Robustness evaluation of an integer programming-based cross-docking schedule using discrete-event simulation

Anne-Laure Ladier  Gülgün Alpan
Grenoble-INP / UJF-Grenoble 1 / CNRS; G-SCOP UMR5272 Grenoble, F-38031
46 avenue Félix Viallet, 38000 Grenoble, France

Allen G. Greenwood
Department of Industrial Engineering, Mississippi State University
Mississippi State, MS 39762, USA

Abstract

Integer programming (IP) models are powerful tools for operations planning, but may encounter problems when applied in a stochastic environment. We propose a methodology to evaluate the robustness of an IP model using discrete-event simulation. In our case, the IP model determines truck and pallet schedules for a cross-docking facility. A portion of the output from the IP model is used as input to a simulation model that represents the facility. The decisions regarding pallet transfers and truck departures are left to the simulation model to ensure flexibility. Random events are induced through the simulation in order to see how robust the IP-based schedules are against the uncertainties in: (i) the transfer time of pallets through the cross-dock, (ii) the unloading time of pallets from trucks, and (iii) the earliness or delay of the trucks arriving at the platform. The difference between the deterministic and the stochastic scenarios are evaluated regarding the number of pallets put into storage, truck docking and sojourn times. Robustness measures are proposed based on numerical experiments.

1. Introduction

To be able to provide their customers with the best products and the best services at the best price, the manufacturing industry has to focus not only on the manufacturing process itself, but also on the logistics aspect. Customers become highly impatient and expect a fast delivery, but are not always willing to pay an extra cost for it. Therefore, supply chains must adapt and become fast and reactive while maintaining or reducing costs.

Cross-docking is one of the logistics techniques that can help companies achieving these goals. In a cross-docking platform (or crossdock), goods are unloaded from incoming trucks, sorted, and directly reloaded onto other trucks bound for their next destination. Typically, goods stay less than 24 hours in a cross-dock, which reduces inventory costs yet accelerates the flow of goods (Apte et al. [2]).

One key element for an efficient and competitive cross-docking system is effective operations planning and coordination. Stock reduction can only be achieved through good coordination between the inbound and outbound trucks and with a good strategy for moving the pallets inside the platform. The resulting optimization problem is called the truck scheduling problem; reviews can be found in Boysen and Fliedner [4] and Van Belle et al. [23]. In an earlier work, Ladier and Alpan [12] proposed an IP-based model and two heuristics to schedule the truck arrivals and departures in a crossdock. Their model takes into account the demands of transportation providers in terms of their preferred arrival and departure times, and creates a feasible schedule that also minimizes the number of items temporarily stored.

The aim of this paper is to provide a methodology to assess the robustness of the previously developed IP-based model using simulation. Rohrer [20] notes that simulation is useful to test different control algorithms before their implementation. Indeed, simulation can provide an environment that is rather close to real-life situations where schedules generated by IP models can be tested to see how they behave in varying environments.

Discrete-event simulation is often used to gather information about crossdock behavior. Magableh et al. [18] simulate a crossdock facility with Arena, in order to find out how the system reacts when demand increases. Deshpande et al. [6] use simulation with Arena to evaluate the performances of two different door assignments (where each door
is allocated to a destination) in a cross-dock over a year. Liong and Loo [15] simulate a warehouse with Arena to understand which parameters influence the stay time of the trucks at the doors. Yang et al. [25] also use Arena to study tactical decisions in a crossdock, such as: should the items be transferred directly or indirectly between the trucks? How many doors should be opened? How many forklifts should be assigned at each door? Liu and Takakuwa [17] use Arena to test an inbound truck schedule and an employees’ schedule in a crossdock; the schedules are adjusted manually if the results of the simulation indicate the schedule is not satisfactory.

There are fewer papers that combine optimization models with simulation in the crossdock literature; we identify four different ways of combining simulation and optimization models:

1. The output of the optimization is used as input to a simulation model.
   Gambardella et al. [7] are, to the best of our knowledge, the only ones applying this technique in the logistics platform environment. They develop a discrete-event simulation model of an intermodal container terminal in order to check the validity of a resource allocation within the terminal, that is generated with an integer linear program. This work, carried out in 1998, relies on a custom-coded simulation program lacking the numerous functions of modern simulation software programs.

