Generating Realistic Benchmarks for Dynamic Truck and Trailer Scheduling using Gaussian Copulas

Joan Alza
j.alza-santos@rgu.ac.uk
Robert Gordon University
Aberdeen, Scotland, UK

Josu Ceberio
University of the Basque Country (UPV/EHU)
Donostia, Basque Country, Spain

Mark Bartlett
Robert Gordon University
Aberdeen, Scotland, UK

John McCall
Robert Gordon University
Aberdeen, Scotland, UK

Abstract

Academic research in dynamic optimisation uses benchmark generators to artificially simulate controlled and reproducible changing-environments to systematically compare algorithmic performance under uncertainty. However, due to the scarcity or difficulty in acquiring real-world data, benchmarks often fail to incorporate real-world features, such as problem constraints or the time-linkage property, where previously made decisions influence future events. This study introduces a Gaussian Copula-based real-world data-driven synthetic data generation model for Dynamic Truck and Trailer Scheduling Problem (DTTSP). The model offers a realistic, privacy-preserving DTTSP benchmark instance generator, which can be used to recreate the dynamism, constraints, heterogeneity, and time-linkage of logistics and supply chain operations. This work examines the utility, fidelity, and privacy of the suggested model in four workday case studies from a local transportation company. The conducted experiments demonstrate the systematical application of Gaussian Copulas to produce accurate, useful, and secure DTTSP benchmark instances that capture the statistical properties and correlation of variables, as well as the temporal patterns, in the original annual data. Nevertheless, the utility analysis of the conditional sampling indicates that there is still room for improvement in the modelling process.

CCS Concepts: • **Computing methodologies → Modeling methodologies**; • **Mathematics of computing → Multivariate statistics**; • **Applied computing → Operations research**.

Keywords: Evolutionary Computation, Dynamic Optimisation Problems, Scheduling, Benchmark Generation, Synthetic Data Generation, Gaussian Copula

FOGA '25, Leiden, Netherlands

ACM ISBN 979-8-4007-1859-5/2025/08
https://doi.org/10.1145/3729878.3746620

ACM Reference Format:
Joan Alza, Josu Ceberio, Mark Bartlett, and John McCall. 2025. Generating Realistic Benchmarks for Dynamic Truck and Trailer Scheduling using Gaussian Copulas. In *Foundations of Genetic Algorithms XVIII (FOGA '25), August 27–29, 2025, Leiden, Netherlands.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3729878.3746620

1 Introduction

Evolutionary Algorithms (EAs) have been widely applied to address complex real-world optimisation problems, such as supply chain management [38], task scheduling [6], and vehicle routing [48]. These optimisation problems are often dynamic, characterised by changes in objectives, constraints, and variables [34]. For example, in Dynamic Vehicle Routing Problems (DVRP), algorithms need to react to changing traffic conditions and new delivery requests [45]. Problems with time-dependent parameters are generally referred to as Dynamic Optimisation Problems (DOPs) in the field of optimisation [13, 34].

Research on dynamic optimisation focuses on understanding the nature of problem-changes and developing algorithms that can effectively track the changing optima [35]. DOPs can be classified based on influence of dynamic features in algorithmic performance [52, 54]. Dynamic features include the time-linkage (i.e. whether past solutions influence future problem states [4]), predictability, visibility, and frequency and severity of changes. Other defining features include the nature of the optimisation problem (e.g. constrained or unconstrained, single- or multi-objective) and the factor of change (i.e. changing objective function, variables, or constraints).

In order to systematically research dynamic features and their influence on algorithm behaviour, the research community has developed standardised benchmark generators. These tools generate a range of dynamic problem-solving scenarios, in a controllable and reproducible way, providing the means to rigorously evaluate and compare the performance of dynamic optimisation algorithms [6, 18, 29, 44, 50]. Typically, these generators simulate dynamism by producing a sequence of optimisation problem instances, where each successive instance is derived from its predecessor by controlled changes to features of the previous instance. Most commonly, empirical DOP analysis focuses on comparing algorithmic performance across generated benchmarks that systematically vary the frequency and magnitude of feature change [2, 39].

Despite the performance of dynamic optimisation algorithms has been thoroughly explored for dynamic features (e.g. objective function change), existing benchmarks often fail to capture real-world features, such as constraints and the time-linkage property [34]. For instance, the Moving Peaks Benchmark (MPB) [6, 7] and the fitness landscape rotation [29, 50] are frequently used for empirically analysing algorithmic performance in the dynamic context due to their simple implementation and controllable parameter setting, which allows adjusting the dynamism of the problem. These generators, however, are generally oversimplified, as they fail to capture domain-specific dynamism (e.g. time-linkage) and complex constraints [35, 52], highlighting the research gap between their academic utility and the need to address practical real-world optimisation scenarios.

This study presents a data-driven benchmark instance generator for a realistic implementation of the Dynamic Truck and Trailer Scheduling Problem (DTTSP), an NP-hard combinatorial problem that is found in supply chain, logistics, and freight transportation planning applications [37, 46]. The DTTSP, which extends the DVRP, intrinsically poses constraints related to assignment, temporality, and resource compatibility [37]. The aim of DTTSP is to optimise the job allocation to a heterogeneous fleet of trucks and trailers, minimising the number of vehicles used, and thus, improving operational efficiency over time [38]. The dynamism is derived from the arrival of new jobs, which increase problem dimensionality and may lead to infeasible solutions, requiring adaptive optimisation strategies. Moreover, due to the dynamic and highly-constrained nature of the problem, the benchmark generator allows producing instances with real-world complexities, including demand variability, the time-linkage property, and problem-specific constraints, such as resource availability and capacity, or required skills for operations.

The proposed synthetic benchmark generation process consists of the following steps. First, the real-world data is analysed to identify key statistical properties and dependencies between variables. Second, a synthetic data generation model (i.e. a Gaussian Copula [40]) is fitted to the real-world data to estimate the multivariate correlation structure and marginal distributions. Third, synthetic datasets are generated by using a conditional sampling from the fitted model. Fourth, the generated datasets are evaluated using statistical metrics based on the fidelity, utility, and privacy of the generated synthetic data over the original data. Finally, the generated benchmark instances are tested to simulate and optimise the dynamic decision-making process of a typical working day at ARR Craib Ltd, the logistics company based in the North-East Scotland that provided us with the data.

The obtained results demonstrate a high fidelity of the suggested Gaussian Copula model to capture multivariate dependencies and statistical properties of the real-world annual data, showing minor discrepancies in capturing the temporal patterns. Nevertheless, the conditional sampling results suggest that the method used should be improved to make instance generation more accurate. Furthermore, the utility analysis and the application of the instances on the simulator suggests that the model generates accurate synthetic data, closely preserving the functional characteristics and practical applicability of the original data while ensuring compliance with data privacy regulations. Finally, the performed data scarcity analysis suggests that promising benchmarking results can still be achieved with a 90% data reduction.

The main contribution of this work is the use of synthetic data generation on a real-world logistics dataset to generate new dynamic DTTSP instances, combining flexible parametric configuration and realism in benchmark design. Furthermore, the controllable parameters of the synthetic data generation process allow for a diversified benchmark design, enabling researchers to systematically evaluate the impact of different problem characteristics on the algorithmic performance and selection under dynamic conditions.

The rest of this paper is structured as follows. Section 2 reviews the literature about the existing research on benchmark generators for DOPs. Section 3 describes the problem and the essential variables in the data for the generation of synthetic instances, and analyses some of its properties and hidden patterns. Section 4 presents the used synthetic data generation model and evaluation metrics, showing their validation and functionality steps. Section 5 performs the essential statistical utility and fidelity analysis of synthetic data, ensuring it reflects the patterns and dependencies of the original data. Section 6 transparently documents the ethical and legal implications related to the data acquisition and the synthetic data generation process. Section 7 concludes the paper and suggests future work.

