Agent Performance in Vehicle Routing when the Only Thing Certain is Uncertainty

Tamas Mahr
Delft Technical University and Almende BV
Rotterdam, The Netherlands
T.Mahr@tudelft.nl

Jordan Srour
RSM Erasmus University
Rotterdam, The Netherlands
JSrour@rsm.nl

Mathijs de Weerdt
Delft Technical University
Delft, The Netherlands
M.M.deWeerdt@tudelft.nl

Rob Zuidwijk
RSM Erasmus University
Rotterdam, The Netherlands
RZuidwijk@rsm.nl

ABSTRACT

While intermodal transport has the potential to introduce efficiency to the transport network, this transport environment also suffers from a lot of uncertainty at the interface of modes. For example, trucks moving containers to and from a port terminal are often uncertain as to when exactly their container will be released from the ship, from the stack, or from customs. This leads to much difficulty and inefficiency in planning a profitable routing for multiple containers in one day.

In this paper, we examine agent-based solutions as a mechanism to handle job arrival uncertainty in the context of a drayage case at the Port of Rotterdam. We compare our agent-based solution approach to a well known on-line optimization approach and study the comparative performance of both systems across four scenarios of varying job arrival uncertainty. We conclude that when less than 50% of all jobs are known at the start of the day then an agent-based approach performs competitively with an on-line optimization approach.

1. INTRODUCTION

Scheduling the routes of trucks to pick-up and deliver containers is a complex problem. In general such Vehicle Routing Problems (VRPs) [19] are known to be NP-complete, and therefore inherently hard and time consuming to solve to optimality. Fortunately, these problems have a structure that can facilitate efficient derivation of feasible (if not optimal) solutions. Specifically, the routes of different trucks are more or less independent. Such "locality" in a problem is a first sign that an agent-based approach may be viable.

Modeling and solving a VRP by coordinating a set of agents can bring a number of advantages over more established approaches in the field of operations research even when using state-of-the-art mixed integer solvers such as CPLEX [7]. Agent advantages include the possibility for distributed computation, the ability to deal with proprietary data from multiple companies, the possibility to react

quickly on local knowledge [5], and the capacity for mixed-initiative planning [1].

In particular, agents have been shown to perform well in uncertain domains. That is, in domains where the problem is continually evolving [5]. In the VRP, for example, a very basic form of uncertainty is that of job arrivals over time. To the best of our knowledge, however, the effect of even this basic level of uncertainty on the performance of agent-based planning in a realistic logistics problem has never been shown.

We think it is safe to assume, based on its long history, that current practice in operations research (OR) outperforms agent-based approaches in settings where all information is known in advance (static settings). However, in situations with a lot of uncertainty, agent-based approaches are expected to outperform these traditional methods [8].

In this paper we investigate whether a distributed agent-based planning approach indeed suffers less from job arrival uncertainty than a centralized optimization-based approach. Our main contribution is to determine at which level of job arrival uncertainty agent-based planning outperforms on-line operations research methods. These results can help transportation companies decide when to adopt an agent-based approach, and when to use an on-line optimization tool, depending on the level of uncertainty job arrivals exhibit in their daily business.

In Section 2 we provide a survey of current work on agent-based approaches to logistics problems. In Section 3 we then introduce the case of a transportation company near the port of Rotterdam. Based on this literature review and the specific nature of our case study VRP, we propose a state-of-the-art agent-based approach where orders are auctioned among trucks in such a way that each order is assigned to the truck that can most efficiently transport the container. Moreover, these trucks continuously negotiate among each other to exchange orders as the routing situation evolves. This agent-based approach is the topic of Section 4. In the following section, Section 5, we describe the centralized on-line optimization approach used in comparison to our distributed agent-based system. The structure of our test problems and the computational results are the topics of Section 6. In our final section we discuss the consequences of our results, summarize our advice to transportation companies, and give a direction for future work.

2. LITERATURE SURVEY

In their frequently cited 1995 paper, Fischer et al. argued that multi-agent models fit the transportation domain particularly well [5]. Their main reasons were that (i) the domain is inherently distributed (trucks, customers, companies etc.); (ii) a distributed agent architecture can cope with multiple dynamic events; (iii) commercial companies may be reluctant to provide proprietary data needed for global optimization and agents can use local information; and (iv) inter-company cooperation can be more easily facilitated by agents. To illustrate the idea, the authors also provided a detailed agent architecture for transportation problems that evolve over time thereby exhibiting uncertainty over time. This architecture makes a distinction between a higher and a lower architectural level. At the higher level, company agents negotiate over transportation requests to eliminate ill-fitting orders. On the lower level, truck agents (clustered per company) participate in simulated market places, where they bid on offered transportation orders. Truck agents use simple insertion heuristics to calculate their costs and use those costs to bid on auctions implementing an extended contract net protocol [17]. Although the heuristics that agents use to make decisions are rather crude, the authors suggested that in dynamic problems (problems with high uncertainty), such methods survive better than sophisticated optimization methods.

