Constraint Programming to Improve Hub Utilization in Autonomous Transfer Hub Networks

Chungjae Lee
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Wirattawut Boonbandansook
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Vahid Eghbal Akhlaghi
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Kevin Dalmeijer[1]
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Pascal Van Hentenryck
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

The Autonomous Transfer Hub Network (ATHN) is one of the most promising ways to adapt self-driving trucks for the freight industry. These networks use autonomous trucks for the middle mile, while human drivers perform the first and last miles. This paper extends previous work on optimizing ATHN operations by including transfer hub capacities, which are crucial for labor planning and policy design. It presents a Constraint Programming (CP) model that shifts an initial schedule produced by a Mixed Integer Program to minimize the hub capacities. The scalability of the CP model is demonstrated on a case study at the scale of the United States, based on data provided by Ryder System, Inc. The CP model efficiently finds optimal solutions and lowers the necessary total hub capacity by 42%, saving $15.2M in annual labor costs. The results also show that the reduced capacity is close to a theoretical (optimistic) lower bound.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming

Keywords and phrases Constraint Programming, Autonomous Trucking, Tranfer Hub Network

Digital Object Identifier 10.4230/LIPIcs.CP.2023.46

Category Short Paper

Funding This research was partly funded through a gift from Ryder and partly supported by the NSF AI Institute for Advances in Optimization (Award 2112533).

Acknowledgements Special thanks to the Ryder team for their invaluable support, expertise, and insights.

1 Introduction

It is widely believed that autonomous trucks will revolutionize the freight industry, and many companies have started exploring its potential [5, 6, 7, 11, 26, 27]. Some of the major players describe the *transfer hub business model* to be the most likely implementation for

[1] Corresponding author

29th International Conference on Principles and Practice of Constraint Programming (CP 2023).
Editor: Roland H. C. Yap; Article No. 46; pp. 46:1–46:11
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

autonomous trucking [21, 25, 28]. An Autonomous Transfer Hub Network (ATHN) is a network of autonomous truck ports (*transfer hubs*) that leverages the strengths of humans and automation in their most effective roles. Autonomous trucks handle the monotonous middle mile to transport goods between the transfer hubs, while humans handle the complex first and last miles through local cities and deal with customer contacts. According to Roland Berger [21], implementing a transfer hub model can lower operational costs by 22% to 40%. A case study in the Southeast of the United States by Ryder System, Inc. and the Socially Aware Mobility lab [22] reports savings from 27% to 40%, supporting earlier estimates. The authors model optimizing ATHN operations as a scheduling problem, and a Constraint Programming (CP) model is used to minimize the empty miles [4]. Subsequent work by [13] presents a column-generation approach and a bespoke network-flow model to quickly find high-quality solutions. These findings are combined by [14] into a flow-based Mixed Integer Programming (MIP) framework that can solve large-scale instances over a long time horizon (e.g., a month) optimally in reasonable time. These capabilities also allow for detailed analyses of the benefits and costs of ATHNs.

This paper builds on previous work by introducing a framework for optimizing both ATHN operations and transfer hub capacity utilization. Capacity planning plays a critical role in an actual implementation of the ATHN for operational scheduling. Furthermore, hub capacity has an impact on the number of essential personnel required, which is a crucial factor to consider in labor planning and policy design. Prior research on ATHN operations primarily focused on routing and scheduling, but neglected the capacity considerations involved. Including capacity constraints in a scalable way is not obvious. The CP method in [4] could be extended to incorporate hub capacity constraints by modeling a Resource-Constrained Project Scheduling Problem [9, 12, 20] at every hub and adding cumulative constraints [1, 24] for hub capacity. However, the poor performance reported in [4] makes it unlikely this method will provide good solutions on the national level. The MIP framework from [14] does handle large scale systems, but does not support a cumulative constraint. Alternatives typically require sophisticated modeling techniques and solution methods such as Time-expanded Networks or Dynamic Discretization Discovery [2, 3, 10, 15, 23, 29, 30], which are not obvious to scale either.

The method presented in this work optimizes ATHN operations and improves hub utilization even for large-scale systems by combining the strengths of MIP and CP. The MIP model is used to generate routes and an initial schedule, while the CP model is used to shift the schedule and minimize the required hub capacities. A case study based on real data is conducted to demonstrate the effectiveness of the new methodology on an ATHN system spanning the United States for a four-week horizon. The proposed CP model efficiently finds optimal solutions and lowers the necessary total hub capacity by 42%. This reduction in capacity may save $15.2M per year in labor cost. Furthermore, it is shown that this is close to the best possible savings for any initial schedule. This paper also includes a sensitivity analysis and provides operational insights for future implementation of the ATHN framework. The remainder of this paper is organized as follows. Section 2 presents the MIP and CP methodology. Section 3 describes the data and experimental settings used in the case study. Results and sensitivity analysis are presented in Section 4. Finally, Section 5 provides conclusions and suggests directions for future research.