2 Background and Related Work

Over the years, academic research in dynamic optimisation has used a wide range of benchmark generators for creating DOP instances [18, 35]. Many of these approaches focus on the continuous space, where mathematical functions change over time by varying certain parameters, such as the frequency and magnitude of changes [21, 39, 52].

One of the first benchmark generator, and probably the most popular benchmark generator, is the Moving Peaks Benchmark (MPB) [6, 7], which changes the height, width, and position of peaks in continuous space to simulate dynamic environments. Similarly, other benchmark generators have been also suggested to simulate different dynamics, such as fitness landscape rotation, which uses the Exclusive-OR (XOR) operator to change the mapping between binary solutions and objective values [44, 50]. In order to extend the generality of these benchmarks, the Generalised Dynamic Benchmark Generator (GDBG) further combines landscape shifting, rotation, and dynamic rules to control the nature of changes in real, combinatorial, and binary spaces [22, 23].

Motivated by the need for more complex benchmarks, the dynamic optimisation community has moved towards more sophisticated frameworks [35]. For example, the Generalised Moving Peaks Benchmark (GMPB) exemplifies the expansion of the original MPB by adding complex variable interactions and various types of dynamisms [53]. Recently, the Dynamic Dataset Generator (DDG) [51] has been introduced as a method that introduces controllable variations in Gaussian components, offering flexibility in simulating dynamic clustering scenarios.

Moreover, the literature also presents some scenario-specific generators to model real-world dynamics by altering certain variables of the problem. These generators include the Dynamic Knapsack Problem (DKP) with varying item profits and capacities [54], the Dynamic Travelling Salesperson Problem (DTSP) with traffic [45], and the DVRP with time windows or pickup and delivery tasks [24, 32]. Moreover, a more realistic extension of the DVRP focuses on the dynamic routing for Mobile Ad hoc Networks (MANETs) [9, 26], where node mobility and communication links changes over time. Similarly, the Dynamic Capacitated Arc Routing Problem (DCARP) has been studied by the community, where the demand and road conditions change over time [27, 31]. Despite these approaches offering enhanced contextual relevance, they remain limited in their capacity to extrapolate research finding to practical applications [35].

Nguyen identifies constraints and time-linkage properties as features found in many real-world DOPs [34]. Cruz et al. [10] listed different dynamic data-driven benchmarks from real-world applications, covering a range from evolutionary robotics to aerospace design. However, due to data privacy concerns, the reproducibility of most of the listed works is limited. Similarly, Yang et al. [49] presented an extensive list of approaches for dynamic combinatorial optimisation problems. However, as pointed out by several authors, the field of dynamic optimisation still requires realistic benchmarks that are easy to implement, as well as general and flexible to simulate different types of dynamism [35, 49].

Data analysts, as well as the machine learning and computer vision communities, have shown an increasing interest in synthetic data generation models, which aim to replicate complex statistical properties of real-world data [12, 16, 19, 30]. Synthetic data generation models, such as Generative Adversarial Networks (GANs) [15] and Diffusion Models [25], learn the statistical properties and patterns of the original data, and then, they generate new instances by sampling from this learned distribution [19]. These models are highly useful, especially when real data is unavailable or limited by privacy or ethical restrictions. The main benefit in producing synthetic data is the possibility to share protected data with statistical characteristics of real-world [16].

Therefore, synthetic data generation models hold considerable promise for benchmarking in dynamic optimisation. For example, GANs have proven efficient to synthetically generate time series data, which could then be used to reproduce the real-world dynamics using an agent-based modelling framework [3, 8].

In conclusion, this work contributes to bridging gaps identified in dynamic optimisation benchmarking, which can be summarised as follows:

1. Benchmark generators for DOPs should incorporate a realistic, flexible, and reproducible implementation, in addition to a simple design. However, most academic benchmark generators are not demonstrably reflective of dynamic features of real-world applications.
2. The application of advanced synthetic data generation models for dynamic benchmark generation remains largely underexplored.

3 Problem Description and Data Analysis

This section describes the considered real-world problem, as well as the structure, key statistical properties, patterns, and dependencies of the donated datasets. In particular, this study investigates the Dynamic Truck and Trailer Scheduling Problem (DTTSP), formulated as a variation of a real-world, constrained, uncertain, and heterogeneous vehicle routing problem with pickup and delivery tasks [24]. In order to formally represent and solve the DTTSP, a mixed-integer linear programming model has been used, with the objective of maximising vehicle fleet efficiency by optimising vehicle schedules to meet customer deadlines and minimising transportation costs [38, 48]. It is worth noting that the DTTSP, as other scheduling problems, is classified as NP-hard [14].

3.1 Truck and Trailer Scheduling Problem

Let the working hours of the company be determined by W_s and W_e, which represent the opening and closing times of the company, respectively.

Let $J = \{J_1, \ldots, J_n\}$ be a set of jobs, where each job J_i contains a collect time J_i^c and a delivery time J_i^d, an expected execution time J_i^s, and the start time J_i^t. The distance between two successive jobs, J_i and J_{i+1}, is denoted by $\delta_{i,i+1}$.

Let $T = \{T_1, \ldots, T_m\}$ be a set of trailers, and a trailer T_j is meant to be subcontracted by the notation $T_j^s = 1$. Similarly, let $L = \{L_1, \ldots, L_l\}$ be a set of trucks, where a truck is subcontracted when $L_k^s = 1$. Let us define the function $\mathcal{T}(L_k)$ to return the trailer associated with a truck L_k. Rigid trucks (i.e. trucks with a trailer constantly attached to them) are denoted as $L_k^r = 1$, which means that a trailer is associated with the truck L_k. Similarly, $C_{J_i,T_j} = 1$ shows that J_i fits in the trailer T_j according to its capacity.

The distance travelled between a trailer T_j and a job J_i is represented as δ_{J_i,T_j}, and the distance travelled between a trailer T_j and a truck L_k is indicated as δ_{T_j,L_k}. Note that the travel time between a job and a trailer, as well as the time between a trailer and a truck, is based on the travel distance and the speed of the trailer type. The function $\tau(J_i, T_j)$ represents the travel time between job J_i and trailer T_j, and $\tau(T_j, L_k)$ returns the travel time between a trailer T_j and a truck L_k, respectively.

Let $S = \{S_1, \ldots, S_w\}$ be the set of *driver-skills* required to perform certain jobs. In order to deal with driver-skills constraints, the function $\mathcal{S}^{job}(J_i, S_s) = 1$ determines whether the job J_i requires the skill S_s, and $\mathcal{S}^{truck}(L_k, S_s) = 1$ if the driver assigned to the truck L_k has the skill S_s.

Finally, the decision variables $X_{J_i,T_j} = 1$ and $Y_{J_i,L_k} = 1$ represent if the trailer T_j and the truck L_k are assigned to job J_i, respectively.

