1.1	0.5	1.6	0	6451	64	1.4	0.4	1.8
neos-839859	2312	64	0.4	0.1	0.5	0	2312	64	0.2	0.1	0.3
neos-839894	21287	64	20.6	1.2	21.8	0	21287	64	20.8	1.2	21.9
neos-841664	5975	64	0.7	0.1	0.8	0	5975	64	0.6	0.1	0.7
neos-847051	2383	64	0.2	0.2	0.4	0	2383	64	0.3	0.1	0.4
neos-847302	2146	64	0.3	0.1	0.5	0	2146	64	0.4	0.1	0.4
neos-848150	1409	64	0.2	0.1	0.3	0	1409	64	0.2	0.0	0.3
neos-848198	11363	64	2.1	0.1	2.2	0	11363	64	2.3	0.1	2.4
neos-848589	1401	64	1.0	3.8	4.8	0	1401	64	1.9	3.7	5.6
neos-848845	7295	64	1.4	0.2	1.7	0	7295	64	1.4	0.3	1.7
neos-849702	5626	64	1.0	0.3	1.2	0	5626	64	1.0	0.3	1.3
neos-850681	25783	64	5.9	0.2	6.0	0	25783	64	5.8	0.2	6.0
neos-856059	580	64	0.1	0.1	0.3	0	580	64	0.1	0.1	0.2

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
neos-859770	294	64	0.3	1.6	1.9	0	294	64	0.3	1.6	2.0
neos-860244	90	64	0.1	0.4	0.5	0	90	64	0.1	0.5	0.6
neos-860300	349	64	0.2	0.6	0.8	0	349	64	0.3	0.6	0.9
neos-862348	1026	64	0.2	0.3	0.5	0	1026	64	0.2	0.3	0.4
neos-863472	142	64	0.0	0.0	0.0	0	142	64	0.0	0.0	0.0
neos-872648	38712	64	176.3	1.8	178.1	0	38712	64	175.4	2.0	177.4
neos-873061	30803	64	140.9	1.7	142.6	0	30803	64	142.5	1.9	144.4
neos-876808	92322	64	98.6	3.1	101.7	0	92322	64	98.4	3.1	101.5
neos-880324	232	64	0.0	0.0	0.0	0	232	64	0.0	0.0	0.0
neos-881765	424	64	0.0	0.0	0.1	0	424	64	0.1	0.0	0.1
neos-885086	3951	64	0.6	0.5	1.2	0	3951	64	0.9	0.5	1.4
neos-885524	205	64	5.4	0.4	5.8	0	205	64	5.6	0.4	6.0
neos-886822	1834	64	0.3	0.1	0.4	0	1834	64	0.2	0.1	0.2
neos-892255	1082	64	0.2	0.1	0.3	0	1082	64	0.2	0.0	0.2
neos-905856	1228	64	0.1	0.1	0.2	0	1228	64	0.2	0.0	0.2
neos-906865	792	64	0.1	0.1	0.1	0	792	64	0.1	0.1	0.1
neos-911880	319	64	0.0	0.0	0.1	0	319	64	0.0	0.0	0.1
neos-911970	258	64	0.0	0.0	0.1	0	258	64	0.0	0.0	0.1
neos-912015	1001	64	0.1	0.1	0.2	0	1001	64	0.2	0.1	0.2
neos-912023	1095	64	0.1	0.1	0.2	0	1095	64	0.2	0.0	0.2
neos-913984	4692	64	2.5	0.2	2.7	0	4692	64	2.8	0.2	3.0
neos-914441	7787	64	1.9	0.4	2.2	0	7787	64	1.6	0.4	2.0
neos-916173	862	64	0.2	0.4	0.6	0	862	64	0.3	0.3	0.7
neos-916792	1165	64	0.2	0.4	0.6	0	1165	64	0.6	0.4	1.0
neos-930752	19648	64	6.0	0.3	6.3	0	19648	64	6.1	0.2	6.3
neos-931517	10863	64	2.1	0.3	2.4	0	10863	64	2.2	0.3	2.5
neos-931538	11171	64	2.4	0.3	2.7	0	11171	64	2.2	0.3	2.5
neos-932721	30030	64	11.8	0.5	12.3	0	30030	64	11.7	0.4	12.1
neos-932816	14154	64	4.1	1.1	5.2	0	14154	64	4.2	1.1	5.4
neos-933364	1547	64	0.1	0.0	0.1	0	1547	64	0.1	0.0	0.1
neos-933550	2301	64	0.5	0.2	0.7	0	2301	64	0.4	0.2	0.6
neos-933562	4067	64	1.2	1.6	2.8	0	4067	64	1.1	1.6	2.7
neos-933638	27131	64	18.3	0.5	18.8	0	27131	64	18.0	0.6	18.6
neos-933815	1147	64	0.1	0.0	0.1	0	1147	64	0.0	0.0	0.0
neos-933966	21146	64	11.1	0.6	11.7	0	21146	64	11.4	0.8	12.2
neos-934184	1547	64	0.1	0.0	0.1	0	1547	64	0.1	0.0	0.1
neos-934278	26638	64	16.5	0.9	17.4	0	26638	64	16.6	1.1	17.7
neos-934441	29117	64	18.7	0.8	19.6	0	29117	64	18.6	0.8	19.4
neos-934531	549	64	0.3	0.3	0.6	0	549	64	0.5	0.6	1.0
neos-935234	27636	64	16.9	0.9	17.9	0	27636	64	17.0	0.7	17.7
neos-935348	28978	64	17.0	0.6	17.6	0	28978	64	16.8	0.6	17.4
neos-935496	4000	64	1.1	0.6	1.7	0	4000	64	0.8	0.7	1.4
neos-935627	30839	64	18.9	0.5	19.4	0	30839	64	18.6	0.4	19.1
neos-935674	4015	64	1.1	0.6	1.7	0	4015	64	1.0	0.4	1.5
neos-935769	24339	64	12.2	0.4	12.6	0	24339	64	12.5	0.2	12.7
neos-936660	26498	64	14.6	0.8	15.4	0	26498	64	14.7	0.6	15.3
neos-937446	26033	64	15.3	0.3	15.6	0	26033	64	15.2	0.2	15.3
neos-937511	24828	64	13.7	0.5	14.2	0	24828	64	13.5	0.3	13.8
neos-937815	33788	64	23.8	1.2	24.9	0	33788	64	23.8	1.1	24.9
neos-941262	26859	64	15.1	0.9	16.0	0	26859	64	15.1	0.7	15.8
neos-941313	46102	64	110.4	2.2	112.6	0	46102	64	112.8	2.2	115.0
neos-941698	682	64	0.1	0.1	0.2	0	682	64	0.1	0.0	0.1
neos-941717	4805	64	0.7	0.2	0.9	0	4805	64	0.8	0.1	0.9
neos-941782	1858	64	0.3	0.1	0.5	0	1858	64	0.2	0.1	0.2
neos-942323	468	64	0.1	0.1	0.1	0	468	64	0.0	0.0	0.1

