informs
https://pubsonline.informs.org/journal/ijaa

INFORMS JOURNAL ON APPLIED ANALYTICS
Vol. 54, No. 1, January–February 2024, pp. 20–36
ISSN 2644-0865 (print), ISSN 2644-0873 (online)

THE FRANZ EDELMAN AWARD
Achievement in Operations Research

Innovative Integer Programming Software and Methods for Large-Scale Routing at DHL Supply Chain

Yibo Dang,[a] Theodore T. Allen,[a,*] Manjeet Singh,[b] Jason Gillespie,[b] Jon Cox,[b] James Monkmeyer[b]

[a] Integrated Systems Engineering, The Ohio State University, Columbus, Ohio 43210; [b] Operations Science & Analytics, DHL Supply Chain North America, Westerville, Ohio 43082

*Corresponding author

Contact: dang.121@osu.edu, https://orcid.org/0000-0002-2474-3234 (YD); allen.515@osu.edu, https://orcid.org/0000-0002-9522-3252 (TTA); manjeet.singh3@dhl.com (MS); Jason.gillespie@dhl.com (JG); jon.cox@dhl.com (JC); jim.monkmeyer@dhl.com (JM)

Accepted: November 16, 2023

https://doi.org/10.1287/inte.2023.0087

Abstract. DHL Supply Chain North America moves more than one billion packages each year for corporate customers. Its transportation planners perform routing, bidding, and improvement tasks for many business projects. Prospective customers require DHL to compete to win their business by solving their delivery problems, improving existing supply chain designs, and guaranteeing savings by using fewer trucks or less fuel. Our new transport network optimization (TNO) software suite gives DHL a significant edge in these bidding and improvement tasks. The four modules in the TNO software are as follows: (1) freight optimization, (2) fleet (sizing) optimization, (3) connection hub or pool point-related optimization, and (4) round-trip optimization. We use innovative integer programming approaches in the TNO software, which we developed and implemented in collaboration with Ohio State University, including a new type of two-color ant colony search to efficiently address outsourcing in the first module and the use of dynamic programming for subproblems. Over 2.5 years since 2020, TNO has led to over $117 million in estimated savings for DHL Supply Chain North America and its customers, contributing a 20% win-rate increase and reducing CO_2 emissions by at least 0.1 megatons.

Keywords: transportation • vehicle routing problem with time regulations • network mode optimization • make-buy decisions • ant colony optimization • Edelman award

Introduction

DHL Supply Chain operates one of the world's largest logistics networks, delivering more than 1 billion packages annually for corporate customers. This process is sometimes called fourth-party (4PL) logistics because we operate the supply chain of other companies at various interaction levels that range from simply supplying delivery vehicles to supporting these companies with demand forecasting capabilities, delivery vehicles, and control tower activities. The most complex interactions involve both freight transportation and vehicle routing, which are topics of increasing interest to *IJAA* readers (Freeman et al. 2020). Because of limited labor and truck resources, DHL Supply Chain North America generally uses more than one transportation mode to reduce shipping costs. On a global basis and in North America, DHL Supply Chain was already the largest logistics company. Yet, we won only a small fraction (less than 40%) of the bids for which we competed.

DHL Group is a multinational package delivery and supply chain management company founded in the United States but headquartered in Germany. DHL Group divisions include Supply Chain, eCommerce, Express, Global Forwarding, Freight, and Post & Parcel Germany. DHL Supply Chain North America is headquartered in Westerville, Ohio, near The Ohio State University main campus in Columbus, Ohio. Our team includes members who are helping to lead both DHL Supply Chain North America and DHL Supply Chain. North America was the development and testing ground for this project, but the outputs are already being used globally.

By investing in in-house advanced analytics, we sought to significantly increase our winning percentage and market share. For the bids we won, the software could generate a complete initial delivery plan and refine plans as conditions changed. Although we considered purchasing existing software, no off-the-shelf software that met the scope of our most advanced supply chain projects was available. Therefore, DHL senior management decided to collaborate with The Ohio State University for the development of a new suite of software, transport network optimizer (TNO), to address our needs.

In our networks, minimizing costs depends on the total number of miles on a route, number of stops, load weight, and load volume. Subcontracting to third or fourth parties is necessary when the total customer demand exceeds the capacity of DHL's fleet, DHL cannot guarantee delivery within the customer's time window, or subcontracting is more economical. TNO uses future demand data provided by the potential customer for the bidding process or cherry-picked historical data to determine which packages DHL should deliver, which packages it should outsource to a third-party carrier, and how such deliveries should occur with minimum cost. DHL runs TNO for each corporate customer separately. Most customers have similar needs over time; therefore, the solutions can achieve a highly accurate estimate of actual transportation costs. DHL uses TNO for the contracts of both existing customers and potential new customers during the bidding process. For existing customers, TNO solutions can be used to consolidate shipments, switch the shipment mode of planned deliveries from dedicated fleets to third-party carriers or vice versa and save costs by combining shipments at pooling points (i.e., cross-dock locations that receive a consolidated truckload (TL)-sized shipment from a shipper and then organize the shipment into less-than-truckload (LTL) shipments to the final destination), thus benefitting from an expanded set of round-trip opportunities. Alternatively, during the bidding process, TNO can help in determining a practical and viable least-cost solution that maximizes the chance of winning bids that will be profitable and avoiding bids that will be unprofitable.

We estimate that our TNO software has already contributed more than $117 million (M) in savings for DHL and its corporate customers over 2.5 years of use. The savings since 2020 are largely documented in contractually agreed reports to customers and primarily result from reduced fuel, driver, capital, and outsourcing costs. Approximately 12% of the total cost reductions result from using fewer resources to accomplish the same task. Therefore, the percentages are essentially the clients' overall expense rates. Thus far, the benefits have primarily been achieved in DHL North America; however, they are accruing worldwide as TNO is rolled out globally.

Our key objective in this paper is to explain how DHL Supply Chain seeks to keep its third-party costs low through a series of optimizations by rightsizing its dedicated fleet, consolidating (i.e., pooling) deliveries, and leveraging the possibility that a given truck might pick up deliveries from multiple depots.

Background

DHL Group is uniquely positioned with a comprehensive range of international express, freight transportation, e-commerce, and supply chain management services. It employs approximately 550,000 employees in more than 220 countries and territories. As the world's leader in contract logistics, DHL Supply Chain offers standardized warehousing, transport, and value-added services that can be combined to form customized supply chain solutions. DHL Supply Chain's total revenue was $12 billion (B) annually in 2021.

Because of the complexities associated with running a logistics network, many companies use professional 4PL logistics companies to manage and operate their supply chain businesses. Large 4PL logistics companies move billions of packages every year. These companies are constrained by limited capacity and growing demand, which is a bottleneck for the logistics industry. In two previous papers (Dang et al. 2021, 2022), we describe our "red-black ant colony search" (RB-ACS) approach to solve extremely large vehicle routing problems (VRPs), which involve creating forecasts and determining how hundreds of thousands of packages and hundreds of trucks will be delivered, including the last-mile delivery to a warehouse or store. We do this while considering the possibility that some packages will be delivered by companies other than DHL. The outputs of RB-ACS include routes and allocations of packages to both DHL and third-party firms.

We refer to the activity of applying RB-ACS generically as "freight optimization" and make four assumptions: (1) there are two delivery modes, (2) fleet sizes are fixed, (3) packages cannot be combined and moved to locations where multiple shipments are delivered for pick up (i.e., pool points or consolidation hubs), and (4) trucks cannot pass through more than a single depot in their round trips. In this paper, we discuss DHL Supply Chain's software extensions, which further refine solutions by progressively relaxing each of these assumptions. Collectively, these four assumptions correspond to the four key modules in the TNO software: (1) freight optimization, (2) fleet (sizing) optimization, (3) pool point optimization (PPO), and (4) round-trip optimization (RTO). We progressively relax each assumption rather than developing a joint solution in a single formulation because of the extremely large sizes of the problems considered. Our analysts routinely solve problems involving package counts approaching one million. In our experience, the relaxation of the round-trip assumption (i.e., Assumption 4), which is newly addressed in this paper, is particularly important.

To prepare competitive bidding proposals, analysts routinely apply all four TNO modules. During this process, the analysts construct detailed routes and the associated costs for the modes of transportation, fleet augmentation costs, repositioning costs, and driver and fuel cost reductions from trucks leaving their depot zones. This facilitates the make-buy decisions, that is, which deliveries should be made by DHL's dedicated fleet, what the delivery sequence should be, and which deliveries should be outsourced to third-party freight providers (i.e., common carriers), to minimize the total transportation costs.

Before starting both the initial freight optimization project and subsequent projects, we talked with DHL employees in several regions, including the operations teams, the solution design teams, and the sales teams, all of whom agreed that optimization tools were needed for guaranteeing operational effectiveness. In addition, virtually all the analysts mentioned that DHL's previous process (i.e., the process used before the TNO implementation) usually required several weeks and extensive geographical and market knowledge. Because of the complexity and the size of the associated delivery problem, current off-the-shelf optimization tools were not a viable solution.

Objectives

TNO helps DHL analysts in preparing low-cost proposals for bidding on new customer contracts, setting fleet sizes, and positioning resources for existing customers. To generate feasible solutions, TNO considers various practical aspects within DHL's transportation network. For example, the scheduled delivery locations can be stores, warehouses, a consolidation hub, or a mix of these locations. Additionally, TNO considers routing constraints and the associated costs based on geography and truck types, product types being shipped, drivers' layovers, and time windows.

