Mühendislik Bilimleri ve Tasarım Dergisi
12(4), 802-820, 2024
e-ISSN: 1308-6693

Journal of Engineering Sciences and Design
DOI: 10.21923/jesd.1445781

Araştırma Makalesi | Research Article

OPTIMIZING READY-MIXED CONCRETE TRANSPORTATION BY A TRUCK MIXER ROUTING MODEL FOR CONCRETE PLANTS

Ozan Aykut DÖNMEZ[1*], Erdinç ÖNER[2]

[1] Yaşar University, Graduate School, Master of Science in Business Engineering, Izmir, Türkiye
[2] Yaşar University, Faculty of Engineering, Department of Industrial Engineering, Izmir, Türkiye

Keywords

Concrete Delivery, Optimization, Time Windows, Truck Mixer, Vehicle Routing Problem.

Abstract

This study aims to develop a truck mixer routing model to increase the efficiency of concrete delivery operations in concrete plants and to propose suitable solution methods for the model. In this study, a mixed-integer linear programming model was developed, based on a variant of the vehicle routing problem known as the capacitated vehicle routing problem with time windows, by incorporating constraints specific to concrete transportation. The objective of the model is to reduce transportation costs by minimizing the total distance travelled by truck mixers and the total number of truck mixers used. The model was first addressed by exact solution methods in Gurobi Optimizer. Since the vehicle routing problem is classified as an NP-hard problem, the complexity of the model increases with the number of customers, leading to a longer computation time. Thus, the model was addressed by a heuristic method developed for this study. The results show that the Gurobi Optimizer provided optimal solutions for up to 15 customers and 15 vehicles within a reasonable computation time, whereas the heuristic method quickly provided near-optimal solutions with a 3.39% cost increase on average.

BETON SANTRALLERİ İÇİN BİR TRANSMİKSER ROTALAMA MODELİ İLE HAZIR BETON TAŞIMACILIĞININ OPTİMİZE EDİLMESİ

Anahtar Kelimeler

Araç Rotalama Problemi, Beton Sevkiyatı, Optimizasyon, Transmikser, Zaman Penceresi.

Öz

Bu çalışma hazır beton santrallerinin beton sevkiyatı verimliliğini arttırmaya yönelik bir transmikser rotalama modeli oluşturmayı ve bu modele uygun çözüm yöntemlerini sunmayı amaçlamaktadır. Bu çalışmada araç rotalama probleminin bir çeşidi olan kapasite kısıtlı ve zaman pencereli araç rotalama problemi kullanılarak, hazır beton taşıma operasyonlarına özgü kısıtlar da eklenerek, bir karma tam sayılı doğrusal programlama modeli oluşturulmuştur. Bu modelin amacı transmikserlerin katettiği toplam mesafeyi ve kullanılan toplam transmikser sayısını azaltarak, taşıma operasyonlarının maliyetini azaltmaktır. Model ilk olarak Gurobi Optimizer kullanılarak kesin çözüm yöntemleriyle ele alınmıştır. Fakat araç rotalama probleminin NP-zor sınıfında olması, müşteri sayısı arttıkça modelin karmaşıklığını arttırmıştır ve bu da problemin çözüm süresini uzatmıştır. Bu nedenle model, bu çalışma için geliştirilen sezgisel bir yöntemle çözülmüştür. Elde edilen sonuçlar, Gurobi Optimizer'in makul bir hesaplama süresiyle 15 müşteriye ve 15 araca kadar optimal çözümler sağladığını, sunulan sezgisel yöntemin ise ortalama 3.39%'lık bir maliyet artışıyla hızlı bir şekilde optimale yakın çözümler verdiğini göstermektedir.

Alıntı / Cite

Dönmez, O. A., Öner, E., (2024). Optimizing Ready-Mixed Concrete Transportation by a Truck Mixer Routing Model For Concrete Plants, Journal of Engineering Sciences and Design, 12(4), 802-820.

Yazar Kimliği / Author ID (ORCID Number)

O. A. Dönmez, 0009-0000-8351-2682
E. Öner, 0000-0002-0503-7588

Makale Süreci / Article Process

Başvuru Tarihi / Submission Date	01.03.2024
Revizyon Tarihi / Revision Date	18.10.2024
Kabul Tarihi / Accepted Date	10.11.2024
Yayım Tarihi / Published Date	25.12.2024

* İlgili yazar / Corresponding author: ozanaykutdonmez@gmail.com, +90-553-316-7619

OPTIMIZING READY-MIXED CONCRETE TRANSPORTATION BY A TRUCK MIXER ROUTING MODEL FOR CONCRETE PLANTS

Ozan Aykut DÖNMEZ[1†], Erdinç ÖNER[2],

[1] Yaşar University, Graduate School, Master of Science in Business Engineering, Izmir, Türkiye
[2] Yaşar University, Faculty of Engineering, Department of Industrial Engineering, Izmir, Türkiye

Highlights

- A mixed-integer linear programming model enhances truck mixer routing efficiency.
- The capacitated vehicle routing problem with time windows, modified with concrete-specific constraints, forms the model's basis.
- A heuristic method, balancing optimal solutions and computation time, is developed.
- The model is validated with three cases generated from the concrete plant data.

Purpose and Scope

This research focuses on enhancing the efficiency of ready-mixed concrete delivery in construction. The aim is to develop a truck mixer routing model for concrete plants, intending to optimize concrete transportation operations.

Design/methodology/approach

The study employs a mixed-integer linear programming model, a variant of the capacitated vehicle routing problem with time windows, tailored for ready-mixed concrete transportation. Split delivery and multi-trip constraints are integrated. The model aims to minimize transportation costs by reducing travel distance and the number of truck mixers used. It was initially solved using Gurobi Optimizer and later approached through a heuristic method developed for this study, including tabu search and nearest neighbour algorithms.

Findings

The model effectively routed the truck mixers at reduced transportation costs, with Gurobi Optimizer yielding optimal solutions for scenarios with up to 15 customers and 15 vehicles. For larger sets, the heuristic method provided near-optimal solutions with only a 3.39% average cost increase in 1.2 seconds. This demonstrates the model's practical utility in real-world scenarios, offering a balance between computation time and solution quality.

Research limitations/implications

The study's primary limitation lies in the increasing complexity with larger customer sets, a prevalent challenge in solving NP-hard problems like the vehicle routing problem. Future research could explore more efficient algorithms or methods to handle larger datasets without significant loss in solution quality or increased computation time.

Practical implications

For ready-mixed concrete companies, the study's findings imply potential cost savings and increased operational efficiency in concrete transportation. Implementing the proposed routing model could lead to more efficient use of resources, contributing to reduced operational costs and enhanced service quality in the construction sector.

Social Implications

The efficient routing of truck mixers not only benefits the construction industry but also positively impacts society by reducing traffic congestion and lowering emissions. Improved delivery efficiency can influence public attitudes towards urban development projects, align with corporate social responsibility goals, and enhance overall quality of life through better infrastructure development.

Originality

This paper introduces a novel routing model for concrete transportation, addressing a specific challenge in ready-mixed concrete delivery. Its originality lies in adapting the vehicle routing problem to the unique demands of concrete transportation, providing valuable insights for industry practitioners and contributing to the body of knowledge in operational research and logistics.

† Corresponding author: ozanaykutdonmez@gmail.com, +90-553-316-7619

1. Introduction

The vehicle routing problem (VRP) is a classic problem related to combinatorial optimization, widely recognized in the areas of operations research and logistics. VRP is generally utilized for optimizing routes of vehicles to minimize transportation costs in logistics operations. Keskintürk et al. (2015) stated that VRP is widely applied in the following areas: waste collection, distribution of products from one or more depots to different customer points, applications in the transportation and logistics industry, distribution and collection problems, determining school bus routes, aircraft routing problems, delivery of online purchases, distribution of newspapers, mail, bread, beverages, etc., patrolling vehicle routing, material collection problems in the inventory area, transportation of disabled individuals, routing of service vehicles, and design of material flow systems.

The first VRP was introduced in an article named "The Truck Dispatching Problem" by Dantzig and Ramser (1959). In their paper, they presented a real-life case involving the distribution of gasoline to service stations and introduced the first algorithmic approach and mathematical programming model (Toth & Vigo, 2014). The classical VRP is classified as an NP-hard problem (Lenstra & Kan, 1981). This means that the number of solutions increases exponentially with the number of nodes, thus solving the problem with exact methods, in large instances, is not accomplished in reasonable computation times. Therefore, heuristics were developed to solve the VRP. El-Sherbeny (2010) stated that in cases where merely obtaining a feasible solution is insufficient and the solution quality is crucial, it gets notable to pursue efficient methods to obtain the best feasible solutions within time constraints deemed reasonable.

The VRP has many variants. The most investigated variants of VRP are the capacitated vehicle routing problem (CVRP) and the vehicle routing problem with time windows (VRPTW). Both CVRP and VRPTW are mixed-integer programming problems in combinatorial optimization and can be solved with exact methods and heuristic approaches. CVRP focuses on finding the optimal route for vehicles with limited capacity to serve a set of customers, whose demands and locations are known, while considering various constraints. In contrast, VRPTW not only aims to determine the most efficient route but also incorporates constraints related to permissible delivery times or time windows, which are governed by delivery deadlines and the earliest allowable delivery times (Solomon, 1987). In addition to these variants, the capacitated vehicle routing problem with time windows (CVRPTW) represents an extension of VRPTW and CVRP. CVRPTW involves the optimization of vehicle routes for a fleet tasked with delivering items to customers, taking into account limitations on vehicle capacity and time window restrictions. CVRPTW takes part in different business areas, including but not limited to delivery scheduling, emergency response planning, and supply chain management.