Fischer et al.'s bi-level approach recognizes that one shortcoming of a fully distributed system is that agents only have access to local information [5]. The need to balance between the omniscience of a centralized model and the agility of a distributed model, was similarly recognized by Mes et al. [12]. They also introduce a higher level of agents, but with a different role than the high-level agents of Fischer et al. Mes et al.'s two high-level agents (the planner and the customer agent) gather information from and provide information to agents assigned beneath them. The role of the higher level agents is to centralize information essential for the lower level agents to make the right decisions.

Some researchers have gone even further in proposing centralized agent-based models. These researchers focused on centralizing the problem information to be able of make better distributed decisions. In one of the few models that is actually applied in a commercial company, Dorer and Calisty cluster trucks geographically, using one agent per cluster [4]. This way, one agent plans for multiple trucks. They use insertion heuristics to initially assign orders to trucks, and then use cyclic transfers [18] to enhance the solution. In an even more centralized model, Leong and Liu use a fully centralized optimizer to initialize the agents [11]. The agents' role is then to change the plans as events are revealed. The authors analyze the performance of their model on a selection of Solomon benchmark sets, and show that it performs competitively.

As noted previously, however, the move towards centralization can hinder the ability of the agents to react quickly on local information. Given the uncertain environment of our problem, we are interested in the competitiveness of a system with fully distributed agents. One example of a fully distributed agent approach in the transportation domain is that of Brückert et al. They proposed a more detailed (holonic) agent model [1]. They distinguished truck, driver, chassis, and container agents that have to form groups (called holons) to serve orders. Already formed holons use the same techniques to allocate tasks as Fischer et al., but the higher agent level is omitted, since they model only a single company case. The main focus of their research is computer-human cooperative planning, and they do not test their model extensively against other models.

Generally, the decision to use a distributed approach is based on the expectation (included already in the reasons of Fischer et al.) that distributed models handle uncertainty better. The agent architecture in these fully distributed models is completely flat, the models avoid centralizing information, and agents can use only local information when making decisions. Having lost the power of using (partial) global information, distributed agents need other ways to enhance their performance.

In the model of Fischer et al., as well as in the models of many of their followers, agents use simple approximation techniques to make decisions. In the related domain of production planning Persson et al. embed optimization in the agents to improve local decisions [13]. They show that optimizing agents outperform the approximating agents, but they also show that central optimization still outperforms the optimizing, but distributed, agents.

While Persson et al. concentrated on making optimal decisions within agents, there is still a need to coordinate between the distributed agents. For example, in the transport problem context, when orders are assigned to trucks sequentially, at every assignment the truck with the cheapest insertion gets the order. Later, however, it might turn out that it would be cheaper to assign the same order together with newly arrived orders to another truck. From the truck point of view it means that trucks that bid early and win assignments might not be able to bid later on more beneficial (better fitting) orders. This problem is called 'the eager bidder problem' [16], and several researchers proposed alternative techniques to solve it. Kohout and Erol introduce an enhancement process that works between agents [9]. The process mimics a well known enhancement technique called 'swapping' or two-exchange [2]. Kohout and Erol implement this swapping process in a fully distributed way, and show that it yields significant improvement.

Perugini et al. extend Fischer's contract-net protocol to allow trucks to place multiple possibly-conflicting bids for partial routes [14]. These bids are not binding, trucks are requested to commit to them only when one of the bids is accepted by an order agent. Since auctions are not necessarily cleared before other auctions are started, agents have a chance to "change their mind" if the situation changes. This extension helps to overcome the eager bidder problem to some extent and thereby produces better results. Another possible way to tackle the same problem is to use leveled commitment contracts introduced by Sandholm and Lesser [15]. Leveled commitment contracts represent agreements between agents that can be withdrawn. If a truck agent finds a new order that fits better, it can decommit an already committed order and take the new one. Hoen and La Poutré employ truck agents that bid for new orders considering decommitting already assigned ones [6]. They show that decommitment yields more optimal plans in a single-company cooperative case.