The inputs include unrouted shipments, common carrier costs, geographical information about distribution centers and demand points, product classes, and truck purchasing costs, including electric trucks. The outputs are the expected costs, routes, schedules, designated transportation modes (i.e., DHL's dedicated fleet or a common carrier), recommended fleet sizes, and recommendations for package consolidations and positioning. Using TNO solutions, the bidders can approach customers with operational transportation cost estimates from DHL within a short amount of time, often delighting customers by quickly conveying a deep understanding of their issues. Then, customers usually compare these estimates with estimates from additional scenarios and costs from the previous processes or quotations from competitors. TNO solutions have helped DHL Supply Chain North America by generating more winning bids (i.e., from approximately 15%–60% with more than 20% generated after 2020) and increasing its revenues by more than 50% since 2015 when TNO precursors were implemented.

The primary goal of developing TNO was to replace a semimanual and iterative planning activity with reliable analytical models to support bidding and operational improvements. Other objectives included shortening the training time needed for new solution designs and the planning time for each project, delighting customers with the speed of generating solutions, and supporting the analysis of multiple scenarios. More efficiently routing the huge LTL network through using third parties, pooling, and repositioning were also important. Addressing these objectives while addressing a wide variety of real situations was required. To address these objectives, we sought to extend the well-known VRP. In general, VRP is concerned with the optimal design of routes in a transportation network. Specifically, the TNO effectively solves a single exceptionally large VRP and thereby addresses a variety of objectives simultaneously.

Literature Review

Since Dantzig and Ramser (1959) proposed the first VRP, much research has focused on solving this combinatorial problem. More recently, there has been an increased focus on multiple transportation modes including "green" options (Seyfi et al. 2022). We refer the reader to Dang et al. (2021, 2022) for a detailed review of this literature. Here, we primarily point out that TNO is designed to address exceptionally large-scale instances of green vehicle routing problems (G-VRP), that is, hundreds of thousands of packages rather than tens or even hundreds. In addition to addressing large-scale problems, we simultaneously consider the complications of time windows, layovers, and relevant scale, which are generally NP-hard (Savelsbergh and Sol 1995). Our review of the literature in this section focuses on such new topics, including fleet (size) optimization, PPO to combine packages and locate them for pickup, and RTO involving multiple depots.

A key aspect of our approach to fleet size/mix optimization is simply iterated applications of RB-ACS from our previous work with different fleet sizes. Whereas our initial approach implemented ant colony search and then made greedy assignments, our current methods build make-buy decision making into the evolution of the symbiotic colonies (Dang et al. 2022). The software also permits designers to study alternative vehicle types as a color option that can address electric vehicles (Dang et al. 2022). Mixed-fleet VRP, including electric vehicles, has been a popular topic in recent years, but we are not aware of any work that thoroughly addresses its complexity and scale as our RB-ACS does (Sassi et al. 2015, Alcaraz et al. 2019, Macrina et al. 2019). Even using metaheuristics, we observe computational instances with at most 100 packages or customers in the literature, whereas DHL Supply Chain analysts routinely consider hundreds of thousands of packages and/or customers in their daily work.

Related to RTO, previous work considered multiple-depot problems (Dondo and Cerdá 2007, Tummel et al. 2013) but constrained packages to regions. Our work considers the possibility that a truck delivers a package in one region and then picks up a package in another region and delivers it as part of its round trip. For this work, the analyst takes the output from earlier steps and finds the longest trips. Then, these long trips are entered into another optimization, which consolidates packages and trucks to potentially visit multiple depots. Therefore, the instances are smaller and in line with the relatively small sizes addressed in the literature (Tummel et al. 2013).

The contribution of our research can be summarized in four points: (1) for fleet sizing and composition, it studies an integrated VRP, which generates complicated route plans, allocations, and mode selections; (2) it provides a high-efficiency heuristic for identifying clusters of packages to be delivered to pool points for pickups in actual situations; (3) it develops exact methods for clustering products to support unconstrained multiple-depot delivery problems; and (4) it describes how the suite of developed tools are successfully applied to bidding and operational improvements at DHL Supply Chain North America.

Problem Description

In routine applications, we analyze data from DHL's existing or potential customers and make transportation mode decisions relative to dedicated fleets and other carrier providers. We might reconsider these decisions after we have won the bid, considering the traffic conditions. A high-quality assignment solution increases the probability of success in bidding for new contracts and offers useful guidance for positioning fleet resources on the network. Here, for confidentiality, we use data from the transportation team at DHL Supply Chain North America, which includes networks in the United States, Mexico, and Canada. The modules within TNO address the following:

- *Make-buy decisions*: TNO's first set of decisions are on whether DHL should service certain shipments, packages, or deliveries with its dedicated fleets or outsource them to common carriers.
- *Vehicle type and number decisions:* TNO's second set of decisions involves choices of the brand, number, and type of specific vehicles, to further reduce total costs.
- *Capacity constraints*: For every leg or "arc" of each route, the total weight and volume associated with the shipments routed through a multiple-stop arc must be less than or equal to the arc's capacity. The "nodes" in our networks are the individual delivery points or customer locations. The "network mode" refers to the type of vehicle used for a delivery, such as DHL, third-party, combustion, or electric vehicle.
- *Time regulations*: Each customer has a prespecified time interval for delivery. In addition, truck drivers are constrained by additional time regulations (e.g., working-hour limits and layover limits). Each time a driver reaches a working-hour limit, that driver must take a layover.
- *Additional constraints*: TNO incorporates detailed operational constraints; examples include the maximum number of layovers per driver, the maximum driving range between each layover, the maximum distance per route, the weekend delivery allowance, and the maximum intranode distance.
- *Cost decomposition*: After generating the optimal solutions for network mode (i.e., the type of vehicle used), TNO solves for the route costs at the shipment level creating an approximate decomposition. This process automates DHL's previous cost modeling (CM) approach, which allows DHL to give its users a quoted cost for each shipment.

The number of feasible solutions for this combinatorial optimization problem increases exponentially as the number of deliveries to be serviced increases. The problem scale may encompass hundreds of thousands of shipments and large numbers of distribution centers.

Manual Freight Optimization Process Before 2019

Before implementing TNO, DHL used the steps we show in Table 1 to make decisions.

The manual process outlined in Table 1 required considerable time and expertise. The first attempt at improving this process of developing proposals to prospective customers (Dang et al. 2021) was based on a single ant colony optimization with some refinements within each route to accelerate the process in 2019.

The 2019 and 2020 Freight Optimization Methods

In 2019, we implemented an ant colony search with greedy route-based outsourcing. The tool, which we initially called D3TO and later TNO, is documented in

Table 1. Solution Procedures at DHL Supply Chain North America Before 2019 and Approximate Time Required for Each Step (Dang et al. 2021)

Steps	Solution procedures before 2019	Average time (hr)
1.	Extract and clean raw shipment information from the database.	8
2.	Preprocess the data, sample the subsets, and manually make initial transportation mode decisions using a given set of constraints.	16
3.	Route the trips and schedule the freight resources using multiple standard commercial software packages based on mileage ranges.	32
4.	Allocate the route cost to each shipment and compare this cost with the transactional common carrier costs obtained from multiple sources.	24
5.	Repeat the routing procedures for the remaining dedicated shipments.	32
6.	Present the quoted cost and "solution" to prospective customers.	8
	Total	120

Table 2. 2019 Solution Procedures at DHL Supply Chain North America and Approximate Time Required for Each Step Before TNO Implementation

Steps	2019 solution procedures	Average time (hr)
1.	Extract and clean raw shipment information from the database.	8
2.	Preprocess the data, sample the subsets, and manually make initial transportation mode decisions using specified constraints.	16
3.	Determine outsourcing and routing decisions using D3TO, which later evolved into TNO's first module.	8
4.	Estimate the costs for the solutions and present proposals to the prospective customers.	8

Dang et al. (2021). It immediately reduced computational times (Table 2) and increased solution quality.

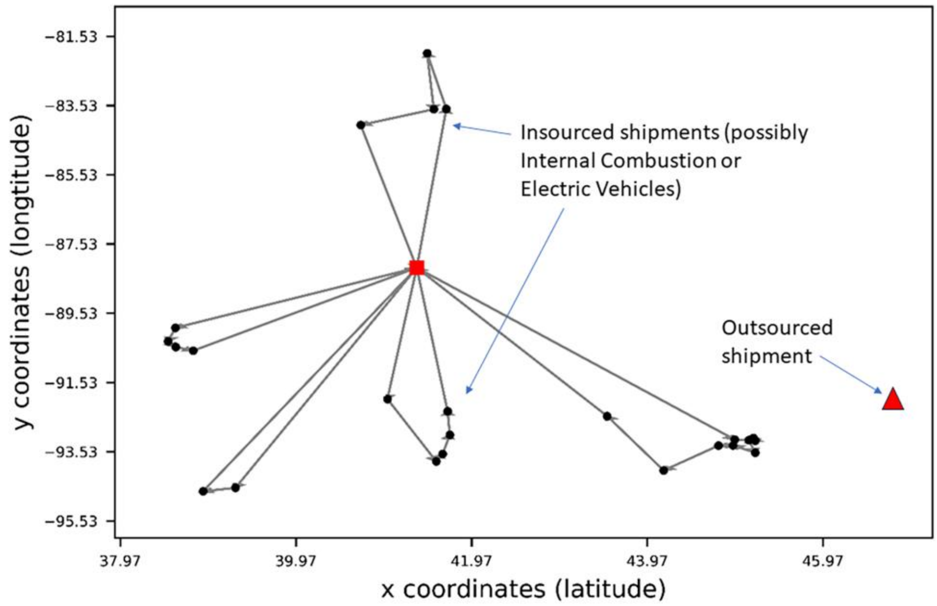
We next improved the initial D3TO freight optimization using RB-ACS with the same scope by combining the route allocation and outsourcing decisions in 2020. The RB-ACS innovation resulted in an additional 9.6% reduction in cost for the largest instance considered, which included 15,000 shipments (Dang et al. 2022). In Dang et al. (2022), we verified and validated the solution quality of RB-ACS using multiple small instances solved within 1% of the Gurobi best solution and achieved superior solution quality compared with tabu search for large instances.