Ready-mixed concrete (RMC) is a form of concrete pre-mixed at a factory or batching plant based on specific concrete mix designs, and subsequently transported to construction sites using truck-mounted transit mixers. (truck mixers). The RMC industry plays a pivotal role in the construction sector, as concrete is the most fundamental material used in construction projects from residential buildings to massive infrastructure developments with a global consumption rate nearing 25 gigatonnes per year (Gursel et al., 2014). RMC industry provides various benefits such as consistent quality, reduced wastage, and speedier construction. RMC suppliers encounter operational difficulties that encompass the procurement of raw materials, production schedule of facilities, and transportation of concrete (Kinable et al., 2014).

RMC suppliers strive to effectively meet their customers' demand and plan the transportation of concrete to the customers' locations while ensuring customer satisfaction and complying with the customers' time windows. However, transportation of RMC from the batching plant to the customers involves several important considerations due to the characteristics of the concrete. The truck mixers used in the RMC industry have different fixed capacities, limiting the volume of concrete they can transport. As the ordered quantity of concrete usually exceeds the capacity of a single truck mixer, it is necessary to schedule several consecutive deliveries to fulfill a single order (Schmid et al., 2010). Furthermore, the travel distance is very important; the longer the concrete stays in the truck, the higher the risk it will begin setting before it's delivered. Efficient routing is imperative not only to ensure timely delivery but also to optimize fuel consumption. Traffic and road conditions can significantly influence delivery times. Moreover, the production schedule of the batching plant must align with delivery timelines. Customers, on their part, have specific time windows within which they require the RMC to be delivered. Furthermore, numerous technical constraints related to the duration of unloading operations need to be factored in (Schmid et al., 2010). Once the delivery and service are complete, the total time taken for the empty truck to return to the plant must be considered. Also, strict adherence to environmental and safety regulations is crucial as spillage or undue delays can lead to environmental hazards and safety risks.

This paper centers on optimizing RMC transportation by utilizing a truck mixer routing model and solution methods. CVRPTW was modified to include split delivery constraints and other constraints specific to the problem.

This paper aims to propose a mathematical model of the problem and suitable solution methods which include an exact solution method and a proposed heuristic method. This study intends to present a practical real-world application of the CVRPTW specifically tailored to a ready mixed-concrete plant. This paper is structured into several key sections. The Literature Survey section offers a concise overview of the variants of the vehicle routing problem. In the Material and Method section, the mathematical model for ready-mixed concrete transportation is introduced, alongside the solution methods employed, including the Gurobi Optimizer and a heuristic method developed for this study. The Experimental Results section provides detailed insights into the outcomes of three distinct case studies. Finally, the Result and Discussion section summarizes the methodologies, solution methods, and key findings of the study, while also highlighting potential areas for future work.

2. Literature Survey

The vehicle routing problem (VRP) represents a fundamental problem in combinatorial optimization, focusing on devising the most efficient routes for a fleet of vehicles tasked with servicing a set of locations. Dantzig and Ramser (1959) proposed the first VRP in a paper titled The Truck Dispatching Problem as a real-world case of gasoline delivery to gas stations. The first algorithmic technique and mathematical programming model that aim to minimize the total route cost for VRP were proposed in this paper. Following this inspiring paper, Clarke and Wright proposed a greedy heuristic algorithm that provided a near-optimal solution to the VRP in their paper in 1964. The academic community has extensively explored the Vehicle Routing Problem (VRP) over the past six decades, with a wealth of articles appearing in leading journals focused on International Operations Research and Transportation Science. These publications have contributed to the development of various mathematical models and the introduction of precise, heuristic, and metaheuristic algorithms aimed at solving different versions of the VRP. In real-life applications, the complexity of capacitated vehicle routing problems often necessitates the use of heuristic and metaheuristic approaches to obtain feasible solutions within a reasonable time (Şahin & Eroğlu, 2015, p. 19)

Table 1. Vehicle Routing Problem Variants

Vehicle Routing Problem Variant	Key Characteristics	Key Constraints
Vehicle Routing Problem (VRP)	Basic vehicle routing optimization	Route optimization
Capacitated Vehicle Routing Problem (CVRP)	Vehicle routing with capacity limits	Vehicle capacity
Vehicle Routing Problem with Time Windows (VRPTW)	Vehicle routing with customer time windows	Time windows for deliveries
Split Delivery Vehicle Routing Problem (SDVRP)	Split deliveries to the same customer	Vehicle capacity; Split deliveries allowed
Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)	Capacity constraints and time windows	Vehicle capacity; Time windows
Multiple Trip Vehicle Routing Problem (MTVRP)	Multiple trips by the same vehicle	Vehicle capacity; Multiple trips per vehicle

Toth and Vigo (2014) declared that the most studied type of VRP is the capacitated vehicle routing problem (CVRP). CVRP is a mixed-integer linear programming (MILP) problem that deals with determining the optimal route for vehicles with limited capacity to carry goods while satisfying constraints. VRP is a complex optimization problem, and solving it with classical methods can be challenging, especially when considering additional constraints like capacity or time windows" (Ünsal & Yiğit, 2018, p. 7). Fukasawa et al. (2006) developed an exact algorithm that combines the branch-and-cut method with the old q-routes approach to derive superior lower bounds. This algorithm provides an optimal solution for the CVRP with up to 100 vertices because the CVRP is a non-polynomial hard (NP-hard) problem. Pichpibul and Kawtummachai (2012) resolved the CVRP by using Clarke and Wright's savings algorithm with an iterative improvement approach. This well-known heuristic method provides a fast and approximate solution for the CVRP. Fitriani et al. (2021) tackled the CVRP using notable heuristics—Clarke and Wright's savings, sequential insertion, and nearest neighbour algorithms to minimize the total distance. Their comparison showed Clarke and Wright's savings algorithm as the top performer in solving the CVRP.

The vehicle routing problem with time windows (VRPTW) is another most-studied variant of the vehicle routing problem (VRP). VRPTW is the extension of the capacitated vehicle routing problem (CVRP) that includes time constraints. In VRPTW, the service time for each customer must begin within a corresponding time interval, called a time window (Toth & Vigo, 2014, p:119). In VRPTW, the two types of time windows are recognized. These are hard time windows and soft time windows. On the one hand, in cases with hard time windows, a vehicle arriving before the scheduled time at a customer's location is obliged to wait until the customer is prepared to initiate service. This implies that any delay occurring prior to the commencement of a time window is not subject to any penalties. On the other hand, in cases with soft time windows, there is some flexibility, and thus Each time window may be breached, subject to the imposition of a penalty cost. The objective of VRPTW is to provide service to several customers within predefined time windows at a possible minimum cost (in terms of distance travelled) without violating the total trip time and capacity constraints for vehicles. Tan et al. (2001) undertook a comparative analysis of the different heuristic methods capable of providing near-optimal solutions to solve VRPTW. They examined the heuristics: local search with λ-interchange, simulated annealing (SA), tabu search (TS), and genetic algorithm (GA). The capacitated vehicle routing problem with time windows (CVRPTW) is an extension of the vehicle routing problem with time windows. CVRPTW schedules the route for a fleet of vehicles that must deliver goods to customers while considering capacity constraints and time window constraints. In real life, CVRPTW arises in different business areas such as delivery scheduling, emergency response planning, and supply chain management., Li (2015) utilized the multiple ant colony system algorithm which is the extended version of ant colony optimization to solve CVRPTW. Multiple ant colony system algorithms provided efficient solutions for CVRPTW. The solutions reduced the number of trucks by 10% and decreased travel time by 9.87% while satisfying all time windows. Bruno (2019) developed software including heuristics for CVRPTW for a food company called Soral. The software consists of the time-oriented Clarke and Wright saving algorithm and the time-oriented nearest neighbour algorithm.

