Dual optimization for Newsvendor-like problem

Chuwen

January 27, 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:

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

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,

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

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 $f^\star, \phi^\star$ be the optimal objective for primal and dual problem, respectively.

1.1 Affine case

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

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

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 $\lambda \in \Lambda$

1.2 Conditions for strong duality

It's well known that strong duality does not hold in general. We review some of the cases here. The Lagrangian duality theory can be found in any standard text.

Theorem 1.1. *if Ω_y is convex then the strong duality holds ..., i.e. $\phi^\star = f^\star$*

add justifications here (slater, ...)

A more interesting result is devoted to mixed integer problems. 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 1.2. *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,*

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

This immediately allows us to have strong duality by definition of perfect formulation, in which case the linear relaxation solves the original problem.

Corollary 1.2.1. *We conclude the strong duality holds since $Y = \{(y,\delta,\epsilon) : y + \delta - \epsilon = b,\ y \in \mathsf{conv}(\Omega_y)\}$ is already* a perfect formulation *in the sense that $Y = \mathsf{conv}(Y)$*

show this or add more conditions to justify

2 Subgradient method

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

$$\lambda_{k+1} = \mathbf{P}(\lambda_k + s_k d_k) \tag{2.1}$$

where $\mathbf{P}$ is the projection onto dual space Λ. d_k is the update direction for current iteration and s_k is the step size using target-based rule:

$$s_k = \gamma_k \frac{\phi^\star - \phi(\lambda_k)}{||d_k||^2} \tag{2.2}$$

Note the direction d_k computed by

$$d_k = \bar{y}_k - b \tag{2.3}$$

where $\bar{y}_k$ is the convex combination of previous iterations $\{y_i\}_{i=1,\dots k}$ and each y_i solves $\phi_i = \phi(\lambda_i)$:

$$\bar{y}_k = \sum_k^i \alpha_k^i y_i, \quad \sum_k^i \alpha_k^i = 1, \alpha_k^i \geq 0 \tag{2.4}$$

Alternatively, one can express the convexity in a recursive manner:

$$\bar{y}_k = (1 - \alpha_k) \cdot \bar{y}_{k-1} + \alpha_k \cdot y_k \tag{2.5}$$

For we simplicity take $g_k = y_k - b$, then g_k is a subgradient of ϕ at λ_k:

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

The direction can be rewritten as the combination of the subgradient and previous directions:

$$d_k = (1 - \alpha_k) \cdot d_{k-1} + \alpha_k \cdot g_k \tag{2.7}$$

The dual subgradient algorithm can be summarized as follows. $\varepsilon, \varepsilon_s$ are the tolerance parameter for objective gap and stepsize, respectively. $\varepsilon > 0, \varepsilon_s > 0$.

Algorithm 1: The Subgradient Algorithm

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

while $\bar{z}_k - \phi_k \geq \varepsilon$ ***and*** $s_k \geq \varepsilon_s$ **do**

Let current iteration be k

Update the multipliers by

$$\lambda_k = \mathbf{P}(\lambda_{k-1} + s_{k-1} d_{k-1})$$

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

Compute γ_k, α_k properly.

Compute current direction by (2.3) or (2.7)

Update $\epsilon_k, \delta_k, \bar{\epsilon}_k, \bar{\delta}_k, z_k, \bar{z}_k$ by the Recovery Algorithm 2

Stepsize is updated by (2.2)

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

$$\begin{aligned}
\epsilon_k &= \max\{y_k - b, 0\} \\
\delta_k &= \max\{b - y_k, 0\} \\
\bar{\epsilon}_k &= \max\{\bar{y}_k - b, 0\} \\
\bar{\delta}_k &= \max\{b - \bar{y}_k, 0\}
\end{aligned} \tag{2.8}$$

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: $\bar{\epsilon}_k \neq \sum_k^i \alpha_k^i \epsilon_i$. 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.

3 Convergence

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 $\hat{\lambda}_k$ instead of λ_k from the best dual bound $\hat{\phi}_k = \max_{i=1,\dots,k} \phi(\lambda_i)$:

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

Since the solution is strictly feasible by implementation of the recovery algorithm (2.8), 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).