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
neos-942830	1444	64	0.2	0.1	0.3	0	1444	64	0.1	0.0	0.2
neos-942886	336	64	0.0	0.1	0.1	0	336	64	0.1	0.0	0.1
neos-948126	30978	64	19.4	1.3	20.7	0	30978	64	19.5	1.4	21.0
neos-948268	11614	64	3.8	0.5	4.2	0	11614	64	3.6	0.5	4.1
neos-948346	4225	64	4.1	0.9	5.0	0	4225	64	4.4	0.9	5.4
neos-950242	1897	64	0.4	0.3	0.8	0	1897	64	0.6	0.2	0.8
neos-952987	531	64	0.3	0.2	0.5	0	531	64	0.6	0.2	0.8
neos-953928	1858	128	1.1	66.5	67.6	1	5020	64	2.8	0.4	3.2
neos-955215	850	64	0.0	0.0	0.1	0	850	64	0.0	0.0	0.0
neos-955800	882	64	0.1	0.1	0.2	0	882	64	0.0	0.0	0.1
neos-957270	515	64	0.2	0.5	0.7	0	515	64	0.3	0.5	0.9
neos-957323	3688	64	3.8	0.9	4.8	0	3688	64	4.3	1.0	5.3
neos-957389	1299	64	0.3	0.5	0.8	0	1299	64	0.5	0.5	1.0
neos-960392	12434	64	9.2	0.4	9.6	0	12434	64	9.8	0.4	10.2
neos-983171	27536	64	16.1	1.3	17.3	0	27536	64	16.2	1.2	17.4
neos-984165	30160	64	19.0	0.9	19.9	0	30160	64	19.1	1.0	20.1
neos1	4509	64	12.5	2.6	15.1	0	4509	64	13.2	2.7	15.9
neos13	2438	64	0.8	4.5	5.2	0	2438	64	0.9	4.7	5.6
neos15	508	64	0.0	0.0	0.0	0	508	64	0.0	0.0	0.1
neos16	475	64	0.0	0.0	0.0	0	475	64	0.0	0.0	0.0
neos18	293	64	0.1	0.1	0.2	0	293	64	0.2	0.1	0.3
neos2	8434	64	34.4	3.2	37.5	0	8434	64	34.8	3.2	38.0
neos6	6681	64	2.1	0.3	2.4	0	6681	64	2.1	0.3	2.4
neos788725	603	64	0.1	0.0	0.1	0	603	64	0.0	0.0	0.1
neos808444	8106	64	4.1	0.5	4.6	0	8106	64	4.0	0.6	4.6
neos858960	2	64	0.0	0.0	0.0	0	2	64	0.0	0.0	0.0
nesm	6164	64	0.6	0.2	0.7	0	6164	64	0.4	0.1	0.5
net12	11945	64	2.9	0.3	3.2	0	11945	64	2.9	0.4	3.3
netdiversion	26331	64	27.9	1.9	29.8	0	26331	64	27.8	1.9	29.7
netlarge2	111407	64	1542.8	17.2	1560.0	0	111407	64	1528.1	17.3	1545.4
newdano	455	64	0.0	0.0	0.1	0	455	64	0.0	0.0	0.0
nl	14132	64	3.9	0.8	4.7	0	14132	64	3.9	0.8	4.7
nobel-eu-DBE	2927	64	0.3	0.0	0.3	0	2927	64	0.1	0.0	0.2
noswot	134	128	0.0	0.1	0.1	1	135	64	0.0	0.0	0.0
npmv07	168166	64	45.2	4.9	50.1	0	168166	64	47.6	4.7	52.3
ns1111636	27016	64	78.9	2.2	81.1	0	27016	64	80.2	2.1	82.2
ns1116954	15636	64	24.2	1.6	25.9	0	15636	64	24.4	1.6	25.9
ns1208400	9117	64	1.7	0.3	2.0	0	9117	64	1.8	0.2	2.0
ns1456591	2343	64	0.5	0.3	0.8	0	2343	64	0.6	0.3	0.9
ns1606230	15157	64	6.2	0.6	6.8	0	15157	64	6.1	0.5	6.6
ns1631475	101171	64	253.3	0.8	254.1	0	101171	64	256.6	0.9	257.5
ns1644855	47488	64	109.7	47.1	156.8	0	47488	64	110.5	47.2	157.7
ns1663818	*1723*	*64*	*24.9*	*7200.0*	*7224.9*	*0*	*1723*	*64*	*25.8*	*7200.0*	*7225.8*
ns1685374	105867	64	779.2	426.5	1205.7	0	105867	64	774.7	426.3	1200.9
ns1686196	157	64	0.1	0.1	0.2	0	157	64	0.1	0.1	0.2
ns1688347	178	64	0.1	0.2	0.2	0	178	64	0.1	0.2	0.3
ns1696083	498	64	0.3	0.5	0.8	0	498	64	0.3	0.5	0.8
ns1702808	58	64	0.0	0.0	0.1	0	58	64	0.0	0.0	0.0
ns1745726	182	64	0.1	0.1	0.3	0	182	64	0.1	0.1	0.2
ns1758913	27608	64	335.6	13.4	349.0	0	27608	64	335.8	13.4	349.2
ns1766074	27	64	0.0	0.0	0.0	0	27	64	0.0	0.0	0.0
ns1769397	206	64	0.2	0.2	0.4	0	206	64	0.2	0.3	0.5
ns1778858	14685	64	5.4	0.5	5.8	0	14685	64	5.4	0.5	5.9
ns1830653	2753	64	0.5	0.2	0.8	0	2753	64	0.7	0.3	0.9
ns1856153	3231	64	0.7	0.5	1.2	0	3231	64	0.7	0.6	1.3

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
ns1904248	42350	128	55.0	2907.3	2962.3	1	42365	64	55.9	2.3	58.2
ns1905797	5721	64	2.5	0.8	3.3	0	5721	64	2.8	0.8	3.6
ns1905800	2057	64	0.5	0.2	0.7	0	2057	64	0.4	0.1	0.5
ns1952667	81	64	0.2	0.3	0.5	0	81	64	0.2	0.3	0.5
ns2017839	60283	64	118.1	2.7	120.8	0	60283	64	117.7	2.7	120.4
ns2081729	115	64	0.0	0.0	0.0	0	115	64	0.0	0.0	0.0
ns2118727	23580	64	111.8	13.3	125.1	0	23580	64	112.7	13.3	126.0
ns2122603	13837	64	3.5	0.6	4.1	0	13837	64	3.8	0.7	4.5
ns2124243	94850	64	312.1	2.7	314.8	0	94850	64	313.0	2.7	315.8
ns2137859	2864	64	5.2	5.2	10.3	0	2864	64	6.2	5.2	11.4
ns4-pr3	10658	64	1.8	0.1	1.8	0	10658	64	1.8	0.0	1.9
ns4-pr9	9965	64	1.5	0.1	1.5	0	9965	64	1.3	0.1	1.4
ns894236	36452	64	16.2	0.3	16.6	0	36452	64	15.4	0.4	15.8