Returning to Fischer's reasoning, however, the primary reason for using distributed agent models is that they are usually expected to outperform central optimization models in problem instances with high levels of uncertainty. Tak-

ing this for granted, researchers usually show that their distributed algorithm is better than the distributed algorithms of others. Experiments studying the behavior of distributed methods over varying levels of uncertainty in comparison to centralized optimization methods are generally absent from the literature.

If advanced swapping and decommitment techniques are used, can fully distributed agents perform competitively with (or better than) centralized optimization in highly uncertain settings? Can the time gained in doing local operations compensate for the loss of not considering crucial global information? In our opinion these questions have not been fully answered. In this paper, we construct a distributed agent model using the most promising techniques as identified in the agent literature and compare this approach via experiments on a real data set to a state-of-the-art centralized on-line optimization approach. The lack of appropriate comparisons between agent-based approaches and existing techniques for transportation and logistics problems possibly indicates a belief on the part of agent researchers that agent-based systems outperform traditional methods [3]. Our goal is to add credibility to this belief by studying a state-of-the-art agent-based system in comparison to a state-of-the-art centralized optimization approach for a real-world dynamic transportation problem. In the following section we define in detail the exact VRP that we use to study both the distributed agent-based and centralized optimization-based approaches.

3. VEHICLE ROUTING PROBLEM

Many of the agent-based approaches for vehicle routing problems are tested on generated data-sets. These data-sets are usually constructed to test specific features of the agent system - often focusing on the extreme ends of the performance spectrum. We, however, want to understand the potential of agent solutions in the highly uncertain real world. To that end we are fortunate to have access to operational data from a mid-sized Dutch logistics service provider (LSP) engaged in the road transport of sea containers. While the LSP that we study is active in several sectors, we focus only on the container division which has a fleet of around 40 trucks, handling an average of 65 customer orders each day.

The process of executing an order starts with receiving an order, generally one day before execution is required. While the orders are often called in one day early, the company does not generally use this information in planning routes or establishing schedules. This is due to the unreliable nature of the order information and the resulting uncertainty encountered during execution. An order is a customer request to the LSP for pickup and transport of a specific container from a container terminal (in the case of an import container) to the customer, with delivery within a certain time window. Arriving at the customer's requested location, the container is then unloaded, and the empty container is brought back to a container terminal or empty depot. This concludes the order, and the truck is ready for its next order. The process is reversed for export containers. What adds uncertainty to this process is that not all containers are available at the time indicated in the received order: either they have not physically left the ship at the expected time or they are delayed for administrative reasons, e.g. an unsettled payment or customs clearing. The LSP can only transport containers that have been released, and are al-

lowed to leave the container terminal. For this reason it is hard to optimize the system in a traditional sense, since not all information is known beforehand, and will only become available at some point in time during the day.

The planning and control of operations is currently performed manually by a team of three human planners, who take care of order intake, arrange the proper amount of trucks based on the expected workload, and assign current orders to trucks. Given the primarily manual method of operations, the addition of a computerized decision support system may greatly enhance the profitability and scalability of the LSP's operations.

To formalize the structure of this case study problem we make several formal assumptions:

- Each demand is available for scheduling at the time it is announced. The announcement of a demand includes all information on: the pick-up location (zipcode), the customer location (zipcode), return drop-off location (zipcode), and the required time windows for arrival at each of these three locations.

- Loading and unloading at the terminals and customer takes time. Picking up a container requires 60 minutes; servicing the container at the customer requires 60 minutes; and returning a container to the final terminal takes 30 minutes.

- All travel times are measured according to data on the Benelux road network.

- No time window violations are allowed; if a job is going to violate time windows then it is rejected at a penalty.

- The penalty for rejecting a job is equal to the loaded time of the job. Given the problem structure defined here, loaded time serves as a proxy for revenue.

- Given the demand structure, the truckload nature of the problem, and the fact that the truck must remain with the container at the customer location, we bundle the pick-up, drop-off, and return activities into one job. The loaded time of a job is then the time spanning the arrival at the pick-up terminal through the completion of service at the return terminal - including all loading and unloading times.

- All trucks in the fleet are equivalent.

Given this context, the objective of this vehicle routing problem is to derive a schedule in real-time that serves as many jobs as possible at the least cost. Cost is defined here in terms of time, as the time spent traveling empty (i.e. non-revenue generating travel) to serve all jobs in addition to the loaded time penalty affiliated with rejecting jobs. By adding a penalty for rejecting jobs equal to the loaded distance (in terms of time) of each job, the obvious cost-minimizing solution of rejecting all jobs is avoided. In this regard, it is important to note that in our setting the loaded distance of an order is approximately four times as great as the empty distance incurred in serving that job.

