Appendices

EC.1. Algorithms

Algorithm 4 Greedy Algorithm to Find the Optimal Solution to (15)

Input: A separable convex function $w(x) = \sum_{r=1}^{R} \breve{\Gamma}_r(x_r)$, a finite jump system J and a feasible solution $x^0 \in J$.

Output: The optimal solution x^* to (15).

Step 1: $x \leftarrow x^0$.

Step 2: If neither of the two conditions below is satisfied, then stop. Otherwise, go to step 3.

(1) There exists a step s such that $x + s \in J$ and $w(x+s) < w(x)$.

(2) There exists steps s, s' such that $x + s \notin J^\tau, x + s + s' \in J^\tau$ and $w(x+s+s') < w(x)$.

Step 3: Compute

$$w_1 \leftarrow \min\{w(x+s) \,|\, x+s \in J, w(x+s) < w(x)\} \quad \text{(EC.1)}$$

$$w_2 \leftarrow \min\{w(x+s) \,|\, x+s \notin J, x+s+s' \in J, w(x+s+s') < w(x)\} \quad \text{(EC.2)}$$

where the minimum over the empty set is defined to be $+\infty$.

Step 4: $\hat{w} \leftarrow \min\{w_1, w_2\}$. If $\hat{w} = w_1$, let $\hat{s}$ be the step s that attains the minimum of (EC.1), $x \leftarrow x + \hat{s}$ and go to step 2. If $\hat{w} = w_2$, let $\hat{s}$ and $\hat{s}'$ be the steps s and s' that attains the minimum of (EC.2), $x \leftarrow x + \hat{s} + \hat{s}'$ and go to step 2.

EC.2. Proofs

Proof of Proposition 1

Let $g(x_r; t_r, p_r) = \log\left(\sum_{m=0}^{x_r - t_r} \binom{x_r}{m} p_r^m (1-p_r)^{x_r - m}\right)$, where $t_r = Q_r^{-1}(\tau) \ge 0$. It is obvious that g is a piecewise-linear function, so it suffices to show that for any integer $x_r \ge t_r + 1$, $2g(x_r; t_r, p_r) \ge g(x_r + 1; t_r, p_r) + g(x_r - 1; t_r, p_r)$, or equivalently, $\left(e^{g(x_r;t_r,p_r)}\right)^2 \ge e^{g(x_r+1;t_r,p_r)} \cdot e^{g(x_r-1;t_r,p_r)}$. To avoid ambiguity, let $\binom{n}{m} = 0$ if $m > n$ or $m < 0$. Then, we have

$$\begin{aligned}\left(e^{g(x_r;t_r,p_r)}\right)^2 &= \left(\sum_{m=0}^{x_r - t_r} \binom{x_r}{m} p_r^m (1-p_r)^{x_r - m}\right)^2 \\ &= \sum_{k=0}^{2(x_r - t_r)} \sum_{m=0}^{\min\{k, x_r - t_r\}} \binom{x_r}{m}\binom{x_r}{k-m} p_r^k (1-p_r)^{2x_r - k}.\end{aligned}$$

Similarly,

$$\begin{aligned}e^{g(x_r+1;t_r,p_r)} \cdot e^{g(x_r-1;t_r,p_r)} &= \left(\sum_{m=0}^{x_r - t_r + 1} \binom{x_r+1}{m} p_r^m (1-p_r)^{x_r + 1 - m}\right)\left(\sum_{m=0}^{x_r - t_r - 1} \binom{x_r-1}{m} p_r^m (1-p_r)^{x_r - 1 - m}\right) \\ &= \sum_{k=0}^{2(x_r - t_r)} \sum_{m=0}^{\min\{k, x_r - t_r - 1\}} \binom{x_r-1}{m}\binom{x_r+1}{k-m} p_r^k (1-p_r)^{2x_r - k}.\end{aligned}$$

Thus, it is sufficient to show that, for any integer $k \in [0, 2(x_r - t_r)]$,

$$\sum_{m=0}^{\min\{k, x_r - t_r\}} \binom{x_r}{m}\binom{x_r}{k-m} \ge \sum_{m=0}^{\min\{k, x_r - t_r - 1\}} \binom{x_r-1}{m}\binom{x_r+1}{k-m}.$$

First, if $k \in [0, x_r - t_r - 1]$, then

$$\begin{aligned}&\sum_{m=0}^{\min\{k, x_r - t_r\}} \binom{x_r}{m}\binom{x_r}{k-m} - \sum_{m=0}^{\min\{k, x_r - t_r - 1\}} \binom{x_r-1}{m}\binom{x_r+1}{k-m} \\ &= \sum_{m=0}^{k} \binom{x_r}{m}\binom{x_r}{k-m} - \sum_{m=0}^{k} \binom{x_r-1}{m}\binom{x_r+1}{k-m} \\ &= \binom{2x_r}{k} - \binom{2x_r}{k} \\ &= 0.\end{aligned}$$