2. The output of a simulation is used as input to an optimization model.
   For instance, Hauser [9] in her dissertation uses a simulation (developed with Arena) of a Toyota manufacturing plant to test different crossdock layouts. The objective is to minimize the walking distance during the dispatching operation, with the idea of eventually balancing the workload. Genetic algorithm are used to decide where each destination goes in the best layout determined by the simulation. Another example is given by Liu and Takakuwa [16], who use a simulation model developed in Arena to determine the workload at a cross-docking center. Data from the simulation are then used as input in an IP model that produces an optimal schedule for the operators.

3. An optimization model is embedded within a simulation model.
   Wang and Regan [24] apply this technique with two time-based algorithms to schedule, in real-time, inbound trucks in a crossdock, with the aim of minimizing pallet transfer time. The algorithms are embedded in an Arena simulation model.

4. A simulation model is embedded within an optimization model.
   This method, often called simulation-optimization, is widely used in diverse fields. In the cross-docking literature, McWilliams [19] generates an inbound truck schedule using this technique. A simulation model is used to evaluate the objective function after each permutation of the meta-heuristics. In a similar way, Aickelin and Adewunmi [1] solve the crossdock truck-to-door assignment problem with a local search; a simulation model evaluates the objective function used in the memetic algorithm.

Our goal in this paper is to fill the gap left in case 1. In the case of cross-docking operations, we demonstrate the use of a simulation model to evaluate the robustness of a solution provided by an IP model.

We use the term robustness following the definition given by Sabuncuoglu and Goren [21]: “a schedule whose performance does not deteriorate much in the face of disruption is called robust.” Specifically, we choose among the many robustness measures proposed by these authors in order to measure the deviation of the realized schedule’s performance from the initial deterministic performance.

The robustness literature gives several examples of robustness evaluation through simulation. Leon et al. [14] propose (slack-based) robustness measures, and evaluate them with a simulation study. Valckenaers et al. [22] review simulation-based studies that analyze scheduling problems, especially rescheduling techniques (repairing the schedule after an unexpected event occurred). They propose a method to evaluate the different rescheduling method. Canon and Jeannot [5] compare different robustness metrics used in the literature, by performing an experimental study and showing how the different metrics relate to each other. Hazır et al. [10] propose a number of slack-based measures for robust project scheduling. They use a Monte-Carlo simulation to see which of these performance measures have the highest correlation with indicators on the project punctuality (percentage of projects ending before a deadline and average delay in the project completion time). Those different papers deal with project scheduling. Our goal in this paper is to propose a method that is applicable to cross-docking operations.

The paper is organized as follows. Section 2 describes the problem being considered and the assumptions. Section 3 explains the interactions between the optimization and the simulation models. The methodology used to evaluate the robustness of the IP model is developed in section 4 while the numerical experiments are carried out in section 5 along with our proposal of robustness evaluation. Conclusive remarks are given in section 6.
2. Scheduling the cross docking operations

We consider a cross-docking platform that operates as follows (the words in italic are used in the rest of the paper to designate each process). When an inbound truck containing pallets for different destinations arrives, it is docked at an available door (docking), then all the content is emptied on the dock, i.e. on the floor in front of the door (unloading). The truck leaves when it is empty. After all the pallets contained in the truck have been placed on the dock, they are controlled and scanned one by one (scanning). In this paper, scanning is considered to be a part of the unloading process. Once the entire truckload has been controlled, the pallets can start being moved. Forklifts are used to move the pallets (transfer) either to a storage location or directly to the outbound truck assigned to the pallet’s specific destination, if the outbound truck is docked at that time (loading). When an outbound truck is full, it is closed, sealed, and can leave the dock.