The objective function can be formulated as follows:

Minimise:

$$\sum_{i=1}^{n} \max\left\{ \sum_{j=1}^{m} X_{J_i,T_j} \cdot T_j^s, \quad \sum_{k=1}^{l} Y_{J_i,L_k} \cdot L_k^s \right\}, \tag{1}$$

subject to

$$\sum_{j=1}^{m} X_{J_i,T_j} = 1, \qquad \forall J_i \in J, \tag{2}$$

$$\sum_{j=1}^{m} Y_{J_i,L_k} = 1, \qquad \forall J_i \in J, \tag{3}$$

$$J_i^c \leq J_i^t + J_i^s \leq J_i^d, \qquad \forall J_i \in J, \tag{4}$$

$$W_s \leq J_i^t + J_i^s \leq W_e, \qquad \forall J_i \in J, \tag{5}$$

$$J_i^t + J_i^s + \delta_{J_i,J_{i'}} - J_{i'}^t \leq \tag{6}$$

$$(2 - X_{J_i,T_j} - X_{J_{i'},T_j}) \cdot M, \quad \forall J_i \neq J_{i'} \in J, J_i^t < J_{i'}^t, \forall T_j \in T, \tag{7}$$

$$J_i^t + J_i^s + \delta_{J_i,J_{i'}} - J_{i'}^t \leq \tag{8}$$

$$(2 - Y_{J_i,L_k} - Y_{J_{i'},L_k}) \cdot M, \quad \forall J_i \neq J_{i'} \in J, J_i^t < J_{i'}^t, \forall L_k \in L, \tag{9}$$

$$\tau(T_j, L_k) + \tau(J_i, L_k) - J_i^t \leq \tag{10}$$

$$(2 - X_{J_i,T_j} - Y_{J_i,L_k}) \cdot M, \qquad \forall J_i \in J, \forall T_j \in T, \forall L_k \in L, \tag{11}$$

$$Y_{J_i,L_k} \cdot \mathcal{S}^{job}(J_i, S_s) \leq \mathcal{S}^{truck}(L_k, S_s), \forall J_i \in J, \forall L_k \in L, \forall S_s \in S, \tag{12}$$

$$X_{J_i,T_j} - C_{J_i,T_j} \leq 0, \qquad \forall J_i \in J, \forall T_j \in T, \tag{13}$$

$$Y_{J_i,L_k} \cdot L_k^r \leq X_{J_i,\mathcal{T}(L_k)} \cdot L_k^r, \qquad \forall J_i \in J, \forall L_k \in L, \tag{14}$$

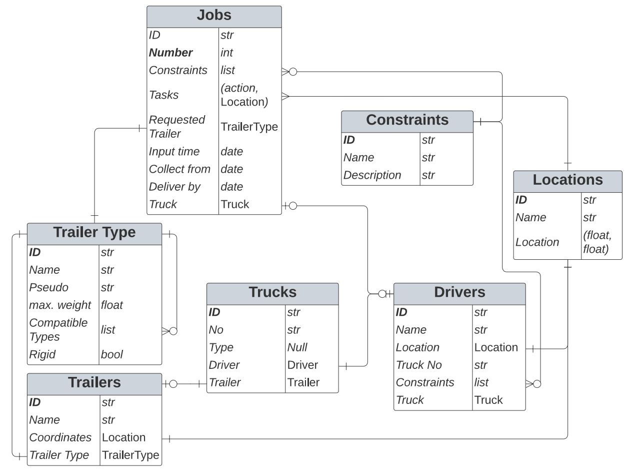

Figure 1. Entity relationship diagram showing the relationships of the data after preprocessing.

where M is a large enough constant that is calculated as follows:

$$M > \max\Big(\max_{J_i,J_{i'} \in J}(J_i^t + J_i^s + \delta_{J_i,J_{i'}}),$$
$$\max_{J_i \in J, T_j \in T, L_k \in L}(\tau(T_j, L_k) + \tau(J_i, L_k))\Big).$$

The objective function in equation (1) aims to minimise the number of subcontracted jobs that are either performed by subcontracted trucks or subcontracted trailers, subject to the following constraints.

Constraints (2) and (3) ensure that a job is performed by one and only one trailer and truck, respectively.

Constraints (4)–(10) are all related to the timing of jobs. Specifically, constraint (4) specifies that all jobs are collected and delivered on time. The fact that a job is performed during the working hours of the company is determined by constraint (5). Constraints (6) and (8) ensure that a trailer and a truck perform a job at a given time, respectively. Constraint (10) guarantees that a truck has enough time to collect a trailer and a job on time.

Finally, constraints (12)–(14) ensure that the assigned resources are compatible with the job in terms of driver-skill capability and compatibility. In particular, constraint (12), referred to as driver-skill constraint, links the skills of the driver assigned to the truck to those required to perform the job. Constraint (13) states that the assigned trailer can perform the specified job. Finally, constraint (14) requires that a job assigned to a rigid truck also be performed by the trailer associated with the rigid truck.

3.2 Data Analysis

The provided real-world data is primarily restricted to commercial confidentiality constraints, which restricts the sharing of sensitive proprietary information in academic and

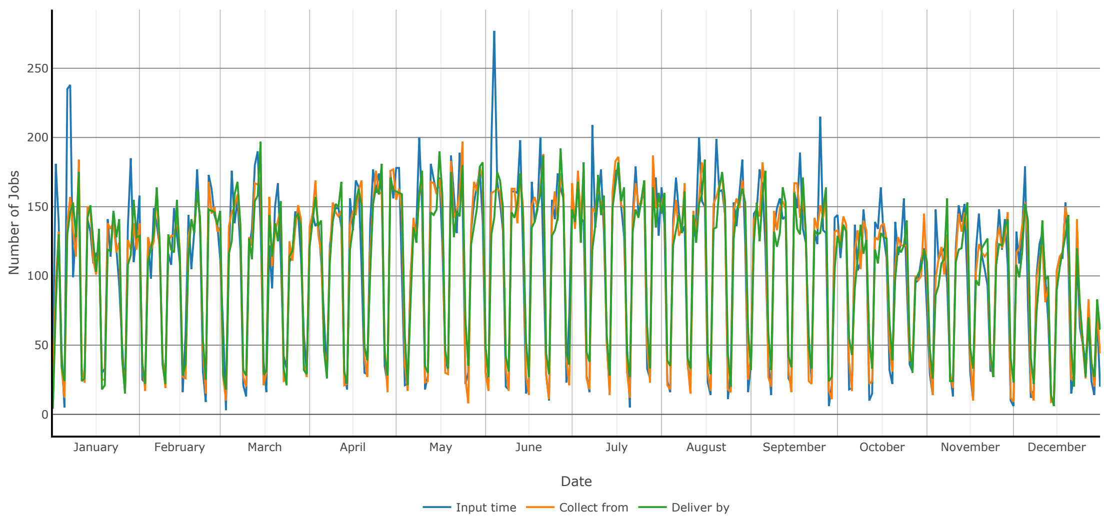

Figure 2. Annual demand of jobs in a daily basis. The colour of the lines show the input time (blue), the collection time (orange) and the delivery time (green).

Table 1. Summary of the data after preprocessing.

Dataset	Number of entries	Number of columns
Job	41, 326	9
Location	1, 055	3
Constraint	50	2
Trailer	1, 306	4
Trailer type	22	5
Truck	296	5
Drivers	296	6

industry research. Moreover, certain data elements (e.g. driver skill information) may be considered as sensitive information. In order to meet the EU General Data Protection Regulation (GDPR)[1], the sensitive information is processed to ensure data integrity and compliance with confidentiality requirements.

Figure 1 illustrates the structure and relationships that compose the original data using an entity relationship diagram. Note that the shown data structure has previously performed processing steps to prevent exposing hidden patterns that could compromise the privacy of the original dataset. Table 1 summarises the number of entries for each dataset after the preprocessing step.

In short, the data provided includes a time series of jobs (i.e. sequences of collect and delivery tasks) performed over a year, historical resource records (i.e. drivers, trucks, trailers, and trailer types), location data, and job-related constraint definitions. This study focuses on the jobs dataset for synthetic data generation, as it contains information on historical actions, locations, and execution times (e.g. "*collect cargo from A at 10 a.m.*"), which are essential for replicating the temporal and spatial dependencies of real-world scheduling scenarios in synthetic benchmarks. The dataset also includes driver-skill constraints for specific jobs and requested time windows for collection and delivery, which further extends the realism and applicability of problem instances.