ns894244	29128	64	21.5	0.9	22.4	0	29128	64	21.7	0.8	22.4
ns894786	18747	64	14.5	0.5	15.0	0	18747	64	14.8	0.6	15.4
ns894788	10998	64	1.5	0.1	1.6	0	10998	64	1.4	0.1	1.5
ns903616	30158	64	22.4	0.6	22.9	0	30158	64	22.4	0.7	23.0
ns930473	10736	64	8.2	0.3	8.6	0	10736	64	8.2	0.3	8.6
nsa	1162	64	0.1	0.1	0.2	0	1162	64	0.1	0.0	0.1
nsct1	4841	64	1.2	1.4	2.6	0	4841	64	1.4	1.4	2.8
nsct2	6810	64	1.3	1.5	2.9	0	6810	64	1.5	1.5	3.0
nsic1	254	64	0.0	0.0	0.0	0	254	64	0.0	0.0	0.0
nsic2	268	64	0.0	0.0	0.0	0	268	64	0.0	0.0	0.0
nsir1	3762	64	0.6	0.4	1.0	0	3762	64	1.0	0.2	1.2
nsir2	3288	64	0.5	0.3	0.7	0	3288	64	0.7	0.4	1.0
nsr8k	102631	64	402.4	196.1	598.5	0	102631	64	397.4	196.8	594.2
nsrand-ipx	152	64	0.1	0.2	0.4	0	152	64	0.4	0.3	0.7
nu120-pr3	6187	64	1.1	0.2	1.2	0	6187	64	1.1	0.1	1.2
nu60-pr9	5736	64	1.1	0.1	1.3	0	5736	64	1.2	0.1	1.3
nug05	161	64	0.0	0.0	0.0	0	161	64	0.0	0.0	0.0
nug06	1117	64	0.1	0.1	0.1	0	1117	64	0.1	0.0	0.1
nug07	7569	64	1.0	0.1	1.1	0	7569	64	0.8	0.1	0.9
nug08	13541	64	2.0	0.2	2.2	0	13541	64	1.7	0.2	1.9
nug08-3rd	43975	64	1840.1	7200.0	9040.1	0	43975	64	1869.3	7200.0	9069.3
nug12	99946	64	160.0	13.7	173.7	0	99946	64	159.6	13.5	173.2
nw04	82	64	0.3	0.5	0.8	0	82	64	1.0	0.5	1.4
nw14	166	64	0.7	0.7	1.4	0	166	64	1.9	0.7	2.7
ofi	*139069*	*128*	*407.6*	*7200.0*	*7607.6*	2	139104	64	412.4	19.8	432.2
opm2-z10-s2	21918	64	111.0	11.3	122.3	0	21918	64	110.7	11.3	122.0
opm2-z11-s8	24349	64	162.2	15.7	177.9	0	24349	64	163.7	15.2	178.9
opm2-z12-s14	28647	64	279.0	32.9	312.0	0	28647	64	283.9	33.8	317.6
opm2-z12-s7	30797	64	313.1	37.7	350.8	0	30797	64	315.2	39.6	354.8
opm2-z7-s2	10304	64	7.3	1.7	9.0	0	10304	64	7.4	1.6	9.0
opt1217	143	64	0.0	0.0	0.0	0	143	64	0.0	0.0	0.0
orna1	1327	64	0.2	2.2	2.3	0	1327	64	0.2	2.3	2.5
orna2	1539	64	0.2	2.3	2.4	0	1539	64	0.2	2.1	2.4
orna3	1754	64	0.2	2.5	2.7	0	1754	64	0.3	2.0	2.3
orna4	2708	64	0.2	3.4	3.7	0	2708	64	0.3	3.6	3.9
orna7	2280	64	0.3	2.3	2.5	0	2280	64	0.3	1.9	2.2
orswq2	148	64	0.0	0.0	0.0	0	148	64	0.0	0.0	0.0
osa-07	966	64	0.5	0.2	0.7	0	966	64	0.5	0.2	0.7
osa-14	2104	64	1.2	0.5	1.7	0	2104	64	1.4	0.5	1.9
osa-30	5628	64	5.9	0.9	6.8	0	5628	64	6.3	0.9	7.2
osa-60	13364	64	38.1	2.1	40.1	0	13364	64	39.1	2.1	41.2
p0033	19	64	0.0	0.0	0.0	0	19	64	0.0	0.0	0.0

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
p0040	13	64	0.0	0.0	0.0	1	23	64	0.0	0.0	0.0
p010	13700	64	1.8	1.0	2.8	0	13700	64	2.2	1.0	3.2
p0201	50	64	0.0	0.0	0.0	1	51	64	0.0	0.0	0.0
p0282	114	64	0.0	0.0	0.0	0	114	64	0.0	0.0	0.0
p0291	27	64	0.0	0.0	0.0	0	27	64	0.0	0.0	0.0
p05	7515	64	0.9	0.6	1.5	0	7515	64	0.8	0.6	1.4
p0548	84	64	0.0	0.0	0.0	0	84	64	0.0	0.0	0.0
p100x588b	235	64	0.0	0.0	0.0	0	235	64	0.0	0.0	0.0
p19	315	64	0.0	0.0	0.1	0	315	64	0.0	0.0	0.0
p2756	73	64	0.0	0.0	0.1	0	73	64	0.0	0.0	0.0
p2m2p1m1p0n100	10	64	0.0	0.0	0.0	0	10	64	0.0	0.0	0.0
p6000	728	64	0.1	0.2	0.3	0	728	64	0.2	0.2	0.4
p6b	548	64	0.1	0.0	0.1	0	548	64	0.1	0.0	0.1
p80x400b	152	64	0.0	0.0	0.0	0	152	64	0.0	0.0	0.0
pcb1000	2684	64	0.3	0.4	0.7	0	2684	64	0.4	0.4	0.8
pcb3000	7719	64	0.9	1.2	2.0	0	7719	64	1.0	1.4	2.4
pds-02	2713	64	0.1	0.1	0.2	0	2713	64	0.1	0.1	0.1
pds-06	10699	64	1.1	0.2	1.3	0	10699	64	1.1	0.2	1.3
pds-10	15362	64	1.9	0.4	2.3	0	15362	64	1.9	0.4	2.3
pds-100	661386	64	4676.7	9.3	4685.9	0	661386	64	4662.1	9.1	4671.2
pds-20	37294	64	13.1	1.0	14.1	0	37294	64	13.4	1.0	14.4
pds-30	65338	64	58.4	0.9	59.3	0	65338	64	57.5	1.2	58.7
pds-40	101787	64	159.1	3.2	162.2	0	101787	64	155.7	3.0	158.7
pds-50	137692	64	289.9	3.6	293.5	0	137692	64	293.2	3.6	296.8
pds-60	188842	64	563.5	5.1	568.6	0	188842	64	563.8	5.3	569.1
pds-70	222389	64	759.3	6.5	765.8	0	222389	64	761.6	6.3	767.9
pds-80	240224	64	823.9	7.9	831.8	0	240224	64	821.3	8.0	829.3
pds-90	512992	64	3214.8	8.7	3223.5	0	512992	64	3210.0	8.8	3218.8
perold	5155	64	0.8	5.5	6.2	0	5155	64	0.6	4.9	5.6