Modeling and Solution Methods

The TNO suite develops the transportation strategies within DHL Supply Chain and helps analysts quickly make better-informed decisions to drive growth and achieve bottom-line savings. All TNO modules effectively solve the same overall problem to reduce costs while meeting demand. Because of the problem's size, all the problem's aspects and assumptions cannot be used in a single formulation. The freight optimization module solves the VRP, including approximate fleet sizing and outsourcing plans. The TNO fleet optimization (FlO) module refines the numbers of vehicles and permits the possibility of including electric vehicles. The PPO module includes the possibility of collecting items for third-party pickup at locations other than the depots for the subset of outsourced routes. Finally, the RTO module allows trucks traveling long distances to visit more than a single depot. We provide details in the appendix. Figure 1 shows a solution in which the triangle on the far right is an outsourced customer or package.

Based on our literature review and DHL's expertise, we identified methods for finding an optimal or high-quality solution for variants of VRPs; these methods include commercial routing software, routing-specific software libraries, heuristic/meta-heuristic methods, and optimization routing software libraries; examples

Figure 1. (Color online) Optimal Solution Generated by TNO, Including 60 Packages, Electric Trucks (Connected Circles), and Outsourcing Option Shown as the Triangle on the Right Side



include Gurobi, ILOG CPLEX, and Google OR-Tools. To choose among these methods, we focused on their ability to address very large-scale problems (e.g., 600,000 packages versus 100 packages). Specifically, we used tailored ant colony optimization (ACO) algorithms that included elements of tabu and genetic algorithm searches. The innovative two-color (i.e., red-black) RB-ACS provides the main structure of our search algorithms, including fleet sizing, with dynamic programming (DP) to optimize the local solutions within neighboring routes. To refine the search performance of regular ACO methods, we adopted some optimization and data structure improvements. DP is one of the key optimization methods in TNO. DP treats the problem as an exponential tree and recursively applies the same reasoning to each condensed subproblem (leaves) to solve the polynomial-sized graphs until reaching the optimum (Cormen et al. 2009).

For PPO, we also developed a custom search using a custom genetic algorithm with a local improvement heuristic to improve the solution. The PPO module identifies near-optimal opportunities to use pool points for a given set of shipments by evaluating the direct LTL shipment costs against costs from predefined pooled lanes. The current process requires analysts to identify fixed pool regions before making any pool point decisions, where a fixed pool region is the pool point to which a shipment is routed if the shipment is to be delivered in a specific geographical region. A software tool is then used to calculate the linehaul TL cost from the origins to the preselected pool points. Finally, the LTL costs from the pool points are added and the software tool determines if using pool points are less expensive than direct LTL shipping within the given period.

For RTO, an integer program is solved using a combination of heuristics and Google OR tools. The RTO module considers parameters such as, the maximum number of stops on the round-trip; the maximum waiting days before the next TL trip; the maximum empty miles between stops; mileage cost and cost-per-delivery cost charged by the carrier; fixed carrier cost per day; cost at each layover; and whether any shift in the base schedule is cost effective during the relevant time period. The RTO module generates cost-saving opportunities for cross-regional TL moves. In essence, the RTO module matches shipments in the geographical area of origin and shipments in the geographical area of destinations. The RTO module is typically used to improve on the schedules derived by other modules that leave empty return trips.

TNO Implementation

In this section, we discuss the TNO suite implementation in more detail. We first briefly review the inputs required and the preprocessing results we generated using the available DHL database (Dang et al. 2021). We then discuss the lessons learned to guide the DHL stakeholders. In addition, we describe the phases in which we improved the TNO algorithms and how these phases are linked to each other in a typical implementation. Finally, we discuss error handling and continuous performance improvement.

TNO Inputs and Preprocessing Efforts

When analysts apply TNO they need to enter data about the client' network and related challenges. The analysts define their scope of interest to be a set of depots in close or sparse geographic areas. They may also define the mix of shipment classes and opportunities to split oversized shipments using consolidation hubs. The ability to change settings may result in different preprocessing options and constraints in the model. Once we have defined the problem, before running the optimization model, it is necessary to select the representative data and transform the initial data into the form that the tool requires as input. Missing values are imputed, and historical data are cherry-picked to estimate future demands (e.g., peak and off-season, statistical tests).

Figure 2 shows a summarized flowchart of the complete process beginning with the extraction of data from DHL transportation network systems.

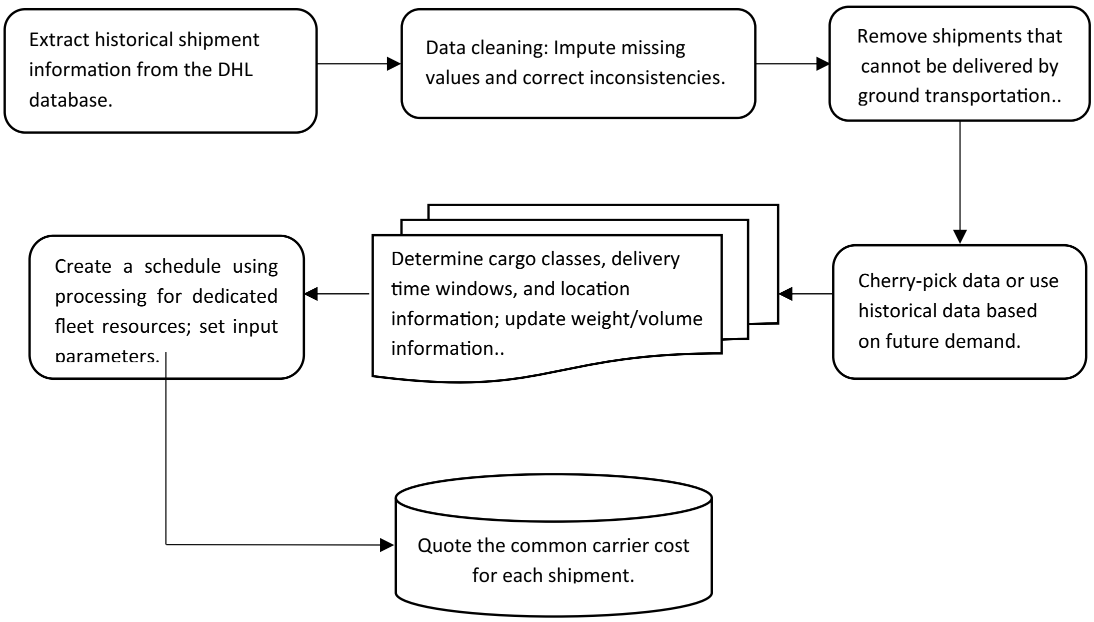
In practice, customers have different delivery requests, which result in different sets of time window requirements. Routes generally have the same capacity on every arc along the trip because most of the trucks are the same size.

Development of TNO

The development and testing of TNO algorithms involved four main phases corresponding to the four key software modules. By releasing the software in phases, the bidding and implementation teams could access the modules as they were developed. The core idea for freight optimization and fleet optimization is the determination of make-buy decisions inside the core ant colony iteration step, effectively creating multiple symbiotic ant colonies. This occurs by comparing route and common carrier costs (Figure 3). In fleet optimization, there is also the potential to consider other modes including electric vehicles, as we indicate in Figure 4. This figure also demonstrates the importance of using outsourcing; deliveries may be so distant from a depot that using any type of dedicated fleet option is not cost effective.

The development of TNO algorithms continues to be a process involving error handling and performance improvement. With repeated tests and assessments by a DHL Supply Chain user, we evaluate the drawbacks of the model, refine them, and eliminate poor solutions (e.g., solutions that require a significant amount of expensive outsourcing). The tool also facilitates reporting and

Figure 2. Preprocessing Data Steps Used in DHL Supply Chain North America's Previous Manual Process and TNO (Dang et al. 2021)

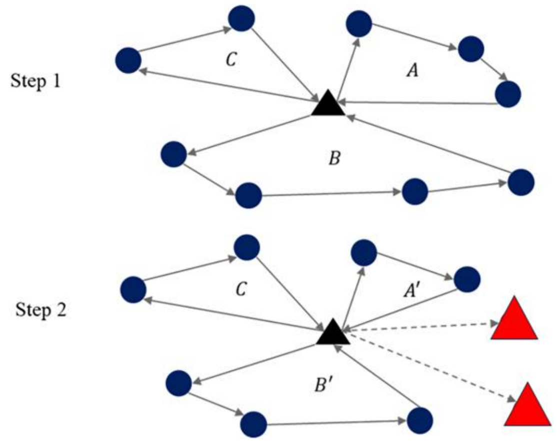


feedback by providing error messages to users who make mistakes in using the tool.

Challenges of Integrating TNO in the Bidding Process

Another important aspect of TNO's implementation at DHL Supply Chain continues to be output processing.

