Combining Simheuristics with Petri Nets for Solving the Stochastic Vehicle Routing Problem with Correlated Demands

Juan I. Latorre-Biel[1], Daniele Ferone[2], Angel A. Juan[3,*], Javier Faulin[1]

[1] *Institute of Smart Cities, Public University of Navarre, Spain*
[2] *Dept. of Mechanical, Energetic and Management Eng., University of Calabria, Italy*
[3] *IN3 – Computer Science Dept., Universitat Oberta de Catalunya, Barcelona, Spain*

Emails: juanignacio.latorre@unavarra.es, danieleferone@gmail.com, ajuanp@uoc.edu, javier.faulin@unavarra.es

Abstract

This paper analyzes a stochastic version of the vehicle routing problem in which customers' demands are not only stochastic but also correlated. In order to solve this stochastic and correlated optimization problem, a simheuristic approach is combined with an adaptive demand predictor. This predictor is based on the use of machine learning methods and Petri nets. The information on real demands, provided by the vehicles as they visit the nodes of the logistic network, allows for a real-time forecast of the demand, as well as for an updated estimate of the correlation between them. A constrained prediction is provided by our hybrid algorithm, which is able to forecast an increase of 50% in the mean value of the demands of all nodes. With a very limited amount of information and reduced computational requirements, our algorithm provides a forecast with a high degree of reliability and a balanced capacity to reject false positives as well as false negatives. To illustrate its effectiveness, the methodology is applied to a wide range of benchmarks. The results show the benefits of applying this methodology in a context of correlated variation of the demands.

1. Introduction

Vehicle routing problems (VRP) are very popular in logistics, since they constitute simplified models of real-world problems found in a wide range of application fields, from long-distance backhaul planning (Belloso et al., 2019) to home healthcare logistics (Fikar et al., 2016). VRPs have received much attention from the research community during the last decades (Laporte, 2009). In their multiple variants, VRPs represent formidable challenges and, for this reason, significant research activity is currently devoted to obtain high-quality solutions using constrained computer resources and a limited time (Oyola et al., 2018). The statement of a VRP includes a depot, containing all available resources –such as products and vehicles–, and a number of nodes representing the customers' facilities –where delivery services may be requested. Several available homogeneous vehicles depart from the depot to deliver a particular product to the customers. A given customer can be served just by a single vehicle. The costs of the distribution process are usually proportional to the distance traveled by the vehicles. However, additional costs and penalties can be considered, such as those related to delivery time, the quality of service to the customers, or the number of required vehicles.

Different variants of the VRP aim at combining the description of relevant features from the real world with a limited level of complexity in the logistics model (Matei et al., 2015; Pop et al., 2013). One of these variants is the capacitated vehicle routing problem (CVRP), where the capacity of the delivery vehicles is constrained. Another variant is the vehicle routing problem with stochastic demands (VRPSD). In the VRPSD, a particularly realistic and challenging feature, randomness, is introduced in the model of the logistic system. In particular, a certain probability distribution represents the stochastic demands of customers, while the actual demand of a given node is only unveiled when the vehicle visits the customer. A methodology that can be applied to plan a set of routes for the vehicles consists of transforming the VRPSD into a VRP, by using one of the parameters of the probability distribution as deterministic demand –usually the mean or expected value of each random demand. A solution, valid for this deterministic VRP, can be taken as a possible solution for the VRPSD. The capacitated vehicle routing problem with stochastic demands (CVRPSD) is a variant of the VRP whose purpose is to find a set of routes for a fleet of homogeneous vehicles, with constrained capacity, to satisfy the stochastic demands of the customers (Marinaki and Marinakis, 2016). Other variants also consider stochastic travel and servicing times (Miranda and Conceição, 2016). The uncertainty associated with the demands of the customers may prevent a route to be completed in case that a given vehicle runs out of products before serving the last customer in the route. This event is called a 'route failure', and requires the implementation of a certain corrective action, named recourse or recovery operation (Figure 1). The vehicle involved in a route failure may return to the depot, so it can be reloaded with products and resume the route at the node where the delivery was interrupted (Hernandez et al., 2019). This recovery process, called detour to depot, increases the costs associated with the solution in an amount that can be given by the distance traveled in the round trip to the depot plus a certain penalty, which can be added to the mentioned cost. Actually, in some cases the structure of the 'penalty costs' associated with a route failure might cause the objective function to become a non-smooth one, as discussed in De Armas et al. (2018).

Other approaches can also be found. For example, a particular variant of the VRP, the chance-constrained VRP (CCVRP) does not specify the recourse actions in case the capacity of a vehicle is exceeded, but it is required that these actions are produced with a low probability. This approach supports certain benefits, such as a more consistent service and a reduced need

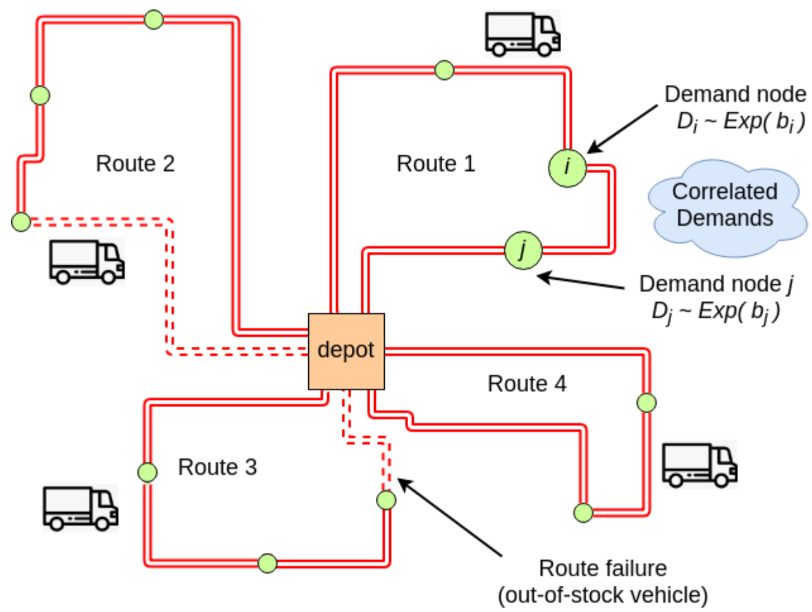


Figure 1: An illustrative example of the problem considered in this study.

for complex and expensive recourse actions (Dinh et al., 2018). Usual assumptions on the stochastic demands of the customers in a VRPSD are their association with a specific probability distribution and their independence (i.e., they are assumed to be non-correlated demands). Some references, including the present research paper, deal with approaches not constraining the types of these probability distributions. Additionally, the research described in this paper considers correlated demands, which may present a common trend –such as a joint increase of their mean values. This behavior is very useful to represent certain features of real-world logistic systems, where demands of different customers may be correlated (Spliet et al., 2014). Common patterns in the behavior of different customers of the same product can usually be found (Shi et al., 2016). Depending on the product to be delivered, diverse external factors, such as weather, festivities / holidays, crisis, fashion, price policies, rumors, panic, euphoria, imitation, shared information, etc., can lead to a correlated variation of the customers' demands. Anticipating any of these situations and predicting a potential correlation in the consumer demands may produce useful information for planning efficiently the routes in a CVRPSD. Some examples of systems that may experience this behavior are courier mail services to deliver and pick up mail or packages, distribution of heating oil, and online dial-a-ride transportation systems. If not predicted, a significant correlated increase in the demand values of the nodes in a CVRPSD may lead to distribution policies where the number of routes per solution is small and, hence, the cost of an *aprioristic* or *planned solution* is reduced. Nevertheless, it is likely for the demands to be larger than expected. That is why it is reasonable to assume that the number of route failures will rise. If the cost of a round trip to the depot or alternative recourse action is relatively high, the solutions obtained without prediction of the correlated variation of the demands are likely to be worse than if a forecast tries to predict this variation. As a consequence, the prediction of a potential correlation in the demands is a very promising area of research to generate low-cost and reliable solutions – i.e., solutions with a small number of route failures.

The rest of the paper is organized as follows. Section 2 makes an overview of the main characteristics of the proposed solving methodology. Section 3 deals with the basic notation and assumptions in the problem that has been investigated. Section 4 discusses relevant scientific literature related to the CVRPSD and the proposed methodology, such as simheuristics, demand prediction, demand correlation, as well as machine learning and Petri nets in combination with simheuristics. Section 5 focuses on the fundamentals of the proposed methodology, while Section 6 details the structure, behavior, integration, and parameters of a Petri net predictor. Section 7 describes a numerical example, while Section 8 discusses the subsequent results. Section 9 is devoted to the conclusions and future work.

2. Methodological Approach Overview

As introduced in the previous section, considering VRPS with stochastic and correlated customers' demands is the main target of this work. Providing a solving approach for that VRP variant constitutes a novel contribution. The methodology proposed to solve the CVRPSD with correlated demands combines simheuristics (Juan et al., 2018) and a demand predictor. The latter is developed as a discrete-event system, and modeled using the paradigm of Petri nets (Reisig, 2012). Simheuristics presents interesting advantages, such as: simplicity, efficiency, flexibility, a reduced parameter-setting stage, and the relaxation of most of the mentioned assumptions (Rabe et al., 2020). In particular, since simheuristics are based on simulation, they can cope with any probability distribution employed to model the random customers' demands. Moreover, our approach will not require these demands to be independent. Additionally, the use of Petri nets to model the predictor provides the following properties to our methodology: relative simplicity, clarity, reliability, and knowledge about the state of the predictor (David and Alla, 2005). Another feature of Petri nets is their flexibility. As a consequence, an extended model with a higher functionality can be easily developed to implement a predictor with a more complex behavior than the simplified one presented in this research paper. Moreover, the forecast is developed thanks to the simulation of the Petri net, which consumes a negligible amount of computer resources.