pf2177	9474	64	3.4	0.6	4.0	0	9474	64	3.3	0.6	3.9
pg	2164	64	0.0	0.1	0.1	0	2164	64	0.1	0.1	0.1
pg5_34	3400	64	0.1	0.1	0.2	0	3400	64	0.1	0.1	0.2
pgp2	3713	64	0.1	0.2	0.3	0	3713	64	0.2	0.2	0.4
pigeon-10	245	64	0.0	0.0	0.0	0	245	64	0.0	0.0	0.0
pigeon-11	237	64	0.0	0.0	0.0	0	237	64	0.0	0.0	0.0
pigeon-12	305	64	0.0	0.0	0.1	0	305	64	0.0	0.0	0.1
pigeon-13	288	64	0.0	0.0	0.1	0	288	64	0.0	0.0	0.1
pigeon-19	608	64	0.0	0.1	0.1	0	608	64	0.1	0.1	0.1
pilot	10227	64	3.6	137.1	140.7	0	10227	64	3.6	136.9	140.5
pilot-ja	11126	64	1.7	11.5	13.2	0	11126	64	1.6	11.4	12.9
pilot-we	5730	64	0.8	1.8	2.7	0	5730	64	0.8	1.6	2.4
pilot4	1483	64	0.2	1.2	1.3	0	1483	64	0.2	1.2	1.4
pilot87	15871	64	9.5	2274.2	2283.7	0	15871	64	9.5	2283.4	2292.9
pilotnov	7134	64	0.9	1.6	2.5	0	7134	64	0.7	1.4	2.1
pk1	101	64	0.0	0.0	0.0	0	101	64	0.0	0.0	0.0
pldd000b	1667	64	0.2	0.8	1.0	0	1667	64	0.1	0.9	1.1
pldd001b	1662	64	0.2	0.9	1.1	0	1662	64	0.2	0.5	0.7
pldd002b	1649	64	0.2	0.9	1.1	0	1649	64	0.3	0.6	0.9
pldd003b	1657	64	0.2	0.8	1.0	0	1657	64	0.3	0.6	0.9
pldd004b	1654	64	0.2	0.8	1.0	0	1654	64	0.2	0.6	0.8
pldd005b	1655	64	0.2	0.7	0.9	0	1655	64	0.2	0.6	0.9
pldd006b	1693	64	0.2	0.8	0.9	0	1693	64	0.3	0.5	0.7
pldd007b	1714	64	0.2	0.7	0.9	0	1714	64	0.3	0.5	0.8
pldd008b	1716	64	0.2	0.8	0.9	0	1716	64	0.2	0.5	0.8
pldd009b	2258	64	0.3	0.9	1.2	0	2258	64	0.4	0.8	1.2

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
pldd010b	2462	64	0.4	0.9	1.3	0	2462	64	0.5	0.9	1.4
pldd011b	2413	64	0.4	0.8	1.2	0	2413	64	0.5	0.5	0.9
pldd012b	2447	64	0.4	0.8	1.2	0	2447	64	0.5	0.8	1.2
pltexpa2-16	1094	64	0.1	0.1	0.1	0	1094	64	0.0	0.0	0.1
pltexpa2-6	411	64	0.0	0.0	0.1	0	411	64	0.0	0.0	0.0
pltexpa3_16	15975	64	4.1	0.8	4.9	0	15975	64	4.4	0.8	5.2
pltexpa3_6	2741	64	0.3	0.2	0.5	0	2741	64	0.5	0.1	0.6
pltexpa4_6	14762	64	3.5	0.5	4.1	0	14762	64	4.0	0.6	4.6
pp08a	145	64	0.0	0.0	0.0	0	145	64	0.0	0.0	0.0
pp08aCUTS	224	64	0.0	0.0	0.0	0	224	64	0.0	0.0	0.0
primagaz	1802	64	0.4	0.1	0.6	0	1802	64	0.4	0.1	0.5
problem	9	64	0.0	0.0	0.0	0	9	64	0.0	0.0	0.0
probportfolio	126	64	0.0	0.0	0.0	0	126	64	0.0	0.0	0.1
prod1	193	64	0.0	0.0	0.0	0	193	64	0.0	0.0	0.1
prod2	344	64	0.0	0.0	0.1	0	344	64	0.1	0.1	0.1
progas	3998	64	0.8	15.4	16.2	0	3998	64	0.8	15.5	16.3
protfold	13331	64	2.8	0.3	3.1	0	13331	64	2.7	0.3	3.0
pw-myciel4	2154	64	0.6	0.1	0.8	0	2154	64	0.7	0.1	0.8
qap10	58606	64	30.2	1.4	31.7	0	58606	64	30.0	1.2	31.3
qiu	1604	64	0.2	0.1	0.3	0	1604	64	0.2	0.0	0.3
qiulp	1604	64	0.2	0.1	0.2	0	1604	64	0.1	0.0	0.1
qnet1	789	64	0.1	0.0	0.1	0	789	64	0.1	0.0	0.1
qnet1_o	406	64	0.0	0.0	0.1	0	406	64	0.0	0.0	0.1
queens-30	13322	64	10.8	748.5	759.3	0	13322	64	13.5	753.5	767.0
r05	7499	64	0.7	0.5	1.1	0	7499	64	0.9	0.5	1.5
r80x800	188	64	0.0	0.0	0.0	0	188	64	0.0	0.0	0.0
rail01	349407	128	4435.1	6310.6	10745.7	1	349603	64	4461.4	1.9	4463.3
rail2586	*30254*	*64*	*675.5*	*7200.0*	*7875.5*	*0*	*30254*	*64*	*679.5*	*7200.0*	*7879.6*
rail4284	*54982*	*64*	*1670.1*	*7200.0*	*8870.1*	*0*	*54982*	*64*	*1678.7*	*7200.0*	*8878.7*
rail507	13033	64	11.2	0.7	12.0	0	13033	64	11.6	0.8	12.5
rail516	6992	64	5.6	0.5	6.1	0	6992	64	5.9	0.5	6.4
rail582	10608	64	9.2	0.6	9.8	0	10608	64	9.5	0.6	10.1
ramos3	19446	64	58.7	975.6	1034.3	0	19446	64	60.3	975.4	1035.7
ran10x10a	133	64	0.0	0.0	0.0	0	133	64	0.0	0.0	0.0
ran10x10b	149	64	0.0	0.0	0.0	0	149	64	0.0	0.0	0.0
ran10x10c	161	64	0.0	0.0	0.0	0	161	64	0.0	0.0	0.0
ran10x12	162	64	0.0	0.0	0.0	0	162	64	0.0	0.0	0.0
ran10x26	423	64	0.0	0.0	0.0	0	423	64	0.0	0.0	0.0
ran12x12	186	64	0.0	0.0	0.0	0	186	64	0.0	0.0	0.0
ran12x21	380	64	0.0	0.0	0.0	0	380	64	0.0	0.0	0.0
ran13x13	236	64	0.0	0.0	0.0	0	236	64	0.0	0.0	0.0
ran14x18	397	64	0.0	0.0	0.0	0	397	64	0.0	0.0	0.0
ran14x18-disj-8	1915	128	0.2	14.1	14.4	1	1916	64	0.3	4.8	5.2
ran14x18.disj-8	1915	128	0.2	14.3	14.6	1	1916	64	0.3	4.8	5.2
ran14x18_1	434	64	0.0	0.0	0.0	0	434	64	0.0	0.0	0.0
ran16x16	379	64	0.0	0.0	0.0	0	379	64	0.0	0.0	0.1
