GUS

Indeks 371262
e-ISSN 2657-9545
ISSN 0033-2372

Przegląd Statystyczny
Statistical Review

Vol. 68 | No. 3 | 2021

GŁÓWNY URZĄD STATYSTYCZNY
STATISTICS POLAND

INFORMATION FOR AUTHORS

Przegląd Statystyczny. Statistical Review publishes original research papers on theoretical and empirical topics in statistics, econometrics, mathematical economics, operational research, decision science and data analysis. The manuscripts considered for publication should significantly contribute to the theoretical aspects of the aforementioned fields or shed new light on the practical applications of these aspects. Manuscripts reporting important results of research projects are particularly welcome. Review papers, shorter papers reporting on major conferences in the field, and reviews of seminal monographs are eligible for submission, but only on the Editor's request.

Since 1 May 2019, the journal has been publishing articles in English.

Any spelling style is acceptable as long as it is consistent within the manuscript.

All work should be submitted to the journal through the ICI Publishers Panel
(https://editors.publisherspanel.com/pl.ici.ppanel-app-war/ppanel/index).

For details of the submission process and editorial requirements please visit
https://ps.stat.gov.pl/ForAuthors.

Przegląd Statystyczny
Statistical Review

Vol. 68 No. 3 2021

EDITORIAL OFFICE'S ADDRESS

University of Gdańsk, ul. Armii Krajowej 101, 81-824 Sopot, Poland

Language editing: Scientific Journals Division, Statistics Poland
Technical editing and typesetting: Statistical Publishing Establishment – team supervised by Wojciech Szuchta

Zakład Wydawnictw Statystycznych

Printed and bound: Statistical Publishing Establishment
al. Niepodległości 208, 00-925 Warsaw, Poland, zws.stat.gov.pl

Website: ps.stat.gov.pl

ISSN 0033-2372
e-ISSN 2657-9545
Index 371262

Information on the sales of the journal: Statistical Publishing Establishment
phone no. +48 22 608 32 10, +48 22 608 38 10

Order no. 340/2021 – 230 printed copies

CONTENTS

Przegląd Statystyczny. Statistical Review, 2021, vol. 68, 3, 1–15 DOI: 10.5604/01.3001.0015.5582
Received: 26.07.2021, revised: 14.09.2021, accepted: 18.10.2021

Forecasting currency covariances using machine learning tree-based algorithms with low and high prices[1]

Sylwester Bejger,[a] Piotr Fiszeder[b]

Abstract. We combine machine learning tree-based algorithms with the usage of low and high prices and suggest a new approach to forecasting currency covariances. We apply three algorithms: Random Forest Regression, Gradient Boosting Regression Trees and Extreme Gradient Boosting with a tree learner. We conduct an empirical evaluation of this procedure on the three most heavily traded currency pairs in the Forex market: EUR/USD, USD/JPY and GBP/USD. The forecasts of covariances formulated on the three applied algorithms are predominantly more accurate than the Dynamic Conditional Correlation model based on closing prices. The results of the analyses indicate that the GBRT algorithm is the best-performing method.

Keywords: machine learning, tree-based ensembles, volatility models, high-low range, covariance forecasting

JEL: C22, C45, C53, C58, C63, G17

1. Introduction

Multivariate volatility models can be used in many financial applications, such as asset pricing, portfolio optimisation, risk management, the estimation of systemic risk in banking, Value-at-Risk estimation or asset allocation. Volatility models of financial instruments that are commonly used are largely based on closing prices only. However, the use of daily low and high prices leads to more accurate estimates and forecasts of variances (e.g. Chou, 2005; Fiszeder & Perczak, 2016; Lin et al., 2012; Molnár, 2016) and covariances (e.g. Chou et al., 2009; Fiszeder, 2018; Fiszeder et al., 2019; Fiszeder & Fałdziński, 2019). Daily low and high prices are almost always available alongside closing prices in financial series. Therefore, making use of them in volatility models is important from a practical viewpoint. The review of multivariate range-based models can be found in Petropoulos et al. (in press).

Recently, the importance of machine learning (ML) algorithms in the forecasting of financial time series has increased considerably (see e.g. de Prado, 2018). ML models, unlike classic (nonlinear) time series analysis, do not require prior

[1] This research was financed by the following National Science Centre projects: 2016/21/B/HS4/00662 entitled 'Multivariate volatility models – the application of low and high prices' and 2019/35/B/HS4/00642 entitled 'Forecasting financial markets using a machine learning approach'.

[a] Nicolaus Copernicus University in Toruń, Faculty of Economic Sciences and Management, Department of Applied Informatics and Mathematics in Economics, ul. Gagarina 13a, 87-100 Toruń, Poland, e-mail: sylwester.bejger@umk.pl, ORCID: https://orcid.org/0000-0001-7900-946X.

[b] Nicolaus Copernicus University in Toruń, Faculty of Economic Sciences and Management, Department of Econometrics and Statistics, ul. Gagarina 13a, 87-100 Toruń, Poland, e-mail: piotr.fiszeder@umk.pl, ORCID: https://orcid.org/0000-0002-9777-2239.

assumptions about the underlying structure of data (Zhang, 2003) and are able to capture recurring nonlinear patterns in time series (see e.g. Fischer et al., 2019). These factors cause machine learning algorithms to outperform most traditional stochastic methods in financial market forecasting (Fiszeder & Orzeszko, 2021; Ryll & Seidens, 2019). The most popular ML approaches in the field of finance are Artificial Neural Network (ANN) and Support Vector Machine (SVM). On average, recurrent neural networks outperform feed-forward neural networks as well as support vector machines (Ryll & Seidens, 2019). However, most models based on ANN and SVM are treated as 'black box' algorithms. A black box model is a system that does not reveal its internal mechanisms, and therefore in machine learning it describes models that cannot be understood by looking at their parameters (e.g. an artificial neural network). Interpretable or Explainable Machine Learning refers to methods and models that make the behaviour and predictions of machine learning systems understandable to humans, which is essential in business forecasting and decision-making (Bejger & Elster, 2020). As an alternative to ANN and SVM models, we want to evaluate the performance of tree-based ensemble algorithms in the forecasting of financial time series. Although ensemble learning algorithms are also referred to as black boxes, if a part of an ensemble is a decision tree, the interpretability of the model and predictions becomes much greater. The decision mechanism (model) of a single decision tree is entirely transparent and interpretable (Barredo Arrieta et al., 2019) due to its intrinsic properties. Among others things, it enables the ranking of the relative significance of predictor variables through variable importance metrics (VIMs) (Biau & Scornet, 2016; Breiman, 2001). In an ensemble of trees, these measures could be used for the global and local interpretability of predictions.

The aim of the paper is to suggest a new approach to forecasting currency covariances based on the combination of machine learning tree-based algorithms with the use of low and high prices. The methods we selected are based on the regression tree concept and Classification and Regression Trees (CART) split criterion (Breiman et al., 1984). We apply the Random Forest Regression (RFR) algorithm (Breiman, 2001), the Gradient Boosting Regression Trees (GBRT) algorithm (Friedman, 2001), and the Extreme Gradient Boosting with tree learner (XGBoost, described in Chen & Guestrin, 2016). Although the most popular ML approaches in the field of finance are ANN and SVM learning methods (Henrique et al., 2019; Ryll & Seidens, 2019), we decided to apply the three above-mentioned algorithms instead, for the following reasons:

- models based on ANN and SVM are treated as 'black box' algorithms with no clear interpretation of hyperparameters and the importance of features, while tree-based methods are interpretable through VIMs (an intrinsic property of tree-based models) and a well-defined hyperparameter meaning;

- they can handle heterogeneous data (ordered or categorical variables, or a mix of both) with minimal preprocessing;
- they can handle highly correlated predictor variables;
- as all tree-based methods, they intrinsically implement feature selection;
- they either do not overfit (RFR) or are easy to control against overfitting (GBRT, XGBoost);
- tree-based methods are still rarely used in forecasting financial markets, despite the fact that they proved to be among the best in competitions such as M4, M5, or those organised by the Kaggle portal.

The applications of the RFR, GBRT, and XGBoost algorithms in the forecasting of financial markets are presented in the works of Ghosh et al. (in press), Islam et al. (2021), Khaidem et al. (2016), Krauss et al. (2017), Kumar and Thenmozhi (2006), Waldow et al. (2021), Yang (2021) and Yang et al. (2021). However, most of those studies are devoted to the forecasting of stock prices or exchange rates. To the best of our knowledge, this study presents the first application of the forecasting of currency covariances. We empirically evaluate the usability of the algorithms on the three most heavily traded currency pairs in the Forex market: EUR/USD, USD/JPY, and GBP/USD. The forecasts of covariances formulated on the three applied algorithms are predominantly more accurate than the Dynamic Conditional Correlation benchmark model based on closing prices.

The remaining part of the paper is organised in the following way: Section 2 describes applied models and methods, in Section 3 we present the data and an outline of the study, Section 4 evaluates the forecasts of the covariance of returns from the RFR, GBRT, XGBoost algorithms and the DCC model, and Section 5 contains the conclusions of the study.

2. Theoretical background

2.1. Tree-based ensemble algorithms

The building block of the machine learning algorithms which we use is a weak learner of a regression tree (e.g. Breiman et al., 1984; Quinlan, 1992). A regression tree is a supervised learning method used to learn a function that combines a set of variables intending to predict another variable. The general idea of a tree learner is to partition feature space X into a set of rectangles and then fit a simple model (like a constant c) in each one (Hastie et al., 2009).

The prediction function of a tree is defined as:

$$f(x) = \sum_{m=1}^{M} c_m I(x, R_m), \tag{1}$$

where M is the number of leaves in the tree; R_m is a region in the feature space (corresponding to leaf m), c_m is a constant corresponding to region m, I is the indicator function (returning 1 if $x \in R_m$, 0 otherwise). The value of c_m is determined in the training phase of the tree. For regression, we partition the predictor space to find a set of regions R that minimise the RSS, given by:

$$\sum_{m=1}^{M} \sum_{i \in R_m} \left(y_i - \hat{y}_{R_m}\right)^2, \tag{2}$$

where $\hat{y}_{R_m}$ is the mean response for the training observations within the m-th region.

As it is infeasible to consider every possible partition of the feature space into M regions, a top-down, greedy algorithm known as recursive partitioning (e.g., CART for a binary tree, Breiman et al., 1984) is used to train the single tree. The essential element of CART is a split criterion, dependent on an impurity measure. A regression tree split criterion computes the (renormalised) difference between the empirical variance in the node before and after a cut is performed.

2.1.1. Random Forest ensemble algorithm

A single regression tree is typically a weak prediction model which is unstable (high variance learner). To reduce prediction variance and prevent bias from increasing, one can combine the prediction of many weak learners (Schapire, 1990), creating an ensemble of learners. A Random Forest (Breiman, 2001; Ho, 1998) is an ensemble (or forest) of decision trees grown from a randomised variant of a tree induction algorithm.

The Random Forest exploits two sources of randomness to reduce the correlation of residuals of base learners, which decreases the general prediction error. The first of them is a bootstrap, where each tree is constructed on the basis of a bootstrap-resampled training data set, thanks to which the trees are different from each other. The second is a split-variable randomisation: each time a split is to be performed, the search for the split variable is limited to a random m subset of the p predictors, which leads to the decorrelation of trees. When the forest's element is a regression tree, such a learning mechanism is called Random Forest Regression.

The RFR prediction is the unweighted average over predictions (1) of the set of K trees:

$$F(x) = \frac{1}{K} \sum_{j}^{K} f_k(x), \tag{3}$$

If $m = p$, we have an ancestor of random forests, bootstrap aggregation, or a bagging ensemble method (Breiman, 2004).

2.1.2. Gradient Boosting Regression Trees

Boosting (Schapire, 1999) is also a technique which additively combines many weak learners to an ensemble. It is a form of a more general concept of additive expansion learning. Boosting algorithms evolved from Adaboost, the first successful boosting algorithm (Freund, 1995; Freund & Schapire, 1997), to its generalisation as a Gradient Boosting that handles various loss functions (Friedman, 2001; Friedman et al., 2000). The GBRT algorithm involves two main steps – fitting (shallow) decision tree h to the 'residuals' from the model, given current tree model F, and updating F by adding h and updating the residuals. Those steps are repeated until an error on the test set starts to arise. The natural idea is to generalise boosting for any differentiable loss function (for example, not sensitive to outliers). In our study, we use the Huber loss function of the following form:

$$L(y,F) = \begin{cases} \frac{1}{2}(y-F)^2 \; |y-F| \leq \delta, \\ \delta\left(|y-F| - \frac{\delta}{2}\right) |y-F| > \delta. \end{cases} \tag{4}$$

The most important differences between GBRT and RFR are the folowing: trees are grown sequentially, which means that each tree is grown using information from the previously grown trees; the method is more sensitive to overfitting, and the number of trees should be controlled ex-post.

2.1.3. Extreme Gradient Boosting algorithm

XGBoost is a scalable machine learning system for tree boosting. It was implemented and described by Chen and Guestrin (2016). XGBoost is widely recognised by practitioners (e.g. Kaggle competitors) and has implementations in many programming languages (R, Python, Java, Scala, Julia, Perl, and others). The method is based on the GBRT idea, but the computational implementation offers more hyperparameters to tune. There is a technical difference in optimising a loss function between GBRT and XGBoost, as GBRT divides the optimisation problem into two parts (the determination of the direction of the minimisation step, the optimisation of the step length). XGBoost tries to determine both in one step directly. It means that at each iteration, both algorithms need to calculate the gradient at the current estimate. Still, XGBoost also needs to calculate the Hessian matrix, so the XGBoost loss function must be twice differentiable.

2.2. Range-based covariance estimator for exchange rates

In the suggested approach to covariance forecasting, we apply the estimator of the covariance of returns calculated on the basis of low and high prices. This estimator has an advantage over that based only on the closing prices, because it uses information about the price changes during the day. Let us consider two exchange rates of currencies A and B in terms of currency C, denoted as A/C and B/C, respectively. In the absence of triangular arbitrage opportunities, the return of the cross rate can be written as:

$$\Delta\ln \mathrm{A}/B = \Delta\ln \mathrm{A}/C - \Delta\ln \mathrm{B}/C. \quad (5)$$

Then, the range-based estimator of covariance for the currency pairs can be represented as:

$$\mathrm{cov}(\Delta\ln \mathrm{A}/C, \Delta\ln \mathrm{B}/C) = 0.5[\mathrm{var}(\Delta\ln \mathrm{A}/C) + \mathrm{var}(\Delta\ln \mathrm{B}/C) - \mathrm{var}(\Delta\ln \mathrm{A}/B)]. \quad (6)$$

As a variance estimator, we use the Parkinson (1980) estimator given as:

$$\mathrm{var}_{Pt} = [ln(H_t/L_t)]^2/(4\ln 2), \quad (7)$$

where H_t and L_t are the daily high and low prices, respectively.

