Contents

1 The repair model

1.1 Formulation

> Notation

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

The demand is stochastic with some distribution $f \in \mathcal{F}$

- $\boldsymbol{d}_t$ - demand/number of planes needed at time t

> 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 DRO/SP model, the goal is to minimize unsatisfied demand and surplus (idle) flights. A minimax objective function that is widely used in Newsvendor problems can be written as follows:

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

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

$$
\begin{aligned}
z = & \min_{x_{it}, u_{it}, \delta_t^+, \delta_t^-} \sum_t (b \cdot \delta_t^+ + h \cdot \delta_t^-) \\
\textbf{s.t.} & \\
& \sum_i u_{it} + \delta_t^+ - \delta_t^- = d_t & \forall t \in T \\
& s_{i,t+1} = s_{it} - \alpha_i u_{it} + \beta_i x_{i,t-\tau} & \forall i \in I, t \in T \\
& x_{it} + u_{i,t} \leq 1 & \forall i \in I, t \in T \\
& x_{it} + x_{i\rho} + u_{i,\rho} \leq 1 & \forall i \in I, t \in T, \rho = t+1, ..., t+\tau \\
& s_{it} \geq L & \forall i \in I, t \in T
\end{aligned} \tag{1.1}
$$

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^+, δ^- be the vector of δ_t^+, δ_t^-, respectively. It allows a more compact formulation.

$$
\begin{aligned}
& \min_{U,X,S} e^\top (b \cdot \delta^+ + h \cdot \delta^-) \\
\textbf{s.t.} & \\
& U^\top e + \delta^+ - \delta^- = d & \forall t \in T \\
& X_{(i,\cdot)}, U_{(i,\cdot)}, S_{(i,\cdot)} \in \Omega_i & \forall i \in I
\end{aligned} \tag{1.2}
$$

We propose a polynomial-time approximation to this problem by Lagrangian relaxation and a subgradient method. At each iteration of the dual search procedure, a set of sub-problems are solved by dynamic programming. The convergence of any black-box subgradient method can be found in (Polyak (1967), and books by Nesterov, Bertsimas, ... to be added), and the complexity are verified on the level of $O(\frac{1}{\epsilon^2})$. We refer further analysis on the rate and convergence of such class of algorithm to Nesterov (2009), Nedić and Ozdaglar (2009). Since regular sugradient method does not grant primal feasibility, there are methods... We use the volume algorithm described in Barahona and Anbil (2000) to update the dual multipliers while approximating a primal feasible solutions to the linear relaxation. The volume algorithm applied to our problem has further properties. Besides the lower bound acquired in the dual relaxation, the convex combination of past iterations in the algorithm gives an upper bound to the original problem. This explicitly bounds the optimal value. Although the solution terminated at the subgradient method is not guaranteed to be integral, it gives a tight interval that asymptotically approaches to the optimal value.

If we allow a tolerance $\epsilon \geq 0$ on subgradient method, the worst overall complexity is $O\left(\frac{1}{\epsilon^2} \cdot \tau \cdot |I| \cdot |T|^3\right)$.

1.2 Lagrangian Relaxation

```
todo
- can do better on complexity
```

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

$$z_{\mathsf{LD}} = -\sum_t \lambda_t d_t + \min_{\delta_t^+,\delta_t^-,U} \sum_t \left[(b+\lambda_t)\cdot\delta_t^+ + (h-\lambda_t)\cdot\delta_t^-\right] + \sum_i \sum_t \lambda_t u_{it}$$

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

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

Next we provide analysis on properties of the subproblem.

1.2.1 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 sol‘ve 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: $O\left(\tau \cdot |T|^3\right)$

1.2.2 Subgradient Method

Lagrange multipliers is updated by the subgradient method, · The parameter α to produce convex combinations of history solutions is updated by the procedure in ... to approach better precision.