4. AGENT-BASED APPROACH

Based on the agent-based modeling literature and the assumptions related to our problem as introduced in Section

3, our goal is to design, using selected techniques from the literature, a distributed agent model that can outperform a centralized optimization approach. Since we are primarily interested in distributed agent models, we use an uncompromisingly flat architecture: no agents can concentrate information from a multitude of other agents. The global idea of our agent-based planning system is to apply an advanced insertion heuristic in a distributed setting and combine this with two heuristics for making (local) improvements: substitution of orders, and random attempts for re-allocation of orders. The only two kinds of agents that participate in this planning system are *truck* agents and *order* (or *container*) agents.

Our order agents represent container orders. The particularity of container orders is identical to the real-world case of the previous section in that they are described by the three stops required: a pick up at a sea-terminal, a delivery at the customer's, and a drop-off return at a possibly different sea-terminal. With each of the three stops there is a time window and a service time associated, which are obeyed by the trucks. Truck agents represent trucks with a single chassis, which means that they can transport only one order at a time. They make plans in order to transport as many containers as they can.

Order agents hold auctions in order of their arrival, and truck agents bid in these auctions. This results in partially parallel sequential auctions. Trucks may bid on multiple orders at the same time; these bids are not binding. If a truck happens to win more than one order, it takes only the first one. All the other orders it won parallel to the first one are rejected, which results in the rejected order agents starting a new auction. Truck agents ultimately accept only one winning bid on parallel auctions as all bids submitted in parallel are highly dependent on the order of previously won and accepted bids. In this way, in the end, the orders are auctioned sequentially, even if they happen to arrive at the same time.

To clear an auction, order agents choose the best bid as winner, and respond positively to the winner and negatively to the others. For this we chose a one-shot auction (and more specifically, a Vickrey auction [20]) for its computational efficiency, as in the model of Hoen and La Poutré [6]. If the winner confirms the deal, a contract is made. These contracts are semi-binding, so truck agents might break it in order to achieve a better allocation.

At the heart of the agent model are the decisions truck agents make. The most important decision they have to make is the bid they submit for a given order. Every truck agent submits a bid that reflects its cost associated with transporting the given order. This cost is a quantity in the time domain. To calculate it, a truck considers *inserting* the new order into its plan, or alternatively *substituting* one of the already contracted orders by the new one.

To calculate the cost of insertion, the truck agent tries to insert the new order in-between every two adjacent orders in their plan (see Figure 1), plus at the beginning and the end. At every position, it calculates the amount of extra empty time it needs to drive if this order is inserted there. Suppose that an agent considers the position between container i and j, and calculates that the empty time the truck needs to travel to pick up j after returning i is d_{ij}. Here we use d_{ij} to represent the distance (in time) between the two jobs i and j, and d_{ii} to denote the loaded distance of job i. The
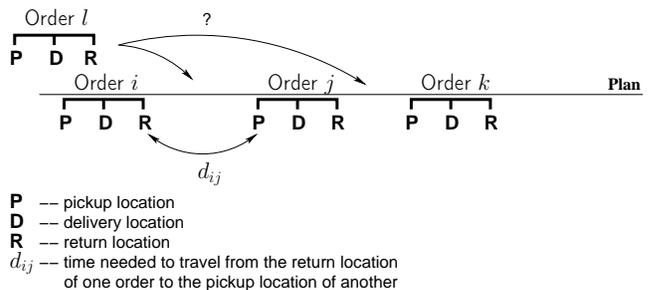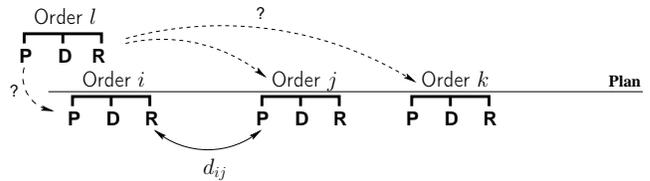

Figure 1: Insertion

Figure 2: Substitution

amount of extra empty time the truck would need to drive for container l then equals $ins_{ij}^l = d_{il} + d_{lj} - d_{ij}$.