Figure 3. (Color online) Iterations of TNO's Freight Optimization Tool, Which Determines Make-Buy Decisions (Dang et al. 2021)



Notes. This figure shows part of an iteration of a red-black ant colony search. At the beginning all the nodes are in the black network and the assignments are generally reasonable, likely having benefited previously from local improvements using dynamic programming. In the next step, the possibility of outsourcing two nearby deliveries is identified, and the black nodes are replaced with triangular (red) nodes. Similarly, the red colony is revised to include these nodes and the metaheuristic continues.

The output from a previous phase is generally the input to a subsequent phase as planned outsourced shipments are reconsidered. While working on the TNO implementation, we received enhancement suggestions internally. Specifically, the analysts suggested the need for fleet sizing, pool points, and round trips. We collected feedback by conducting numerous interdepartmental meetings, analysts' interviews, and surveys. The TNO team then studied the standard workflows of the bidding projects, reworked the output formats, and updated the cost decomposition formulae. Furthermore, some analysts commented that inexperienced users who were not familiar with TNO might have limited knowledge of how to avoid errors. To correct this problem, we wrote internal TNO user manuals and are continuing to revise these manuals for ease of understanding.

Another challenge we identified relates to education. When DHL Supply Chain attempted to replicate results in parts of the world outside of North America, we encountered delays because qualified well-trained analysts took longer to identify than we anticipated. This highlights the role of The Ohio State University both as a partner in innovations and also as trainers of many DHL analysts on whom we continue to rely to achieve savings and win contracts.

Current Process with TNO and Quantified Time Savings

The steps we now follow after the TNO implementation are presented in Table 3. The steps used most frequently in a given project are freight optimization and RTO; therefore, we marked the other steps as optional.

Figure 4. (Color online) TNO Implementation Timeline Overview

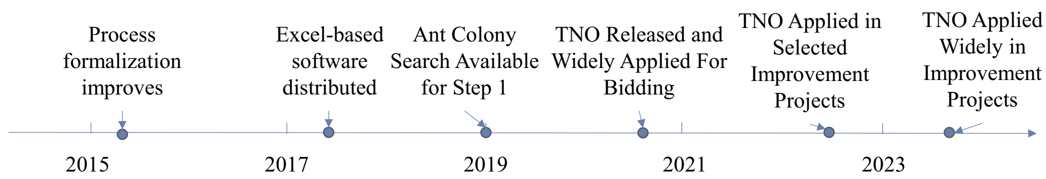


Figure 4 shows an overall timeline of the TNO implementation. Overall, TNO automates and improves all the previous manual processes by supporting an integrated bidding and system improvement process with clear metrics. It evaluates multiple solutions using a novel ACO-based hybrid algorithm and genetic algorithms and generates a better solution than the previous manual process prior to 2019, which took approximately 120 hours over a period of up to two weeks (Table 1). The TNO built-in algorithm iteratively searches across billions of potential solutions so that analysts can spend more time on visualizing the solutions, cost modeling, and preparing bidding proposals. Based on our survey of the transportation team at DHL Supply Chain North America, we know that transportation analysts work on more than 100 projects annually. Currently, TNO is used in virtually all these projects, especially those on cost estimation and bidding processes.

Validation and Savings Estimation

In this section, we present our estimation of the savings resulting from the TNO implementation from initial planning through the bidding process and improving existing processes. The estimates of savings for each project identified here generally leverage the existing process established through contracts with the customers, which relate to fuel, driver, and truck savings. The savings from increasing the win percentage is only indirectly considered because if DHL Supply Chain North America did not win the projects, these savings would not have been achieved.

Savings Through the Bidding Process

The bidding team served as our close partner during TNO development. Therefore, bidding team analysts, having provided initial motivation and feedback throughout the development process, implemented the process first. The team easily accepted the process because it was like the processes that they had previously used but was supported with advanced analytics via TNO software. We begin by defining some notation:

s_B: Yearly savings effectively achieved through planning using TNO in the bidding process and achieved for DHL customers on bids that we won.

s_O: Yearly savings achieved by process improvements for existing contracts.

p_F: Fraction of the existing projects that applied TNO in the past year.

In particular, the fraction of the sales that DHL Supply Chain North America would have obtained without the TNO implementation is difficult to determine. The market is increasingly competitive and other companies are improving their analytical capabilities. Yet, DHL Supply Chain North America has increased its sales by an estimated 20% in only two years as part of a longer-term growth in which sales increased from 15% to 60%. Between 2010 and 2023, DHL Supply Chain North America's revenues have more than doubled. It is also difficult to delineate when TNO was first widely adopted in the bidding process. We estimate $\tau = 2.5$ years of operation. Our early work was a somewhat straightforward implementation of ant colony search with availability as early as 2018 (Dang et al. 2021). In 2022, we bid for the projects shown in Table 4 using TNO and achieved the estimated savings shown. These results, which are approximate, were achieved for each of the past $\tau = 2.5$ years, and similar results can be expected going forward, leading to an estimate of $s_B =$ \$38.6M annually. Therefore, Table 4 represents the 2022 projects known to have fully used TNO at bidding

Table 3. Procedures DHL Supply Chain North America Followed After TNO Implementation in 2020, Including Average Analyst Bid Preparation Times

Steps	TNO solution procedures	Average time (hr)
1.	Extract and clean raw shipments information from the database.	8
2.	Preprocess the data, sample the subsets, and make initial transportation mode decisions using human judgment and specified constraints.	16
3.	Determine initial outsourcing decisions using freight optimization.	8
4	(Optional) Create recommendations for fleet size changes.	8
5	(Optional) Improve pickups for third parties using pool point optimization (PPO).	8
6	Determine additional routings and consolidations using round-trip optimization (RTO).	8
7	Present the solutions and proposals to the customers for the bid.	8

Table 4. New Projects Started in 2022 Using the Current TNO Suite, Contract Length of Each Project, Estimated Savings from TNO, and Average Bid Preparation Time

New business award	Annual freight spend ($M)	Contract length (yr)	TNO savings (%)	Preparation hours
Aerospace Manufacturer NorAm LLP	112.10	5	5.0	42
Battery Retailer LLP Renewal	4.80	3	3.4	9
CPG Spinoff Network Management	18.84	3	8.2	20
Confectioner Canada LLP	18.68	5	10.7	24
Tier 2 Auto Supplier LLP	17.29	5	12.4	14
Pharma TMS RFP	47.95	5	5.6	19
Pharma Network Consolidation	32.91	5	6.9	21
Automotive Manufacturer LLP	26.90	5	11.2	18
Tier 1 Auto Supplier NAFTA LLP	89.80	5	10.2	29
Auto Aftermarket Supplier LLP	45.67	3	6.9	17
Ecom Tire Retailer LLP	39.45	3	15.8	16
Pharma LLP	7.21	5	9.4	15
Totals	461.6			244

time. We won each bid, and the lower costs reflect the savings (from reduced fuel, driver, capital, and outsourcing costs) built into the plan.

Savings Through Improvements on Existing Processes

Approximately one year after the implementation of TNO for bidding, the operations group started using TNO for operational improvements and achieved monetary savings on the 19 recent projects relating to seven customers in 2022 that involved active DHL Supply Chain management (Table 5). In Table 5, we list each module implementation on a separate line and show the estimated savings. We anticipate achieving similar results on an expected 60 projects annually by the middle of 2024. We estimate that s_O = $60M for each year when rolled out to 60 projects per year.

Combining all the past savings, we have

$$\text{Past Savings} = \tau s_B + p_F s_O = 2.5 \times \$38.6\text{M} + \$20\text{M} \cong \$117\text{M}. \tag{1}$$

Approximately 0.1 megatons of CO_2 savings from the reduction in fuel consumption are associated with these savings. Moving forward (after mid-2024 and assuming 60 improvement projects per year), the savings rate estimate range is

$$\text{Yearly Savings} = s_B + s_O = \frac{\$98.6\text{M}}{\text{year}}. \tag{2}$$

These are the estimated savings realized for DHL customers through reduced fuel, driver, and other costs. The amount added to the DHL Supply Chain North America bottom line is a different calculation. This relates mainly to the increase in win percentage from approximately 15% in 2015 to approximately 60% in 2022. Many factors, including the pandemic, influenced revenues; however, revenues have increased more than 50% since 2015 to more than $5B per year. Although we cannot disclose internal profit numbers, the savings are significant. Being able to propose efficient solutions at bidding time is critical to winning and maintaining a contract. The incremental TNO savings in money and time (Table 4) compared with the period before 2020 are somewhat small (e.g., only around 15% better solutions as measured by objective values and through conservative analyses). Yet, they are important for winning the right contracts. Moreover, the ease of using TNO provides major efficiency gains for retaining existing customers. Without the ease of training and implementation of optimized solutions from TNO, DHL analyst user reports suggest that few (if any) of these operational benefits would have been achieved.

Managerial Insights and Business Impact

The biggest insight we gained from the TNO implementation was that being able to quickly generate

Table 5. Estimated Savings for Existing Projects in Reduced Fuel, Personnel, and Other Costs from Operational Improvements to These Projects Using the Current TNO Suite

Customer	Sector	Estimated savings
Customer 1	Engineering/manufacturing	$760,000
Customer 1	Engineering/manufacturing	$675,660
Customer 1	Engineering/manufacturing	$126,000
Customer 1	Engineering/manufacturing	$52,800
Customer 2	Engineering/manufacturing	$4,000,000
Customer 3	Consumer	$393,141
Customer 4	Consumer	$612,000
Customer 4	Consumer	$2,674,520
Customer 4	Consumer	$845,550
Customer 5	Consumer	$977,199
Customer 5	Consumer	$161,850
Customer 6	Engineering/manufacturing	$300,000
Customer 6	Engineering/manufacturing	$102,732
Customer 6	Engineering/manufacturing	$82,330
Customer 6	Engineering/manufacturing	$147,879
Customer 6	Engineering/manufacturing	$50,089
Customer 7	Retail electronics	$1,100,000
Customer 7	Retail electronics	$6,139,432
Customer 7	Retail electronics	$974,604
	Total	$20,175,786

plausible delivery bids provided a major edge in the bidding process. DHL Supply Chain North America's win rates have increased from approximately 15% to 60% with more than 20% generated after 2020. Customers were often pleased that DHL Supply Chain North America could not only generate a bid quickly but also analyze many scenarios quickly and accurately.

