Production, Manufacturing and Logistics

Measures of dynamism and urgency in logistics

Rinde R. S. van Lon, Eliseo Ferrante, Ali E. Turgut, Tom Wenseleers, Greet Vanden Berghe, Tom Holvoet

ABSTRACT

Dynamism was originally defined as the proportion of online versus offline orders in the literature on dynamic logistics. Such a definition however, loses meaning when considering purely dynamic problems where all customer requests arrive dynamically. Existing measures of dynamism are limited to either (1) measuring the proportion of online versus offline orders or (2) measuring urgency, a concept that is orthogonal to dynamism, instead. The present paper defines separate and independent formal definitions of dynamism and urgency applicable to purely dynamic problems. Using these formal definitions, instances of a dynamic logistic problem with varying levels of dynamism and urgency were constructed and several route scheduling algorithms were executed on these problem instances. Contrary to previous findings, the results indicate that dynamism is positively correlated with route quality; urgency, however, is negatively correlated with route quality. The paper contributes the theory that dynamism and urgency are two distinct concepts that deserve to be treated separately.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Logistic optimization problems aim at minimizing costs while serving customers’ transportation requests. The most common problem formalization is the vehicle routing problem (VRP) (Dantzig & Ramser, 1959). Roads are treated as edges of a graph and a traveling salesman problem (TSP) is solved for one or more vehicles represented in such a graph (Flood, 1956). In practice, vehicle schedules are devised offline, after all customer requests have been received, and are applied later on without the possibility to modify the schedules once the vehicles have started servicing.

A number of technological advances have fostered new interest and transformed problems in the domain of logistics. Such advances are the introduction of the Global Positioning System (GPS) in 1996, the increasing accuracy of Geographic Information Systems (GIS), and more recently the development and spread of tablets and smart phones with high-bandwidth internet. Online changing of routes or devising completely new routes is now possible due to the availability of accurate information on the position of all vehicles. These developments open new avenues for increasing customer satisfaction (i.e. relatively fast shipping of goods, even at the day of ordering), while operational costs and environmental impact can be further decreased. In the dynamic variant, the typical dynamic aspect is the arrival time of the request containing the useful information needed to compute optimal routes for the vehicles (Pillac, Gendreau, Gueret & Medaglia, 2013).

Dynamic logistics is a well researched topic continuing to receive widespread attention (Berbeglia, Cordeau, & Laporte, 2010; Parragh, Doerner, & Hartl, 2008; Pillac et al., 2013). Psaraftis (1995) and later Eksioglu, Vural, and Reisman (2009) devised taxonomies for the (dynamic) VRP, but did not formally define dynamism as such. Pillac et al. (2013) suggested that a better formalization of the dynamics would allow more precise classification of problem instances. Based on such a classification, it would be possible to scientifically assess the quality of algorithms for dynamic logistic problems in different circumstances. For instance, datasets such as those presented in Li and Lim (2001), Mitrović-Minić and Laporte (2004) and Gendreau, Guertin, Potvin, and Seguin (2006) could be classified and compared quantitatively and it would be possible to find specific dynamic properties within dynamic logistics where one class of algorithms performs better
than others. The cornerstone of a formalization of dynamics in logistics is a formal definition of dynamism. Intuition suggests that the frequency of change should be part of such a definition of dynamism. A more dynamic problem is characterized by a more continuous distribution of request arrivals. Static problems, on the other hand, have all requests available at the same time or, alternatively, become available in bursts and thus have a more varying request arrival frequency. Furthermore, different optimization algorithms likely differ in their ability to find near-optimal solutions for highly dynamic problems. When information is clustered together, the available time can be used for devising a good schedule, contrasting, frequent changes of the problem definition make scheduling in advance almost useless and favor a completely reactive strategy instead.

Lund, Madsen, and Rygaard (1996) proposed the first formal measure for quantifying dynamism in logistic problems. They define dynamism as the proportion of requests known after the scheduling phase (i.e. when vehicles are already shipping) with respect to the total number of requests. Their measure considers a problem where all requests arrive during shipping as 100 percent dynamic. Contrary to our intuition, the relative timing of the requests does not influence the value of this dynamism measure. Larsen, Madsen, and Solomon (2002) recognized the limitation of the measure by Lund et al. and aimed at fixing it by taking into account the urgency of a request. Larsen et al.’s measure considers a request to be more dynamic when announced closer to its deadline. However, this approach fails to measure what intuitively could be considered dynamism, since it does not measure the relative distribution of request announcements. On closer inspection, the concept of urgency is included in the degree of dynamism considered by Larsen et al. Moreover, Larsen et al. showed that for problems with a high dynamism value, the algorithms tested produced a lower quality schedule. Based on their experimental setup, concluding whether the negative correlation between their measure and schedule quality is the result of dynamism, urgency or a combination thereof is nearly impossible.

The present paper investigates whether the experimental observations reported by Larsen et al. are caused by dynamism, urgency or both. We analyze whether splitting urgency and dynamism into separate concepts is desirable. To conduct a sound scientific evaluation, we need to be able to formally define both dynamism and urgency as two separate concepts and to develop the tools for classifying logistic scenarios. These tools enable generating instances of logistic problems with varying levels of dynamism and urgency. The instances are realistic, while capable of sharing common characteristics, excepting differing levels of dynamism and urgency. The dataset thus generated contains instances of the dynamic pickup and delivery problem with time windows (PDPTW), a special case of the VRP that is sufficiently relevant to allow general claims. Further, the dataset, the simulator and all code is available online to allow reproducibility of all results.