Hence, the main original contributions of this paper can been summarized as follows: *(i)* it proposes a predictor, based on a discrete-event system, for shared trends in correlated demands of the CVRPSD –this predictor will be able to represent the system structure and state; *(ii)* it designs a structurally simple and efficient Petri net, which is then integrated within a simheuristic framework to develop a learning methodology that allows for improving the quality of the solutions found in the presence of correlated demands; *(iii)* it proposes a methodology that uses the data provided by the vehicles –as they visit the different nodes–, to develop a continuously updated prediction on the demands of the remaining nodes in a route; *(iv)* it studies the feasible types of VRP and their circumstances, where the demand prediction can be applied successfully to improve the quality of the *aprioristic* solutions; and *(v)* it tests the proposed methodology into a large set of CVRPSD benchmarks, and analyze the impact of the Petri net predictor in the quality of the results.

3. Basic Notation and Assumptions

In the basic version of the CVRPSD, a typical instance i of the problem contains a set of $n_i + 1$ nodes (numbered from 0 to n_i). All the vehicles depart from the depot, node 0, to deliver products to the remaining nodes, which represent customers. In the traditional version of the problem, customers' demands are deterministic and known beforehand. In the stochastic version, customers' demands can follow different random variables, and their specific value for a given customer is only revealed when the vehicle visits the customer.

The basic CVRPSD constitutes a simplified version of a real-life problem in which customers' demands might share a common trend. Therefore, these demands might not be independent but, instead, correlated. Our methodology provides a forecasting procedure that allows to compute efficient solutions in the case of correlated demands. In particular, two different scenarios are allowed. It is assumed that the demands of the nodes can either be: *(i)* uncorrelated –their value is obtained from a certain probability distribution, whose mean value is previously known; or *(ii)* correlated, in which case we assume an increase of 50% in the mean value of the probability distribution modeling the demand of each node. The specific scenario associated with a certain instance of the CVRPSD is not known beforehand. The forecasting system is in charge of predicting which scenario is active from the information of the real demands given by the vehicles as they visit the customers.

The statement of the CVRPSD considered in this paper unveils the real demands of the nodes once the delivery vehicle visits the customer. Before this visit, only their probability distributions and their mean values are known, as well as the real demands of the already visited nodes. The proposed methodology uses the information gathered during the time span of the instance. A forecasting mechanism to predict shared trends in customers' demands has been implemented in the code developed to solve the CVRPSD with correlated demands. It is based on a discrete-event system, applied for collecting information from the customers visited by the delivery vehicles. This information is used to forecast the trend followed by the demands of the remaining customers left to be visited. As long as each vehicle visits a node, the real demand is known. This information feeds the forecasting system, and the prediction is updated. The discrete-event system is modeled by a Petri net, and its state is updated every time a vehicle visits a customer. The information provided by the Petri net is used to forecast the expected demands of the remaining nodes to be visited.

The mechanism for forecasting the demand correlation can be applied to find a high-quality solution for different variants of the VRP, where the data obtained during the route execution could be applied in a subsequent route-planning stage. Some of these variants are the dynamic VRP (Ritzinger et al., 2015; Spliet et al., 2014), multi-trip VPR (Cattaruzza et al., 2016), VRP with time windows (Bräysy and Gendreau, 2005), as well as the periodic VRP (Campbell and Wilson, 2013).

4. Modular Components of our Approach

The proposed methodology is based in three fundamental topics: simheuristics –as a methodology to solve the CVRPSD–, machine learning –for developing an adaptive predictor of the demand–, and the paradigm of the Petri nets –as a formal language to describe the predictor.

4.1. Simheuristics Applied to VRPs

Simheuristic algorithms integrate simulation and metaheuristics in order to find good quality solutions for a problem in a short time (Rabe et al., 2020). They are specially designed to solve a wide range of combinatorial optimization problems that take into account real-life complexity and uncertainty (Pagès-Bernaus et al., 2019; Quintero-Araujo et al., 2019). In that sense, simheuristics can be seen as a natural extension of the metaheuristics concept (Ferone et al., 2019). In particular, the combination of Monte Carlo simulation (MCS) and metaheuristics has been successfully applied to obtain near-optimal solutions for different VRPSD variants. MSC allows determining the total cost of any planned solution, including the cost of the route failures and their subsequent recourse actions, as well as its reliability (Faulin et al., 2008). As a result, promising solutions can be provided, with the associated cost and reliability, for supporting the decision-making process. Hence, Gruler et al. (2017a,b) report successful approaches, based on simheuristics, to solve the route planning in waste collection management under uncertainty scenarios. Similarly, Reyes-Rubiano et al. (2019) tackles the routing of electric vehicles with limited driving ranges and stochastic travel times, Calvet et al. (2019) address the multi-depot vehicle routing problem with stochastic demands, while Gruler et al. (2018, 2020) propose simheuristic approaches for single- and multi-period inventory routing problems with stochastic demands.

4.2. Demand Correlation in VRP

Stochastic VRPs (SVRP) contain stochastic parameters, associated with certain probability distributions. VRPs with stochastic demands (VRPSD) is a particular case of SVRP, where the random variables are related to the demands of the customers (Hernandez et al., 2019). Additionally, dynamic and stochastic VRPs envisage the use of real-time information, acquired after the creation of a planned solution, to update the routes to a new context (Ritzinger et al., 2015). In the VRPSD literature, it is usual to assume the demands of the customers to be independent (Oyola et al., 2018). However, this feature is not always found in real-world distribution problems. On the contrary, in many practical cases it is possible to find correlations between the demands of the nodes. For example, Chiang (2007) presents the correlated VRP, where a correlation between the periodic demands of the nodes is considered in the solution of the problem. The consideration of this correlation contributes to the prediction of differences between the real aggregated demands of the planned routes and the capacity of the vehicles. Shi et al. (2016) consider a VRP with potential demands, soft time windows (VRP-PDTW), and split delivery (several vehicles can serve the same customer). In the analyzed scenario, it is considered probable that once a customer knows the initial demand of other customers, this customer could make some adjustments to his or her own demand. The potential demand of a customer j is computed as a function of the difference between the initial demand of customer j and the ones of the remaining customers. This function can be linear, logarithmic, or semi-logarithmic, according to the authors. However, in most of the reported experimental tests, the authors generate the potential demands randomly. Dinh et al. (2018) describe a methodology to solve the chance-constrained VRP, which is a VRPSD with a limited probability of exceeding the capacity of every vehicle. In their approach, correlations between random demands are allowed, although the probability distribution of the demands is unknown.

4.3. Demand Prediction in VRPs

It is not always possible to know in advance the effects of correlations in the variation of the real demands. For this reason, demand forecasting may be considered as a particularly critical source of information to produce high-quality planned solutions for the VRPSD, where there is a correlation in the stochastic demands. In the CVRPSD with correlated demands, some information might not be known during the construction stage of an *aprioristic* or *planned solution*, despite this information can significantly influence the total cost (Ritzinger et al., 2015). Among the updated information that could be useful to plan the routes, we could consider travel times, service times, new or canceled customer requests, as well as customers' demands (Zou and Dessouky, 2018). However, this information is not always available at the time, when the *aprioristic* or *planned solution* is constructed. As a consequence, it is of great interest to forecast the values of the unknown parameters of relevance to the problem. In the literature, it is common to use historical data (Ehmke et al., 2012). Hence, Markov et al. (2016) present a methodology to solve a rich routing problem for collecting recyclable waste, where a daily demand is predicted by the statistical process of historical data of waste level in containers, which was obtained via ultrasonic sensors. Zou and Dessouky (2018) propose a look-ahead dynamic partial routing for the VRP with dynamic customer requests. In particular, it uses historical data to predict if some dynamic customers will request a delivery service once the planning stage has finished. Ge et al. (2018) develops a methodology to solve the two-echelon VRP – a VRP with intermediary facilities to transship the freight among different vehicles. This methodology uses historical distribution data of logistics companies, gathered from multiple delivery cycles, to forecast the demand of the customers. The historical demand data of each customer corresponds to a period of 30 days. Chiariotti et al. (2018) describe a methodology for the dynamic re-balancing of a bike-sharing system, which is tested using the data of the New York city system. In order to improve the estimation of the demand patterns, the authors propose to consider not only historical data but also current trends and weather data.