Second, if $k \in [x_r - t_r, 2(x_r - t_r)]$, then

$$\sum_{m=0}^{\min\{k, x_r - t_r\}} \binom{x_r}{m}\binom{x_r}{k-m} - \sum_{m=0}^{\min\{k, x_r - t_r - 1\}} \binom{x_r-1}{m}\binom{x_r+1}{k-m}$$

$$
\begin{aligned}
&= \sum_{m=0}^{x_r-t_r} \binom{x_r}{m}\binom{x_r}{k-m} - \sum_{m=0}^{x_r-t_r-1} \binom{x_r-1}{m}\binom{x_r+1}{k-m} \\
&= \sum_{m=0}^{x_r-t_r} \left[\binom{x_r-1}{m} + \binom{x_r-1}{m-1}\right]\binom{x_r}{k-m} - \sum_{m=0}^{x_r-t_r-1} \binom{x_r-1}{m}\left[\binom{x_r}{k-m} + \binom{x_r}{k-m-1}\right] \\
&= \binom{x_r-1}{x_r-t_r}\binom{x_r}{k-x_r+t_r} + \sum_{m=0}^{x_r-t_r} \binom{x_r-1}{m-1}\binom{x_r}{k-m} - \sum_{m=0}^{x_r-t_r-1} \binom{x_r-1}{m}\binom{x_r}{k-m-1} \\
&= \binom{x_r-1}{x_r-t_r}\binom{x_r}{k-x_r+t_r} + \sum_{m=0}^{x_r-t_r-1} \binom{x_r-1}{m}\binom{x_r}{k-m-1} - \sum_{m=0}^{x_r-t_r-1} \binom{x_r-1}{m}\binom{x_r}{k-m-1} \\
&= \binom{x_r-1}{x_r-t_r}\binom{x_r}{k-x_r+t_r} \\
&\geq 0.
\end{aligned}
$$

Therefore, from all of the above, we can see that the desired inequality holds.

Proof of Proposition 2

We let $x, y \in J^\tau$ be two feasible solutions to (8). A randomly selected step s from x to y is $\{0, 0, ..., 1, ..., 0\}$ (or $\{0, 0, ..., -1, ..., 0\}$ which is omitted as the following proof can be established similarly). Since $\sum_{r=1}^{R} x_r + s_r > K$, then $x + s \notin J^\tau$ which violates condition (1) in Definition 2. For condition (2), there is always a step s' from $x + s$ to y such as $\{0, ..., -1, ..., 0\}$ satisfying $\sum_{r=1}^{R} x_r + s_r + s'_r = K$, that is $x + s + s' \in J^\tau$. To sum up, condition (2) can be always satisfied and therefore J^τ is a jump system.

Proof of Proposition 3

Let t^* denote the optimal solution to problem (1). In Step 2, solving (8) for any τ_k such that $\tau_k < t^*$ would result in $\psi^* < \log(1-\alpha)$. Conversely, solving (8) for any τ_k such that $\tau_k \geq t^*$ would result in $\psi^* \geq \log(1-\alpha)$. These facts guide the binary search in Step 3, which either increases τ_k if the corresponding $\psi^* < \log(1-\alpha)$ or decreases τ_k if the resulting $\psi^* \geq \log(1-\alpha)$. The former is done by increasing the lower bound of the search interval, t_l, whereas the latter is done by reducing the upper bound of the search interval, t_u. Throughout the procedure, the boundaries of the search interval, $[t_l, t_u]$, are computed such that solving (8) for $\tau_k = t_l$ would result in $\psi^* < \log(1-\alpha)$, whereas solving (8) for $\tau_k = t_u$ would result in $\psi^* \geq \log(1-\alpha)$. In the last iteration, $t_u - t_l \leq \epsilon$, implying that the returned τ_k is at most ϵ apart from the optimal t^*.

The time complexity of the binary search algorithm is $\mathcal{O}(\log(\mathcal{W}/\epsilon))$ (Sabharwal 2019) in which $\mathcal{W}$ is the width of the initial interval. In each iteration, Algorithm 2 solves (8) in pseudo-polynomial time (Shioura and Tanaka 2007). Thus, the proposed binary search procedure (Algorithm 1) returns the ϵ-efficient solution to (7) in pseudo-polynomial time.

Proof of Proposition 4

The proof is straightforward. First, recall that the relaxation at each node is solved optimally with Algorithm 4, and that we prune the nodes with LBs that are higher than the best $\hat{\text{UB}}$ found thus far. Second, recall that after solving the relaxation on a node and obtaining solution $\bar{x}$, we will branch further if there exists r such that $\breve{\Gamma}_r^{[a_r,b_r]}(\bar{x}_r) \neq \Gamma_r(\bar{x}_r)$ (i.e., if LB $<$ UB). Thus, after termination of the B&B procedure, the solution with the lowest UB must be optimal.