3.1 Analysis outline

- we've showed $\phi^\star = f^\star = z^\star$
- we show λ_k converges to $\lambda^\star \in \Lambda^\star$ for our choices of γ_k, α_k
- we show primal solution $\bar{z}_k$ converges to $z^\star$

Lemma 3.1. *ϵ-subgradient.*

$$\begin{aligned} g_k^\top(\lambda_k - \lambda) &\leq \phi_k - \phi(\lambda) \\ d_k^\top(\lambda_k - \lambda) &\leq \phi_k - \phi(\lambda) + \epsilon_k \end{aligned} \tag{3.1}$$

where

$$\epsilon_k = \sum_k^i \alpha_k^i \cdot [g_i^\mathsf{T}(\lambda_k - \lambda_i) + \phi_i - \phi_k] \tag{3.2}$$

Notice ϵ_k can be further simplified by the definition of ϕ:

$$\epsilon_k = \sum_k^i \alpha_k^i \cdot (g_i^\mathsf{T}\lambda_k - \phi_k) \tag{3.3}$$

Lemma 3.2. *Dual convergence, ?. The subgradient method is convergent if ϵ_k satisfies:*

$$\frac{1}{2}(2-\gamma_k)(\phi_k - \phi^\star) + \epsilon_k \le 0 \tag{3.4}$$

Proof. The proof can be done by showing the monotonic decrease of $\|\lambda_k - \lambda^\star\|$ via the iterative equations.

$$\|\lambda_{k+1} - \lambda^\star\|^2 \le ||\lambda_k - \lambda^\star||^2 + 2\cdot\gamma_k\frac{(\phi^\star - \phi_k)}{\|d_k\|^2}d_k^\mathsf{T}(\lambda_k - \lambda^\star) + (\gamma_k)^2\frac{(\phi^\star - \phi_k)^2}{\|d_k\|^2} \tag{3.5}$$

Notice:

$$\begin{aligned} 2\cdot d_k^\mathsf{T}(\lambda_k - \lambda^\star) + \gamma_k(\phi^\star - \phi_k) \le &\quad 2(\phi_k - \phi^\star + \epsilon_k) + \gamma_k(\phi^\star - \phi_k) \\ =&(2-\gamma_k)(\phi_k - \phi^\star) + 2\epsilon_k \le 0 \end{aligned} \tag{3.6}$$

and we have the convergence. □

Now we visit properties for primal solutions.

Theorem 3.3. *Recovery Algorithm* (2.8)

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

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

(b)

$$\bar{z}_k \ge d_k^\mathsf{T}\lambda_k$$

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.

$$
\begin{aligned}
(\mathbf{P}) \quad & \min_{\delta,\epsilon} p^\intercal \delta + h^\intercal \epsilon \\
\mathbf{s.t.} \quad & t + \delta - \epsilon = 0 \\
& \delta \in \mathbb{R}^n_+, \epsilon \in \mathbb{R}^n_+
\end{aligned}
\tag{3.7}
$$

and

$$
\begin{aligned}
(\mathbf{D}) \quad & \max_{\lambda} t^\intercal \lambda \\
\mathbf{s.t.} \quad & -p \leq \lambda \leq h, \lambda \in \mathbb{R}^n
\end{aligned}
\tag{3.8}
$$

Since **P** is well-defined. The dual problem **D** is straight-forward to solve by comparing t to 0 for each dimension:

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

This corresponds to the part (a) and recovery algorithm (2.8) 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 3.4. *Suppose the subgradient is bounded, that is,* $\exists L > 0$ *such that*

$$
\|g_k\| \leq L \tag{3.9}
$$

and ???