2 Methodology

This paper uses the methods by [14] to design an ATHN and to optimize the routes of the autonomous vehicles with a MIP. The resulting solution minimizes the cost of the system, but does not take into account the necessary capacity at each of the hubs, which may lead to low hub utilization. To address this issue, this paper introduces a CP model to shift the schedule in such a way that the original time windows remain satisfied, and the necessary hub capacities are minimized.

2.1 ATHN Design and Operations

The input data for the design and optimization of ATHN consists of a set of loads with origins, destinations, and release times. Following [14], K-means clustering is used to determine hub locations H, and loads are assigned to hubs according to a hub-assignment rule that minimizes the total driving distance (with autonomous miles discounted by a factor γ).

To optimize the ATHN operations, first define a set of tasks $T = \{1, 2, \ldots\}$, where each task $t \in T$ corresponds to moving a load from origin hub $h_t^+ \in H$ to destination hub $h_t^- \in H$ with an autonomous truck. Each task $t \in T$ is associated with a desired pickup time p_t and a flexibility $\Delta \geq 0$, resulting in a pickup time window of $[p_t - \Delta, p_t + \Delta]$. A *task graph* $G = (V, A)$ is introduced to find the optimal sequence of tasks for every vehicle. The set $V = T \cup \{0\} \cup \{|T| + 1\}$ consists of vertices that correspond to the tasks, together with a source node 0 and a sink node $|T|+1$. Choosing an arc $a \in A$ indicates that the corresponding tasks are performed sequentially by the same vehicle. An arc between two tasks $t, t' \in T$ represents loading at h_t^+, moving freight from h_t^+ to h_t^-, unloading at h_t^-, and relocating from h_t^- to $h_{t'}^+$ to be ready for the next task. Each arc is associated with a corresponding time τ_a. The time for loading or unloading is given by a parameter σ, and OpenStreetMap times are used for driving and relocation [17].

The cost is calculated in two parts: d_t is the *direct cost*, which represents the cost of serving task $t \in T$ directly with a conventional truck without using any of the hubs. This cost is taken to be the total distance for delivery and empty return. Note that the current non-autonomous system corresponds to using only direct trips. The second component is a *cost differential* c_a associated with each arc $a \in A$. For arc $a = (t, t')$ this represents the difference in cost to switch task t from conventional to autonomous delivery, including the relocation cost from h_t^- to $h_{t'}^+$. That is, d_t is the direct trip cost, and $d_t + c_{tt'}$ is the cost to serve task $t \in T$ autonomously and relocate to the start of task t'. The autonomous middle miles are discounted by a factor $\alpha \in [0, 1]$ and it is assumed that a fraction $\beta \in [0, 1)$ of the implied first/last miles are empty. Additional details on how τ_a and c_a are calculated are provided by [14].

Let y_a be a binary variable that indicates that arc $a \in A$ is selected, and let x_t be the start time of task $t \in T$. For convenience, let δ_v^+ and δ_v^- denote the out-arcs and in-arcs of vertex $v \in V$, respectively. For a given number of autonomous trucks K, MIP (1) asks for a set of at most K routes from source to sink that cover different tasks. Objective (1a) minimizes the total cost. If a task is not covered it means it is served by conventional means and the cost is that of a direct trip d_t. If the task is covered, the cost differential $c_{tt'}$ is added to calculate the autonomous cost. Constraints (1b) enforce that every task is performed at most once. Constraints (1c) are flow conservation constraints, and Constraint (1d) limits the number of vehicles to K. Constraints (1e) are Miller-Tucker-Zemlin constraints [16] that ensure sufficient time passes between subsequent tasks, where M is a sufficiently large constant. These constraints also eliminate cycles. Finally, the variables and their domains are given by Equations (1f) and (1g). The MIP Model (1) is solved with a blackbox solver after applying the acceleration techniques detailed in [14]. The arc-flows (y-variables) are translated into a set of routes, which are represented as sequences of tasks, by tracing the flows from the source node 0 to the sink node $|T| + 1$. Note that for given optimal routes, the start times (x-variables) are typically not unique. For consistent analysis, the start times are shifted to as early as possible in post processing.

$$
\begin{aligned}
\min \quad & \sum_{t \in T} d_t + \sum_{a \in A} c_a y_a, && && \text{(1a)} \\
\text{s.t.} \quad & \sum_{a \in \delta_t^+} y_a \leq 1 && \forall t \in T, && \text{(1b)} \\
& \sum_{a \in \delta_t^+} y_a = \sum_{a \in \delta_t^-} y_a && \forall t \in T, && \text{(1c)} \\
& \sum_{a \in \delta_0^+} y_a \leq K, && && \text{(1d)} \\
& x_{t'} \geq x_t + \tau_{tt'} - M(1 - y_{tt'}) && \forall t, t' \in T, (t, t') \in A, && \text{(1e)} \\
& x_t \in [p_t - \Delta, p_t + \Delta] && \forall t \in T, && \text{(1f)} \\
& y_a \in \mathbb{B} && \forall a \in A. && \text{(1g)}
\end{aligned}
$$

Figure 1 Mixed Integer Program for Optimizing ATHN Operations.