ran17x17	258	64	0.0	0.0	0.0	0	258	64	0.0	0.0	0.0
ran4x64	515	64	0.0	0.0	0.0	0	515	64	0.0	0.0	0.0
ran6x43	447	64	0.0	0.0	0.0	0	447	64	0.0	0.0	0.0
ran8x32	402	64	0.0	0.0	0.0	0	402	64	0.0	0.0	0.0
rat1	2870	64	0.8	5.8	6.6	0	2870	64	0.7	5.8	6.5
rat5	3024	64	1.1	609.6	610.7	0	3024	64	1.4	611.8	613.1
rat7a	11319	64	12.3	7200.0	7212.3	0	11319	64	13.7	7200.0	7213.7
rd-rplusc-21	138	64	6.3	42.3	48.6	1	139	64	7.7	41.9	49.5
reblock166	3359	64	1.4	0.4	1.8	0	3359	64	1.3	0.5	1.7

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
reblock354	12854	64	7.9	0.7	8.6	0	12854	64	8.2	0.9	9.1
reblock420	9480	64	11.5	1.0	12.5	0	9480	64	11.6	1.1	12.7
reblock67	972	64	0.1	0.1	0.2	0	972	64	0.1	0.0	0.2
recipe	40	64	0.0	0.0	0.0	0	40	64	0.0	0.0	0.0
refine	21	64	0.0	0.0	0.0	0	21	64	0.0	0.0	0.0
rentacar	6483	64	1.2	0.3	1.5	0	6483	64	1.1	0.2	1.3
rgn	85	64	0.0	0.0	0.0	0	85	64	0.0	0.0	0.0
rlfddd	825	64	0.2	0.3	0.5	0	825	64	0.3	0.3	0.6
rlfdual	8652	64	1.7	0.6	2.3	0	8652	64	1.7	0.6	2.3
rlfprim	5532	64	1.2	0.4	1.6	0	5532	64	1.2	0.5	1.7
rlp1	248	64	0.0	0.0	0.0	0	248	64	0.0	0.0	0.0
rmatr100-p10	6260	64	1.6	0.2	1.8	0	6260	64	1.5	0.1	1.7
rmatr100-p5	10885	64	3.1	0.2	3.3	0	10885	64	3.4	0.2	3.6
rmatr200-p10	13646	64	11.7	0.5	12.2	0	13646	64	11.8	0.5	12.3
rmatr200-p20	11166	64	7.7	0.4	8.1	0	11166	64	8.2	0.5	8.7
rmatr200-p5	17650	64	18.8	0.6	19.4	0	17650	64	18.8	0.7	19.5
rmine10	13871	64	20.9	2.9	23.8	0	13871	64	21.6	3.0	24.7
rmine14	92846	64	1720.4	22.5	1742.9	0	92846	64	1714.0	22.6	1736.5
rmine6	2058	64	0.7	0.2	0.9	0	2058	64	0.6	0.2	0.8
rocII-4-11	751	64	0.3	0.9	1.1	0	751	64	0.5	0.9	1.3
rocII-7-11	1075	64	0.5	1.8	2.3	0	1075	64	0.8	1.8	2.5
rocII-9-11	1523	64	0.8	2.6	3.4	0	1523	64	1.1	2.5	3.6
rococoB10-011000	4263	64	0.8	0.1	0.9	0	4263	64	0.7	0.0	0.8
rococoC10-001000	2344	64	0.3	0.1	0.4	0	2344	64	0.2	0.0	0.2
rococoC11-011100	10112	64	1.6	0.1	1.8	0	10112	64	1.8	0.2	2.0
rococoC12-111000	11920	64	3.4	0.3	3.7	0	11920	64	3.2	0.3	3.5
roll3000	2627	64	0.3	0.1	0.5	0	2627	64	0.2	0.1	0.3
rosen1	898	64	0.2	0.2	0.3	0	898	64	0.1	0.1	0.2
rosen10	2401	64	1.0	0.6	1.6	0	2401	64	0.9	0.6	1.6
rosen2	1627	64	0.6	0.3	0.9	0	1627	64	0.6	0.2	0.8
rosen7	332	64	0.0	0.0	0.1	0	332	64	0.0	0.1	0.1
rosen8	662	64	0.1	0.1	0.2	0	662	64	0.2	0.1	0.3
rout	293	64	0.0	0.0	0.0	0	293	64	0.0	0.0	0.0
route	2239	64	1.1	0.4	1.5	0	2239	64	1.2	0.4	1.6
roy	119	64	0.0	0.0	0.0	0	119	64	0.0	0.0	0.0
rvb-sub	512	64	0.8	1.5	2.3	0	512	64	1.4	1.5	3.0
satellites1-25	9023	64	4.7	0.6	5.3	0	9023	64	4.9	0.5	5.4
satellites2-60	79705	64	373.8	2.6	376.4	0	79705	64	376.6	2.6	379.2
satellites2-60-fs	65144	64	270.6	7.0	277.6	0	65144	64	271.8	6.5	278.3
satellites3-40	190485	64	2657.9	26.6	2684.5	0	190485	64	2654.6	26.3	2680.9
satellites3-40-fs	199929	128	2345.5	4825.0	7170.5	1	199931	64	2342.0	34.6	2376.6
sc105	99	64	0.0	0.0	0.0	0	99	64	0.0	0.0	0.0
sc205	217	64	0.0	0.0	0.0	0	217	64	0.0	0.0	0.0
sc205-2r-100	1109	64	0.1	0.1	0.2	0	1109	64	0.2	0.0	0.2
sc205-2r-16	171	64	0.0	0.0	0.0	0	171	64	0.0	0.0	0.0
sc205-2r-1600	9189	64	4.7	0.3	5.0	0	9189	64	4.9	0.2	5.2
sc205-2r-200	2195	64	0.4	0.1	0.5	0	2195	64	0.5	0.0	0.5
sc205-2r-27	327	64	0.0	0.0	0.0	0	327	64	0.0	0.0	0.0
sc205-2r-32	331	64	0.0	0.0	0.0	0	331	64	0.0	0.0	0.0
sc205-2r-4	55	64	0.0	0.0	0.0	0	55	64	0.0	0.0	0.0
sc205-2r-400	4393	64	1.1	0.2	1.3	0	4393	64	1.3	0.2	1.5
sc205-2r-50	621	64	0.1	0.0	0.1	0	621	64	0.0	0.0	0.0
sc205-2r-64	651	64	0.1	0.0	0.1	0	651	64	0.0	0.0	0.1
sc205-2r-8	100	64	0.0	0.0	0.0	0	100	64	0.0	0.0	0.0
sc205-2r-800	8729	64	3.4	0.4	3.8	0	8729	64	3.4	0.5	3.9

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
sc50a	45	64	0.0	0.0	0.0	0	45	64	0.0	0.0	0.0
sc50b	49	64	0.0	0.0	0.0	0	49	64	0.0	0.0	0.0
scagr25	784	64	0.0	0.0	0.1	0	784	64	0.1	0.0	0.1
scagr7	178	64	0.0	0.0	0.0	0	178	64	0.0	0.0	0.0
scagr7-2b-16	717	64	0.0	0.0	0.1	0	717	64	0.1	0.0	0.1