Then the primal-dual bound by the recovery algorithm (2.8) *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:

$$\epsilon_k = d_k^\intercal \lambda_k - \phi_k \leq \frac{1}{2}(2-\gamma_k)(\phi^\star - \phi_k) \to 0 \tag{3.10a}$$

$$\Rightarrow \quad d_k^\intercal \lambda_k \to \phi^\star \tag{3.10b}$$

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

As shown in 3.3, by (3.7), (3.8), suppose $\exists\ \mu_k \in [-p, h]$ such that $\mu_k \in \arg\max_\lambda d_k^\intercal \lambda$

It's equivalent to show:

$$\mu_k^\top d_k - \lambda_k^\top d_k \to 0 \tag{3.11}$$

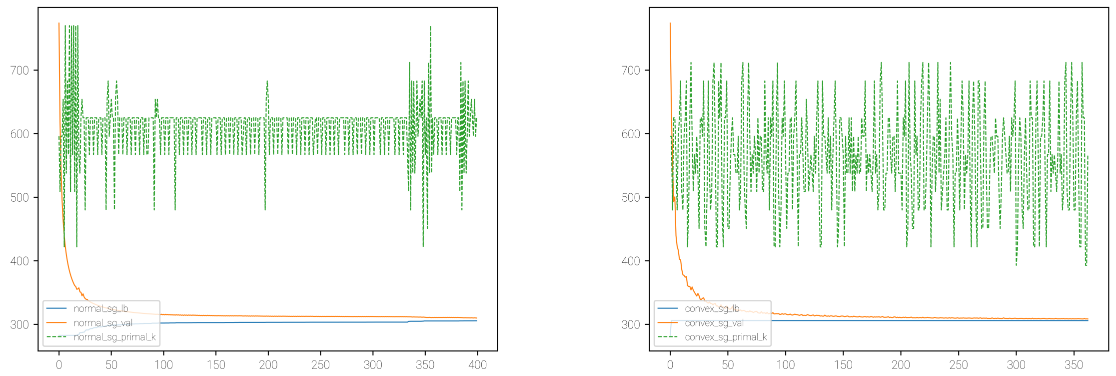
□

3.2 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 **??**). 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. (2.3)

As shown in Figure 1, our results show that averaged solution by recovery algorithm (2.8) 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 (2.8). 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.

4 Applications

4.1 Flight Maintenance Problem

Consider a fleet where 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.

- $\boldsymbol{d}_t$ - 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:

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

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

$$\begin{aligned}
z = &\min_{x_{it}, u_{it}, \delta_t, \epsilon_t} \sum_t (b \cdot \delta_t + h \cdot \epsilon_t) \\
\textbf{s.t.} & \\
& \sum_i u_{it} + \delta_t - \epsilon_t = d_t & \forall t \in T \\
& s_{i,t+1} = s_{it} - \alpha_i u_{it} + \beta_i x_{i,t-\tau} & \forall i \in I, t \in T \\
& x_{it} + u_{i,t} \leq 1 & \forall i \in I, t \in T \\
& x_{it} + x_{i\rho} + u_{i,\rho} \leq 1 & \forall i \in I, t \in T, \rho = t+1, ..., t+\tau \\
& s_{it} \geq L & \forall i \in I, t \in T
\end{aligned}$$

We define the last four sets of constraint as Ω_i, which describe the non-overlapping requirements during a maintenance for each 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.

$$\begin{aligned}
&\min_{U,X,S} e^\top (b \cdot \delta + h \cdot \epsilon) \\
\textbf{s.t.} & \\
& U^\top e + \delta - \epsilon = d & \forall t \in T \\
& X_{(i,\cdot)}, U_{(i,\cdot)}, S_{(i,\cdot)} \in \Omega_i & \forall i \in I
\end{aligned}$$

4.1.1 Lagrangian Relaxation

The Lagrangian is introduced by relaxing the equality constraint, so we have:

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

$\phi(\lambda)(\lambda)$ is unbounded unless $-b \le \lambda_t \le h$, it reduces to a set of low dimensional minimization problems for each i:

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

Next we provide analysis on properties of the subproblem.

4.1.2 Subproblem for each plane

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

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

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

Define state: $y_t = [m_t, s_t]^\top$, where m_t **denotes the remaining time of the undergoing maintenance**. s_t is the remaining lifespan. At each period t we decide whether the plane i is idle or waiting (for the maintenance), working, or starting a maintenance, i.e.:

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

We have the Bellman equation:

$$V_n(u_t, x_t | m_t, s_t) = \lambda_t \cdot u_t + \min_{u,x} V_{n-1}(...)$$

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: $|s| = \sum_i^{|T|} \sum_j^{|T|-i} 1 = (|T|+1)(\frac{1}{2}|T|+1)$ 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: $O\left(\tau \cdot |T|^3\right)$

4.1.3 Subgradient Method

Lagrange multipliers is updated by a subgradient method. (volume algorithm, etc.)