The paper is organized as follows. First, the relevant literature is discussed (Section 2). Second, dynamic pickup and delivery problems (PDPs) are formally defined and dynamism and urgency are explained intuitively (Section 3). The novel measures which form the main contribution of the paper are explained (Section 4) and the empirical evaluation is discussed (Section 5). Finally, the conclusions based on the experimental evaluation are presented and the usefulness of the proposed measures to advance the field of dynamic logistics and beyond is discussed (Section 6).

2. Related work

The VRP was first introduced (Dantzig & Ramser, 1959) as a generalization of the TSP (Flood, 1956). A dynamic version of VRP was first studied considering a dynamic version of a special case of VRP transportation of people (Wilson et al., 1977): the dial-a-ride-problem (DARP) (Cordeau & Laporte, 2003). The customer requests (trips from a source to a destination) in a DARP appear dynamically. These type of requests were later formally defined in Psaraftis (1980) as immediate requests, distinguished from advanced requests that are received before the beginning of the planning horizon.

In this section, we review the existing literature on previously proposed dynamism measures. We also briefly review the state of the art on the dynamic PDPTW.

2.1. Dynamism and measures

The first dynamism measure was introduced by Lund et al. (1996) and later refined by Larsen et al. (2002). Section 4.2 discusses these measures in detail after an intuitive definition of dynamism is presented. Larsen (2000) proposed a framework that distinguishes between weakly, moderately and strongly dynamic systems. The intention of this framework is to quickly find an appropriate algorithm based on the problem’s classification.

Beside these works, we have no knowledge of any work that defines measures for dynamism within the field of operations research. Nevertheless, several authors make interesting observations related to dynamism in logistics.

A first observation, by Kilby, Prosser, and Shaw (1998), is that the arrival rate of new tasks in a dynamic VRP is important. If the problem updates constantly, an algorithm will require more restarts than in the case where requests arrive in widely separated bursts. Similarly, Pillac et al. (2013) note that the frequency of updates in problem information have a dramatic impact on the time available for optimization. The statements made by Kilby et al. and Pillac et al. align with what intuitively could be considered dynamism since the arrival rate of requests is similar to the relative distribution of request announcements.

A second observation, also by Kilby et al. (1998), is about the time at which a commitment to serve a customer at a particular time must be made. The time of the commitment is one of the fundamental questions in dynamic routing. Kilby et al. (1998) define a dynamism-related measure called the commit horizon, which denotes the period where the schedule is fixed before the latest possible commit time. The latter is problem-dependent but is often defined as the operation’s starting time. Although we did not consider the commit horizon in our study, it may be an interesting property to investigate related to dynamism.

A third noteworthy observation about dynamism in logistics is made by Borndörfer, Grötschel, Klostermeier, and Küttner (1999). In the static DARP, the computed schedule and the schedule executed on the next day often differ significantly because of cancellations of requests, spontaneous requests, vehicle breakdowns and other unpredictable events. This observation suggests that static DARPs are exceptional in practice.

2.2. Literature review on the dynamic PDPTW

Gendreau and Potvin (1998) discussed application domains in which dynamic vehicle routing problems occur, such as dial-a-ride (taxi) problems and courier and repair services. Berbeglia et al. (2010) presented an extensive overview of variants of dynamic PDPs. The dynamic PDPTW is a special case of the dynamic VRP. It should be noted that the dynamic PDPTW is often seen as a stochastic problem, in which some knowledge about the nature of the arrivals is known in advance in a stochastic way, while the actual requests become known only during the operation day (Fu, 2002; Pillac et al., 2013). Psaraftis (1995) remarked, without formally defining near-term, that in dynamic vehicle routing
near-term events are more important than long-term events. Research on the dynamic PDPTW has mainly concentrated on algorithm development, in this section, we only review work in which the dynamic PDPTW is seen from a completely dynamic perspective, without any a priori knowledge.

Gendreau et al. (2006) presented a dynamic version of tabu search with a neighboring structure based on ejection chains. The optimization procedure is run while the environment is static. When new requests arrive, or when a vehicle has finished pickup or delivery, the algorithm performs insertion and ejection moves. Madsen, Ravn, and Rygaard (1995) developed an insertion heuristic to tackle the dynamic DARP with time windows for moving elderly and disabled people in Denmark.

Mitrović-Minić, Krishnamurti, and Laporte (2004) presented an approach based on two time horizons: a short time horizon aimed at achieving the short-term goal of minimizing the distance travelled, and a longer time horizon aimed at achieving the long-term goal of facilitating the insertion of future requests. Five more rolling horizon heuristics were considered and compared in Yang, Jallot, and Mahmassani (2004). Mitrović-Minić and Laporte (2004) first considered two very simple heuristics: drive-first, that requires a vehicle to drive as soon as it is feasible according to the earliest departure time, and wait-first that instead requires the vehicle to wait at its current location as long as it is feasible. Two more waiting strategies aim to achieve a trade-off between the first two heuristic and at better handling waiting times in order to facilitate insertion of future requests.

Pureza and Laporte (2008) proposed two strategies, a waiting and a buffering strategy. Differently from Mitrović-Minić and Laporte (2004), the waiting strategy exploits extra information provided by the computation of the faster path. In order to minimize the earliness of a service at a location, the buffering strategy, instead, postpones the assignment of the least urgent new requests to the latest possible time. The idea underlying both strategies is to schedule requests in batches, retrieving as much information as possible to produce better schedules.