scagr7-2b-4	189	64	0.0	0.0	0.0	0	189	64	0.0	0.0	0.0
scagr7-2b-64	11614	64	1.2	0.4	1.6	0	11614	64	1.6	0.4	2.0
scagr7-2c-16	684	64	0.0	0.0	0.1	0	684	64	0.0	0.0	0.1
scagr7-2c-4	186	64	0.0	0.0	0.0	0	186	64	0.0	0.0	0.0
scagr7-2c-64	2727	64	0.2	0.1	0.3	0	2727	64	0.2	0.1	0.3
scagr7-2r-108	4538	64	0.3	0.2	0.5	0	4538	64	0.4	0.2	0.6
scagr7-2r-16	707	64	0.0	0.0	0.1	0	707	64	0.0	0.0	0.1
scagr7-2r-216	9082	64	0.8	0.3	1.1	0	9082	64	1.0	0.3	1.4
scagr7-2r-27	1144	64	0.0	0.1	0.1	0	1144	64	0.0	0.1	0.1
scagr7-2r-32	1335	64	0.1	0.1	0.1	0	1335	64	0.0	0.1	0.1
scagr7-2r-4	185	64	0.0	0.0	0.0	0	185	64	0.0	0.0	0.0
scagr7-2r-432	16562	64	2.5	0.5	3.0	0	16562	64	2.8	0.5	3.3
scagr7-2r-54	2299	64	0.1	0.1	0.2	0	2299	64	0.1	0.1	0.2
scagr7-2r-64	2763	64	0.2	0.1	0.2	0	2763	64	0.2	0.1	0.3
scagr7-2r-8	357	64	0.0	0.0	0.0	0	357	64	0.0	0.0	0.0
scagr7-2r-864	32972	64	12.0	1.0	13.0	0	32972	64	12.6	0.9	13.6
scfxm1	447	64	0.0	0.0	0.1	0	447	64	0.0	0.0	0.0
scfxm1-2b-16	4522	64	0.4	0.2	0.6	0	4522	64	0.5	0.2	0.7
scfxm1-2b-4	1135	64	0.1	0.1	0.1	0	1135	64	0.1	0.1	0.1
scfxm1-2b-64	26349	64	8.8	0.8	9.6	0	26349	64	9.2	0.8	10.0
scfxm1-2c-4	1071	64	0.1	0.1	0.1	0	1071	64	0.1	0.0	0.1
scfxm1-2r-128	23788	64	7.7	0.7	8.5	0	23788	64	8.0	0.7	8.7
scfxm1-2r-16	4672	64	0.5	0.2	0.7	0	4672	64	0.5	0.1	0.6
scfxm1-2r-256	47280	64	30.8	1.3	32.1	0	47280	64	31.7	1.3	33.0
scfxm1-2r-27	8158	64	0.8	0.3	1.1	0	8158	64	0.6	0.1	0.8
scfxm1-2r-32	8924	64	1.0	0.3	1.3	0	8924	64	0.8	0.3	1.1
scfxm1-2r-4	1203	64	0.1	0.1	0.1	0	1203	64	0.0	0.1	0.1
scfxm1-2r-64	13133	64	1.8	0.5	2.2	0	13133	64	2.1	0.5	2.6
scfxm1-2r-8	2252	64	0.2	0.1	0.2	0	2252	64	0.1	0.1	0.2
scfxm1-2r-96	16931	64	3.7	0.6	4.3	0	16931	64	3.9	0.6	4.5
scfxm2	1119	64	0.1	0.1	0.1	0	1119	64	0.1	0.1	0.1
scfxm3	1992	64	0.1	0.1	0.2	0	1992	64	0.1	0.1	0.2
scorpion	245	64	0.0	0.0	0.0	0	245	64	0.0	0.0	0.0
scrs8	608	64	0.0	0.0	0.1	0	608	64	0.0	0.0	0.1
scrs8-2b-16	88	64	0.0	0.0	0.0	0	88	64	0.0	0.0	0.0
scrs8-2b-4	22	64	0.0	0.0	0.0	0	22	64	0.0	0.0	0.0
scrs8-2b-64	296	64	0.0	0.1	0.1	0	296	64	0.0	0.0	0.1
scrs8-2c-16	91	64	0.0	0.0	0.0	0	91	64	0.0	0.0	0.0
scrs8-2c-32	179	64	0.0	0.0	0.1	0	179	64	0.0	0.0	0.0
scrs8-2c-4	22	64	0.0	0.0	0.0	0	22	64	0.0	0.0	0.0
scrs8-2c-64	358	64	0.0	0.1	0.1	0	358	64	0.1	0.0	0.1
scrs8-2c-8	45	64	0.0	0.0	0.0	0	45	64	0.0	0.0	0.0
scrs8-2r-128	543	64	0.1	0.1	0.2	0	543	64	0.1	0.1	0.2
scrs8-2r-16	96	64	0.0	0.0	0.0	0	96	64	0.0	0.0	0.0
scrs8-2r-256	1119	64	0.2	0.2	0.4	0	1119	64	0.3	0.2	0.5
scrs8-2r-27	103	64	0.0	0.0	0.0	0	103	64	0.0	0.0	0.0
scrs8-2r-32	192	64	0.0	0.0	0.0	0	192	64	0.0	0.0	0.1
scrs8-2r-4	24	64	0.0	0.0	0.0	0	24	64	0.0	0.0	0.0
scrs8-2r-512	2532	64	0.4	0.4	0.8	0	2532	64	0.6	0.5	1.0
scrs8-2r-64	384	64	0.0	0.1	0.1	0	384	64	0.0	0.1	0.1

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
scrs8-2r-64b	271	64	0.0	0.1	0.1	0	271	64	0.0	0.1	0.1
scrs8-2r-8	41	64	0.0	0.0	0.0	0	41	64	0.0	0.0	0.0
scsd1	97	64	0.0	0.0	0.0	0	97	64	0.0	0.0	0.0
scsd6	423	64	0.0	0.0	0.1	0	423	64	0.1	0.1	0.1
scsd8	1837	64	0.2	0.1	0.3	0	1837	64	0.3	0.0	0.3
scsd8-2b-16	295	64	0.0	0.1	0.1	0	295	64	0.0	0.0	0.1
scsd8-2b-4	47	64	0.0	0.0	0.0	0	47	64	0.0	0.0	0.0
scsd8-2b-64	2056	64	0.1	0.3	0.4	0	2056	64	0.4	0.4	0.8
scsd8-2c-16	198	64	0.0	0.0	0.1	0	198	64	0.0	0.0	0.1
scsd8-2c-4	47	64	0.0	0.0	0.0	0	47	64	0.0	0.0	0.0
scsd8-2c-64	2056	64	0.2	0.4	0.6	0	2056	64	0.4	0.3	0.8
scsd8-2r-108	1114	64	0.1	0.2	0.3	0	1114	64	0.1	0.2	0.4
scsd8-2r-16	231	64	0.0	0.0	0.1	0	231	64	0.0	0.0	0.0
scsd8-2r-216	2314	64	0.1	0.4	0.5	0	2314	64	0.3	0.3	0.6
scsd8-2r-27	286	64	0.0	0.1	0.1	0	286	64	0.0	0.0	0.1
scsd8-2r-32	429	64	0.0	0.1	0.1	0	429	64	0.1	0.0	0.1
scsd8-2r-4	47	64	0.0	0.0	0.0	0	47	64	0.0	0.0	0.0
scsd8-2r-432	4530	64	0.2	0.5	0.8	0	4530	64	0.5	0.5	1.1
scsd8-2r-54	600	64	0.1	0.1	0.2	0	600	64	0.1	0.1	0.2
scsd8-2r-64	1188	64	0.1	0.1	0.2	0	1188	64	0.1	0.1	0.2
scsd8-2r-8	97	64	0.0	0.0	0.0	0	97	64	0.0	0.0	0.0