Before the implementation of TNO, we held many rounds of discussions with the project managers and the transportation analysts. Their feedback confirmed that TNO could quickly help them determine cost-saving potential and efficient options relating to outsourcing (i.e., dedicated fleet and/or common carriers). Typically, a bidding project may require the analyst to spend a few weeks or even months preparing the proposal; transportation mode optimization occupies at least half of this time.

Culture and cultural change are also important. TNO has truly enabled DHL Supply Chain North America to become even more of a nimble, high-technology organization. We can train workers worldwide much faster, offer our customers innovative features related to analyzing scenarios, and test fleet makeup in ways that were previously impossible. There is also an increased regard for the value of partnering with and benefiting from elite research institutions such as The Ohio State University. The Ohio State University collaborators helped to provide and train the workforce needed to develop TNO.

Conclusions

In this paper, we discuss a business problem encountered by DHL Supply Chain North America, which routinely operates large and complicated transportation networks. We describe the previous manual processes for generating bids and improving existing project supply chain designs and associated weaknesses. We present a tailored high-performance TNO suite that we designed to solve several large-scale NP-hard problems. We estimate that the implementation of this algorithm in the DHL Supply Chain North America has saved in excess of \$117M over 2.5 years since 2020 for DHL and its customers.

The optimization models were developed in collaboration with Ohio State University. TNO is currently in use for planning shipment allocations, vehicle routes, and purchases of third-party logistics. The methods are contributing to millions of dollars in added profits for DHL Supply Chain and its customers by saving personnel, fuel, and third-party costs annually in North America. An extension of this work to other branches of DHL is planned to be the next step. The software is quite adaptable and is already being used in South America and India. In the meantime, we are developing and implementing another improved version of TNO to deal more accurately with heterogeneous fleet sizes. We are also evaluating combining methods such as outsourcing with relocation and round-trips to achieve even higher levels of solution quality.

Appendix. Models and Methods

Freight Optimization

In this section, we present our formulation for the vehicle routing problem with time regulations and common carriers (VRPTRCC). We use the following notations for the problem parameters and variables. The overall formulation selected is a type of deterministic integer program. We had briefly considered addressing demand uncertainty but focused on deterministic modeling because of the repeatability of the demands involved and the large scale of the problem. Specifically, we formulate the problem with an "arc-based model" because the compact formulations with vehicle indices are suitable for most VRP cases (Adulyasak et al. 2015). If the vehicle routes were predefined or there were fewer possibilities to detour to other clusters, we might have selected set-partition formulations.

Sets

I: Deliveries or packages from depot d, where $i \in I = \{I_1, I_2, \ldots, I_d\}$.

I_0: Nodes from a particular depot, which includes the customer set I and the depot 0.

V: Vehicles or trucks from depot d, where $k \in V = R = \{1, 2, \ldots, V_d\}$.

Parameters

e_i: Earliest service time at node i, which is the earliest acceptable delivery time.

A_i: Latest service time at node I, which is defined as the latest scheduled delivery time.

c_{ij}: Distance cost matrix from nodes i to j (i.e., the distance multiplied by the cost per mile).

λ_i: Rated common carrier cost for shipment i.

f: Fixed cost per day by using vehicle k.

p: Cost per stop along the route.

d^{max}: Maximum allowable distance between internal nodes on the same route.

q: Capacity of the homogeneous fleet vehicles.

μ_i: Unloading time at customer node i.

b: Layover time (a fixed value of 10 hours).

δ: Maximum allowable working time per layover.

g: The estimated average speed of each vehicle k (a fixed value of 55 miles per hour).

t^{max}: Maximum allowable time of duration for one route.

w_i: Demand at node i.

Variables

s_{ijk}: An integer variable indicating the service time of node j from i on vehicle k.

x_{ijk}: A binary variable indicating whether node i immediately precedes j by vehicle k.

y_i: A binary variable denoting whether (or not) shipment i is assigned to a common carrier.

l_{jk}: A binary variable indicating vehicle k taking a layover at or immediately before node j.

z_k: A binary variable indicating whether (or not) vehicle k needs to have the first layover.

r_k: A binary variable indicating whether (or not) vehicle k needs the second layover.

t_k: Total time duration (in hours) of vehicle v on the route.

u_i: Product weights accumulated in a vehicle up to node i (an integer variable).

θ_k: Total number of days elapsed on route k.

Objective Function (A.1) states that we want to minimize the overall cost of transporting goods. Specifically, the cost includes the fixed vehicle cost on each route ($f \cdot \theta_k$), the summation of the mileage cost, and the stop cost along each arc (i, j) by each vehicle k, $\sum_{i \in I_0} \sum_{j \in I_0} x_{ijk} \cdot (c_{ij} + p)$. Because this term involves the cost of one extra stop charged at the depot, we need to subtract $\sum_{i \in I_0} x_{i0k} \cdot p$. Next, we sum the previous costs over the dedicated fleet and add in the cost of shipments from the common carriers $\sum_{i \in I} \lambda_i \cdot y_i$.

The first group of constraints are related to the flow degrees in the network. Constraints (A.2) ensure that the flows that enter in node j by vehicle k should be equal the flows leaving from j by vehicle k. Constraints (A.3) state that the flow within the same node is invalid (i.e., $x_{iik} = 0$). Constraints (A.4) ensure that the package of each customer i should be shipped by exactly one mode—either on a dedicated fleet or by being assigned to a common carrier. Constraints (A.5) specify that each vehicle should be used at most once (i.e., $\sum_{j \in I_0} x_{0jk} \le 1$). The second group of constraints are related to the time window constraints. The service time of each customer j by vehicle k from its potential predecessor i (s_{ijk}) is constrained with the time window of customer j multiplied by x_{ijk}. If i does not precede j, then $x_{ijk} = 0$ and the service time equals zero. This concept is specified in Constraints (A.6). Constraints (A.7) are tighter for the service time s_{ijk} because its value should be less than or equal to the service time at node i ($\sum_{h \in I_0} s_{hik}$) plus the time spent traveling from node i to node j $\left(\frac{d_{ij}}{g}\right)$, the unloading time of customer i (μ_i), and the layover time, if node i is the immediate predecessor of j. The term d_{ij} is the distance between i and j, whereas g is the estimated average speed of the vehicle. The term $b \cdot l_{jk}$ specifies whether the driver should take a layover before or when reaching customer j. If $l_{jk} = 1$, then the driver of vehicle k will take a 10-hour layover immediately and continue the drive at the end of the layover. Furthermore, if i and j are not connected, the service time variable s_{ijk} is zero, which is forced by Constraints (A.6). Constraints (A.8) state the same concept as Constraints (A.7) in a reverse direction. That is, if i and j are connected, Constraints (A.7) and (A.8) will give us the actual service time at customer j as $\sum_{h \in I_0} s_{hik} + x_{ijk} \cdot \left(\frac{d_{ij}}{g} + \mu_i\right) + b \cdot l_{jk}$. Otherwise, the term $(1 - x_{ijk})$ is one, and a "Big M" (a common approach to efficiently address inequalities in the context of integer and linear programming) is multiplied to relax this inequality. To accelerate the speed of the MILP solver, the Big M value should be tight; here we use the latest delivery time among the deliveries (in hours) as M. Next, we use Constraints (A.9)–(A.10) to check whether vehicle k's driver should take layovers on the trip. If taking the first layover is necessary for driver k (i.e., $z_k = 1$), the total working time, $\sum_{i \in I_0} \sum_{j \in I_0} x_{ijk} \cdot \left(\frac{d_{ij}}{g} + \mu_i\right)$, must be greater than δ.

Whenever the second layover is required, the first layover should be applied ($z_k = 1$) and the total working time on vehicle k, $\sum_{i \in I_0} \sum_{j \in I_0} x_{ijk} \cdot \left(\frac{d_{ij}}{g} + \mu_i\right)$, must be greater than twice δ, which is 28 hours. In that case, $z_k = 1$ and $r_k = 1$, and Constraints (A.9) and Constraints (A.10) are in effect. In Constraints (A.11), we compute the number of layovers required on vehicle k ($r_k + z_k$) and let the value equal $\sum_{j \in I_0} l_{jk}$. This inequality assigns the layovers to each node j if it is on vehicle k. Constraints (A.12) state that the value of the binary variable l_{jk} should be at most $\sum_{i \in I} x_{ijk}$. Constraints (A.12) and (A.13) ensure the layovers are assigned to the earliest node j on vehicle k; otherwise, if j is not on k, then $l_{jk} = 0$. Constraints (A.13) and (A.14) are weight capacity constraints, which eliminate the infeasible routes that exceed the capacity of the trucks. On the one hand, we restrict the upper bound of weight up to node i. If i is directly preceded by the depot, then u_i should be less than or equal to the demand w_i. Also, every demand should be satisfied. Then, we conclude $u_i = w_i$. On the other hand, if node i is not preceded by the depot 0, then u_i is constrained by q, and $\sum_{k \in V} x_{0ik} = 0$. Constraints (A.14) calculate the accumulated weights up to u_i by the summation of its demand w_i and the accumulated weights from its immediate predecessors by vehicle k, $\sum_{j \in I} x_{jik}$. These two inequalities also ensure there must be an arc going out from the depot; otherwise, there will be an infeasible inner loop (subtour). We have proved the combo of subtour elimination Constraints (A.13) and (A.14) to be a valid inequality.