The Volume Algorithm HERE

Notice:

- At iteration k, suppose $-b \leq \lambda_t^k \leq h, \forall t \in T$, we use dynamic programming to solve the relaxed minimization problem, then the (integral) solution (X^k, S^k, U^k) is also feasible for the original problem (compute δ, ϵ accordingly). The primal value z^k is the upper bound for optimal solution $z^\star$: $z^k \geq z^\star$.
- In the volume algorithm, we consider the convex combination $\bar{X}$ of past iterations $\{X^1, ..., X^k\}$. We update $\bar{X} \leftarrow \alpha X^k + (1-\alpha)\bar{X}$. It's easy to verify $\bar{z} \geq z^\star \geq z_{\mathsf{LD}}^k$, where $\bar{z}$ is the primal objective value for $\bar{X}$ and z_{LD}^k is the dual value for X^k. By the termination criterion $|\bar{z} - z_{\mathsf{LD}}^k| \leq \epsilon_z$ for some small value $\epsilon_z > 0$, we conclude the $\bar{z}$ converges to the optimal value $z^\star$.
- While $\bar{z} \to z^\star$, there is no guarantee for the solution $\bar{X}, \bar{U}$ being integral via the volume algorithm; $\bar{X}, \bar{U}$ is feasible only to the linear relaxation.
- Remark:
  - The projection for dual variables is simple since there is only a box constraint. More computation would be needed if we use the minimax objective function, i.e., $q \geq h \cdot (U^\top e - d), q \geq b \cdot (d - U^\top e)$, in which case two set of multipliers are needed, say $\lambda, \mu \geq 0$, and the projection should be done onto:

$$\{(\lambda, \mu) | \lambda + \mu \leq 1\}$$

4.1.4 Rounding

- *compute $\min c^\top |x - x^\star|$ where $x^\star$ is the (possibly) fractional solution achieving the best bound, using DP.

still working on this.

4.1.5 Numerical Experiments

In this section, In this section, we report numerical results to demonstrate the efficiency and effectiveness of our proposed algorithms for solving the repair problem (**ref here**). We parallelize the subproblems to available cores solved by dynamic programming.

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

	$\|I\|$	$\|T\|$	bench			normal			convex		
			$\hat{\phi}$	$\bar{z}$	time (s)	time (s)	ϕ_gap	$\bar{z}$_gap	time (s)	ϕ_gap	$\bar{z}$_gap
68	12	25	1332.00	1332.00	2.49	62.95	-0.00%	0.50%	58.28	0.00%	0.56%
6	12	15	828.00	828.00	0.64	22.65	-0.02%	0.48%	21.30	0.00%	0.57%
63	16	30	2102.93	2106.00	1.25	139.24	-0.02%	0.53%	124.89	0.15%	0.59%
60	16	30	1650.52	1656.00	1.89	137.05	-0.02%	0.67%	124.76	0.33%	0.74%
16	16	25	2196.00	2196.00	1.25	79.32	-0.05%	0.45%	71.44	0.00%	0.50%
44	12	20	756.00	756.00	16.86	44.41	-0.05%	1.29%	41.69	0.00%	0.77%
5	12	15	774.00	774.00	0.53	21.38	-0.06%	0.50%	20.08	0.00%	0.56%
70	20	15	985.02	990.00	103.11	32.87	-0.06%	1.63%	32.68	0.50%	1.47%
76	20	30	2589.23	2592.00	1.71	155.53	-0.07%	0.52%	129.33	0.11%	0.64%
20	16	15	954.00	954.00	1.10	28.71	-0.07%	0.55%	28.58	-0.00%	1.00%
21	16	15	756.00	756.00	1.50	25.99	-0.08%	1.06%	25.99	-0.00%	1.85%
19	16	25	1890.00	1890.00	1.44	83.96	-0.11%	0.47%	70.26	0.00%	0.56%
1	24	25	2445.43	2448.00	2.55	127.85	-0.11%	0.57%	106.56	0.11%	0.70%
51	24	20	2124.00	2124.00	2.36	81.04	-0.11%	0.72%	68.38	0.00%	0.59%
9	12	15	540.00	540.00	1.60	25.58	-0.11%	1.03%	23.93	0.00%	0.87%
11	20	25	2214.00	2214.00	1.39	93.73	-0.15%	0.49%	77.50	0.00%	0.59%
73	20	15	1278.00	1278.00	0.61	34.89	-0.16%	0.52%	29.43	0.00%	0.62%
35	24	30	3258.00	3258.00	2.01	179.31	-0.17%	0.49%	150.41	0.00%	0.59%
59	16	20	1332.00	1332.00	9.22	57.39	-0.19%	0.53%	47.99	0.00%	0.64%