4.4. Petri Nets Applied to VRPs

Petri nets constitute a mathematical formalism especially suited to model and analyze discrete event systems, which may show complex behavior, such as concurrency and synchronization. A discrete-event system is a discrete-state and event-driven system, i.e., a system whose states present discrete values and may change after the occurrence of an event (Silva, 2018). The graphical representation of a Petri net provides with an intuitive and self-documented specification that describes explicitly the state of the modeled system (Silva, 1993). An equivalent matrix-based representation is appropriate for computer simulation, where the application of simple rules allows to study the evolution of the Petri net in different scenarios. Petri nets count on a broad body of theoretical results, facilitating both, the structural and the performance analysis of a net. Structural analysis allows checking qualitative properties, such as liveness, deadlock-freeness, reversibility, and boundedness. Janssens et al. (2009) reports an application of Petri nets aimed at solving the routing and scheduling problems in scenarios with uncertain travel times, such as the vehicle routing problem with time windows. Latorre-Biel et al. (2016) propose a methodology to combine simheuristics with a Petri net model, applied to cope with instances of the CVRPSD. The routing problem in a smart city through the use of a colored Petri net model is addressed in Latorre-Biel et al. (2017) to develop a mesoscopic traffic simulator. Essani and Haider (2018) describe a methodology to solve the multiple traveling salesman problem through its transformation into a colored Petri net. Finally, a Petri net model to describe the logistics in a smart factory in the frame of the paradigm of Industry 4.0 is presented in Latorre-Biel et al. (2018).

5. Fundamentals of the Methodology

Typically, a simheuristic algorithm starts with the calculation of a solution for a deterministic version of the CVRPSD in the form of a CVRP. This deterministic solution distributes the nodes in routes, which are assigned to different delivery vehicles. The quality of one of these *aprioristic* solutions is then estimated by Monte Carlo simulation. Stochastic values for the demands of the customers are obtained, following a certain probability distribution, and the potential route failures are evaluated. As a result, an average value of the cost associated with the implementation of the planned solution can be calculated. A solution with high quality (low cost) can then be selected and implemented. The previously described approach has been modified to implement the forecast of a potential shared trend in the correlated demands of the customers. In this new approach, once a solution for the active instance of the CVRPSD has been chosen and its application starts, a Petri net for demand prediction is activated to receive the real demands of the nodes, as long as they are visited by the delivery vehicles. The state of the Petri net, as well as the correlation forecast, is updated every time that new data feeds the predictor. The application of this forecast to the calculation of a planned solution is carried out once the route execution stage of a solution to a previous instance of the CVRPSD has finished. In other words, the correlation forecast is used when all the customers have been served and its purpose is to calculate a solution for the next solving iteration. The correlation forecast provided by the Petri net is then used to update (if necessary) the mean values of the probability distributions associated with the nodes, which are used as deterministic values of the demands, when planning a new solution for the CVRPSD.

Notice that the amount of information used by the Petri net predictor is just n_i real numbers, where n_i is the number of visited nodes in the execution of the routes of an *aprioristic* solution of the i-th instance of a CVRPSD (excluded the depot, node number 0). The Petri net predictor is integrated in the traditional simheuristic methodology as described in the following steps:

1. Use a simheuristic algorithm for solving a CVRPSD instance, i.e.: *(i)* solve the deterministic problem using average values; *(ii)* use MCS to evaluate each new 'promising' solution in a stochastic environment; and *(iii)* repeat the steps above while the termination condition has not been met yet.
2. Apply the best solution to solve the active instance of the CVRPSD and feed the Petri net system to forecast the correlation among customers' demands.
3. If necessary, re-compute the mean value of the demands associated with the different nodes, using the updated forecast of the Petri net.
4. Repeat the entire process using the updated expected value for each demand.

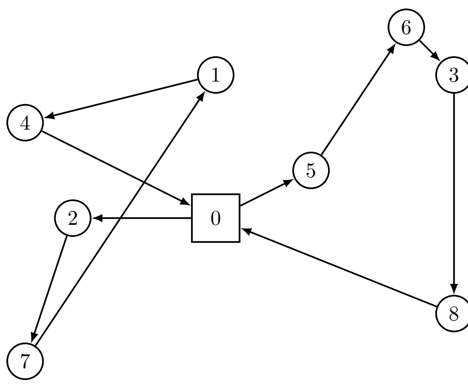
Algorithm 1 reports the pseudo-code of the approach. Moreover, Figure 2 reports an example of solutions obtained by the Simheuristic. Figure 2(a) is a solution obtained after the execution of the `BR-CWS` algorithm. The solution is not locally optimum and is improved by `localSearch`, obtaining the solution in Figure 2(b). The last step of the Simheuristic consists in the evaluation of the solution in the stochastic environment through simulation. A possible realization of the process of simulation can be found in Figure 2(c). In this case, the real demands along the routes exceeded the capacity of the vehicles, and the routes have been repaired with two additional trips to the depot for a vehicle reload. The results obtained by the simulation are used by the Petri net to adjust the predictions and obtain more accurate solutions in successive iterations.

6. The Petri Net Predictor

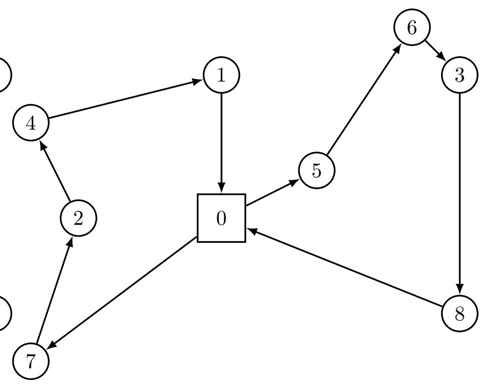
The Petri net that is used to forecast a potential correlation between the demands of the nodes in a CVRPSD is an interpreted ordinary Petri net composed by 5 places and 6 transitions (David and Alla, 2005; Silva, 1993). In our case, the Petri net is non-pure and n-bounded ($n = \mathbf{m}_0(p0)$, initial marking of place p_0 of the Petri net predictor). It is not live, since, after the firing of a certain number of transitions, a deadlock is always reached no matter what the initial marking is. There are four structural conflicts in the Petri net (involving the output transitions of p_0, p_1, p_2, and p_3). The interpretation of the Petri net prevents the structural conflict associated to p_0 from becoming an effective conflict, since the transitions involved in it,

Algorithm 1: Simheuristic and Petri net integration.

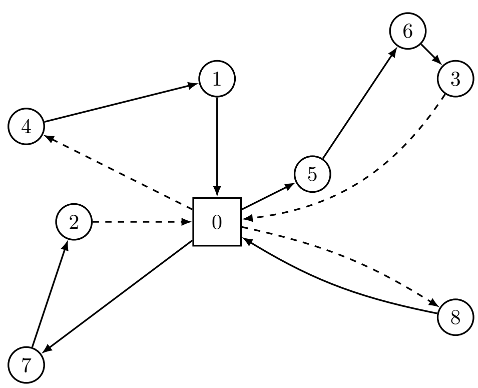
```
   // Solve the deterministic version of instance i using
      expected demands
1  initSol ← CWS(i);
   // Evaluate the solution in a stochastic environment
2  stochCost(initSol) ← simulation(initSol, sSim)
3  bestSol ← initSol;
4  while stopping criterion not met do
5  |   newSol ← BR-CWS(i);
6  |   newSol ← localSearch(newSol) ;                    // 2-Opt
7  |   stochCost(newSol) ← simulation(newSol, sSim) ;
8  |   if stochCost(newSol) < stochCost(bestSol) then
9  |   |   bestSol ← newSol
   // Petri net simulation
10 Apply bestSol to solve active instance i;
11 foreach j = 1, ..., n_i do
12 |   Classify node j into type 1, 2, or 3 ;    // see Section 6.2
13 |   Simulate the PN and qualitative forecast demand trend in
   |     remaining nodes;
14 Update expected demands;
15 while more time available repeat from 1
```



(a) Solution obtained after CWS.



(b) Solution obtained after local search.



(c) Solution after simulation.

Figure 2: Example of solution.

$\{t_0, t_1, t_2\}$, are synchronized with the occurrence of external events. These external events are the visit of a new node in the application of a solution for the CVRPSD and its subsequent classification. The mentioned classification is performed according to the comparison of the real demand with the mean of the probability distribution of the node. Regarding the type of node, the guard functions will allow the enabling of a single transition in the set $\{t_0, t_1, t_2\}$, assuming single server semantics: *(i)* C_0 is a Boolean guard function of transition t_0; it is active (true) after a vehicle has visited a certain node and its demand has been classified as showing a higher value than expected; *(ii)* C_1 is a Boolean guard function of transition t_1; it is active (true) after a vehicle has visited a certain node and its demand has been classified as meeting the expected value; and *(iii)* C_2 is a Boolean guard function of transition t_2; it is active (true) after a vehicle has visited a certain node and its demand has been classified as showing a lower value than expected. Once one of these transitions has been triggered, the associated guard function is deactivated (false). The Petri net then evolves until a deadlock is reached, before a new external enabling is produced. As a consequence of the mentioned interpretation of the Petri net, the other structural conflicts never become effective, since it is not possible for more than one of the transitions involved in each structural conflict to be enabled. Figure 3 depicts the Petri net in its initial marking.