In search of a dispatching algorithm that could imitate a human being, Potvin, Dufour, and Rousseau (1993) presented a learning system based on linear programming. The system is able to learn an optimal policy taking into account expert decisions in former situations. The same idea was implemented by Shen, Potvin, Rousseau, and Roy (1995), who based an approach on neural networks and by Benyahia and Potvin (1998), where genetic programming was used.

An alternative approach to the dynamic PDPTW relies on a decentralized multi-agent system (MAS) instead of a centralized decision maker. Every vehicle is assumed to be able to perform some computation and to communicate with any other vehicle anytime. A first comparison between centralized and decentralized approaches was performed by Mes, van der Heijden, and van Harten (2007). They compared traditional heuristics developed in earlier work (van der Heijden, Ebben, Gademann, & van Harten, 2002) against novel distributed MAS that use a Vickrey auction to bid for new pickup and delivery requests when they appear. It was shown that the performance of the MAS approach is often at least as good as traditional heuristics. Mes, van der Heijden, and Schuur (2010) further improved the performance of the MAS by introducing a look-ahead mechanism. Bidding uses value functions to estimate the expected future revenue of inserting a new order in an agent’s schedule. Other look-ahead strategy combinations were proposed in Mes, van der Heijden, and Schuur (2013).

3. Dynamic pickup-and-delivery problems

In PDPs, a fleet of vehicles deals with customer transportation requests. A request is handled when an item is transported from pickup to delivery location as requested by the customer. In dynamic PDPs, requests may arrive at any time during the fleet’s operating hours, necessitating the maintenance of a flexible schedule. No prior information is available about the number of requests that may still arrive nor about their locations or time windows.

3.1. Formal definition

The definition of the dynamic PDP used throughout this paper is based on Gendreau et al. (2006). A scenario, which describes the unfolding of a dynamic PDP, is defined as a tuple:

\[ (T, \mathcal{E}, \mathcal{V}) := \text{scenario}, \]

where

\[ (0, T) := \text{time frame of the scenario}, \quad T > 0 \]
\[ \mathcal{E} := \text{list of events}, \quad |\mathcal{E}| \geq 2 \]
\[ \mathcal{V} := \text{set of vehicles}, \quad |\mathcal{V}| \geq 1 \]

\[ (0, T) \] is the period in which the fleet of vehicles \( \mathcal{V} \) have to handle all customer requests. The events represent customer requests. We distinguish between advance events and dynamic events. Advance events are known before time 0 of the time frame of the scenario. Dynamic events are instead revealed between time 0 and time \( T \) and describe new transportation requests, or can possibly introduce other new information. Each event \( e_i \in \mathcal{E} \) is defined by the following variables:

\[ a_i := \text{announce time} \]
\[ p_i := [p^L_i, p^R_i] = \text{pickup time window}, \quad p^L_i < p^R_i \]
\[ d_i := [d^L_i, d^R_i] = \text{delivery time window}, \quad d^L_i < d^R_i \]
\[ ps_i := \text{pickup service time span} \]
\[ dst_i := \text{delivery service time span} \]
\[ ploc_i := \text{pickup location} \]
\[ dloc_i := \text{delivery location} \]
\[ tt_i := \text{travel time from pickup location to delivery location} \]

Similar to Larsen et al. (2002) we define reaction time as the length of the interval between the order arrival time \( a_i \) and the closing of the pickup time window \( p^R_i \):

\[ r_i := p^R_i - a_i = \text{reaction time} \quad (1) \]

The time window related variables of a transportation request are visualized in Fig. 1.

Furthermore we assume that:

- vehicles start at a depot and have to return after all orders are handled;
- the fleet of vehicles \( \mathcal{V} \) is homogeneous;
- the cargo capacity of vehicles is infinite \((\text{e.g. courier service})\);
- the vehicle is either stationary or driving at a constant speed;
- contrasting to Gendreau et al. (2006), vehicle diversion is allowed, meaning that a vehicle can divert from its destination at any time;
- vehicle fuel is infinite and driver fatigue is not an issue;
- each location can be reached from any other location; and,
- the scenario is completed when all pickup and deliveries have been executed and all vehicles have returned to the depot.
Vehicle schedules are subject to both hard and soft constraints. The openings of time windows are hard constraints and therefore:

\[ sp_{ij} \geq p_{ij}^l \]  \hspace{1cm} (2)
\[ sd_{ij} \geq d_{ij}^l \]  \hspace{1cm} (3)

\( sp_{ij} \) is the start of the pickup operation of order event \( e_i \) by vehicle \( v_j \); similarly, \( sd_{ij} \) is the start of the delivery operation of order event \( e_i \) by vehicle \( v_j \). The time windows closings (\( p_{ij}^r \) and \( d_{ij}^r \)) are soft constraints. They are incorporated into the objective function, which is defined similarly to Gendreau et al. (2006) and needs to be minimized:

\[
\min := \sum_{j \in V} \left( vtt_j + td\{bd_j, T\} \right) + \sum_{i \in E} \left( td\{sp_{ij}, p_{ij}^r\} + td\{sd_{ij}, d_{ij}^r\} \right)
\]  \hspace{1cm} (4)

where

\[ td\{\alpha, \beta\} := \max\{0, \alpha - \beta\} = \text{tardiness} \]  \hspace{1cm} (5)

\( vtt_j \) is the total travel time of vehicle \( v_j \); \( bd_j \) is the return time of vehicle \( v_j \) to the depot. The objective function computes the total vehicle travel time, the total tardiness of vehicles returning to the depot and the total pickup and delivery tardiness. The objective function determines the route cost of a solution, where a fleet of vehicles executes a scenario. A low route cost corresponds with a high quality route.

