8. Appendix: Supplemental Material for "ECLIPSE: An Extreme-Scale Linear Program Solver for Web-Applications"

8.1. Proofs and Technical Details

Proof of Lemma 1.

$$\begin{aligned} g_\gamma(\lambda) =& \min_{x\in\mathcal{C}} \ \left\{c^T x + \frac{\gamma}{2}x^T x + \lambda^T(Ax-b)\right\} \\ \geq& \min_{x\in\mathcal{C}} \ \left\{c^T x + \lambda^T(Ax-b)\right\} + \min_{x\in\mathcal{C}} \frac{\gamma}{2}x^T x \\ \geq& g_0(\lambda) \end{aligned} \tag{13}$$

where, in arriving at the last line we used the fact that $\min_{x\in\mathcal{C}} \frac{\gamma}{2}x^T x \geq 0$.

On the other hand, we have that:

$$\begin{aligned} g_0(\lambda) =& \min_{x\in\mathcal{C}} \ \left\{c^T x + \lambda^T(Ax-b)\right\} \\ =& \min_{x\in\mathcal{C}} \ \left\{c^T x + \frac{\gamma}{2}x^T x + \lambda^T(Ax-b) - \frac{\gamma}{2}x^T x\right\} \\ \geq& \min_{x\in\mathcal{C}} \ \left\{c^T x + \frac{\gamma}{2}x^T x + \lambda^T(Ax-b)\right\} \\ & - \max_{x\in\mathcal{C}} \frac{\gamma}{2}x^T x \\ =& g_\lambda(\lambda) - \frac{\gamma}{2}\vartheta \end{aligned} \tag{14}$$

where,

$$\vartheta = \max_{x\in\mathcal{C}} x^T x = \sum_{i\in[I]} \max_{x_i\in\mathcal{C}_i} x_i^T x_i$$

If $\mathcal{C}_i$ is the unit simplex then $\max_{x_i\in\mathcal{C}_i} x_i^T x_i = 1$ If $\mathcal{C}_i$ is the unit hypercube, then $\max_{x_i\in\mathcal{C}_i} x_i^T x_i = |J|$.

Combining the results from (13) and (14) we arrive at the result of the Lemma. □

Proof of Lemma 2. We present an outline of the proof following Mangasarian & Meyer (1979). Note that LP (1) can be written in the following form:

$$\Gamma := \min \ \ c^T x \ \ \text{s.t.} \ \ \tilde{A}x \geq \tilde{b} \tag{15}$$

where, the inequality $\tilde{A}x \geq \tilde{b}$ describes the polyhedral constraint set in (1)

$$\{x : Ax \leq b\} \cap \{x : x \in \mathcal{C}\}.$$

Note that Γ is the optimal objective value of (15).

We wish to show that for all $\gamma > 0$ sufficiently small, a solution to the following perturbed problem

$$\min \ \ c^T x + \gamma/2\|x\|_2^2 \ \ \text{s.t.} \ \ \tilde{A}x \geq \tilde{b} \tag{16}$$

solves Problem (15). The idea of the proof works as follows. Starting with a primal solution to (15) (and a solution for the dual of this LP), we propose a candidate primal-dual pair that satisfies the optimality conditions of (16). The construction of this primal-dual pair depends upon the following problem:

$$\min \ \ \|x\|_2^2 \ \ \text{s.t.} \ \ \tilde{A}x \geq \tilde{b}, \ \ c^T x \leq \Gamma \tag{17}$$

which by construction requires knowing the optimal objective value Γ for Problem (15).

Let $\bar{x}$ be an optimal solution to (17); and let $\bar{v}, \bar{u}$ be the optimal dual variables corresponding to the constraints $\tilde{A}x \geq \tilde{b}$ and $c^T x \leq \Gamma$ respectively. One can write down the Karush-Kuhn-Tucker (KKT)-optimality conditions for Problem (17):

$$\begin{aligned} 2\bar{x} - \tilde{A}^T\bar{v} + \bar{u}c &= 0 \\ \bar{v}\cdot(\tilde{A}\bar{x} - \tilde{b}) &= 0 \\ c^T\bar{x} &= \Gamma \end{aligned} \tag{18}$$

and in addition we have feasibility conditions:

$$(\tilde{A}\bar{x} - \tilde{b}) \geq 0, \ \ \bar{v} \geq 0, \ \ \bar{u} \geq 0.$$

Note that $\bar{x}$ is also an optimal solution to the LP (15) — let $\bar{w}$ be a corresponding optimal dual solution.

