

Using simulation to evaluate how multi-agent transportation planners cope with truck breakdowns

Tamás Máhr

Delft University of Technology,
PO Box 5031, 2600 GA Delft,
The Netherlands,
tamas.mahr@gmail.com

F. Jordan Srour

The American University of Beirut,
P.O. Box 11-0236, Riad El Solh, Beirut 1107 2020,
Lebanon,
fjsrour@gmail.com

Mathijs de Weerd

Delft University of Technology,
PO Box 5031, 2600 GA Delft,
The Netherlands,
M.M.deWeerd@tudelft.nl

Abstract—In most real-world settings, a transportation plan requires modifications during execution. A thorough evaluation of transportation planning methods thus requires testing and comparison in a dynamic environment. We give conditions on a simulation environment that follow from this requirement, and propose a multi-agent simulator meeting these conditions. In addition, we propose a new measure that captures robustness in such dynamic settings. The multi-agent simulator and the robustness measure are then used to compare three different transportation methods (two multi-agent planners and one online optimization approach) in settings with release time uncertainty and truck breakdown incidents.

I. INTRODUCTION

In order for multi-agent transportation planning methods to gain ground, it is necessary to compare the agent-based planning results to other well known algorithms on realistic benchmark instances. When considering instances in which all relevant planning information is known in advance, the evaluation procedure is simple: each instance is given to each planning method, which in turn computes a plan, and the resulting plans are then evaluated using a predefined cost measure. In contrast, in dynamic or uncertain instances, when new information is revealed to the planning method only after an initial plan has been produced, the evaluation is not so straightforward. Instead of evaluating only a single plan, one must examine a series of plan versions, produced in response to the dynamic events. In this paper, we propose an evaluation method that does not directly measure the produced plans, but rather the simulated traces of the trucks executing the plans.

The goal of this paper is tripartite. First, we propose a framework, based on the traces of truck movements, allowing for the comparison of fundamentally different planning approaches on dynamic data. Second, we introduce a new measure of robustness to capture the merits of the planning approaches in dynamic settings relative to static settings. Third, we demonstrate the value of our framework and robustness measure by comparing distributed agent-based and centralized optimization-based planning systems for a transportation problem with uncertain release times and truck breakdowns.

II. RELATED WORK

Biennially, *OR/MS Today*, the membership publication of the Institute for Operations Research and Management Science, publishes a review of vehicle routing software. The main attraction of this review is a series of charts directly comparing the features of each package. In 2010, 15 out of the 23 surveyed packages provide real-time re-routing capabilities, while 22 of the 23 provide daily route planning support; all 23 packages provide route plan analysis [1]. What is obscured by this chart is how these route planning differences translate into real-world plan performance. To ascertain that, one must dig deeper into the literature.

Doing this, one finds that when considering on-line, real-time, or dynamic vehicle routing problems, three planning mechanisms tend to emerge: on-line optimization methods (reactive methods, dependent on mathematical programming techniques, working on a rolling-horizon framework), stochastic programming or look-ahead methods (methods dependent on the incorporation of forecasting likely instance characteristics), and heuristic or metaheuristic methods (methods that try to exploit problem structure for fast computation times) [2], [3]. The fact that these three distinct areas are so readily discernable belies the reality that most planning mechanism evaluations are performed in an inter-domain rather than intra-domain manner. Comparing heuristic/metaheuristic approaches to on-line optimization is, however, necessary if these approaches, such as multi-agent systems, are to enter the mainstream.

On-line optimization techniques for dynamic vehicle routing problems work by breaking the full time-span of the instance into multiple planning horizons or decision epochs. Then at each epoch the existing plan is either revised by heuristic methods or fully re-optimized via the use of a mathematical program. In this domain, [4] demonstrate the superiority of an exact mixed integer programming formulation of the PDPTW. They compare their reoptimization approaches to three heuristic approaches (a simple round robin assignment, an insertion heuristic, and a reordering approach). This comparison reveals that the reoptimization approaches systematically outperform the heuristic approaches by about 10% in terms of cost. This line of work is continued by [5] by examining a hybrid rule-based heuristic and optimization

approach. These results are interesting as they utilize empty-travel and cost related metrics recorded for each truck at each decision epoch to compare planning approaches across two methodologies — on-line optimization and heuristics. In many ways our work is an extension of this paper, with one major difference, the heuristics they considered were designed to make decisions similar to the optimization approach, but faster and at a (possibly) lower quality. Our heuristic approaches are agent-based, with the agents representing a structural decomposition of the vehicle routing problem into trucks and jobs.