2.2 Minimizing Required Hub Capacities

The routes and schedule obtained from the MIP do not take hub utilization into account. Therefore, it may happen that many trucks are loading and unloading at the same hub at the same time. To address this issue, a CP model is introduced to shift the schedule to minimize the necessary loading/unloading capacity at the hubs, while satisfying the original time windows and maintaining the same route (sequence of tasks) for every vehicle.

The CP model is based on $\mathcal{K}$; the set of routes obtained from MIP (1). Every route $k \in \mathcal{K}$ is split into an sequence of jobs $\mathcal{J}_k = \{1, 2, \ldots\}$. For notational convenience, the jobs are numbered sequentially by the order in which they are performed, rather than using the original task numbers. Figure 2 provides a visualization, which will serve as a running example. Each job $j \in \mathcal{J}_k$ has up to four properties:

- type$(j) \in \{\mathtt{l}, \mathtt{d}, \mathtt{u}, \mathtt{r}, \mathtt{p}\}$; type of job: load, drive, unload, relocate, park, respectively.
- duration(j) (only for types `l`, `d`, `u`, `r`); duration of this job.
- task$(j) \in T$ (only for types `l`, `d`, `u`); task associated with this job.
- hub$(j) \in H$ (only for types `l`, `u`, `p`); hub associated with this job.

Each route is modeled with the same repeating sequence of loading at the origin hub (`l`), waiting before driving (`p`), driving (`d`), waiting at the destination hub before unloading (`p`), unloading (`u`), waiting before relocating to the next task if any (`p`), relocating (`r`), and waiting at the origin hub of the next task until loading (`p`). Note that jobs of type `l`, `d`, `u`, `r` have a fixed duration, while the duration of `p` jobs is flexible and can be zero. When no relocation is necessary (the previous destination is equal to the next origin), the `r` job is still defined with duration zero for convenience. Every `l`, `d`, `u` job is trivially associated with an original task $t \in T$. Jobs `l`, `u`, `p` for which the vehicle is standing still are associated with a hub as described above.

CP Model

The CP Model (2) is based on variables S_j^k that indicate the start time of job $j \in \mathcal{J}_k$ in route $k \in \mathcal{K}$, and variables C_h that indicate the necessary loading/unloading capacity at hub $h \in H$. For convenience, $S_{|\mathcal{J}_k|+1}^k$ is defined to represent the time at which the final job

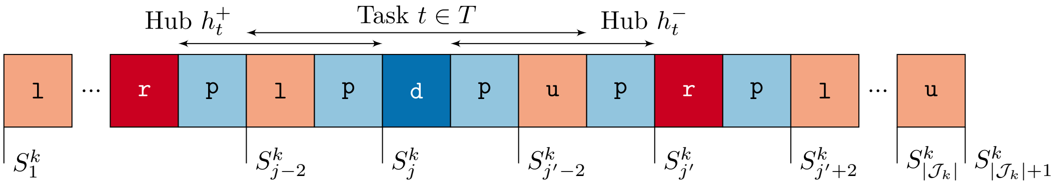


Figure 2 Jobs $\mathcal{J}_k$ for Route $k \in \mathcal{K}$ (jobs $j, j' \in \mathcal{J}_k$ are used as examples in the main text).

$$
\begin{aligned}
\min \quad & \sum_{h \in H} C_h, && && \text{(2a)}\\
\text{s.t.} \quad & I_j^k = \texttt{Interval}([S_j^k, S_{j+1}^k]) && \forall k \in \mathcal{K}, j \in \mathcal{J}_k, && \text{(2b)}\\
& \texttt{Cumulative}\left([I_j^k | k \in \mathcal{K}, j \in \mathcal{J}_k, \text{type}(j) \in \{\texttt{l}, \texttt{u}\}, \text{hub}(j) = h], C_h\right) && \forall h \in H, && \text{(2c)}\\
& S_{j+1}^k = S_j^k + \text{duration}(j) && \forall k \in \mathcal{K}, j \in \mathcal{J}_k, \text{type}(j) \in \{\texttt{l}, \texttt{d}, \texttt{u}, \texttt{r}\}, && \text{(2d)}\\
& \text{dom}(S_j^k) = [\underline{s}_j^k, \overline{s}_j^k] && \forall k \in \mathcal{K}, j \in \mathcal{J}_k. && \text{(2e)}
\end{aligned}
$$