Next, we want to know the total time elapsed t_k on the vehicle k (travel + unloading + layovers) to check if the travel time limit is violated and determine the days on which vehicle k has been used. The total time elapsed on vehicle k is evaluated by Constraints (A.15) where the value should be greater or equal to the return time of vehicle k to the depot ($\sum_{i \in I_0} s_{i0k}$) minus the serving time of this vehicle at the first stop ($\sum_{j \in I_0} s_{0jk}$) plus the time taken from the depot to the first stop $\left(\sum_{j \in I_0} x_{0jk} \cdot \left(\frac{d_{0j}}{g}\right)\right)$. Constraints (A.15) also set the upper time limit for each vehicle. If the limit is violated, the route is not feasible. We then divide t_k by 24 hours to obtain the number of days, θ_k, of using vehicle k. This value is an integer, which is the roundup of $\frac{t_k}{24}$. For any arc (i, j) to be feasible, the distance (d_{ij}) should be within the intra-node distance threshold d^{max}. Constraints (A.18) break the symmetric solutions caused by the homogeneous fleet size. Finally, integrality Constraints (A.19) and nonnegativity Constraints (A.20) are given.

Fleet Optimization

We provide additional details regarding the fleet optimization (FlO) program functionality and operation here. The FlO program operates with the many of the same

Table A.1. MILP Formulation of the VRPTRCC

Type		Equation number
Objective function	$$Min \sum_{k\in V}\left\{f\cdot\theta_k+\sum_{i\in I_0}\sum_{j\in I_0}x_{ijk}\cdot(c_{ij}+p)-\sum_{i\in I_0}x_{i0k}\cdot p\right\}+\sum_{i\in I}\lambda_i\cdot y_i$$	(A.1)
Subject to:		
Degree constraints	$$\sum_{i\in I_0}x_{ijk}=\sum_{h\in I_0}x_{jhk},\quad \forall k\in V,\quad \forall j\in I_0$$	(A.2)
	$$X_{iik}=0,\quad \forall i\in I_0,\ \forall k\in V$$	(A.3)
	$$\sum_{k\in V}\sum_{j\in I_0}x_{ijk}+y_i=1,\quad \forall i\in I$$	(A.4)
	$$\sum_{j\in I_0}x_{0jk}\le 1,\quad \forall k\in V$$	(A.5)
Time window constraints	$$a_j\cdot x_{ijk}\ge s_{ijk}\ge e_j\cdot x_{ijk},\quad \forall i\in I_0,\ \forall j\in I,\ \forall k\in V$$	(A.6)
	$$s_{ijk}\le\sum_{h\in I_0}s_{hik}+x_{ijk}\cdot\left(\frac{d_{ij}}{g}+m_i\right)+b\cdot l_{jk},\ \forall i\in I,\ \forall j\in I_0,\ \forall k\in V$$	(A.7)
	$$s_{ijk}\ge\sum_{h\in I_0}s_{hik}+x_{ijk}\cdot\left(\frac{d_{ij}}{g}+m_i\right)+b\cdot l_{jk}-(1-x_{ijk})\cdot M,\ \forall i\in I,\ \forall j\in I_0,\ \forall k\in V$$	(A.8)
Layovers constraints	$$\sum_{i\in I_0}\sum_{j\in I_0}x_{ijk}\cdot\left(\frac{d_{ij}}{g}+\mu_i\right)\le\delta+2\cdot\delta\cdot z_k,\quad \forall k\in V$$	(A.9)
	$$\sum_{i\in I_0}\sum_{j\in I_0}x_{ijk}\cdot\left(\frac{d_{ij}}{g}+\mu_i\right)\le 2\cdot\delta+\delta\cdot r_k,\quad \forall k\in V$$	(A.10)
	$$r_k+z_k=\sum_{j\in I_0}l_{jk},\ \forall k\in V$$	(A.11)
	$$l_{jk}\le\sum_{i\in I_0}x_{ijk},\ \forall j\in I_0,\ \forall k\in V$$	(A.12)
Truck capacity constraints	$$q+(w_i-q)\cdot\sum_{k\in V}x_{0ik}\ge u_i\ge w_i,\quad \forall i\in I$$	(A.13)
	$$u_i-u_j+q\cdot\left(\sum_{k\in V}x_{ijk}\right)\le q-w_j,\ \forall i,j\in I$$	(A.14)
Maximum travel time constraints	$$t^{max}\ge t_k\ge\sum_{i\in I_0}s_{i0k}-\sum_{j\in I_0}s_{0jk}+\sum_{j\in I_0}x_{0jk}\cdot\left(\frac{d_{0j}}{g}\right),\ \forall k\in V$$	(A.15)
	$$\theta_k\ge\frac{t_k}{24},\ \forall k\in V$$	(A.16)

Table A.1. (Continued)

Type		Equation number
Intranode distance constraints	$d_{ij} \cdot x_{ijk} \le d^{max},\ \forall i,\ j \in I_0,\ \forall k \in V$	(A.17)
Symmetry breaking inequalities	$t_k \ge t_{k+1},\ \forall k \in V$	(A.18)
Integrality and nonnegativity constraints	$x_{ijk},\ y_i,\ l_{ik},\ r_k,\ z_k \in \{0,1\},\ y_0 = 0,\ \forall i \in I_0,\ \forall j \in I_0,\ \forall k \in V$	(A.19)
	$t_k \ge 0,\ \theta_k \in Z_+,\ s_{ijk} \ge 0,\ u_i \ge 0,\ \ \forall i \in I,\ \forall j \in I_0,\ \forall k \in V$	(A.20)

parameters and variables used by the exemplary FlO, which are summarized in Table A.1. Additional parameters and variables associated with the FlO program are shown in Table A.2.

In accordance with the black transition rules, at each node i, potential next moves are evaluated, and decisions are made according to probability distributions. Here, τ_{ij} is a number, which is called a "pheromone concentration" on edge (i,j) and which is equal to the amount of pheromone accumulated on the path between the current node i and a possible move j. The visibility value η_{ij} is regarded as the short-term possibility to serve a customer j, which is expressed by $\eta_{ij} = d_{ij}^{-1}$ and which is the inverse of the length of edge (i,j). The decision about which customer to serve next depends on the short-term visibility value η_{ij} and the long-term pheromone value τ_{ij}. The black transition rule is the transition rule in the FrO program. If at any step a black ant decides to explore a new path $(q > q_0)$, it selects the most attractive customer j to visit according to a simplified probability distribution. The FlO program uses the same local updating rule and global updating rule as the FrO program.

An exemplary RB-ACS consists of two interchangeable pheromone trails. Although the black ants follow their transition rules to find high-quality routes (black), red ants search the potential direct LTL deliveries (red) and try to find cost-efficient closed loop routes. At each customer i, the RB-ACS first determines whether i is labeled red or black. The colors of the deliveries depend on the color of the ants (i.e., routes). In addition, each customer i can shift its color once the color of a route changes. If i is labeled as black, then it will follow the black transition rules (Dang et al. 2021). This rule allows red ants to exploit the most attractive black nodes and to explore better routes from other red nodes according using

$$\widehat{r_{ij}} = \begin{cases} e^{-\frac{d_{ij}}{m_j}}, & \text{if } \Delta(i,j) < \varepsilon \cdot m(i,j),\ j \in V_1 \\ 0, & \text{otherwise.} \end{cases} \tag{A.21}$$

Assume the red node i is the current stop and the red ant wants to pick the successor customer j. The RB-ACS creates a sample of U[0,1] distributed random variable and denotes the obtained value by $\tilde{t}$. When $\tilde{t} \ge t$, then j from the black nodes is exploited by this ant. When $\tilde{t} < t$, then j is selected from the red nodes according to the following discrete distribution with the probability $\widehat{r_{ij}}$ given by Equation (A.10). The threshold value t is called the red-black exploration rate. The constant red-black penalty parameter (ε) is set to allow a few expensive closed-loop routes to exist for the local improvement heuristics.

The local updating rule for the red pheromone trails is then applied. After a single iteration is finished, a global updating rule is performed on the paths connected to the red node $i \in V_1$, which is given by

$$\widehat{r_{ij}} = \widehat{r_{ij}}^{red} = \begin{cases} e^{-\frac{d_{ij}}{m_j}}, & \text{if } (i,j) \in \{L_{global}\},\ j \in V_1 \\ \tau_0, & \text{otherwise,} \end{cases} \tag{A.22}$$

$$\widehat{r_{ji}} = \widehat{r_{ji}}^{red} = \tau_0. \tag{A.23}$$

Therefore, after each single iteration, the red routes from the global best solution L_{global} are intensified and the other arcs that flow in and out are downgraded to their initial pheromone values.

Table A.2. Additional Parameters That the FlO Program Uses

Parameters	Descriptions
$\widehat{r_{ij}}$	Probability of choosing red node j from the current red node i
$\Delta(i,j)$	Updated closed-loop route cost found by adding arc (i,j) to the current route
ε	Constant red-black penalty parameter
$m(i,j)$	Total direct LTL cost after adding nodes i and j to the current route
V_0	Set of black deliveries in the current iteration
V_1	Set of red deliveries in the current iteration
S_{max}	The maximum number of solutions in each iteration
s_k	A closed-loop route, which is operated by fleet k

The red-black ant search procedure is represented here.