We further impose the following hard constraints on the construction of scenarios to ensure consistency and feasibility of individual orders:

\[ r_i \geq 0 \]  \hspace{1cm} (6)
\[ d_i^g \geq p_{ij}^g + pst_i + tt_i \]  \hspace{1cm} (7)
\[ d_i^l \geq p_{ij}^l + pst_i + tt_i \]  \hspace{1cm} (8)

These constraints are visualized in Fig. 2.

The reaction time constraint (Eq. (6)) ensures that an order is always announced before its due date. The time window constraints (Eqs. (7) and (8)) ensure that pickup and delivery time windows are compatible. These constraints ensure that a pickup operation started at any time within \( p_i \) allows a delivery within \( d_i \), given that a vehicle is available and respecting vehicle capacity, service time and travel time constraints.

4. Measure design

Although we aim at measuring properties of dynamic PDPTWs, the concepts of dynamism and urgency are not limited to measuring properties of the problem class dynamic PDPTW. In general, properties of a series of events should be measured.

4.1. Intuitive definitions

Dynamism and urgency are abstract variables that capture two aspects of dynamic PDPTWs. We consider these variables to be problem related as opposed to algorithm related; the applied algorithm should have no influence on the value of the measures. However, the dynamism and urgency measures may assist in choosing an appropriate algorithm for a PDPTW instance. Further, the measures should be conceptually orthogonal, i.e. a measure for one concept should not be (partially) mixed with aspects of other concepts. Therefore, urgency and dynamism should not be correlated.

4.1.1. Dynamism

We base our notion of dynamism on the meaning of the word dynamism: “marked by usually continuous and productive activity or change” (Merriam-Webster.com, 2014). Therefore we consider the degree of dynamism to be the continuity of change. A very dynamic scenario is one that changes continuously while a less dynamic scenario only changes occasionally. This is visualized in Fig. 3.

We further define a change to be an event that introduces additional information to the problem, such as an order event as defined formally in Section 3.1. In our interpretation, knowing the dynamism of a problem does not give any extra information on the predictability of events.

4.1.2. Urgency

Urgency is an indicator of the reaction time available for responding to an incoming order. Urgency can be expressed in time units and defined as the difference between order arrival time and closing of the pickup time window as shown in Fig. 4.

Fig. 3. Visualization of order arrival times, each red bar indicates an event in which a new order is announced. The figures (a–f) are presented in decreasing order of dynamism. In (a) the events have equal interarrival times and are nicely distributed over the period, in (b, c) we see that changes occur less frequently. In (d, e) all events arrive in one or two batches making it less continuous and therefore less dynamic. In (f) all 10 events arrive at the same time resulting in a scenario with no dynamism. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)
4.2. Degree of dynamism

The degree of dynamism was defined by Lund et al. (1996):

\[
\text{dod} := \frac{\text{Number of dynamic requests}}{\text{Total number of requests}}
\]

(9)

It is the proportion of dynamic requests with respect to the total requests (including dynamic and advance requests). The definition ignores information related to dynamism defined in Section 4.1.1. For instance, a scenario where all events are announced in a relatively short burst has the same dod as a scenario where events are announced more evenly throughout the day (see Fig. 5).

This means that the applicability of dod is limited and not suitable for purely dynamic scenarios (i.e., scenarios without advance events). Since this measure does not measure dynamism as we conceive it, we propose to rename it to the proportion of dynamic requests.

Larsen et al. (2002) recognized the limitations of Eq. (9) and designed the effective degree of dynamism in an attempt to measure dynamism more accurately:

\[
\text{edod}_{tw} := \sum_{i=1}^{n_{imm}} \left( \frac{a_i}{T_i} \right)
\]

(10)

Where \(n_{imm}\) is the number of dynamic requests and \(n_{tot}\) is the total number of events. They also proposed a similar measure that takes time windows into account:

\[
\text{edod}_{tw} := \frac{1}{n_{tot}} \sum_{i=1}^{n_{tot}} \left( 1 - \frac{r_i}{T_i} \right)
\]

(11)

Since the problem under investigation includes time windows, the analysis focuses on edod\(_{tw}\) (Eq. (11)), without loss of generality.

Fig. 6 shows three scenarios with their respective value for the edod\(_{tw}\) measure.

Fig. 6 (a) and (b) show two similar scenarios that differ in the arrival times of the transportation requests. They have the same value for edod\(_{tw}\), even though their arrival times are quite different. It differs from the definition of dynamism introduced in the present paper (Section 4.1.1), where the scenario depicted in Fig. 6(b) has a higher degree of dynamism than that of Fig. 6(a).

The arrival times in Fig. 6(b) have been multiplied by 2 in order to obtain Fig. 6(c). The reaction times are the same. According to our definition of dynamism these two scenarios should have the same level of dynamism, but according to edod\(_{tw}\) they are different. This difference is problematic because it means that edod\(_{tw}\) is dependent on the length of the scenario \(T\), hence scenarios of different length cannot be compared using the edod\(_{tw}\) measure.