6.1. Description of the Places and Transitions of the Petri Net

Figure 3 summarizes the main elements of our Petri net, whose details are presented next:

- $\mathbf{p_0}$ represents the nodes to visit;
- $\mathbf{p_1}$ and $\mathbf{p_2}$ represent, respectively, the excess of visited nodes with a stochastic demand that is identified as higher or lower than expected (regarding a certain window, quantified in a parameter of the Petri net, w, which needs to be tuned up);
- $\mathbf{p_3}$ allows the detection of an excess of visited nodes with a stochastic demand that is identified as being different than expected, while $\mathbf{p_4}$ represents a forecast of correlated variation of demands;
- $\mathbf{t_0}$, $\mathbf{t_1}$, and $\mathbf{t_2}$ fire only if, after a node is visited, the stochastic demand is higher, equal, or lower than expected, respectively;

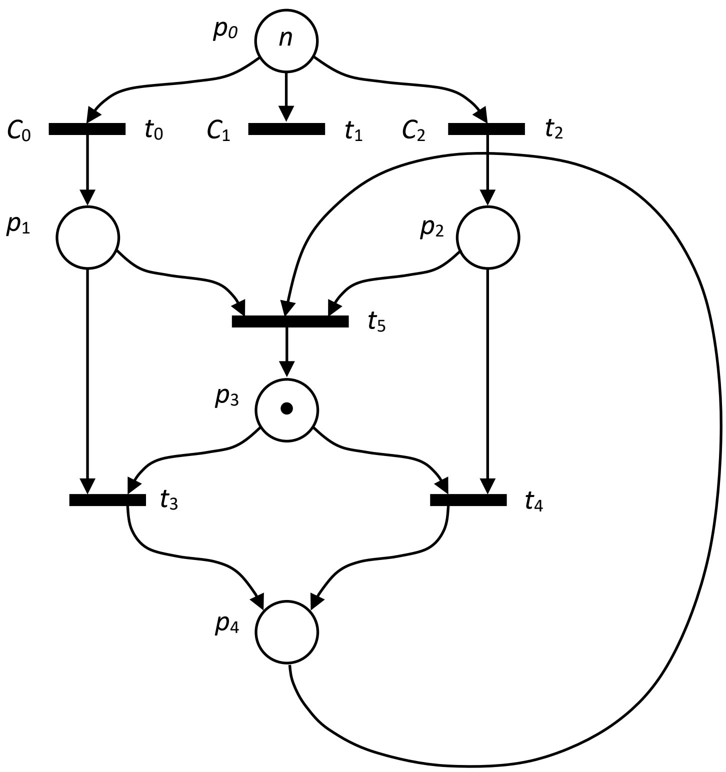


Figure 3: A graphical representation of the Petri net predictor.

- $\mathbf{t_3}$ and $\mathbf{t_4}$ fire only if a correlation is not predicted yet but there is an excess of nodes visited with higher or lower demand than expected, respectively;

- finally, $\mathbf{t_5}$ fires only if a correlation is already predicted and it is registered simultaneously an excess of both types of nodes (with higher and lower demand than expected); this transition removes the marking representing the smaller excess of one of both types of nodes by canceling a node of both types of nodes; after firing this transition there might be only one type of node in excess.

6.2. Operation of the Petri Net

The operation of the Petri net is performed in a series of steps, including a preliminary one required to furnish the forecast system with suitable information. This preliminary step makes use of the existing simheuristic algorithm. When the algorithm obtains the real demand of the j-th node (i.e. a delivery vehicle visits the node), this information is sent to the Petri net predictor so it can forecast a potential correlation in the real demands. This prediction might improve the estimation of the demands in the remaining nodes. Notice that $1 \leq j \leq n_i$, where n_i is the number of nodes to be visited (excluding the depot, node number 0).

1. The last visited node (node j) is classified by the Petri net into one of the following three types: *(i)* type 1, whenever the real demand of the node is higher than the expected demand; *(ii)* type 2, when the real demand of the node is exactly as expected; or *(iii)* type 3, whenever the real demand of the node is lower than the expected demand. This classification is applied by defining an interval around the mean of the probabilistic distribution that models the stochastic demand of the j-th node. The width of this interval depends on a parameter w. If the real demand of the j-th node falls inside this interval, this node would be considered as one of type 2. In case the real demand does not belong to this interval, the node would be of type 1 (higher-than-expected values) or 3 (lower-than-expected values).
2. In order to simulate the evolution of the Petri net, we need to provide a marking describing the initial state of the net. If the node is not the first one to be visited in the application of the *aprioristic* solution, this initial marking is the final marking from the simulation performed

after the previous visit of a node by any of the delivery vehicles. This requirement is based on the fact that the demand forecast is updated with the cumulative information provided by the nodes as they are visited. The simulation of the evolution of the Petri net starts when any of the three transitions $\{t_0, t_1, t_2\}$ is triggered. These transitions classify a new node into one of the three types mentioned in the previous step. This simulation finishes when there is not any enabled transition (deadlock). Once the evolution of the Petri net reaches a deadlock, it is possible to forecast the trend of the correlated demands of the nodes. In our methodology, the forecast is computed by comparing the number of visited nodes –with a detected change of trend– with a threshold, t. If this threshold is exceeded, then a shared increase in the correlated demands of the non-visited nodes is predicted. Otherwise, the forecast is that no change is expected. This threshold is a parameter of the forecast system that should be set. The simulation of the Petri net predictor requires two inputs: the real demand of a node and an initial marking. Additionally, the simulation of this Petri net leads to two outputs: *(i)* the qualitative forecast of the trend followed by the correlated real demands of the remaining nodes; and *(ii)* the final marking of the simulation of the Petri net –before the following simulation sequence– as a consequence of visiting another node.

3. Return to the initial step until all nodes have been already visited.
4. Compute the updated value of the expected demands on the nodes to be visited as a consequence of the qualitative forecast of the Petri net simulation. If there is a prediction of a real demand increase, this step is implemented by modifying accordingly one of the statistic parameters of the probability distribution associated with the real demands. For example, the average value can be used for this purpose.
5. Once all the nodes have been visited, an *aprioristic* solution can be carried out. For this purpose, the updated values of the expected demands in the non-visited nodes can be used. This step might imply a change in the delivery strategy applied so far.

6.3. Parameters of the Petri Net

As already mentioned, the Petri net predictor makes use of two parameters. The first one is the *interval around the expected values of the customers' demands*. Hence, once a node is visited and its real demand is revealed, this value is compared to the expected one. If the real demand falls inside this interval, we assume that the demand of the present node has not changed. On the contrary, if the real demand falls outside the interval, we assume that there is a variation in the customer's demand. This interval is quantified by a parameter $w \in [0, 1]$ as follows: let us consider $E[X]$ as the mean of a probability distribution and D_i the stochastic demand of the i-th node. Then, w defines an interval $[E[X] \cdot (1 - w), E[X] \cdot (1 + w)]$. The second parameter is the *overall excess of one of the three types of nodes among the visited ones*, t. This potential excess leads the Petri net to forecast a certain trend in the correlated demands. Let us consider that $n_i + 1$ is the number of nodes in the i-th instance of a CVRPSD with m sequential solving iterations. Also, let us assume that the current marking of place p_4 is $\mathbf{m}(p_4)$, being $\mathbf{m}_0(p_0) = n_i$ the initial marking of p_0. Then, we consider a threshold given by a real number $t\ in[0, 1]$. Now, whenever $m(p_4) \geq (i/m) \cdot t \cdot \mathbf{m}_0(p_0)$ we can assume that there is a correlated increase in the demands of the nodes. Under the assumption of correlation, a quality parameter, $Q_1(w, t)$, is computed as an average of all instances tested: $Q_1(w, t) = \frac{R11}{R11+R10}$, where $R11$ denotes the number of correct predictions (i.e., the correlation value is 1 and the predicted response is 1), and $R10$ is the number of incorrect ones (i.e., the correlation value is 1 and the predicted response is 0). Likewise, under the assumption of no correlation, a quality parameter $Q_2(w, t)$ is computed as an average of all instances tested: $Q_2(w, t) = \frac{R00}{R00+R01}$, where $R00$ denotes the number of correct predictions and $R01$ is the number of incorrect ones. An average value of both quality parameters, $Q_1(w, t)$ and $Q_2(w, t)$, is obtained as follows: $Q_T(w, t) = \frac{Q_1(w,t)+Q_2(w,t)}{2}$. Finally, an additional parameter $Q_B(w, t)$ is also defined to quantify the balance of successful prediction rates –both when a correlated increase in the demands exists (Q_1) and when it does not (Q_2). Specifically, it is computed as: $Q_B(w, t) = \frac{1}{|Q_1(w,t)-Q_2(w,t)|}$.

6.4. Example of Application

In order to illustrate the proposed methodology, an example based on instance *A-n32-k5* (available from https://bit.ly/3eGxGx9) is considered. In this example, there is one depot and 31 additional nodes with demands, whose mean values are known beforehand and constrained to the interval $[1, 24]$. Additionally, there are 6 delivery vehicles, each of them with a capacity of 100 units. An initial solution, composed of 6 routes, is computed using a simheuristic algorithm. In this solution, the first route is defined by the sequence of nodes $(24, 21, 22, 5, 16)$. Node 24 has an average demand of 24, while the real demand –unveiled once the vehicle reaches the node–, takes the value of 49.37. This information is sent to the Petri net predictor by means of the guard functions of transitions t_0, t_1, and t_2, which classify the node by comparing the real demand with a window, w, around the mean value of the demand. Let us consider $w = 0.4$. Then, an upper limit of this window, given by the value $mean(1{+}\mathrm{w}) = 24(1 + 0.4) = 33.6$, is compared to the value 49.37 (real demand). Since the latter is higher than the former, the firing of transition t_0 increases the marking of place p_1, which represents the number of nodes with a demand above the window.