Table 1: (continued)

23	16	15	952.79	954.00	0.72	25.50	-0.20%	1.02%	22.26	0.13%	0.69%
69	12	25	1242.00	1242.00	2.25	74.43	-0.21%	0.55%	67.85	0.00%	0.61%
46	12	30	1422.00	1422.00	1.45	111.73	-0.21%	0.57%	101.88	0.00%	0.64%
36	24	30	2826.00	2826.00	2.89	177.13	-0.23%	0.60%	146.85	0.00%	0.74%
47	12	30	1220.49	1224.00	3.43	93.64	-0.25%	1.13%	94.00	0.20%	2.40%
77	20	30	2340.00	2340.00	1.83	165.61	-0.26%	0.58%	137.36	0.00%	0.70%
56	16	20	1206.00	1206.00	1.12	53.11	-0.29%	0.60%	48.09	0.00%	0.66%
45	12	30	1206.00	1206.00	1.07	98.67	-0.33%	0.66%	89.92	0.00%	0.76%
10	20	25	1638.00	1638.00	1.76	105.16	-0.34%	0.66%	94.73	0.00%	0.72%
40	12	20	1078.94	1080.00	1.51	48.66	-0.44%	1.71%	42.05	0.10%	0.55%
7	12	15	594.00	594.00	0.69	26.63	-0.52%	1.49%	23.49	0.00%	0.74%
33	20	20	1402.16	1404.00	15.10	65.98	0.04%	0.63%	57.33	0.13%	0.72%
31	20	20	1923.43	1926.00	1.49	59.74	0.05%	0.46%	49.63	0.13%	0.56%
4	24	25	2785.33	2790.00	3.12	127.48	0.05%	0.47%	105.67	0.17%	0.57%
3	24	25	2276.87	2286.00	5.05	138.55	0.05%	0.62%	115.09	0.40%	0.77%
25	24	15	1510.24	1512.00	1.35	35.67	0.06%	0.52%	29.54	0.12%	0.60%
30	20	20	1580.59	1584.00	2.58	67.05	0.06%	0.56%	55.52	0.22%	0.70%
39	24	30	2489.16	2502.00	3.44	191.46	0.09%	0.92%	190.77	0.51%	1.36%
28	24	15	1668.00	1674.00	1.56	38.92	0.09%	1.80%	32.62	0.36%	0.61%
0	24	25	2475.97	2484.00	2.02	113.65	0.12%	0.53%	94.05	0.32%	0.66%
66	12	25	968.73	972.00	8.09	78.03	0.14%	1.08%	77.34	0.34%	0.74%
17	16	25	1272.11	1278.00	6.76	98.75	0.15%	1.50%	89.32	0.46%	0.76%
18	16	25	1956.46	1962.00	1.65	89.64	0.18%	0.44%	75.18	0.28%	0.55%
78	20	30	2871.00	2880.00	1.37	145.98	0.18%	0.46%	121.96	0.31%	0.56%
58	16	20	1561.50	1566.00	0.91	54.82	0.20%	0.42%	45.30	0.29%	0.52%
50	24	20	1863.90	1872.00	2.27	78.09	0.20%	0.57%	65.59	0.43%	0.69%
29	24	15	1364.62	1368.00	1.34	33.98	0.22%	0.79%	28.69	0.25%	0.72%
38	24	30	2704.73	2718.00	2.95	190.06	0.23%	0.61%	158.32	0.49%	0.73%
14	20	25	2100.99	2106.00	2.18	98.94	0.23%	0.63%	98.80	0.23%	0.91%
55	16	20	1416.62	1422.00	1.19	54.22	0.24%	0.50%	49.50	0.38%	0.57%
72	20	15	1382.40	1386.00	1.26	30.06	0.25%	0.49%	26.53	0.26%	0.62%