The main difficulty in the optimisation process is characterised by the dynamic nature of the problem, driven by the arrival of new jobs over time. That is, the system simply detects the arrival of jobs, and if they require completion within the same day, they are treated as *dynamic jobs*, where scheduling plans need to be adjusted ad-hoc. A special class of dynamic jobs, called *flexible jobs*, are those that can be scheduled for future completion if the solver cannot complete them in the requested day. Finally, *static jobs* arrive either before the beginning of the workday, or are entered during the day for execution on a later day, allowing for planning. Note that the arrival of dynamic jobs increases the dimensionality of the solution space, while job completions reduce it, creating a fluctuating search landscape for optimisation algorithms.

In order to create accurate realistic benchmark instances and ensuring their utility, it is essential to perform a temporal analysis to check the existence of cyclic patterns in the obtained data. Figure 2 shows the annual distribution of time-dependent job variables on a daily basis. The figure reveals a weekly seasonality in the data, where job volumes are higher

[1]https://gdpr-info.eu/

on weekdays compared to weekends, except for specific periods of the year, such as Christmas period. Annual trends also reveal increased job-input fluctuation after Christmas and during a specific week in June, due to customer holidays and pre-planning.

Moreover, Figure 3 provides additional information on the daily seasonality of the data by showing the day-time distributions of input, collection, and delivery times for dynamic, flexible, and static jobs in the annual data. The daily patterns in the figure indicate that job input times are concentrated at the start of the working day, reflecting the operational nature of the logistics workflow. Collection and delivery times exhibit a binomial distribution, with peaks corresponding to morning and afternoon operational windows.

The scheduling process is further complicated by a range of static hard and soft constraints, which adds complexity to the feasibility of solutions. Constraints include customer deadlines, international regulations, such as the European Agreement concerning the International Carriage of Dangerous Goods by Road [33], and resource management limitations, such as *driver-skill* constraints, that requires specific trailers or specialised driver expertise.

A fundamental feature of this problem is its time-linkage property, where current decisions influence subsequent states. For instance, assigning a driver to a job with specific skill requirements limits their availability for other tasks, also affecting future scheduling decisions.

The heterogeneity of the considered problem can be found in two ways. On the one hand, the geographic heterogeneity is a typical challenge in real-world applications, since algorithms require calculating travel distances and regional operational demands [5]. On the other hand, resource heterogeneity further complicates the problem, as the availability of trucks, trailers, and drivers varies daily due to maintenance schedules or driver unavailability.

Finally, the absence of historical operator decisions excludes optimal-based evaluations, which are ignored to meet with the privacy requirements. Therefore, this work uses alternative metrics to evaluate algorithmic performance, such as measuring Key Performance Indicators (KPIs).

4 Synthetic Benchmark Generation

Synthetic data generation refers to the use of real-world data to generate similar (but not identical) data by capturing the hidden structure, statistical properties, and patterns in the real data, by using algorithms or mathematical models [19]. In particular, synthetic data serves as a powerful approach to cope with data scarcity, address privacy concerns, and accelerates model development across a range of industries and domains, such as computer vision [25], natural language processing [30], and healthcare [12].

The synthetic data generation process involves several stages, including data gathering, preprocessing, modelling, and evaluation. That is, a model is first constructed by fitting the processed original data, and then, the model is used to generate new samples to approximate the probability distributions and patterns in the original data (i.e. marginal and joint probability distributions), although more complex patterns may require more sophisticated modelling techniques and a deeper understanding of the underlying data characteristics.

This work presents a Gaussian copula, a simple and efficient statistical model for capturing complex correlations and data structure, to generate realistic benchmark instances. The use of Gaussian Copulas is motivated by their simplicity and interpretability over complex models (e.g. GANs), offering a straightforward parametric framework with robust performance in capturing distributions, correlations, and dependencies in the original data. Its applicability has been demonstrated in different fields, such as finance, medicine, and engineering time series analysis [17, 20, 36].

Copulas are based on Sklar's theorem [40], which states that any multivariate distribution can be written in terms of univariate marginal distributions and a copula (i.e. probability distribution function). Specifically, each random variable X_i is defined with a particular cumulative distribution F_{X_i} and a density function f_{X_i}. Hence, the copula defines the dependence structure between random variables, independent of their marginal distributions.

The synthetic data generation of Gaussian Copulas is based on the model learning and sampling. That is, first, the model learns the structure of the data by mapping the marginal distributions of each variable to standard normal distributions. Next, it computes the correlation matrix that captures the dependencies between these transformed variables. Finally, it generates synthetic data by sampling from a multivariate normal distribution defined by this correlation matrix, preserving the statistical dependencies present in the original data.

A highly useful feature of synthetic data generation models is the conditional sampling, which allows generating specific instances while maintaining the learnt distributions and patterns. Gaussian Copulas perform the conditional sampling by fixing values of certain variables and sampling the remaining variables from the derived conditional distribution [17]. In particular, the conditional sampling is used to generate typical workday instances for the problem described in Section 3.

This study uses a Gaussian Copula ensemble that combines samples from two submodels trained on distinct subsets of the jobs dataset. In short, this model captures the dynamic dependencies in the original data by (i) dividing the jobs dataset into dynamic and static subsets, (ii) training separate sub-models on each subset, and (iii) combining the sub-model outputs to generate synthetic data, since there is no dependency between dynamic and static jobs. Note that

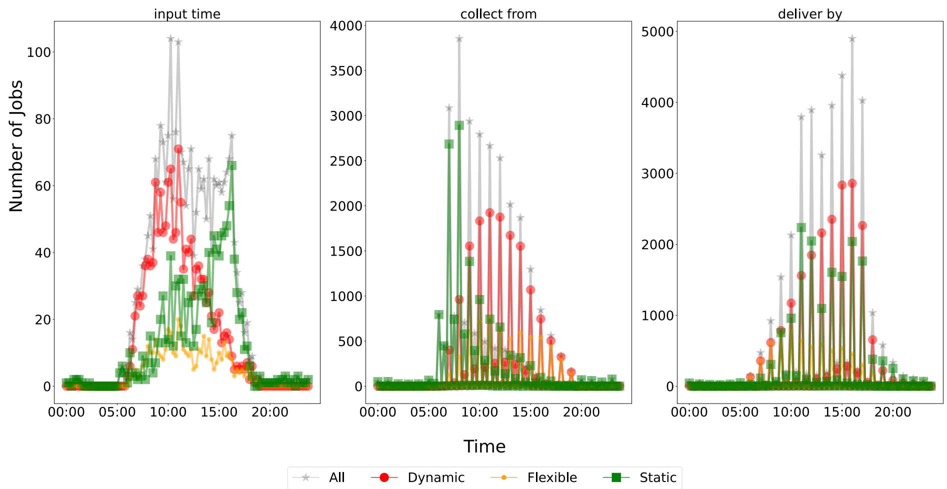

Figure 3. Half-hourly time distributions of static (green), flexible (orange), and dynamic (red) jobs for the input, collection, and delivery day-time seasonality analysis. The cumulative amount of job volume is also shown (grey).

each sub-model can generate both dynamic and static job types.

In order to capture the temporal dependencies of the original data, the model engineers (i.e. transforms) the time-related variables by calculating the time difference between input and delivery times to the collection time of jobs. This method has been selected after a thorough evaluation of its structure, parameters, and fitting process, as detailed in [1]. In particular, the cited work refers to the model as the ensemble Gaussian Copula with the different temporal approaches on the data with the weekly decomposition.