EC.3. Node Routing Formulation

Here, we give an example of a node routing problem whose optimal value is the shortest turnaround time achievable by a fixed number of vehicles. The problem is defined on a directed graph $G = (V, A)$, where V is the node set consisting of the customer node set V_c and the depot node 0, and A represents the arc set. The set of homogeneous trucks is denoted by K, where each truck $k \in K$ has a salt load capacity Q. The salt consumption at each customer node $i \in V_c$ is denoted by d_i, and the time required to traverse arc $(i, j) \in A$ is denoted by t_{ij}. A binary variable x_{ij}^k takes a value of 1 if truck k travels from node $i \in V$ to node $j \in V$. A positive variable s_i^k indicates the arrival time of truck k at node i, which is constrained within the interval $[a_i, b_i]$. The problem is formulated as follows:

$$\textbf{min-max VRP:} \quad \min_{x_{ij}^k \in \{0,1\}, s_i^k \geq 0, z} \quad z \tag{EC.3}$$

$$\text{s.t.} \sum_{i \in V_c} x_{0j}^k = 1, \quad \forall k \in K \tag{EC.4}$$

$$\sum_{i \in V_c} x_{i0}^k = 1, \quad \forall k \in K \tag{EC.5}$$

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \quad \forall i \in V_c \tag{EC.6}$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{ji}^k = 0, \quad \forall i \in V_c, \forall k \in K \tag{EC.7}$$

$$\sum_{i \in V_c} d_i \sum_{j \in V} x_{ij}^k \leq Q, \quad \forall k \in K \tag{EC.8}$$

$$s_i^k + t_{ij} - M_{ij}(1 - x_{ij}^k) \leq s_j^k, \quad \forall (i,j) \in A, \forall k \in K \tag{EC.9}$$

$$a_i \leq s_i^k \leq b_i, \quad \forall i \in V_c, \forall k \in K \tag{EC.10}$$

$$z \geq \sum_{i \in V} \sum_{j \in V} t_{ij} x_{ij}^k, \quad \forall k \in K. \tag{EC.11}$$

The objective function (EC.3) computes the turnaround time. Constraints (EC.4) and (EC.5) ensure that all trucks depart from and return to the depot node 0. Constraints (EC.6) guarantee that each customer node $i \in V_c$ is served exactly once, and constraints (EC.7) ensure flow conservation for each truck. Constraints (EC.8) ensure that the salt usage remains within the capacity of the spreader. Constraints (EC.9) require the arrival time s_i^k of truck k at node i to occur before the arrival time s_j^k of truck k at the following node j; the large constants M_{ij} can be decreased to $\max_{(i,j)\in A}\{b_i + t_{ij} - a_j\}$. Lastly, constraints (EC.10) represent time windows and help to eliminate subtours.

Most of the regions are small enough that the VRP can be solved to optimality using mathematical programming software. However, there are a few regions with more complex networks: for example, in the largest region (Centerville), the VRP has 433 nodes, which is quite large by the standards of the literature. In these cases, we used a combination of routing heuristics including cheapest insertion, crossover, and exchange procedures, repeating them until no further improvement could be made. This approach returns a solution in under 30 minutes. To evaluate the performance of the solution, one can run branch-and-bound on the VRP and use the best lower bound (LB) returned by the solver. For example, if we consider Centerville with six trucks, the heuristic approach produces a solution whose turnaround time is 140 minutes, while leaving branch-and-bound to run overnight produces an LB of 128.56 minutes. The optimality gap is thus no more than 9%, and likely smaller, as LBs are notoriously slow to converge in this problem class.

EC.4. Routes for the Truck Allocation Based on Problem (1)

Bothwell

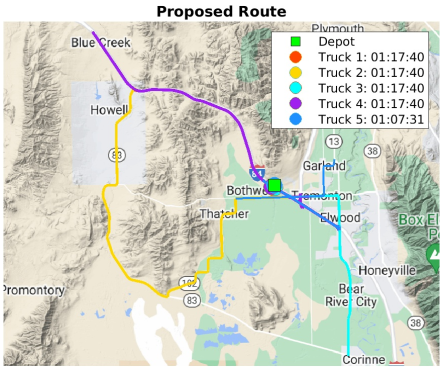

Table EC.1: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	77.72	46.99	18.57	12.93	4.26
# 2	77.42	46.67	29.06	17.85	3.60
# 3	76.27	37.33	15.32	6.61	3.84
# 4	77.40	54.74	51.17	36.62	2.27
# 5	67.52	35.68	30.46	17.86	2.23
Total	376.33	221.41	144.58	91.87	16.19
Max	77.72	54.74	51.17	36.62	4.26