4.3. Dynamism measure

We define the list of interarrival times \(\Delta\) as follows:

\[
\Delta := \{\delta_0, \delta_1, \ldots, \delta_{|\mathcal{E}|-1}\} = \{a_j - a_i| j = i + 1 \land \forall a_i, a_j \in \mathcal{E}\}
\]

(12)

\[
|\Delta| := |\mathcal{E}| - 1
\]

(13)

Based on the visualization of a scenario with 100 percent dynamism in Fig. 3(a) we can define a perfect interarrival time that is required for 100 percent dynamism as follows:

\[
\theta := \text{perfect interarrival time} = \frac{T}{|\mathcal{E}|}
\]

(14)

The perfect interarrival time enables computing the deviation of an interarrival time relative to the 100 percent case:

\[
\sigma_i := \begin{cases} 
\theta - \delta_i & \text{if } i = 0 \text{ and } \delta_i < \theta \\
\theta - \delta_i + \frac{\theta - \delta_i}{\theta} \times \sigma_{i-1} & \text{if } i > 0 \text{ and } \delta_i < \theta \\
0 & \text{otherwise}
\end{cases}
\]

(15)
Consequently, the deviation of an entire scenario is defined as:

$$\sum_{i=0}^{\lfloor |\Delta| \rfloor} \sigma_i$$  \hspace{1cm} (16)

Since bursts are defined as interarrival times that are smaller than $\theta$, this definition focuses on interarrival times that are smaller than $\theta$. In case $\delta_i < \theta$, a recursive penalty, expressed by the term $\frac{\theta - \delta_i}{\theta} \times \sigma_{i-1}$, is applied. This penalty proportionally adds the deviation of the previous interarrival time. In short, the penalty term is used to recognize bursts, to measure their size, and to take their contribution into account. The motivation for this recursive penalty can best be explained using an example. Consider the scenario shown in Fig. 7.

Following the description of dynamism in Section 4.1.1, the scenario in Fig. 7(a) is more dynamic than the one in Fig. 7(b). When examining the interarrival times of both scenarios, it shows that $\theta = 1$ and both have five interarrival times of .1 and four interarrival times of 2:

$$\Delta^a = \{.1, .2, .1, .2, .1, .2, .1, .2\}$$

$$\Delta^b = \{.1, .1, .1, .1, .2, 2, 2, 2\}$$

Computing only the deviations from the perfect interarrival time is not enough to distinguish between these scenarios. Therefore the recursive penalty used in Eq. (15) distinguishes between these two scenarios by taking into account the deviation of the preceding interarrival time. In this example (Fig. 7) these deviations become:

$$\sigma^a = \{.9, 0, .9, 0, .9, 0, .9\}$$

$$\sigma^b = \{.9, 1.71, 2.439, 3.0942, 3.68478, 0, 0, 0\}$$

The deviation of the event series from the 100 percent case has to be normalized with respect to the theoretical maximum deviation for a scenario (i.e. the 0 percent case). We compute the maximum as follows:

$$\sum_{i=0}^{\lfloor |\Delta| \rfloor} \hat{\sigma}_i$$  \hspace{1cm} (17)

where

$$\hat{\sigma}_i := \theta + \begin{cases} \frac{\theta - \delta_i}{\theta} \times \sigma_{i-1} & \text{if } i > 0 \text{ and } \delta_i < \theta \\ 0 & \text{otherwise} \end{cases}$$  \hspace{1cm} (18)

In Eq. (18), the perfect interarrival time is multiplied by the number of interarrival times and the recursive penalty is also added in order to dynamically increase the maximum. Adding the recursive penalty to the maximum prevents $\hat{\sigma}$ from becoming greater than $\sigma$.

Combining Eqs. (15) and (17) the definition of dynamism becomes:

$$\text{dynamism} := 1 - \frac{\text{deviation}}{\text{max deviation}} = 1 - \frac{\sum_{i=0}^{\lfloor |\Delta| \rfloor} \sigma_i}{\sum_{i=0}^{\lfloor |\Delta| \rfloor} \hat{\sigma}_i}$$  \hspace{1cm} (19)

Continuing with the scenarios in Fig. 7, dynamism is 50 percent for scenario A and 27.6 percent for scenario B (see Appendix A for the complete calculation).

4.4. Urgency measure

Based on Fig. 4 we can define a measure for urgency of a single order event:

$$\text{urgency}(e_i) := p_i^R - a_i = r_i$$  \hspace{1cm} (20)

Urgency is the reaction time expressed in time units. In order to get an indication of the urgency of an entire scenario, one can compute the mean and standard deviation of urgency.

Note that this definition is similar to Larsen et al.’s effective degree of dynamism (Eq. (11)) but differs in a key aspect. The urgency value is not normalized to the length of the scenario. We believe that the length of a scenario and urgency should be independent and should therefore not be coupled in the definition of urgency.

5. Evaluation

We evaluate the dynamism and urgency measures by investigating their impact to route quality generated by scheduling algorithms. Route quality is defined as the inverse of the cost of a route, where route cost is computed using the objective function defined in Eq. (4). Three hypotheses are investigated:

- When increasing the dynamism of a scenario the average route quality of an algorithm decreases
- When increasing the urgency of a scenario the average route quality of an algorithm decreases
- When increasing both dynamism and urgency the average route quality of an algorithm decreases

5.1. Dataset generator

To evaluate the influence of dynamism and urgency on strategies developed for dynamic PDPTW, it is imperative to be able to create scenarios with any level of urgency and dynamism. Furthermore, these scenarios have to be as similar as possible, except for their possibly different urgency and dynamism, while still being the result of a stochastic process. Therefore, a dataset generator has been constructed and used to generate a dataset with 11 levels of dynamism (0–100 percent with steps of 10 percent) and 10 levels of urgency (0–45 minutes with steps of 5 minutes). This results in 110 different scenario settings. We produced 20 different instances for each setting resulting in 2200 scenarios.