More details about the applied range-based covariance estimator and its properties can be found in Fiszeder and Orzeszko (2021), who employ this estimator in a new methodology for dynamic modeling and forecasting covariance matrices based on support vector regression.

2.3. The DCC model

In this section, we describe the DCC model of Engle (2002). It is one of the most popular multivariate volatility models (see e.g. Bauwens et al., 2012) and is often used as a benchmark model in empirical studies. Let us assume that $\boldsymbol{\varepsilon}_t$ ($N \times 1$ vector) is the innovation process for the conditional mean and can be written as:

$$\boldsymbol{\varepsilon}_t | \psi_{t-1} \sim N(0, \mathbf{cov}_t), \quad (8)$$

where ψ_{t-1} is the set of all information available at time $t-1$, N is the multivariate normal distribution, and $\mathbf{cov}_t$ is the $N \times N$ symmetric conditional covariance matrix.

The DCC(P, Q) model can be presented as:

$$\mathbf{cov}_t = \mathbf{D}_t \mathbf{cor}_t \mathbf{D}_t, \tag{9}$$

$$\mathbf{cor}_t = \mathbf{Q}_t^{*-1} \mathbf{Q}_t \mathbf{Q}_t^{*-1}, \tag{10}$$

$$\mathbf{Q}_t = \left(1 - \sum_{i=1}^{Q} \zeta_i - \sum_{j=1}^{P} \theta_j\right)\mathbf{S} + \sum_{i=1}^{Q} \zeta_i (\mathbf{z}_{t-i}\mathbf{z}'_{t-i}) + \sum_{j=1}^{P} \theta_j \mathbf{Q}_{t-j}, \tag{11}$$

where $\mathbf{D}_t = \text{diag}(h_{1t}^{1/2}, h_{2t}^{1/2} \dots, h_{Nt}^{1/2})$, conditional variances h_{kt} (for $k = 1,2, \dots, N$) are described as univariate GARCH models (equations (12–13)), $\mathbf{z}_t$ is the standardised $N \times 1$ residual vector assumed to be serially independently distributed given as $\mathbf{z}_t = \mathbf{D}_t^{-1}\boldsymbol{\varepsilon}_t$, $\mathbf{cor}_t$ is the time varying $N \times N$ conditional correlation matrix of $\mathbf{z}_t$, $\mathbf{S}$ is the unconditional $N \times N$ covariance matrix of $\mathbf{z}_t$ (it can also be estimated with other parameters of the model, but this makes estimation more difficult) and $\mathbf{Q}_t^*$ is the diagonal $N \times N$ matrix composed of the square root of diagonal elements of $\mathbf{Q}_t$. Parameters ζ_i (for $i = 1,2, \dots, Q$ and θ_j (for $j = 1,2, \dots, P$) are nonnegative and satisfy the $\sum_{i=1}^{Q} \zeta_i + \sum_{j=1}^{P} \theta_j < 1$ condition.

The univariate GARCH(p, q) model applied in the DCC model can be written as:

$$\varepsilon_{kt} | \psi_{t-1} \sim N(0, h_{kt}), \quad k = 1,2, \dots, N, \tag{12}$$

$$h_{kt} = \alpha_{k0} + \sum_{i=1}^{q} \alpha_{ki} \varepsilon_{k\,t-i}^2 + \sum_{j=1}^{p} \beta_{kj} h_{k\,t-j}, \tag{13}$$

where $\alpha_{k0} > 0, \alpha_{ki} \geq 0, \beta_{kj} \geq 0$ (for $k = 1,2, \dots, N$; $i = 1,2, \dots, q$; $j = 1,2, \dots, p$), weaker conditions for nonnegativity of the conditional variance can be assumed (see Nelson & Cao, 1992). The requirement for covariance stationarity of ε_{kt} is $\sum_{i=1}^{q} \alpha_{ki} + \sum_{j=1}^{p} \beta_{kj} < 1$.

Parameters of the DCC model can be estimated by the quasi-maximum likelihood method using a two-stage approach. Let the parameters of model $\boldsymbol{\Theta}$ be written in two groups, i.e. $\boldsymbol{\Theta}' = (\boldsymbol{\Theta}'_1, \boldsymbol{\Theta}'_2)$, where $\boldsymbol{\Theta}_1$ is the vector of the parameters of conditional means and variances, and $\boldsymbol{\Theta}_2$ is the vector of the parameters of the correlation part of the model. The log-likelihood function can be written as the sum of two parts:

$$L(\boldsymbol{\Theta}) = L_{Vol}(\boldsymbol{\Theta}_1) + L_{Corr}(\boldsymbol{\Theta}_2 | \boldsymbol{\Theta}_1), \tag{14}$$

where $L_{Vol}(\boldsymbol{\Theta}_1)$ represents the volatility part:

$$L_{Vol}(\boldsymbol{\Theta}_1) = -\frac{1}{2} \sum_{t=1}^{n} (N \ln(2\pi) + ln|\mathbf{D}_t|^2 + \boldsymbol{\varepsilon}'_t \mathbf{D}_t^{-2} \boldsymbol{\varepsilon}_t), \tag{15}$$

while $L_{Corr}(\mathbf{\Theta}_2|\mathbf{\Theta}_1)$ can be viewed as the correlation component:

$$L_{Corr}(\mathbf{\Theta}_2|\mathbf{\Theta}_1) = -\frac{1}{2}\sum_{t=1}^{n}(\ln|\mathbf{cor}_t| + \mathbf{z}'_t\mathbf{cor}_t^{-1}\mathbf{z}_t - \mathbf{z}'_t\mathbf{z}_t). \tag{16}$$

$L_{Vol}(\mathbf{\Theta}_1)$ can be written as the sum of the log-likelihood functions of N univariate GARCH models:

$$L_{Vol}(\mathbf{\Theta}_1) = -\frac{1}{2}\sum_{k=1}^{N}\left(n\ln(2\pi) + \sum_{t=1}^{n}\left(ln(h_{kt}) + \frac{\epsilon_{kt}^2}{h_{kt}}\right)\right). \tag{17}$$

In the first stage, the parameters of univariate GARCH models can be estimated separately for each of the assets and the estimates of h_{kt} can be obtained. In the second stage, residuals transformed by their estimated standard deviations are used to estimate the parameters of the correlation part ($\mathbf{\Theta}_2$) conditioning on the parameters estimated in the first stage ($\widehat{\mathbf{\Theta}}_1$).

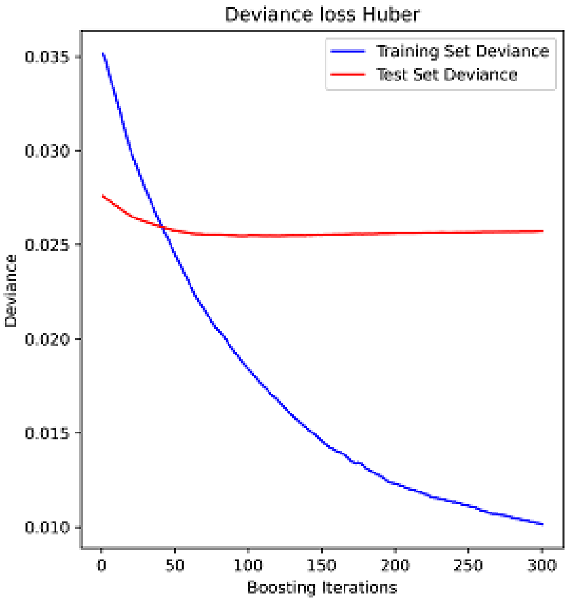
3. Data and description of the research

We evaluate the accuracy of the proposed procedure of covariance forecasting based on data from the Forex market, for the purpose of which we examine three most heavily traded currency pairs, namely EUR/USD, USD/JPY and GBP/USD. Daily data for the period from 2 January 2004 to 30 December 2016 are used. A total sample of 3,365 observations is split into a training set (period: 14 Janury 2004 to 31 December 2014, size: 2,846 observations, the first eight observations are truncated during the construction of the analytical dataset) and a test set (period: 2 January 2015 to 30 December 2016, holdout set size: 519 observations).

The target variable is the covariance of returns of currency pairs given in (6). This estimator is more efficient than the one based on closing prices only. The set of predictors contains the following time series: – target_lag_1 until target_lag_8, min_A/C_lag1, min_B/C_lag1, max_A/C_lag1, max_B/C_lag1, close_A/C_lag1, close_B/C_lag1, lnzwr_A/C_lag1 and lnzwr_B/C_lag1. We also add categorical predictors commonly used in training time series models: month, weekofyear, dayofweek, dayofyear, dayofmonth. Categorical variables are integer-coded (which is a better option for the tree-based methods than one-hot encoding).

We implement machine learning pipelines for the random forest, GBRT and XGBoost in the Python 3.6.3 environment, and use standard libraries for data processing and machine learning, i.e. numpy, scikit-learn, pandas and datetime. Additionally, we use the XGBoost library and the scikit-learn wrapper interface for XGBoost. The training set, containing about 85% of the samples, is used to tune hyperparameters using the time series k-fold cross-validator (*TimeSeriesSplit* method). Standard cross-validation techniques assume that samples are independent and identically distributed, and would result in an unreasonable correlation between training and testing instances (yielding poor estimates of the generalisation error) on time series data. In the *TimeSeriesSplit* method, successive training sets are supersets of those that come before them. It also adds all surplus data to the first training partition, which is always used to train the model. The preliminary tuning of hyperparameters is done by searching the space of the parameters (*RandomizedSearchCV* or *GridSearchCV* methods) with the above described k-fold cross-validation. In a random forest, the additional calibration of the *min_samples_split*, *max_depth*, *min_impurity_decrease* and *min_samples_leaf* split hyperparameters is performed. In the case of boosting (GBRT), the ex-post control for overfitting (the value of the *n_estimators* hyperparameter) is done (see the Figure).

Figure. Train and test set deviance against boosting iterations



Source: authors' calculations.

For tree-based supervised learning, the critical model's elements are impurity measures, which determine the split quality and a loss function, influencing the quality of predictions. In regression on time series data, the MSE and RMSE impurity measures are applied. As a loss function, we utilise the Huber loss function in the GBRT model, and the square loss in XGBoost.

The three ensemble models are tuned, trained, and applied on the out-of-sample instances to generate forecasts. Parameters of the DCC model are estimated each day on a rolling sample of a fixed size of 500 observations (approximately 2 years).

4. Comparison of covariance forecasts

This section compares out-of-sample one-day-ahead forecasts of the covariance of returns from three machine learning algorithms (RFR, GBRT and XGBoost), with the forecasts from the DCC model. We evaluate forecasts for a two-year period from 2 January 2015 to 30 December 2016.

The sum of products of 15-minute returns (the realised covariance) is employed as a proxy of the daily covariance for the evaluation of the forecasts. We assess the forecasts from the models based on two primary measures, i.e. the mean squared error (MSE) and the mean absolute error (MAE). In order to evaluate the statistical significance of the results, the Diebold-Mariano test (Diebold & Mariano, 1995) is applied. We perform a pairwise comparison with respect to the DCC benchmark model. The forecasting performance results are presented in the Table.

Table. Evaluation of covariance forecasts for selected exchange rates

Method	Forecast evaluation criteria			
	MSE	DM test *p*-value	MAE	DM test *p*-value
EUR/USD-JPY/USD				
DCC	0.2016	–	0.1741	–
RFR	0.1935	0.0420	**0.1455**	0.0000
GBRT	**0.1889**	0.0004	0.1477	0.0000
XGBoost	0.1925	0.0445	0.1464	0.0000
EUR/USD–GBP/USD				
DCC	0.3122	–	0.1662	–
RFR	0.2905	0.1416	0.1870	0.0130
GBRT	**0.2799**	0.0258	**0.1536**	0.1060
XGBoost	0.2903	0.0729	0.1833	0.0199
JPY/USD–GBP/USD				
DCC	0.7503	–	0.1717	–
RFR	**0.6408**	0.0132	0.1407	0.0085
GBRT	0.6476	0.0141	**0.1373**	0.0053
XGBoost	0.6349	0.0158	0.1486	0.0545

Note: the evaluation period is from 2 Janury 2015 to 30 December 2016, the realised covariances are used as the real values of covariance and estimated as the sum of products of 15-min. returns. The lowest values of the MSE and MAE are marked in bold. The *p*-values of the Diebold-Mariano test are presented for pairs of models: the selected algorithm and the DCC benchmark. A *p*-value lower than the significance level means that the forecasts of covariance from the selected method with a lower evaluation measure are significantly more accurate.

Source: authors' calculations.

Under the MSE criterion, the forecasts of covariance from all the three machine learning algorithms are more accurate than the forecasts based on the DCC model. According to the Diebold-Mariano test, the advantage of these algorithms is statistically significant at the level of 10%, except the EUR/USD-GBP/USD relation for RFR. Under the MAE measure, the forecasts based on the analysed machine learning methods are again significantly more accurate than the forecasts from the DCC model for the EUR/USD-JPY/USD and JPY/USD-GBP/USD relations. For EUR/USD-GBP/USD, the lowest value of the criterion occurs for the GBRT algorithm, but this result is not statistically significant. Predominantly, both of the loss functions indicate the GBRT algorithm as the best performing method.

5. Conclusions

The machine learning ensemble method is a method that combines a set of weak learners to create a (more potent) learner that performs better than any of the individual ones. Ensemble methods help reduce bias and/or variance. We use a decision tree as a base, weak learner. We examine the performance of three popular tree-based ensemble algorithms: random forest (regression), GBRT and XGBoost. These algorithms exploit two different approaches to ensemble learning. Random forest trains individual estimators independently over bootstrapped subsets of data (bagging) and incorporates the second level of randomness. When optimising each node split, only a random subsample (without replacement) of the attributes will be evaluated, with the purpose of the further decorrelating of the estimators. Both GBRT and XGBoost utilise a boosting technique that is different from a random forest. In boosting, individual trees are fitted sequentially, observations are weighted differently in each iteration, and poor-performing trees are excluded. All the three algorithms belong to a group of machine learning algorithms which are most popular and widely-used in many fields. It is worth noticing that a variant of the gradient boosting algorithm, LightGBM, has won the M5 forecasting competition (Makridakis et al., 2020). Tree-based ensembles are also becoming increasingly popular in financial forecasting (Henrique et al., 2019; de Prado, 2018).