4.1 Evaluation Metrics

This study employs a set of evaluation metrics to validate the similarity, utility, and privacy of the generated synthetic data, ensuring its suitability for dynamic optimisation benchmarking. Table 2 summarises the metrics used to quantitatively measure the similarity, validity, and privacy of the generated synthetic data.

4.1.1 Fidelity Analysis.

Researchers typically use different metrics to evaluate the statistical fidelity of synthetic data, as the underlying distributions of original data are typically unknown [3]. Data fidelity analysis measures the statistical similarity (i.e. patterns and dependencies) between the original and synthetic data [11]. Specifically, data fidelity can be categorised into three types of analyses: (i) marginal distribution similarity comparison between original and synthetic data, (ii) bivariate distribution (correlation) similarity analysis for all pairs of columns in the original and synthetic data, and (iii) temporal dependency analysis of time-related variables.

Table 2. Summary of the evaluation and validation metrics used to analyse the fidelity, utility, and privacy preservation of the synthetic.

Evaluation Type	Evaluation Metric	
Fidelity analysis:	**Continuous variables**	**Discrete variables**
- Marginal distr. comparison ($\overline{MDC}$)	Kolmogorov-Smirnov test	Total variation distance
- Correlation comparison ($\overline{CC}$)	Correlation similarity	Contingency similarity
- Temporal distr. comparison ($\overline{\mathcal{T}}$)	Equations (19) and (20)	
Utility analysis:		
- Discriminative analysis ($\overline{\text{AUC-ROC}}$)	Discriminative logistic regression model performance	
- Simulation-based applicability	Integration of instances in the simulator [38]	
Privacy analysis:		
- Exact matches	New rows	

Marginal distribution comparison. The marginal distribution comparison ($\overline{MDC}$) evaluates the similarity between original and synthetic data marginals. Specifically, for each marginal, a $MDC \in [0, 1]$ score reflects the distributional difference, where a score of 1 indicates identical distributions and 0 indicates maximal dissimilarity. The overall $\overline{MDC}$ score is the average marginal similarity. The distribution comparison is quantified using the Kolmogorov-Smirnov test [41] for continuous variables (i.e. numerical and date), whereas the total variation distance [47] is used for discrete variables (i.e. categorical and boolean).

Formally, let $F_{R_X}(x)$ and $F_{S_X}(x)$ be the cumulative probability functions for the continuous variable X in the original and synthetic data, R_X and S_X, respectively. Then, the Kolmogorov-Smirnov score for the column X is computed as:

$$KS(X) = \left(1 - \max_{x \in R_X \cup S_X} \left\{ \left| F_{R_X}(x) - F_{S_X}(x) \right| \right\} \right), \quad (15)$$

where $x \in R_X \cup S_X$ represents all possible values in the combined range of samples R_X and S_X. $KS \in [0,1]$ calculates the score based on the largest difference between $F_{R_X}(x)$ and $F_{S_X}(x)$. Note that the higher the score, the higher the quality of the generated data.

Similarly, given $F_{R_X}(x)$ and $F_{S_X}(x)$, the total variation distance score for the column X is calculated as follows:

$$TV(X) = \left(1 - \frac{1}{2} \sum_{x \in R_X \cup S_X} \left|F_{R_X}(x) - F_{S_X}(x)\right|\right), \tag{16}$$

where the fraction $1/2$ ensures the normalisation $TV \in [0,1]$. Note that the score may be biased by a low number of samples, since the frequency of missing values in the synthetic data would be 0. A score $TV(X) = 1$ indicates a perfect match between marginal X in the original and synthetic data.

Correlation comparison. The correlation comparison ($\overline{CC}$) evaluates the average similarity of the correlation between column pairs in original and synthetic data. Specifically, the correlation similarity measures the correlation between continuous variables, while contingency tables analyse correlations between discrete variables. Note that, for continuous and discrete variable relationship analysis, contingency tables are used after discretising the continuous variable.

Given R_X and R_Y to be a pair of continuous columns in the original data, and S_X and S_Y to be the same columns in the synthetic data, the bivariate correlation score for the columns X and Y can be calculated as:

$$\begin{aligned}\rho(X,Y) =& \rho(Y,X) = \\ & \left(1 - \frac{1}{2}\left|\mathrm{Corr}(R_X,R_Y) - \mathrm{Corr}(S_X,S_Y)\right|\right),\end{aligned} \tag{17}$$

where $\mathrm{Corr}(R_X,R_Y), \mathrm{Corr}(S_X,S_Y) \in [0,1]$ represent the correlation function (i.e. Pearson's and Spearman's rank correlation) that returns the correlation coefficient between the given random variables. This study uses the Pearson's correlation as the correlation function. A score of $\rho(X,Y) = 1$ means a total pairwise match between correlation of the same columns in the original and synthetic data.

The absolute difference between two normalised contingency tables can be calculated using the total variation distance. Formally, let X and Y be two discrete variables representing two columns of the original and synthetic data, $\mathbf{R}$ and $\mathbf{S}$, and let $C^R_{X,Y}$ and $C^S_{X,Y}$ denote their normalised contingency tables, respectively. Then, the contingency similarity score for X and Y can be measured as:

$$\begin{aligned}&\mathcal{S}^c(X,Y) = \\ &\left(1 - \frac{1}{2}\sum_{x \in X}\sum_{y \in Y}\left|C^R_{X,Y}(x,y) - C^S_{X,Y}(x,y)\right|\right),\end{aligned} \tag{18}$$

where $C^R_{X,Y}(x,y), C^S_{X,Y}(x,y) \in [0,1]$ represent the probability of $x \in X$ and $y \in Y$ in the original and synthetic data, respectively. Note that the higher the score, the more similar the two distributions are.

Note that the overall $\overline{CC} \in [0,1]$ score represents the mean of absolute correlation scores, where a score of 1 indicates perfect correlation.

Temporal distribution comparison. As described in Section 3, capturing the complex temporal dependencies (i.e. chronology) of time-related variables (i.e. the input, collection, and delivery times) of jobs in the synthetic data is essential. That is, the input time of samples must be no later than the requested collection time, and the collection time must be strictly before the delivery time.

This work suggest quantifying the similarity of the time-related variables by using the Kolmogorov-Smirnov test (Equation (15)). Thus, the overall time-distribution similarity score is calculated as follows:

$$\overline{\mathcal{T}} = \frac{1}{|\boldsymbol{T}|}\sum_{T \in \boldsymbol{T}} KS(T), \tag{19}$$

where $T \in \boldsymbol{T}$ is a time-related variable, and $KS(T)$ is the Kolmogorov-Smirnov score for the variable T. A higher score $\overline{\mathcal{T}} \in [0,1]$ indicates a higher distributinal similarity between the time-related variables in the original and synthetic data.

Note that $\overline{\mathcal{T}}$ can be applied to different seasonal components, such as annual ($\overline{\mathcal{T}}_{annual}$) and weekly ($\overline{\mathcal{T}}_{weekly}$) distribution comparison. Nevertheless, for the daily distribution similarity it may be more appropriate to also incorporate the dynamism in the formulation. Hence, the daily distribution comparison ($\overline{\mathcal{T}}_{daily}$) is represented as follows:

$$\overline{\mathcal{T}}_{daily} = \frac{1}{|\boldsymbol{T}|}\sum_{T \in \boldsymbol{T}} \frac{1}{2}\left(KS(T_{dyn}) + KS(T_{stat})\right), \tag{20}$$

where T_{dyn} and T_{stat} represent the values of the time-related variable T for dynamic and static jobs, respectively.