Route animation: https://youtu.be/dnLowWtO-Tc

Brigham City

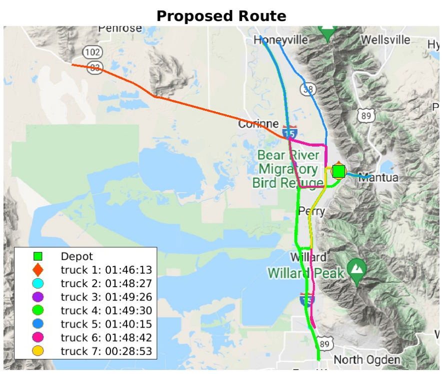

Table EC.2: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	106.22	61.20	12.98	5.67	6.94
# 2	108.45	63.95	54.23	32.49	3.93
# 3	109.44	69.64	27.26	14.74	6.86
# 4	109.50	68.07	36.50	24.60	5.43
# 5	100.25	52.42	40.08	23.37	3.63
# 6	108.70	57.63	20.10	10.76	5.86
# 7	28.88	14.24	2.24	1.02	1.65
Total	671.44	387.15	193.39	112.65	34.31
Max	109.44	69.64	54.23	32.49	6.94

* Route animation: https://youtu.be/p77IbAbIEY8

Centerville

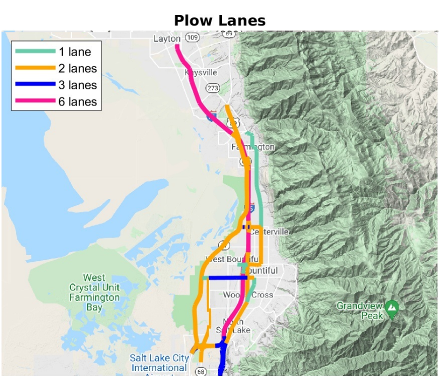

Table EC.3: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	104.89	55.14	6.65	3.18	6.49
# 2	104.94	55.24	18.95	10.21	5.62
# 3	104.82	55.19	13.35	7.75	5.93
# 4	104.52	60.90	23.93	15.24	5.70
# 5	104.93	51.35	31.90	15.13	4.52
# 6	104.66	48.72	20.76	10.08	4.83
# 7	104.71	43.85	17.57	8.05	4.47
# 8	101.00	47.37	36.88	18.42	3.61
Total	834.47	417.76	169.99	88.06	41.21
Max	104.94	60.90	36.88	18.42	6.49

Clearfield

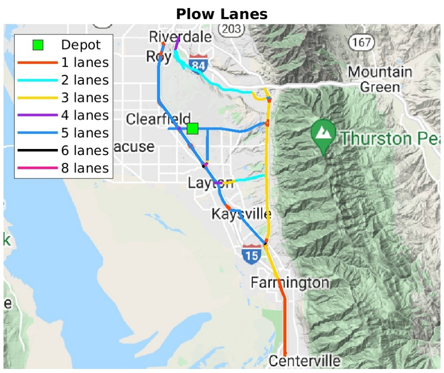

Table EC.4: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	104.46	54.87	38.75	19.42	4.43
# 2	102.41	42.21	29.94	14.42	3.47
# 3	104.54	51.93	47.02	23.42	3.56
# 4	104.24	52.08	29.48	13.94	4.76
# 5	102.22	52.22	49.83	25.00	3.40
# 6	104.54	60.82	96.09	52.48	1.04
# 7	77.36	38.86	40.58	20.46	2.30
Total	699.77	352.99	331.69	169.14	22.98
Max	104.54	60.82	96.09	52.48	4.76

Huntsville

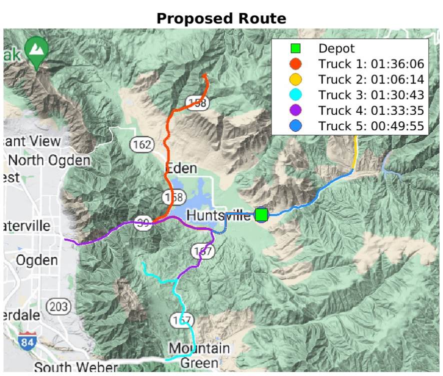

Table EC.5: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	96.10	40.04	20.40	8.50	3.94
# 2	66.23	27.60	13.43	5.60	2.75
# 3	90.72	37.80	32.87	13.70	3.01
# 4	93.59	39.00	31.92	13.30	3.21
# 5	49.92	20.79	24.95	10.39	1.30
Total	396.56	165.23	123.57	51.49	14.21
Max	96.10	40.04	32.87	13.70	3.94

Route animation: https://youtu.be/9KRST3-Iwmw

Morgan

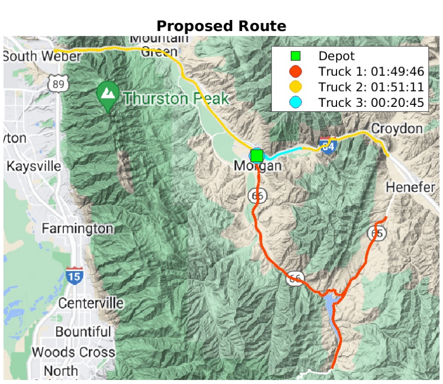

Table EC.6: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	109.76	49.28	1.26	0.42	6.11
# 2	111.19	62.30	93.39	56.33	0.75
# 3	20.75	9.07	11.55	5.99	0.39
Total	241.7	120.65	106.2	62.74	7.24
Max	111.19	62.3	93.39	56.33	6.11