Figure 3 Constraint Programming Model for Minimizing Required Hub Capacities.

u is completed (see Figure (2)). Objective (2a) minimizes the total necessary hub capacity. To calculate the hub capacity, the model first defines an interval variable I_j^k for every job (Equation (2b)), which implicitly enforces $S_j^k \leq S_{j+1}^k$. Equation (2c) defines a cumulative constraint for every hub $h \in H$ to collect the intervals of the jobs with type l and u at that hub, and to assign the neccesary hub capacity to C_h. Note that this cumulative constraint has a variable as the capacity. It is also worth mentioning that Constraints (2c) span all the vehicles. Constraints (2d) ensure the correct job duration when a duration is defined (p jobs are flexible). Finally, Equation (2e) defines the domains of the S-variables, where constants $\underline{s}_j^k$, $\overline{s}_j^k$ remain to be defined. This paper focuses on loading/unloading capacity, but note that CP Model (2) is easily adapted to other objectives, such as minimizing parking space.

To ensure that the time flexibility Δ is respected, the domains of the S-variables need to be defined accordingly. For job $j \in \mathcal{J}_k$ of route $k \in \mathcal{K}$ and type$(j) = \texttt{l}$, the domain is defined around the desired pickup time of the task to match the MIP: dom$(S_j^k) = [p_{\text{task}(j)} - \Delta, p_{\text{task}(j)} + \Delta]$. This domain is translated to the following d and u jobs to make sure that the flexibility is not exceeded until the task is complete. With slight abuse of notation, this gives the translated domains dom$(S_j^k) = $ dom$(S_{j-2}^k) + $ duration$(j-2)$ for jobs of type$(j) \in \{\texttt{d}, \texttt{u}\}$ (see Figure 2).

Redundant Bounds and Constraints

Non-trivial bounds for jobs of type r and p are not strictly necessary, but it is straightforward to derive the following:

$$
\text{dom}(S_j^k) = \begin{cases} [\underline{s}_{j-2}^k + \text{duration}(j-2), \overline{s}_{j+2}^k - \text{duration}(j)] & \text{if type}(j) = \texttt{r},\\ [\underline{s}_{j-1}^k + \text{duration}(j-1), \overline{s}_{j+1}^k] & \text{if type}(j) = \texttt{p}. \end{cases} \tag{3}
$$

Relocation can only start after unloading is completed, and has to start to be in time for the next loading (see job j' in Figure 2). In a similar way, parking can only start after the previous job is completed, and parking for duration zero is possible until the upper bound of the next job.

Table 1 Baseline Parameter Values for the Case Study.

Parameter	Value
n	6842 loads
$\lvert H\rvert$	100 transfer hubs
γ	40% discount for autonomous mileage during hub-assignment
β	25% first/last-mile inefficiency
α	25% discount for autonomous mileage
Δ	1 hour pickup-time flexibility
σ	30 minutes autonomous truck loading/unloading time
K	100 autonomous trucks

For the current Objective (2a), the following observation is used to reduce the search space: To minimize the loading/unloading capacity, it does not matter when relocation takes place between the jobs `u` and `l`. It is therefore possible without loss of generality to impose that relocation starts immediately after unloading:

$$S_j^k = S_{j-1}^k \quad \forall k \in \mathcal{K}, j \in \mathcal{J}_k, \text{type}(j) = \texttt{r}. \tag{4}$$

3 Case Study

This paper conducts a realistic case study based on data from Ryder System, Inc. (Ryder), one of the largest transportation and logistics companies in North America. Ryder has provided a dataset that is representative for its dedicated transportation business in the US, reducing the scope to orders that are strong candidates for automation. Following [14], the case study focuses on orders that are *challenging* in the sense that they would currently induce an empty return trip. Every order represents a load with an origin, a destination, and a scheduled release time. The hubs are chosen based on data from October to December 2019, while the experiments are based on 6842 loads in the first four weeks of October.

Experimental Settings

The parameter settings are taken from [14] and summarized in Table 1. All methods from Section 2 are implemented in Python 3.9. The MIP models are solved with Gurobi 9.5.2, and the CP models are solved with CP-SAT 9.6 [19]. All time-related data is rounded to the nearest minute to facilitate the integer domains that are required by CP-SAT. The experiments are conducted on a Linux machine with dual Intel Xeon Gold 6226 CPUs on the PACE Pheonix cluster [18]. Each experiment is assigned to use at most 24 cores and 192GB of RAM. If Gurobi runs out of memory, the experiment is repeated on a machine with 384GB of RAM, and if that fails, the number of cores is halved until the solver terminates successfully. CP-SAT did not encounter memory issues at any point. The Gurobi time limit is set to three hours, except for the sensitivity analysis for flexibility Δ, which is given 12 hours to obtain better solutions and is warm started with a MIP start. The CP-SAT solver is warm started with the solution obtained by the MIP, which is provided as a hint.