Daily low and high prices contain important information about the variability of the prices of financial instruments, but they are very seldom used for the estimation of volatility models. We combine machine learning tree-based algorithms with the usage of low and high prices and suggest a new approach to forecasting currency covariances. We conduct an empirical evaluation of this procedure on the basis of three most heavily traded currency pairs in the Forex market: EUR/USD, USD/JPY and GBP/USD. The forecasts of covariances formulated on the three applied algorithms are in most part more accurate than the DCC model, used as a benchmark model based on closing prices. The results of the analyses indicate that the GBRT algorithm is the best-performing method.

Research on tree-based machine learning methods in covariance forecasting can be further developed, for example in the area of the analysis of the importance of predictors or studies on the interpretability of the optimal values of hyper-parameters. Other issues, such as modifying a loss function in boosting-based methods and examining the performance of random forest and gradient boosting variants (e.g. quantile regression forests, dynamic random forests, and the LightGBM algorithm) seem worth further investigation as well.

References

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2019, December 26). *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI*. https://arxiv.org/pdf/1910.10045.pdf.

Bauwens, L., Hafner, C., & Laurent, S. (Eds.). (2012). *Handbook of Volatility Models and Their Applications*. John Wiley & Sons. https://doi.org/10.1002/9781118272039.

Bejger, S., & Elster, S. (2020). Artificial Intelligence in economic decision making: how to assure a trust?. *Ekonomia i Prawo / Economics and Law*, *19*(3), 411–434. https://doi.org/10.12775/EiP.2020.028.

Biau, G., & Scornet, E. (2016). A random forest guided tour. *TEST*, *25*(2), 197–227. https://doi.org/10.1007/s11749-016-0481-7.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324.

Breiman, L. (2004). Bagging predictors. *Machine Learning*, *24*, 123–140. https://doi.org/10.1007/BF00058655.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees* (1st edition). Chapman & Hall/CRC. https://doi.org/10.1201/9781315139470.

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco.

Chou, R. Y. (2005). Forecasting Financial Volatilities with Extreme Values: The Conditional Autoregressive Range (CARR) Model. *Journal of Money, Credit and Banking*, *37*(3), 561–582. https://doi.org/10.1353/mcb.2005.0027.

Chou, R. Y., Wu, C. C., & Liu, N. (2009). Forecasting time-varying covariance with a range-based dynamic conditional correlation model. *Review of Quantitative Finance and Accounting*, *33*(4), 327–345. https://doi.org/10.1007/s11156-009-0113-3.

Diebold, F. X., & Mariano, R. S. (1995). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263. https://doi.org/10.2307/1392185.

Engle, R. F. (2002). Dynamic Conditional Correlation: A Simple Class of Multivariate Generalized Autoregressive Conditional Heteroskedasticity Models. *Journal of Business and Economic Statistics*, *20*(3), 339–350. https://doi.org/10.1198/073500102288618487.

Fischer, T. G., Krauss, C., & Deinert, A. (2019). Statistical Arbitrage in Cryptocurrency Markets. *Journal of Risk and Financial Management*, *12*(1), 1–15. https://doi.org/10.3390/jrfm12010031.

Fiszeder, P. (2018). Low and high prices can improve covariance forecasts: The evidence based on currency rates. *Journal of Forecasting*, *37*(6), 641–649. https://doi.org/10.1002/for.2525.

Fiszeder, P., & Fałdziński, M. (2019). Improving forecasts with the co-range dynamic conditional correlation model. *Journal of Economic Dynamics and Control*, *108*, 1–16. https://doi.org/10.1016/j.jedc.2019.103736.

Fiszeder, P., Fałdziński, M., & Molnár, P. (2019). Range-based DCC models for covariance and value-at-risk forecasting. *Journal of Empirical Finance*, *54*, 58–76. https://doi.org/10.1016/j.jempfin.2019.08.004.

Fiszeder, P., & Orzeszko, W. (2021). Covariance matrix forecasting using support vector regression. *Applied Intelligence*, *51*(10), 7029–7042. https://doi.org/10.1007/s10489-021-02217-5.

Fiszeder, P., & Perczak, G. (2016). Low and high prices can improve volatility forecasts during periods of turmoil. *International Journal of Forecasting*, *32*(2), 398–410. https://doi.org/10.1016/j.ijforecast.2015.07.003.

Freund, Y. (1995). Boosting a Weak Learning Algorithm by Majority. *Information and Computation*, *121*(2), 256–285. https://doi.org/10.1006/inco.1995.1136.

Freund, Y., & Schapire, R. E. (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139. https://doi.org/10.1006/jcss.1997.1504.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232. https://doi.org/10.1214/aos/1013203451.

Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, *28*(2), 337–407. https://doi.org/10.1214/aos/1016218223.

Ghosh, P., Neufeld, A., & Sahoo, J. K. (in press). Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. *Finance Research Letters*. https://doi.org/10.1016/j.frl.2021.102280.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inteference, and Prediction* (2nd edition). Springer. https://doi.org/10.1007/978-0-387-84858-7.

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, *124*, 226–251. https://doi.org/10.1016/j.eswa.2019.01.012.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(8), 832–844. https://doi.org/10.1109/34.709601.

Islam, S. F. N., Sholahuddin, A., & Abdullah, A. S. (2021). Extreme gradient boosting (XGBoost) method in making forecasting application and analysis of USD exchange rates against rupiah. *Journal of Physics: Conference Series*, *1722*, 1–11. https://doi.org/10.1088/1742-6596/1722/1/012016.

Khaidem, L., Saha, S., & Dey, S. R. (2016, April 29). *Predicting the direction of stock market prices using random forest*. https://arxiv.org/pdf/1605.00003.pdf.

Krauss, Ch., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, *259*(2), 689–702. https://doi.org/10.1016/j.ejor.2016.10.031.

Kumar, M., & Thenmozhi, M. (2006). *Forecasting stock index movement: A comparison of support vector machines and random forest*. 9th Capital Markets Conference, Vashi.

Lin, E. M. H., Chen, C. W. S., & Gerlach, R. (2012). Forecasting volatility with asymmetric smooth transition dynamic range models. *International Journal of Forecasting*, *28*(2), 384–399. https://doi.org/10.1016/j.ijforecast.2011.09.002.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020, October 6). *The M5 Accuracy competition: Results, findings and conclusions*. https://www.researchgate.net/publication/344487258_The_M5_Accuracy_competition_Results_findings_and_conclusions.

Molnár, P. (2016). High-low range in GARCH models of stock return volatility. *Applied Economics*, *48*(51), 4977–4991. https://doi.org/10.1080/00036846.2016.1170929.

Nelson, D. B., & Cao, C. Q. (1992). Inequality Constraints in the Univariate GARCH Model. *Journal of Business & Economic Statistics*, *10*(2), 229–235. https://doi.org/10.2307/1391681.

Parkinson, M. (1980). The Extreme Value Method for Estimating the Variance of the Rate of Return. *The Journal of Business, 53*(1), 61–65. https://doi.org/10.1086/296071.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Oliveira, F. L. C., Baets, S. D., Dokumentov, A., Ellison, J., Fiszeder, P., Franses, P. H., Frazier, D. T., Gilliland, M., Gönül, M. S., Goodwin, P., Grossi, L., Grushka-Cockayne, Y., Guidolin, M., Guidolin, M., Gunter, U., Guo, X., Guseo, R., Harvey, N., Hendry, D. F., Hollyman, R., Januschowski, T., Jeon, J., Jose, V. R. R., Kang, Y., Koehler, A. B., Kolassa, S., Kourentzes, N., Leva, S., Li, F., Litsiou, K., Makridakis, S., Martin, G. M., Martinez, A. B., Meeran, S., Modis, T., Nikolopoulos, K., Önkal, D., Paccagnini, A., Panagiotelis, A., Panapakidis, I., Pavía, J. M., Pedio, M., Pedregal, D. J., Pinson, P., Ramos, P., Rapach, D. E., Reade, J. J., Rostami-Tabar, B., Rubaszek, M., Sermpinis, G., Shang, H. L., Spiliotis, E., Syntetos, A. A., Talagala, P. D., Talagala, T. S., Tashman, L., Thomakos, D., Thorarinsdottir, T., Todini, E., Arenas, J. R. T., Wang, X., Winkler, R. L., Yusupova, A., & Ziel, F. (in press). Forecasting: Theory and Practice. *International Journal of Forecasting*.

de Prado, M. L. (2018). *Advances in Financial Machine Learning*. John Wiley & Sons.

Quinlan, J. R. (1992). *C4.5 Programs for Machine Learning*. Morgan Kaufmann.

Ryll, L., & Seidens, S. (2019, July 6). *Evaluating the Performance of Machine Learning Algorithms in Financial Market Forecasting: A Comprehensive Survey*. https://arxiv.org/pdf/1906.07786.pdf.

Schapire, R. E. (1990). The Strength of Weak Learnability. *Machine Learning*, *5*(2), 197–227. https://doi.org/10.1007/BF00116037.

Schapire, R. E. (1999). *A Brief Introduction to Boosting*. Sixteenth International Joint Conference on Artificial Intelligence, Stockholm.

Waldow, F., Schnaubelt, M., Krauss, C., & Fischer, T. G. (2021). Machine Learning in Futures Markets. *Journal of Risk and Financial Management*, *14*(3), 1–144. https://doi.org/10.3390/jrfm14030119.

Yang, Y. (2021). *Market Forecast using XGboost and Hyperparameters Optimized by TPE*. 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID), Guangzhou.

Yang, Y., Wu, Y., Wang, P., & Jiali, X. (2021). *Stock Price Prediction Based on XGBoost and LightGBM*. 2021 International Conference on Economic Innovation and Low-carbon Development, Qingdao.

Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, 59–175. https://doi.org/10.1016/S0925-2312(01)00702-0.

Przegląd Statystyczny. Statistical Review, 2021, vol. 68, 3, 16–40 DOI: 10.5604/01.3001.0015.5584
Received: 30.09.2021, revised: 22.11.2021, accepted: 24.11.2021

A genetic algorithm for vehicle routing in logistic networks with practical constraints

Grzegorz Koloch,[a] Michał Lewandowski,[b] Marcin Zientara,[c] Grzegorz Grodecki,[d] Piotr Matuszak,[e] Igor Kantorski,[f] Adam Nowacki[g]

Abstract. We optimise a postal delivery problem with time and capacity constraints imposed on vehicles and nodes of the logistic network. Time constraints relate to the duration of routes, whereas capacity constraints concern technical characteristics of vehicles and postal operation outlets. We consider a method which can be applied to a brownfield scenario, in which capacities of outlets can be relaxed and prospective hubs identified. As a solution, we apply a genetic algorithm and test its properties both in small case studies and in a simulated problem instance of a larger (i.e. comparable with real-world instances) size. We show that the genetic operators we employ are capable of switching between solutions based on direct origin-to-destination routes and solutions based on transfer connections, depending on what is more beneficial in a given problem instance. Moreover, the algorithm correctly identifies cases in which volumes should be shipped directly, and those in which it is optimal to use transfer connections within a single problem instance, if an instance in question requires such a selection for optimality. The algorithm is thus suitable for determining hubs and satellite locations. All considerations presented in this paper are motivated by real-life problem instances experienced by the Polish Post, the largest postal service provider in Poland, in its daily plans of delivering postal packages, letters and pallets.

Keywords: rich vehicle routing problem, brownfield, hubs and satellites, genetic algorithm

JEL: C60, C61, L87

[a] SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: gkoloch@sgh.waw.pl, ORCID: https://orcid.org/0000-0001-5506-2864.

[b] SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: mlewan1@sgh.waw.pl, ORCID: https://orcid.org/0000-0002-6003-1859.

[c] SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: mzient@sgh.waw.pl, ORCID: https://orcid.org/0000-0002-3984-2665.

[d] SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: ggrode@sgh.waw.pl, ORCID: https://orcid.org/0000-0001-6558-3217.

[e] SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: pmatusz@sgh.waw.pl, ORCID: https://orcid.org/0000-0002-5312-9698.

[f] SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: ikanto@sgh.waw.pl, ORCID: https://orcid.org/0000-0001-9870-0803.

[g] SGH Warsaw School of Economics, Institute of Econometrics, Decision Analysis and Support Unit, ul. Madalińskiego 6/8, 02-513 Warszawa, e-mail: anowack@sgh.waw.pl, ORCID: https://orcid.org/0000-0003-4635-6347.

1. Introduction

We consider a realistic variant of a postal delivery problem with time and capacity constraints, which is a special case of a general vehicle routing problem (VRP) (see Toth & Vigo, 2002). Duration of routes is constrained since volumes need to be shipped on time (Spliet & Gabor, 2015). The capacity of vehicles is constrained, as Borčinová (2017) or Ralphs et al. (2003) notice, so only a limited volume of postal cargo can be shipped within each vehicle. Keskin et al. (2019) also assert that the capacity of postal operations outlets – nodes of the network – is constrained both in terms of vehicles that can be served by an outlet, and in terms of volumes that can be processed there.

Within any logistic plan, streams of volumes (demands) can be delivered directly or they can be shipped using some outlets as transfer locations, in which volumes are processed, merged and shipped to their final destinations. We are interested in a technology capable of identifying such outlets within the network, so that they can be recommended as prospective hubs when the network expands. It should be remembered that within a brownfield type of analysis, such prospective hubs do not have to overlap with outlets in which currently most of the volume is processed.

We use a genetic algorithm and argue that with a natural representation of solutions and with relatively simple specifications of genetic operators, subject to the representation in question, we are capable of obtaining such technology as that tested in small case studies and in a real-life problem instance. In particular, the employed genetic operators are capable of switching between direct and transfer connections, depending on what is optimal in a given problem instance, and also of selecting which volumes should be shipped directly and which should be shipped using transfer connections within a single problem instance – if the instance in question requires such a selection for optimality. All considerations presented in this paper are motivated by real-life problem instances which we analysed for the Polish Post (Poczta Polska), the largest, state-owned provider of postal services in Poland.

2. Problem Statement

The problem discussed in this paper can be considered as a version of a VRP which contains elements of various VRP specifications, e.g. VRPs with split pickups and split deliveries (see Casazza et al., 2018; Wassan & Nagy, 2014), VRPs with time windows (see Benjamin & Beasley, 2013; Bräysy & Gendreau, 2005), VRPs with capacity constraints (see Baldacci et al., 2012) and VRPs with multiple depot locations (see Crevier et al., 2007; Nagy & Salhi, 2005).

