Online Resource

The figures in the paper examined scaling used in conjunction with the simplex algorithm. Figures 15–19 in this online resource correspond to Figures 2, 5, 8, 9, and 12 in the paper. The difference between these two sets of figures is that the figures in the online resource depict the use of CPLEX's primal simplex algorithm, while the figures in the paper depict the use of CPLEX's dual simplex algorithm. No scaling technique in either set of figures shows a discernible advantage over its peers.

The solution iterations without scaling appear to be higher (but within a standard deviation) in Figures 16 and 19. In Figure 19, the mean solution iterations without scaling does exceed the mean plus one standard deviation for all instances of IBM MPSX scaling. This fact leads to the conclusion that as a constraint matrix becomes increasingly poorly scaled, some scaling may be beneficial.

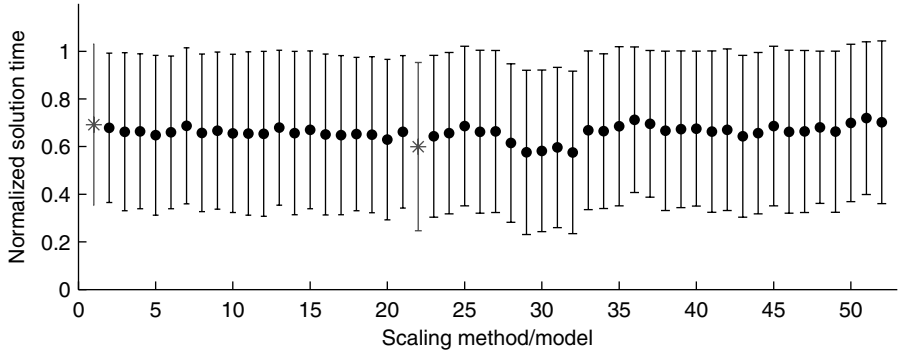

Fig. 15 The normalized solution time with primal simplex using CPLEX

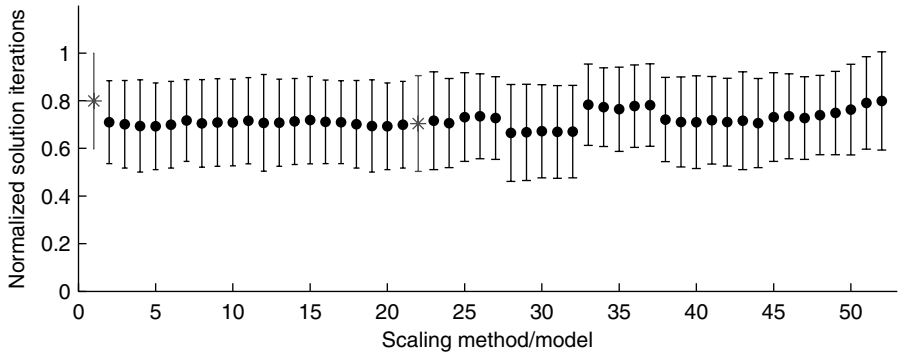

Fig. 16 The normalized solution iterations with primal simplex using CPLEX

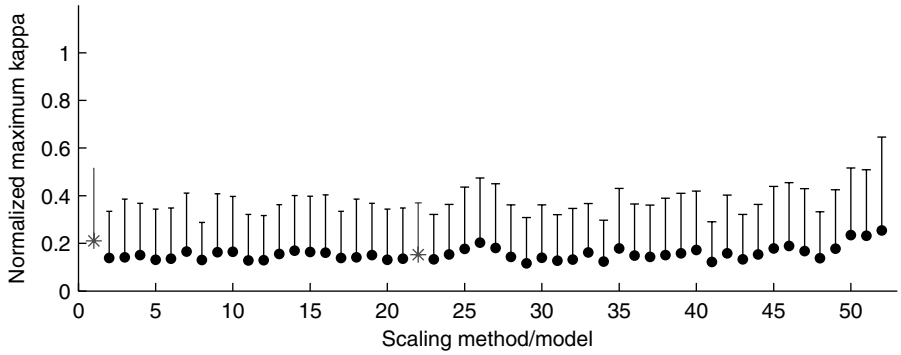

Fig. 17 The normalized maximum condition number with primal simplex using CPLEX

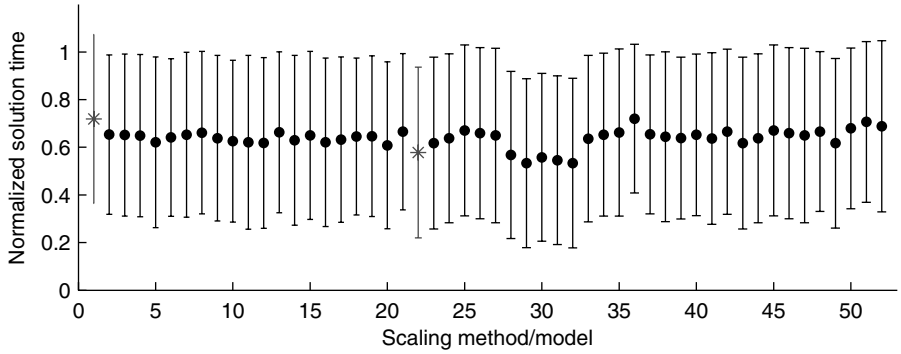

Fig. 18 The normalized solution time with primal simplex using CPLEX

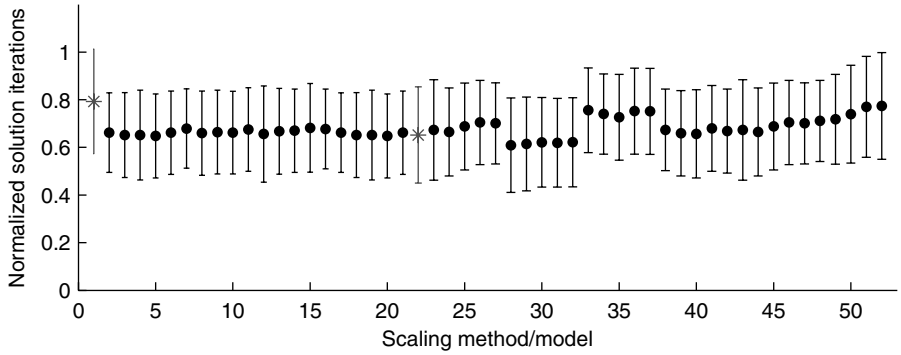

Fig. 19 The normalized solution iterations with primal simplex using CPLEX

The last experiments that were performed involved studying the combination of two scaling techniques on a given problem. The results from these experiments are given in Figures 20–33. Algorithm 1 (below) can be used to generate a complete listing of the scaling combinations used and the legends associated with these figures. In these figures, equilibration scaling followed by IBM MPSX scaling is highlighted in gray. In terms of using combinations, one might consider this particular combination a baseline for measurement. Certainly, IBM MPSX scaling was used by MPSX and OSL, and geometric mean scaling is an option in CPLEX.