The next two nodes present a real demand of 30.5 and 13, respectively. However, their expected demands are 12 and 4, respectively. Hence, both nodes increase the marking of p_1. However, the fourth node, number 5, presents a real demand of 1.38, and an average demand of 7. In this case $mean(1{-}\mathrm{w}) = 7(1 - 0.4) = 4.2$, which is the lower limit of the window around the mean value, is higher than the real demand. As a consequence, the Petri net increases the counter of nodes with a demand below the window by firing transition t_2 and, as a consequence, increasing the marking of place p_2. At this point, the Petri net removes a token from p_1 and another one from p_2, since t_5 is enabled . In this predictor, the pairs of nodes –one with a real demand above the window and another below it–, do not have any influence in the prediction. Only the excess of one type of nodes may influence the prediction itself by firing either t_3 or t_4 to add a token to p_4. Once all the nodes have been visited and the real demands are known, the Petri net generates a prediction by taking into account the marking of p_1 or p_2, representing the excess of nodes with a demand higher or lower than the expected one. The final value of p_1 is $m_f(p1) = 8$. A threshold $t = 0.2$ is defined to compute $0.2 \cdot 32 = 6.4$, the minimal value in p_1 and p_2 required to predict a correlated variation in the demands. Notice that 32 is the number of nodes in the network. In this example $m_f(p1) = 8 > 6.4$. Therefore, the Petri net predicts a correlated increase in the demand values. This prediction can be used to compute a new solution to the CVRPSD for subsequent deliveries of products to the nodes.

7. Numerical Experiments

In order to test the proposed methodology and evaluate its performance, a set of numerical experiments have been carried out. Following the approach employed in Gonzalez-Martin et al. (2018) for the arc routing problem, the classical VRP benchmarks have been extended into stochastic ones by using random demands instead of the deterministic ones. In addition, expected values of these demands have been increased by 50% to represent the cases in which a correlated increase of the real demand is produced. In particular, 53 classical VRP instances have been transformed into VRPSD instances by changing the deterministic demands into random demands following an exponential probability distribution. The entire dataset can be found at `https://bit.ly/3eGxGx9`. The expected values of these random demands are given by the original deterministic demands. Thus, for every instance an (*aprioristic*) and deterministic solution has been obtained by using the VRP algorithm proposed in Quintero-Araujo et al. (2017). Next, this solution has been run in a simulation environment to generate observations of the random demands, which were then used to feed the Petri net predictor. As a result, a forecast on the trend followed by the potentially correlated demands has been obtained. This forecast may predict either that the mean values of the demands remain constant or that they have been increased by a 50% percentage. Next, this forecast has been applied for computing a new deterministic solution. In summary, a simheuristics approach has been combined with the Petri net predictor to get a stochastic solution for each instance.

Every instance has been solved in four different scenarios. Scenarios 1 and 2 consider that the actual demands are not increased in a correlated way, while scenarios 3 and 4 assume that the real demands are increased by 50% in a correlated way with respect to their original values. While in scenarios 1 and 3 the applied methodology does not rely on the Petri net predictor, this predictor is employed in scenarios 2 and 4. Tables 1 and 2 show the results obtained after solving the instances in each of the aforementioned scenarios. The values included in these tables are described next: *(i) planned solution* is the cost of the deterministic solution, based on the knowledge of the position and deterministic demands of each node; *(ii) stochastic cost* refers to the additional cost generated by the extra trips to the depot after a route failure occurs; *(iii) total cost* is obtained by simply adding the two previous costs; *(iv) number of routes* refers to the number of independent sequences of nodes created in the planned solution, where each route is assigned to a different vehicle; *(v) route failures* refers to the number of round trips to the depot that have been completed by the vehicles for reloading – i.e., after having run out of products before finishing their routes; *(vi) forecast* refers to the prediction produced by the Petri net for a particular solution, where a single forecast is considered for each CVRPSD instance and scenario –if a correlation is predicted, it takes the value 1, being 0 otherwise; *(vii)* the *percentage of variation of the total cost*, ($\%tc$), compares the total costs of a planned solution when the Petri net predictor is applied and when it is not, i.e., $\%tc = \left(\frac{TC(PN)_i - TC_i}{TC_i}\right) \cdot 100$; and *(vii)* the *percentage of variation of the route failures*, ($\%\ rf$), compares the number of route failures of a planned solution when the Petri net predictor is applied and when it is not, i.e., $\%rf = \left(\frac{NRF(PN)_i - NRF_i}{NRF_i}\right) \cdot 100$.

8. Analysis of Results

Tables 1 and 2 include the main results obtained in our numerical tests. The type of forecast produced by the Petri net, as well as its resulting accuracy, are relevant to understand these results. The 'Forecast' column in Table 1 represents the prediction outcome of the Petri net. Table 1 corresponds to the scenarios where the nodes do not present a correlated increase of the demands. For this reason, a success in the prediction is represented by 0 in Table 1, while a value of 1 indicates a wrong prediction (false positive). In Table 1 there are 45 predictions that are correct and 8 that are wrong. The success rate is 84.9%, which corresponds to an expected value of 84%, achieved in the tuning process of the Petri net parameters, where $Q_T(w,t) = 0.84$. Table 2 refers to tests performed under the assumption that there is a correlated increase in the demands of the nodes. Hence, the 'Forecast' column in Table 2 shows the predictions of the Petri net, which can be right (1) or wrong (0 or false negative). There are 45 correct forecasts and 8 incorrect ones, which are the same figures shown in Table 1. Thus, the success rate is 84.9%. The detection of false positives (Table 1) and false negatives (Table 2) is balanced, since the forecast provided by the Petri net presents a similar success rate, both with and without correlation in the nodes' demands. This result could have been expected from the quality parameter $Q_B(w,t)$.

Tables 1 and 2 allow to compare the solutions obtained from the application of simheuristics. This version of simheuristics implements a particular Petri net predictor to forecast the stochastic demand of the nodes. In these instances, the behavior of the correlated demands, which can be predicted, is constrained to just two possibilities: *(i)* the mean value of their probability distributions are kept constant; or *(ii)* this value can be increased by a 50% percentage. Each row in Table 1 provides the solutions obtained in both cases, depending on whether the forecast of the Petri net predictor is applied

Table 1: Results for scenarios 1 and 2.