We consider two cases. (i) If $\bar{u} = 0$, then the multiplier corresponding to the constraint $c^T x \leq \Gamma$ is inactive. In this case, recall that $\bar{x}$ is an optimal primal solution to (16). Furthermore, for any $\gamma \geq 0$, the candidate $\bar{w} + \gamma\bar{v}$ is an optimal solution for the dual variables corresponding to the constraint $\tilde{A}x \geq \tilde{b}$ in (16). That is, for any $\gamma \geq 0$, $(\bar{x}, \bar{w} + \gamma\bar{v})$ is a primal-dual pair that satisfies the KKT system of Problem (16).

We consider the second case (ii) that is, $\bar{u} > 0$. In this case, the following primal-dual tuple $(\bar{x}, \bar{v}/\bar{u})$ is an optimal solution to the KKT-system of Problem (16) for $\gamma = 1/\bar{u}$. In fact, investigating the KKT-system for Problem (17) above, any convex combination of $\bar{w}$ and $\bar{v}/\bar{u}$ will satisfy the KKT system for Problem (16) – more precisely,

$$(\bar{x}, (1-\tau)\bar{w} + \tau(\bar{v}/\bar{u}))$$

is a KKT point for (16) for any $\gamma = \tau/\bar{u} \leq 1/\bar{u}$ with $\tau \in [0, 1]$. This completes the proof of the lemma. □

Proof of Lemma 3. Let us recall the definition of the dual function $g_\gamma(\lambda)$:

$$\begin{aligned} g_\gamma(\lambda) =& \min_{x\in\mathcal{C}} \ \left\{(c + \lambda^T A)^T x + \frac{\gamma}{2}x^T x - \lambda^T b\right\} \\ =& \psi_\gamma(c + A^T\lambda) - \lambda^T b \end{aligned} \tag{19}$$

where,

$$\psi_\gamma(w) := \min_{x\in\mathcal{C}} \left\{w^T x + \frac{\gamma}{2}x^T x\right\}.$$

For every w, let $\hat{x}(w) \in \operatorname{argmin}_{x\in\mathcal{C}} \left\{w^T x + \frac{\gamma}{2} x^T x\right\}$. For $\gamma > 0$, the minimizer $\hat{x}(w)$ is unique; and hence the function $w \mapsto \psi_\gamma(w)$ is differentiable. By Danskin's Theorem (Bertsekas, 1999), the gradient of the map $w \mapsto \psi_\gamma(w)$ is given by: $\nabla\psi_\gamma(w) = \hat{x}(w)$.

For $\gamma = 0$, $\hat{x}(w)$ is not unique, and hence $w \mapsto \psi(w)$ is not differentiable. In this case, a subgradient of $\psi(w)$ exists and is given by $\partial\psi(w) = \hat{x}(w) \in \operatorname{argmin}_{x\in\mathcal{C}} w^T x$.

Let $\hat{x}(\lambda)$ be the minimizer of the optimization problem wrt x in (19) (for a given λ). Then we have (using the chain rule for example),

$$\nabla g_\gamma(\lambda) = A\hat{x}(\lambda) - b.$$

We also have the following chain of inequalities:

$$\begin{aligned}\|\nabla g_\gamma(\lambda) - \nabla g_\gamma(\lambda')\| &= \|A(\hat{x}(\lambda) - \hat{x}(\lambda'))\| \\ &\le \|A\|_S \|\hat{x}(\lambda) - \hat{x}(\lambda')\|. \end{aligned} \quad (20)$$

where $\|\cdot\|_S$ deonotes the Spectral norm. Now recall that:

$$\hat{x}(\lambda) = \Pi_{\mathcal{C}}\left(\frac{-A^T\lambda - c}{\gamma}\right)$$