Route animation: https://youtu.be/3KS7Js1C3QI

North Logan

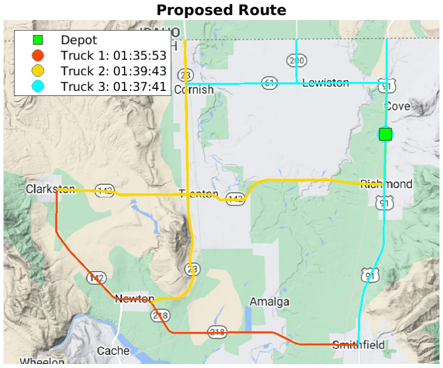

Table EC.7: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	95.88	51.14	11.21	5.98	5.65
# 2	99.72	53.18	3.45	1.84	6.42
# 3	97.69	52.10	3.45	1.84	6.28
Total	293.29	156.42	18.11	9.66	18.34
Max	99.72	53.18	11.21	5.98	6.42

Route animation: https://youtu.be/k27Skj1_qn8

Ogden

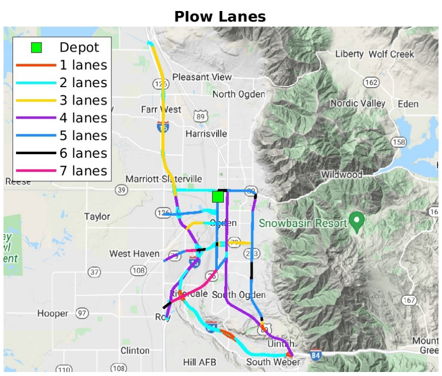

Table EC.8: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	105.37	50.83	45.59	21.76	3.63
# 2	105.98	59.89	24.33	12.16	5.97
# 3	105.85	54.60	42.77	21.53	4.13
# 4	105.47	38.30	4.09	1.62	4.59
# 5	105.46	60.99	44.57	26.12	4.36
# 6	105.56	55.25	19.86	8.53	5.84
# 7	105.71	43.67	24.20	10.00	4.21
# 8	105.82	38.44	19.62	6.94	3.94
# 9	72.85	27.87	7.46	2.94	3.12
Total	918.07	429.84	232.48	111.60	39.78
Max	105.98	60.99	45.59	26.12	5.96

Riverside

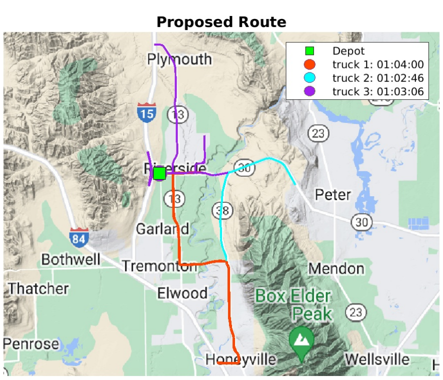

Table EC.9: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	64.00	33.85	1.36	0.74	4.13
# 2	62.76	35.76	7.86	4.34	3.92
# 3	63.10	33.16	14.54	8.20	3.12
Total	189.86	102.77	23.76	13.28	11.19
Max	64.00	35.76	14.54	8.20	4.13

Route animation: https://youtu.be/cF0TCiryk54

South Logan

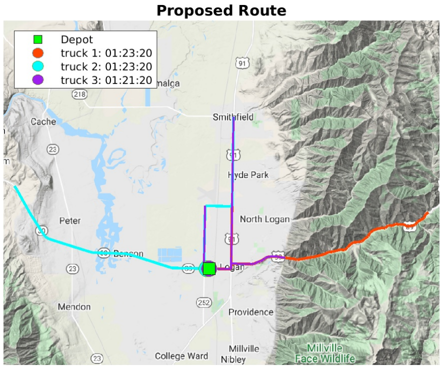

Table EC.10: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
#1	83.82	43.81	0.00	0.00	5.47
#2	84.01	43.18	1.25	0.52	5.33
#3	81.33	33.88	8.28	3.45	3.80
Total	249.16	120.87	9.53	3.97	14.61
Max	84.10	43.81	8.28	3.45	5.47

Route animation: https://youtu.be/A-_WOT530pQ

Wellsville

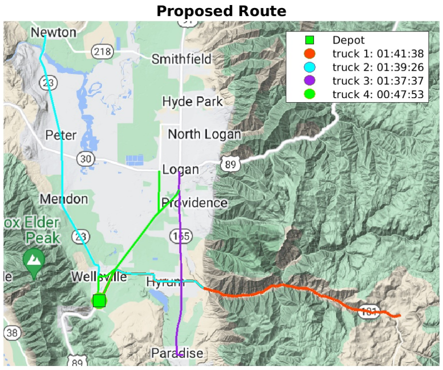

Table EC.11: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
#1	101.64	45.48	12.89	6.94	4.82
#2	99.44	53.18	14.41	7.14	5.76
#3	97.61	56.66	3.36	2.24	6.80
#4	47.89	28.52	0.00	0.00	3.56
Total	346.58	183.84	30.66	16.32	20.94
Max	101.64	56.67	14.41	7.14	6.80