This problem differs from the typical ones, most importantly due to technical capacity constraints imposed on nodes of the network, and because of the fact that transported volumes can be split or merged in the nodes of the network in an arbitrary way. Additionally, we assume that an overall constraint is imposed on the length of the route (expressed here by the number of locations that can serve as transfer nodes within each route), whereas it is typical for VRPs that the length of the routes, i.e. sequences of locations visited by vehicles, is not constrained. More specifically, we assume that the route between the origin and the destination locations of a postal shipment can take the form of either a direct connection or a transfer connection, with one facility at most in which the vehicle stops along the route. Using direct connections can be beneficial, since they tend to avoid heavy traffic, and as a consequence smaller costs are incurred by vehicles and drivers. Moreover, if a substantial volume is processed on such a connection, i.e. a large vehicle (or vehicles) can be fully loaded, the number of utilised vehicles decreases, and so does the unit cost of the shipment.

On the other hand, transfer connections make it possible to split the stream into a given number of sub-streams and distributing them over the network, possibly merging with streams originating from different locations. It is also beneficial, since it reduces the number of vehicles needed to serve the deliveries and may help preventing vehicles from getting caught in heavy traffic at some locations. However, when a volume is shipped using a transfer location, it needs to be unloaded and processed by the staff and machinery of a transfer outlet, then re-loaded into either the same or another vehicle, and finally delivered to its destination location. Processing streams within postal operations outlets requires time and incurs costs.

The incorporation of these technical features of operation outlets into the definition of the analysed problem is crucial when we are interested in a brownfield analysis and the identification of prospective hubs. The solver must balance the above-mentioned elements of the problem.

The version of the postal delivery problem which we consider in this article is related to a number of other problems and approaches in the field of logistics and optimal planning, both in the context of the identification of transfer connections and in the context of the divisibility of the delivery. As regards the former, a useful, although a slightly outdated summary of formulations and solution heuristics used in point-to-point delivery systems (including postal systems) is proposed by Leung at al. (1990). Also de Camargo et al. (2013), Çetiner et al. (2010) and Karimi and Setak (2018) investigate how transfer locations are determined on the basis of demands and the topology of the postal delivery network. In the context of the divisibility of the product stream, Kerivin et al. (2008) present an up-to-date formulation of a splittable pickup and delivery problem with reloads, Archetti and Speranza (2012) also consider a framework with split deliveries, whereas Baumung et al. (2015), for example, focus on designing optimal plans for a postal delivery network with parcels and letter mail.

The set of orders (streams, demands) that need to be processed will be denoted by S, and a single generic stream by $s \in S$. The execution of a stream consists in delivering a given positive volume of divisible cargo from the origin location to the destination location, using a logistic network with its operating characteristics like distances and travel times between nodes, capacity of vehicles, capacities of the nodes, time windows in which the nodes operate, costs of the utilisation of vehicles, and costs of volume processing within the nodes. The objective is to deliver all streams on time and at the smallest possible cost. In the further parts of the paper we will provide a more formal description of the analysed problem.

2.1. Logistic network

A logistic network is specified by an ordered pair (N, E), where $N = \{1,2,\dots,n\}$ denotes an n-element set of nodes (locations of postal operations outlets), and E denotes a set of directed edges. In the case of the company whose problem is analysed here, there is a direct connection between each pair of nodes $(i,j) \in E$, as long as $i \neq j$, hence $E = N^2 \backslash \{(i,i), i \in N\}$. The term direct connection between a pair of locations, say $i \in N$ and $j \in N$, does not necessarily mean that the physical route between location i and location j does not go through any $p \in N$ location (or locations). It means, however, that when delivering the stream of volume from i to j, we do not stop in postal operations outlets located in such nodes.

2.2. Postal operations outlets

Each postal operations outlet has the following technical characteristics:

a) a time window in which the outlet operates: $tw_i = [open_i, close_i]$, where $open_i, close_i$ are timestamps (points in time in a day), defined with the precision of up to a minute;

b) the capacity of the outlet, in terms of the number of vehicles it can process, $capVeh_i > 0$;

c) the capacity of the outlet, in terms of the volume it can process, $capVol_i > 0$;

d) the unit cost of processing volume $cVol > 0$;

e) time needed to process a unit of volume $tVol > 0$.

In practice, the values of $open_i$ and $close_i$ are similar for the respective $i \in N$ outlets, but as they do not have to be identical, they are indexed by i. They represent, respectively, the earliest and the latest time at which outlet i is operational throughout the day, i.e. the earliest when any volume can enter it (the unloading of volumes from vehicles can start), and the latest any volume has to leave it (the loading of all volumes into vehicles has to finish). When a vehicle enters a postal operations outlet, what happens first is unloading the volume from it. The volume contained in a vehicle is the sum of the volumes of streams which were delivered by this vehicle[1] to the outlet. Each stream needs to be processed in an outlet, which takes time and incurs costs. Streams whose destination is a given outlet are processed in a simplified way, i.e. without a vehicle assignment, since they are not delivered to any further nodes of the network. But still, the time needed to process the stream without a vehicle assignment in its destination node is accounted for, along with the cost of processing.

The cargo of a stream finishing in an outlet is delivered to final destinations (post offices) outside of the model within the last mile.[2] If the time needed to implement the last mile around node i is denoted by t_i^{lm} (it involves uploading the volumes onto vehicles which perform the last mile, delivering the volumes to their final destinations, i.e. post offices, and unloading them there) and the deadline for the delivery of stream s is denoted by $\bar{t}_s$ (the time by which the stream must reach the post office within the last mile), then the last possible departure time of this stream from location i is not $close_i$, but $\min\left(close_i, \bar{t}_s - t_i^{lm}\right)$. Indeed, streams whose destination is location i must leave this location at the $\bar{t}_s - t_i^{lm}$ time at the latest, and streams for which location i is a transfer node can be processed until the $close_i$ time, and leave it by $\bar{t}_s - t_i^{lm} < close_i$.

Analogically, streams which originate in a given postal operations outlet are delivered to this outlet from outside the network within the first mile. If the time needed to implement the first mile around node i is denoted by t_i^{fm}, and the starting time of the implementation of the first mile for stream s is denoted by $\underline{t}_s$, then the effective earliest time at which this stream can enter outlet i for processing is not $open_i$, but $\max\left(open_i, t_i^{fm} + \underline{t}_s\right)$. Indeed, streams which originate in location i can leave this location at $t_i^{fm} + \underline{t}_s$, and streams for which location i is a transfer node can be processed starting from the $open_i$ time, and arrive there from $t_i^{fm} + \underline{t}_s > open_i$. Note that the first mile correction applies only to streams which originate in node i, and the last mile correction applies only to streams whose destination is location i (i.e. it does not apply to the streams which use location i as a transfer node).

[1] A vehicle can ship the entire volume of a given order or a fraction of it.

[2] In our study the last mile is neglected. It is assumed that it is optimised locally outside of the main model.

2.3. Vehicles

We assume that vehicles are available on demand in each location. This is consistent with the common practice of many postal operators who often find it profitable to outsource the maintenance of their fleet of vehicles. This implies that the objective function abstracts from the possible cost which arises if a fleet of vehicles needs to be expanded (i.e. vehicles need to be purchased), but involves solely the costs of the operations of the fleet, so the costs incurred by the daily operations of the postal operator. Each vehicle has its technical characteristics, namely:

a) the capacity in terms of the number of postal pallets it can carry $vehVolPall_i > 0$. Although the number of pallets must be an integer, we consider fractional numbers as well, which represent a pallet which is not fully loaded;
b) the capacity in terms of the mass of the volume it can carry $vehVolKG_i > 0$;
c) the cost of vehicle utilisation per kilometre $vehCostKM > 0$;
d) the cost of vehicle utilisation per hour $vehCostH > 0$;
e) the cost of a driver per hour $vehCostDrvH > 0$.

2.4. Execution of streams

Let $k_s \in N$ denote the origin node, and $l_s \in N$ the destination node of stream $s \in S$. The volume associated with stream s, which flows over the edge (i, j), will be denoted by v_{sij}. Naturally, $v_{sij} \geq 0$ and $v_{sij} \leq D_s$, where $D_s > 0$ denotes the volume (demand) of stream s. Since k_s and l_s are the origin node and the destination node of stream s, we know that $v_{sk_sj} > 0$ for at least one $j \in N \backslash \{k_s\}$, and $v_{sil_s} > 0$ for at least one $i \in N \backslash \{l_s\}$. If the flow of volume from the origin node k_s to the destination node l_s is a direct one, then $v_{sk_sl_s} > 0$. If a transfer connection is used (with one transfer node), we have $0 < v_{sk_sw_{sp}} = v_{sw_{sp}l_s} \leq v_s$ for some indices $p \in \{1,2,\dots,m_s\}$, where $1 \leq m_s \leq n-2$ denotes the number of transfer connections which are used for stream s, and, naturally, $w_{sp} \in N \backslash \{k_s, l_s\}$. It can also be the case that none such w_{sp} node exists in the solution (i.e. $m_s = 0$), which means that stream s is processed using a (single) direct connection. It is also possible that a given stream is processed using only transfer connections, in which case $v_{sk_sl_s} = 0$.

2.5. The search space

An order can be processed in the form of a direct connection or in the form of a transfer connection. There is one way of processing an order in the form of a direct connection, whereas in relation to a transfer connection (or connections), an order can be processed in $\sum_{k=1}^{n-2}\binom{n-2}{k}=2^{n-2}-1$ ways, therefore altogether there are 2^{n-2} possibilities. Since deliveries from each node of the network are carried out to all other nodes, there are n^2-n orders placed. Hence, there are $2^{n(n-1)(n-2)}$ possible solutions, without taking into account time and capacity constraints. We have $n\approx 40$, which yields 2^{59280} ways in which orders can be processed, and this is without vehicle assignment. This precludes any approach based on direct enumeration of solutions.

3. The Algorithm

To optimise the problem outlined in the previous section, we employ a stochastic heuristic procedure which belongs to the family of genetic algorithms (see Boussaïd et al., 2013; Dréo et al., 2006 or Katoch et al., 2021), which is one of many possible approaches that are used in case of logistic problems (see Arnold & Sörensen, 2019; Baker & Ayechew, 2003; Baldacci et al., 2010; Cordeau et al., 2002; Laporte et al., 2000, or Prins, 2002). To employ such a method, a solution must be encoded in an appropriate way, as e.g. Kadri and Boctor (2018) demonstrate.

In the following parts of the paper, we present a simplified narrative, i.e. our point is made without a detailed discussion of the possible formulations, in the form of a mixed-integer program (MIP). The implementation of the problem we deal with in the form of an MIP, which is then solved with techniques such as branch and bound, cutting plane, etc., would require the introduction of many additional variables, like index binaries showing if any volume is shipped through a given edge for each stream or not, time-related variables, etc. (see Granada-Echeverri et al., 2019; Rieck & Zimmermann, 2010 or Theurich et al. (2021). Also, as shown by Boland et al. (2017), in such a case the reformulation of the problem in the form of a time-extended graph is possible.

3.1. Representation of solutions

Problem formulation presented in the previous section can be translated into a mathematical programming form with variables $v_{sij}\geq 0$, which denote volumes of stream $s\in S$ assigned to edges (i,j) for $i,j\in N$, $i\neq j$, and $veh_{ij}\in\mathbb{Z}_+$, which denote a number of vehicles operating on edge (i,j). This would yield s olutions of the form $\left(v_{sij}, veh_{ij}, s\in S, i,j\in N, i\neq j\right)$ belonging to search space $\left(S\times\mathbb{R}_+^{n(n-1)},\mathbb{Z}_+^{n(n-1)}\right)$. Although a genetic algorithm could, in principle, be applied to such a solution representation, there are more natural, concise and efficient ways in which solutions can be represented and genetic operations performed on them.

By 'natural' we mean that the representation of solutions resembles a constructive nature of a genetic algorithm, i.e. the fact that it constructs new solutions from the olds ones (in the process of a crossover). A mathematical programming formulation is generic and it does not reflect the inner workings of any optimisation technique, including that of a genetic algorithm. We will employ an approach in which the representation of solutions resembles the physical process of the construction of routes within a logistic plan.

By concise we mean that only relevant information is stored in the structure (object) which represents the solution. For example, we cannot rule out that in most situations (problem instances), at each stage of the search space exploration, most variables v_{sij} will be equal to zero, since most routes do not make sense from the economic point of view. Moreover, a set of such variables can vary at different stages of the optimisation for the algorithm runs which coordinate the search towards different local minima, therefore most of such variables cannot be set to zero *a priori*. As a consequence, potentially large objects need to be stored in memory, with significant access time to their elements, capable of representing many and diverse solutions, although only a fraction of the solutions is effectively processed, i.e. visited in the process of the search space exploration.[3] We will employ an approach in which the representation of solutions is minimal, i.e. only the relevant information is stored.

Finally, the criterion of efficiency is related to the fact that crossover and mutation operators either produce feasible solutions, in which case it has to be a tailor-made operator, compatible with constraints imposed in the problem in question, or a repair operator must be applied to the result of the work of the operator, so that the produced solutions, possibly infeasible at first, are projected[4] on the feasible region of the search space. It is also possible that the algorithm allows search space explorations through infeasible regions, but, in either case, final solutions produced by the optimiser must be feasible, for example due to increasing penalty applied to the objective function for infeasibility. For the problem in question, due to a multitude of the imposed feasibility constraints and due to the way in which crossovers and mutations are typically specified for problems formulated as mixed integer mathematical programmes using variables like v_{sij} and veh_{ij}, no matter which approach is adopted, the production of feasible solutions can be computationally expensive,[5] especially if based on stochastic trials until a feasible solution is obtained, or can substantially restrict the way in which new solutions are

[3] Moreover, sparse matrices do not prove an appropriate approach due to efficiency reasons. See below.

[4] The term projection is used here in a casual way, without a strict mathematical meaning of a projection operator.

[5] Crossovers are inherently stochastic procedures, which means that producing a feasible solution by chance can take substantial computational time.

constructed from the old ones, when tailored-made feasibility-preserving operators are employed. The caveat is that such feasibility-preserving operators tend to lead to a shrinkage of the subset of the feasible region that can be effectively explored by the solver.

In both the above-described cases, the efficiency of the search space exploration deteriorates – either due to a computational burden or a low diversity of the exploration which, at the end of the day, also translates into an increased computational burden: with less diversity in the search space exploration, more iterations are needed to explore the search space effectively. In the further parts of the paper, we will take advantage of the natural way in which solutions are represented (as discussed above), so that efficient feasibility-preserving operators of reproduction and mutation can be specified. Our representation of the solution is, informatively, equivalent to a specific representation in the form of a mathematical programme, i.e. a bijective correspondence between these two representations can easily be obtained.