Hence,

$$\begin{aligned}&\|\hat{x}(\lambda) - \hat{x}(\lambda')\| \\ &\le \left\|\Pi_{\mathcal{C}}\left(\frac{-A^T\lambda - c}{\gamma}\right) - \Pi_{\mathcal{C}}\left(\frac{-A^T\lambda' - c}{\gamma}\right)\right\| \\ &\le \frac{1}{\gamma}\|A\|_S\|\lambda - \lambda'\|, \end{aligned} \quad (21)$$

where, the last inequality follows from the observation that the projection operator $\Pi_{\mathcal{C}}(\cdot)$ is a contraction:

$$\|\Pi_{\mathcal{C}}(u) - \Pi_{\mathcal{C}}(v)\| \le \|u - v\|;$$

and the definition of the spectral norm of A. Therefore, combining (20) and (21) we get:

$$\|\nabla g(\lambda) - \nabla g(\lambda')\| \le \frac{1}{\gamma}\|A\|_S^2\|\lambda - \lambda'\| \quad (22)$$

which concludes the proof of the Lemma. □

8.2. Details on Computing $\|A\|_S$ with regard to step-size selection

Note that in this paper, we are interested in matrices of the form: $A = [A^{(1)}; A^{(2)}]$ (cf Section 2) where, $A^{(1)}$ has $m = O(1)$-many rows and $A^{(2)}$ is of the form:

$$A^{(2)} = \begin{pmatrix} D_{11} & \dots & D_{1I} \\ & \dots & \\ D_{m_2 1} & \dots & D_{m_2 I} \end{pmatrix} \quad (23)$$

where, D_{ij} are diagonal matrices.

We proceed to obtain an upper bound on the largest singular value of A — to this end, we make use of the following chain of inequalities (in the display below we take an $x \in \mathbb{R}^n$):

$$\begin{aligned}\|Ax\| &= \left\|\begin{pmatrix} A^{(1)}x \\ A^{(2)}x \end{pmatrix}\right\|_2 \\ &= \left(\|A^{(1)}x\|_2^2 + \|A^{(2)}x\|_2^2\right)^{1/2} \\ &\le \left(\|A^{(1)}\|_S^2 + \|A^{(2)}\|_S^2\right)^{1/2}\|x\|_2 \end{aligned}$$

We now consider $x \in \mathbb{R}^m$; and using a partition for $x = [x_1; x_2]$ such that $x_1 \in \mathbb{R}^{m_1}$ and $x_2 \in \mathbb{R}^{m_2}$, we have:

$$\begin{aligned}\|A^T x\|_2 &= \|(A^{(1)})^T x_1 + (A^{(2)})^T x_2\|_2 \\ &\le \|(A^{(1)})^T x_1\|_2 + \|(A^{(2)})^T x_2\|_2 \\ &\le \|A^{(1)}\|_S\|x_1\|_2 + \|A^{(2)}\|_S\|x_2\|_2 \\ &\le C(\|x_1\|_2 + \|x_2\|_2) \\ &\le \sqrt{2}C\|x\|_2 \end{aligned}$$

where, above $C = \max\{\|A^{(1)}\|_S, \|A^{(2)}\|_S\}$; and the last inequality follows by using the fact that $\sqrt{(a^2+b^2)/2} \ge (a+b)/2$ for any two scalars a, b. Therefore, combining the above inequalities we have:

$$\max_{\|x\|_2=1} \|A^T x\|_2 \le \min\left\{\sqrt{2}C, \left(\|A^{(1)}\|_S^2 + \|A^{(2)}\|_S^2\right)^{1/2}\right\}. \quad (24)$$

In light of (24), to obtain a bound on the spectral norm (i.e., largest singular value) of A, we compute the spectral norms of $A^{(1)}$ and $A^{(2)}$.

As the matrix $A^{(1)}$ has a small number of rows i.e., $O(1) = m_1 \ll n$, in order to compute the spectral norm of $A^{(1)}$, we find the largest singular value of the $m_1 \times m_1$ matrix product $A^{(1)}(A^{(1)})^T$ (this computation is done once at the start of the algorithm—in a distributed fashion). As m_1 is usually very small, this matrix operation costs $O(n)$. When m_1 is larger than a few thousand, we use a randomized power method (Halko et al., 2011) to compute the largest eigenvalue. The matrix $A^{(2)}$ composes of diagonal matrices (23)—to compute the largest singular value of $A^{(2)}$, we use the power method (the randomized power method (Halko et al., 2011) is also found to work well). Every iteration of this algorithm requires multiplications of the form $A^{(2)}u$ and $(A^{(2)})^T v$ for vectors u, v of suitable dimensions. These multiplications are cheap due to

- the diagonal-concatenation structure of $A^{(2)}$
- the diagonal blocks have very few nonzero entries (See Sections 4, 5 and 6 for discussions on the structure of these diagonal blocks and anticipated number of nonzeros in typical applications that we consider).