Table 2. Literature Review on Vehicle Routing Problems Solutions

Author(s) & Year	Vehicle Routing Problem Variant	Methodology
Babaee Tirkolaee, E., Abbasian, P., Soltani, M., & Ghaffarian, S. A. (2019)	The Multi-Trip Vehicle Routing Problem with Time Windows (MTVRPTW)	CPLEX solver and proposed heuristic consisting of Simulated Annealing with local search algorithms
Brandao, J. (2006)	The Vehicle Routing Problem with Backhauls (VRPB)	Tabu Search Algorithm with K-Trees solution
Bruno, L. (2019)	The Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)	Prosed Method, Clarke and Wright's Algorithm, and Nearest Neighbour Algorithm
Chen, P., Huang, H. K., & Dong, X. Y. (2010)	The Capacitated Vehicle Routing Problem (CVRP)	Iterated variable neighborhood descent (IVND) algorithm with relocation, swap, 2-opt*, 2-opt and cross-exchange
Chen, S., Golden, B., & Wasil, E. (2007)	The Split Delivery Vehicle Routing Problem (SDVRP)	Proposed Heuristics: combining Clarke and Wright's Algorithm with Record- to- Record Tavel Algorithm
Fitriani, N. A., Pratama, R. A., Zahro, S., Utomo, P. H., & Martini, T. S. (2021)	The Capacitated Vehicle Routing Problem (CVRP)	Clarke and Wright's Saving Algorithm, The Sequential Insertion Algorithm, and The Nearest Neighbor Algorithm
Fukasawa, R., Longo, H., Lysgaard, J., Aragão, M. P. D., Reis, M., Uchoa, E., & Werneck, R. F. (2006)	The Capacitated Vehicle Routing Problem (CVRP)	Branch-and-Cut and Price Algorithm with old q routes approach
Kek, A. G., Cheu, R. L., & Meng, Q. (2008)	The Distance-Constrained Vehicle Routing Problems (DCVRP)	ILOG OPL Studio Cplex and Branch and Bound Algorithm with a node selection strategy
Li, X. (2015)	The Capacitated Vehicle Routing Problem with Time Windows (CVRPTW)	Multiple Ant Colony Optimization Algorithm System (MACS)
Solomon, M. M. (1987)	The Vehicle Routing Problem with Time Windows (VRPTW)	Clarke and Wright Savings algorithm, time-oriented Nearest Neighbor, Insertion Heuristics, time-oriented Sweep Heuristics
Tan, K. C., Lee, L. H., Zhu, Q. L., & Ou, K. (2001)	The Vehicle Routing Problem with Time Windows (VRPTW)	Local Search with λ-interchange, Simulated Annealing, Tabu Search, and Genetic Algorithm

Table 2 presents a variety of approaches to solving different vehicle routing problem (VRP) variants, showcasing both heuristic and exact methods. Researchers have employed exact methods like branch-and-cut, branch-and-bound, and CPLEX solvers to find optimal solutions, particularly for complex problems like the capacitated (CVRP). The Capacitated Vehicle Routing Problem (CVRP) is classified as NP-Hard, and exact solutions in real-life applications are often impractical, which leads to the adoption of heuristic methods for solving it (Karagül, Tokat, & Aydemir, 2016, p. 217). In addition, heuristic methods such as Clarke and Wright's savings algorithm, Nearest Neighbour, and Tabu Search are also included, demonstrating their effectiveness in finding near-optimal solutions for large-scale and time-constrained VRPs. Recent advancements in heuristic algorithms have focused on hybrid approaches, combining elements of multiple algorithms to enhance solution accuracy while maintaining computational efficiency.

In ready-mix concrete (RMC) transportation, it is essential to effectively meet customer demands and comply with their specified time windows for the successful progression of construction projects. Therefore, concrete transportation can be conceptualized as a capacitated vehicle routing problem with time windows (CVRPTW) modified by the characteristics of other VRPs, such as split delivery constraints and multi-trip considerations. CVRPTW is an NP-hard problem and such modifications increased the complexity of the CVRPTW, which means more computation time is required to solve the model. This model was represented as a mixed-integer linear programming (MILP). For exact solutions to address the model, software like Gurobi Optimizer and IBM ILOG CPLEX can be utilized for optimal solutions. Additionally, exact methods like the branch-and-bound, the cutting-plane methods, and the branch-and-cut method combine both of these methods. On the heuristic methods side, there are several popular heuristic methods, including Clarke and Wright's savings algorithm, nearest neighbour algorithm, and sweep algorithm. These heuristics are frequently used for constructing near-optimal solutions. When it is required to improve these solutions, local search algorithms, including the 2-opt, 3-opt, k-opt exchanges, the λ-interchange, and the insertion sort can be beneficial. Metaheuristic methods, particularly Tabu Search and Genetic Algorithms, offer flexibility and adaptability, making them highly suitable for real-world scenarios where problem constraints may change dynamically. For metaheuristic approaches, the tabu search algorithm is considered one of the best options.

3. Material and Method

This section presents a mathematical model developed for the transportation of ready-mixed concrete (RMC) and proposes exact and heuristic solution methods applied to solve the model. To understand the nature of the problem, data from a medium-sized concrete plant was utilized. The concrete plant aims to route the truck mixers in a way that is both cost-effective and time-efficient. The concrete plant operates with a limited number of truck mixers, and therefore the plant uses these trucks multiple times after serving the customers. Additionally, the truck mixers must visit customers on specified time windows and move only between the customers and the concrete plant. However, when customer demand exceeds the truck mixer capacity, the concrete plant meets this demand with split deliveries. In these split deliveries, the successor truck mixer must wait at the plant to proceed to the customer until the preceding truck mixer has completed the service of the concrete due to the setting time of the concrete. Given the specific requirements of the problem, the capacitated vehicle routing problem with time windows (CVRPTW) was selected and modified to better suit these needs for the mathematical model.

The proposed model is based on the framework of CVRPTW and includes split delivery constraints and multi-trip considerations. Due to the complexity of RMC transportation, some assumptions were made when the model was being developed. Two solution methods both exact and heuristic were implemented as Python scripts to address the problem. First, the model was solved with the Gurobi Optimizer which uses an exact solution method, branch and cut method. Since the model is classified as an NP-hard problem, the model's complexity increases with the number of customers, which leads to longer computation times. Hence, the model was also solved by a heuristic method named the Urgent Demand Algorithm. The algorithm is based on a greedy algorithm incorporating time windows, capacity constraints, and split delivery constraints. The algorithm was inspired by the nearest neighbour algorithm and the tabu search. The heuristic method, while not guaranteed to find the optimal solution, provides a faster alternative to exact methods, making it suitable for large-scale problems.

3.1. Assumptions

Due to the complexity of the concrete transportation problem and the data set, the following assumptions have been made to integrate the problem into the mathematical model and find the optimal solution:

- The service time (h) for transferring concrete to customers, whether by conveyor belt or concrete pump, is assumed to be the same for all types.
- It is assumed that all customers require the same strength class of concrete in MPa. This assumption

simplifies the production and delivery process, as it eliminates the need for varying mix designs and allows for uniform batching.

- The fleet is considered homogenous and the truck mixers have the same capacity (q). This simplifies the model by eliminating variability in vehicle size, making route optimization more straightforward. In real-world applications, this assumption reflects many construction operations where standardized truck mixers are used, ensuring consistent delivery volumes and reducing complexity in scheduling and load planning. This uniformity helps streamline logistics and ensures efficient use of resources across all delivery routes.
- The distance between customers i and j is defined by combining the distance from the plant to customer i and the distance from the plant to customer j.
- It is assumed that the time windows for each customer are set with a duration of a 1-hour gap, accounting for the setting time of the concrete.
- It is assumed that the waiting time between truck mixers making consecutive deliveries is set equal to the service time (h), in consideration of the concrete setting time. This ensures that the concrete does not set prematurely and remains in optimal condition upon arrival at the site. Additionally, this adjustment ensures that the truck mixers can efficiently return to the plant, load fresh concrete, and deliver it on time. In practice, this timing constraint helps maintain concrete quality and aligns the model with real-world operational challenges, such as batching schedules and traffic delays, thereby enhancing overall efficiency.
- All traffic conditions and environmental factors are assumed to be optimal and all truck mixers move at a constant velocity; thus, cost per distance in kilometres for truck mixer k (c_k) is a constant parameter.
- The driver cost (D_k) is assumed to be based on the daily driver cost for operating a truck mixer, calculated from the monthly gross salary of a truck driver. Since the model provides daily routes, the monthly salary is converted into a daily rate.

3.2. Mathematical Model

The ready-mixed concrete (RMC) transportation problem can be modeled by using the framework of the capacitated vehicle routing problem with time windows (CVRPTW) modified by split delivery constraints and multi-trip considerations. Toth and Vigo (2014) declared that CVRPTW is defined on the directed graph G = (V, A), where the batching plant is represented by the two vertices 0 and n + 1. All feasible truck mixer routes correspond to paths in G that start from node 0 and end at node n + 1. Let V be the vertices, $N = V\backslash\{0, n+1\}$ be the set of customer vertices, $A = (V\backslash\{n+1\}) \times (V\backslash\{0\})$ be the set of arcs and be the K set of homogenous truck mixers. Binary variable $x_{ijk} = 1$ if and only if truck mixer $k \epsilon K$ moves over the arc $(i,j) \epsilon A$. Time windows are associated with, $[e_1, l_1] = [e_n, l_n] = [E, L]$ where E and L stand for earliest and latest service start times for the truck mixers at the customer. Additionally, zero demands (w) and service times (s) are defined for these two nodes $w_0 = w_{n+1} = s_0 = s_{n+1} = 0$. Assuming that the travel time matrix satisfies these equations, feasible solutions exists only if $e_0 \le min_{i \in V/\{0\}}\{l_i - t_{0i}\}$ and $l_0 \ge max_{i \in V\backslash\{0\}}\{max\{e_0 + t_{0i}, e_i\} + s_i + t_{i,n+1}\}$. Moreover, an arc $(i,j) \in A$ can be excluded because of temporal considerations, if $e_i + s_i + t_{ij} > l_j$, or capacity limitations or $w_i + w_j > q$ or by other factors. Toth and Vigo (2014) stated that in split delivery vehicle routing problem (SDVRP), a feasible solution to the problem with |K| vehicles exists if and only if $\sum_{i \in N} w_i \le Kq$ holds. Also, delivery amounts of all routes serving a customer $i \in N$ sum up to the demand w_i. This mixed integer linear programming (MILP) model for RMC transportation entails defining specific sets, indices, parameters and decision variables, followed by the formulation of objective function and associated constraints:

Sets:

N Set of customers, $N = \{1,2, \dots, n\}$

V Set of nodes, customers and the batching plant denoted as (0), $V = N \cup \{0\}$

K Set of homogenous fleet of mixer trucks, $K = \{1,2, \dots, k\}$

Indices:

i, j Indices for customers and the batching plant, $\forall i, j \in V$

k, m Indices for preceding and successor trucks $\forall k \in K, \forall m \in [k+1, K]$

Parameters:

$\boldsymbol{d_{ij}}$ Distance between node i and node j, $\forall i, j \in V,\ i \neq j$, $\boldsymbol{d_{ij}} = d_{0i} + d_{0j}$

$\boldsymbol{w_i}$ Demand of customer i in cubic meters (m^3), $\forall i \in N$

$\boldsymbol{c_k}$ Cost per distance in kilometres for truck mixer k, $\forall k \in K$

$\boldsymbol{D_k}$ Driver cost for truck mixer k, $\forall k \in K$

$\boldsymbol{q}$ Capacity of truck k in cubic meters, $\forall k \in K$

$\boldsymbol{t_{ij}}$ Travel time from station i to station j, $i,j \in V$, $\boldsymbol{t_{ij}} = t_{0i} + t_{0j}$

$\boldsymbol{t_{0i}}$ Travel time from the plant to station i, including loading time of truck mixer

$\boldsymbol{e_i}$ Earliest departure time from station i, $\forall i \in V$

$\boldsymbol{l_i}$ Latest arrival time from station i, $\forall i \in V$

$\boldsymbol{h}$ Service time of truck mixer k at the customer, $\forall k \in K$, $\forall i \in N$

Decision Variables:

$$\boldsymbol{x_{ijk}} = \begin{cases} 1, & \text{If the truck mixer k moves from node i to node j } (\forall i,j \in V, \forall k \in K) \\ 0, & \text{If the truck mixer k does not move from node i to node j} \end{cases}$$

$\boldsymbol{s_{ik}}$ Time at which truck mixer k starts service at customer i, $\forall i \in N, \forall k \in K$

$\boldsymbol{f_{ik}}$ Percentage of demand of customer i delivered by truck mixer k

Objective Function:

$$Minimize \ \sum_{i\in V}\sum_{j\in V, i\neq j}\sum_{k\in K} c_k \times d_{ij} \times x_{ijk} + \sum_{k\epsilon K}\sum_{j\epsilon V} x_{0jk} * D_k \quad (1)$$

subject to;

$$\sum_{j=1}^{V} x_{0jk} \le 1 \qquad \forall\, k \in K\,, \quad (2)$$

$$\sum_{k=1}^{K} f_{ik} = 1 \qquad \forall\, i \in N, \forall\, k \in K\,, \quad (3)$$

$$w_i f_{ik} \le q \qquad \forall\, i \in N,\ k \in K\,, \quad (4)$$

$$\sum_{j=0}^{V} x_{jik} \ge f_{ik} \qquad \forall i \in N, k \in K\ , \quad (5)$$

$$\sum_{i\in V} x_{ijk} - \sum_{i\in V} x_{jik} = 0 \qquad \forall j \in N,\ \forall k \in K\ , \quad (6)$$

$$x_{00k} = 0 \qquad \forall k \in K\ , \quad (7)$$

$$e_i \le s_{ik} \qquad \forall\, i \in N, k \in K\ , \quad (8)$$

$$l_i \ge s_{ik} \qquad \forall\, i \in N, k \in K\,, \quad (9)$$

$$s_{ik} + t_{ij} + h - M * (1 - x_{ijk}) \le s_{jk} \qquad \forall i \in V, \forall j \in N, \forall k \in K\,, \quad (10)$$

$$s_{im} - s_{ik} \ge h * \left(\sum_{j=0}^{V}(x_{jik} + x_{jim}) - 1\right) \qquad \forall i \in N, i \neq j, \forall k \in K, \forall m \in [k+1, K]\,, \quad (11)$$

$$x_{ijk} \in \{0,1\} \qquad \forall i,j \in V, \forall k \in K\,, \quad (12)$$

$$s_{ik} \ge 0 \qquad \forall i \in N, k \in K\,, \quad (13)$$

$$f_{ik} \ge 0 \qquad \forall i \in N, k \in K\,. \quad (14)$$

The objective function (1) minimizes the total cost in terms of the total distances traveled by the truck mixers and the driver cost of trucks, respectively. Constraint (2) ensures that each truck can depart from the batching plant to any customer at most once to start its route. Constraint (3) ensures that split deliveries fulfill the entire demand of each customer. Capacity constraint (4) ensures that the volume of concrete supplied by a truck to any customer should not exceed the truck's capacity. Constraint (5) ensures that if a split delivery of demand for any customer is delivered by a truck, then there must be a corresponding route for the truck to the customer. Constraint (6) maintains flow continuity, which means that for every route where a truck visits a customer, there is a corresponding route exiting that customer. Constraint (7) ensures that trucks cannot have a route that starts and ends at the batching plant without visiting customers. Constraints (8) and (9) ensure that the trucks start service between the customers' earliest and latest acceptable service times. Constraint (10) ensures that the trucks serve customers within the time windows by regulating service start times based on travel and service times. M is a sufficiently large constant, often referred to as the big-M Constraint. (11) ensures that when the two trucks supply concrete to the same customer through split deliveries, there is a time difference of at least the allowed time lag between them. Constraint (12) ensures that x_{ijk} is a binary decision variable. Constraints (13) and (14) ensure that s_{ik} and f_{ik} are non-negative continuous decision variables. Subtour elimination prevents vehicles from forming cycles (subtours) where they revisit the same point without completing the entire route. When truck mixers travel between customers, they must visit the concrete batching plant. This condition and existing constraints ensure that subtours do not occur in the model. Thus, adding subtour constraints would lead to redundant complexity.

3.3. Proposed Methods

The mathematical model of the problem was addressed using both an exact method and a heuristic method. The exact method was represented by the Gurobi Optimizer, which utilizes the branch and cut method. Additionally, a heuristic method named the Urgent Demand algorithm was developed specifically for this problem. Both solution methods were implemented as Python scripts to effectively solve the problem.

3.3.1. Branch and Cut Method in Gurobi Optimizer

The branch and cut method is a highly successful technique for solving numerous types of integer programming problem, and it can ensure optimality. Most types of vehicle routing problem can be formulated as mixed integer linear programming models so they can be solved by the branch and cut method. The branch-and-cut method consists of the well-known exact methods: branch-and-bound method and the cutting plane method. These techniques are employed to address a series of linear programming relaxations that represent an approximation of the original integer programming challenge. The cutting plane technique refines this approximation to achieve a closer representation of the integer programming issue. Subsequently, the branch-and-bound method applies a sophisticated divide-and-conquer strategy to effectively resolve these problems (Mitchell, 2002).

Branch and Cut method begins by initializing the problem and creating a list of active nodes. Each node represents a subproblem which is a version of the original problem with additional constraints. The algorithm enters a cycle where a node is selected and its linear programming (LP) relaxation is solved to find the best LP solution. If the LP solution improves the current best solution (incumbent) and is integer, it updates the incumbent. If not, and if the solution cannot be improved by adding cutting planes, the node is partitioned into subproblems which are added to the list of active nodes. Cutting planes are linear inequalities that exclude the current non-integer solution without excluding any feasible integer solutions. They are added to tighten the LP relaxation and cut off fractional solutions, improving the algorithm's efficiency. Throughout the process, nodes are pruned based on several criteria: if the solution is worse than the incumbent, if it's infeasible, or if it's an integer solution that doesn't improve the incumbent. The algorithm terminates when there are no more nodes to explore, meaning the list of active nodes is empty. The best integer solution found is then the output, along with its corresponding objective value. This solution is optimal within the explored space, assuming all potential solutions have been considered through the branching and cutting process.

Initially, the mathematical model was implemented as a Python script utilizing Gurobi Optimizer, along with pandas and NumPy libraries. Gurobi Optimizer is a mathematical optimization software that uses the branch and cut method as one of its key techniques for solving mixed-integer linear programming problems (Linderoth & Lodi, 2010). On the other hand, NumPy is a general-purpose package for scientific computing in Python. NumPy provides support for large arrays and matrices (including multidimensional arrays), together with a large number of high-level mathematical functions to operate on these arrays (Van Der Walt et al., 2011). Additionally, pandas is a powerful, open-source data analysis and manipulation package for Python. Pandas provide fast and flexible, data structures, designed to make working with relational and labelled data both intuitive and easy (Chen, 2017).