4.1.2 Utility Analysis. This evaluation measures the usability of the synthetic data for benchmarking, ensuring it adequately substitutes the real-world data without compromising analytical integrity. This analysis can be divided into two different sections: (i) performing a discriminative analysis using machine learning, where a classifier is trained to distinguish between real-world and synthetic data, and (ii) evaluating of the applicability of benchmark in the simulator, which involves directly using the generated synthetic data as input for the DTTSP simulation.

Discriminative analysis. This process involves evaluating the ability of a classification algorithm (e.g. logistic regression) to distinguish between real and synthetic data based. The overall utility score can be given by the average Area Under the Receiver Operating Characteristic (AUC-ROC) score across all the cross-validation splits, which can be formulated as follows:

$$\overline{\text{AUC-ROC}} = 1 - (2 \times \max(\text{AUC-ROC}, 0.5) - 1)\,. \tag{21}$$

This formulation ensures that the overall score is always between 0 and 1, even if the raw AUC-ROC is below 0.5 (i.e. random classifier). A higher score indicates better replication of the statistical properties in the original data, making synthetic data more suitabile for benchmarking.

it should be noted that a high discriminative analysis score may also imply that the synthetic data reproduces exact or near-exact records from the real-world data, thus, compromising commercial confidentiality. This complementary privacy aspect is covered in Section 4.1.3.

Simulation-based analysis. In order to analyse the validity of the generated benchmark instances, they are applied into the automated real-time vehicle tracking and decision-making support simulator, developed in [38]. This integration allows for a practical evaluation of the performance of the simulator with synthetic data inputs, gaining insights into the ability of the synthetic data to represent realistic scenarios and support effective decision-making.

In short, the simulator assumes that all trucks are based in the depot at the start of the working day, while trailers can be distributed heterogeneously across the considered geographical area. The simulator employs a constructive algorithm to to minimise vehicle idle time, while maximising fleet productivity, thereby reducing the reliance on subcontracted trucks, trailers, and drivers. Note that the algorithm is said to be an offline method, as it iteratively plans the scheduling of jobs after the arrival of each new job.

4.1.3 Privacy Analysis.

The aim of privacy analysis is to protect confidential and sensitive information in the synthetic data generation process. Different privacy metrics are employed to evaluate the privacy characteristics of the synthetic data. Nevertheless, the suggested synthetic data generation framework is ethically compliant and safe, as it focuses on company-related operational aspects, ensuring the output does not replicate sensitive data, such as personal identifiable driver information.

This study uses one simple measure to measure privacy, the exact match score, which identifies and quantifies identical records in both the original and synthetic data. Note that lower scores indicate a better privacy, meaning that little or no real-world information is revealed in the synthetic data.

5 Experimental Study

The experimentation of this work is divided in two parts: (i) the fidelity, utility, and privacy analyses for the synthetic data are performed to capture the distribution, correlations, and dependencies in the whole real-world data, and (ii) the conditional sampling is analysed to evaluate the distributions and dependencies of the generated synthetic instances. Complete implementation and results are available online[2].

[2]https://zenodo.org/records/15308781

Table 3. Fidelity, utility, and privacy analysis of the ensemble Gaussian Copula on the annual data.

Evaluation Metric	Score (%)
Fidelity analysis:	
- $\overline{MDC}$	98.98%
- $\overline{CC}$	85.75%
- $\overline{\mathcal{T}}_{annual}$	99.28%
- $\overline{\mathcal{T}}_{weekly}$	96.87%
- $\overline{\mathcal{T}}_{daily}$	92.59%
Utility analysis:	
- $\overline{\text{AUC-ROC}}$	100%
Privacy analysis:	
- Exact matches	99.05%

5.1 Evaluation of the Gaussian Copula

A summary of the fidelity, utility, and privacy scores of the ensemble Gaussian Copula is shown in Table 3. The table shows the fidelity, utility, and privacy scores (as percentages) of the model. Specifically, the utility analysis is determined by the overall marginal distribution ($\overline{MDC}$) and correlation ($\overline{CC}$) similarity scores, as well as the annual, weekly, and daily similarity scores (i.e. $\overline{\mathcal{T}}_{annual}$, $\overline{\mathcal{T}}_{weekly}$, and $\overline{\mathcal{T}}_{daily}$, respectively).

The results obtained show a good performance of the used Gaussian Copula to capture the distributions, patterns, and dependencies in the original data. In order to better understand the daily seasonality similarity, the time distributions are displayed in Figure 4. The figure shows the day-time distribution of the input, collection, and delivery times of the synthetic data.

The distributions of the Gaussian Copula-based results reveals a rough approximation of distributions from the original data (Figure 3), in particular for the collection time, where the distributions of static and dynamic jobs are closely matched with the original distributions. Nevertheless, the input time- and delivery time distributions show different shapes to the distributions shown in Figure 3. This may be caused by the relative values (i.e. time-difference to collection time of jobs) used for the time-related variables, which perturb the mathematical properties of these variables.

Moreover, the outstanding utility and privacy scores reflect that the synthetic data reproduces the statistical properties of the original data without replicating original records. These results validate the developed Gaussian Copula model to produce high-fidelity and privacy-preserving benchmark instances.

5.2 Conditional Sampling

In order to illustrate the validity of the model to produce benchmark instances, the performance of the model using conditional sampling needs to be evaluated. Table 5 presents the results of using the developed Gaussian Copula model to

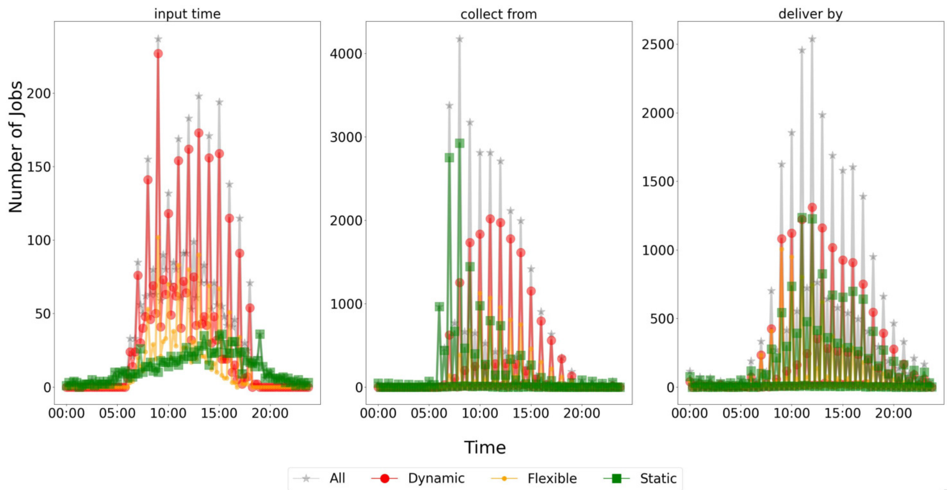

Figure 4. Half-hourly time distributions of the used ensemble Gaussian Copula model. The plots show static (green), flexible (orange), and dynamic (red) jobs for the input, collection, and delivery day-time seasonality analysis, respectively. The cumulative amount of job volume is also shown (grey).

Table 4. Data characteristics for the considered typical day instances, extracted from the real-world data.

	21 January	1 March	17 April	19 April
Weekday	Monday	Friday	Tuesday	Thursday
Input jobs	141	112	169	106
Collect jobs	138	135	144	169
Deliver jobs	119	147	154	151
Static jobs	75	71	67	93
Dynamic jobs	63	64	77	76
Flexible jobs	27	9	8	28

synthesise job income data for four typical workdays. The typical days are selected based on random seeds (i.e. $1-4$).