	Scenario 1 (no correlation. no prediction)					Scenario 2 (no correlation. Petri net prediction)							
Instance	Planned solution	Stochastic costs	Total costs	#routes	Route failures	Planned solution	Stochastic costs	Total costs	#routes	Route failures	Forecast	%tc	%rf
A-n32-k5	1822.70	263.20	2086.00	7	449	1817.80	280.40	2098.20	6	536	0	0.59	19.38
A-n33-k5	1547.30	212.70	1760.00	6	555	1513.60	195.50	1709.10	7	513	1	-2.89	-7.57
A-n33-k6	1572.50	314.20	1886.70	9	746	1612.00	226.30	1838.30	9	609	0	-2.56	-18.36
A-n37-k5	1640.60	221.80	1862.40	5	565	1709.30	154.70	1864.10	6	428	0	0.09	-24.25
A-n38-k5	1939.60	158.90	2098.60	8	486	1937.40	190.30	2127.70	8	550	0	1.39	13.17
A-n39-k6	2047.50	274.40	2321.90	8	764	2187.50	160.10	2347.60	9	539	1	1.11	-29.45
A-n45-k6	2644.60	257.50	2902.10	10	587	2570.70	298.40	2869.10	9	607	0	-1.14	3.41
A-n45-k7	2506.40	391.80	2898.20	10	608	2677.90	384.10	3062.00	11	708	0	5.65	16.45
A-n55-k9	2740.30	498.50	3238.80	10	1248	2830.10	494.80	3324.90	11	1204	0	2.66	-3.53
A-n60-k9	3498.70	591.40	4090.10	13	1133	3488.80	608.40	4097.30	12	1004	0	0.18	-11.39
A-n61-k9	2808.30	470.00	3278.30	11	1255	2930.90	398.80	3329.70	12	1130	0	1.57	-9.96
A-n63-k9	4186.00	770.40	4956.30	12	1149	4264.40	713.90	4978.30	13	1140	0	0.44	-0.78
A-n65-k9	3585.80	528.40	4114.20	11	1166	3671.60	486.20	4157.80	12	1064	0	1.06	-8.75
A-n80-k10	5348.40	753.80	6102.20	18	1120	5387.10	776.90	6164.00	16	1138	0	1.01	1.61
B-n31-k5	1453.10	221.80	1674.90	9	358	1464.60	239.20	1703.70	8	404	0	1.72	12.85
B-n35-k5	2135.20	500.90	2636.10	5	707	2225.40	465.00	2690.40	6	668	1	2.06	-5.52
B-n39-k5	2112.20	169.90	2282.20	8	506	2089.80	182.10	2271.80	8	555	0	-0.45	9.68
B-n41-k6	2423.40	259.70	2683.10	11	521	2499.90	232.80	2732.70	12	519	1	1.85	-0.38
B-n45-k5	2267.90	218.60	2486.50	7	532	2195.80	286.60	2482.40	6	668	1	-0.16	25.56
B-n50-k7	2385.20	321.90	2707.10	8	769	2451.30	345.00	2796.30	8	915	0	3.29	18.99
B-n52-k7	2725.20	252.10	2977.30	9	682	2701.50	264.70	2966.20	10	774	0	-0.37	13.49
B-n56-k7	2692.10	210.10	2902.20	12	658	2691.10	216.30	2907.40	12	681	0	0.18	3.50
B-n57-k9	3621.00	705.10	4326.10	12	916	3591.40	815.20	4406.60	12	1004	0	1.86	9.61
B-n64-k9	2953.00	441.10	3394.10	11	1248	2882.00	449.80	3331.80	11	1325	0	-1.84	6.17
B-n67-k10	3401.20	411.00	3812.20	19	1072	3435.70	480.20	3915.80	16	1176	0	2.72	9.70
B-n68-k9	4172.60	476.50	4649.20	16	803	4461.20	515.30	4976.50	18	815	0	7.04	1.49
B-n78-k10	4240.90	560.20	4801.00	16	1267	4335.60	522.20	4857.80	14	1159	0	1.18	-8.52
E-n22-k4	654.20	175.60	829.80	4	644	688.00	131.70	819.80	5	484	0	-1.21	-24.84
E-n30-k3	1373.60	143.60	1517.20	4	304	1354.40	152.40	1506.70	3	365	0	-0.69	20.07
E-n33-k4	1658.90	369.70	2028.60	5	492	1708.60	304.10	2012.70	6	393	0	-0.78	-20.12
E-n51-k5	1468.00	144.50	1612.40	6	657	1536.10	146.30	1682.40	6	613	0	4.34	-6.70
E-n76-k7	2368.40	178.10	2546.50	8	759	2397.10	230.80	2627.90	7	929	0	3.20	22.40
E-n76-k10	2557.50	272.10	2829.60	13	1361	2496.80	303.50	2800.30	12	1305	1	-1.04	-4.11
E-n76-k14	2570.10	510.20	3080.30	16	2160	2536.90	496.30	3033.20	16	2187	0	-1.53	1.25
F-n72-k4	1235.30	41.40	1276.70	9	325	1197.20	43.60	1240.90	8	344	0	-2.80	5.85
M-n101-k10	3946.50	320.10	4266.60	11	1164	3897.50	377.90	4275.40	11	1158	0	0.21	-0.52
M-n121-k7	8013.00	116.70	8129.70	27	182	8181.00	189.50	8370.50	30	421	0	2.96	131.32
P-n19-k2	387.70	34.10	421.80	3	130	390.60	43.60	434.10	3	180	0	2.93	38.46
P-n20-k2	421.90	34.20	456.10	3	170	424.30	39.50	463.80	3	161	0	1.69	-5.29
P-n22-k2	450.40	35.20	485.60	3	140	462.20	28.00	490.20	4	133	0	0.95	-5.00
P-n22-k8	853.00	316.30	1169.40	10	1079	861.80	341.20	1203.10	10	1133	0	2.88	5.00
P-n40-k5	1203.80	107.10	1311.00	6	561	1200.90	82.90	1283.80	6	464	1	-2.07	-17.29
P-n50-k8	1537.90	230.20	1768.10	10	1052	1515.00	269.60	1784.60	9	1282	0	0.93	21.86
P-n50-k10	1564.80	294.00	1858.80	11	1444	1584.80	297.30	1882.10	12	1354	0	1.26	-6.23
P-n51-k10	1656.70	347.90	2004.50	11	1607	1665.50	333.70	1999.20	12	1409	0	-0.26	-12.32
P-n55-k7	1591.00	198.50	1789.50	7	976	1629.70	162.80	1792.50	8	747	0	0.16	-23.46
P-n55-k15	1797.30	534.20	2331.50	18	2370	1822.20	554.20	2376.40	18	2400	0	1.93	1.27
P-n60-k10	1893.30	349.60	2242.90	11	1562	1881.60	306.40	2188.10	11	1465	0	-2.45	-6.21
P-n65-k10	2090.30	369.10	2459.40	11	1429	2091.80	275.20	2367.00	11	1392	0	-3.76	-2.59
P-n70-k10	2315.80	325.80	2641.60	12	1368	2280.60	351.20	2631.80	11	1507	0	-0.37	10.16
P-n76-k4	2391.90	49.30	2441.20	6	257	2331.10	93.70	2424.90	5	405	0	-0.67	57.59
P-n76-k5	2348.90	139.00	2487.90	6	543	2406.10	124.90	2531.00	7	571	1	1.73	5.16
P-n101-k4	3271.40	65.60	3337.00	5	294	3291.30	55.60	3346.90	5	271	0	0.30	-7.82

Table 2: Results for scenarios 3 and 4.

	Scenario 3 (correlation. no prediction)					Scenario 4 (correlation. Petri net prediction)							
Instance	Planned solution	Stochastic costs	Total costs	#routes	Route failures	Planned solution	Stochastic costs	Total costs	#routes	Route failures	Forecast	%tc	%rf
A-n32-k5	1918.60	735.50	2654.10	6	1219	1958.00	540.00	2498.00	8	896	1	-5.88	-26.50
A-n33-k5	1482.40	647.80	2130.20	6	1501	1623.30	429.70	2053.00	8	980	1	-3.62	-34.71
A-n33-k6	1659.80	570.90	2230.60	8	1621	1716.50	408.30	2124.80	11	1164	1	-4.74	-28.19
A-n37-k5	1771.00	445.70	2216.70	6	1285	1775.70	294.30	2070.10	8	832	1	-6.62	-35.25
A-n38-k5	1977.70	525.70	2503.30	9	1308	1849.90	427.40	2277.30	10	983	1	-9.03	-24.85
A-n39-k6	2024.50	652.00	2676.50	7	1610	2139.20	430.60	2569.80	10	1101	1	-3.98	-31.61
A-n45-k6	2628.00	869.30	3497.40	9	1885	2647.60	678.00	3325.60	11	1427	1	-4.91	-24.30
A-n45-k7	2724.70	1032.30	3757.00	12	1629	2802.70	762.10	3564.90	14	1319	1	-5.11	-19.03
A-n55-k9	2793.40	1147.00	3940.40	11	2685	3002.90	876.40	3879.30	15	2002	1	-1.55	-25.44
A-n60-k9	3549.90	1501.90	5051.70	14	2519	3707.60	1314.00	5021.60	15	2324	0	-0.60	-7.74
A-n61-k9	2975.50	1065.60	4041.10	12	2820	2874.00	1020.80	3894.80	12	2704	0	-3.62	-4.11
A-n63-k9	4420.40	1810.10	6230.50	14	2775	4269.30	1789.40	6058.70	14	2658	0	-2.76	-4.22
A-n65-k9	3664.70	1206.20	4870.90	14	2799	3800.30	861.00	4661.30	17	1901	1	-4.30	-32.08
A-n80-k10	5498.40	1883.80	7382.20	17	2852	5697.20	1209.70	6906.90	20	1879	1	-6.44	-34.12
B-n31-k5	1604.20	491.50	2095.70	10	921	1623.80	350.70	1974.60	12	563	0	-5.78	-38.87
B-n35-k5	2241.40	882.50	3124.00	6	1265	2335.60	745.30	3080.80	8	1012	1	-1.38	-20.00
B-n39-k5	2035.50	464.50	2500.00	8	1248	2098.50	327.10	2425.50	9	763	1	-2.98	-38.86
B-n41-k6	2499.00	695.50	3194.50	12	1405	2471.30	514.50	2985.80	13	1155	1	-6.53	-17.79
B-n45-k5	2209.80	627.10	2837.00	6	1669	2302.40	470.10	2772.50	9	1109	1	-2.27	-33.55
B-n50-k7	2342.10	881.30	3223.40	8	2180	2438.80	516.00	2954.80	12	1288	1	-8.33	-40.92
B-n52-k7	2741.10	731.30	3472.40	10	1842	2661.10	589.00	3250.00	11	1455	1	-6.40	-21.01
B-n56-k7	2797.00	591.80	3388.80	13	1705	2833.00	459.80	3292.80	14	1247	1	-2.83	-26.86
B-n57-k9	3914.40	1648.40	5562.80	16	1979	3968.20	1428.90	5397.00	16	1756	1	-2.98	-11.27
B-n64-k9	2830.00	1118.70	3948.70	10	3115	3172.50	675.40	3847.90	17	1957	1	-2.55	-37.17
B-n67-k10	3510.20	1015.70	4525.90	19	2579	3661.80	757.30	4419.10	22	1809	1	-2.36	-29.86
B-n68-k9	4580.50	1293.90	5874.50	19	2380	4682.90	943.80	5626.80	21	1604	1	-4.22	-32.61
B-n78-k10	4348.10	1279.10	5627.20	17	2879	4395.10	922.10	5317.30	19	1999	1	-5.51	-30.57
E-n22-k4	673.90	355.50	1029.50	4	1387	767.70	234.20	1001.90	7	896	1	-2.68	-35.40
E-n30-k3	1415.30	364.60	1779.90	3	944	1447.50	244.20	1691.70	5	578	1	-4.95	-38.77
E-n33-k4	1608.50	950.60	2559.10	5	1196	1835.30	582.70	2418.00	7	772	1	-5.51	-35.45
E-n51-k5	1520.30	347.60	1867.80	7	1538	1610.80	222.30	1833.10	9	1022	1	-1.86	-33.55
E-n76-k7	2411.30	513.80	2925.10	9	2145	2546.40	259.50	2805.80	13	1211	1	-4.08	-43.54
E-n76-k10	2511.30	771.00	3282.30	12	3480	2569.20	542.90	3112.10	17	2398	1	-5.19	-31.09
E-n76-k14	2638.90	1098.40	3737.30	16	4967	2598.80	1038.20	3637.10	16	4953	0	-2.68	-0.28
F-n72-k4	1232.60	138.20	1370.80	9	1061	1267.60	71.70	1339.30	10	556	1	-2.30	-47.60
M-n101-k10	4002.50	911.00	4913.60	12	2954	4063.90	576.20	4640.10	17	1896	1	-5.57	-35.82
M-n121-k7	8142.20	577.50	8719.70	33	1411	8378.10	240.60	8618.70	36	519	1	-1.16	-63.22
P-n19-k2	405.40	118.70	524.10	3	536	409.20	96.90	506.10	4	416	1	-3.45	-22.39
P-n20-k2	405.30	128.80	534.10	3	549	395.80	157.40	553.20	3	567	0	3.59	3.28
P-n22-k2	416.40	133.60	550.00	3	509	479.60	117.20	596.80	4	526	0	8.50	3.34
P-n22-k8	910.10	573.20	1483.30	10	2006	899.00	616.20	1515.30	9	2162	0	2.16	7.78
P-n40-k5	1221.00	258.00	1479.00	7	1236	1225.30	223.30	1448.60	8	978	1	-2.05	-20.87
P-n50-k8	1497.50	649.50	2146.90	9	2931	1583.60	417.50	2001.10	14	1872	1	-6.79	-36.13
P-n50-k10	1581.40	702.70	2284.10	12	3189	1686.50	499.30	2185.80	17	2339	1	-4.30	-26.65
P-n51-k10	1707.60	735.60	2443.20	12	3294	1820.40	553.90	2374.30	18	2313	1	-2.82	-29.78
P-n55-k7	1502.40	484.80	1987.20	7	2207	1670.40	297.00	1967.40	11	1406	1	-1.00	-36.29
P-n55-k15	1856.60	1057.30	2913.90	18	4943	2066.00	775.70	2841.70	28	3510	1	-2.48	-28.99
P-n60-k10	1870.30	797.00	2667.20	11	3344	1977.80	568.40	2546.20	16	2346	1	-4.54	-29.84
P-n65-k10	2099.50	767.80	2867.30	12	3210	2294.80	483.50	2778.20	18	1987	1	-3.11	-38.10
P-n70-k10	2321.70	808.00	3129.70	12	3381	2408.40	555.50	2963.90	17	2356	1	-5.30	-30.32
P-n76-k4	2406.90	223.20	2630.10	6	1002	2370.30	138.50	2508.80	8	549	1	-4.61	-45.21
P-n76-k5	2341.50	412.70	2754.20	6	1713	2432.00	256.10	2688.10	9	1046	1	-2.40	-38.94
P-n101-k4	3209.10	336.70	3545.80	5	1199	3216.40	170.40	3386.80	6	751	1	-4.48	-37.36

