Dual optimization for Newsvendor-like problem

Chuwen

February 11, 2021

Contents

1 Introduction

This paper is concerned with minimizing a newsvendor-like objective $f : \mathbb{R}^n \to \mathbb{R}$,

$$
\begin{aligned}
&\min f(\delta, \epsilon) \\
\textbf{s.t.}& \\
&y + \delta - \epsilon = b \\
&y \in \Omega_y \subseteq \mathbb{R}^n, \delta \in \mathbb{R}^n_+, \epsilon \in \mathbb{R}^n_+
\end{aligned}
\tag{1.1}
$$

where f is a convex function of δ, ϵ. The right-hand-side on the binding constraints is in the positive orthant: $b \in \mathbb{R}^n_+$. In the basic settings, let y be the ordering quantity quantities in a multi-item multi-period newsvendor problem, one minimizes the total expected cost:

$$
\min_{y \in \mathbb{R}_+} \mathbf{E}\left(h \cdot e^\top \max\{y - b, 0\} + p \cdot e^\top \max\{b - y, 0\}\right)
$$

Once the expectation operator is dropped, it is easy to verify the equivalence of such deterministic version to the problem (1.1) above. This problem is motivated from applications in device maintenance, inventory management, crew scheduling and so on.

Let $\lambda \in \mathbb{R}^n$ be the Lagrangian multiplier, we have the Lagrangian dual function,

$$
\begin{aligned}
\phi(\lambda) = &\min_{\delta, \epsilon} f(\delta, \epsilon) + \lambda^\top \delta - \lambda^\top \epsilon + \min_y \lambda^\top y - \lambda^\top b \\
\textbf{s.t.}& \\
&y \in \Omega_y \\
&\delta \in \mathbb{R}^n_+, \epsilon \in \mathbb{R}^n_+
\end{aligned}
\tag{1.2}
$$

with two independent subproblems. For δ, ϵ we have a convex optimization problem in the positive orthant. We also assume minimizing the linear objective under $y \in \Omega_y$ can be well-solved. In later sections we show some special cases where Ω_y may be further decomposed into smaller problems.

Denote $f^\star$ is the optimal value of (1.1), $\phi^\star$ is the optimal value of $\max \phi(\lambda)$.

1.1 Affine case

Let $f = p^\top \delta + h^\top \epsilon, p, h \in \mathbb{R}^n_+$, we have

$$
\phi(\lambda) = \min_{\delta, \epsilon} (p + \lambda)^\top \delta + (h - \lambda)^\top \epsilon + \min_y \lambda^\top y - \lambda^\top b \tag{1.3}
$$

Then ϕ is unbounded unless $\lambda \in \Lambda$ where $\Lambda = \{\lambda : \lambda \in [-p, h]\}$, in which case

$$\phi(\lambda) = \min_{y \in \Omega_y} \lambda^\mathsf{T} y - \lambda^\mathsf{T} b,\ \lambda \in \Lambda$$

and $\delta^\star(\lambda), \epsilon^\star(\lambda) = 0$ are corresponding optimizers for any $\lambda \in \Lambda$

2 Dual Optimization

2.1 Conditions for strong duality

We know Lagrangian relaxation produces a bound up to linear relaxation of a problem with the ”easy” constraints and the convex hull of relaxed constraints.

(Review Here).

Lemma 2.1. *if* $\Omega_y = \{y \in \mathbb{R}^n : y \in \Omega, y \in \mathbb{Z}^n\}$. *Then we have the following relation for dual function,*

$$\phi^\star = \min_{\delta,\epsilon} f(\delta, \epsilon) \quad \textbf{\textit{s.t.}}\ y + \delta - \epsilon = b,\ y \in conv(\Omega_y)$$

The above lemma immediately allows us to have strong duality by definition of perfect formulation. We first define set Y be the intersection of the hyperplane and $\mathsf{conv}(\Omega_y)$,

$$Y = \{(y, \delta, \epsilon) : y + \delta - \epsilon = b,\ y \in \mathsf{conv}(\Omega_y)\} \tag{2.1}$$

Then we have the following results,

Corollary 2.1.1. *We conclude the strong duality holds since Y is already* a perfect formulation *in the sense that $Y = \mathsf{conv}(Y)$.*

We conclude $\phi^\star = f^\star$ since $f^\star = \min_{(y,\delta,\epsilon)\in \mathsf{conv}(Y)} f(\delta, \epsilon)$.