In the figures of experiments using a combination of two scaling techniques, error bars were not included because of the size of the figures. There are ten different scaling techniques that are tested across one, two, four, six, and eight iterations, and equilibration. There are 2350 different combinations of these scaling techniques and models. Hence, the clusters of about 250 data points, which are identifiable in Figures 20–21, are representative of the division of data points along a scaling technique. For example, the first roughly 250 data points correspond to the scaling combinations where arithmetic mean is first applied.

Figures 20–21 show the normalized scaling time without and with presolve, respectively. Naturally, the combinations of scaling techniques are going to exhibit a higher overall scaling time than the single scaling techniques exhibited. A corresponding reduction in solution time would be required to warrant the use of these combinations. Unfortunately, no such result can be deduced from Figures 22–25, which depict the normalized solution time with various solution algorithms, and with and without presolve. These figures indicate no real trend or pattern in solution time relative to the combinations of scaling techniques employed here. The same lack of pattern is evident in the normalized solution iterations depicted in Figures 26–29.

Lastly, the normalized maximum condition number of the bases with various solution algorithms, with and without presolve, are given in Figures 30–33. Several different outlying combinations make the figures difficult to interpret. However, the results again show the result seen in the single scaling application experiments. That is, IBM MPSX scaling is relatively efficient at controlling the maximum condition number of the bases. However, its combination with some scaling techniques proved to degrade its performance relative to this measure.