Algorithm 1: The Volume Algorithm

initialize multipliers $\bar\lambda = e$, stepsize s^0, solve $z_{\mathsf{LD}}(\bar\lambda)$ to obtain primal solution $(\bar X, \bar U, \bar S)$;
while $||\nabla_\lambda^k|| \leq \epsilon_\nabla$ *and* z **do**
  compute subgradient $\nabla_\lambda^k = \bar U^\top e - d$, let $\lambda^k = \bar\lambda + s^k \cdot \nabla_\lambda^k$, where stepsize s^k is computed from ..., ;
  solve $z_{\mathsf{LD}}^k = z_{\mathsf{LD}}(\lambda^k)$ to obtain primal solution (X^k, U^k, S^k);
  update primal solution $\bar U, \bar X, \bar S$ by the same routine, $\bar X$ for example.

$$\bar X = (1 - \alpha)\bar X + \alpha X^k$$

  if $z_{\mathsf{LD}}^k > \bar z_{\mathsf{LD}}$ **then**
    update: $\bar\lambda \leftarrow \lambda^k$, $\bar z \leftarrow z^k$
  end
  recover solution to primal problem by 1.2:
  set iteration number $k \leftarrow k + 1$
end

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^+, δ^- 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:

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

1.2.3 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.

1.2.4 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.

(details on the algorithm, parameters, et cetera.)

Convergence of Lagrange Relaxation We randomly generated 5-8 instances for each problem class with size $|I| = 10, 15, 20$ and $|T| = 25, 30$. We use Gurobi 9.1 to compute benchmarks: lower bound bench_lb and primal objective value bench_sol within 300 seconds. The value and bound for subgradient methods are subgrad_val, subgrad_lb, respectively. We set the maximum iterations to 400 so that the subgradient method terminates at a comparable time with Gurobi. At last, we compare primal_gap and bound_gap in the last two columns. All the computations have been performed on a Mac mini (2018) with 3.2 GHz 6-Core Intel Core i7 processor and a RAM of 32 GB.

It can be observed that the subgradient method performed closely to commercial mixed-integer linear solver.

Table 1: Computational results on convergence to optimal solution $z^\star$

$\|I\|$	$\|T\|$	bench_lb	bench_sol	subgrad_val	subgrad_lb	primal_gap	bound_gap
15	25	50.545103	52.000000	51.567854	50.593575	-0.83%	0.10%
20	25	58.583531	76.000000	70.658525	69.975157	-7.03%	19.45%
20	25	144.000000	144.000000	144.715790	143.375955	0.50%	-0.43%
10	30	50.160760	52.000000	52.366787	52.000000	0.71%	3.67%
10	30	50.000000	50.000000	50.394692	50.000000	0.79%	0.00%
15	25	88.000000	88.000000	88.725505	88.000000	0.82%	0.00%
10	25	49.999998	49.999998	50.420987	50.000000	0.84%	0.00%
10	25	48.968605	50.000000	50.422554	50.000000	0.85%	2.11%
10	30	81.999998	82.000000	82.718656	81.938236	0.88%	-0.08%
20	25	52.198486	53.999995	54.480932	53.974198	0.89%	3.40%
20	25	146.000000	146.000000	147.319376	145.898549	0.90%	-0.07%
20	25	118.000000	118.000000	119.063879	118.000000	0.90%	0.00%
15	25	88.000000	88.000000	88.788297	88.000000	0.90%	0.00%
10	30	28.507084	30.000000	30.268855	30.000000	0.90%	5.24%
20	25	136.236668	138.000000	139.251805	137.915722	0.91%	1.23%
10	30	86.655991	88.000000	88.818326	88.000000	0.93%	1.55%
10	30	60.000000	60.000000	60.565028	60.000000	0.94%	0.00%
10	25	72.000000	72.000000	72.698958	72.000000	0.97%	0.00%
15	25	84.000000	84.000000	84.837668	84.000000	1.00%	0.00%
10	30	44.775258	50.000000	50.583272	49.454663	1.17%	10.45%
10	25	40.000000	40.000000	40.766048	39.747977	1.92%	-0.63%
10	30	33.175978	35.999999	36.747449	34.966511	2.08%	5.40%
10	25	9.386239	14.000000	14.342575	14.000000	2.45%	49.15%
15	25	34.000000	34.000000	35.802835	34.000000	5.30%	0.00%

Reference

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

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

Nesterov Y (2009) Primal-dual subgradient methods for convex problems. *Mathematical programming* 120(1):221–259.

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