In the Python script, the mathematical formulation of the model is defined in Gurobi Optimizer format. Hardcoded data is read, pre-processed, and manipulated using the pandas library. NumPy is then utilized for numerical operations related to the problem, such as matrix operations. The performance of the Gurobi Optimizer model was eventually evaluated by three distinct generated from the concrete plant data. Figure 1 represents the routing part of the Python script of the Gurobi Optimizer model for Case 1.

```
from gurobipy import *

import numpy as np
import pandas as pd

N=5
V=10
t_model=Model()
w=[0,35,25,50,12]
d=np.zeros((N,N))
t=np.zeros((N,N))
distance_to_depot=[0,30,25,30,20]
time_to_depot=[0,30,25,30,20]

loading_time = 10  # 10 minutes loading time at the depot

for i in range(N):
   for j in range(N):
      d[i][j] = distance_to_depot[i] + distance_to_depot[j]

      if i == 0:
         # Add loading time only for trips starting from the depot
         t[i][j] = time_to_depot[j] + loading_time
      elif j == 0:
         # Decide whether to add loading time for trips returning to the depot
         # Else, no loading time for returning trips
         t[i][j] = time_to_depot[i]
      else:
         t[i][j] = time_to_depot[i] + time_to_depot[j] + loading_time

driver=1000
c=28
e=[0,60,90,120,60]
l=[0,120,150,180,120]

x=t_model.addVars(N,N,V,vtype=GRB.BINARY,name='x')
f=t_model.addVars(N,V,vtype=GRB.CONTINUOUS,name='f')
s=t_model.addVars(N,V,vtype=GRB.CONTINUOUS,name='s') #arrival time

h=15 #service_time 15 min
q=12 #q=12 m3 for each truck
M=1000 #Big M Constant
obj_fn=sum(c*d[i][j]*x[i,j,k] for i in range(N) for j in range(N) for k in range (1,V))+sum(driver*x[0,j,k] for j in range (1,N) for k in range (1,V))
t_model.setObjective(obj_fn, GRB.MINIMIZE)

c1=t_model.addConstrs((sum(x[0,j,k] for j in range(1,N))<=1) for k in range(1,V))
c2=t_model.addConstrs((sum(f[i,k]for k in range (1,V))==1)for i in range (1,N))
c3=t_model.addConstrs((w[i]*f[i,k]<=q) for i in range (1,N) for k in range (1,V))
c4=t_model.addConstrs((sum(x[j,i,k] for j in range (N))>=f[i,k])for i in range (1,N) for k in range (1,V))
c5=t_model.addConstrs((sum(x[i,j,k] for j in range(N))==sum(x[j,i,k] for j in range(N))) for k in range(1,V) for i in range (1,N))
c6=t_model.addConstrs((x[0,0,k]==0)for k in range (1,V))
c7=t_model.addConstrs((e[i]<=s[i,k])for i in range(1,N) for k in range (1,V))
c8=t_model.addConstrs((s[0,k]>=e[0])for k in range(1,V))
c9=t_model.addConstrs((s[i,k]<=l[i]) for i in range(1,N) for k in range (1,V))
c10=t_model.addConstrs((s[i,k]+t[i,j]+h-M*(1-x[i,j,k])<=s[j,k])for i in range(N) for j in range (1,N)for k in range(1,V))
c11=t_model.addConstrs((((s[i,m]-s[i,k])>=(h*(sum(x[j,i,k]+x[j,i,m] for j in range (N))-1))) for i in range (1,N) for k in range(1,V) for m in range (k+1,V))
c12 = t_model.addConstrs((sum(x[0,j,k] for j in range(1,N)) <= sum(x[0,j,k-1] for j in range(1,N))) for k in range(2,V))

t_model.optimize()
print(t_model.objVal)
```

Figure 1. Python Code of Routing Part in Gurobi Optimer Model

3.3.2. The Urgent Demand Algorithm

In exact solution methods for vehicle routing problems such as the branch and cut method, finding the optimal solution requires significant computational time, especially with a high number of customers and vehicles. Heuristic methods are commonly used for solving the large-sized NP-hard problems especially in real-life scenarios, because they are capable to compute near-optimal solutions in a reasonable time (Chen et al., 2010). Therefore, a heuristic method named the Urgent Demand Algorithm was developed for this study, inspired by well-known algorithms such as the nearest neighbour algorithm and the tabu search.

The nearest neighbour algorithm is the most natural heuristic method for the vehicle routing problem. It involves a routing process where the vehicle route proceeds from the customer nearest to the depot. Then, the vehicle visits the customer nearest to the one just visited and this process is iterated until all of the customers have joined the route (Fitriani et. al., 2021). The nearest neighbour algorithm is a very straightforward method for solving the VRPs and performs quickly, however it can occasionally miss out shorter routes due to its greedy character (Pop et.al., 2011). Introduced by Fred Glover in 1986, Tabu Search is a metaheuristic algorithm that utilizes a memory-driven technique to extend the exploration of the local search method beyond local optima. It maintains a record of previously executed moves or investigated solutions to guide its search process (Tan et.al., 2001).

The Urgent Demand Algorithm can be classified as a greedy algorithm. This algorithm was developed by involving characteristics of both tabu search and nearest neighbourhood algorithms. The algorithm selects the customer based on the earliest time which resembles the nearest neighborhood algorithm's strategy of choosing the next closest or most immediate node. Furthermore, it includes more complex features such as time windows, logical demand handling and iterative improvement, which are characteristics of the tabu search algorithm. Despite these similarities, the proposed algorithm is not fully similar to either algorithm.

The Urgent Demand Algorithm offers several advantages, including its ability to quickly generate near-optimal solutions, making it well-suited for time-sensitive tasks like ready-mix concrete delivery. The algorithm's simplicity ensures ease of implementation and fast computation. However, it has limitations when applied to more complex real-world scenarios. The algorithm assumes static demand and routing conditions, which may not account for dynamic changes such as fluctuating customer needs or traffic delays. It also doesn't address heterogeneous fleets or varying truck capacities, which can reduce its flexibility in large-scale or more complex logistics systems.

The Urgent Demand Algorithm, along with the mathematical model and hard-coded data, is implemented as a Python script. This script utilizes the NumPy and Pandas libraries for numerical operations and data processing. The algorithm's performance was evaluated by using the same three cases as in the Gurobi Optimizer model. The evaluation focused on the algorithm's ability to manage unmet demands within the customer's time windows and to create efficient routes for truck mixers in shorter times. Figure 2 represents the pseudocode of the Urgent Demand Algorithm, explaining the logic of the algorithm. The following paragraphs explain the logic and methodology of the Urgent Demand Algorithm, focusing on optimizing truck mixer routing, prioritizing timely deliveries, and managing unmet demand efficiently

First, the Urgent Demand algorithm begins by initializing variables and data structures, followed by calculating the distance, time, and demand matrices. It then lists the possible arrival times for each customer within their respective time windows. The routing process starts from the concrete plant, with each truck mixer beginning its route from the plant. If unmet demand exists and the earliest possible time for a customer exceeds a realistic threshold (e.g., more than 1000), the algorithm adjusts by reducing the time by 1000 to prioritize that customer.

For each customer, the algorithm updates the earliest service times, considering the time window constraints. It then selects the most urgent customer, defined as the one with the earliest service time, for the next visit. The algorithm calculates whether the truck mixer can arrive at the selected customer before their latest service time; if not, it skips to the next customer. If arrival within the time window is feasible, the customer is added to the truck mixer's route.

The process continues by moving on to the next customer in line. After visiting a customer, the algorithm adjusts the arrival interval for that customer and updates the possible arrival times for the remaining customers. It also updates the unmet demand based on the truck mixer's capacity and the demand of the customer just served. Additionally, if a customer's demand is fully met or the customer is deemed inaccessible at the current time, the algorithm deprioritizes them by adjusting their earliest service time. The process concludes when all customer demands are met, or no feasible routes are left.

A_i: Available arrival interval time for customer i
E_i: Available earliest time for customer i
L_i: Available latest time for customer I
U_i: Unmet demand of customer i
d_{ij}: Distance between customers i and j
t: Current time
p: Current destination
c: Candidate customer
z: Candidate arrival time
s: Service time of a truck mixer
M: Maximum number of trials for each truck mixer
R_k: Set of customers assigned to truck mixer k
S_k: Set of service start time assigned to truck mixer k
V: Number of truck mixers
C: Number of customers
Initialization:
$A_i = [E_i, L_i]$ for all customers i
$R_k = \emptyset$ for all truck mixers k
$S_k = \emptyset$ for all truck mixers k
$W_i = \min E_i$ for all customers i
Route Construction for Truck Mixers:
for k=1 to V
  if $W_i \geq 1000$ *and* $U_i > 0$ **then**
    $W_i = W_i - 1000$
  t =0
  p =depot
try = 0
  while (try<M)
      $c = \text{argmin}(W_i)$ #Calculates the closest available arrival times for each customer i
      if $U_c > 0$ *and* $t + d_{pc} \leq L_c$ **then**
        $z = \max\{t + d_{pc},\ W_c\}$
        if $z \notin A_C$ *and* $z < \max\{A_i\}$ **then**
          while $z \notin A_C$
            $z = z + 1$
        if $z \in A_C$ **then**
          $S_k = S_k \cup \{z\}$
          $R_k = R_k \cup \{c\}$
          $p = c$
          $t = z + s$
          $A_c = A_c \backslash [z - s + 1, z + s - 1]$
          $W_c = \min A_c$
          if $U_c \geq 12$ **then**
          $U_c = U_c - 12$
            $W_c = W_c + 1000$
          else
          $U_c = 0$
      else if $U_c = 0$ *and* $W_c < 1000$ **then**
        $W_c = W_c + 1000$
      else if $W_c < 1000$ **then**
        $W_c = W_c + 1000$
  try=try+1

Figure 2. Pseudocode of the Urgent Demand Algorithm

3.4. Data Generation

This section presents the data from the concrete plant and operational details. Figure 3 shows the distances between the concrete plant denoted as 0 and the customers in Case 1. Truck mixers must move in the manner illustrated in Figure 3 for each case. This plant is part of a larger chain owned by a company and operates in the Izmir province of Türkiye.