or not to the calculation of the *aprioristic* solution. Notice that there is not any instance where there is a shared increase in the correlated demands. The values in the rows, where the prediction of the Petri net is right (0), have no significance here, since the algorithm is exactly the same regardless of whether the Petri net predictor has been used or not. In these cases, the differences in the results are a consequence of the stochastic nature of the demands. Furthermore, the computational cost of implementing the Petri net prediction is negligible compared to the application of the simheuristic algorithm.

From Table 1, it can be concluded that the use of a Petri net predictor leads to worse results in a certain percentage of the false positives, where these false positives are in turn the 15.1% of the total instances solved. This result could have been expected, since false positives tend to lead to solutions with a higher number of routes. As a consequence, they are expected to present higher costs in the planned solutions, as well as a smaller number of route failures –which, in turn, leads to smaller stochastic costs. These slightly worse results from the use of a Petri net predictor are clearly compensated by the better results obtained when there is an increase in the correlated demands. Thus, the effectiveness of the application of a Petri net predictor is shown in Table 2, which contains solutions obtained in scenarios, where the nodes present a correlated increase in demands. Notice that the forecast of the Petri net presents a positive impact in the majority of the solutions, since a successful prediction would allow to find more realistic *aprioristic* solutions. In the same way, Table 2 contains some values lacking significance for the analysis aimed at this section. These values, 8 in total, correspond to wrong forecasts of the Petri net (false negatives or forecast $R10$), since the planned or *aprioristic* solutions are computed with the same knowledge on the demands of the nodes and the same solving methodology –regardless of whether a Petri net is employed or not. Furthermore, all those solutions in Table 2 whose forecast have been correct correspond to 45 out of the 53 instances. They also present smaller total stochastic costs and a smaller number of route failures. The improvement in the total cost ranges from 1% to 9%, depending on the instance, while the route failures are reduced between 11% and 63%. Figure 4 shows a comparison between the standard approach (base scenario, represented by the horizontal line $y = 0$) and the enriched approach with the Petri net prediction. Notice that, when no correlation exists, the average gap in total cost between both approaches is quite small (0.68%). In other words, the Petri net predictor offers no advantage over the traditional method. However, in a scenario with correlation this situation changes. Now, the average gap is negative and larger in absolute value (-3.52%), which shows how the use of our Petri net predictor can provide noticeable reductions in total cost when correlation exists. A similar effect can be observed for the route failures indicator: in absence of correlation, our Petri net predictor does not provide any noticeable advantage over the traditional method (average gap of 4.05%). However, when correlation exists, our approach contributes to significantly reduce the number of route failures (the average gap is now about -28%).

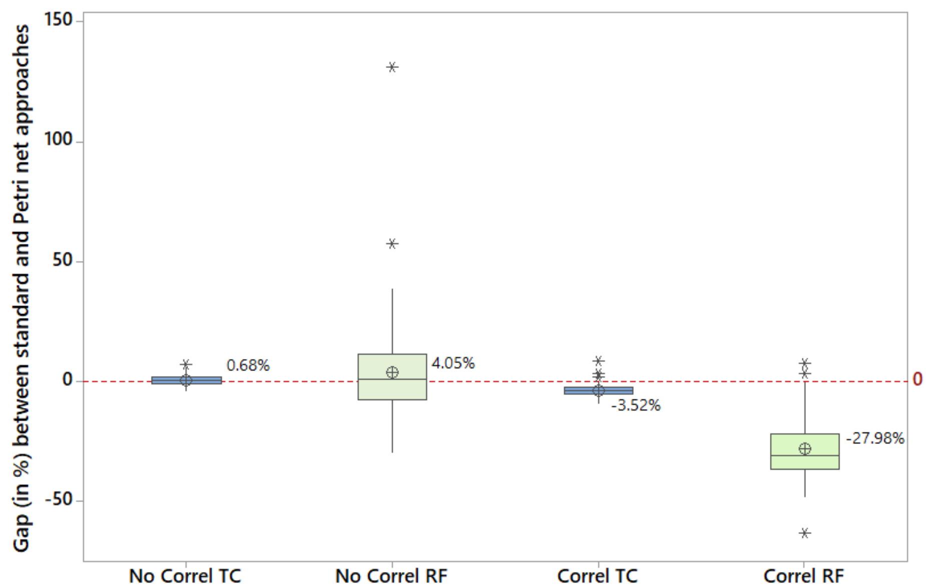


Figure 4: A comparison of the standard approach and the one using Petri nets.

9. Conclusions and Future Work

In real logistic networks, which inspire the definition of vehicle routing problems (VRP) in most of its variants, there are some features that are not usually considered in stochastic VRP models. One of these features is the potential correlations among customers' demands. As a consequence of these correlations, the real demands might present common variations. A Petri net predictor has been implemented to forecast the correlated behavior of the demands in a simple scenario, where the probability distributions representing the demands are constrained to two possibilities: their mean value can be constant or they can experience a correlated percentage increase of 50%. This simplified scenario facilitates the illustration of the structure and operation of this predictor, its application, and the interpretation of the results.

A set of instances have been solved in scenarios with correlated increase of the demands and without it, as well as with the application of a Petri net predictor and without it. The results of the tests lead to the following conclusions:

1. False positives present a reduced impact on the quality of the solutions since: *(i)* forecasts with false positives have an effect on just about 15% of the solutions –in case demands are not really correlated; *(ii)* not all forecasts with false positives have led to solutions with higher costs; and *(iii)* only half of the tests correspond to a correlated increase in demands –thus, the false positives affect about 7.5% of the total number of tests.
2. False negatives do not imply a reduction in the quality of the solutions, since their only effect is to lose the opportunity to improve the solution obtained using the Petri net forecast.
3. Under the correlated scenario, using our Petri net predictor provides better solutions in about 85% of the cases.

All in all, the Petri net predictor has been successfully employed to solve VRP instances with stochastic and correlated demands. In addition, the implementation of the Petri net predictor can be done without assuming a high computational effort. Moreover, this predictor only requires a limited amount of information, which facilitates its implementation in practice.