Let x_s denote the representation of the way in which stream $s \in S$ is executed. If the delivery of stream s involves a direct connection, then

$$d(x_s) = \big((i,j), v_{sd}\big), \tag{1}$$

where $(i,j) = (k_s, l_s)$ and $0 < v_{sd} \leq v_s$ represents the volume of s which is delivered using a direct connection. Note that $(i,j) = (k_s, l_s)$ does not have to be stored in x_s explicitly, since the structures $(k_s, s \in S)$ and $(l_s, s \in S)$ must be stored anyway. If the delivery of s does not involve a direct connection, then[6] $d(x_s) = \emptyset$ and $v_{sd} = 0$. If the delivery of stream s involves at least one transfer connection, then

$$t(x_s) = \Big(\big((i, w_{sp}, j), p \in \{1,2,\dots,m_s\}\big), \big(v_{sp}, p \in \{1,2,\dots,m_s\}\big)\Big), \tag{2}$$

where, again, $(i,j) = (k_s, l_s)$, $w_{sp} \in N \backslash \{k_s, l_s\}$ with $w_{sp_a} \neq w_{sp_b}$ for any $p_a, p_b \in \{1,2,\dots,m_s\}$, $p_a \neq p_b$, and $0 < v_{sp} \leq v_s$ represents the volume of s which is delivered using a transfer connection that passes through node w_{sp}, i.e. the vehicle travelling from k_s to w_{sp} stops in the postal operations outlet located in w_{sp}, and the stream v_{sp} is processed in w_{sp}. Otherwise $t(x_s) = \emptyset$, and each $v_{sp} = 0$.

A solution is represented by $x = (x_s, s \in S)$. Such a representation is natural, since routes themselves are stored, along with volumes flowing through the edges. It is also concise, since only used connections are physically represented. As we will see, it also allows the implementation of efficient reproduction and selection operators.

[6] We use symbol $\emptyset$ to denote any null, empty or nonexistent data structure.

To give an example, let us consider a small problem with three nodes $V = \{1,2,3\}$, and three streams s_1, s_2 and s_3, with origins and destinations given by $\left(k_{s_1}, l_{s_1}\right) = (1,2)$, $\left(k_{s_2}, l_{s_2}\right) = (1,3)$, and $\left(k_{s_3}, l_{s_3}\right) = (3,1)$, and associated volumes equal to $v_{s_1} = 10$, $v_{s_2} = 20$ and $v_{s_3} = 5$. Let us assume that stream s_1 is executed using a direct connection only, therefore we have $d\left(x_{s_1}\right) = ((1,2), 10)$ and $t\left(x_{s_1}\right) = \emptyset$. Similarly, let us assume stream s_2 is executed using a direct connection (volume of 12) and a transfer connection (volume of 8), therefore we have $d\left(x_{s_2}\right) = ((1,3), 12)$ and $t\left(x_{s_2}\right) = ((1,2,3), 8)$. Finally, let us assume that stream s_3 is executed using a transfer connection only, therefore we have $d\left(x_{s_3}\right) = \emptyset$ and $t\left(x_{s_3}\right) = ((3,2,1), 5)$.

3.2. Initial population

For genetic optimisation, a population of solutions is needed, which we denote by $P_k = \{x_i, i = 1,2, \dots, |P|\}$, where $k \geq 0$ is an iteration index of the algorithm (with P_0 being the initial population), and $|P|$ denotes the size of the population, which is kept constant through the iterations. The initial population consists of $|P|$ solutions, each of which is constructed according to the procedure outlined below, which is sequentially applied to each of the m streams.

Firstly, the number of transfer connections is drawn from set $\{0,1,2, \dots, \min(\lceil v_s \rceil, n-2)\}$, according to probabilities proportional to 2^{-k}, respectively. If the sampled number of transfer connections is bigger that 0, we sample, uniformly and without replacement, respective routes for transfer connections $\left(k_s, w_{sp_i}, l_s\right)$ with $w_{sp_i} \in N \setminus \{k_s, l_s\}$ for $i = 1,2, \dots, |t(x_s)|$. Secondly, if $|t(x_s)| < n-2$, a direct connection (k_s, l_s) is added to x_s with the probability of 1 or p_d, respectively. Finally, the volume v_s gets distributed over the connections of x_s. Since there are, by construction, $|t(x_s)| + |d(x_s)| \geq 1$ connections within x_s, volume v_s is partitioned into volumes v_{si}, $i = 1,2, \dots, |t(x_s)| + |d(x_s)|$, such that $v_{si} > 0$ and $\sum_{i=1}^{|t(x_s)|+|d(x_s)|} v_{si} = v_s$. This makes the explicit construction of $t(x_s)$ and $d(x_s)$ possible, and therefore the construction of x_s as well. The procedure is applied for each $s \in S$, which yields generic solution x_i. The procedure is repeated for $i = 1,2, \dots, |P|$.

3.3. Genetic operators

A genetic algorithm usually consists in the sequential application of the following three genetic operators: selection, reproduction and mutation. Selection is responsible for the stochastic promotion of high quality solutions to the reproduction phase (see Vajda et al., 2008), which in turn combines structural features of solutions producing new solutions, possibly of higher quality than the old ones (see Kora & Yadlapalli, 2017 or Umbarkar & Sheth, 2015).[7] Mutation, by introducing random disturbances into offspring produced by reproduction, is mainly responsible for increasing the diversity of the search space exploration and for the prevention of premature convergence, as Lim et al. (2017) and Squillero and Tonda (2016) demonstrate. Our implementation of the genetic algorithm follows the idea, hence we have $P_{k+1} = M\left(R\big(S(P_k)\big)\right)$, where S stands for selection, R for reproduction, and M for mutation operators.

3.3.1. Selection

A tournament selection is employed as a selection operator (see Bäck et al., 2018). It boils down to selecting $|P|$ times the two solutions $x, y \sim U(P_k)$ from P_k, i.e. two uniform draws from P_k, with the replacement and the promotion of x to the reproduction phase if $f(x) > f(y)$, and the promotion of y otherwise. We augment, however, the typical formulation of a tournament selection by the rule of elitism, which assumes that the $\alpha \in (0,1)$ share of the best solutions from P_k is automatically promoted to the recombination phase, and they constitute share $(0,1) \ni \beta \geq \alpha$ of parental pool $S(P_k)$. If $\beta > \alpha$, then the $(\beta - \lambda)|P|$ solutions, which are integer by construction, randomly drawn from the α share of P_k, are duplicated. The above shows how $S(P_k)$ is constructed.

3.3.2. Reproduction

After the choice of the selection operator, we obtain population $S(P_k) = (p_1, p_2, \dots, p_k, p_{k+1}, \dots, p_{|P|-1}, p_{|P|})$, which contains $\frac{|P|}{2}$ ordered tuples of solutions in the form of (p_k, p_{k+1}) for $k = 1,3, \dots, |P| - 1$. Each such pair of solutions – parents (p_k, p_{k+1}) produces a new pair of solutions – offspring (o_k, o_{k+1}), according to a *1-stream-interchange procedure*, which we define below.

First, index q is drawn uniformly from set $\{1,2, \dots, m\}$, the elements of which correspond to streams $s \in S$. The letter m denotes the number of streams in a given problem instance, i.e. $|S| = m$. So far it has not been important to emphasize, but the solution $x = (x_s, s \in S)$ comes in the form of an ordered set, a sequence, therefore streams $s \in S$ are indexed and we can equivalently represent the solution as $x = (x_k, k = 1,2, \dots, m)$, where x_k represents the way in which the k-th sequence is executed in solution x.

[7] Obviously, the recombination can also produce offspring worse than 'parents', in particular with the objective function below the average in the population, but such offspring, in terms of the expected value, will not be promoted by the selection operator in the next iteration.

Secondly, the execution of the q-th stream in solution p_k, denoted by $x_q(p_k)$ and represented by $d\left(x_q(p_k)\right)$ and $t\left(x_q(p_k)\right)$, is interchanged with the execution of the q-th stream in solution p_{k+1}, denoted by $x_q(p_{k+1})$ and represented by $d\left(x_q(p_{k+1})\right)$ and $t\left(x_q(p_{k+1})\right)$. Such an interchange operation can be represented as:

$$\left(d\left(x_q(p_k)\right), t\left(x_q(p_k)\right)\right) \leftrightarrow \left(d\left(x_q(p_{k+1})\right), t\left(x_q(p_{k+1})\right)\right),$$

or, in short, as $x_q(p_k) \leftrightarrow x_q(p_{k+1})$. The above demonstrates how $R\big(S(P_k)\big)$ is constructed.

3.3.3. Mutation

Mutation introduces changes in each stream x_s of each solution $x \in R\big(S(P_k)\big)$ independently, with an exogenous probability of $p_m \in (0,1)$. We assume three possible changes, employed with equal probabilities. The first change consists in removing a random connection from stream x_s. The connection to be removed is chosen uniformly from $\{1,2,\dots,|t(x_s)| + |d(x_s)|\}$. Indeed, if $|d(x_s)| = 1$, then index 1 represents a direct connection, and indices $2,3,\dots,|t(x_s)| + |d(x_s)|$ represent consecutive transfer connections. If $|d(x_s)| = 0$, indices $1,2,\dots,|t(x_s)| + |d(x_s)|$ represent transfer connections.

The removal of a connection implies that some volume, say v, needs to be distributed over the remaining $|t(x_s)| + |d(x_s)| - 1$ connections. It can happen that $|t(x_s)| + |d(x_s)| - 1 = 0$, so in other words, that x_s contained exactly one route, which was removed. In such a case, a random connection is added to x_s, either a direct or a transfer one, with probabilities p_d and $1 - p_d$, respectively, and the volume from the removed connection is assigned to it. If $|t(x_s)| + |d(x_s)| - 1 \geq 1$, then there is at least one route which is left after the removal of the sampled route.

The second possible change involves adding a random route to stream x_s. The added route can either be a direct connection, if x_s does not have one, or a transfer connection, if $|d(x_s)| < n - 2$. If x_s does not already have a direct connection and it is possible to add a transfer connection, then the choice is made randomly with the probability of $p_d{}'$ and $p_t = 1 - p_d{}'$, respectively. If a transfer connection is added, then the node for the connection is drawn uniformly from $N \setminus \left\{\{k_s, l_s\} \cup \{w_{sp}, p = 1,2,\dots,m_s\}\right\}$. If x_s already has a direct connection, then a transfer connection is added. When a direct or transfer connection is added, a random proportion of the volume of some other connection is shifted to it. The connection from which the volume is shifted is chosen uniformly. If the shifted volume equals the entire volume of the connection in question, it is then removed from x_s. If a volume is perfectly divisible, this does not happen, but when it has to be modelled as an integer value, such a situation can happen.

The third possible change reallocates the volume within the existing connections. More specifically, two connections are drawn at random, and their volumes are interchanged. In the way described above genetic operators are implemented.

4. Numerical Experiments

In this section, we present and discuss a selection of case studies. Our aim here is to verify the desired features of the algorithm, and to provide some insight into the ability of the algorithm to guide brownfield analyses for realistic-size problem instances.

Case studies are divided into the following two groups:

a) stylised problem instances – to verify if the algorithm is capable of selecting direct and transfer connections appropriately;
b) practical problem instances – to observe convergence and to compare the generated solution with a benchmark. The way to verify the solution is to check if it is satisfactory from the practical point of view. A possible practical benchmark is a solution in which all volumes are shipped directly from the origins to the depots.

4.1. Stylised case studies

To give an example of stylised case studies, we will demonstrate how it can be experimentally confirmed that the algorithm is capable of producing good-quality solutions in three cases:

a) when streams should be executed directly;
b) when merging of streams is optimal;
c) when there are distinct areas in which shipments should be executed separately.

Case a). Assume that the objective function is given by the total length of all routes of all streams counted separately, i.e. without merging volumes:

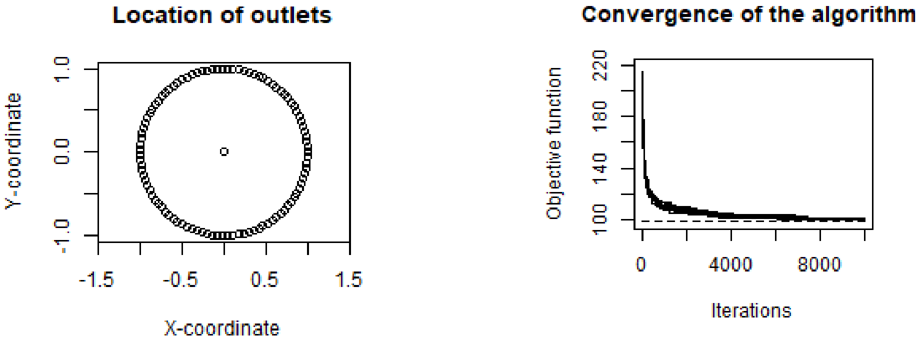
$$f(x) = \sum_{s \in S} \left(\chi_{sd} d_{(k_s,l_s)} + \chi_{st} \left(\sum_{p=1}^{m_s} d_{(k_s,w_{sp})} + d_{(w_{sp},l_s)} \right) \right), \tag{3}$$

where:

$$\chi_{sd} = \begin{cases} 1, & d(x_s) \neq \emptyset \\ 0, & d(x_s) = \emptyset \end{cases} \text{and}\, \chi_{st} = \begin{cases} 1, & t(x_s) \neq \emptyset \\ 0, & t(x_s) = \emptyset \end{cases}, \tag{4}$$

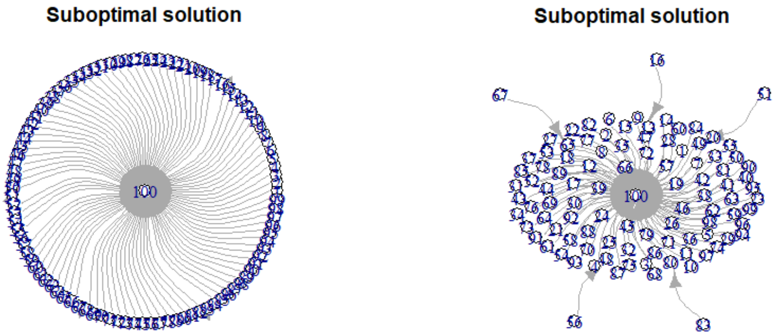
i.e. χ_{sd} and χ_{st} are indicator variables which show if x_s involves a direct connection (χ_{sd}), or at least one transfer connection (χ_{st}), and $d_{(i,j)}$ denotes the distance between locations $i \in N$ and $j \in N$. In such a case, the optimal solution boils down to using only direct connections, no matter how the network is organised. We also assume that $v_s = v$ for all $s \in S$ and that $vehCap = v$ for all vehicles, but due to the specification of f, having $vehCap > v$ does not change the fact that the optimal solution involves only direct connections. For the graphical presentation of the results, we assumed that depots of respective streams are located in points (x_i, y_i), $i = 1,2,\dots,n$, on a circle with the center in (0,0), i.e. for $x_i = rcos(\phi_i)$ and $y_i = rsin(\phi_i)$, where $\phi_i = i\frac{2\pi}{n+1}$. The centre of the circle serves as a single destination point for all streams, as shown in Figure 1.