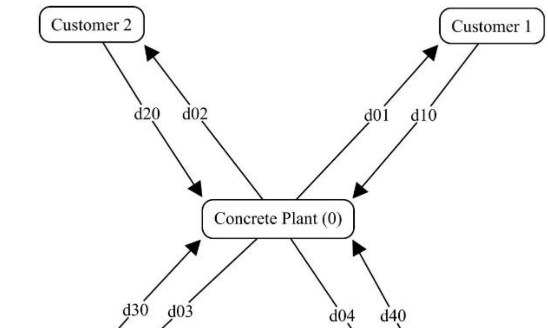

Figure 3. Network Representation of Concrete Plant in Case 1

Table 3 shows the parameters recently collected from the medium-sized concrete plant. The plant receives customer orders about a week in advance, and they're typically finalized around three days. Then, the plant makes a schedule for each work day. Due to the adverse effects of sun and heat on concrete pouring, customers generally request that the concrete be delivered during the early hours of the morning, especially before 01:00 p.m.

The concrete plant needs to source additional truck mixers from other plants during days of high customer demand. Therefore, the production rate of the plant varies depending on the customer demands and number of truck mixers available.

Table 3. Parameters of the Concrete Plant

Parameters of the Concrete Plant	Value
Daily Driver Cost (D)	1000 TRY/truck
Cost per kilometer (c)	28 TRY/km
Truck Mixer Capacity (q)	12 m^3/truck
Loading Time of Truck Mixer at the Plant	10 min
Service Time (h)	15 min
Velocity of Truck Mixer	60 km/h
Production Rate of the Plant	70 m^3/h

In Table 3, the driver cost (D) value represents the daily driver cost for operating a truck mixer and the cost per kilometer (c) value represents the costs for each truck mixer per km in terms of diesel fuel consumption. Driver cost (D) and cost per kilometer (c) values were calculated using the average monthly gross salary for truck mixer drivers and the average diesel fuel prices determined by the Republic of Türkiye's Energy Market Regulatory Authority in 2023. In this concrete plant, the fleet is standardized to utilize exclusively a single type of truck mixer, which has a capacity of 12 m3. It was assumed that the service time (h) for transferring concrete, whether by conveyor belt or concrete pump, is 15 minutes for all types. The velocity of the truck mixer was determined based on the average velocity of the trucks in delivery operations at the concrete plant. The production rate was determined based on the theoretical maximum production of concrete at the plant.

Table 4. Time Windows and Demands for each Customer in Case 1

Information about 4 Customers					
Customer No	**Demand (m^3)**	**Earliest Time Value**	**Latest Time Value**	**Earliest Time (e_i)**	**Latest Time (l_i)**
1	35	60.0	120.0	07:00	08:00
2	25	90.0	150.0	07:30	08:30
3	50	120.0	180.0	08:00	09:00
4	12	60.0	120.0	07:00	08:00

In Table 4, the earliest time (ei) and latest time (li) indicate time windows. The earliest and latest time values were converted into the HH:MM format by using the base time of 06:00 a.m. plus the values in minutes (06:00 + x min= HH:MM). For Customer 1, the earliest time value of 60.0 minutes was converted to 07:00 a.m., and the latest time value of 120.0 minutes was converted to 08:00 a.m. Time windows for all customers were set to 1 hour, since the concrete begins to set within 60 minutes, depending on the type of cement and environmental conditions.

Table 5. Distance Matrix in Case 1

d_{ij} (km) (j) (i)	0	1	2	3	4
0	0	30	25	30	20
1	30	0	55	60	50
2	25	55	0	55	45
3	30	60	55	0	50
4	20	50	45	50	0

Table 5 shows the distances between the nodes including customers and the concrete plant. The plant serves customers within a maximum distance of 30 kilometers.

Table 6. Time Matrix in Case 1

t_{ij} (min) (j) (i)	0	1	2	3	4
0	0	40	35	40	30
1	30	0	65	70	60
2	25	65	0	65	55
3	30	70	65	0	60
4	20	60	55	60	0

Table 6 shows the travel times between the nodes including customers and the concrete plant. Travel times were calculated by dividing the distances listed in Table 4.3 by an average truck mixer velocity of 60 km/h and adding a 10-minute loading time of the truck mixer at the plant. However, for the travel times of customers returning to the plant, the 10-minute loading time was not added, as there is no loading involved in these return trips.

4. Experimental Results

In this section, the results for three cases were obtained using the Gurobi Optimizer and the Urgent Demand algorithm on the specified computer. The Python script for the exact solution was executed with the Gurobi Optimizer, version 10.0.3 (build v10.0.3rc0 for Windows 64-bit). The system's CPU was Intel(R) Core(TM) i7-10750H, operating at 2.60 GHz, and supported the instruction set [SSE2|AVX|AVX2].

4.1. Results of Case 1 from Gurobi Optimizer

This section presents the optimal solution for Case 1 including route details and delivery schedule

Table 7. Route Details for Case 1 from Gurobi Optimizer

Route Details					
Route Number	**Truck Number**	**Route**	**Route Duration(min)**	**Route Cost (TRY)**	**Total Cost (TRY)**
1	1	0-3-0	85	2680	26760
2	2	0-3-0	85	2680	
3	3	0-1-0-3-0	170	4360	
4	4	0-1-0-3-0	170	4360	
5	5	0-2-0-3-0	160	4080	
6	6	0-4-0-2-0	140	3520	
7	7	0-1-0	85	2680	
8	8	0-2-0	75	2400	

Table 7 shows that the Gurobi Optimizer model used 8 of 10 truck mixers in the optimal solution of Case 1. For example, truck mixer 6 which followed route 6, moved from the concrete plant (0) to customer 4 after being loaded with concrete, then returned to the plant for getting concrete and subsequently moved to customer 2 and returned to the plant. The total cost of route 6 is 3520 TRY. The route cost for a truck mixer includes the total distance travelled and the daily driver cost assigned to the truck mixer. The total duration of route 6 is 140 minutes. The total duration of a route includes the travel time between stations, truck service times at customers, and the loading time of the truck at the plant. The total cost of the optimal solution is 26760 TRY, calculated by summing all the route costs.

Table 8. Delivery Schedule for Case 1 from Gurobi Optimizer

Delivery Schedule					
Customer Number	**Demand (m³)**	**Truck Number**	**Percentage of Demand**	**Split Delivery Amount (m³)**	**Service Start Time (a.m.)**
1	35	3	31.43%	11	07:00
		4	34.29%	12	07:15
		7	34.29%	12	08:00
2	25	5	48.00%	12	07:30
		6	4.00%	1	08:10
		8	48.00%	12	08:30
3	50	1	4.00%	2	08:00
		2	24.00%	12	08:15
		3	24.00%	12	08:30
		4	24.00%	12	08:45
		5	24.00%	12	09:00
4	12	6	100.00%	12	07:00

Table 8 shows the distribution of customer demands and the percentage of customer demands that each truck mixer delivers to them in the optimal solution of Case 1. For example, the demand of customer 1, amounting to 35 m^3 was supplied by trucks 3, 4, and 7 respectively through split deliveries. Truck 3 started service at customer 1 at 07:00 a.m. and delivered 11 m^3 of concrete which constituted 31.43% of customer 1's total demand. After the completion of service of truck 3, truck 4 started service at 7:15 a.m., followed by truck 7 at 08:00 a.m.

4.2. Summary of Results

In this study, the Gurobi Optimizer was utilized as the exact method and provided optimal solutions, while the proposed heuristic method, the Urgent Demand algorithm, provided near-optimal solutions. These two methods were evaluated across three cases in terms of the total cost of solutions, computation times, and number of trucks used.