In addition to insertion, a truck agent also considers substitution (analogous to what others call decommitment). To calculate the cost of substituting one of the already contracted orders by the new one, it sums up the cost components. The first component is the insertion cost of the new order at the place of the substituted order, the second component is the lost profit on the substituted order, and the third component is a penalty term. For example, we compute the cost of substituting order j with order l ($subs_j^l$) in Figure 4. Here $subs_j^l = ins_{ik}^l + profit_j + d_{jj}$. The insertion term ins_{ik}^l is the same as defined above. The value of $profit_j$ is the difference of the price received for order j and its insertion cost: $profit_j = price_j - ins_{ik}^j$. This term represents the market position of the substituted order in the bid. If the competition for order j is fierce, the profit on j would be low (since the second-best bid was hardly higher than the winning bid). This results in a low substitution cost, therefore such orders are more likely to be substituted. An order that is well suited for a specific truck is likely to produce a high profit for that truck, therefore it will have a high substitution cost. The last term in this expression, the amount of loaded time of order j, serves as a penalty on substituting that job. Using such a penalty discourages the substitution of long orders that may be harder to fit somewhere else. Additionally, the orders that are finally rejected (those that do not manage to make a contract with any truck agents) will be shorter, which will result in a better total cost. Algorithm 1 describes how new orders are dealt with.

In addition to bidding on auctions for new orders, truck agents have another way to enhance the overall solution. At random time intervals, every truck randomly selects an order in its plan and releases it. Trucks never select the order they are currently serving and also not one, for which the execution is about to begin (the pick-up time of the container is less than 10 seconds away – this small time

---
Algorithm 1 Insertion and substitution of orders
---

1. Compute the extra costs for every possible insertion and every possible substitution

2. Order the merged list of insertions and substitutions in increasing order of these costs

3. Iterate over this list

   (a) If the new order's time windows are violated, continue with the next alternative.

   (b) If a time window of an order after the new one is violated, continue with the next alternative.

   (c) Else the cheapest feasible position is found. Return this position.

---

buffer is selected to provide as much opportunity for route improvement as possible). Note, the same time limit is also applied to the insertion and substitution decisions explained earlier. An order agent that is released (just as those order agents that are substituted) initiates a new auction to find another place. In most cases, these auctions result in the very same allocation as before the release. Nevertheless, sometimes they do manage to find a better place and make a contract with another truck.

Whenever an order agent finalizes a contract with a truck agent, it sends a message to all other order agents to notify them about the changed plan of the given truck. This is important for order agents that do not have a contract yet. Any change in the trucks' plans may be their chance to find their place in a truck. Those order agents will start an auction in response to the notification message in the hope of finally making a contract.

To summarize the agent-based approach, let us list the main techniques that characterize it:

- Orders are allocated to trucks via second-price auctions sequentially, at the time they become known to the agent system.

- Truck agents consider insertion and substitution of new orders in their plan. Substituted orders are released from the truck. Released order agents hold a new auction to find another place. If a truck cannot deliver an order within the time windows, it rejects it.

- Truck agents randomly release contracted order agents. Randomly released order agents also hold a new auction to find a place.

- Order agents notify each other whenever they change the plan of a truck (make a contract). Rejected orders (without a contract) thereby get a chance to hold a new auction and find a truck.

To evaluate this approach, we implemented a real-time truck simulator that we connected to the agent system. Every truck agent assumes responsibility for a simulated truck. In the coupled agent-truck-simulator system, agents send plans to trucks for execution. Simulated trucks drive along the road network of the Benelux as the plans prescribe. They periodically report their position as well as their activities to the agents. This way truck agents can follow the execution
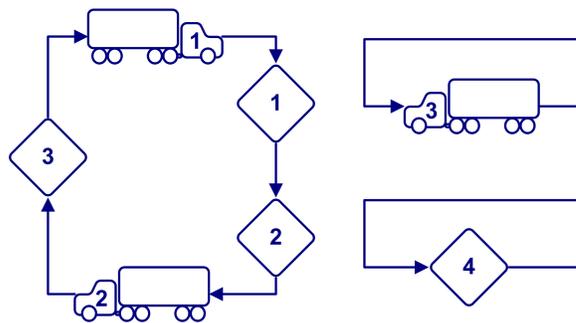

Figure 3: Cycles in the MIP solution structure.

of the plans and make decisions with the knowledge of what is happening in the (simulated) world.