Figure 1. Location of origin nodes – the circle, and of the destination node – the center, convergence of optimisation



Source: authors' calculations based on stylised data described in Case a).

Figure 2. Best suboptimal solution generated by the algorithm



Source: authors' calculations based on stylised data described in Case a).

For $n = 100$, which is large enough from the practical perspective, we know that the optimal value of the objective function is $f = 99$, hence it is easy to observe how the algorithm converges. Figure 1 plots the value of the objective function over 10,000 iterations for 10 independent runs. The dashed horizontal line presents the level of the optimal solution. In all runs, the algorithm converges closely to the optimal value. Figure 1 shows 10,000 iterations with f converging to, respectively: 101.52, 100.13, 100.64, 100.32, 102.06, 100.20, 100.68, 100.82, 101.08 and 100.39.

Figure 2 plots the best sub-optimal solution generated by the algorithm, with $f(x) = 100.13$, which is within 1.1% from the optimal solution. The left panel presents the solution with outlets aligned as in the input data of the problem. We know that there are transfer connections in this solution, but the alignment of the locations makes them hard to see. Therefore, on the right panel, we also present the same sub-optimal solution, but with a different alignment of nodes, so that we can observe that there are five transfer connections still left in the solution.[8]

The conclusion is that the employed genetic operators allow the selection of solutions based on direct connections if the problem instance in question requires such coordination.

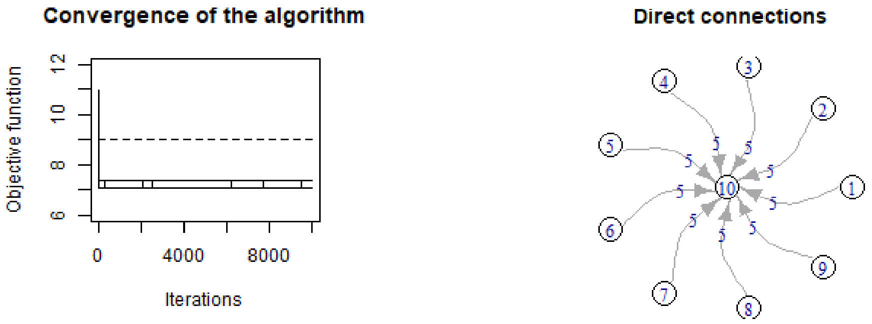
Case b1). Assume that the locations of all the outlets are the same as in Case a), and also that the volume of each stream stays the same, i.e. $v_s = v$ for all $s \in S$. Assume also that the objective function is modified, so that it is beneficial to merge volumes:

$$f(x) = \sum_{\substack{(i,j)\in N^2 \\ i\neq j}} veh_{ij} d_{(i,j)}, \tag{5}$$

where veh_{ij} represents the number of vehicles assigned to the (i,j) edge, i.e. the minimal number of vehicles capable of loading volume $v_{ij} = \sum_{s\in S} v_{sij}$ which flows through this edge. To make the merging of volumes possible, the capacities of vehicles are increased to a value larger than v. We assume that $vehCap = 3v$ for each vehicle. Now it seems reasonable to set up local hubs and coordinate the direct flows to the destination point from the hubs.

[8] Within 25,000 iterations, all the runs converged to the optimal solution.

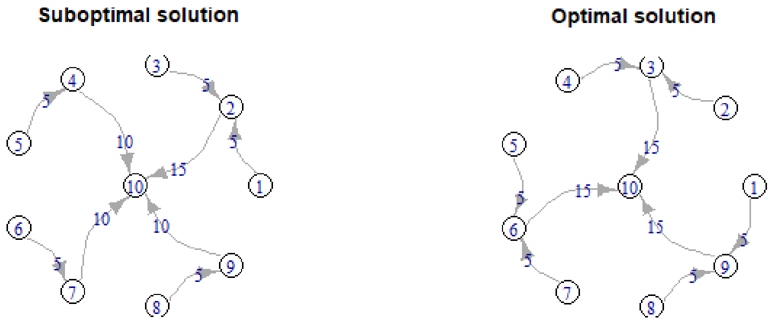
Figure 3. Convergence of the objective function and solution based on direct connections only



Source: authors' calculations based on stylised data described in Case b1).

For clarity, in Figures 3 and 4 we present the results for $n = 10$, but an analogical behaviour of the algorithm is observed for greater values of n.[9] Figure 3 shows the trajectory of the objective function over 10,000 iterations for 10 independent runs of the algorithm, where the dashed horizontal line represents the value of the objective function when only direct connections are used, with $f = 9$ (the left panel), along with a solution which uses only direct connections (the right panel). Figure 4 presents two types of solutions generated by the algorithm, one of which is optimal with $f(x) = 7.10$ (the right panel), while the other is sub-optimal with $f(x) = 7.42$ (the left panel). As arrows on the figure suggest, we assumed that $v = 5$.

Figure 4. The suboptimal solution and the optimal solution with coordination of volumes in selected nodes

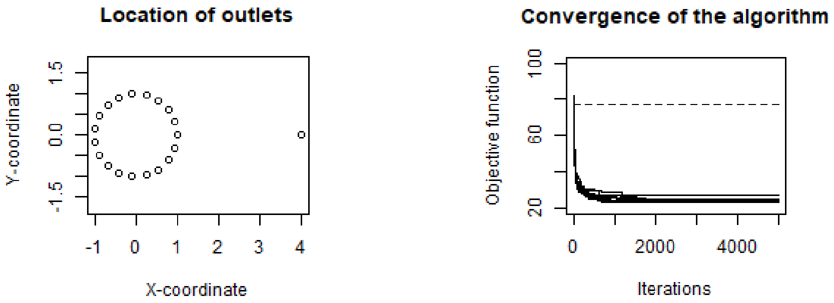


Source: authors' calculations based on simulated data described in Case b1).

[9] Since emergent properties of solutions, i.e. the selection of direct connections, when needed, and the selection of transfer connections with the coordinated merging of volumes, when needed, come as a consequence of the way in which genetic operators were implemented, they do not depend on the size of the problem. Efficiency is bigger for smaller test cases, therefore in the current section we also include the presentation of a real-life size of the problem.

Case b2). Now let us assume that the locations of nodes are given as presented in the left panel of Figure 5. Location of origin nodes is the same as in Case a) and b1), with the exception of the location of the single destination node, which was shifted from $(0,0)$ to $(0,4)$. Let us assume that $vehCap = 100v$, which in practical terms means that the capacities of vehicles are unlimited. The objective is defined as in Case b1). The right panel of Figure 5 presents the trajectory of the objective function over $5{,}000$ iterations for 10 independent runs of the algorithm. The dashed horizontal line represents the value of the objective function when only direct connections are used.

Figure 5. Location of origins and destinations and the convergence of the objective function



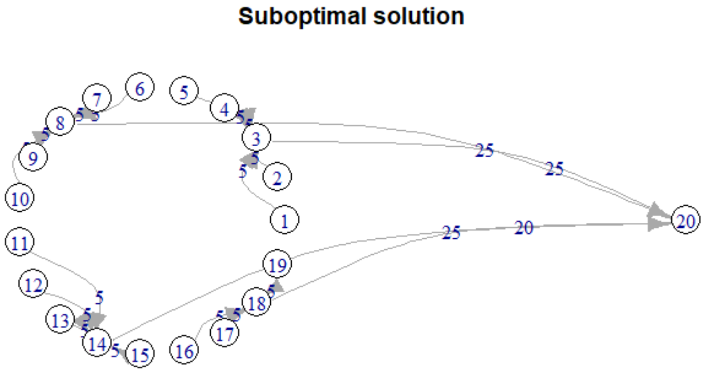
Location of outlets

Convergence of the algorithm

Source: authors' calculations based on stylised data described in Case b2).

Figure 6 presents the optimal solution generated by the algorithm with $f(x) = 23.25$. For clarity, we present an example of a solution for $n = 20$, but an analogical behaviour of the algorithm is observed for larger values of n. The figure shows that the algorithm is capable of coordinating the execution of streams by setting up four hubs – for regions NE, NW, SE and SW, to which volumes are shipped from nearby locations and from which merged volumes are shipped by single vehicles directly to the destination location. Please note that streams originating in locations selected as hubs are shipped directly to their destination points. The conclusion from Cases b1) and b2) is that the employed genetic operators are capable, whenever necessary, of selecting hubs in which volumes are merged and from which they are transferred directly to their destinations.

Figure 6. A solution in which four locations are selected as hubs where volume is aggregated and transported to the destination location by direct connections

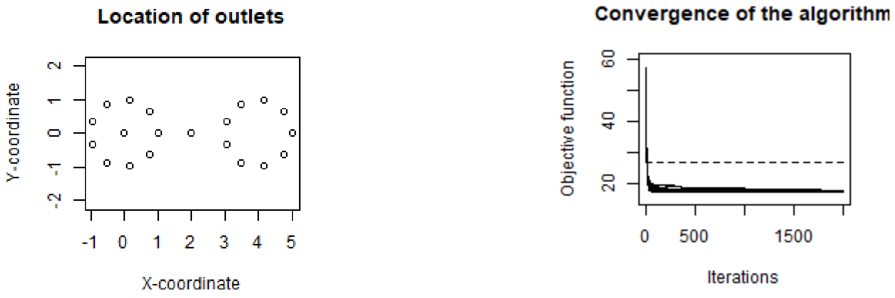


Suboptimal solution

Source: authors' calculations based on stylised data described in Case b2).

Case b3). This case is analogical to b2), but we introduce two regions – one consists of locations on the circle centred at (0,0), as in the a) and b1) Cases, and the other is analogical to the b2 Case, but with the circle centred in (4,0) and the single destination in (2,0). In Figure 7, locations are presented on the left panel and the convergence of the objective on the right one (the dashed line represents the benchmark solution with $f = 28.14$). Figure 8 presents both the optimal and the suboptimal solutions, the latter being very close to the optimal solution as far as the value of f is concerned ($f = 17.47$ for the optimal solution, and $f = 17.49$ for the suboptimal one). Note that although the difference in the objective is small in the two presented instances, the implied logistic organisation of the East region is substantially different, especially as far as the status of node 16 is concerned (which was given the status of a hub node in the right case, but no such status in the left one). Also the hub status in the NE region is interchanged between nodes 12 and 13.

Figure 7. Location of depots and destinations, convergence of the objective function

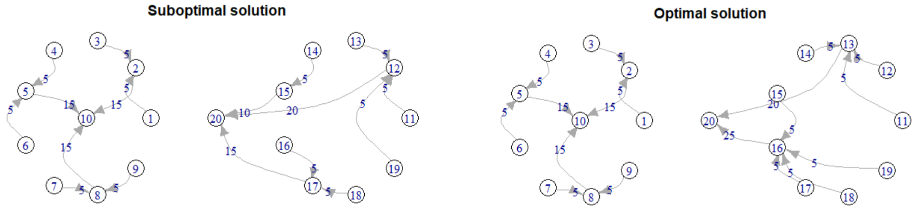


Location of outlets

Convergence of the algorithm

Source: authors' calculations based on stylised data described in Case b3).

Figure 8. Two solutions with a very similar value of the objective function, yet with a substantially different organisation of the East region



Source: authors' calculations based on stylised data described in Case b3).

4.2. Practical case study

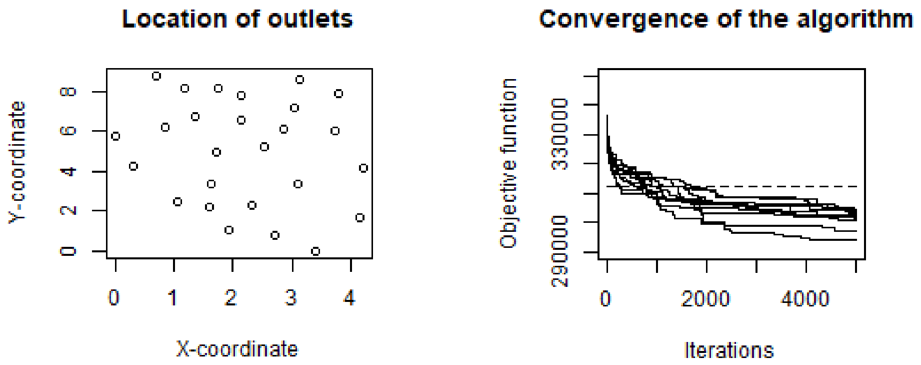
To show some sample results of a brownfield analysis supported by our algorithm, let us consider a problem in which locations correspond to 26 chosen facilities from the logistic network of an existing postal service provider (left panel of Figure 9), with distances and travel times calculated on the basis of real measurements, with realistic assumptions regarding demands and technical characteristics of vehicles and outlets, and with the objective defined as the cost of the execution of the plan.[10] The cost is calculated by a simulation of the execution of the solution and the calculation of the following variables: V – total volume processed, i.e. $V = \sum_{i \in N} v(i)$, where $v(i)$ denotes the total volume processed in outlet $i \in N$, D – total distance travelled by vehicles, T – total time in which the vehicles travelled, T_D – total time of the work of drivers (which is longer than T since it includes situations when drivers have to wait for the volume to be processed in transfer nodes). These variables are then crossed with cost characteristics $cVol$, $vehCostKM$, $vehCostH$ and $vehCostDrvH$. The right panel of Figure 9 presents the trajectory of the objective function for 1,000 iterations for 10 independent runs of the algorithm. The dotted line at $f = 312{,}300$ represents the cost of the benchmark solution. We present a sample solution which is at least 5% better than the benchmark. This suboptimal solution obtains $f = 294{,}000$. Figure 10 presents the outflows and inflows of the volume for the respective nodes of the network, for the benchmark solution (in blue) and for the presented solution (in red). Please note that for each node, both the outflow and the inflow of volume is larger for the suboptimal solution than for the benchmark one, in which outflows and inflows correspond to volumes originating and finishing in the respective nodes. The residual outflow and inflow (the net flow) equals the sum of the total outflow from a node decreased by the outflow which

[10] Due to property rights, all data were transformed, so that units do not represent any meaningful quantities.

originates in the node and the total inflow of the node decreased by the inflow which finishes in that node. The net flow grows due to the utilisation of transfer connections and the merging of volumes.