Table 9. Total Costs and Computation Times of Solutions Overview

	Overview of Total Costs of Solutions and Computation Times								
	4 Customers and 10 Trucks Available (Case 1)			10 Customers and 15 Trucks Available (Case 2)			15 Customers and 15 Trucks Available (Case 3)		
Solution Method	**Total Cost (TRY)**	**Number of Routes**	**Computation Time (s)**	**Total Cost (TRY)**	**Number of Routes**	**Computation Time (s)**	**Total Cost (TRY)**	**Number of Routes**	**Computation Time (s)**
Gurobi Optimizer	26760	8	1.5	51160	10	15.0	64040	10	310170.0
Urgent Demand Algorithm	26760	8	1.0	53160	12	1.2	68040	14	1.5

Table 9 shows the total cost of solutions and computation times for the solution methods for each case. In Case 1, both the Gurobi Optimizer and the Urgent Demand algorithm provided equally cost-effective solutions, each with a total cost of 26760 TRY for 8 routes. Both solution methods performed well in this case with close computation times (1.5 seconds and 1.0 seconds, respectively). In Case 2, Gurobi Optimizer provided a more cost-effective solution with a total cost of 51160 TRY for 10 routes, as compared to the Urgent Demand algorithm, which provided a solution with a total cost of 53160 TRY for 12 routes. Notably, the Urgent Demand algorithm performed in a lower computation time (1.2 seconds). In Case 3, Gurobi Optimizer again provided a more cost-effective solution with a lower total cost of 64040 TRY for 10 routes, as compared to the Urgent Demand algorithm, which provided a solution with a total cost of 68040 TRY for 14 routes. However, the Gurobi Optimizer's computation time was substantially higher at 310170.0 seconds, as compared to the Urgent Demand Algorithm, which solved the problem in only 1.5 seconds. It is important to note that in a solution, each route was followed by a single truck.

The total cost of a solution was then calculated by summing the costs of all the routes. These route costs included the total distance travelled and the driver cost.

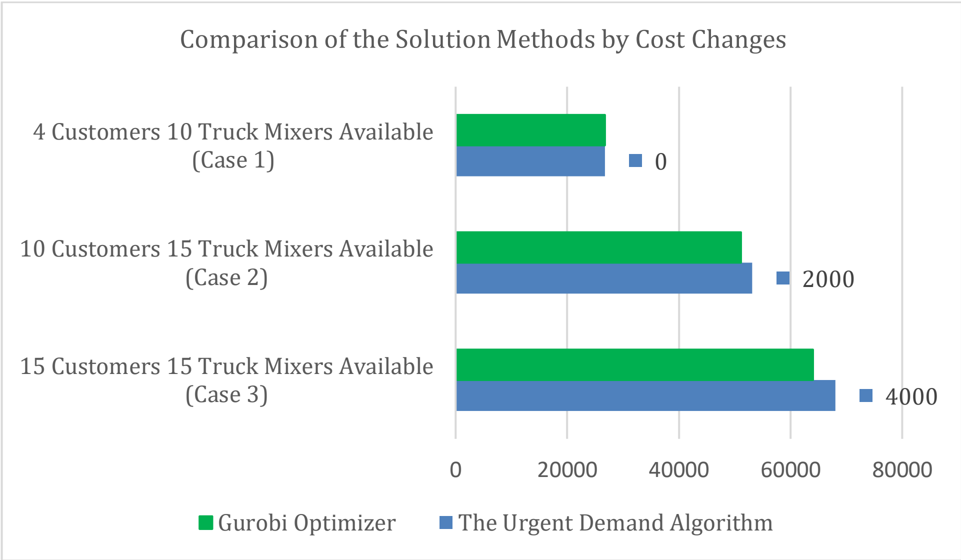

Figure 4. Comparison of the Solution Methods by Cost Change

Figure 4 shows the cost change between the optimal solution and the near-optimal solution for each case. In Case 1, the Urgent Demand Algorithm provided a near-optimal result with no increase in cost, indicating a 0% increase. In contrast, Case 2 demonstrated a different outcome where the algorithm provided the near-optimal solution but with an additional cost of 2000 TRY, representing a 3.91% increase in cost of the solution. Similarly, in Case 3, the algorithm provided the near-optimal solution; however, this was accompanied by a 4000 TRY increase in cost, equating to a 6.25% increase in cost of the solution. In this analysis, the Urgent Demand algorithm produced near-optimal solutions, with an average cost increase of 3.39% as calculated by averaging the percentage increases of total cost in all three cases.

A fourth case is also developed to evaluate the scalability and applicability of the Urgent Demand Algorithm in a larger logistics scenario than typically seen in concrete delivery operations. While the standard problem does not involve such a large number of customers, simulating a scenario where a concrete plant serves 30 customers with 30 trucks available allows for a comprehensive assessment of the algorithm's ability to manage increased complexity in routing and fleet coordination. This experiment demonstrates the algorithm's potential adaptability to other logistics sectors, such as, food and beverage distribution, and medical supply logistics, where optimizing large fleets and adhering to strict time windows are critical. By testing the algorithm in a more demanding environment, this study provides insights into its potential for broader application in various logistics operations facing similar challenges, including real-time decision-making and route efficiency optimization. In case 4, the Urgent Demand algorithm utilized 22 out of 30 available trucks to service 30 customers, constructing 22 optimized routes with a total cost of 145480 TRY in just 2.0 seconds. While the Urgent Demand algorithm demonstrates promising efficiency in its current form, further development and refinement could enhance its ability to handle more complex routing problems, making it even more suitable for real-time applications in larger-scale operations.

Average cost increase of 3.39%, observed in the The Urgent Demand Algorithm is considered acceptable for several reasons, particularly in the context of ready-mix concrete transportation. This marginal increase allows for flexibility in handling real-time operational challenges, such as unexpected traffic conditions, changes in customer demands, or urgent deliveries. The Urgent Demand Algorithm provides near-optimal solutions quickly, making it particularly suitable for time-sensitive applications like concrete delivery, where even small delays can lead to significant project disruptions or material spoilage.

5. Result and Discussion

This paper has systematically investigated optimizing the transportation of ready-mixed concrete (RMC) within the context of logistics and supply chain management in the construction sector. The primary goal was to propose a truck mixer routing model to improve the efficiency of concrete delivery operations in terms of cost-effectiveness and efficient routing. In the process of formulating the problem into the mathematical model, it was formulated as a mixed-integer linear programming (MILP) model based on the capacitated vehicle routing problem with time windows (CVRPTW). To address the unique challenges of RMC transportation, CVRPTW was modified with split

delivery constraints, multi-trip characteristics, and other constraints. This model was addressed by employing both an exact method and a heuristic method. Firstly, the model was addressed by the exact method, the Gurobi Optimizer, which uses the branch and cut method. Given that the vehicle routing problem (VRP) is NP-hard, the complexity of the model increased with more customers and vehicles, which led to longer computation times in solving the model. Therefore, a heuristic method named as Urgent Demand Algorithm was developed for the problem by inspiring the characteristics of tabu search and nearest neighbour algorithm. The data was collected from a concrete plant operating in Turkey. Three individual cases were generated from this data to evaluate the performance of the solution methods. In the first case, there were 4 customers and 10 truck mixers available at the plant. Then, in the second case, there were 10 customers and 15 truck mixers available. Lastly, for the third case, there were 15 customers and 15 truck mixers available.
Gurobi Optimizer provided the optimal solutions for all cases in a reasonable computation time. However, in the third case, it was observed that the computation time required was significantly longer (310170.0 seconds) compared to the first and second cases. In contrast, the proposed heuristic provided near-optimal solutions rapidly (1.2 seconds on average) for all cases with about a 3.39% average cost increase. In the routes derived from the Gurobi Optimizer solutions, the use of truck mixers was lower compared to the proposed heuristic method. These results showed that the proposed heuristic algorithm was an effective alternative to the exact methodologies for solving the problem, especially large-scale problems. This study has revealed that the proposed model and its solution methods effectively constructed routes for truck mixers, being time-effective and cost-effective in the transportation of concrete.

This study provided new perspectives on filling the gap between theoretical vehicle routing problems and practical applications in real-world logistical problems of the construction sector. It showed the potential of advanced optimization techniques to enhance efficiency and sustainability in construction logistic solutions. Practically, it proposed a robust truck routing model and solution methodologies for ready-mixed concrete suppliers. This model routed the truck mixers efficiently and enhanced concrete delivery efficiency in terms of fuel-oil and driver costs. The model also contributed to environmental sustainability by minimizing fuel consumption, which indirectly reduced vehicle emissions.

While this study successfully developed a model, there were certain limitations particularly related to computational feasibility for large-scale problems and assumptions made to simplify complex real-world problems. These assumptions included uniform service time, homogeneous fleet, same concrete strength class, and ideal traffic conditions. The proposed heuristic method provided near-optimal solutions rapidly for the problem, but it slightly increased the cost of the solutions. These limitations underlined the necessity for further refinement in methodological approaches and expanding the model's scalability. Despite these limitations, the model and its heuristic approach can be adapted and applied to other industries where logistics optimization is crucial. For instance, the algorithm could be beneficial in food and beverage distribution, where perishability and time-sensitive deliveries are key factors. Moreover, industries like medical supply logistics, where timely delivery of critical supplies is essential, could also leverage the model's adaptability to handle urgent demands under time constraints. These applications demonstrate the model's flexibility in addressing complex routing challenges across various sector

Given the results of this study were promising, this study was limited by assumptions and its reliance on the concrete plant data. Future research should focus on enhancing the scalability of the truck mixer routing model and address the existing limitations of concrete delivery. Key focus areas should include adapting the model for larger and more complex cases, such as integrating real-time dynamic routing and traffic conditions. Also, incorporating environmental sustainability factors could improve the model's role in sustainable logistics. Exploring advanced heuristic algorithms and machine learning techniques for predictive routing could improve computational efficiency. Additionally, the model’s applicability should be investigated in other time-sensitive industries, urban logistics, and smart city development. Lastly, understanding the impact of customer satisfaction on routing efficiency and adapting the model to accommodate various concrete types and strengths would be important for its broader application and effectiveness.