Given this very different heuristic mechanism, we can no longer compare empty-travel and cost metrics recorded for each truck only at decision epochs as in [4]. We must now consider full traces of truck activity across the full time range of the instance in order to make a fair comparison of planning methodologies. The idea of comparing the spatial-temporal traces of vehicles to ascertain the quality of a routing policy is not new in the field of transportation [6]. It is, however, more commonly used when considering personal vehicle routing as opposed to freight vehicle routing. While the truck trace provides a powerful tool for deriving new metrics with which to compare plans, the simulator responsible for executing those plans should also be carefully designed.

The simulation framework presented in this paper is an extension of an earlier on-line optimization and agent-based comparison [7]. In that work, the researchers recognized the need for a time-based simulation, but the two planning approaches were coded to run on separate simulators. While the use of distinct time-based simulators had no adverse impact on the results, it did have a detrimental impact on the ability to easily compare additional planning methods. Given this realization, we continue this article with a description of the requirements for any planning comparison framework.

III. REQUIREMENTS FOR THE TESTBED

To provide a fair comparison of transportation planning algorithms, any framework for comparison must accommodate certain requirements. We document the most essential of these requirements in the following three subsections.

A. Capability to consider unforeseen events

Our main goal is to evaluate transportation planning methods for problem instances with uncertainty. This means that in any instance, at the start of the planning horizon, there are events that must remain unknown to all planning methods until a specified point in time. For example, these dynamic or uncertain events may be defined in the instance data deterministically (e.g. at 3pm truck X breaks down completely), but the planning methods, should not know about these events before their occurrence (e.g. before 3pm). Thus, the simulation must be structured to ensure that from the point of view of the planning method, the problem instances contain uncertainty.

B. Capability to accommodate different computation times

Different transportation planning methods have different run-time requirements. When solving dynamic transportation

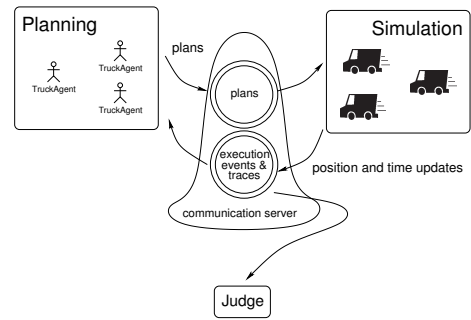


Fig. 1. Evaluating a planner by measuring execution of simulated trucks

problems, the run-time of a planning method can severely influence the speed at which unforeseen changes can be dealt with. Algorithms with a long run time can potentially miss their chance to respond adequately to a given incident. To be able to make realistic conclusions about the behavior of planning systems in the real-world, it is thus essential to simulate the effect of long run times. Therefore, we propose not using event-based simulations, because in such simulations there is usually no direct relation between computation time and simulated time, since simulated time is usually moved forward only after all computation related to an event has been completed.

C. Capability to separate planning and execution

To ensure a fair comparison of transportation planning algorithms, an evaluation method should use the same component to generate execution traces for each planning algorithm. This suggests that the planning and execution components should be separated (code-wise). This structure gives researchers flexibility by allowing reuse of the execution component (truck movement simulator) for each planning method. In turn, this lowers the development effort required to conduct the experiments and prevents drawing false conclusions from anomalies caused by different programming errors in the different execution codes.

To summarize, being able to include dynamic events during execution, explicitly consider the computation time needed by the planning methods, and reuse the same plan-execution component are, in our opinion, the essential properties of a fair evaluation method for transportation planning methods. In the following, we briefly describe the comparison framework that was developed to meet these requirements and obtain the results presented in this paper.

IV. SIMULATION TESTBED

Our framework for comparing different planning methods consists of a planning component, a simulation component, a communication server, and a judge. We begin by describing the planning component, which implements the planning methods under evaluation, and the simulation component, which produces the execution traces. In Figure 1, the planning component consists of agents cooperating to produce transportation plans. In reality, the planning component can use any type of internal plan generating mechanism (e.g.

agents, mathematical programming, hybrid systems). The only requirement of the planning component is that it must compute plans, and it must send those plans to the simulation component via the communication server.