A few days or weeks in advance, the transport providers for the inbound and outbound trucks indicate at what time they wish to arrive and leave the crossdock. If their trucks serve different clients, this reservation process enables them to organize their rounds. The manager considers the inbound and outbound trucks reservations and creates a feasible schedule; this is called the truck scheduling process. The manager then contacts the transport providers to give them their final appointment times.

Ladier and Alpan [12] propose a model to handle truck scheduling when the transport providers make reservations. The integer program schedules the inbound and outbound trucks, maximizing the transportation providers’ satisfaction and minimizing the total quantity put into storage. One main assumption is that the unloading, scanning, transferring, loading and leaving operations are all done within the same time period, long enough to ensure that the pallets can be processed in masked time. The distance of the transfer (thus the location of the doors) is not taken into account. More details and justifications can be found in Ladier and Alpan [12].

3. Linking optimization and simulation model

The relationships between the simulation and optimization models are shown in Figure 1. The outputs of the IP model are the truck schedules (arrival and departure times for the inbound and the outbound trucks) and the detailed pallet moves (number of pallets moved from one point to another at each time period).

The simulation model takes as input the truck arrival times that are determined by the IP model. We assume that the manager has called the transport providers in order to set up their arrival time according to the IP model results. However, we do not force the truck departure times – the inbound trucks leave when they are empty, and the outbound trucks leave when they are full.

Figure 1: Interactions between the IP model and the simulation model (example with 3 inbound and 3 outbound doors)
For the simulation model to be able to react to a planning change, the pallet moves have to stay flexible. If each pallet was required to move only at the time and to the location decided by the IP model, the simulation would be totally blocked when a truck is late, or operators would stay idle in front of an early truck. Therefore, instead of using the exact pallet move times determined by the IP model, the simulation uses a greedy algorithm to decide which pallet is pulled from which inbound dock and sent to which outbound dock. The data on the pallet moves determined by the IP model, when aggregated, tell the manager how many pallets are moved per hour, and therefore what staffing levels are needed for the transfer each hour. We assume that the manager has staffed the platform accordingly and therefore use the output of the IP model to limit the hourly capacity of pallet transfer in the simulation model.

The simulation model is developed using FlexSim\(^1\). It is validated by ensuring that, when the simulation model is entirely deterministic, the IP model and the simulation model behave in similar ways. The possible causes of deviations between the IP model and the simulation model, and suggestions for dealing with the differences, are explained in Ladier et al. \(^{13}\).

### 4. Robustness evaluation: methodology

The goal of this paper is to evaluate the robustness of schedules obtained with an IP model that uses deterministic input data to generate truck schedules. The IP model also has some restrictive assumptions on transfer times, which may experience uncertainty in the actual operating environment. We want to see how perturbed the system becomes when subjected to stochastic events. We consider three possible sources of variability:

- **Time needed to complete the transfer** of a pallet due to the performance of the workers doing the transfer, the traveling distance, or the congestion of the platform.
- **Time needed to unload** a pallet due to the way trucks are loaded, number of workers working on the same truck, and skills of the workers. Both activities, transfer and unloading, are not explicitly taken into account in the IP model: they are assumed to be performed in masked time. Thus, it is interesting to see how sensitive the schedule is to variations in process times.
- **Truck arrival times** due to, for example, traffic or weather conditions.

This section details the methodology used to test the different cases. Section 4.1 explains how variability is modeled. Section 4.2 defines the indicators that we use to measure the perturbations, and section 4.3 describes the instances that are used in the tests.

### 4.1 Modeling variability

In the simulation model, the transfer and the unloading times are modeled with triangular distributions. Such a distribution can be used when limited data about a process is available \(^{(11)}\). It also has the advantage of being bounded (which is not the case of the normal or exponential distributions). A triangular distribution is defined by its location parameters \(a\) (minimum value), \(b\) (maximum value) and \(m\) (mode). We use only symmetric triangle distributions, for which the mode \(m\) equals the mean \(\mu\). Having a symmetric distribution eliminates the bias a skewed distribution could introduce. This simplifying assumption is not contradictory with our industrial data.