2.2 Subgradient Method

To solve the reduced problem for λ, we consider a class of subgradient methods:

$$\lambda_{k+1} = \mathbf{P}(\lambda_k + s_k d_k) \tag{2.2}$$

where $\mathbf{P}$ is the projection onto dual space Λ, in the affine case (1.3), we compute the projection onto box $\Lambda = [-p, h]$. d_k is the update direction for current iteration and s_k is the step size using target-based rule:

$$s_k = \gamma_k \frac{\phi^\star - \phi(\lambda_k)}{\|d_k\|^2} \tag{2.3}$$

In practice, we take primal feasible values which over estimate the $\phi^\star$.

During the progress on dual problem, we compute an weighted average solution $\bar{y}_k$ from the convex combination of previous iterations: $\{y_i\}_{i=1,...k}$ and each y_i solves $\phi_i = \phi(\lambda_i)$, $y_i = \arg\min_{y\in\Omega_y} \lambda_i^\mathsf{T} y$.

$$\begin{aligned}
\bar{y}_k &= \sum_k^i \alpha_i^{(k)} y_i, \quad \sum_k^i \alpha_i^{(k)} = 1, \alpha_i^{(k)} \geq 0 && (2.4)\\
&= (1-\alpha_k)\cdot \bar{y}_{k-1} + \alpha_k \cdot y_k && (2.5)
\end{aligned}$$

The second equation (2.5) rephrases the convexity in a recursive manner that may help in programming. By taking $g_k = y_k - b$, then g_k is a subgradient of ϕ at λ_k:

$$g_k \in \partial\phi_k \tag{2.6}$$

The search direction can be computed from subgradient.

$$\begin{aligned}
\lambda_{k+1} &= \mathbf{P}(\lambda_k + s_k g_k)\\
s_k &= \gamma_k \frac{\phi^\star - \phi(\lambda_k)}{\|g_k\|^2}
\end{aligned} \tag{2.7}$$

As a comparison, we can use a convex alternative such that $d_k = \bar{y}_k - b$. Similarly, it can be expressed as convex combinations.

$$d_k = (1-\alpha_k)\cdot d_{k-1} + \alpha_k \cdot g_k \tag{2.8}$$

For simplicity, we refer to the convex choice (2.8) whenever term d_k is used.

The dual subgradient algorithm can be summarized as follows. $\varepsilon, \varepsilon_s$ are the tolerance parameter for objective gap and stepsize, respectively. $\varepsilon > 0, \varepsilon_s > 0$. At each iteration k, let $\gamma_k < 2, \alpha_k = \frac{1}{k}$.

Algorithm 1: The Subgradient Algorithm

Initialization. $\alpha_0 = 1, \lambda_0 = e, \gamma_0 = 1$

while $\phi^\star - \phi_k \geq \varepsilon$ ***and*** $s_k \geq \varepsilon_s$ **do**

  Let current iteration be k

  Update the multipliers by direction and stepsize, either by (2.2), (2.3) or (2.7).

  Solve dual problem ϕ_k by (1.2) and compute subgradient g_k respectively.

end

It is obvious to see the solutions during dual optimization $(y, \epsilon, \delta) = (y_k, 0, 0)$ are feasible if and only if we can find $y_k = d$, which in general will not hold. This motivates the following algorithm based on linear programming theory.