5.1.1. Controlling dynamism of time series

A homogeneous Poisson process is a common model for the arrival of stochastic events, e.g. phone calls (Willkomm, Machiraju, Bolot, & Wolisz, 2009) and requests of an individual document on a web server (Arilit & Williamson, 1997). The dynamism of time series obtained using a Poisson process was first investigated. Interestingly, the results show that time series generated using a homogeneous Poisson process are in a range of 45–60 percent dynamism. Other methods for generating time series with dynamism lower than 45 percent and dynamism higher than 60 percent have been investigated. A time period $T = 12$ and number of events...
\[ |E| = 360 \] for generating time series was used. The homogeneous Poisson process has a constant intensity function, which is defined as:

\[
\lambda(t) = \frac{|E|}{T} = 30
\]

(21)

A non-homogeneous Poisson process has a variable intensity function. It is based on a sine wave to control the dynamism of a scenario. By varying the parameters of the sine wave, the properties of event bursts can be controlled:

\[
\lambda(t) = a \cdot \sin(t \cdot f \cdot 2\pi - \pi \cdot p) + h
\]

(22)

\[ a = \text{amplitude} \]

(23)

\[ f = \text{frequency} \]

(24)

\[ p \sim U(0, 1) \text{ phase shift} \]

(25)

\[ h \sim U(-.99, 1.5) \text{ height} \]

(26)

In order to keep the total number of events constant with different levels of dynamism, the amplitude and height parameters are rescaled such that the total area under the intensity function equals \(|E|\). After rescaling, the resulting events following a non-homogeneous Poisson process are generated using the thinning method (Lewis & Shedler, 1979). Fig. 8 visualizes the effect of the height parameter on the event intensity and therefore on the dynamism of a scenario.

Since the non-homogeneous Poisson process only generates scenarios with a dynamism lower than or equal to scenarios generated with the homogeneous Poisson process, a different method had to be used for generating more dynamic scenarios. A method that generates higher levels of dynamism is attained by drawing interarrival times from a normal distribution. We used the truncated normal distribution \(\lambda\left(\frac{I}{T}, 0.04\right)\) with a lower bound of 0 and a standard deviation of 0.04 was found experimentally to yield the best results. If a value \(x\) was drawn such that \(x < 0\), a new number was drawn from the distribution. Truncating a normal distribution actually shifts the mean, hence the mean was rescaled to make sure the effective mean was equal to \(\frac{I}{T}\).

The fourth method for generating interarrival times is a uniform distribution with mean \(\frac{I}{T}\) and a maximum deviation \(\sigma\). The \(\sigma\) value is (for each scenario again) drawn from the truncated normal distribution \(\lambda(1, 1)\) with bounds \([0, .25]\). If a value \(\sigma\) is obtained from the distribution such that \(\sigma > .25\) or \(\sigma < 0\), a new value is drawn. The mean is not scaled, and therefore the effective mean of \(\sigma\) is higher than 1.

An experiment was conducted where each previously described method was used to generate 1000 samples (time series). For each sample the dynamism was computed using Eq. (19). We repeated this experiment until we found the parameters that produce scenarios in the entire range of 0–100 percent dynamism. Fig. 9 shows the final results of this experiment as a frequency diagram.

Based on these experimental results we conclude that these four methods can be used to generate scenarios with dynamism ranging from 0 to 100 percent. Table 1 shows the time series generators and the levels of dynamism they generate.

The probability that one of the time series generated has exactly the desired dynamism value is very small. A radius of 1 percent has therefore been defined around each dynamism value. For this dataset, we consider a scenario with dynamism \(d\) such that \(n - 1 < d < n + 1\) to have dynamism \(n\), where \(n\) is one of the dynamism values listed in Table 1.
Table 1
Overview of dynamism values used in the dataset and the corresponding time series generator.

<table>
<thead>
<tr>
<th>Dynamism values</th>
<th>Time series generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 5, 10, 15, 20, 25, 30, 35, 40, 45</td>
<td>Non-homogeneous Poisson process</td>
</tr>
<tr>
<td>50, 55</td>
<td>Homogeneous Poisson process</td>
</tr>
<tr>
<td>60, 65</td>
<td>Normal distribution</td>
</tr>
<tr>
<td>70, 75, 80, 85, 90, 95, 100</td>
<td>Uniform distribution</td>
</tr>
</tbody>
</table>

5.1.2. Generating comparable scenarios with different dynamism and urgency levels

The aim of the dataset is to have a set of scenarios where all settings are the same except for the dynamism and urgency levels. Also, we strive to minimize any interactions between variables, e.g. dynamism should not correlate with time window intervals. This ensures that any effect measured is solely the result of the difference in dynamism and urgency.

The dataset generators are stochastic functions. Although all parameters are set as to make desired results as likely as possible, undesirable scenarios cannot be completely avoided. Therefore, we employ a filter that only accepts scenarios corresponding with the following requirements:

- All scenarios must have exactly |S| events, (the time series generators all produce time series which have on average the correct number of events, but not always).
- Scenarios must have a dynamism that fits in one of the dynamism bins from Table 1.