We want to test the behavior of the system when the variability of the transfer time increases; therefore, we want to increase the coefficient of variation while keeping a constant skewness (equal to 0 since the distribution is symmetric) and a constant mean. Using the table proposed by Jannat and Greenwood \(^{11}\), we calculate the values of \(a\) and \(b\) when the coefficient of variation \(c_v\) increases. Since the distribution represents a process time, we keep only the cases when \(a > 0\). The parameters of the resulting triangular distributions are shown in Table 1.

<table>
<thead>
<tr>
<th></th>
<th>Standard process time</th>
<th>Experimental values</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(c_v = 0.1)</td>
<td>(c_v = 0.2)</td>
</tr>
<tr>
<td><strong>Unloading</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td>3.5</td>
<td>3.10</td>
</tr>
<tr>
<td>(b)</td>
<td>4.7</td>
<td>5.10</td>
</tr>
<tr>
<td><strong>Transfer</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td>2.8</td>
<td>2.67</td>
</tr>
<tr>
<td>(b)</td>
<td>4.3</td>
<td>4.39</td>
</tr>
</tbody>
</table>

\(^1http://www.flexsim.com\)
The means chosen for the distributions are obtained from the values of \(a\) and \(b\) detailed in the “Standard process time” column of Table 1. Those values are determined using the classic crossdock sizes given by Bartholdi and Gue [3], and standard process times for logistic operations [8]. These values are close to the case \(c_v = 0.1\), which is thus the more realistic.

To test the effect of variability in the truck arrival times, we apply a random deviation \(d\) to each scheduled truck arrival time \(t_0\), after the data are imported from the IP model. The deviations represent early or late arrivals, or on time, if the deviation is zero. Since most deviations are very short (a few minutes) and large deviations occur only occasionally, we assume the arrival deviations \(d\) are exponentially distributed (with mean \(\delta\)). In order to avoid unrealistically large time deviations, we truncate the distribution such that no value can be greater than \(10 \times \delta\). We also define a multiplier \(\sigma\) that take the values: -1 for an early arrival, +1 for a late arrival, and 0 for an on-time arrival. The probability mass function for \(\sigma\) is specified, as \(P(\sigma = +1), P(\sigma = -1)\) and \(P(\sigma = 0)\). Therefore, for each truck arrival, its simulated arrival time is \(t_a = \max(0; t_0 + d \times \sigma)\) where \(d\) and \(\sigma\) are random samples from their respective distributions.

### 4.2 Measuring perturbations

We want to check how perturbed the system is when we apply different degrees of variability to the transfer times, unloading times and truck arrival times. Our goal is to measure the deviation between the performance of the realized schedule, and the initial deterministic performance (Sabuncuoglu and Goren [21]). We therefore use the following measurement indicators:

- **Total number of pallets in stock** \(I_1\)
- **Error in docking time** (inbound \(I_2\) and outbound \(I_3\)): for each truck which docks later than expected, we compute the absolute difference between the scheduled docking time and the time in which the truck actually docks, in minutes. The indicator is the sum of those deviations for all the inbound/outbound trucks.
- **Error in staying time** (inbound \(I_4\) and outbound \(I_5\)): for each truck which stays docked longer than expected, we compute the absolute difference between the scheduled time spent at the dock, and the actual time spent at the dock by the truck, in minutes. The indicator is the sum of those deviations for all the inbound/outbound trucks.

The time-related indicators \(I_2\) to \(I_5\) are only considered for the trucks that arrive and/or leave later than planned. In order to keep the number of indicators to follow reasonable, and because it does not threaten the applicability of our schedule, earliness is not explicitly taken into account. Since the trucks arriving early can impact the stock level, earliness is implicitly included in \(I_1\).

### 4.3 Choosing the instances

An instance is a schedule for a certain size facility, time duration and number of trucks.

The parameters of our instances are described in Table 2. \(M\) indicates the maximum number of pallets transferred at each time interval. Since a standard truck can carry 33 European pallets, we assume for all our tests that the inbound and outbound trucks all have a capacity of 33 pallets. The instances are generated such that all the inbound trucks have exactly 33 pallets, and that the total amount of incoming pallets per destination is a multiple of 33.