The simulation component executes those plans by simulating trucks driving between the job locations as dictated by the plans. It is important that the location coordinates are interpreted in the same way by the planning component and the simulation component. In principle, the coordinates could be numbers relative to a planar coordinate system and the travel distances could be based on an Euclidean travel metric. Alternatively, the coordinates can refer to geographical coordinates (i.e. latitude and longitude) and the travel distances can be derived from the actual road network. In our realization, we used the latter mechanism.

In addition to these spatial considerations, the planning and simulation components should also have the same sense of time, as discussed in the previous section. This is important, because the simulation executes truck movements, and thereby generates traces in parallel with the planning component that calculates new plans. As a consequence, the evaluation framework does not assume that either the computation of new plans or the communication of those plans are instantaneous. The longer the computation and communication of the new plans take, the more time is spent with the simulated trucks executing outdated plans.

The planning and simulation components communicate via the communication server, therefore monitoring the messages is trivial; it only requires listening to the communication server. The execution events and traces circle in Figure 1 represents a component saving all messages, allowing several measurements of a given planning method. The sequence of position update messages from the simulation component, for example, reveal the simulation traces and provide the ultimate means for evaluation. This saved execution log is subsequently processed by our last component, the judge.

The judge can perform various computations on the saved logs to derive measurements of the execution. For a given execution log and problem instance, the judge can compute, for example, the time (or distance) spent traveling empty by the trucks, the time spent idle by the trucks, and the aggregated time of lateness in arriving to delivery locations. The power of the judge rests in its ability to accommodate a host of performance measures as required for each experiment. One common evaluation criterion is the measure corresponding to the objective function explicitly used by the planning method — the judge can easily accommodate this need.

To demonstrate the power of this framework for comparing transportation planning methods in dynamic settings, we present the results of a comparison made between a multi-agent planning system and an on-line optimization planning system. The dynamic events that occur during execution are the revelation of job release times and truck breakdowns.

V. EXPERIMENT SETUP

In the following subsections, we first present the problem instances that were used in our experiments. We then define

two measures — a traditional cost measure and an alternative, so called *robustness* measure. These form the basis of our comparison between the transportation planning methods described at the end of this section.

A. Problem instances

The problem instances used in this study are derived from the real-world data published in [8]. There, a set of 33 instances were defined as static instances (denoted as R_0). Each static instance contains 65 truckload (container) orders and 40 trucks. Container orders are defined by three locations (pick up, delivery and return) and three time windows, one for each location. In the static instances, every order is available for transport at the beginning of the day, 6am, and must be transported before 6pm. To add uncertainty to these static instances, dynamic release times were added to different subsets of the orders; thus creating release-time uncertainty instances (R_x). These orders were then, no longer available at the beginning of the day, but rather at a point in time dictated by their release-time. In total, four release-time uncertainty scenarios were defined based on the percentage of dynamic orders in the instances. These were the R_{25} , R_{50} , R_{75} , and R_{100} scenarios, where respectively 25%, 50%, 75%, and 100% of the jobs were dynamic.

In this paper, we extend the previously defined uncertainty scenarios with a new source of uncertainty: truck breakdowns. To model truck-breakdown uncertainty, we assigned breakdown times to randomly selected trucks in the static (R_0) instances. In the instances, the points in time at which trucks breakdown were randomly generated according to a uniform distribution over the full day (6am to 6pm). With this method, we prepared three different truck-breakdown scenarios, with three (B_3), five (B_5), and ten (B_{10}) trucks breaking down each day.

Truck-breakdown uncertainty is a natural extension to the previously defined release-time uncertainty. While in release-time uncertainty cases the planners have to solve problems with an increasing number of jobs, here the change manifests itself in a decreasing number of resources that can be used to deliver a constant number of containers. Comparing the performance of plans with such uncertain instances requires unique evaluation criteria and methods — the topic of the next subsection.

B. Evaluation criteria

We define two evaluation criteria that can be computed from the execution traces. The first is a classical cost measure that is used (partly) as a minimization criterion by the planning methods. The second is a novel measure characterizing the robustness of the planning methods in the face of uncertainty.