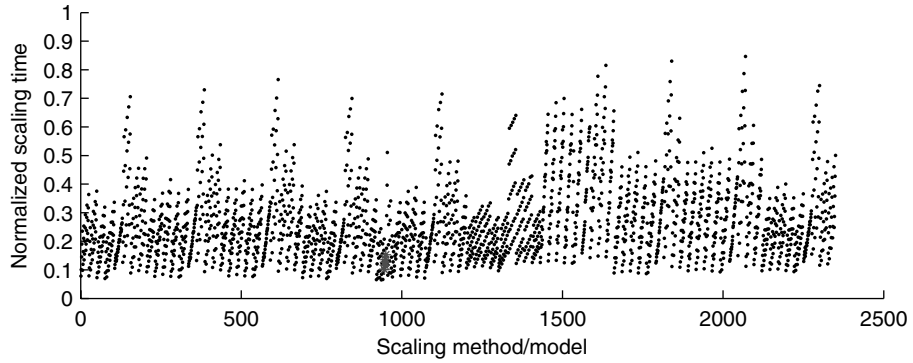

Fig. 20 The scaling time without presolve

```
begin
    Input: method[] ← { 'Arithmetic mean', 'de Buchet p = 1',
                        'de Buchet p = 2', 'Entropy', 'Equilibration',
                        'Geometric mean', 'IBM MPSX', 'L1-norm',
                        'L2-norm', 'Linf-norm', 'Binormalization' }
           iterat[]  ← { 1, 2, 4, 6, 8 }
    forall the method1 of method[] do
        for j ← 1 to 5 do
            forall the method2 of method[] do
                for i ← 1 to 5 do
                    print method1 and iterat[ j ] followed by
                          method2 and iterat[ i ]
                    if method2 == 'Equilibration' then break
                end
            end
            if method1 == 'Equilibration' then break
        end
    end
end
```

Algorithm 1: A generator for the legend for Figures 20–33: This legend indicates the scaling method/model and the associated iteration count. The order in which the scaling applications are generated corresponds to their position on the x-axis from left to right in the figures.