FlO program (single iteration): Ant search procedure in a single iteration

Result: an updated route: $s = (0, i_1, i_2, \ldots, i, 0)$
Input:
Current partial route $(0, i_1, i_2, \ldots, i)$, candidate list for customer i: $\Lambda(i)$;
While closed loop route $(0, i_1, i_2, \ldots, i, 0)$ feasible do
$q_0 = q_{0s}$;
if $\Lambda(i) = \emptyset$ then
complete the current route s;
break
else
* The ant decides to do exploration *
if $q > q_0$ then
if $i \in V_1$ then
* decide whether to explore from the red deliveries or the black deliveries *
if $t \le \tilde{t}$ then
explore j from the tabu list s.t. $j \in V_1$ and $j \in \Lambda(i)$;
pick $j = \arg\max_{j \in \Lambda(i),\, j \in V_1} \widehat{r_{ij}}$;
else
explore j from the tabu list s.t. $j \in V_0$ and $j \in \Lambda(i)$;
end
else
* Exploitation – no matter which color i is labeled *
Exploit j from the tabu list s.t. $j \in \Lambda(i)$;
end
add j after i to the current partial route;
update the node indices $(i \leftarrow j)$ and form a new closed loop route s;
update the route information – weight, time, distance, fleet costs, layover, tabu lists, etc.
end
end while

The FlO program adopts mutation and permutation heuristics to improve the algorithmic performance. The permutation procedure associated with the FlO program may be the same as we describe previously with respect to the *FlO program (single iteration): Ant search procedure in a single iteration* process. In this case, the only difference is the input and the output; because the FlO program is concerned with closed-loop routing, the current partial route is $s_k = (0, i_1, i_2, \ldots, i, 0)$, which includes a backhaul arc to the origin. A feasibility check evaluates the efficacy of the backhaul arc, and the cost $z(s_k)$ considers the backhaul cost and varies by different fleet sizes. Similarly, the output considers the closed loop routing, which gives $z(s_k) \leftarrow \min\{f(0,0), z(s_k)\}; (0, i'_1, \ldots, i'_n, i, 0)$.

For long routes (typically more than 20 nodes), the FlO program adopts a rolling-window to do the permutation. The concept of this heuristic is to fix the length of the partial sequence (w) and roll w over the kth route. For each w, the FlO program calls our DP (permutation) procedure and combines all the instances of w to form a new sequence. If the new sequence is feasible, it means the new sequence is better than the original sequence, and the FlO program assigns this solution to the kth ant.

Table A.3. Fleet Size Pheromone Matrix

Deliveries	Truck I_0	Truck II_0	Truck III_0	Max indicator
1	50	0	0	I_0
2	2	45	3	II_0
3	39	11	0	I_0
4	0	0	50	III_0
...	...	...	...	...
$\lvert I \rvert$	3	5	42	III_0

Note. These numbers build up during the iterations and help guide inclusion decisions about specific trucks in future ACS iterations.

A completed solution (a colony) is generated in a given iteration after all the ants have finished their tours. The mutation heuristic can then be applied to help the RB-ACS reach better solutions in the search space by randomly mutating the routes and, hence, producing a new colony that is better but not far from the original colony. In this operation, every ant is regarded as a black ant and the RB-ACS tries to improve the solution. The steps for the mutation heuristic in the RB-ACS are the same as in the single iteration FlO procedure. The mutation and permutation probability at the iteration k is defined in Dang et al. (2021).

The FlO program performs a fleet size reduction just before the completion of each iteration. Before the fleet size reduction, all the closed-loop routes are built with the largest available vehicle type, and the reduction operation is conducted for every solution generated in the iteration.

Next, the RB-ACS collects the iteration best solution $z(A)$ and the global best solution $z^*(A)$. A fleet pheromone matrix is created to accumulate the fleet type assigned by these two solutions. For instance, given three types of available fleet types, I_0, II_0, and III_0, shipment i can be allocated to II by $z^*(A)$ and allocated to I by $z(A)$ in the kth iteration. In this case, the matrix is added by one at the position of [i, I_0] and [i, II_0]. Moreover, to prevent ties, the algorithm chooses the iteration number N that cannot be divided by the number of available fleet types $|V|$.

After the specified iterations, the maximum counts over the rows of the matrix are the indicators of the optimized fleet sizes for the shipments. Table A.3 presents an exemplary fleet pheromone matrix, which represents 50 iterations and three fleet types.

With the assigned fleet sizes, RB-ACS continues its iterations for $|V|$ different clusters. Within each cluster, all the shipments will be fulfilled by the same fleet type. The main RB-ACS procedure contains $2N$ Iterations. Although the first N iterations are used to find the most appropriate fleet types for each shipment, the later N iterations are run separately for each fleet type for better convergence.

The RB-ACS procedure may be represented as follows:

FlO program: Red-black ant colony system procedure

Result: $z^*(A)$; optimal closed-loop routes and direct LTL assignments
Initialization:
Assign same amount of pheromone π_0 on each arc $(i, j) \in A$;
* initialize the network cost with the total direct LTL costs*
$z^*(A) \leftarrow \sum_{i \in I} m_i$
for $iteration = 1 \rightarrow N$ do
while $solution \le S_{max}$ do
* construct routes with the largest Fleet type for each solution *
while $i \in I$ *not visited* do
call the FlO single iteration procedure to build a complete route s;
if $length(s) \ge 1$ then
call FlO Program - Ant search procedure in a single iteration;
end
end while
call mutation procedures between routes in this solution;
fleet size reduction on each route s, update the cost and route information;
reset the deliveries to unvisited, move to next *solution*;
end while
update $z^*(A)$, $z(A)$ and the optimal closed-loop routes and direct LTL assignments;
label the black and red routes;
call global updating rule by Equation (A.21) for black routes;
call global updating rule by Equations (A.22) and (A.23) for red routes;
update the fleet pheromone matrix for each shipment by $z^*(A)$, $z(A)$;
end
divide the shipments into different clusters by their assigned fleet types V;
for $size = 1 \rightarrow |V|$ do
for $iteration = (N+1) \rightarrow 2N$ do
Repeat: the same procedures for the first N iterations;
End
* store the new best cost z^H*
let $z^H = z^H + z^*_{size}(A)$;
end
Comparison:
* update the optimal results if improved*
if $z^H \le z^*(A)$ then
$z^*(A) \leftarrow z^H$;
End

As we discuss previously, the FlO program aims to determine and output the lowest-cost cargo transportation solution by comparing costs between closed-loop route shipments and direct LTL shipments. To that end, the FlO program calculates the optimized cost for delivering each shipment by the following rule:

Suppose route A is the route to be considered, then
if *A is a closed − loop route (black)* :
for i in route A :

$$C_i = \frac{1}{2}\cdot\frac{w_i}{w_A}\cdot[C_A - (|A|-1)\cdot p] + \frac{1}{2}\cdot\frac{d_{0i}}{d_{0A}}\cdot[C_A - (|A|-1)\cdot p] + p \quad \text{(A.24)}$$

else :
for i in route A :

$$C_i = m_i. \quad \text{(A.25)}$$

This means that if route A is an insourced route (i.e., in-house by DHL personnel), the cost for each shipment i on the route is calculated by Equation (A.24). However, if route A is not cost efficient as a fleet route, each shipment i on the route is assigned to be a direct LTL shipment with the initial common carrier cost m_i, as reflected in Equation (A.25).

Pool Point Optimization

We next briefly explore some symmetry breaking rules in addition to the main symmetry breaking Constraint (A.18). Some trips with different indices can slow down a mixed-integer programming (MIP) solution by requiring the solver to explore many alternatives, equivalent solutions—so-called *symmetric solutions* (Sherali and Smith 2001). To avoid this issue, we rank paths as follows:

The costing functionality of an exemplary PPO program may be represented as follows:

PPO program: Costing functions

Input: shipment information: locations, assigned pool points, weights, shipment class
function LTLCost(i)
Step 1:
Given the six steps of weight (lbs) in the LTL tariff, [500, 1,000, 2,000, 5,000, 10,000, 20,000], find the min value x^* that is greater or equal to weight(i) and the max value y^* that is less than weight(i);
Step 2:
$cost = \min\{weight(i) * unit\ price(y^*),\ x^* * unit\ price(x^*)\}$;
Step 3:
compare the cost with the baseline cost;
$LTL\ cost(i) = max\{cost * class\ change\ multiplier * discount, min\ charge\}$;
return LTL cost(i)
function TLCost(origin, pool point)
$TL\ Cost = Linehaul\ cost + mileage * TL\ fuel\ surcharge\ percentage$;
Input: a solution z;
function cost_for_shipment(z)
for i in I:
if i is assigned to a pool point **then**
$cost\ for\ i = direct\ LTL\ cost\ for\ i$;
else
$cost\ for\ i = LTLCost(i) * (1 + LTL\ surcharge\ percent)$;
end
* add the allocated linehaul TL cost to the shipment i *
for k in pool point list:
if $weight\ assigned\ to\ k \neq 0$ **then**
get the number of trucks needed from origin to k;
$allocated\ TL\ cost\ for\ i = TL\ Cost(origin,\ pool\ point) * numberoftrucks * \frac{weight(i)}{weigh\ assigned\ to\ k}$;
label the truck ID for each shipment;
end

if i is assigned to a pool point **then**
cost for i+ = *allocated TL cost for i*;
end
return cost for i

The genetic procedure used in the first part of the PPO program is a probabilistic search, which imitates the process of natural selection and evolution to evolve a population of initial solutions. A solution stands for a complete choice of pool points. For instance, solution $A = [Columbus : 1, Boston : 0]$, which means that the pool point named Columbus is chosen (with value 1) and the pool point named Boston is not chosen (with value 0). An exemplary genetic procedure may be represented as follows:

PPO program: Genetic procedure

Inputs: shipment files, zip-to-zip distance matrix
Step 1:
Set parameters–number of generations N, number of solutions in each generation M;
number of parents P, number of pool points K;
Step 2:
for m in M:
initialize the select or not select decisions (1,0) for the 50 pool points in solution m;
Step 3: (crossover)
for m in $[P, M]$:
exchange the pool point decisions between the parents randomly;
store the offspring solutions;
Step 4: (mutation)
for m in $\left[\frac{M}{2}+1, M\right]$:
for k in K:
switch the value of zero and one on pool point k to its opposite;
Step 5: (score and sort)
Call PPO program: Costing functions to get the cost for the M solutions;
sort the M solutions by their objective function values;
Step 6:
Repeat Steps 3–5 until the Nth generation.