The total cost of transportation for instance i , $c(i)$, is computed from the execution traces as the sum of the time that the trucks were traveling empty ($t_e(i)$), plus the sum of the rejection penalties for all orders that were not transported ($p(i)$), plus the sum of the time the trucks were late as

compared to the designated time-windows ($t_l(i)$):

$$c(i) = t_e(i) + p(i) + t_l(i). \quad (1)$$

As all elements of the total cost are in the time domain, the cost is defined in seconds.

As stated previously, the sum of the time all trucks in instance i spent traveling empty is denoted by $t_e(i)$. For one truck, empty travel is the time spent traveling from the return of a container to the pick up of the next container. The rejection penalty ($p(i)$) is defined as the sum of the rejection penalties of all non-transported orders in instance i . The rejection penalty of one order is the loaded time of the order, that is the time spent traveling between the pick up and the return of the container. The sum of all lateness that occurred in instance i is denoted by $t_l(i)$. Lateness is defined as the difference of the delivery time and the end time of the customer time window; this takes a non-zero value for an order only if it was delivered after the end of the time window at the customer location.

From the cost of all instances in scenario S , the average cost of the scenario is denoted by $c(S)$. While this cost measure is defined for any scenario, static (R_0) or dynamic (e.g. R_{100} or B_{10}), the robustness measure is only defined for dynamic scenarios. This measure expresses the degradation that the given planning method suffers from the uncertainty present in the instances of scenario S . To appreciate this measure, note that any dynamic instance j was generated from a static instance i by adding uncertainty as defined by scenario S . Accordingly, the robustness of a given planning method for an instance j is defined as the ratio of its total cost for instance i in the static scenario R_0 and the total cost of the corresponding instance j in the dynamic scenario S :

$$r(j) = \min\left(1, \frac{c(i)}{c(j)}\right) \quad (2)$$

In this measure, the $\min()$ function ensures that if the cost of a method for a dynamic instance is lower than in the corresponding static instance, its robustness is maximized at 1. The average robustness on a set of instances S is then denoted by $r(S)$.

These two measures, $c()$ and $r()$ describe two different properties of a method. The cost measure, $c()$, reveals how efficient the method is on the instances of a scenario. The robustness measure, $r()$, tells us how fragile this efficiency is. It defines the (normalized) reduction in performance under uncertainty. Clearly, these measures are strongly related, which is illustrated by the following proposition.

Proposition 1: Given a dynamic instance j and the related static instance i , if method B is more robust than method A on j , then the total costs of B for j are less than A , or the total costs of A for i are less than B (or both).

Proof: To arrive at a contradiction, assume that B is more robust than A on j , the costs of B for j are at least as high as A , and the costs of A for i are at least as high as B . Denote the costs of method X on instance k by $c^X(k)$.

TABLE I

THE RELATION BETWEEN ROBUSTNESS AND COST IN CASES WITH A HIGHER COST FOR DYNAMIC INSTANCES (j)

static instance i	uncertain instance j	robustness
$c^A(i) = c^B(i)$	$c^A(j) = c^B(j)$	$r^A(j) = r^B(j)$
$c^A(i) < c^B(i)$	$c^A(j) < c^B(j)$	any
$c^A(i) \leq c^B(i)$	$c^A(j) > c^B(j)$	$r^A(j) < r^B(j)$
$c^A(i) > c^B(i)$	$c^A(j) > c^B(j)$	any
$c^A(i) > c^B(i)$	$c^A(j) \leq c^B(j)$	$r^A(j) > r^B(j)$

Then this assumption implies that

$$\min\left(\frac{c^B(i)}{c^B(j)}, 1\right) > \min\left(\frac{c^A(i)}{c^A(j)}, 1\right) \quad (3)$$

$$c^B(j) \geq c^A(j), \text{ and} \quad (4)$$

$$c^A(i) \geq c^B(i). \quad (5)$$

Dividing the left hand side of 5 by the right hand side of (4), and the right hand side of (5) by the left hand side of (4), we obtain

$$\frac{c^A(i)}{c^A(j)} \geq \frac{c^B(i)}{c^B(j)}. \quad (6)$$

There are two cases. If $c^B(i) \geq c^B(j)$ then with (5) and (4) $c^A(i) \geq c^A(j)$. Thus both methods have a robustness value of 1, contradicting that B is more robust than A .