For the set of instances with 3 inbound and 3 outbound doors, that we call \(set_{3+3}\), the schedule is obtained with the IP model. Since the size of the instances with 25 inbound and 25 outbound doors (\(set_{25+25}\)) is too large to be handled.
by the integer program used previously, the truck schedules are calculated using the heuristic H1, as detailed in Ladier and Alpan [12]. This heuristic has been chosen because it is faster and better for larger instances according to [12].

4.4 Simulation parameters
The simulation is run until the platform is empty and all the trucks have left. This occurs after about 10 simulation hours due to the structure of the instances, but in some cases the operations are delayed and the operations finish later. Each of the 21 instances detailed in Table 2 is tested over a number of scenarios. Each scenario (set of different values for the experiment parameters) is tested over 20 different replications (which provides sufficient precision for analysis). For each replication, the value of each indicator $I_i$ is compared to the value $d_i$ of the deterministic case. We check whether or not this value is in the interval $d_i \pm \varepsilon_i$, where $\varepsilon_i$ represents an acceptable tolerance for indicator $I_i$. $\varepsilon_1$ is a number of pallets, $\varepsilon_2$ to $\varepsilon_5$ are in minutes. For each scenario, we calculate the average value of the performance measures $I_i$ over the 20 replications and the 21 instances. The percentage of replications off-limits obtained depends on $\varepsilon_i$.

Note that in general, the platform manager knows the tolerances $\varepsilon$ of his/her organization. For example, some companies give financial penalties to their transport providers if they are more than 15 minutes late; implicitly, the company assumes it can absorb delays smaller than 15 minutes, but not larger. In next section, we use the simulation model to estimate $\varepsilon$ in different cases, and propose robustness indicators linked to this tolerance.

5. Robustness evaluation: results
Following the methodology detailed in section 4, we use the simulation model to see how the schedules obtained with the IP model react when submitted to variability (in transfer times, unloading times and truck arrival times). The goal is to propose, based on the numerical experiments, a robustness indicator for each cause of variability studied.

5.1 Variability in transfer time
In this section, unloading time is equal to zero; transfer times are stochastic as detailed in section 4.1. We carry out three different types of analysis: a detailed analysis for all the indicators defined in section 4.2, an aggregated analysis over the temporal indicators, and finally a correlation analysis to see if the different indicators are linked together.

Analysis per indicator
Figure 2 shows the value of the percentage off-limits for each indicator $I_i$, for two different sets of values for $\varepsilon_i$, for the set of instances set3+3. The results are similar on set25+25.

We observe that a higher coefficient of variation leads to a higher percentage of results off-limits (increasing curves), although the different indicators are less sensitive to changes in the coefficient of variation in the case of set25+25. The total number of pallets in storage and the error in stay time for the inbound side are the most sensitive indicators when the transfer times become more variable.

The shapes of the curves for the different indicators are similar; therefore, for the sake of readability, in the remaining we aggregate the temporal indicators $I_2$ to $I_5$ together.

Aggregated analysis
Figure 3 shows how the average percentage off-limits (average of $I_2$ to $I_5$, with 20 replications for each instance) varies with different values of the tolerances, set such that $\varepsilon_2 = \varepsilon_3 = \varepsilon_4 = \varepsilon_5 = \varepsilon$. 
For \( \text{set}_3+3 \) and for a coefficient of variation \( c_v = 0.1 \), the deviation drops to zero for \( \epsilon \geq 60 \) minutes. The percentage off-limits is very sensitive for tolerances smaller than 15 minutes, and almost insensitive when the tolerances are greater than 30 minutes. Instances of \( \text{set}_25+25 \) are less sensitive than those of \( \text{set}_3+3 \); this is due to their structure. Having a great number of doors provides more flexibility: when a pallet is unloaded it is more likely that a corresponding truck is available, even when the system is perturbed. A similar behavior is observed for indicator \( I_1 \). Since the curves are continuous and monotonous, we propose the following measure for the robustness due to variability in transfer time:

\[
R_{\text{transfer}} = \text{tolerance } \epsilon (\text{in min}) \text{ to get 10} \% \text{ off limits when } c_v = 0.1
\]

The tolerance \( \epsilon \) is the average of \( \epsilon_2, \epsilon_3, \epsilon_4 \) and \( \epsilon_5 \) defined in section 4.4. This robustness measure, being a single numerical value, is easier to exploit than a full set of data as represented in Figure 3. It can be used to quantify how different IP models are able to absorb variations in transfer time.