scsd8-2r-8b	97	64	0.0	0.0	0.0	0	97	64	0.0	0.0	0.0
sct1	15178	64	7.4	5.3	12.7	0	15178	64	7.6	5.3	12.9
sct32	14051	64	3.8	1.0	4.8	0	14051	64	3.7	0.8	4.6
sct5	6217	64	3.1	0.5	3.6	0	6217	64	3.5	0.6	4.1
sctap1	262	64	0.0	0.0	0.0	0	262	64	0.0	0.0	0.1
sctap1-2b-16	323	64	0.0	0.0	0.0	0	323	64	0.0	0.0	0.0
sctap1-2b-4	83	64	0.0	0.0	0.0	0	83	64	0.0	0.0	0.0
sctap1-2b-64	4773	64	0.6	0.3	0.9	0	4773	64	0.7	0.3	1.0
sctap1-2c-16	329	64	0.0	0.0	0.0	0	329	64	0.0	0.0	0.0
sctap1-2c-4	85	64	0.0	0.0	0.0	0	85	64	0.0	0.0	0.0
sctap1-2c-64	1146	64	0.1	0.1	0.1	0	1146	64	0.1	0.1	0.2
sctap1-2r-108	2109	64	0.2	0.1	0.4	0	2109	64	0.4	0.1	0.5
sctap1-2r-16	282	64	0.0	0.0	0.0	0	282	64	0.0	0.0	0.0
sctap1-2r-216	4248	64	0.5	0.3	0.8	0	4248	64	0.8	0.3	1.1
sctap1-2r-27	536	64	0.0	0.0	0.1	0	536	64	0.0	0.0	0.1
sctap1-2r-32	559	64	0.0	0.0	0.1	0	559	64	0.0	0.0	0.1
sctap1-2r-4	84	64	0.0	0.0	0.0	0	84	64	0.0	0.0	0.0
sctap1-2r-480	9373	64	1.5	0.6	2.2	0	9373	64	1.9	0.6	2.5
sctap1-2r-54	1069	64	0.1	0.1	0.2	0	1069	64	0.1	0.1	0.2
sctap1-2r-64	1123	64	0.1	0.1	0.2	0	1123	64	0.2	0.1	0.2
sctap1-2r-8	147	64	0.0	0.0	0.0	0	147	64	0.0	0.0	0.0
sctap1-2r-8b	161	64	0.0	0.0	0.0	0	161	64	0.0	0.0	0.0
sctap2	505	64	0.1	0.0	0.1	0	505	64	0.1	0.0	0.1
sctap3	604	64	0.1	0.0	0.1	0	604	64	0.1	0.0	0.1
seba	6	64	0.0	0.0	0.0	0	6	64	0.0	0.0	0.0
self	13495	64	48.6	6101.6	6150.2	0	13495	64	49.7	6095.1	6144.8
set1ch	513	64	0.0	0.0	0.0	0	513	64	0.0	0.0	0.0
set3-10	2279	64	0.1	0.1	0.3	0	2279	64	0.2	0.1	0.4
set3-15	2269	64	0.2	0.1	0.3	0	2269	64	0.2	0.1	0.4
set3-20	2344	64	0.1	0.1	0.3	0	2344	64	0.2	0.1	0.4
seymour	3858	64	1.0	0.2	1.2	0	3858	64	0.9	0.2	1.2
seymour-disj-10	5700	128	1.8	63.8	65.6	1	5702	64	2.1	8.6	10.7
seymour.disj-10	5700	128	1.7	63.8	65.5	1	5702	64	2.1	8.4	10.5
seymourl	3858	64	0.9	0.2	1.1	0	3858	64	1.1	0.1	1.2

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
sgpf5y6	*153350*	*128*	*167.8*	*7200.0*	*7367.8*	1	197667	64	221.6	7.1	228.7
share1b	217	64	0.0	0.0	0.0	0	217	64	0.0	0.0	0.0
share2b	159	64	0.0	0.0	0.0	0	159	64	0.0	0.0	0.0
shell	595	64	0.0	0.0	0.1	0	595	64	0.0	0.0	0.0
ship04l	473	64	0.0	0.0	0.1	0	473	64	0.0	0.0	0.1
ship04s	383	64	0.0	0.0	0.1	0	383	64	0.0	0.1	0.1
ship08l	810	64	0.1	0.1	0.1	0	810	64	0.1	0.1	0.2
ship08s	513	64	0.0	0.1	0.1	0	513	64	0.1	0.0	0.1
ship12l	1085	64	0.1	0.1	0.2	0	1085	64	0.2	0.1	0.2
ship12s	648	64	0.0	0.1	0.1	0	648	64	0.1	0.1	0.2
shipsched	2174	64	0.6	0.2	0.8	0	2174	64	0.9	0.3	1.2
shs1023	*174484*	*128*	*604.0*	*7200.0*	*7804.0*	5	177407	64	618.0	7.5	625.4
siena1	28441	64	19.0	3.2	22.2	0	28441	64	19.7	3.2	22.9
sierra	640	64	0.0	0.1	0.1	0	640	64	0.0	0.0	0.1
sing2	37541	64	40.3	0.7	41.0	0	37541	64	40.9	0.7	41.6
sing245	218115	64	1608.1	7.0	1615.1	0	218115	64	1611.8	6.9	1618.7
sing359	*352087*	*64*	*5830.9*	*7200.0*	*13030.9*	*0*	*352087*	*64*	*5799.4*	*7200.0*	*12999.4*
slptsk	4856	64	1.2	12.8	14.0	0	4856	64	1.8	13.0	14.8
small000	557	64	0.0	0.1	0.1	0	557	64	0.0	0.0	0.1
small001	725	128	0.0	0.8	0.9	1	737	64	0.0	0.0	0.1
small002	834	128	0.0	1.0	1.1	1	850	64	0.1	0.1	0.2
small003	681	128	0.0	0.8	0.8	1	693	64	0.1	0.1	0.1
small004	465	128	0.0	0.8	0.8	1	476	64	0.1	0.1	0.1
small005	601	128	0.0	0.9	1.0	1	621	64	0.1	0.1	0.2
small006	518	128	0.0	0.8	0.8	1	540	64	0.1	0.1	0.1
small007	517	128	0.0	1.0	1.1	1	547	64	0.1	0.1	0.1
small008	489	128	0.0	0.9	0.9	1	511	64	0.1	0.1	0.1
small009	416	128	0.0	0.9	0.9	1	435	64	0.1	0.1	0.1
small010	328	128	0.0	0.8	0.8	1	342	64	0.1	0.1	0.1
small011	337	128	0.0	0.8	0.9	1	348	64	0.0	0.0	0.1
small012	287	64	0.0	0.1	0.1	0	287	64	0.1	0.1	0.1
small013	289	64	0.0	0.1	0.1	0	289	64	0.1	0.1	0.1
small014	341	64	0.0	0.1	0.1	0	341	64	0.0	0.1	0.1
small015	338	64	0.0	0.1	0.1	0	338	64	0.1	0.1	0.1
small016	338	64	0.0	0.1	0.1	0	338	64	0.0	0.1	0.1
south31	23654	64	14.9	2.4	17.3	0	23654	64	15.2	2.5	17.8
sp97ar	6984	64	2.9	1.3	4.2	1	6986	64	3.3	0.5	3.8
sp97ic	3102	64	0.8	0.5	1.3	0	3102	64	0.7	0.6	1.3