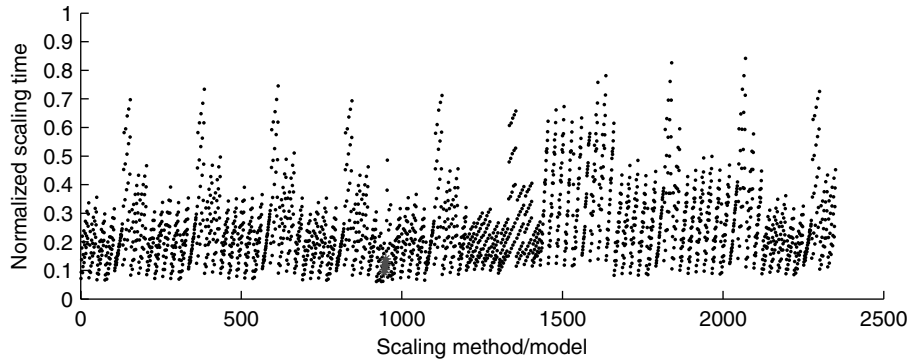

Fig. 21 The scaling time with presolve

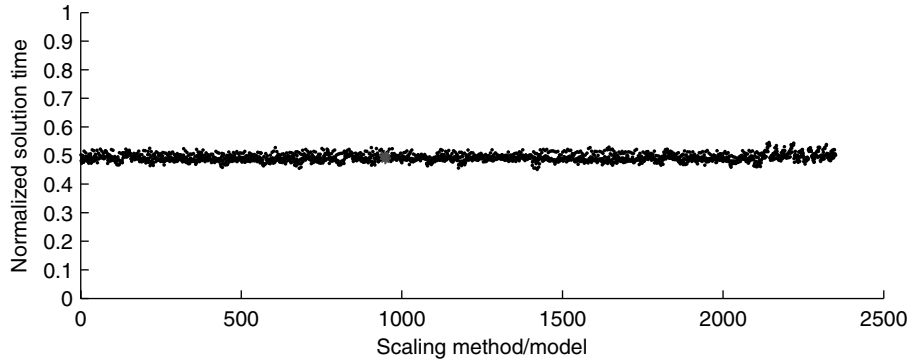

Fig. 22 The solution time without presolve using CPLEX primal simplex

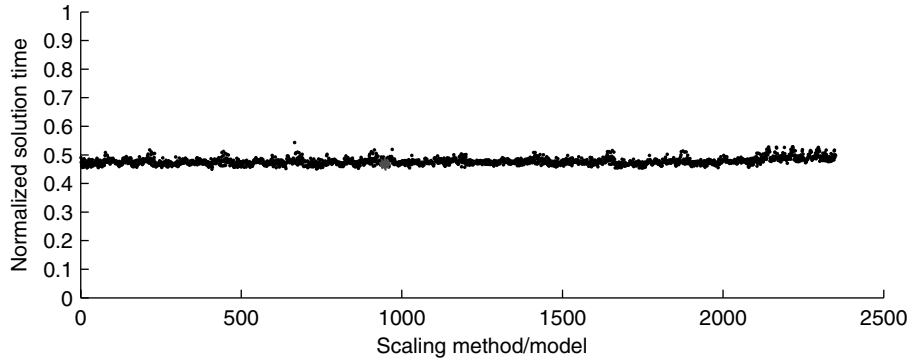

Fig. 23 The solution time with presolve using CPLEX primal simplex

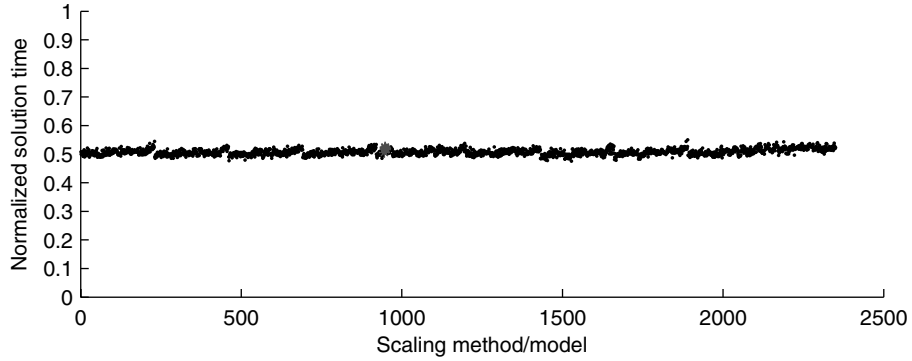

Fig. 24 The solution time without presolve using CPLEX dual simplex

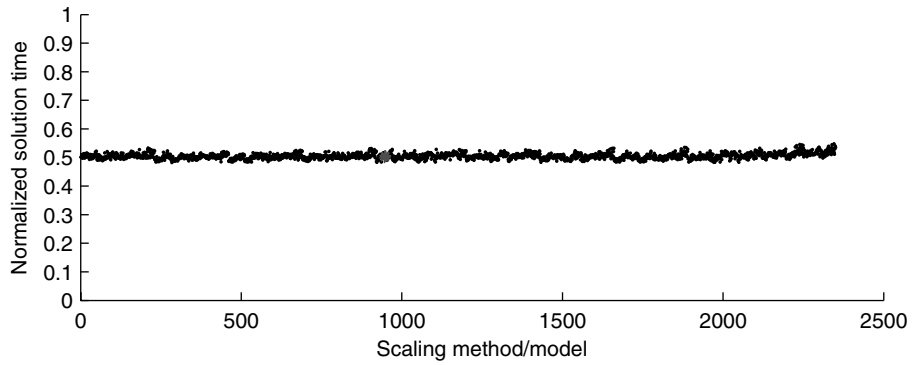

Fig. 25 The solution time with presolve using CPLEX dual simplex