Table 3 separates the indicators related to inbound and outbound trucks of figure 3b. We observe that the propagation is not the same on the inbound and outbound sides: the outbound indicators are less sensitive. This is because, in the simulation, the transfer algorithm favors the outbound side, which is thus more robust to perturbations.

### Table 3: Percentage off-limits, separating the inbound- and outbound-related indicators (\( \text{set}_3+3 / \text{set}_25+25 \))

<table>
<thead>
<tr>
<th>Tolerance</th>
<th>( c_v = 0.1 )</th>
<th>( c_v = 0.4 )</th>
<th>( c_v = 0.1 )</th>
<th>( c_v = 0.4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>65% / 27%</td>
<td>71% / 27%</td>
<td>26% / 10%</td>
<td>27% / 9%</td>
</tr>
<tr>
<td>5</td>
<td>17% / 3%</td>
<td>44% / 8%</td>
<td>8% / 0%</td>
<td>21% / 0%</td>
</tr>
<tr>
<td>10</td>
<td>4% / 0%</td>
<td>31% / 1%</td>
<td>2% / 0%</td>
<td>14% / 0%</td>
</tr>
<tr>
<td>15</td>
<td>2% / 0%</td>
<td>23% / 0%</td>
<td>2% / 0%</td>
<td>12% / 0%</td>
</tr>
</tbody>
</table>

**Correlation analysis**

Instances from \( \text{set}_25+25 \) have no errors in docking time on the inbound side; there are enough doors to accommodate all the inbound trucks, even when the transfer time is long. For instances from \( \text{set}_3+3 \), a correlation of coefficient \( r \geq 0.75 \) exists between the error in docking time and the error in staying time in the inbound side (indicators \( I_2 \) and \( I_4 \)). Note that an error in docking time occurs when a truck cannot dock at the scheduled time, because the dock is occupied by another truck. This happens when the previous truck stayed docked for too long. When the correlation is 1, the total error in docking time is explained by the total error in stay time. It means that all the trucks which stayed longer than planned are considered “critical” tasks in the planning; i.e., a set of inbound trucks that stayed \( x \) minutes longer caused trucks that followed to dock \( x \) minutes late, exactly. When the correlation is zero, there may be error in the stay time of some inbound trucks, but they do not cause any error in the docking time; i.e., those trucks are considered not critical. When the correlation is between 0 and 1, the situation is mixed; i.e., among the truck staying longer than planned, some are critical and some are not. A solution to improve the robustness of the IP model would thus be to find ways to minimize the number of critical trucks.

**5.2 Variability in unloading time**

In this set of experiments, the transfer time is deterministic and set equal to 3.5 minutes. The unloading time is stochastic and follows the triangular distributions described in section 4.1. Results show a pattern similar to the one in
Figure 3: a higher coefficient of variation implies a higher percentage of cases off-limits, for all of the tolerance values. Similarly to what is done in Section 5.1, this property enables us to propose an indicator to measure the robustness of the model to the variability in unloading time:

\[ R_{\text{unloading}} = \text{tolerance } \varepsilon \text{ (in min) to get 10\% off limits when } c_{v_{\text{unloading}}} = 0.1 \]

5.3 Variability in truck arrival time

We vary the percentages of trucks arriving late, early, and on time such that the total is 100%. We observe the percentage off-limits (aggregated over \( I_2 \) to \( I_5 \)) as a function of the tolerance \( \varepsilon \), with different values of \( \delta \). An example of result obtained when 40% of trucks arrive late, 27% are early, 33% unchanged) is displayed in Figure 4.