4 Results

Figure 4 summarizes the results of using the CP model to minimize the necessary hub capacity for loading and unloading trucks. "Before CP" shows the required capacity if the MIP solution were implemented immediately, while "After CP" shows the results after applying the CP model. The ATHN problem was solved to optimality for all instances except

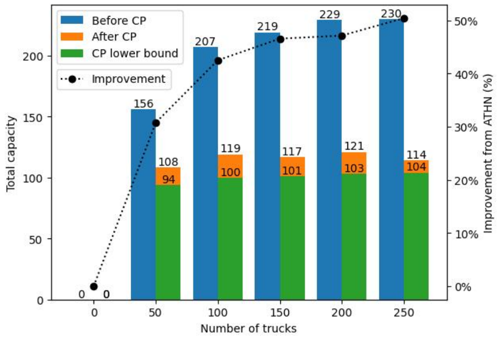


Figure 4 Capacity Before and After CP.

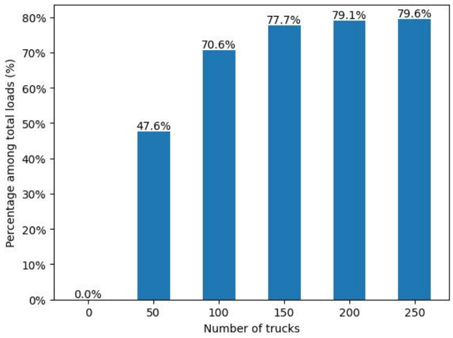
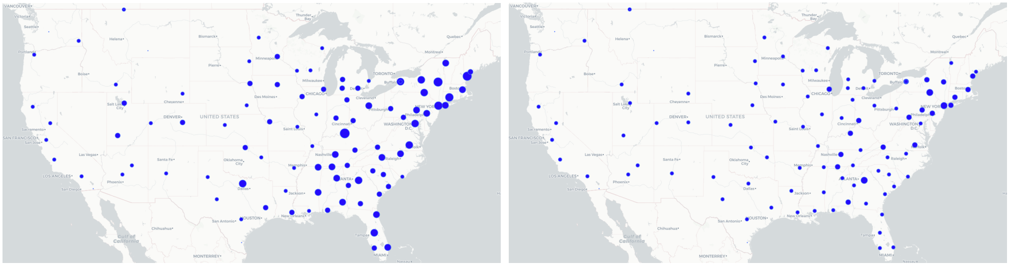


Figure 5 Loads Served Autonomously.



(a) Before CP Optimization.

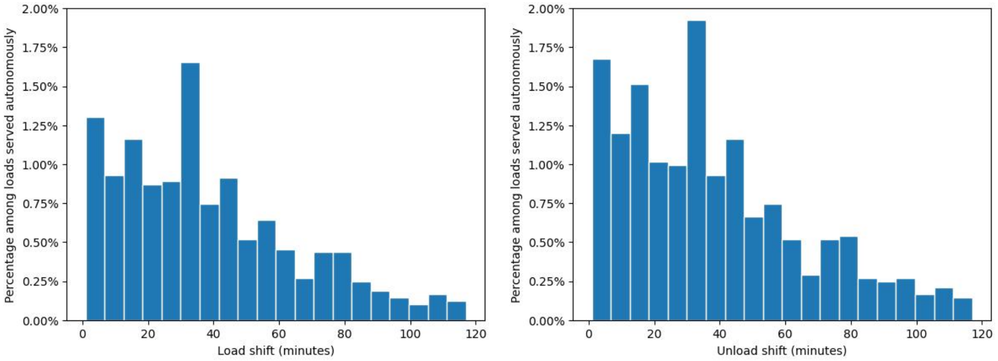
(b) After CP Optimization.

Figure 6 Capacity Reduction for the 100 Truck Case (circle area proportional to hub capacity).

when $K = 50$, which remained at a 0.05% optimality gap within the given time limit. For the CP problem, all instances were solved to optimality in under 30 seconds. The figure shows that the CP model is highly effective in reducing the total hub capacity for all instances, reducing the necessary capacity by 31% up to 50%. For the base case of $K = 100$ the CP model can reduce capacity by 42%. Note that this may correspond to significant monetary savings: If each unit of loading/unloading capacity requires the assistance of a mechanic around the clock (three shifts of $57,557 per year [8]), the CP model reduces the annual labor cost by $15.2 million. Figure 4 also shows that the resulting capacity is close to the lower bound *for any ATHN solution*, which is obtained by removing the time constraints between subsequent tasks, i.e., treat every task as if it is the only task on the route.