If $c^B(i) < c^B(j)$ then $\frac{c^B(i)}{c^B(j)} < 1$ and thus from (3) it follows that $\frac{c^B(i)}{c^B(j)} > \frac{c^A(i)}{c^A(j)}$. This contradicts Equation (6). ■

Proposition 1 highlights the underlying dependency between the cost and the robustness measures for a certain subset of the cases. Table I shows how robustness depends on the relation of the cost measures of the two methods on the *same* instances when both have a higher cost on a dynamic instance j than on the related static instance i (which is most common). When Proposition 1 can be applied, then there is a clear winner. In the other cases, however, robustness depends on the actual cost values of the compared methods.

C. The planning methods

Three planning methods are compared in this paper, two distributed agent-based methods and one centralized optimization. All three methods are fundamentally related to the same algorithms as those published in [7]. The common goal of the methods is to minimize empty travel time plus the rejection penalty, while all three strictly disallow lateness.

The two agent algorithms employ various heuristics to compute and adapt plans in any given situation. Both use the classical insertion heuristic, order exchange and order relocation heuristics. The difference between the two is that one of the agent systems uses a modified insertion heuristic extended to consider order replacement. This allows already committed orders to be released in favor of a new, “better” (i.e less cost) order. The simpler agent algorithm using the original insertion algorithm is called *insertion-relocation-exchange (ire)*, while the one using the modified insertion algorithm is called *substitution-relocation-exchange (sre)*.

TABLE II
PARAMETERS OF THE EXPERIMENTS

Benchmark settings	
number of trucks	40
number of orders	65
pick-up and delivery time	1 hour
return time	half an hour
delivery time-window length	2 hours
time-window start points	uniform from [8am,5pm]
start time	6am
simulation speed	6 times real time
<i>variable release times</i>	
<i>variable number of truck breakdowns</i>	
Parameters of the agent approaches	
reallocation frequency	1 per 1 hour per job
exchange frequency	1 try per 1 hour per truck
Parameters of central optimization	
M value ensures timing constraints	9000
decision horizon	30 seconds real time
(except the first decisions that are unlimited)	

The on-line optimization approach, extensively documented in [7]–[9], is based on the mixed-integer programming model of [4].

A summary of the parameters used to set up the experiments is provided in Table II. The elements typeset in italics were varied throughout the experiments, the rest remained fixed.

VI. SIMULATION RESULTS

The simulation results presented in this section were obtained by utilizing the three transportation planning methods and an off-line (a posteriori) method as part of the proposed evaluation framework running the described problem instances. A portion of these results (the cost measure comparisons of the on-line optimization and the *sre* agent method on the release-time uncertainty scenarios) was already published in [7]. Nevertheless, for the sake of completeness, we discuss the results of all three methods on both uncertainty types in terms of both cost efficiency and robustness.

Fig. ?? repeats the earlier findings of [7] regarding the total cost of the different methods across the range of release-time uncertainty scenarios. While in the static instances the on-line optimization clearly outperforms the agent solutions, there is a point beyond which the advantage of the on-line optimization disappears in the face of uncertainty. When more than 50% of the jobs are dynamic, the agent solutions start to take the lead.

In earlier work, when service time uncertainties were also examined, it appeared that this trend holds across all uncertainty types. However, the truck-breakdown scenarios, introduced in this paper, break this trend. An increase in breakdown events seems to have an equal effect on all three planning methods. The on-line optimization method maintains its advantage against the agent-based methods even when 25% of the trucks breakdown, which can be seen in Figure ??.

The final two figures depict the robustness results of the algorithms on the different scenarios. In the case of release-time uncertainty (Fig. ??), the robustness of the on-line