* Route animation: https://youtu.be/kA5m4n7p0Gc

Laketown

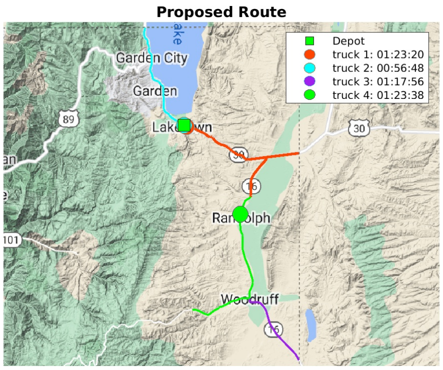

Table EC.12: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
#1	84.00	42.00	0.00	0.00	5.25
#2	56.80	28.40	0.00	0.00	3.55
#3	77.93	41.94	0.00	0.00	5.24
#4	83.64	41.82	42.28	21.14	2.58
Total	302.37	154.16	42.28	21.14	16.62
Max	84.00	42.00	42.28	21.14	5.25

* Route animation: https://youtu.be/uMUMBMNgyhY

EC.5. Routes for the Truck Allocation Based on Problem (2)

Bothwell

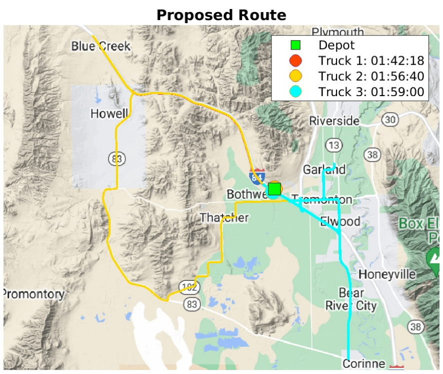

Table EC.13: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	102.30	62.72	31.61	22.31	5.05
# 2	116.67	68.75	53.07	32.44	4.54
# 3	116.22	64.51	18.78	11.69	6.60
Total	335.19	195.98	103.46	66.44	16.19
Max	116.67	68.75	53.07	32.44	6.60

Route animation: https://youtu.be/gWYHBSU5i9I

Brigham City

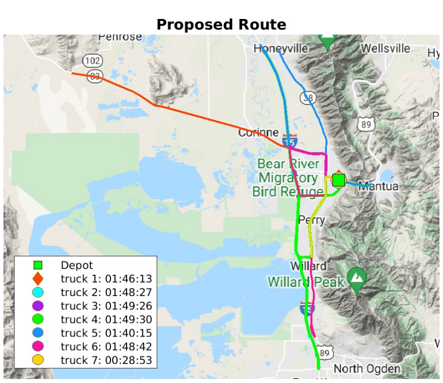

Table EC.14: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	106.22	61.20	12.98	5.67	6.94
# 2	108.45	63.95	54.23	32.49	3.93
# 3	109.44	69.64	27.26	14.74	6.86
# 4	109.50	68.07	36.50	24.60	5.43
# 5	100.25	52.42	40.08	23.37	3.63
# 6	108.70	57.63	20.10	10.76	5.86
# 7	28.88	14.24	2.24	1.02	1.65
Total	671.44	387.15	193.39	112.65	34.31
Max	109.44	69.64	54.23	32.49	6.94

Route animation: https://youtu.be/p77IbAbIEY8

Centerville

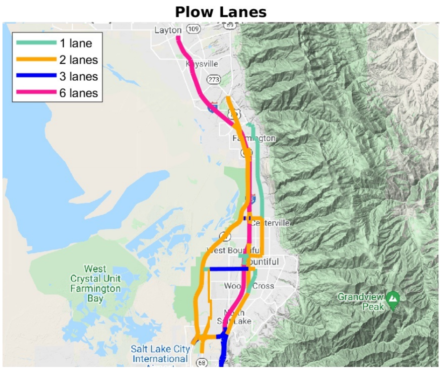

Table EC.15: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	94.54	50.38	2.57	0.86	6.19
# 2	94.91	49.14	16.29	9.84	4.91
# 3	94.97	50.24	7.39	5.56	5.59
# 4	94.92	49.91	13.09	6.90	5.38
# 5	94.75	53.91	28.42	19.39	4.32
# 6	94.76	44.47	21.08	10.17	4.29
# 7	94.72	44.24	30.83	14.90	3.67
# 8	94.70	40.72	22.23	10.37	3.79
# 9	75.57	34.22	19.24	9.56	3.08
Total	833.84	417.23	161.14	87.55	41.21
Max	94.97	53.91	30.83	19.39	6.19