We further define the concept of office hours as the period [0, O) in which new orders are accepted. To ensure feasibility of individual orders we need to take into account the travel time, service time durations and urgency:

\[ O = T - pst_{\text{max}} - dst_{\text{max}} - \begin{cases} 2 \cdot tt_{\text{max}} & \text{if } u < \frac{1}{2} \cdot tt_{\text{max}} \\ 1 \frac{1}{2} \cdot tt_{\text{max}} - u & \text{otherwise} \end{cases} \] (27)

Here, \( pst_{\text{max}} \) and \( dst_{\text{max}} \) are the maximum pickup and delivery service times respectively, \( tt_{\text{max}} \) is the maximum travel time between a pickup and delivery location and \( u \) is urgency.

The pickup and delivery time windows have to be randomly chosen while respecting the constraints set out by the urgency level and the announce time. The \( p_i^b \) is defined as the sum of \( a_i \) and \( u \), hence it follows that \( p_i^b \) needs to be between \( a_i \) and the sum of \( a_i \) and \( u \):

\[ p_i^b = \begin{cases} \sim t(d(a_i, p_i^b - 10)) & \text{if } u > 10 \\ a_i & \text{otherwise} \end{cases} \] (28)

Here, 10 is the minimum pickup time window length unless urgency is less than 10, in that case the urgency level is the pickup time window length. The upper bound of \( d_i^b \) can be defined as:

\[ ubd_i^b = T - tt(dloc, depot_{loc}) - dst_i \] (29)

This translates as the latest possible time to start the delivery operation such that the delivery time window constraints are met and the vehicle can still be back at the depot on time. The lower bound of \( d_i^b \) was already defined in Eq. (8):

\[ lbd_i^b = p_i^b + pst_i + tt_i \] (30)

We define a minimum delivery time window length of 10, which results in an upper bound of \( d_i^b \):

\[ ubd_i^b = ubd_i^b - 10 \] (31)

Based on these bounds we draw the opening of the delivery time window from the following uniform distribution:

\[ d_i^b \sim U(lbd_i^b, \text{max}(lbd_i^b, ubd_i^b)) \] (32)

To find \( d_i^b \) we need to redefine the lower bound (from Eq. (7)) by using the actual value of \( d_i^b \):

\[ lbd_i^b = \min\{\max\{p_i^b + pst_i + tt_i, d_i^b + 10\}, ubd_i^b\} \] (33)

Finally, the closing of the delivery time window is defined as:

\[ d_i^b \sim U(lbd_i^b, ubd_i^b) \] (34)

All locations in a scenario are points on the Euclidean plane. It has a size of 10 by 10 kilometer with a depot at the center of this square. At the start of the scenario all 10 vehicles are at the depot. The vehicles have a constant travel speed of 50 kilometer per hour. All pickup and delivery locations are drawn from a two dimensional uniform distribution \( t_d(0, 10) \).

For simulating a scenario we use the discrete time simulator RinSim (van Lon & Holvoet, 2012) version 3.0.0 (van Lon, 2014b). The time unit is set to milliseconds and the tick size to 1000 milliseconds. The pickup and delivery service times \( pst \) and \( dst \) are set to 5 minutes. For reproducibility, all code (van Lon, 2016a) and the data and results (van Lon, 2016b) are published.

5.2. Heuristic algorithms used to solve dynamic PDPTW

The cheapest insertion heuristic (Algorithm 1) and the 2-opt optimization procedure (Algorithm 2) were used in the experiments. Since the 2-opt procedure requires a complete schedule as input, it uses the cheapest insertion heuristic for inserting new orders to yield a complete schedule. These two heuristics have been used in earlier work for vehicle routing problems (Coslovich, Pesenti, & Ukovich, 2006; Pasarats, 1983; Solomon, 1987) and the heuristics are general enough not to have a bias towards scenarios with specific levels of dynamism or urgency.

Algorithm 1: Cheapest insertion heuristic, source code available in van Lon (2014a).

```
Input: (T, E, V);
Data: S;
/* A scenario as input */
S_{next} = ∅

for e ∈ E, e ∉ S do
  /* generate all PDPT insertion points in the current schedule: */
  insertions = generate_insertion_points(S)
  for i ∈ insertions do
    /* construct a new schedule by inserting e at insertion i */
    S_{new} = construct(S,e)
    if cost(S_{new}) < cost(S_{best}) then
      S_{best} = S_{new}
  end
end
```

Algorithm 2: 2-opt procedure, source code available in van Lon (2014a).

```
Input: S
S_{best} = S
swaps = generate_swaps(S)

foreach e ∈ swaps do
  S_{new} = swap(S,e)
  if cost(S_{new}) < cost(S_{best}) then
    S_{best} = S_{new}
end

/* If a better schedule has been found, we start another iteration */
if S_{best} ≠ S then
  2-opt(S_{best})
end
```
Each time a new order is announced, the algorithms are executed to produce a new schedule for the fleet of vehicles. It is assumed that execution of the algorithm is instantaneous with respect to the dynamics of the simulations.

5.3. Results and analysis

20 scenarios have been generated per level of dynamism and urgency. For each scenario both algorithms were used for controlling the fleet of vehicles. Figs. 10 and 11 show the experimental results.

The first hypothesis: increasing dynamism decreases average route quality, is not supported by the results (Figs. 10 and 11). In fact, for very urgent scenarios, the route costs decreased when dynamism was increased (Figs. 10(b) and 11(b)). We believe that the relatively small effect of dynamism on route costs can be explained by the assumed instant response of the algorithms. This means that the algorithms never need to or can be interrupted during calculations and will always give their best possible answer. However, it is expected (Kilby et al., 1998) that when requests arrive at a constant rate, an algorithm has relatively little computation time due to restarts triggered by new requests. Therefore, an interesting direction for future work is to investigate advanced algorithms that require more computation time and can also be interrupted.