Roughly speaking, these products can be computed with cost: $\sum_{i\in[I]} Km_2$, where K denotes the average number of

non-zeros in each diagonal block — usually in applications (as discussed in the main text), K is at most a thousand. Moreover, the product operation can be parallelized across $i \in [I]$. Once again, this can be computed off-line (for once) before the start of the algorithm.

8.3. Simulation Experimental Details

The first two simulation experiments are run on macOS 10.15.1, 2.4 GHz 8-Core Intel Core i9, with 32GB memory.

8.3.1. Validity Experiments

For the validity experiment shown in Figure 2, the full details are given below.

- The data has been simulated using a uniform distribution with a fixed seed. The size of the dataset is $I = 10000$ and $J = 100$ with $n = IJ = 10^6$. We are attaching the code, through which the data can be generated.
- No data was excluded in any pre-processing steps.
- Throughout the simulation we have chosen $\gamma = 10^{-3}$ and initialized all entries of λ by 1.
- To show the plot, we have done a single run. It can be easily modified to do multiple runs so that we can have similar plots. We found the trend to hold across simulations.
- We display the Objective value of the optimization problem (primal and dual) along the y-axis and the iterations on the x-axis.
- The stopping criteria is the relative duality gap given by

$$\frac{|\text{primalObj}_t - \text{dualObj}_t|}{1 + |\text{primalObj}_t| + |\text{dualObj}_t|} < \epsilon$$

  where we choose $\epsilon = 10^{-5}$ and the primal objective is obtained using the *feasible* primal solution—this is generated by modifying $\hat{x}(\lambda)$ so that it becomes feasible for $Ax \leq b$. In our example, it is possible to do that because there are only two constraints in A. We identify,

$$\delta^* = \max\{\delta : \delta a_i^T \hat{x}(\lambda) \leq b_i \ \ \forall i\}$$

  and choose $\hat{x}_{feasible} = \delta^* \hat{x}(\lambda)$.

8.3.2. Scalability Experiments

The details for the scalability experiment plot (Figure 3) are given below:

- Similar to before, we consider random samples from the uniform distribution with different seeds. We choose $I = 100, 500, 1000, 5000, 10000$ and $J = 100$ and $n = IJ$. The data can be generated by running the script.
- No data was excluded in any pre-processing steps.
- We choose $\gamma = 10^{-2}$ and initialized all entries of λ by 1.
- The stopping criteria was chosen exactly as the previous experiment with $\epsilon = 10^{-5}$.
- We run the system for 50 evaluations and report the average.
- We report the average running time of the different algorithm along with the error bars.

8.3.3. Comparison with Average Method in Extreme Scale

For the extreme-scale problem, the details of the results for Table 1 are given below.

- We choose $I = 10^4, 10^5, 10^6$ and $J = 100$. This makes $n = 10^6, 10^7, 10^8$. The same simulation from uniform distribution is done with a fixed seed. We pick split size as 1000 and 10000.
- No data was excluded in any pre-processing steps.
- We split users accordingly to the split size and solve the problem in each group for the split method.
- Similar to before, we choose $\gamma = 10^{-3}$, initialized $\lambda = 1$ and the stopping criteria was same as above with $\epsilon = 10^{-5}$.
- We show the primal objective value $\hat{x}^T c$ and the primal residual which is defined as

$$\text{Primal Residual} = \|(A\hat{x} - b)\|/(1 + \|b\|) \quad (25)$$

- We also compute the primal feasibility violation which is defined as $\|(A\hat{x} - b)_+\|/(1 + \|b\|_2)$. In our simulations we observe that this primal feasibility violation is about 33% of the primal residual for ECLIPSE. For the averaging methods the primal solution was found to be feasible (this is by construction). However, these solutions were found to be in the interior of the feasible region, leading to a sub-optimal objective value.
- We run this specific large scale simulation on a Red Hat Enterprise Linux Workstation 7.6 (Maipo) with 16 Processors each running Intel Xeon Silver 4108 CPU @1.80GHz and a total memory of 64 GB.