Clearfield

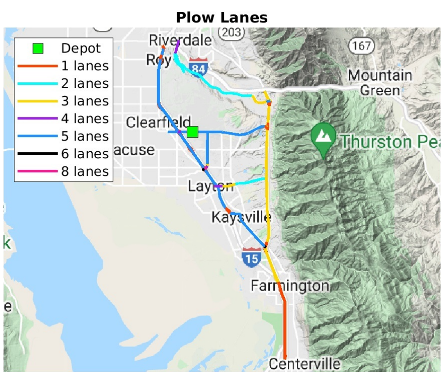

Table EC.16: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	93.81	49.77	44.28	22.81	3.37
# 2	92.89	37.44	28.97	13.92	2.94
# 3	93.61	48.55	47.67	24.05	3.06
# 4	93.39	49.36	33.13	17.17	4.02
# 5	90.78	47.78	47.95	24.11	2.96
# 6	92.59	43.36	47.53	22.56	2.60
# 7	93.66	55.25	80.20	45.13	1.27
# 8	72.20	35.43	26.75	13.33	2.76
Total	722.93	366.94	356.48	183.08	22.98
Max	93.81	55.25	80.2	45.13	4.02

Huntsville

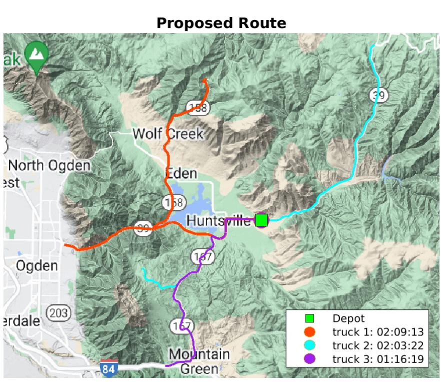

Table EC.17: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	129.22	53.84	11.52	4.80	6.13
# 2	123.36	51.40	11.52	4.80	5.82
# 3	76.32	31.80	32.88	13.70	2.26
Total	328.90	137.04	55.92	23.30	14.21
Max	129.22	53.84	32.88	13.70	6.13

Route animation: https://youtu.be/6nFSDeGjUGY

Morgan

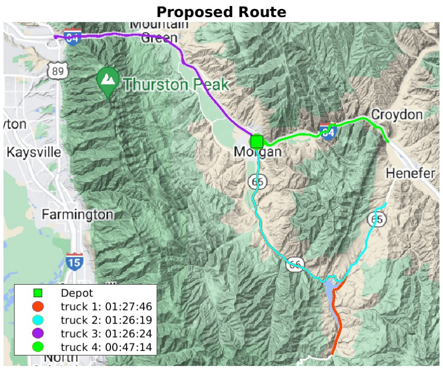

Table EC.18: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	87.76	38.28	65.58	27.22	1.38
# 2	86.32	37.80	0.00	0.00	4.72
# 3	86.40	48.10	76.76	44.85	0.41
# 4	47.23	23.27	29.86	17.49	0.72
Total	307.71	147.45	172.20	89.56	7.23
Max	87.76	48.10	76.76	44.85	4.72

Route animation: https://youtu.be/4q8-50DdDI4

North Logan

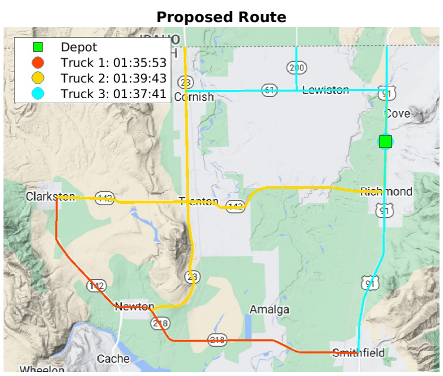

Table EC.19: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	95.88	51.14	11.21	5.98	5.65
# 2	99.72	53.18	3.45	1.84	6.42
# 3	97.69	52.10	3.45	1.84	6.28
Total	293.29	156.42	18.11	9.66	18.34
Max	99.72	53.18	11.21	5.98	6.42

Route animation: https://youtu.be/k27Skj1_qn8

Ogden

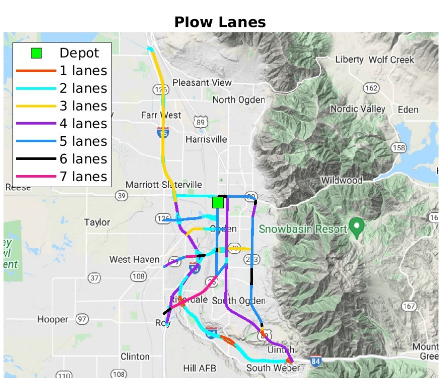

Table EC.20: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	105.37	50.83	45.59	21.76	3.63
# 2	105.98	59.89	24.33	12.16	5.97
# 3	105.85	54.60	42.77	21.53	4.13
# 4	105.47	38.3	4.09	1.62	4.59
# 5	105.46	60.99	44.57	26.12	4.36
# 6	105.56	55.25	19.86	8.53	5.84
# 7	105.71	43.67	24.2	10.00	4.21
# 8	105.82	38.44	19.62	6.94	3.94
# 9	72.85	27.87	7.46	2.94	3.12
Total	918.07	429.84	232.48	111.60	39.78
Max	105.98	60.99	45.59	26.12	5.96