sp98ar	10342	64	2.9	1.4	4.2	1	10349	64	3.1	0.5	3.6
sp98ic	3276	64	0.7	0.6	1.3	0	3276	64	0.8	0.6	1.4
sp98ir	2967	64	0.7	0.5	1.2	1	2970	64	0.5	0.2	0.7
square15	208626	64	2424.8	8.4	2433.2	0	208626	64	2424.6	8.5	2433.0
stair	658	64	0.1	4.0	4.1	0	658	64	0.1	4.0	4.1
standata	50	64	0.0	0.0	0.0	0	50	64	0.0	0.0	0.0
standmps	189	64	0.0	0.0	0.0	0	189	64	0.0	0.0	0.0
stat96v4	144049	64	395.8	3588.8	3984.6	0	144049	64	402.2	3590.3	3992.5
stat96v5	*11838*	*128*	*18.4*	*7200.0*	*7218.4*	*1*	*13119*	*64*	*21.2*	*7200.0*	*7221.2*
stein27	32	64	0.0	0.0	0.0	0	32	64	0.0	0.0	0.0
stein45	59	64	0.0	0.0	0.0	0	59	64	0.0	0.0	0.0
stocfor1	108	64	0.0	0.0	0.0	0	108	64	0.0	0.0	0.0
stocfor2	1994	64	0.3	0.1	0.4	0	1994	64	0.2	0.1	0.4
stocfor3	16167	64	6.2	0.5	6.7	0	16167	64	6.5	0.6	7.1
stockholm	25891	64	20.3	11.6	32.0	1	26971	64	21.7	0.8	22.5
stormG2_1000	*732642*	*64*	*4648.3*	*7200.0*	*11848.3*	*0*	*732642*	*64*	*4679.4*	*7200.0*	*11879.5*
stormg2-125	92103	64	45.6	1.9	47.5	0	92103	64	47.4	1.9	49.3

Table 6 continued

	SoPlex$_9$+QSopt_ex					SoPlex$_{50}$+QSopt_ex					
Instance	iter	p_{ex}	t_9	t_{ex}	t	R_0	iter	p_{ex}	t_{50}	t_{ex}	t
stormg2-27	20754	64	2.4	0.5	2.9	0	20754	64	2.7	0.4	3.1
stormg2-8	6863	64	0.6	0.2	0.8	0	6863	64	0.4	0.2	0.6
stormg2_1000	*732642*	*64*	*4660.6*	*7200.0*	*11860.6*	*0*	*732642*	*64*	*4685.5*	*7200.0*	*11885.5*
stp3d	148139	64	1238.6	7.2	1245.8	0	148139	64	1242.3	7.1	1249.5
sts405	434	64	0.5	0.5	1.1	0	434	64	0.7	0.4	1.1
sts729	907	64	1.3	1.5	2.7	0	907	64	1.8	1.5	3.2
swath	127	64	0.1	0.1	0.2	0	127	64	0.1	0.1	0.2
sws	1150	64	0.3	0.4	0.7	0	1150	64	0.3	0.3	0.7
t0331-4l	15487	64	14.7	8.2	22.9	0	15487	64	15.4	8.2	23.7
t1717	12220	64	7.0	2.0	9.0	0	12220	64	7.6	2.0	9.6
t1722	10612	64	3.3	0.6	3.9	0	10612	64	3.6	0.6	4.2
tanglegram1	478	64	0.3	1.4	1.6	0	478	64	0.3	1.4	1.8
tanglegram2	236	64	0.1	0.1	0.2	0	236	64	0.0	0.1	0.2
testbig	8010	64	4.4	0.2	4.6	0	8010	64	4.5	0.3	4.8
timtab1	20	64	0.0	0.0	0.0	0	20	64	0.0	0.0	0.0
timtab2	36	64	0.0	0.0	0.0	0	36	64	0.0	0.0	0.0
toll-like	717	64	0.1	0.0	0.1	0	717	64	0.1	0.1	0.1
tr12-30	696	64	0.0	0.0	0.0	0	696	64	0.0	0.0	0.0
transportmoment	9617	64	1.7	0.7	2.4	0	9617	64	1.9	0.6	2.4
triptim1	68167	64	93.3	1.2	94.5	0	68167	64	92.8	1.2	94.1
triptim2	204335	64	572.2	4.3	576.4	0	204335	64	567.7	4.3	572.0
triptim3	89514	64	156.7	2.7	159.4	0	89514	64	161.5	2.6	164.1
truss	21892	64	4.7	0.2	4.9	0	21892	64	4.6	0.1	4.7
tuff	212	64	0.0	0.0	0.1	0	212	64	0.0	0.0	0.1
tw-myciel4	11588	64	2.6	0.1	2.7	0	11588	64	2.3	0.1	2.3
uc-case11	39603	64	75.9	1.1	77.0	0	39603	64	75.3	1.1	76.4
uc-case3	34156	64	65.4	1.2	66.6	0	34156	64	66.7	1.2	67.9
uct-subprob	2988	64	0.5	0.1	0.6	0	2988	64	0.5	0.0	0.6
ulevimin	125858	64	108.8	1.0	109.8	0	125858	64	108.6	1.1	109.6
umts	5537	64	0.9	0.5	1.5	2	5549	64	0.7	0.2	0.8
unitcal_7	21824	64	8.9	0.7	9.5	0	21824	64	9.1	0.8	9.9
us04	338	64	0.2	0.2	0.4	0	338	64	0.5	0.2	0.7
usAbbrv-8-25_70	2434	64	0.1	0.1	0.2	0	2434	64	0.1	0.1	0.1
van	11014	64	5.2	1.8	6.9	0	11014	64	6.0	1.7	7.7
vpm1	130	64	0.0	0.0	0.0	0	130	64	0.0	0.0	0.0
vpm2	192	64	0.0	0.0	0.0	0	192	64	0.0	0.0	0.0
vpphard	11075	64	11.3	1.0	12.3	0	11075	64	11.5	1.0	12.5
vpphard2	8024	64	32.4	4.1	36.5	0	8024	64	33.1	4.2	37.2
vtp-base	75	64	0.0	0.0	0.0	0	75	64	0.0	0.0	0.0
wachplan	2033	64	0.5	0.2	0.7	0	2033	64	0.4	0.1	0.5
watson_1	*188103*	*128*	*366.3*	*7200.0*	*7566.3*	1	188496	64	369.2	10.6	379.8
watson_2	*333044*	*128*	*1519.4*	*7200.0*	*8719.4*	1	333083	64	1518.6	19.5	1538.1
wide15	229488	64	2733.8	8.5	2742.3	0	229488	64	2734.6	8.7	2743.3
wnq-n100-mw99-14	764	64	5.0	13.3	18.2	0	764	64	5.8	13.2	19.0
wood1p	142	64	0.1	0.8	1.0	0	142	64	0.2	0.8	1.1
woodw	1832	64	0.6	0.2	0.8	0	1832	64	0.6	0.1	0.7
world	70204	128	131.2	7200.0	7331.2	12	70320	64	132.1	4.5	136.6
zed	31	64	0.0	0.0	0.0	0	31	64	0.0	0.0	0.0
zib54-UUE	1855	64	0.1	0.1	0.2	0	1855	64	0.1	0.1	0.2