52	24	20	2043.68	2052.00	2.67	71.74	0.25%	0.51%	59.82	0.41%	0.62%
79	20	30	2581.10	2592.00	2.17	175.35	0.25%	0.51%	145.95	0.42%	0.62%
22	16	15	913.85	918.00	1.43	26.61	0.25%	0.59%	22.43	0.45%	0.70%
67	12	25	896.26	900.00	17.62	74.01	0.27%	0.74%	66.90	0.42%	0.82%
41	12	20	861.46	864.00	1.54	40.29	0.28%	0.67%	40.32	0.30%	1.40%
53	24	20	2186.99	2196.00	1.75	74.49	0.29%	0.49%	62.76	0.41%	0.59%
61	16	30	1863.00	1872.00	1.87	151.73	0.30%	0.60%	131.44	0.48%	0.68%
26	24	15	1882.99	1890.00	1.45	35.32	0.31%	0.44%	28.54	0.37%	0.55%
42	12	20	986.91	990.00	1.24	44.64	0.31%	0.54%	39.41	0.31%	0.61%
27	24	15	1307.42	1314.00	1.09	38.37	0.33%	0.59%	32.33	0.50%	0.78%
8	12	15	662.15	666.00	1.72	23.08	0.35%	0.58%	21.10	0.58%	0.66%
12	20	25	2058.75	2070.00	300.02	109.29	0.37%	0.57%	91.62	0.55%	0.68%
24	16	15	1164.18	1170.00	0.53	26.36	0.39%	0.45%	22.16	0.50%	0.55%
54	24	20	2220.91	2232.00	2.16	67.87	0.39%	0.46%	56.28	0.50%	0.56%
34	20	20	1880.52	1890.00	1.80	62.09	0.39%	0.47%	51.36	0.50%	0.59%
2	24	25	2292.19	2304.00	26.96	132.40	0.39%	0.76%	131.94	0.51%	0.63%
64	16	30	2292.23	2304.00	1.14	140.09	0.40%	0.46%	116.67	0.51%	0.55%
15	16	25	1773.58	1782.00	1.65	99.49	0.40%	0.53%	83.24	0.47%	0.65%
74	20	15	1271.49	1278.00	2.52	30.54	0.40%	0.84%	25.28	0.51%	0.62%
37	24	30	2883.32	2898.00	2.88	224.79	0.42%	0.59%	220.62	0.51%	1.26%
32	20	20	1809.64	1818.00	1.06	64.22	0.45%	0.49%	53.86	0.46%	0.61%
75	20	30	2310.74	2322.00	2.69	180.16	0.45%	0.55%	181.30	0.48%	1.96%
13	20	25	2310.22	2322.00	2.46	104.55	0.46%	0.46%	87.16	0.51%	0.55%
48	12	30	1665.18	1674.00	2.51	101.36	0.47%	0.49%	101.66	0.53%	0.86%
43	12	20	1235.38	1242.00	2.84	33.27	0.49%	0.46%	34.40	0.54%	0.50%
57	16	20	1468.37	1476.00	1.21	51.02	0.51%	0.55%	51.03	0.51%	1.04%
49	12	30	1252.32	1260.00	300.02	114.56	0.58%	0.86%	114.81	0.61%	1.73%

Table 1: (continued)

62	16	30	1495.06	1512.00	300.02	140.53	0.62%	1.09%	139.62	1.05%	1.87%
71	20	15	1031.76	1044.00	300.01	32.00	1.15%	1.20%	32.43	0.92%	3.13%
65	12	25	1133.18	1152.00	300.03	79.02	1.59%	0.93%	69.02	1.66%	0.63%

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