Each solution provided by the PPO program is treated individually; its score is defined by a corresponding objective function value (transportation cost) and an infinity penalization to the decisions of choosing a pool without assigning any shipment thereto.

With the solution and detailed pool point assignments in place, the local improvement heuristic is called by the second part of the PPO program to further improve and optimize the solution. In this regard, consider an exemplary implementation of the PPO program in the United States, which includes 50 available pool points across the country. In this example, the PPO program considers the closest 10 of the 50 pool points for each shipment to complete a reassignment. Then, the PPO program reassigns shipments based on the local improvement heuristic. After application of the local improvement heuristic, a near-optimal, reliable solution is generated. An exemplary local improvement heuristic may be represented as follows:

PPO program: Local improvement heuristic

Step 1:
Make all shift and swap movements that improve the solution. Let the final cost be C_{old}. Make this solution the current one.
Step 2:
For each customer, calculate the cost C_{new} of shifting it from the current assigned pool point to each of the 10 selected points in the solution.
Step 3:
Determine the difference $d = C_{new} - C_{old}$.
Step 4:
if $d \le 0$ then
new assignment of the pool point is accepted.
end if
Go to Step 6.
Step 5:
if $d > 0$ then
determine the probability of the new assignment being accepted: $p = e^{-\frac{d}{t}}$,
where t is the temperature control parameter.
* To accomplish this, generate a $U[0,1]$ distributed random number r; if $r \le p$, assignment is accepted and made current (i.e., C_{new} and the current solution are both updated accordingly); otherwise ($r > p$), keep the current assignment.
Step 6:
Repeat Steps 2–5 until all shipments have been evaluated.

Finally, the allocated cost for each shipment is calculated and the summary output is presented (e.g., as a printed Excel worksheet). Because there are an exponential number of combinations of pool point decisions, the PPO program may take a while to determine an efficient solution. To hasten determining the solution, large shipment files may be divided into groups and multiprocessed by origins and time periods.

In light of the foregoing description, an exemplary PPO program may be represented as follows:

Main PPO Program

while $i \le$ *number of iterations*:
call PPO Program - Genetic procedure ;
collect the best solution z^H generated from the genetic procedure;
assign $C_{old} = z^H$ and call our local improvement heuristic;
call PPO program: Costing functions—cost_for_shipment() to get the allocated cost for each shipment;
$i+ = 1$;
end while.

Round-Trip Optimization

Finally, we briefly explore some symmetry-breaking rules in addition to the main symmetry breaking Constraint (A.18).

Here, we provide additional details regarding RTO program functionality and operations; however, we first summarize the various parameters and variables used by an exemplary RTO program in Table A.4.

RTO program

Input: weekly shipment file, constraint parameters, costs
Step 1:
Divide the shipment file into M sets by week;
Step 2:
Cluster the shipments by origins for each of the M sets;

Table A.4. Parameters That the RTO Program Uses

Parameters	Descriptions
a	Cost per mile for continuous move route
c_{ij}	Common carrier cost between i, j
L	Limit on the maximum number of shipments
x_{ij}^k	$= \begin{cases} 1, \text{ use round} - \text{trip truck } k \text{ between location } i, j \\ 0, \text{ otherwise} \end{cases}$
y_{ij}	$= \begin{cases} 1, \text{ use direct } TL/\text{LTL between location } i, j \\ 0, \text{ otherwise} \end{cases}$

Consolidate shipments that share the same origin and destination;
Step 3:
Optimize model $\wp$ with Google OR Tool solver;
Step 4:
Repeat Steps 2–3 until the specified iterations;
Step 5:
Collect the solution from the solver, find the round-trip arcs if $x_{ij}^k = 1$;
Output: round-trips and direct TL/LTL shipments.

An exemplary RTO program may be represented as follows:
The output of the RTO program may include two levels of information. For the round-trip level, the RTO program may report, for example, actual delivery time, allocated cost for each TL trip, empty-miles percentage, total distance of the round-trip, total transit days, savings percentage, and weight carried. For the shipment level, the RTO program may report, for example, detailed shipment information, delivery sequences, round-trip IDs, and allocated savings.

References

Adulyasak Y, Cordeau JF, Jans R (2015) The production routing problem: A review of formulations and solution algorithms. *Comput. Oper. Res.* 55(March):141–152.

Alcaraz J, Caballero-Arnaldos L, Vales-Alonso J (2019) Rich vehicle routing problem with last-mile outsourcing decisions. *Transportation Res. Part E* 129(September):263–286.

Cormen TH, Leiserson CE, Rivest RL, Stein C (2009) *Introduction to Algorithms*, 3rd ed. (MIT Press, Cambridge, MA).

Dang Y, Allen TT, Singh M (2022) A heterogeneous vehicle routing problem with common carriers and time regulations: Mathematical formulation and a two-color ant colony search. *Comput. Industry Engrg.* 168(June):108036.

Dang Y, Singh M, Allen TT (2021) Network mode optimization for the DHL supply chain. *INFORMS J. Appl. Anal.* 51(3):179–199.

Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Management Sci.* 6(1):1–14.

Dondo R, Cerdá J (2007) A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *Eur. J. Oper. Res.* 176(3):1478–1507.

Freeman NK, Keskin BB, McCullough C (2020) IJAA: Past, present, and future. *INFORMS J. Appl. Anal.* 50(6):355–372.

Macrina G, Pugliese LD, Guerriero F, Laporte G (2019) The green mixed fleet vehicle routing problem with partial battery recharging and time windows. *Comput. Oper. Res.* 101(January):183–199.

Sassi O, Cherif-Khettaf WR, Oulamara A (2015) Iterated tabu search for the mix fleet vehicle routing problem with heterogenous electric vehicles. Thi HAL, Dinh TP, Nguyen NT, eds. *Modelling, Computation and Optimization in Information Systems and Management Sciences* (Springer, Cham, Switzerland), 57–68.

Savelsbergh MW, Sol M (1995) The general pickup and delivery problem. *Transportation Sci.* 29(1):17–29.

Seyfi M, Alinaghian M, Ghorbani E, Çatay B, Sabbagh MS (2022) Multi-mode hybrid electric vehicle routing problem. *Transportation Res. Part E Logist. Transportation Rev.* 166(October):102882.

Sherali HD, Smith JC (2001) Improving discrete model representations via symmetry considerations. *Management Sci.* 47(10): 1396–1407.

Tummel C, Franzen C, Hauck E, Jeschke S (2013) The multi-depot heterogeneous fleet vehicle routing problem with time windows and assignment restrictions (M-VRPTWAR). Jeschke S, Isenhardt I, Hees F, eds. *Automation, Communication and Cybernetics in Science and Engineering* (Springer, Berlin), 767–779.

Yibo Dang is an applied scientist at Amazon. He received his PhD from The Ohio State University in integrated systems engineering and worked at DHL Supply Chain North America and later JP Morgan Chase. His interests include demand modeling and large-scale integer programming. He was the lead programmer and modeler for the transport network optimizer.

Theodore T. Allen is a professor of integrated systems engineering and computer science engineering (courtesy) at The Ohio State University. He earned his PhD degree from the University of Michigan. His prior roles include serving on the board of the QSR section and as past president of the Social Media Analytics section. He is currently the vice president of communications for the PSOR section. He helped with the design of some of the algorithms and was the lead for project documentation.

Manjeet Singh is the senior director of AI, data & analytics CoE at DHL Supply Chain. He received his PhD from Ohio University and previously held the position of operations research lead at Net Jets. His diverse interests include large-scale optimization, warehouse modeling and optimization, and extensive vehicle routing with outsourcing. He led the development of the transport network optimizer overseeing Yibo.

Jason Gillespie is the senior director and head of continuous improvement and innovation for transportation at DHL Supply Chain North America. He earned an MBA from Franklin University and a BS in organizational psychology from The Ohio State University. A former U.S. Army Ranger, he led his team in the implementation of the transport network optimizer, focusing on the continuous improvement of existing projects.

Jon Cox serves as the senior director of solutions design at DHL Supply Chain North America. He is instrumental in supporting the acquisition and renewal of business by delivering value through effective supply chain designs. He holds MBA and BS degrees from The Ohio State University, Fisher College of Business. His team played a pivotal role as the lead customer and partner in the development.

James Monkmeyer is the president of transportation and LLP North America at DHL Supply Chain. He holds an MS in transportation from Northwestern University and a BS from the University of Wisconsin (Madison). His prior experience includes roles at England Logistics, A3 Integration, and JP Morgan Chase. He commissioned, oversaw the development, and implemented key projects for the transport network optimizer.