Obtained results show a significant decline in the performance for the ensemble Gaussian Copula with conditional sampling. In particular, the model effectively captures the daily patterns in the data, but underperforms in capturing distributions, correlations, and dependencies in the real-world typical days. This underperformance is primarily caused by the rejection process of the conditional sampling, which iteratively discards infeasible solutions requiring several iterations to satisfy temporal dependencies and conditional requirements (i.e. specific collection time).

It is worth noting that the model is fitted using annual data, and conditional sampling is employed to sample specific dates (and therefore, problem instances) from the model. However, the distributions for specific dates may differ from those observed annually, which can affect the similarity scores. Nevertheless, despite this variability, the model fitted with annual data is sufficiently general to approximate annual distributions and produce realistic synthetic instances using conditional sampling.

Table 5. Utility analysis of the conditional sampling of the ensemble Gaussian Copula.

Date	$\overline{MDC}$	$\overline{CC}$	$\overline{\mathcal{T}}_{annual}$	$\overline{\mathcal{T}}_{weekly}$	$\overline{\mathcal{T}}_{daily}$
21 January	65.67%	43.93%	15.69%	37.38%	85.75%
1 March	63.76%	42.17%	5.33%	39.35%	90.13%
17 April	68.72%	44.12%	29.39%	38.41%	88.86%
19 April	67.63%	43.47%	28.82%	62.31%	89.63%

Finally, the integration of synthetic benchmark instances into the simulator has proven effective for reproducing realistic job income workflow for a typical day. This performance validates the utility of the generated synthetic instances for simulation-based decision-making for logistics.

In summary, conditional sampling shows promise for creating specific synthetic instances based on a model derived from annual data. However, it should be used cautiously as a benchmark generator, since the distributions of the generated instances may differ from those in the original data.

6 Discussion

The obtained results demonstrate the potential of using a Gaussian Copula-based synthetic data generation model to effectively replicate the multivariate dependencies and statistical properties of the DTTSP. Nevertheless, the average analysis presented in Section 5 presents several limitations that require further discussion. In the following paragraphs, a list of limitations is discussed, highlighting areas where the framework could be improved and where further research on the modelling would be needed.

First, the flexibility of the current methodology could be extended to a broader parameter selection. That is, although the proposed Gaussian Copula model generates benchmark

Table 6. Impact of the dataset size on the ensemble Gaussian Copula model performance.

Training size	$\overline{MDC}$	$\overline{CC}$	$\overline{\mathcal{T}}_{annual}$	$\overline{\mathcal{T}}_{weekly}$	$\overline{\mathcal{T}}_{daily}$
80%	98.87%	87.64%	99.27%	96.95%	92.51%
50%	98.61%	86.67%	98.80%	96.85%	92.32%
30%	98.41%	85.74%	98.63%	96.87%	92.46%
10%	97.23%	84.29%	97.57%	96.76%	90.67%

instances that simulate the job income for specific working days, other data elements (see Figure 1) are assumed to be fixed (e.g. types and quantities of trucks, trailers). Moreover, incorporating a wider range of dynamic events (e.g. resource availability and failures or job cancellations) may present a more realistic framework. This presents an opportunity to evaluate the impact of different dynamic characteristics of the problem on the algorithmic performance.

Second, the performed utility analysis has demonstrated a reduction of the marginal and correlation scores to meet the temporal dependencies in the original data. This limitation of the model and its failure to capture original time distributions indicate the need to explore alternative and more suitable synthetic data generation techniques, such as using a sequential synthetic data generation method for the generation of synthetic time series data [3].

Finally, data scarcity is considered one of the main challenges for producing synthetic data in many domains [12, 28, 30]. This applies to situations where data collection may be expensive or time-consuming, or computational resources may be limited. Note that smaller datasets can cause overfitting, whereas very large datasets can be computationally expensive. Therefore, identifying the minimal amount of data required to achieve satisfactory performance without excessive computational costs is crucial. Certainly, it is generally believed that the performance and robustness of the model typically improve with large datasets without noise, outliers, or irrelevant information, since they allow for a better pattern, correlation, and distribution learning.

This study introduces a data-scarcity analysis to evaluate the validity of the model by exploring the amount of data needed for generating valid and useful DTTSP instances. Table 6 illustrates the *learning curve* of the model by displaying the fidelity scores achieved during training and evaluation with varying amounts of real-world data, i.e. 10%, 20%, 50%, and 80% of random collection days picked from the original data, as similarly done in [43].

The scarcity analysis of the obtained results reveals similar marginal distribution and correlation metrics to the whole data, as shown in Table 3. However, the model shows a lower capacity to capture temporal patterns as the amount of data decreases, specifically in terms of annual and daily seasonalities. Nevertheless, note that the model maintains reasonably high fidelity scores even with a 10% reduction of the available data, suggesting that data augmentation may not result in fidelity improvements.

Hence, this examination suggests understanding the balance between data quantity and quality, since increasing training data may not improve model performance. Future research could explore techniques to deal with data scarcity to improve the performance of synthetic data generation models in data-limited applications.

7 Conclusion and Future Work

This study introduces a framework for creating realistic data-driven benchmark instances for the DTTSP. This enables the development and systematic comparative evaluation of dynamic optimisation algorithms capable of tackling realistic problem features identified in the logistics domain. In particular, the work provides a novel framework for the use of synthetic data generation to overcome data scarcity and privacy concerns that are often a barrier to exposure of the academic community to real-world data. The study also documents limitations of the developed synthetic data generation approach in fully capturing the complex dependencies and dynamic patterns of real-world data.

There is considerable scope for future research on the development and applicability of synthetic data generation models for DOP benchmarking. Possible extensions of this particular study include implementing a more suitable model to capture the temporal patterns in the data, performing an instance space analysis to better understand benchmark characteristics [42], using the provided data to develop further benchmarks for related DOPs (e.g. vehicle allocation), and integrating additional constraints, such as driving-hour regulations or fuel usage of trucks.

Acknowledgments

This publication is based upon work from COST Action CA22137 "Randomised Optimisation Algorithms Research Network" (ROAR-NET), supported by COST (European Cooperation in Science and Technology).

Josu Ceberio has been partially supported by the Spanish Ministry of Science (MICIU/AEI/10.13039/PID2023-149195-NB-I00), and the Research Groups 2022-2025 (IT1504-22) and Elkartek (KK-2024/00030) programs from the Basque Government.

The authors acknowledge ARR Craib Ltd for allowing them to use their data as part of the contributions to this publication.

References

[1] Joan Alza. 2025. *Adaptive Challenge for Algebraic and Realistic Dynamic Optimisation Benchmarks.* Ph. D. Dissertation. Robert Gordon University.

[2] Joan Alza, Mark Bartlett, Josu Ceberio, and John McCall. 2023. On the Elusivity of Dynamic Optimisation Problems.