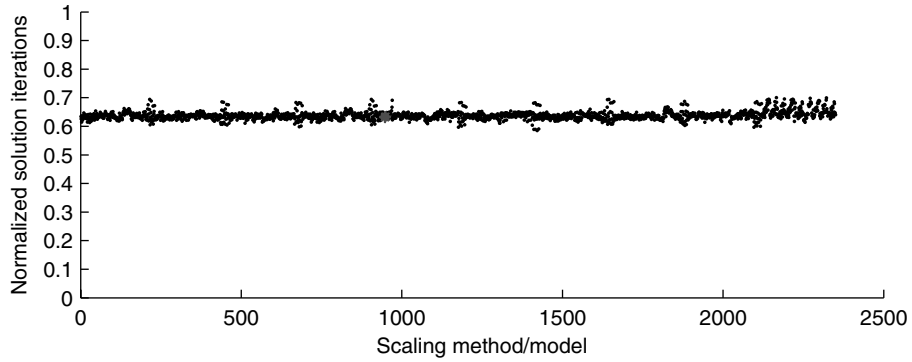

Fig. 26 The solution iterations without presolve using CPLEX primal simplex

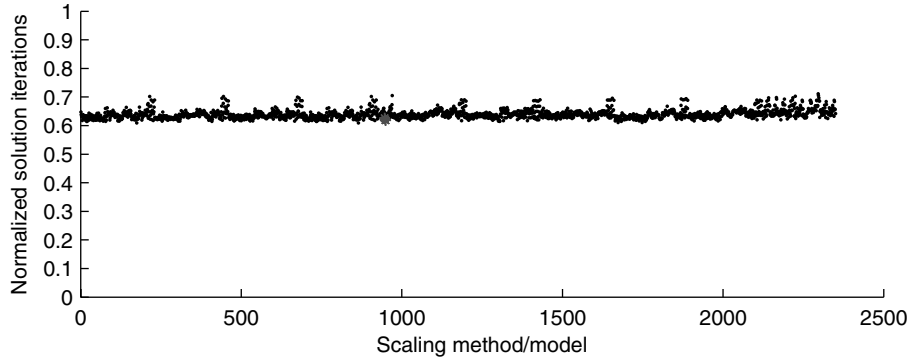

Fig. 27 The solution iterations with presolve using CPLEX primal simplex

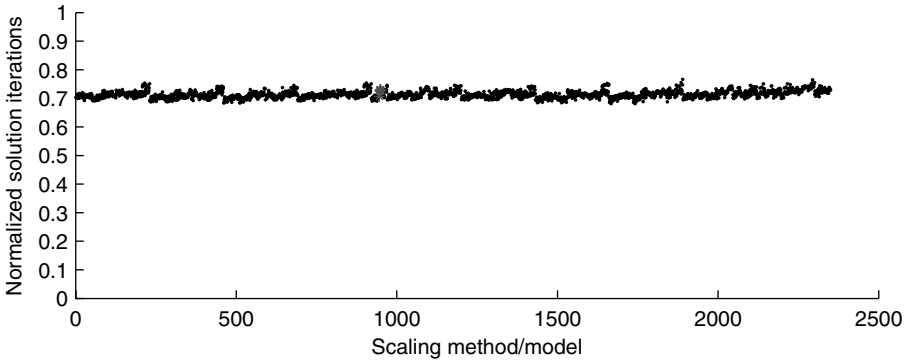

Fig. 28 The solution iterations without presolve using CPLEX dual simplex

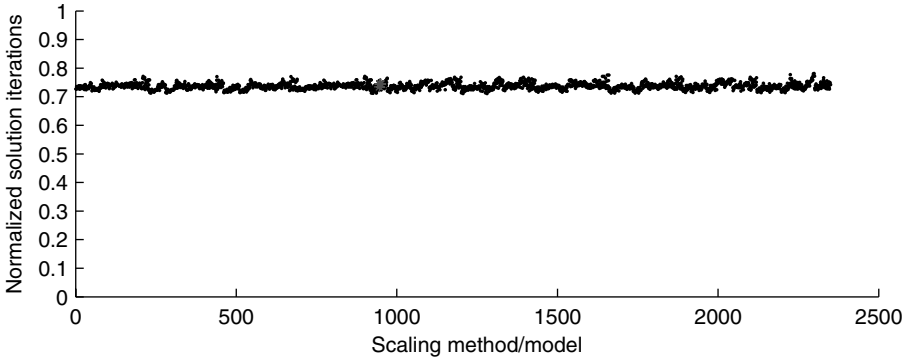

Fig. 29 The solution iterations with presolve using CPLEX dual simplex