8.4. Real-World Experimental Details

8.4.1. Validity Experiments

To validate the extreme-scale solver that has been implemented in Scala/Spark, we consider a small problem that can be easily solved and then we artificially replicate the problem. By doing so, the solution keeps on repeating. That

is, we start with a small problem of the form

$$\begin{aligned} \underset{x}{\text{Maximize}} \quad & x^T p \\ \text{subject to} \quad & \sum_i x_{ij} r_{ij} \leq b_j \qquad \forall j \\ & \sum_j x_{ij} = 1 \qquad \forall i \\ & 0 \leq x \leq 1 \end{aligned} \tag{26}$$

where $I = 5, J = 5$ and $n = IJ = 25$. p, r are chosen once of length n from the uniform distribution and fixed. We can easily solve this problem and let $\hat{\lambda}_{5\times 1}$ be the dual solution.

We then replicate this problem intentionally, by keeping using the same p and r. That is, with a repeat factor of K, we create K more users and K more items. Thus we have,

$$\tilde{p}_{kJ+i} = \begin{pmatrix} 0 & \dots & p_{i1}, p_{i2}, \dots, p_{iJ} & \dots & 0 \end{pmatrix},$$

for $k = 0, \dots, K-1$ where the original p_i vector is set at the k-slot with 0 padded on either side such that we increase the total items to $J = 5K$ but the user still only has the original 5 items. As a simple example, for $I = 2$, $J = 3$ and $K = 2$, if we have

$$\begin{aligned} p_1 &= \begin{pmatrix} 0.1 & 0.2 & 0.3 \end{pmatrix} \\ p_2 &= \begin{pmatrix} 0.4 & 0.5 & 0.6 \end{pmatrix} \end{aligned}$$

then

$$\begin{aligned} \tilde{p}_1 &= \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0 & 0 & 0 \end{pmatrix} \\ \tilde{p}_2 &= \begin{pmatrix} 0.4 & 0.5 & 0.6 & 0 & 0 & 0 \end{pmatrix} \\ \tilde{p}_3 &= \begin{pmatrix} 0 & 0 & 0 & 0.1 & 0.2 & 0.3 \end{pmatrix} \\ \tilde{p}_4 &= \begin{pmatrix} 0 & 0 & 0 & 0.4 & 0.5 & 0.6 \end{pmatrix} \end{aligned}$$

With a similar translation to $\tilde{r}$ from r it is easy to see that if we replicated problem, it has the form

$$\begin{aligned} \underset{x}{\text{Maximize}} \quad & x^T \tilde{p} \\ \text{subject to} \quad & \sum_i x_{ij} \tilde{r}_{ij} \leq b_j \qquad \forall j \\ & \sum_j x_{ij} = 1 \qquad \forall i \\ & 0 \leq x \leq 1 \end{aligned} \tag{27}$$

where $\tilde{b} = (b, \dots, b)$ is repeated K times. Because of this separable structure, the new dual solution is

$$\tilde{\lambda}_{5K\times 1} = (\hat{\lambda}_{5\times 1}, \dots, \hat{\lambda}_{5\times 1})$$

repeated K times. We use this data generation mechanism to test our production solver. The code to generate the data is given in the supplementary material. We do not consider the explicit separable nature in the problem, but we were able to match the exact dual upto a precision of 10^{-2}.

- We considered $I = 100, J = 50, K = 10^7$ and were still able to converge in a relatively short amount of time. We choose $\gamma = 10^{-3}$, initialized all entries of λ by 1, and the stopping criteria was the relative change in the norm of the dual variables with a threshold of 10^{-6}.
- These experiments were run on the developmental cluster with 800 executors and 8GB of memory per executor.

8.4.2. Scalability Experiments

After the above validity experiment, we ran experiments with real datasets, the results of which are presented in Table 2. For a comparison, we also implemented SCS (an ADMM based method) in Scala/Spark, but it could not scale to the problems at hand as it did not leverage the problem structure. The details of our experiments are as follows.

- The data was collected from live traffic and internal email/notification generation pipelines. We also considered the candidate sets from the people recommendation system.
- Due to privacy concerns we will not be able to share this data. All simulated data and the data generation mechanism have been shared.
- We choose $\gamma = 10^{-3}$, initialized all entries of λ by 1, and the stopping criteria was the relative change in the norm of the dual variables with a threshold of 10^{-6}. Note that since we were working with the general system, we could not use the relative duality gap as obtaining a feasible solution is non-trivial.
- We let the algorithms run till convergence was reached.
- We describe the running time using our internal datasets.
- As before, the experiments were run on the developmental cluster with 800 executors and 8GB of memory per executor.