Our second hypothesis is that more urgent scenarios (lower urgency values) result in increased route costs. This prediction is strongly supported, under all levels of dynamism considered. The
data further suggests that when scenarios become less urgent, route costs decrease diminish. This result is expected. When new requests need to be handled urgently by a fleet of vehicles still busy handling previous requests, it is natural that some delays are introduced. Further, when requests are urgent there is less room for optimizing the length of routes, leading to longer processing times, which eventually results in higher total routing costs.

The third hypothesis states that increasing both dynamism and urgency leads to a lower average route quality. The computational results do not support this hypothesis, as can be seen by the lack of a peak in the top left of Figs. 10(a) and 11(a). Rather surprisingly, an interaction effect can be observed in the lower left corner of the same image. This effect is significantly highly non-linear according to a multiple regression polynomial model. Very urgent scenarios with low dynamism seem to be the hardest scenarios for the algorithms to solve. An explanation for this area of most difficult scenarios is possibly that these scenarios incorporate large bursts of very urgent orders. These bursts may be as big or bigger than the fleet of available vehicles, which therefore quickly run out of time to meet the time windows of requests.

Model selection based on the Akaikie Information Criterion revealed that the route cost was best predicted by a multiple regression model in which dynamism was included as a quadratic polynomial, urgency as a cubic polynomial, plus the interaction between both polynomials (overall model fit for Algorithm 1: adjusted $R^2 = 0.609$, $p < 2 \cdot 10^{-16}$; $\text{AIC} = 37591$. for Algorithm 2: $R^2 = 0.7322$, $p < 2 \cdot 10^{-16}$; $\text{AIC} = 36833$, significance of dynamism, urgency and their interaction effect in Algorithms 1 and 2 was always $p < 10^{-16}$). Furthermore, calculated eta squared values show that urgency had a much larger effect than dynamism or the interaction between both factors (for Algorithm 1: eta squared is 0.54, 0.04 and 0.02, for Algorithm 2: eta squared is 0.70, 0.04 and 0.01).

Based on these results it can be concluded that dynamism and urgency are two different concepts, each affecting the problem in a different way. This justifies the presented theory that dynamism and urgency should be separated, as opposed to the measure by Larsen (2000), in two different measures.

6. Conclusion

The present paper argues that urgency and dynamism are conceptually different and we propose separate measures for both concepts. In support of this conceptual separation, the experimental results show that the degree of dynamism and urgency have a different influence on the solution quality in dynamic logistic problems. Interestingly, the degree of dynamism is negatively correlated with operating costs while more urgent scenarios are correlated with significantly higher operating costs.

The negative correlation between degree of dynamism and operating cost is possibly explained by the algorithms’ assumptions. It can be expected that with a real time setup, where an algorithm can be interrupted during computation, a positive correlation may exist. Furthermore, the correlations between urgency and operating cost are expected to be general in kind, since urgent requests constrain the fleet of vehicles, regardless of the algorithm that is being used.

During the realization of this article the authors published a new benchmark dataset for dynamic PDPTW with different levels of dynamism, urgency and scale (van Lon & Holvoet, 2015), where scale is a combination of number of vehicles, number of requests and area size. The dataset enables systematic comparison of the performance of a broad range of algorithms under varying conditions. Similar to Máhř, Srou, de Weerdt, and Zuidwijk (2010), a comparison of centralized and decentralized approaches applied to this dataset should shed more light on the strengths and weaknesses of both approaches. Such a comparison should clarify whether problems with different levels of dynamism, urgency and scale can better be addressed with either of the two approaches.

Acknowledgments

This research was funded by the Research Fund KU Leuven, the Fund for Scientific Research (FWO) – Flanders (12N7515N), the ESF “H2SWARM” program (GA133.11 ESF), and the KULeuven projects IDO-BioCo3 and Excellence Center Financing PF/2010/007. A.E. Turgut acknowledges the Scientific and Technological Research Council of Turkey (TUBITAK-2219).

Appendix A. Dynamism calculation example

Two examples of computing dynamism using Eq. (19).

For Algorithm 1:

\[
\begin{align*}
\epsilon^a &= \{9, 1, 3, 1.1, 5.1, 5.2, 7.2, 7.3, 9, 9.4\} \\
\theta^a &= 10 \\
\Delta^a &= \{1.2, 1.2, 1.2, 1.2, 1.1\} \\
\sigma^a &= \{9, 0, 9, 0, 0, 0, 0\} \\
\delta^a &= 1, 1, 1, 1, 1, 1, 1 \\
\end{align*}
\]

Dynamism$^a = 1 - \frac{4.5}{9} = 0.5$

For Algorithm 2:

\[
\begin{align*}
\epsilon^b &= \{5.6, 7.8, 8.9, 1.3, 5, 7.9\} \\
\theta^b &= 10 \\
\Delta^b &= \{1.1, 1, 1.1, 1.2, 2, 2\} \\
\sigma^b &= \{9, 1.7, 2.439, 3.0942, 3.68478, 0, 0, 0, 0\} \\
\delta^b &= 1, 1.81, 2.539, 3.1951, 3.78478, 1, 1, 1, 1 \\
\end{align*}
\]

Dynamism$^b = 1 - \frac{11.82798}{16.32888} = 0.2756$

References