This research not only provided a novel approach to the specific problem of RMC transportation but also established a precedent for future studies in the broader disciplines of supply chain management and logistics, particularly in construction logistics. It underlined the importance of balancing efficiency, cost, and practical applicability in complex logistical operations.

Conflict of Interest

No conflict of interest was declared by the authors.

References

Aydemir, E., Karagül, K., & Tokat, S. (2016). KAPASİTE KISITLI ARAÇ ROTALAMA PROBLEMLERİNDE BAŞLANGIÇ ROTALARININ KURULMASI İÇİN YENİ BİR ALGORİTMA. Mühendislik Bilimleri Ve Tasarım Dergisi, 4(3), 215-226. https://doi.org/10.21923/jesd.60313

Baldacci, R., Mingozzi, A., & Roberti, R. (2012). Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. European Journal of Operational Research, 218(1), 1-6. https://doi.org/10.1016/j.ejor.2011.07.037

Borcinova, Z. (2017). Two models of the capacitated vehicle routing problem. Croatian Operational Research Review, 463-469. https://doi.org/10.17535/crorr.2017.0029

Brandao, J. (2006). A new tabu search algorithm for the vehicle routing problem with backhauls. European Journal of Operational Research, 173(2), 540-555. https://doi.org/10.1016/j.ejor.2005.01.042

Bräysy, O., & Gendreau, M. (2005). Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. Transp. Sci., 39, 104-118. https://doi.org/10.1287/trsc.1030.0056

Bruno, L. (2019). Solving a food-delivery problem with a Vehicle Routing Problem-based approach (Doctoral dissertation, Politecnico di Torino). http://webthesis.biblio.polito.it/id/eprint/10456

Chen, D. Y. (2017). Pandas for everyone: Python data analysis. Addison-Wesley Professional.

Chen, S., Golden, B., & Wasil, E. (2007). The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. Networks: An International Journal, 49(4), 318-329. https://doi.org/10.1002/net.20181

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations research, 12(4), 568-581. https://doi.org/10.1287/opre.12.4.568

Cook, W., & Rich, J. L. (1999). A parallel cutting-plane algorithm for the vehicle routing problem with time windows. https://hdl.handle.net/1911/101910

Cordeau, J. F., & Laporte, G. (2005). Tabu search heuristics for the vehicle routing problem (pp. 145-163). Springer US. https://doi.org/10.1007/0-387-24977-X_9

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. Management science, 6(1), 80-91. https://doi.org/10.1287/mnsc.6.1.80

Dönmez, O. A. (2024). Optimizing ready-mixed concrete transportation by a truck mixer routing model for concrete plants [Master’s thesis, Yaşar University]. Council of Higher Education (YÖK) Thesis Center (Thesis No. 874566). https://tez.yok.gov.tr/UlusalTezMerkezi

El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. Journal of King Saud University-Science, 22(3), 123-131. https://doi.org/10.1016/j.jksus.2010.03.002

Energy Market Regulatory Authority. (2023). Akaryakıt Fiyatları. EPDK. https://www.epdk.gov.tr/Detay/Icerik/21-48/akaryakit-fiyatlari

Erdoğan, S., & Miller-Hooks, E. (2012). A green vehicle routing problem. Transportation research part E: logistics and transportation review, 48(1), 100-114. https://doi.org/10.1016/j.tre.2011.08.001

Fitriani, N. A., Pratama, R. A., Zahro, S., Utomo, P. H., & Martini, T. S. (2021). Solving capacitated vehicle routing problem using saving matrix, sequential insertion, and nearest neighbor of product ‘X’in Grobogan district. https://doi.org/10.1063/5.0039295

Fukasawa, R., Longo, H., Lysgaard, J., Aragão, M. P. D., Reis, M., Uchoa, E., & Werneck, R. F. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. Mathematical programming, 106, 491-511. https://doi.org/10.1007/s10107-005-0644-x

Gursel, A. P., Masanet, E., Horvath, A., & Stadel, A. (2014). Life-cycle inventory analysis of concrete production: A critical review. Cement and Concrete Composites, 51, 38-48. https://doi.org/10.1016/j.cemconcomp.2014.03.005

Keskintürk, T., Topuk, N., & Özyeşil, O. (2015). Araç rotalama problemleri ve çözüm yöntemleri. İşletme Bilimi Dergisi, 3(2), 77-107. https://dergipark.org.tr/en/pub/jobs/issue/22921/245436

Kinable, J., Wauters, T., & Berghe, G. V. (2014). The concrete delivery problem. Computers & Operations Research, 48, 53-68. https://doi.org/10.1016/j.cor.2014.02.008

Kumar, S. N., & Panneerselvam, R. (2012). A survey on the vehicle routing problem and its variants. https://doi.org/10.4236/iim.2012.43010

Laporte, G., Gendreau, M., Potvin, J. Y., & Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. International transactions in operational research, 7(4-5), 285-300. https://doi.org/10.1111/j.1475-3995.2000.tb00200.x

Lenstra, J. K., & Kan, A. R. (1981). Complexity of vehicle routing and scheduling problems. Networks, 11(2), 221-227. https://doi.org/10.1002/net.3230110211

Li, X. (2015). Capacitated vehicle routing problem with time windows: a case study on pickup of dietary products in nonprofit organization. Arizona State University. https://keep.lib.asu.edu/_flysystem/fedora/c7/142371/Li_asu_0010N_15360.pdf

Lin, C., Choy, K. L., Ho, G. T., Chung, S. H., & Lam, H. Y. (2014). Survey of green vehicle routing problem: past and future trends. Expert systems with applications, 41(4), 1118-1138. https://doi.org/10.1016/j.eswa.2013.07.107

Linderoth, J. T., & Lodi, A. (2010). MILP software. Wiley encyclopedia of operations research and management science, 5, 3239-3248.

Lysgaard, J., Letchford, A. N., & Eglese, R. W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. Mathematical programming, 100, 423-445. https://doi.org/10.1007/s10107-003-0481-8

Mitchell, J. E. (2002). Branch-and-cut algorithms for combinatorial optimization problems. Handbook of applied optimization, 1(1), 65-77.

Nazif, H., & Lee, L. S. (2012). Optimised crossover genetic algorithm for capacitated vehicle routing problem. Applied Mathematical Modelling, 36(5), 2110-2117. https://doi.org/10.1016/j.apm.2011.08.010

Nurcahyo, G. W., Alias, R. A., Shamsuddin, S. M., & Sap, M. N. M. (2002). Sweep algorithm in vehicle routing problem for public

transport. Jurnal Antarabangsa Teknologi Maklumat, 2, 51-64.
Özaydın, E. (2003). Capacitated vehicle routing problem with time windows (Doctoral dissertation). https://research.sabanciuniv.edu/id/eprint/8533
Pichpibul, T., & Kawtummachai, R. (2012). An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem. ScienceAsia, 38(3), 307-318. https://doi.org/10.2306/scienceasia1513-1874.2012.38.307
Pop, P. C., Zelina, I., LupÅŸe, V., Sitar, C. P., & Chira, C. (2011). Heuristic algorithms for solving the generalized vehicle routing problem. International Journal of Computers Communications & Control, 6(1), 158-165. https://www.univagora.ro/jour/index.php/ijccc/article/view/2210
Schmid, V., Doerner, K. F., Hartl, R. F., &tas Salazar-González, J. J. (2010). Hybridization of very large neighborhood search for ready-mixed concrete delivery problems. Computers & operations research, 37(3), 559-574. https://doi.org/10.1016/j.cor.2008.07.010
Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research, 35(2), 254-265.
Şahin, Y., & Eroğlu, A. (2015). SİPARİŞ TOPLAMA VE KAPASİTE KISITLI ARAÇ ROTALAMA PROBLEMLERİNİN HİYERARŞİK ÇÖZÜMÜ. Mühendislik Bilimleri Ve Tasarım Dergisi, 3(1), 15-28.
Tan, K. C., Lee, L. H., Zhu, Q. L., & Ou, K. (2001). Heuristic methods for vehicle routing problem with time windows. Artificial intelligence in Engineering, 15(3), 281-295. https://doi.org/10.1016/S0954-1810(01)00005-X
Toth, P., & Vigo, D. (2002). Models, relaxations and exact approaches for the capacitated vehicle routing problem. Discrete Applied Mathematics, 123(1-3), 487-512. https://doi.org/10.1016/S0166-218X(01)00351-1
Toth, P., & Vigo, D. (Eds.). (2014). Vehicle routing: problems, methods, and applications. Society for industrial and applied mathematics.
Ünsal, Ö., & Yiğit, T. (2018). YAPAY ZEKA VE KÜMELEME TEKNİKLERİ KULLANILARAK GELİŞTİRİLEN YÖNTEM İLE OKUL SERVİSİ ROTALAMA PROBLEMİNİN OPTİMİZASYONU. Mühendislik Bilimleri Ve Tasarım Dergisi, 6(1), 7-20. https://doi.org/10.21923/jesd.340220
Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. Computing in science & engineering, 13(2), 22-30. https://doi.org/10.1109/MCSE.2011.37