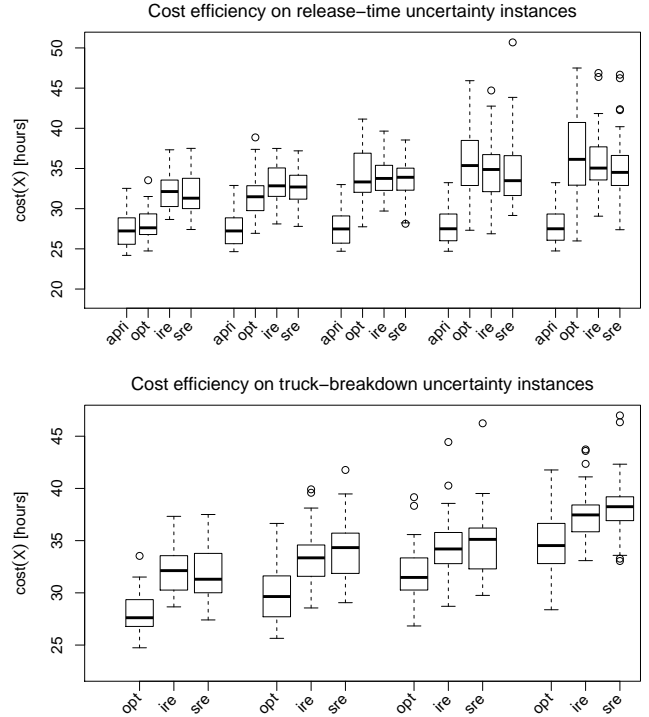


Fig. 2. Cost-measure results on the release-time, and truck-breakdown uncertainty scenarios.

optimization method is consequently lower than that of the agent methods, indicating that on-line optimization is more susceptible to performance loss (in terms of cost) than the agent methods. The most robust method, with a robustness value close to one, is the a posteriori optimal method. The fact that its robustness is less than one means that each instance becomes more difficult as release-time uncertainty increases; even optimal results taken by solving the cases, assuming an end-of-the-day perspective with all relevant information known, are worse than in the static case.

Thus far, the main trend observable in the results is that the method which is better cost-wise, is also better in terms of robustness. However, the truck-breakdown scenarios do not follow this trend (see Fig. ??). Although the on-line optimization has consistently lower transportation-cost results on these scenarios, it still proves to be less robust than the agent methods.

VII. CONCLUSIONS AND FUTURE WORK

In this paper we argued that any evaluation of transportation planning methods should be able to simulate unforeseen events, take care of the consequences of (long) computation times in a realistic manner, and separate planning and execution to be able to compare completely different methods in a consistent way. We provided a detailed description of a comparison framework that meets these requirements through the use of planning, simulation, and judge components.

For static planning approaches, a cost function is usually minimized and the emergent value is used to compare

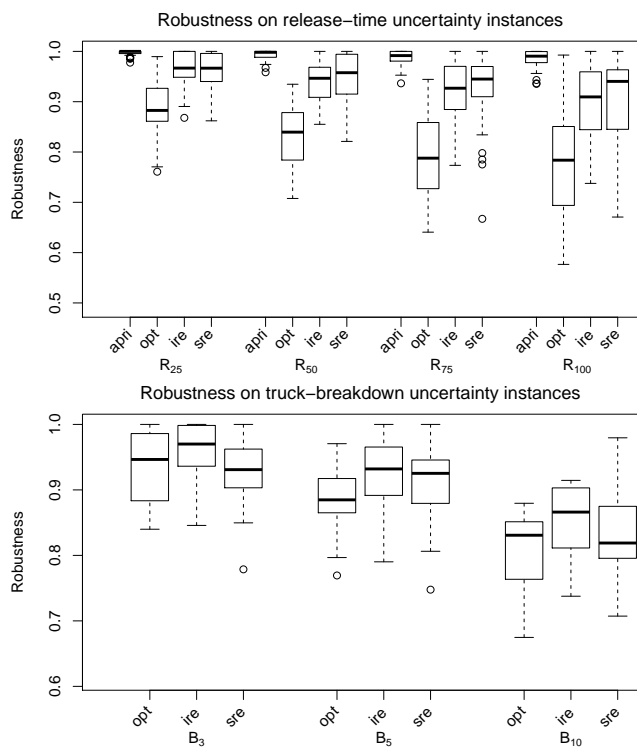


Fig. 3. Robustness-measure results on the release-time, and truck-breakdown uncertainty scenarios.

different methods. However in a dynamic setting, in addition to cost minimization, the effect of unforeseen events on these costs is also an important measure. This paper provided such a measure for robustness.