Riverside

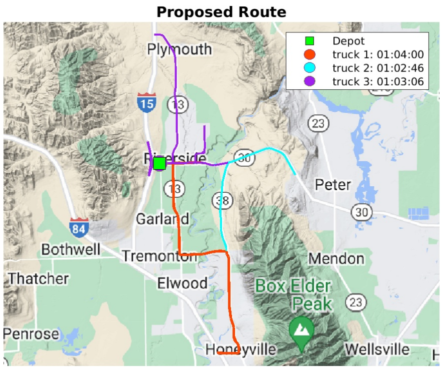

Table EC.21: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	(miles)	Salt (tons)
# 1	64.00	33.85	1.36	0.74	4.13
# 2	62.76	35.76	7.86	4.34	3.92
# 3	63.10	33.16	14.54	8.20	3.12
Total	189.86	102.77	23.76	13.28	11.19
Max	64.00	35.76	14.54	8.20	4.13

Route animation: https://youtu.be/cF0TCiryk54

South Logan

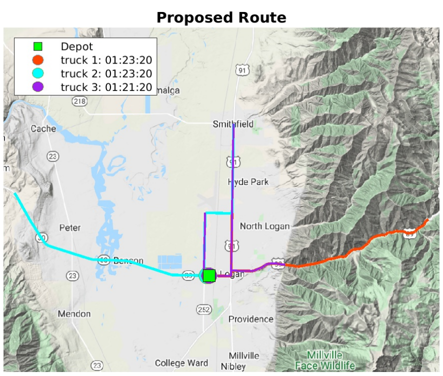

Table EC.22: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	(miles)	Salt (tons)
#1	83.82	43.81	0.00	0.00	5.47
#2	84.01	43.18	1.25	0.52	5.33
#3	81.33	33.88	8.28	3.45	3.80
Total	249.16	120.87	9.53	3.97	14.61
Max	84.10	43.81	8.28	3.45	5.47

Route animation: https://youtu.be/A-_WOT530pQ

Wellsville

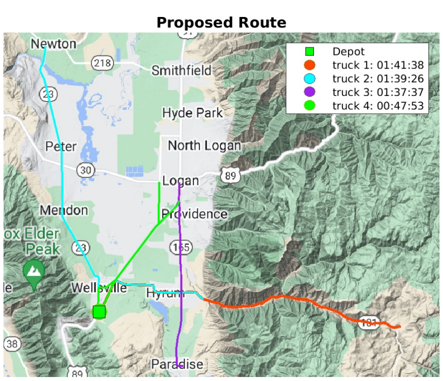

Table EC.23: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	(miles)	Salt (tons)
#1	101.64	45.48	12.89	6.94	4.82
#2	99.44	53.18	14.41	7.14	5.76
#3	97.61	56.66	3.36	2.24	6.80
#4	47.89	28.52	0.00	0.00	3.56
Total	346.58	183.84	30.66	16.32	20.94
Max	101.64	56.67	14.41	7.14	6.80

Route animation: https://youtu.be/kA5m4n7p0Gc

Laketown

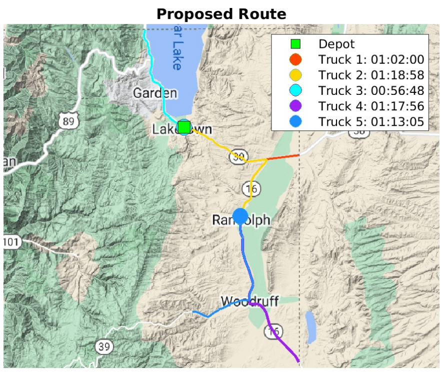

Table EC.24: Route performance with new fleet size

Truck ID	Duration (mins)	Distance (miles)	Deadhead (mins)	Deadhead (miles)	Salt (tons)
# 1	62.00	31.00	0.00	0.00	3.88
# 2	78.96	39.48	46.40	23.20	2.04
# 3	56.80	28.40	0.00	0.00	3.55
# 4	77.93	41.94	21.14	10.57	3.92
# 5	73.08	36.54	21.14	10.57	3.25
Total	348.77	177.36	88.68	44.34	16.62
Max	78.96	41.94	46.40	23.20	3.92

Route animation: https://youtu.be/U4l8wJpuNow

References

Sabharwal, C.L. 2019. Blended root finding algorithm outperforms bisection and regula falsi algorithms. *Mathematics* **7**(11) 1118:1–1118:16.

Shioura, A., K. Tanaka. 2007. Polynomial-time algorithms for linear and convex optimization on jump systems. *SIAM Journal on Discrete Mathematics* **21**(2), 504–522.