Algorithm 2: Recovery Algorithm

$$\begin{aligned}
\epsilon_k &= \max\{y_k - b, 0\} \\
\delta_k &= \max\{b - y_k, 0\} \\
\bar{\epsilon}_k &= \max\{\bar{y}_k - b, 0\} \\
\bar{\delta}_k &= \max\{b - \bar{y}_k, 0\}
\end{aligned} \tag{2.9}$$

To simplify our presentation, let z be a function of y such that $z_k = z(y_k) = f(\delta_k, \epsilon_k)$, then z is also convex in y since both function f and $\max\{\cdot, 0\}$ are convex. It's also worth to notice that $\bar{\epsilon}_k$ should not be calculated as running averages: $\bar{\epsilon}_k \neq \sum_k^i \alpha_i^{(k)} \epsilon_i$. For such an "averaged" solution, we let $\bar{z}_k = z(\bar{y}_k)$. We later find the recovery algorithm achieves at the optimal objective.

2.3 Overview

We first review several features for the subgradient method regarding parameters γ_k, α_k and search direction d_k produced from convex combinations.

The target based rule are well-known as the Polyak rule Polyak (1967). The idea of using previous searching directions is introduced to accelerate the subgradient method and provide a better stopping criterion, see Camerini et al. (1975), Brännlund (1995), Barahona and Anbil (2000). Brännlund (1995) showed that with convex combinations the optimal choice of stepsize is equivalent to the Camerini-Fratta-Maffioli modification, it also provides an analysis on its linear convergence rate.

From the primal perspective, our method is close to *primal averaging method*. Nedić and Ozdaglar (2009) gives a line of analysis on convergence and quality of the primal approximation by averaging over all previous solutions with a constant stepsize. They use a simple averaging scheme that can be rephrased into a recursive equation with $\alpha_k = 1/k$ such that:

$$\bar{y}_k = \frac{k-1}{k} \cdot \bar{y}_{k-1} + \frac{1}{k} \cdot y_k$$

then it gives lower and upper bounds for the averaged solution that involve the primal violation, norm of the subgradient, etc. Furthermore, they only analyze the case for constant stepsize $s_k = s, s \geq 0$ and the search direction defined solely by the subgradient. We refer to Kiwiel et al. (2007) for target based stepsizes. The volume algorithm proposed by Barahona and Anbil (2000) is close to the case mentioned in Brännlund (1995) in a dual viewpoint while adopting $\tilde{\lambda}_k$ instead of λ_k from the best dual bound $\tilde{\phi}_k = \max_{i=1,\ldots,k} \phi(\lambda_i)$:

$$\lambda_{k+1} = \mathbf{P}(\tilde{\lambda}_k + s_k d_k)$$

Since the solution is strictly feasible by implementation of the recovery algorithm (2.9), i.e., there is no need to bound for feasibility gap as has been done in most of literature covering the **primal recovery**. Instead, we focus on the quality of the recovery, i.e.:

$$|\bar{z}_k - \phi_k| \text{ or } |\bar{z}_k - z^\star|$$

We found its convergence is closely related to strong duality of the problem. Accounting for performance, we suggest several specific choices of parameters regarding the subgradient method (γ, α, d).

2.4 Convergence Analysis

Comments:

1. From Lemma 2.1 and Corollary 2.1.1, we see $\phi^\star = f^\star$.

2. We assume convergence of dual method: $\lambda_k \to \lambda^\star$, we don't limit our choice of direction here (the simplest one should also work).

3. $z_k = z(y_k)$ is not convergent, see the *green* lines in Figure 1, nor $\min_k z(y_k)$.

4. $\bar{y}_k$ is feasible to linear relaxation (may not be integral) and $z(\bar{y})$ is convergent. see the *orange* lines

Now we visit properties for primal solutions.