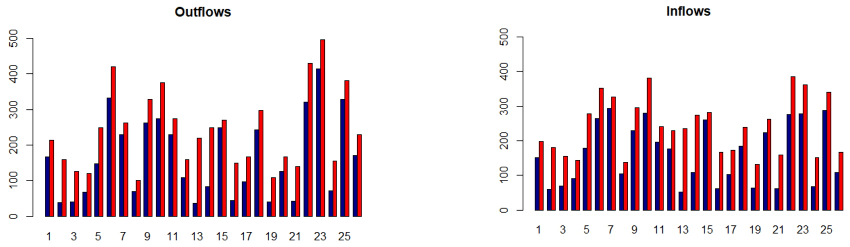
Net flows are shown in Figure 11 as nominal values (the upper panel), as percentages of outflows and inflows of the suboptimal solution (the middle panel), and as percentages of outflows and inflows of the benchmark solution (the lower panel). When the net flow is big in absolute terms, as well as relatively to the outflow which originates in the node and relatively to the inflow which finishes in the node, the node is identified as a hub. Note that, for example, nodes 13 and 14, which, relatively to some other nodes in the network, are not the origin or destination of any great volume of shipments, are considered as prospective hubs, whereas nodes 22 and 23, where a substantial amount of volume originates and finds destination, are considered so to a lesser extent.

Figure 9. Location of points and trajectory of the objective function



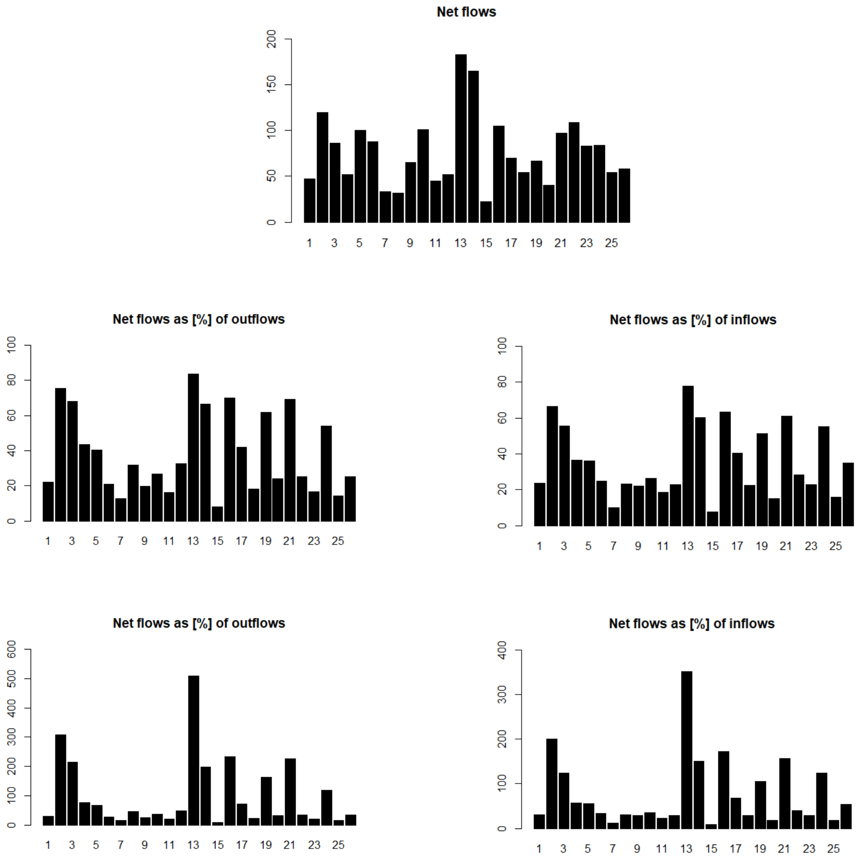
Source: authors' calculations based on a subset of scaled real data.

Figure 10. Inflows and outflows of volume for respective facilities for the benchmark solution (in blue) and for the suboptimal one (in red)



Source: authors' calculations based on a subset of scaled real data.

Figure 11. Net flows assigned to outlets in terms of nominal values (upper panel), [%] of outflows and inflows of the suboptimal solution (the middle panel), [%] of outflows and inflows of the benchmark solution (lower panel)



Source: authors' calculations based on a subset of scaled real data.

5. Conclusions

We studied a realistic variant of a postal delivery problem with pickups and deliveries, split pickups and deliveries, time constraints and capacity constraints, the latter imposed both on vehicles and operations outlets. Our main conclusion is that the implementation of the genetic algorithm we used (with a relatively simple specification of genetic operators) makes it possible to select streams which should be delivered directly and those which should be merged with other streams along the routes, and allows the determination of transfer nodes within the network. The implementation with such features, when used in a real-life case study with relaxed constraints on the capacities of outlets, allows the determination which outlets will be able to serve as prospective hubs when the network expands. From the practical point of view, such a conclusion means that the algorithm in question can be helpful in the brownfield type of analyses of postal companies, especially large ones, which operate on a country-wide infrastructure. Strong trends that we nowadays observe on the market of postal shipments, which are driven e.g. by a substantial surge in the volumes of orders that consumers place via the Internet (vs. stationary shopping), which will most probably continue and strengthen in the future, pose a serious challenge for the operator. On the other hand, such a situation opens the way for the development of optimisation techniques in the process of both tactical and strategic planning. Our research focused on one aspect of this kind of planning, namely on the identification of prospective hubs within the postal network, and on the approach to this problem. We assumed that transfer connections could pass through arbitrary transfer locations, which allowed the identification of hub locations. The next step would be to introduce an optimisation model in which hub locations would be fixed and transfer connections could be executed only through them, whereas peripheral locations would never serve as transfer nodes. We plan to add such second part of the model to obtain a closed, two-step procedure for postal optimisation. Moreover, in the model presented in this study, we accounted only for the overall time feasibility constraints, i.e. the time of logistic operations and transport which did not exceed the time windows in which outlets operate. The second part of the model would need to take into consideration time constraints resulting from vehicles' queuing in the network nodes and from the necessity of waiting in a hub location for the volumes originating from peripheral locations to be merged there. The above would be a meaningful follow-up to the study presented in this paper.

Acknowledgements

The research results presented in this paper were obtained with the support of the National Centre for Research and Development's grant no. POIR.04.01.04-00-0054/17-00.

References

Archetti, C., & Speranza, M. G. (2012). Vehicle routing problems with split deliveries. *International Transactions in Operational Research*, *19*(1–2), 3–22. https://doi.org/10.1111/j.1475-3995.2011.00811.x.

Arnold, F., & Sörensen, K. (2019). What makes a VRP solution good? The generation of problem-specific knowledge for heuristics. *Computers & Operations Research*, *106*, 280–288. https://doi.org/10.1016/j.cor.2018.02.007.

Baker, B. M., & Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, *30*(5), 787–800. https://doi.org/10.1016/S0305-0548(02)00051-5.

Bäck, T., Fogel, D. B., & Michalewicz, Z. (Eds.). (2018). *Evolutionary Computation 1: Basic Algorithms and Operators*. CRC Press. https://doi.org/10.1201/9781482268713.

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.

Baldacci, R., Toth, P., & Vigo, D. (2010). Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*, *175*(1), 213–245. https://doi.org/10.1007/s10479-009-0650-0.

Baumung, M. N., Gündüz, H. I., Müller, T., & Sebastian, H.-J. (2015). Strategic Planning of Optimal Networks for Parcel and Letter Mail. In H.-J. Sebastian, P. Kaminsky & T. Müller (Eds.), *Quantitative Approaches in Logistics and Supply Chain Management* (pp. 81–103). Springer. https://doi.org/10.1007/978-3-319-12856-6_4.

Benjamin, A. M., & Beasley, J. E. (2013). Metaheuristics with disposal facility positioning for the waste collection VRP with time windows. *Optimization Letters*, *7*(7), 1433–1449. https://doi.org/10.1007/s11590-012-0549-6.

Boland, N., Hewitt, M., Marshall, L., & Savelsbergh, M. (2017). The Continuous Time Service Network Design Problem. *Operations Research*, *65*(5), 1303–1321. https://doi.org/10.1287/opre.2017.1624.

Borčinová, Z. (2017). Two models of the capacitated vehicle routing problem. *Croatian Operational Research Review*, *8*(2), 463–469. https://doi.org/10.17535/crorr.2017.0029.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences*, *237*, 82–117. https://doi.org/10.1016/j.ins.2013.02.041.

Bräysy, O., & Gendreau, M. (2005). Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science*, *39*(1), 104–118. https://doi.org/10.1287/trsc.1030.0056.

Casazza, M., Ceselli, A., & Calvo, R. W. (2018). A branch and price approach for the Split Pickup and Split Delivery VRP. *Electronic Notes in Discrete Mathematics*, *69*, 189–196. https://doi.org/10.1016/j.endm.2018.07.025.

Çetiner, S., Sepil, C., & Süral, H. (2010). Hubbing and routing in postal delivery systems. *Annals of Operations Research*, *181*(1), 109–124. https://doi.org/10.1007/s10479-010-0705-2.

Crevier, B., Cordeau, J. F., & Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, *176*(2), 756–773. https://doi.org/10.1016/j.ejor.2005.08.015.

Cordeau, J.-F., Gendreau, M., Laporte, G., Potvin, J.-Y., & Semet, F. (2002). A guide to vehicle routing heuristics. *Journal of the Operational Research Society*, *53*(5), 512–522. https://doi.org/10.1057/palgrave.jors.2601319.

de Camargo, R. S., de Miranda, G., & Løkketangen, A. (2013). A new formulation and an exact approach for the many-to-many hub location-routing problem. *Applied Mathematical Modelling*, *37*(12-13), 7465–7480. https://doi.org/10.1016/j.apm.2013.02.035.

Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2006). *Metaheuristics for Hard Optimization: Methods and Case Studies*. Springer-Verlag. https://doi.org/10.1007/3-540-30966-7.

Granada-Echeverri, M., Toro, E. M., & Santa, J. J. (2019). A mixed integer linear programming formulation for the vehicle routing problem with backhauls. *International Journal of Industrial Engineering Computations*, *10*(2), 295–308. https://doi.org/10.5267/j.ijiec.2018.6.003.

Kadri, R. L., & Boctor, F. F. (2018). An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. *European Journal of Operational Research*, *265*(2), 454–462. https://doi.org/10.1016/j.ejor.2017.07.027.

Karimi, H., & Setak, M. (2018). A bi-objective incomplete hub location-routing problem with flow shipment scheduling. *Applied Mathematical Modelling*, *57*, 406–431. https://doi.org/10.1016/j.apm.2018.01.012.

Katoch, S., Chauhan, S. S., & Kumar, V. (2021). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications*, *80*(5), 8091–8126. https://doi.org/10.1007/s11042-020-10139-6.

Kerivin, H. L. M., Lacroix, M., Mahjoub, A. R., & Quilliot, A. (2008). The splittable pickup and delivery problem with reloads. *European Journal of Industrial Engineering*, *2*(2), 112–133. https://doi.org/10.1504/EJIE.2008.017347.

Keskin, M., Laporte, G., & Çatay, B. (2019). Electric vehicle routing problem with time-dependent waiting times at recharging stations. *Computers & Operations Research*, *107*, 77–94. https://doi.org/10.1016/j.cor.2019.02.014.

Kora, P., & Yadlapalli, P. (2017). Crossover Operators in Genetic Algorithms: A Review. *International Journal of Computer Applications*, *162*(10), 34–36. https://doi.org/10.5120/ijca2017913370.

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.

Leung, J. M. Y., Magnanti, T. L., & Singhal, V. (1990). Routing in Point-to-Point Delivery Systems: Formulations and Solution Heuristics. *Transportation Science*, *24*(4), 245–260. https://doi.org/10.1287/trsc.24.4.245.

Lim, S. M., Sultan, A. B. M., Sulaiman, M. N., Mustapha, A., & Leong, K. Y. (2017). Crossover and Mutation Operators of Genetic Algorithms. *International Journal of Machine Learning and Computing*, *7*(1), 9–12. http://doi.org/10.18178/IJMLC.

Nagy, G., & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research*, *162*(1), 126–141. https://doi.org/10.1016/j.ejor.2002.11.003.

Prins, C. (2002). Efficient Heuristics for the Heterogeneous Fleet Multitrip VRP with Application to a Large-Scale Real Case. *Journal of Mathematical Modelling and Algorithms*, *1*(2), 135–150. https://doi.org/10.1023/A:1016516326823.

Ralphs, T. K., Kopman, L., Pulleyblank, W. R., & Trotter, L. E. (2003). On the Capacitated Vehicle Routing Problem. *Mathematical Programming*, *94*(2–3), 343–359. https://doi.org/10.1007/s10107-002-0323-0.

Rieck, J., & Zimmermann, J. (2010). A new mixed integer linear model for a rich vehicle routing problem with docking constraints. *Annals of Operations Research*, *181*(1), 337–358. https://doi.org/10.1007/s10479-010-0748-4.

Spliet, R., & Gabor, A. F. (2015). The Time Window Assignment Vehicle Routing Problem. *Transportation Science*, *49*(4), 721–731. https://doi.org/10.1287/trsc.2013.0510.

Squillero, G., & Tonda, A. (2016). Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization. *Information Sciences*, *329*, 782–799. https://doi.org/10.1016/j.ins.2015.09.056.

Theurich, F., Fischer, A., & Scheithauer, G. (2021). A branch-and-bound approach for a Vehicle Routing Problem with Customer Costs. *EURO Journal on Computational Optimization*, *9*, 1–11. https://doi.org/10.1016/j.ejco.2020.100003.

Toth, P., & Vigo, D. (Eds.). (2002). *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics. https://doi.org/10.1137/1.9780898718515.

Umbarkar, A. J., & Sheth, P. D. (2015). Crossover operators in genetic algorithms: a review. *ICTACT Journal on Soft Computing*, *6*(1), 1083–1092. http://doi.org/10.21917/ijsc.2015.0150.

Vajda, P., Eiben, A. E., & Hordijk, W. (2008). Parameter Control Methods for Selection Operators in Genetic Algorithms. In G. Rudolph, T. Jansen, N. Beume, S. Lucas & C. Poloni (Eds.), *Parallel Problem Solving from Nature – PPSN X* (pp. 620–630). Springer. https://doi.org/10.1007/978-3-540-87700-4_62.