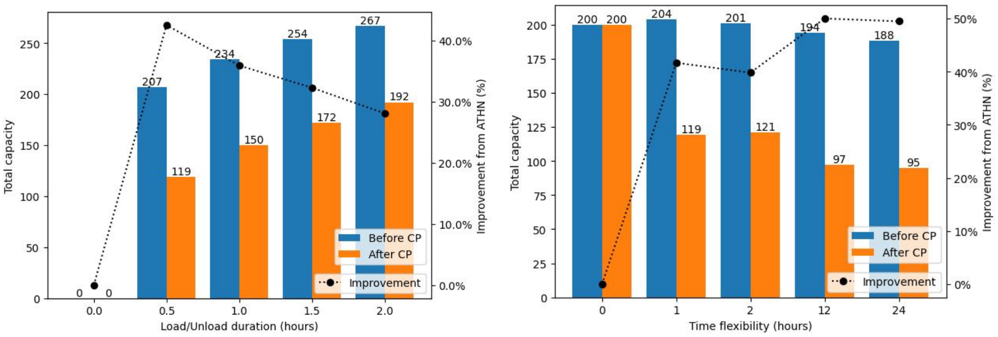
As the number of trucks increases, the ATHN starts serving more loads autonomously, as shown by Figure 5. Without the CP model, this leads to a substantial increase in hub capacity as more loads are added to the system. The CP model completely mitigates this effect, and is able to maintain an almost stable hub capacity as the workload increases. This reveals a suprising robustness to accomodate new orders that the CP model is able to exploit. When making investment decisions and hiring personnel to operate the hubs, this is a very desirable property. The maps in Figure 6 demonstrate that the necessary capacity is reduced throughout the system, and the peaks in the South and Northeast have decreased significantly. The capacity at the hub near Louisville, Kentucky for example was reduced from 7 to 2.

Interestingly, only a small portion of the jobs needs to be rescheduled to obtain the substantial savings in hub capacity. E.g., in the 100-truck case only 13% of loadings were rescheduled. Figure 7 provides histograms for the absolute size of the shift for the loading and unloading times that were shifted. It can be seen that a majority of these jobs were



(a) Histogram of Shifted Loading Times. **(b)** Histogram of Shifted Unloading Times.

Figure 7 Shifted Schedule Times for the 100 Truck Case.



(a) Impact of Loading/Unloading Time σ. **(b)** Impact of Flexibility Δ.

Figure 8 Sensitivity Analysis Loading/Unloading Time and Flexibility for the 100 Truck Case.

moved by no more than 30 minutes, which shows that the ATHN solutions are sufficiently flexible to be adjusted without propagating delays through the schedule. The fact that the CP model is easy to solve and only makes small modifications also makes it an attractive tool at the operational level: if order details are changed or delays are encountered, the CP model can quickly be re-solved to avoid causing overlap in loading and unloading at the hubs.

Sensitivity Analysis

The loading/unloading time σ and the flexibility Δ are two significant factors that affect the ATHN. Figure 8 summarizes how these parameters impact the total hub capacity and the CP model's performance. The ATHN is solved for loading/unloading time from zero to two hours, and for no flexibility up to one day of flexibility. Note that loading/unloading time up to two hours may be realistic if the vehicle is inspected every time it leaves or enters a hub.

For σ, the ATHN solver found optimal solutions for all instances and the CP problem found optimal schedules within 15 seconds. As σ increases, the total hub capacity increases because jobs overlap more frequently. The CP model is not able to fully compensate for this effect, but can still improve the hub capacity by at least 28%.

For Δ, optimal solutions were found for $\Delta \in \{0\text{h}, 1\text{h}\}$, a near-optimal solution within 0.05% optimality gap was found for $\Delta = 2\text{h}$, while $\Delta \in \{12\text{h}, 24\text{h}\}$ remained at a 3% optimality gap. The CP solver took 7 minutes for $\Delta = 12\text{h}$ and 13 minutes for $\Delta = 24\text{h}$

Table 2 CP Solving Time Speedup compared to Omitting Redundant Bounds and Constraints.