Theorem 2.2. *Recovery Algorithm* (2.9)

(a) For fixed $y = y_k$, (ϵ_k, δ_k) is the optimal solution for the restricted primal problem.

$$f(\epsilon_k, \delta_k) \leq f(\epsilon, \delta), \quad \forall \delta \geq 0, \epsilon \geq 0, y = y_k$$

(b)

$$\bar{z}_k \geq d_k^\intercal \lambda_k$$

Proof. We first notice a strong duality pair with *fixed* $t \in \Omega_y$, for example, t may take values in $y_k, \bar{y}_k, k = 1, 2, ...$ in the subgradient iterations.

$$\begin{aligned} (\mathbf{P}) \quad & \min_{\delta,\epsilon} p^\intercal \delta + h^\intercal \epsilon \\ \mathbf{s.t.} \quad & t + \delta - \epsilon = 0 \\ & \delta \in \mathbb{R}^n_+, \epsilon \in \mathbb{R}^n_+ \end{aligned} \tag{2.10}$$

and

$$\begin{aligned} (\mathbf{D}) \quad & \max_{\lambda} t^\intercal \lambda \\ \mathbf{s.t.} \quad & -p \leq \lambda \leq h, \lambda \in \mathbb{R}^n \end{aligned} \tag{2.11}$$

Since $\mathbf{P}$ is well-defined. The dual problem $\mathbf{D}$ is straight-forward to solve by comparing t to 0 for each dimension:

$$\mu_j^\star = \begin{cases} h_j & \text{if } t_j > 0 \\ p_j & \text{else} \end{cases} \quad \forall j = 1, ..., n$$

This corresponds to the part (a) and recovery algorithm (2.9) by taking $t = g_k$.

Similarly, take $t = d_k$ we can show part (b).

$$\bar{z}_k = p^\intercal \bar{\delta}_k + h^\intercal \bar{\epsilon}_k \geq d_k^\intercal \lambda_k$$

□

Theorem 2.3. *Suppose the subgradient is bounded, that is, $\exists L > 0$ such that*

$$\|g_k\| \leq L \tag{2.12}$$

and ???

Then the primal-dual bound by the recovery algorithm (2.9) *converges to* 0*, specifically:*

$$\bar{z}_k - \phi^\star \to 0$$

Proof. We first notice

$$\phi^\star - \phi_k \leq g_k^\intercal(\lambda^\star - \lambda_k) \leq \|g_k\|\|\lambda^\star - \lambda_k\| \Rightarrow \phi_k \to \phi^\star$$

This immediately follows:

$$\epsilon_k = d_k^\intercal \lambda_k - \phi_k \leq \frac{1}{2}(2 - \gamma_k)(\phi^\star - \phi_k) \to 0 \tag{2.13a}$$

$$\Rightarrow \quad d_k^\intercal \lambda_k \to \phi^\star \tag{2.13b}$$

We now show the convergence from $\bar{z}$ to $\lambda_k^\intercal d_k$?

As shown in 2.2, by (2.10), (2.11), suppose $\exists\ \mu_k \in [-p, h]$ such that $\mu_k \in \arg\max_\lambda d_k^\intercal \lambda$

It's equivalent to show:

$$\mu_k^\top d_k - \lambda_k^\top d_k \to 0 \tag{2.14}$$

□

2.5 Computational Results

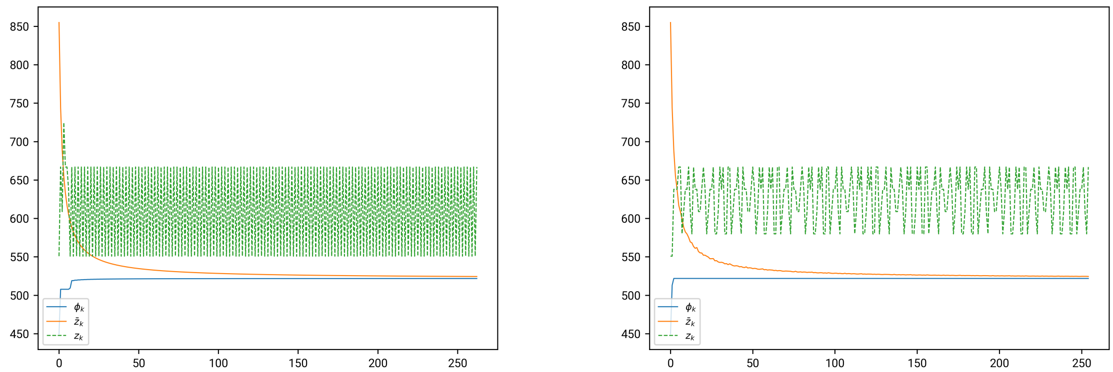
We present our computational results to validate the convergence analysis on subgradient method. The experiments are done on the Fleet Maintenance Problem (see 3.1). The baseline is set by MILP modeled in Gurobi 9.1 to provide lower bound and best integral solution. We implement subgradient methods mentioned in our paper in Python 3.7. Specifically, we test on two specific subgradient variants:

1. Normal subgradient, labelled as normal_sg. This is the simplest subgradient method using iteration:
$$\lambda_{k+1} = \mathbf{P}(\lambda_k + s_k g_k)$$

2. Convex subgradient, convex_sg, using:
$$\lambda_{k+1} = \mathbf{P}(\lambda_k + s_k d_k)$$

where d_k is averaged over past iterations, cf. (2.8)

As shown in Figure 1, our results show that averaged solution by recovery algorithm (2.9) converges to the best lower bound.



(a) Normal subgradient method using g_k

(b) Convex subgradient method using d_k

Figure 1: An instance illustrating the convergence of the subgradient methods and the recovery algorithm (2.9). sg_lb and sb_val are lower bound for the subgradient method and averaged primal value from the recovery algorithm, respectively. primal_k is the primal value at iteration k without averaging. We find that averaged solution avoids the zig-zag behavior of primal_k.

We summarize all test cases in Table 1.

3 Applications

3.1 Flight Maintenance Problem

In the Flight Maintenance Problem (FMP), recurrent maintenance is needed for each airplane to ensure safety.

At each time $t \in T$ there is a demand of quantity $d_t \geq 0$ associated with withdraw cost $b \geq 0$. If the size of the fleet at current time is greater than demand, then it incurs the idle cost $h > 0$. Each airplane $i \in I$ deteriorates with rate α_i and there is a lower bound L on the lifespan representing the current condition. If the airplane approaches to the worst-allowed-condition then it cannot be assigned to any flights. A maintenance plan should be scheduled to improve the current condition for plane i by rate β_i. Once scheduled, a plane comes back after τ time periods.

The goal is to minimize the total cost by uncovered demand and surplus flights. We summarize the notation as follows:

> Notation

- I, T - set of plane, time periods, respectively
- b, h - demand withdraw and plane idle cost, respectively
- τ - lead time for maintenance

We first assume the demand is deterministic.

- d_t - demand, number of planes needed at time t

We make a plan to define work and maintenance schedules.

> Decision

- x_{it} - 0 - 1 variable, 1 if plane i starts a maintenance at time t
- u_{it} - 0 - 1 variable, 1 if plane is working at time t
- $s_{it} \geq 0$ - the lifespan of plane i at time t

The objective can be written in the Newsvendor style:

$$\min_{u,x,s} b \cdot (d_t - \sum_i u_{it})_+ + h \cdot (\sum_i u_{it} - d_t)_+ \tag{3.1}$$

Alternatively, we use the following objective function with δ_t, ϵ_t indicating unsatisfied demand and surplus, respectively.

$$f = \min_{x_{it}, u_{it}, \delta_t, \epsilon_t} \sum_t (b \cdot \delta_t + h \cdot \epsilon_t) \tag{3.2}$$

s.t.