Several research lines still remain open for future works. Among them: *(i)* to develop an alternative solving approach, based on the combination of simulation with machine learning or time series analysis; *(ii)* to consider larger-size instances in order to investigate how adding more correlated nodes affects the performance of the Petri net predictor in comparison to other approaches; *(iii)* to consider scenarios with a wider range of possible values for the correlated demands, which might reduce the accuracy of some prediction methods; and *(iv)* to analyze an entire supply chain network with nodes that present correlated demands.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science, Innovation, and Universities (PID2019-111100RB-C21/C22, RED2018-102642-T) and the SEPIE Erasmus+ Program (2019-I-ES01-KA103-062602). We also want to acknowledge the support received from the CAN Foundation in Navarre, Spain (Grant ID 903 100010434 under the agreement LCF/PR/PR15/51100007).

References

Belloso, J., Juan, A.A., Faulin, J., 2019. An iterative biased-randomized heuristic for the fleet size and mix vehicle-routing problem with backhauls. International Transactions in Operational Research 26, 289–301.

Bräysy, O., Gendreau, M., 2005. Vehicle routing problem with time windows, part I: route construction and local search algorithms. Transportation Science 39, 104–118.

Calvet, L., Wang, D., Juan, A.A., Bové, L., 2019. Solving the multidepot vehicle routing problem with limited depot capacity and stochastic demands. International Transactions in Operational Research 26, 458–484.

Campbell, A.M., Wilson, J.H., 2013. Forty years of periodic vehicle routing. Networks 63, 2–15.

Cattaruzza, D., Absi, N., Feillet, D., 2016. Vehicle routing problems with multiple trips. 4OR 14, 223–259.

Chiang, C.P., 2007. The correlated vehicle routing problem, in: Proceedings of the 2007 International Conference on Wireless Communications, Networking and Mobile Computing, IEEE. pp. 3824–3828.

Chiariotti, F., Pielli, C., Zanella, A., Zorzi, M., 2018. A dynamic approach to rebalancing bike-sharing systems. Sensors 18, 512.

David, R., Alla, H., 2005. Discrete, continuous, and hybrid Petri nets. volume 1. Springer.

De Armas, J., Ferrer, A., Juan, A.A., Lalla-Ruiz, E., 2018. Modeling and solving the non-smooth arc routing problem with realistic soft constraints. Expert Systems With Applications 98, 205–220.

Dinh, T., Fukasawa, R., Luedtke, J., 2018. Exact algorithms for the chance-constrained vehicle routing problem. Mathematical Programming 172, 105–138.

Ehmke, J.F., Steinert, A., Mattfeld, D.C., 2012. Advanced routing for city logistics service providers based on time-dependent travel times. Journal of Computational Science 3, 193–205.

Essani, F., Haider, S., 2018. An algorithm for mapping the asymmetric multiple traveling salesman problem onto colored Petri nets. Algorithms 11, 143.

Faulin, J., Juan, A.A., Serrat, C., Bargueno, V., 2008. Predicting availability functions in time-dependent complex systems with saedes simulation algorithms. Reliability Engineering & System Safety 93, 1761–1771.

Ferone, D., Gruler, A., Festa, P., Juan, A.A., 2019. Enhancing and extending the classical grasp framework with biased randomisation and simulation. Journal of the Operational Research Society 70, 1362–1375.

Fikar, C., Juan, A.A., Martinez, E., Hirsch, P., 2016. A discrete-event driven metaheuristic for dynamic home service routing with synchronised trip sharing. European Journal of Industrial Engineering 10, 323–340.

Ge, X., Xue, G., Wen, P., 2018. Proactive two-level dynamic distribution routing optimization based on historical data. Mathematical Problems in Engineering 2018, 1–15.

Gonzalez-Martin, S., Juan, A.A., Riera, D., Elizondo, M.G., Ramos, J.J., 2018. A simheuristic algorithm for solving the arc routing problem with stochastic demands. Journal of Simulation 12, 53–66.

Gruler, A., Fikar, C., Juan, A.A., Hirsch, P., Contreras-Bolton, C., 2017a. Supporting multi-depot and stochastic waste collection management in clustered urban areas via simulation–optimization. Journal of simulation 11, 11–19.

Gruler, A., Panadero, J., de Armas, J., Moreno, J.A., Juan, A.A., 2018. Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs. Computers & Industrial Engineering 123, 278–288.

Gruler, A., Panadero, J., de Armas, J., Moreno, J.A., Juan, A.A., 2020. A variable neighborhood search simheuristic for the multiperiod inventory routing problem with stochastic demands. International Transactions in Operational Research 27, 314–335.

Gruler, A., Quintero-Araújo, C.L., Calvet, L., Juan, A.A., 2017b. Waste collection under uncertainty: a simheuristic based on variable neighbourhood search. European Journal of Industrial Engineering 11, 228–255.

Hernandez, F., Gendreau, M., Jabali, O., Rei, W., 2019. A local branching matheuristic for the multi-vehicle routing problem with stochastic demands. Journal of Heuristics 25, 215–245.

Janssens, G.K., Caris, A., Ramaekers, K., 2009. Time Petri nets as an evaluation tool for handling travel time uncertainty in vehicle routing solutions. Expert Systems with Applications 36, 5987–5991.

Juan, A.A., Kelton, W.D., Currie, C.S., Faulin, J., 2018. Simheuristics applications: dealing with uncertainty in logistics, transportation, and other supply chain areas, in: Proceedings of the 2018 Winter Simulation Conference, IEEE. pp. 3048–3059.

Laporte, G., 2009. Fifty years of vehicle routing. Transportation Science 43, 408–416.

Latorre-Biel, J.I., Faulin, J., Jiménez, E., Juan, A.A., 2017. Simulation model of traffic in smart cities for decision-making support: Case study in Tudela (Navarre, Spain), in: Proceedings of the 2017 International Conference on Smart Cities, Springer. pp. 144–153.

Latorre-Biel, J.I., Faulin, J., Juan, A.A., 2016. Enriching simheuristics with Petri net models: potential applications to logistics and supply chain management, in: Proceedings of the 2016 Winter Simulation Conference, IEEE Press. pp. 2475–2486.

Latorre-Biel, J.I., Faulin, J., Juan, A.A., Jiménez-Macías, E., 2018. Petri net model of a smart factory in the frame of industry 4.0. IFAC-PapersOnLine 51, 266–271.

Marinaki, M., Marinakis, Y., 2016. A glowworm swarm optimization algorithm for the vehicle routing problem with stochastic demands. Expert Systems with Applications 46, 145–163.

Markov, I., Bierlaire, M., Cordeau, J.F., Maknoon, Y., Varone, S., 2016. Inventory routing with non-stationary stochastic demands. URL: http://infoscience.epfl.ch/record/221364.

Matei, O., Pop, P.C., Sas, J.L., Chira, C., 2015. An improved immigration memetic algorithm for solving the heterogeneous fixed fleet vehicle routing problem. Neurocomputing 150, 58 – 66.

Miranda, D.M., Conceição, S.V., 2016. The vehicle routing problem with hard time windows and stochastic travel and service time. Expert Systems with Applications 64, 104–116.

Oyola, J., Arntzen, H., Woodruff, D.L., 2018. The stochastic vehicle routing problem, a literature review, part I: models. EURO Journal on Transportation and Logistics 7, 193–221.

Pagès-Bernaus, A., Ramalhinho, H., Juan, A.A., Calvet, L., 2019. Designing e-commerce supply chains: a stochastic facility–location approach. International Transactions in Operational Research 26, 507–528.

Pop, P.C., Matei, O., Sitar, C.P., 2013. An improved hybrid algorithm for solving the generalized vehicle routing problem. Neurocomputing 109, 76 – 83.

Quintero-Araujo, C.L., Caballero-Villalobos, J.P., Juan, A.A., Montoya-Torres, J.R., 2017. A biased-randomized metaheuristic for the capacitated location routing problem. International Transactions in Operational Research 24, 1079–1098.

Quintero-Araujo, C.L., Gruler, A., Juan, A.A., Faulin, J., 2019. Using horizontal co-operation concepts in integrated routing and facility-location decisions. International Transactions in Operational Research 26, 551–576.

Rabe, M., Deininger, M., Juan, A., 2020. Speeding up computational times in simheuristics combining genetic algorithms with discrete-event simulation. Simulation Modelling Practice and Theory 103, 102089.

Reisig, W., 2012. Petri nets: an introduction. volume 4. Springer Science & Business Media.

Reyes-Rubiano, L., Ferone, D., Juan, A.A., Faulin, J., 2019. A simheuristic for routing electric vehicles with limited driving ranges and stochastic travel times. SORT. Statistics and Operations Research Transactions , 3–24.

Ritzinger, U., Puchinger, J., Hartl, R.F., 2015. A survey on dynamic and stochastic vehicle routing problems. International Journal of Production Research 54, 215–231.

Shi, C., Li, T., Bai, Y., Zhao, F., 2016. A heuristics-based parthenogenetic algorithm for the VRP with potential demands and time windows. Scientific Programming 2016, 1–12.

Silva, M., 1993. Introducing Petri nets, in: Practice of Petri Nets in Manufacturing. Springer Netherlands, pp. 1–62.

Silva, M., 2018. On the history of discrete event systems. Annual Reviews in Control 45, 213–222.

Spliet, R., Gabor, A.F., Dekker, R., 2014. The vehicle rescheduling problem. Computers & Operations Research 43, 129–136.

Zou, H., Dessouky, M.M., 2018. A look-ahead partial routing framework for the stochastic and dynamic vehicle routing problem. Journal on Vehicle Routing Algorithms 1, 73–88.