		Number of Trucks				
Redundant Bounds	Redundant Constraints	50	100	150	200	250
✗	✗	1.0x	1.0x	1.0x	1.0x	1.0x
✓	✗	1.1x	1.4x	1.2x	0.9x	0.8x
✗	✓	3.9x	6.8x	7.4x	3.9x	5.4x
✓	✓	3.5x	5.5x	6.3x	3.1x	5.5x

to find optimal schedules, which suggests that the CP computation time is more sensitive to changes in Δ than in σ. This is explained by the fact that increasing the flexibility significantly increases the search space. Again, the CP model shows substantial improvement over the initial solution with improvements ranging from 39% to 50% for the case study. When the flexibility increases, the necessary capacity goes down even before CP is applied, but the CP model is able to benefit more from the additional freedom.

Impact of Redundant Bounds and Constraints

Table 2 shows the impact of the redundant bounds and constraints introduced in Section 2. These results are based on the baseline parameter settings for different numbers of trucks. It can be seen that only adding redundant bounds can both speed up or slow down the solver. For example, the solver becomes 1.4 times faster for $K = 100$ trucks, but a factor 0.8 slower for $K = 250$ trucks. The main benefit comes from adding redundant constraints, which speeds up the solver by up to 7.4 times for $K = 150$ trucks. When the redundant bounds and constraints are combined, performance may be improved further, as is the case for $K = 250$ trucks, but it appears that only including redundant constraints is the most efficient setting for these experiments.

5 Conclusion

The Autonomous Transfer Hub Network (ATHN) is one of the most promising ways to adapt self-driving trucks for the freight industry. This paper proposes a framework for optimizing ATHN operations with respect to hub utilization. To accomplish this, a MIP model generates autonomous vehicle routes and an initial schedule, which is then optimized using a CP model to reduce the required hub capacity for loading and unloading trucks. Results from the Ryder case study demonstrate the effectiveness of this approach, with the CP model reducing the total capacity by at least 42% while requiring only minor schedule modifications. This may save $15.2M per year in labor cost and is close to the lowest possible capacity for any initial schedule. Sensitivity analysis on loading/unloading duration and flexibility provides practical insights for ATHN operations and systematically shows the benefit of the CP model. Future work may explore alternative objective functions, such as minimizing parking space. Another interesting direction is to attempt to jointly optimize routes and hub capacity directly.

References

1 Abderrahmane Aggoun and Nicolas Beldiceanu. Extending chip in order to solve complex scheduling and placement problems. *Mathematical and Computer Modelling*, 17(7):57–73, 1993. doi:10.1016/0895-7177(93)90068-A.

2 Natashia Boland, Mike Hewitt, Luke Marshall, and Martin Savelsbergh. The Continuous-Time Service Network Design Problem. *Operations Research*, 65(5):1303–1321, 2017. doi:10.1287/opre.2017.1624.

3 Natashia Boland and Martin Savelsbergh. Perspectives on integer programming for time-dependent models. *TOP*, 27:147–173, 2019. doi:10.1007/s11750-019-00514-4.

4 Kevin Dalmeijer and Pascal Van Hentenryck. Optimizing Freight Operations for Autonomous Transfer Hub Networks. *arXiv:2110.12327*, 2021. arXiv:2110.12327.

5 FleetOwner. TuSimple among autonomous truck companies to join Self-Driving Coalition. https://www.fleetowner.com/, 2021. URL: https://www.fleetowner.com/technology/autonomous-vehicles/article/21152006/tusimple-among-autonomous-truck-companies-to-join-selfdriving-coalition.

6 Forbes. Plus Partners With IVECO To Develop Automated Trucks For Global Deployment. https://www.forbes.com/, 2021. URL: https://www.forbes.com/sites/richardbishop1/2021/04/12/plus-partners-with-iveco-to-develop-automated-trucks-for-global-deployment.

7 FreightWaves. Gatik, Isuzu to partner on autonomous truck platform. https://www.freightwaves.com/, 2021. URL: https://www.freightwaves.com/news/gatik-isuzu-to-partner-on-autonomous-truck-platform.

8 Glassdoor. How much does a Truck Mechanic make? https://www.glassdoor.com/Salaries/truck-mechanic-salary-SRCH_KO0,14.htm, 2023. Accessed: 2023-04-28.

9 Sönke Hartmann and Dirk Briskorn. An updated survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 297(1):1–14, 2022. doi:10.1016/j.ejor.2021.05.004.

10 Edward He, Natashia Boland, George Nemhauser, and Martin Savelsbergh. An exact algorithm for the service network design problem with hub capacity constraints. *Networks*, 80(4):572–596, 2022. doi:10.1002/net.22128.

11 Heavy Duty Trucking. Daimler's Redundant Chassis for Autonomous-Truck Operation. https://www.truckinginfo.com/, 2021. URL: https://www.truckinginfo.com/10157338/daimlers-redundant-chassis-for-autonomous-truck-operation.