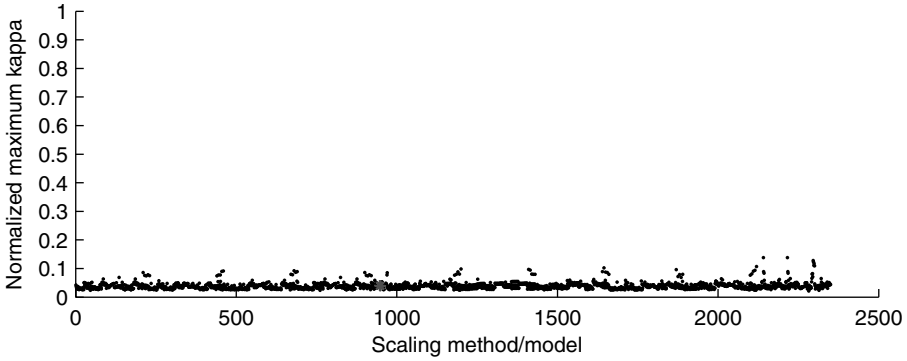

Fig. 30 The maximum condition number without presolve using CPLEX primal simplex

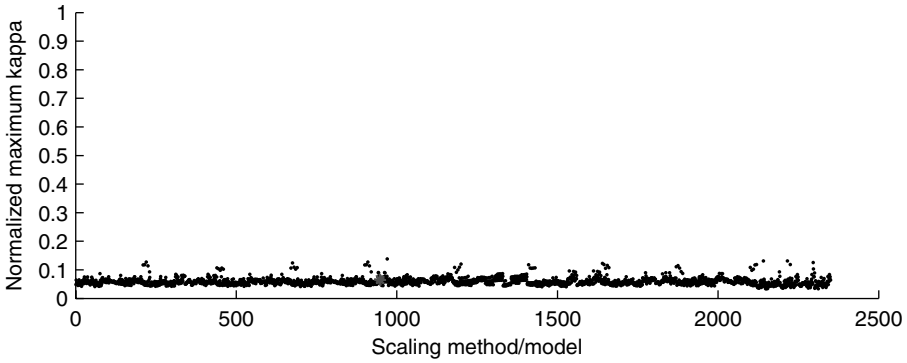

Fig. 31 The maximum condition number with presolve using CPLEX primal simplex

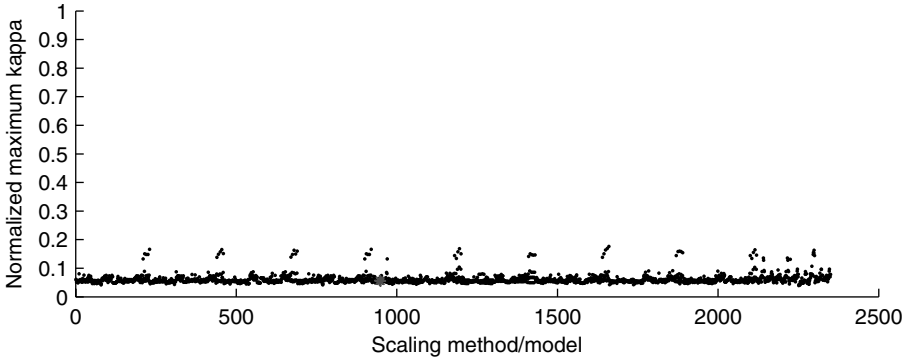

Fig. 32 The maximum condition number without presolve using CPLEX dual simplex

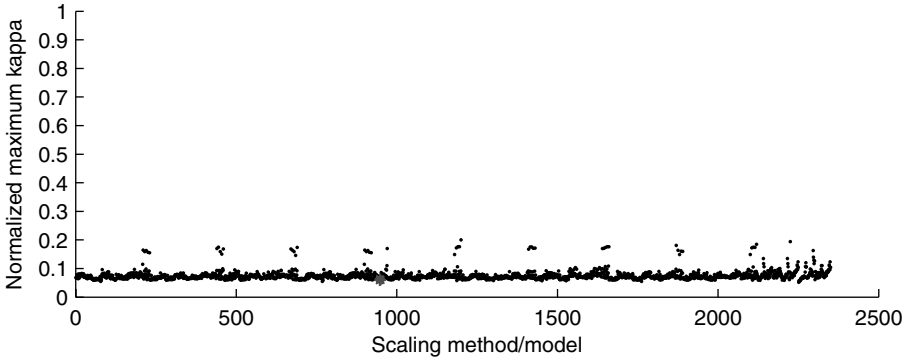

Fig. 33 The maximum condition number with presolve using CPLEX dual simplex