$$\sum_i u_{it} + \delta_t - \epsilon_t = d_t \qquad \forall t \in T \tag{3.3}$$

$$s_{i,t+1} = s_{it} - \alpha_i u_{it} + \beta_i x_{i,t-\tau} \qquad \forall i \in I, t \in T \tag{3.4}$$

$$x_{it} + u_{i,t} \leq 1 \qquad \forall i \in I, t \in T \tag{3.5}$$

$$x_{it} + x_{i\rho} + u_{i,\rho} \leq 1 \qquad \forall i \in I, t \in T, \rho = t+1, ..., t+\tau \tag{3.6}$$

$$s_{it} \geq L \qquad \forall i \in I, t \in T \tag{3.7}$$

The objective function (3.2) and the binding constraint (3.3) follow the same routine for Newsvendor objective, cf. (1.1). The last four sets of constraint describe the non-overlapping requirements during a maintenance for each i. (3.4) tracks the lifespan at each period t, (3.5) describes the utility status of each plane. The non-overlapping requirements for working and maintenance is indicated in (3.6). We summarize (3.4) - (3.7) as Ω_i.

Let $U, X, S \in \mathbb{R}_{+}^{|I|\times|T|}$ be the matrix of u_{it}, x_{it} and s_{it}, $U_{(i,.)}$ be the ith row of U. Let δ, ϵ be the vector of δ_t, ϵ_t, respectively. It allows a more compact formulation.

$$\min_{U,X,S} e^{\mathsf{T}}(b \cdot \delta + h \cdot \epsilon) \tag{3.8}$$

s.t.

$$U^{\mathsf{T}} e + \delta - \epsilon = d \qquad \forall t \in T \tag{3.9}$$

$$X_{(i,\cdot)}, U_{(i,\cdot)}, S_{(i,\cdot)} \in \Omega_i \qquad \forall i \in I \tag{3.10}$$

3.1.1 Dual Optimization

Similar to previous analysis, the Lagrangian is introduced by relaxing the equality constraint (3.9), so we have:

$$\phi(\lambda) = -\sum_t \lambda_t d_t + \min_{\delta_t, \epsilon_t, U} \sum_t [(b + \lambda_t) \cdot \delta_t + (h - \lambda_t) \cdot \epsilon_t] + \sum_i \sum_t \lambda_t u_{it}$$

It reduces to a set of low dimensional minimization problems for each $i \in I$:

$$\begin{aligned} &\phi(\lambda) = -\sum_t \lambda_t d_t + \min_U \sum_i \sum_t \lambda_t u_{it} \\ &\textbf{s.t.} \\ &\quad X_{(i,\cdot)}, U_{(i,\cdot)}, S_{(i,\cdot)} \in \Omega_i \\ &\quad - b \le \lambda_t \le h \end{aligned} \tag{3.11}$$

Lagrange multipliers is updated by a subgradient method (2.2), and the primal solution is computed by the Recovery Algorithm (2.9) using the averaged scheme.

Next we provide analysis on properties of the subproblem.

3.1.2 Subproblem

In the dual search process, one should solve a set of subproblems $\forall i \in I$ with respect to λ defined as follows:

$$\min_{\Omega_i} \sum_t \lambda_t \cdot u_{i,t} \tag{3.12}$$

The model seeks to minimize total cost while keeping the lifespan safely away from the lower bound L. We solve this by dynamic programming.

Define state: $y_t = [m_t, s_t]^\intercal$, where m_t denotes current working status is the remaining lifespan. At each period t we decide whether the plane i is idle, working, or starting a maintenance, i.e.:

$$(u_t, x_t) \in \{(1,0), (0,0), (0,1)\}$$

We have the Bellman equation:

$$V_t(u_t, x_t|y_t) = \lambda_t \cdot u_t + \min_{u,x} V_{t-1}(u_{t-1}, x_{t-1}|y_{t-1}) \tag{3.13}$$

Complexity: let s_0 be the initial lifespan and finite time horizon be $|T|$, we notice the states for remaining maintenance waiting time is finite, $m_t \in \{0, 1, ..., \tau\}$.