12 Willy Herroelen, Bert De Reyck, and Erik Demeulemeester. Resource-constrained project scheduling: A survey of recent developments. *Computers & Operations Research*, 25(4):279–302, 1998. doi:10.1016/S0305-0548(97)00055-5.

13 Chungjae Lee, Kevin Dalmeijer, and Pascal Van Hentenryck. Optimization Models for Autonomous Transfer Hub Networks. *arXiv:2201.06137*, 2022. arXiv:2201.06137.

14 Chungjae Lee, Kevin Dalmeijer, Pascal Van Hentenryck, and Peibo Zhang. Optimizing Autonomous Transfer Hub Networks: Quantifying the Potential Impact of Self-Driving Trucks. *arXiv:2305.03119*, 2023. arXiv:2305.03119.

15 Luke Marshall, Natashia Boland, Martin Savelsbergh, and Mike Hewitt. Interval-Based Dynamic Discretization Discovery for Solving the Continuous-Time Service Network Design Problem. *Transportation Science*, 55(1):29–51, 2021. doi:10.1287/trsc.2020.0994.

16 Clair E. Miller, Albert W. Tucker, and Richard A. Zemlin. Integer Programming Formulation of Traveling Salesman Problems. *Journal of the ACM*, 7(4):326–329, 1960. doi:10.1145/321043.321046.

17 OpenStreetMap. Planet dump retrieved from https://planet.osm.org, 2021. URL: https://www.openstreetmap.org.

18 PACE. *Partnership for an Advanced Computing Environment (PACE)*, 2017. URL: http://www.pace.gatech.edu.

19 Laurent Perron and Vincent Furnon. Google OR-Tools v9.6. https://developers.google.com/optimization/, 2023.

20 A. Alan B. Pritsker, Lawrence J. Waiters, and Wolfe Philip M. Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach. *Management Science*, 16(1):93–108, 1969. doi:10.1287/mnsc.16.1.93.

21 Roland Berger. Shifting up a gear – Automation, electrification and digitalization in the trucking industry. https://www.rolandberger.com/, 2018. URL: https://www.rolandberger.com/publications/publication_pdf/roland_berger_trucking_industry.pdf.

22 Ryder System, Inc. and Socially Aware Mobility Lab. The Impact of Autonomous Trucking: A Case-Study of Ryder's Dedicated Transportation Network. *Ryder Newsroom*, 2021. URL: https://newsroom.ryder.com/news/news-details/2021/Ryder-Teams-Up-with-Georgia-Tech-for-Industrys-First-Data-Driven-Study-on-Impact-of-Autonomous-Trucking/.

23 Yannick Oskar Scherr, Mike Hewitt, Bruno Albert Neumann-Saavedra, and Dirk Christian Mattfeld. Dynamic discretization discovery for the service network design problem with mixed autonomous fleets. *Transportation Research Part B: Methodological*, 141:164–195, 2020. doi:10.1016/j.trb.2020.09.009.

24 Andreas Schutt, Thibaut Feydy, Peter J. Stuckey, and Mark G. Wallace. Explaining the cumulative propagator. *Constraints*, 16:250–282, 2011. doi:10.1007/s10601-010-9103-2.

25 Mohsen Shahandasht, Binaya Pudasaini, and Sean Logan McCauley. Autonomous Vehicles and Freight Transportation Analysis. Technical report, The University of Texas at Arlington, 2019. doi:10.13140/RG.2.2.28484.78726.

26 TechCrunch. Aurora and Volvo partner to bring autonomous long-haul trucks to North America. https://www.techcrunch.com/, 2021. URL: https://techcrunch.com/2021/03/30/aurora-and-volvo-partner-to-bring-autonomous-long-haul-trucks-to-north-america/.

27 Transport Topics. Navistar, TuSimple Partner to Launch Self-Driving Trucks in 2024. https://www.ttnews.com/, 2020. URL: https://www.ttnews.com/articles/navistar-tusimple-partner-launch-self-driving-trucks-2024.

28 Steve Viscelli. Driverless? Autonomous Trucks and the Future of the American Trucker. Center for Labor Research and Education, University of California, Berkeley, and Working Partnerships USA, 2018. URL: http://driverlessreport.org/.

29 Duc Minh Vu, Mike Hewitt, Natashia Boland, and Martin Savelsbergh. Dynamic Discretization Discovery for Solving the Time-Dependent Traveling Salesman Problem with Time Windows. *Transportation Science*, 54(3):703–720, 2020. doi:10.1287/trsc.2019.0911.

30 Haotian Wu, Ian Herszterg, Martin Savelsbergh, and Yixiao Huang. Service Network Design for Same-Day Delivery with Hub Capacity Constraints. *Transportation Science*, 57(1):273–287, 2023. doi:10.1287/trsc.2022.1155.