[3] Fouad Bahrpeyma, Mark Roantree, Paolo Cappellari, Michael Scriney, and Andrew McCarren. 2021. A Methodology for Validating Diversity in Synthetic Time Series Generation.
[4] Peter A. N. Bosman. 2005. Learning, Anticipation and Time-deception in Evolutionary Online Dynamic Optimization.
[5] Raouf Boucekkine, Giorgio Fabbri, Salvatore Federico, and Fausto Gozzi. 2022. Managing Spatial Linkages and Geographic Heterogeneity in Dynamic Models with Transboundary Pollution.
[6] Jürgen Branke. 1999. Memory Enhanced Evolutionary Algorithms for Changing Optimization Problems.
[7] Jürgen Branke. 2002. *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publishers, Boston, MA.
[8] Eoin Brophy, Zhengwei Wang, Qi She, and Tomás Ward. 2023. Generative Adversarial Networks in Time Series: A Systematic Literature Review.
[9] Hui Cheng, Shengxiang Yang, and Xingwei Wang and. 2012. Immigrants-enhanced Multi-Population Genetic Algorithms for Dynamic Shortest Path Routing Problems in Mobile Ad Hoc Networks.
[10] Carlos Cruz, Juan R. González, and David A. Pelta. 2011. Optimization in Dynamic Environments: A Survey on Problems, Methods and Measures.
[11] DataCebo, Inc. 2024. *Synthetic Data Metrics*. DataCebo, Inc. Version 0.13.0.
[12] Alvaro Figueira and Bruno Vaz. 2022. Survey on Synthetic Data Generation, Evaluation Methods and GANs.
[13] H. Fu, P. R. Lewis, B. Sendhoff, K. Tang, and X. Yao. 2014. What are Dynamic Optimization Problems?
[14] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*.
[15] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks.
[16] Shuang Hao, Wenfeng Han, Tao Jiang, Yiping Li, Haonan Wu, Chunlin Zhong, Zhangjun Zhou, and He Tang. 2024. Synthetic Data in AI: Challenges, Applications, and Ethical Implications.
[17] José Miguel Hernández-Lobato, James Robert Lloyd, and Daniel Hernández-Lobato. 2013. Gaussian Process Conditional Copulas with Applications to Financial Time Series.
[18] Yaochu Jin and Jürgen Branke. 2005. Evolutionary Optimization in Uncertain Environments – A Survey.
[19] James Jordon, Lukasz Szpruch, Florimond Houssiau, Mirko Bottarelli, Giovanni Cherubin, Carsten Maple, Samuel N. Cohen, and Adrian Weller. 2022. Synthetic Data – What, Why and How?
[20] Sanket Kamthe, Samuel Assefa, and Marc Deisenroth. 2021. Copula Flows for Synthetic Data Generation.
[21] Changhe Li, Trung Thanh Nguyen, Sanyou Zeng, Ming Yang, and Min Wu. 2018. An Open Framework for Constructing Continuous Optimization Problems.
[22] Changhe Li and Shengxiang Yang. 2008. A Generalized Approach to Construct Benchmark Problems for Dynamic Optimization.
[23] Hongjian Li and Geng Zhang. 2022. Designing Benchmark Generator for Dynamic Optimization Algorithm.
[24] CKY Lin. 2011. A Vehicle Routing Problem with Pickup and Delivery Time Windows, and Coordination of Transportable Resources.
[25] Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, and Junbin Gao. 2023. Diffusion Models for Time Series Applications: A Survey.
[26] Pao-Cheng Lin. 2009. Multi-Population Genetic Algorithms with Immigrants Scheme for Dynamic Shortest Path Routing Problems in Mobile Ad Hoc Networks.
[27] Min Liu, Hemant Kumar Singh, and Tapabrata Ray. 2014. A Benchmark Generator for Dynamic Capacitated Arc Routing Problems.
[28] Yingzhou Lu, Minjie Shen, Huazheng Wang, Xiao Wang, Capucine van Rechem, and Wenqi Wei. 2023. Machine Learning for Synthetic Data Generation: A Review.
[29] Michalis Mavrovouniotis, Shengxiang Yang, and Xin Yao. 2012. A Benchmark Generator for Dynamic Permutation-Encoded Problems.
[30] Scott McLachlan. 2017. *Realism in Synthetic Data Generation*. Master's thesis. Massey University of New Zeland.
[31] Yi Mei, Xiaodong Li, and Xin Yao. 2009. A Hybrid Ant Colony Optimization Algorithm for the Extended Capacitated Arc Routing Problem.
[32] Roberto Montemanni, Luca Maria Gambardella, Andrea Emilio Rizzoli, and Alberto V. Donati. 2005. A Hybrid Ant Colony Optimisation Technique for Dynamic Vehicle Routing.
[33] United Nations. 2016. European Agreement Concerning the International Carriage of Dangerous Goods by Road (ADR).
[34] Trung Thanh Nguyen. 2011. *Continuous Dynamic Optimisation using Evolutionary Algorithms*. Ph. D. Dissertation. University of Birmingham.
[35] Trung Thanh Nguyen, Shengxiang Yang, and Jürgen Branke. 2012. Evolutionary Dynamic Optimization: A Survey of the State of the Art.
[36] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. 2016. The Synthetic Data Vault.
[37] Isis Torres Pérez, José Luis Verdegay, Carlos Cruz Corona, and Alejandro Rosete Suárez. 2013. Models and Solutions for Truck and Trailer Routing Problems: An Overview.
[38] Olivier Regnier-Coudert, John McCall, Mayowa Ayodele, and Steven Anderson. 2016. Truck and Trailer Scheduling in a Real World, Dynamic and Heterogeneous Context.
[39] Philipp Rohlfshagen, Per Kristian Lehre, and Xin Yao. 2009. Dynamic Evolutionary Optimisation: An Analysis of Frequency and Magnitude of Change.
[40] Martin Sklar. 1959. Fonctions de Répartition an Dimensions et Leurs Marges.
[41] Nikolai V Smirnov. 1939. Estimate of Deviation between Empirical Distribution Functions in Two Independent Samples.
[42] Kate Smith-Miles and Mario Andrés Muñoz. 2023. Instance Space Analysis for Algorithm Testing: Methodology and Software Tools.
[43] Giovanni Tataranno, Federico Bigi, and Francesco Viti. 2025. Dynamic OD Matrix Estimation using Data-Driven Modelling under Data-Scarcity: an application of Sparse Variational Gaussian Process.
[44] Renato Tinós and Shengxiang Yang. 2014. Analysis of Fitness Landscape Modifications in Evolutionary Dynamic Optimization.
[45] Hao Tong, Miqing Li, Jialin Liu, and Xin Yao. 2025. How Do Dynamic Events Change the Fitness Landscape of Traveling Salesman Problems?
[46] Thomas Weise, Alexander Podlich, Kai Reinhard, Christian Gorldt, and Kurt Geihs. 2009. Evolutionary freight transportation planning.
[47] M. B. Wilk and R. Gnanadesikan. 1968. Probability Plotting Methods for the Analysis of Data.
[48] Xiaoshu Xiang, Jianfeng Qiu, Jianhua Xiao, and Xingyi Zhang. 2020. Demand Coverage Diversity based Ant Colony Optimization for Dynamic Vehicle Routing Problems.
[49] Shengxiang Yang, Yong Jiang, and Trung Thanh Nguyen. 2013. Metaheuristics for Dynamic Combinatorial Optimization Problems.
[50] Shengxiang Yang and Xin Yao. 2003. Dual Population-based Incremental Learning for Problem Optimization in Dynamic Environments.
[51] Danial Yazdani, Jürgen Branke, Mohammad Sadegh Khorshidi, Mohammad Nabi Omidvar, Xiaodong Li, Amir H. Gandomi, and Xin Yao. 2024. Clustering in Dynamic Environments: A Framework for Benchmark Dataset Generation With Heterogeneous Changes.
[52] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. 2021. A Survey of Evolutionary Continuous Dynamic Optimization Over Two Decades—Part B.
[53] Danial Yazdani, Mohammad Nabi Omidvar, Ran Cheng, Jürgen Branke, Trung Thanh Nguyen, and Xin Yao. 2022. Benchmarking Continuous Dynamic Optimization: Survey and Generalized Test Suite.
[54] Abdunnaser Younes, Paul Calamai, and Otman Basir. 2005. Generalized Benchmark Generation for Dynamic Combinatorial Problems.