![Figure 4](image)

Figure 4: Percentage off limit for the aggregated indicators \( I_2 \) to \( I_5 \) as a function of the tolerance interval \( \varepsilon \) for the truck arrival time variation when \( P(\sigma = -1) = 27\% \), \( P(\sigma = 0) = 33\% \), \( P(\sigma = +1) = 40\% \)

Again, the curves’ patterns are similar to the observations done in sections 5.1 and 5.2: it is possible to deduce a similar indicator from these measurements. But the proportion of trucks arriving early, late and on time is a new parameter compared to is done in the previous section. Figure 5 studies the effect on the tolerance to be set in order to get 10% off-limits for different values of \( \delta \), when 33% of the trucks arrive on time, and the other 67% are either delayed and early with varying proportions.

![Figure 5](image)

Figure 5: Tolerance to get 10% off-limits function of the proportion of early/delayed trucks, for different values of \( \delta \).

We note that the curves in Figure 5 are rather flat, which shows that early arrivals tend to compensate delayed ones. Tolerance \( \varepsilon \) is not very sensitive in that case: for example in set 3+3, when the delays follow an exponential distribution of mean \( \delta = 5 \) min, the tolerance to get 10% off-limit is always around 50 minutes, no matter what the proportion of delayed/early trucks is.

In order to get more variations and thus a monotonous curve from which an indicator can be derived, we vary the percentages of trucks arriving late \( P(\sigma = 1) \), or the percentage of trucks arriving early \( P(\sigma = -1) \), but not both at the same time – the rest are trucks arriving on time. Figure 6 shows the tolerance to be set in that case, in order to get 10% off-limits for different values of \( \delta \). For instance, if 20% of the trucks are delayed and the truck delays follow an exponential distribution of mean \( \delta = 10 \) minutes, the tolerance has to be set to 50 minutes.
Figure 6: Tolerance to get 10% off-limits (aggregating $I_2$ to $I_5$) as a function of the proportions of early/late truck, for different values of $\delta$ (set3+3).

Note also on Figure 6 that a given percentage of early trucks creates a smaller perturbation than the same percentage of late trucks. This confirms the intuitive idea that delayed trucks are “worse” than early trucks – early trucks can wait, while the delay of a truck arriving late can be difficult to compensate. Our indicator thus focuses on delayed trucks.

The curves are all strictly increasing when the percentage of trucks early or late increase. Since the curves are continuous and monotonous, we can use the tolerance needed to get 10% off-limits as an indicator for the robustness due to variability in arrival time:

$$R_{\text{arrival}} = \text{tolerance } \varepsilon \text{ (in min) to get 10\% off limits when } \begin{cases} P(\sigma = +1) = 20\% \\ P(\sigma = 0) = 80\% \\ P(\sigma = -1) = 0\% \end{cases} \text{ and } \delta = 10 \text{ min}$$

6. Conclusion

This paper demonstrates how a discrete-event simulation model can be used to assess the robustness of an IP-based optimization model. In this case, the application is the scheduling of trucks and pallet flows at cross-docking platforms. This joint use of optimization and simulation is not common in the logistics field.

We propose a method and a set of measures to evaluate the robustness of an IP model when truck arrival times vary. The simulation model enables the testing of many schedules under stochastic conditions. The tolerance needed to get 10% off-limit is proposed as an indicator to test the robustness of future optimization models; the values determined here will be used as reference for future comparisons.

The perspective for this work is to develop new integer programs that are more robust (for instance by minimizing the number of critical trucks as suggested in section 5.1), and to compare their performances using the proposed robustness measures. Another perspective would be to use the robustness indicators developed in order to design a simulation-optimization approach.

The tolerances introduced for the indicators reflect an industrial reality. The insights gathered can support the managers in setting the tolerances by showing the effect of a tolerance change on the overall performance.

Acknowledgements

This work has been supported by the Rhône-Alpes Region.

G-SCOP is partner of the LabEx PERSYVAL-Lab (ANR-11-LABX-0025).

References


Ladier, Alpan and Greenwood