Both the simulator and the robustness measure are illustrated by comparing three state of the art transportation planning methods: one using a centralized on-line optimization technique and two variants of multi-agent planners. The comparisons are done on a set of known benchmarks with release time uncertainty, and a new set of benchmarks with truck breakdown uncertainty.

From these experiments we conclude that the simulator is able to provide a fair comparison between very different types of planning methods, and that the robustness measure adequately captures the effect of uncertainty on such planning methods. Additionally, the experiments provided a surprising result regarding the effect of random truck breakdowns. In previous work multi-agent systems were shown to be more efficient than the centralized optimization approach under high uncertainty levels; this is not the case when considering random truck breakdowns. In cases where this is the only cause of uncertainty, online optimization significantly outperforms, in terms of cost, both of the two state-of-the-art multi-agent planning approaches. However, these multi-agent approaches appear to be slightly more robust; the relative increase in costs is a bit smaller, probably because of the slack in the initial plans without uncertainty.

The results presented here not only highlight the strengths and weaknesses of three different transportation planning

systems, but they also elucidate the strengths and weaknesses of our proposed simulation framework. Specifically, while the recommended component based structure provides a plug-and-play feel, actually making the planning system-simulator connection is still labor intensive. Any labor intensive process can, in turn, become an error prone process. Future versions of the dynamic simulation framework should be streamlined. One mechanism to streamline the framework is the development of a web-based simulation service.

Aside from ameliorating the simulation framework, there is also future work for our transportation scenarios. Specifically, we plan to consider travel time uncertainty. We seek to model travel time uncertainty that stems not only from road way disturbances, but from the driver's behavior — e.g. the selection of routes other than the geographically shortest route. As an extension of this change in the scenarios, we would also like to consider multi-agent and optimization planning methods that can take the stochastic nature of travel time explicitly into account.

In addition to travel time considerations, transportation planning in the real-world includes other, tacit forms of uncertainty. To capture the value of a planning method in the real-world, comparisons between a dispatcher (human) planned route and those derived from an automated planning system, would be extremely valuable.

We believe that the simulation environment put forward in this paper, as well as the robustness measure presented, as demonstrated on three planning approaches, support performing such future experiments.

REFERENCES

- [1] J. Partyka and R. Hall, "Vehicle routing software survey: On the road to connectivity," *ORMS Today*, vol. 37, no. 1, 2010.
- [2] B. Eksioglu, A. V. Vural, and A. Reisman, "The vehicle routing problem: A taxonomic review," *Computers & Industrial Engineering*, vol. 57, no. 4, pp. 1472 – 1483, 2009.
- [3] A. Larsen, O. B. Madsen, and M. M. Solomon, "Recent developments in dynamic vehicle routing systems," in *The Vehicle Routing Problem: Latest Advances and New Challenges*, ser. Operations Research/Computer Science Interfaces Series, R. Sharda, S. Vo, B. Golden, S. Raghavan, and E. Wasil, Eds. Springer US, 2008, vol. 43, pp. 199–218.
- [4] J. Yang, P. Jaillet, and H. Mahmassani, "Real-time multi-vehicle truckload pick-up and delivery problems," *Transportation Science*, vol. 38, no. 2, pp. 135–148, 2004.
- [5] H. S. Mahmassani, Y.-J. Kim, and P. Jaillet, "Local optimization approaches to solve dynamic commercial fleet management problems," *Transportation Research Record*, vol. 1733, pp. 71–79, 2000.
- [6] C. Sommer and F. Dressler, "Progressing toward realistic mobility models in VANET simulations," *Communications Magazine, IEEE*, vol. 46, no. 11, pp. 132 –137, 2008.
- [7] T. Máhr, J. Srou, M. de Weerd, and R. Zuidwijk, "Can agents measure up? a comparative study of an agent-based and on-line optimization approach for a drayage problem with uncertainty," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 1, pp. 99 – 119, 2010.
- [8] F. J. Srou, T. Máhr, M. M. de Weerd, and R. A. Zuidwijk, "MIPLIB truckload PDPTW instances derived from a real-world drayage case," Erasmus Research Institute of Management (ERIM), Tech. Rep., 2010. [Online]. Available: <http://hdl.handle.net/1765/20883>
- [9] F. J. Srou, "Dissecting drayage: An examination of structure, information, and control in drayage operations," Ph.D. dissertation, Erasmus University Rotterdam, 2010.