Let total number of possible periods to initiate a maintenance be n_1, and working periods be n_2. If we ignore lower bound L on s, total number of possible values of s is bounded above: $|s| \leq (|T| + 1)(\frac{1}{2}|T| + 1)$ since $n_1 + n_2 \leq |T|$. For each subproblem we have at most 3 actions, thus we conclude this problem can be solved by dynamic programming in polynomial time, the complexity is: $O\left(\tau \cdot |T|^3\right)$

3.1.3 Numerical Experiments

In this section, we report numerical results to demonstrate the efficiency and effectiveness of our proposed algorithms for solving the FMP. We parallelize the subproblems to available cores solved by dynamic programming. We summarize all deterministic test cases in Table 1.

References

Barahona F, Anbil R (2000) The volume algorithm: producing primal solutions with a subgradient method. *Mathematical Programming* 87(3):385–399, publisher: Springer.

Brännlund U (1995) A generalized subgradient method with relaxation step. *Mathematical Programming* 71(2):207–219, ISSN 1436-4646, URL http://dx.doi.org/10.1007/BF01585999.

Camerini PM, Fratta L, Maffioli F (1975) On improving relaxation methods by modified gradient techniques. *Nondifferentiable optimization*, 26–34 (Springer).

Kiwiel KC, Larsson T, Lindberg PO (2007) Lagrangian relaxation via ballstep subgradient methods. *Mathematics of Operations Research* 32(3):669–686, publisher: INFORMS.

Nedić A, Ozdaglar A (2009) Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization* 19(4):1757–1780, publisher: SIAM.

Polyak BT (1967) A general method for solving extremal problems. *Soviet Mathematics Doklady* 5.

Appendix

Table 1: Computational Results of the Fleet Maintenance Problem

ID	$\|I\|$	$\|T\|$	bench			normal			convex		
			$\hat{\phi}$	$\bar{z}$	time (s)	time (s)	ϕ_gap	$\bar{z}$_gap	time (s)	ϕ_gap	$\bar{z}$_gap
16	12	120	5512.25	5526.00	243.51	859.71	-0.02%	0.77%			
6	8	160	5161.12	5184.00	14.94	817.31	0.02%	0.35%			
7	8	160	4459.49	4482.00	26.96	1165.90	0.14%	0.84%			
12	4	80	1345.09	1350.00	10.93	151.32	0.26%	0.55%			
2	8	120	3598.68	3618.00	300.01	601.46	0.27%	0.78%			
15	12	120	6654.00	6678.00	18.73	779.93	0.28%	0.56%			
4	12	80	4663.20	4680.00	8.27	433.26	0.30%	0.55%			
5	12	80	4318.83	4338.00	14.05	446.71	0.37%	0.62%			
21	8	80	1872.36	1890.00	300.00	307.95	0.39%	1.08%			
17	12	120	7847.93	7884.00	14.55	712.11	0.43%	0.47%			
1	8	120	4370.33	4392.00	13.77	638.05	0.43%	0.56%			
0	8	120	4871.84	4896.00	7.26	498.11	0.46%	0.50%			
22	8	80	3707.39	3726.00	7.45	138.29	0.48%	0.48%			
8	8	160	4972.18	5004.00	300.02	1356.18	0.56%	0.72%			
19	4	120	1588.91	1602.00	300.01	262.45	0.71%	0.78%			
20	4	120	1369.57	1386.00	300.01	244.54	0.75%	0.94%			
23	8	80	2493.00	2520.00	300.01	224.12	1.02%	0.70%			
3	12	80	3774.44	3834.00	300.01	414.66	1.36%	0.76%			
13	4	80	900.42	918.00	300.01	178.94	1.53%	0.98%			
18	4	120	1093.55	1116.00	300.02	336.67	1.70%	1.40%			
11	4	160	963.51	990.00	300.01	635.89	2.11%	2.24%			
10	4	160	2090.03	2142.00	300.01	654.02	2.43%	0.75%			
14	4	80	1312.70	1350.00	300.01	122.79	2.77%	0.58%			
9	4	160	798.37	846.00	300.02	461.87	2.81%	0.81%			

$\bar{z}$_gap is the relative gap from averaged primal solution to benchmark solution. ϕ_gap is the gap for best lower bound at termination.