Wassan, N. A., & Nagy, G. (2014). Vehicle Routing Problem with Deliveries and Pickups: Modelling Issues and Meta-heuristics Solution Approaches. *International Journal of Transportation*, *2*(1), 95–110. http://dx.doi.org/10.14257/ijt.2014.2.1.06.

Przegląd Statystyczny. Statistical Review, 2021, vol. 68, 3, 41–46
Received: 27.10.2021, revised: 28.10.2021, accepted: 2.11.2021

Report from the 30th Scientific Conference: 'Classification and Data Analysis – Methodology and Practice'

Krzysztof Jajuga,[a] Grażyna Dehnel,[b] Marek Walesiak[c]

The 30th Scientific Conference: 'Classification and Data Analysis – Methodology and Practice' (the 25th Taxonomic Conference) of the Classification and Data Analysis Section of the Polish Statistical Association (SKAD) took place on 8–10 September 2021, in Poznań, Poland. The conference was organised by the Department of Statistics of the Poznań University of Economics and Business and was held online through the Microsoft Teams communication platform. Grażyna Dehnel, PhD, DSc, Assoc. Prof. was the chairman of the Organising Committee, consisting of Maciej Beręsewicz, PhD, Tomasz Klimanek, PhD, DSc, Assoc. Prof., Wojciech Roszka, PhD, Marcin Szymkowiak, PhD, DSc, Assoc. Prof., Hanna Wdowicka, PhD and Kamil Wilak, PhD. Basic information about the conference is available at: https://SKAD2021.ue.poznan.pl.

The following topics were addressed during the conference:

- theoretical aspects (taxonomy, discriminant analysis, linear ordering methods, multivariate statistical analysis, methods of analysing continuous variables, methods of discrete variables analysis, symbolic data analysis, graphical methods);
- practical applications (financial data analysis, marketing data analysis, spatial data analysis, other areas of data analysis application – medicine, psychology, archaeology, etc., computer application of statistical methods).

The main objective of the SKAD conference was to present the current research and to create a platform for the exchange of ideas relating to theoretical and applied aspects of classification and data analysis. This annually held forum provides an opportunity to present and promote state-of-the-art research and to indicate possible directions for its further development.

The conference featured 102 participants who are faculty members and doctoral students of the following universities and institutions: AGH University of Science and Technology, Calisia University – Kalisz, Poland, Statistics Poland, the Institute of Rural and Agricultural Development of the Polish Academy of Sciences,

[a] Wrocław University of Economics and Business, ul. Komandorska 118/120, 53-345 Wrocław, Poland, e-mail: krzysztof.jajuga@ue.wroc.pl, ORCID: https://orcid.org/0000-0002-5624-6929.
[b] Poznań University of Economics and Business, al. Niepodległości 10, 61-875 Poznań, Poland, e-mail: grazyna.dehnel@ue.poznan.pl, ORCID: https://orcid.org/0000-0002-0072-9681.
[c] Wrocław University of Economics and Business, ul. Komandorska 118/120, 53-345 Wrocław, Poland, e-mail: marek.walesiak@ue.wroc.pl, ORCID: https://orcid.org/0000-0003-0922-2323.

Iowa State University, Gdańsk University of Technology, Cracow University of Technology, RTI International, Warsaw University of Life Sciences, SGH Warsaw School of Economics, Medical University of Silesia, University of Agricultural Sciences Bangalore, University of Economics in Katowice, Cracow University of Economics, Poznań University of Economics and Business, Wrocław University of Economics and Business, University of Gdańsk, Adam Mickiewicz University in Poznań, Łódź University, Poznań University of Life Sciences, University of Szczecin, University of Białystok, the Statistical Office in Łódź, the Statistical Office in Poznań, Utrecht University, University of Southampton, West Pomeranian University of Technology in Szczecin.

54 presentations introducing research results relating to the theory and applications of classification and data analysis were delivered during 3 plenary sessions and 15 parallel sessions. The sessions were chaired by Andrzej Dudek (two sessions), Elżbieta Gołata, Iwona Markowicz, Beata Bieszk-Stolorz, Dominik Rozkrut, Kamila Migdał-Najman, Andrzej Sokołowski, Marek Walesiak, Marcin Szymkowiak, Barbara Pawełek, Jacek Kowalewski, Paweł Lula, Krzysztof Najman, Agnieszka Stanimir, Grażyna Dehnel, Ewa Roszkowska and Józef Pociecha.

Selected papers presented at the conference will be published in a book entitled 'Classification and Data Analysis – Methodology and Practice', edited by K. Jajuga, G. Dehnel, M. Walesiak, published by Springer. Below is a list of all papers presented during the conference:

Peter van der Heijden, *Multiple system estimation using covariates having missing values and measurement error: estimating the size of the Maori population in New Zealand*;

Dominik Rozkrut, *Digital transformation, data governance, data stewardship and official statistics*;

Stephanie Eckman, *Using Passive Data to Supplement or Replace Survey Data*;

Jae-Kwang Kim, *Propensity score estimation using the density ratio model under item nonresponse*;

Grażyna Trzpiot, *Longevity risk versus longevity dividend*;

Elżbieta Gołata, *Changes in demographic structures – consequences for the labour market in Central and Eastern Europe*;

Marcin Szymkowiak, Mirosław Krzyśko, Waldemar Wołyński, *Using functional distance correlation and functional Hilbert-Schmidt correlation to select features for cluster analysis of the labour market in Poland*;

Andrzej Sokołowski, Małgorzata Markowska, *Measure of cluster stability in dynamic classification*;

Dorota Rozmus, *Stability studies in an aggregated approach in taxonomy*;

Cyprian Kozyra, *Problems with exploratory and confirmatory factor analysis*;

Lilia Karpinska, Sławomir Śmiech, *Mapping regional vulnerability to energy poverty in Poland*;

Dwijendra Nath Dwivedi, Katarzyna Wójcik, Anilkumar Guntipalli Vemareddy, *Identification of key concerns and sentiments towards data quality and data strategy challenges using Sentiment Analysis and Topic Modelling*;

Mateusz Marszałkowski, Maciej Beręsewicz, *Extending the Demand for Labour survey by administrative data using data integration methods*;

Marek Walesiak, Grażyna Dehnel, Andrzej Dudek, *Dynamic approach in relative taxonomy and robust measures of central tendency*;

Marta Kusterka-Jefmańska, Bartłomiej Jefmański, Ewa Roszkowska, *Application of the Intuitionistic Fuzzy Synthetic Measure in subjective quality of life measurement based on aggregated data*;

Marcin Pełka, *Outlier identification for symbolic data with the application of the DBSCAN algorithm*;

Victor Shevchuk, *Determinants of the real estate prices in Poland*;

Anna Gdakowicz, Ewa Putek-Szeląg, *Attributes affecting exposure time of a residential property*;

Radosław Trojanek, *Are transaction prices the best source of information for constructing apartment price indexes on the primary market?*;

Iwona Markowicz, Beata Bieszk-Stolorz, *Changes in the share prices of macro-sector companies on the Warsaw Stock Exchange in response to the COVID-19 pandemic*;

Barbara Batóg, Katarzyna Wawrzyniak, *Comparison of the results of linear ordering of companies listed on the Warsaw Stock Exchange by means of different propositions of transformations of nominants into stimulants*;

Dominik Krężołek, *The impact of the COVID-19 pandemic on extreme risk in the metals market – application of GARCH-type models with alpha-stable distributed error*;

Michał Pietrzak, Tomasz Józefowski, Tomasz Klimanek, Andrzej Młodak, *Optimisation of the selection of statistical disclosure control methods on the example of microdata from the Polish survey of accidents at work*;

Romana Głowicka-Wołoszyn, Andrzej Wołoszyn, *Spatial effects in regional inequality analysis of own income potential among Polish gminas*;

Radosław Murkowski, *Multidimensional analysis of the excessive number of deaths related to the COVID-19 pandemic in European countries by gender and age*;

Agata Majkowska, Kamila Migdał-Najman, Krzysztof Najman, Katarzyna Raca, *Graphic characters as Twitter age group identifiers*;

Beata Bieszk-Stolorz, Krzysztof Dmytrów, Sebastian Majewski, Wojciech Zbaraszewski, *Application of random forests in the study of the differences in the perception of the neighbourhood of national parks in the Pomerania Euroregion*;

Iwona Bąk, Katarzyna Cheba, *Fuzzy cognitive maps as a tool for structuring new research problems*;

Izabela Michalska-Dudek, Andrzej Dudek, *Deep neural network model for predicting ROPO (Research Online, Purchase Offline) behaviour of tourists*;

Michel Voss, Maciej Beręsewicz, *Detection of solar panels using deep learning methods*;

Urszula Cieraszewska, Paweł Lula, Magdalena Talaga, *Successes and failures of scientific journals and their determinants*;

Urszula Cieraszewska, Anna Drabina, Ewelina Paluch, Janusz Tuchowski, *The co-authorship study among the Cracow University of Economics employees in years 2010–2020*;

Hanna Wdowicka, *Multilevel modelling of the foot placement control law proposed by Hof in different conditions of walk*

Ewelina Paluch, Janusz Tuchowski, Katarzyna Wójcik, Marcela Zembura, *Analysis of the importance of IT issues in scientific publications in the field of medicine in the years 2000–2020*;

Jan Kubacki, Alina Jędrzejczak, *Determinants and estimation of private farm income using methods of small area estimation and multivariate statistical analysis*;

Alina Szkop, *Estimation of variables from the DG-1 survey using time series analysis*;

Paweł Lańduch, *Estimation of selected variables in enterprise statistics using mass imputation*;

Ewa Kowalka, *'A picture is worth more than a thousand words' – graphical presentation of numerical data*;

Barbara Batóg, Jacek Batóg, *Classification of local administrative units in the period 2006–2018: a spatial approach*;

Agnieszka Stanimir, *Perception of climate change – differences between the Y and BB generation*;

Marcin Pełka, Andrzej Dudek, *Symbolic data analysis as a tool for credit fraud detection*;

Sylwia Filas-Przybył, Tomasz Klimanek, Jacek Kowalewski, *Employment-related commuting to provincial capital cities and their functional areas in 2016*;

Mirosława Kaczmarek, *Cash and non-cash payments for in-store purchases during the COVID-19 pandemic – similarities and differences between Generation X and Generation Y consumers*;

Aleksandra Łuczak, Sławomir Kalinowski, *Measuring subjective poverty: methodological and application aspects*;

Iwona Markowicz, Paweł Baran, *The use of the duration analysis in the study of the time of running the activity of Polish enterprises on the intra-Community market*;

Wioletta Grzenda, *Survival trees and direct adjusted survival curve – prediction in duration analysis*;

Marcin Salamaga, *Application of survival analysis methods in the study of foreign divestment in Poland*;

Dorota Żebrowska-Suchodolska, *Similarity of investment funds during the pandemic – stock and bond funds*;

Wojciech Roszka, *Statistical data integration in scientific productivity research*;

Maciej Beręsewicz, Dagmara Nikulin, Marcin Szymkowiak, Kamil Wilak, *COVID-19 and the gig economy in Poland*;

Ewa Łaźniewska, Tomasz Górecki, Klaudia Plac, *Regional labour markets as a result of the impact of the COVID-19 pandemic – a Polish-German borderland case study*;

Aleksandra Matuszewska-Janica, *The impact of the COVID-19 pandemic on the labour market in European Union countries*;

Marcin Szymkowiak, Kamil Wilak, Tomasz Józefowski, *Estimation of selected variables associated with the labour market in the light of non-response*;

Olga Kutera, *Fine wine in risk minimising portfolios.*

Members of the Classification and Data Analysis Section of the Polish Statistical Association (SKAD PTS) held their annual meeting on the first day of the conference. The meeting was chaired by Krzysztof Jajuga, PhD, DSc, ProfTit, and its agenda included the following items:

- report on SKAD activities;
- information about planned domestic and international conferences;
- organisation of SKAD conferences in 2022 and 2023;
- election of the representative of SKAD PTS in the IFCS Council for the 2022–2025 term;
- other issues.

The meeting was opened by Prof. Krzysztof Jajuga.

The report describing the activities undertaken by SKAD was presented by the Secretary of the SKAD Council, Barbara Pawełek, PhD, DSc, Assoc. Prof. at Cracow University of Economics. As reported, SKAD had at that moment 232 members (the section's by-laws and membership applications are available on the SKAD website).

Subsequently, Prof. Pawełek presented information about a book containing papers delivered during the previous SKAD conference (which was held in Sopot on 7–9 September 2020), adding that the report from that conference could be found in issue 3/2020 of the Statistical Review.

Prof. Barbara Pawełek introduced the details concerning conferences held in 2021: DSSV (Data Science, Statistics & Visualisation) – ECDA (the European Conference on Data Analysis), held online on 7–9 June; an IFCS online event – 7 September, featuring Grażyna Trzpiot, PhD, DSc, ProfTit from the University of Economics in Katowice as the keynote speaker; 13th Scientific Meeting of the

Classification and Data Analysis Group (ClaDAG), held online on 9–11 September (with two papers delivered by Polish authors).

The following conferences are scheduled to take place in late 2021 and in 2022: the 39th International Conference on Multivariate Statistical Analysis (MSA 2021), 8–10 November 2021 in Łódź; the 15th International Conference in memory of Professor Aleksander Zeliaś on Modelling and Forecasting Socio-economic Phenomena, 9–12 May 2022 in Zakopane; the 17th conference of the International Federation of Classification Societies (IFCS 2022), Porto, Portugal, 19–23 July 2022 (https://ifcs2022.fep.up.pt/).

The next point on the agenda concerned the organisation of future SKAD conferences. After a discussion, the participants agreed that a decision on this matter would be made later.

In the next part of the meeting the participants elected a representative of SKAD PTS in the IFCS Council for the 2022–2025 term. The election was presided over by Prof. Andrzej Dudek. Dr. Kamil Wilak was appointed the scrutineer. After Prof. Andrzej Dudek asked the participants of the videoconference to put forward their candidates and Prof. Marek Walesiak proposed Krzysztof Jajuga, who was the only candidate and agreed to stand as one. The scrutineer conducted a secret vote by means of the Doodle application. 23 members of SKAD PTS cast 23 valid votes, all in favour.

In the absence of any further points on the agenda, Prof. Krzysztof Jajuga closed the meeting.

On the last day of the conference, at the closing session, Prof. Marek Walesiak informed all the participants about the election of Krzysztof Jajuga as the representative of SKAD PTS in the IFCS Council and the possibility of having conference papers published in the Springer monograph. Prof. Krzysztof Jajuga thanked the conference organisers and all the participants and invited everyone to the next SKAD conference.