Finally, we have a third element in the system, whose role is to monitor both the agents and the simulator, thereby gathering all information necessary to evaluate the performance of the agents, and to calculate the total cost of the routing. Just as described in Section 3, the ultimate objective of the agents is to minimize the total cost of the routing which is specified in terms of the time trucks travel empty plus the loaded-travel-time penalty associated with rejecting a container. The next section describes the on-line optimization approach that is used in comparison to the agent-based approach, based on this total cost.

5. ON-LINE OPTIMIZATION APPROACH

To estimate the value of the agent-based solution approach (described in Section 4), we study it in comparison to an optimization-based solution approach, reflective of those currently embedded in commercially available vehicle routing decision support software (DSS). We therefore examine two optimization based solution approaches: (i) a mixed-integer program for solving the static *a priori* case in order to provide a baseline benchmark, and (ii) an on-line optimization approach, comparable to the agent approach, and designed to represent current vehicle routing DSS.

At the core of both the static *a priori* solution and the on-line optimization is a mixed integer program (MIP) for a truck-load vehicle routing problem with time windows, which is given to CPLEX [7]. This MIP is based on the formulation put forth by Yang et al. [21]. The complete description of our modifications to Yang et al.'s MIP is the focus of this section. Before introducing the notation and mathematical formulation for this problem, we begin with a small example to illustrate exactly how Yang et al.'s MIP works to exploit the structure of this truckload pick-up and delivery problem with time windows.

Imagine a scenario with three trucks and four jobs. The model of Yang et al. is constructed such that it will find a set of least cost cycles describing the order in which each truck should serve the jobs. For example, as depicted in Figure 3, the outcome may be a tour from truck 1 to job 1, then job 2, then truck 2, then job 3, then back to truck 1. This would indicate that truck 1 serves job 1 and 2, while truck 2 serves job 3. The cycle including only truck 3 indicates that truck 3 remains idle. Similarly, the cycle including only job 4 indicates that job 4 is rejected.

Given the assumptions in Section 3, we designate the fol-

lowing notation for the given information.

- K — the total number of vehicles available in the fleet.
- N — the total number of known demands.
- d_{ij} — as introduced in 4, the travel time required to go from demand i's return terminal to the pick-up terminal of demand j. Note, if $i = j$ then the travel time d_{ii} represents the loaded distance of job i.
- d_{0i}^k — the travel time required to move from the location where truck k started to the pick-up terminal of demand i.
- d_{iH}^k — the travel time from the return terminal of demand i to the home terminal of vehicle k.
- v^k — the time vehicle k becomes available.
- l_i — the loaded time required of job i (time from pick up at originating terminal to completion of service at the return terminal). Note, $l_i = d_{ii}$.
- τ_i^- — earliest possible arrival at demand i's pick-up terminal.
- τ_i^+ — latest possible arrival at demand i's pick-up terminal.
- M — a large number set to be $2 \cdot \max_{i,j}\{d_{ij}\}$.

Note: τ_i^- and τ_i^+ are calculated to ensure that all subsequent time windows (at the customer location and return terminal) are respected. Given the problem of interest, we specify the following two variables.

- x_{uv} — a binary variable indicating whether arc (u, v) is used in the final routing; $u, v = 1, \ldots, K + N$.
- δ_i — a continuous variable designating the time of arrival at the pick-up terminal of demand i.

Using the notation described above, we formulate a MIP that explicitly permits job rejections, based on the loaded distance of a job.

$$\min \quad \sum_{k=1}^{K}\sum_{i=1}^{N} d_{0i}^k x_{k,K+i} + \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij} x_{K+i,K+j}$$
$$+ \sum_{i=1}^{N}\sum_{k=1}^{K} d_{iH}^k x_{K+i,k} \tag{1}$$

such that

$$\sum_{v=1}^{K+N} x_{uv} = 1 \qquad \forall u = 1, \ldots, K+N \quad (2)$$

$$\sum_{v=1}^{K+N} x_{vu} = 1 \qquad \forall u = 1, \ldots, K+N \quad (3)$$

$$\delta_i - \sum_{k=1}^{K}(d_{0i}^k + v^k)x_{k,K+i} \geq 0 \qquad \forall i = 1, \ldots, N \quad (4)$$

$$\begin{aligned} &\delta_j - \delta_i - M x_{K+i,K+j} + \\ &(l_i + d_{ij})x_{K+i,K+i} \qquad \forall i,j = 1, \ldots, N \quad (5)\\ &\geq l_i + d_{ij} - M \end{aligned}$$

$$\tau_i^- \leq \delta_i \leq \tau_i^+ \qquad \forall i = 1, \ldots, N \quad (6)$$

$$\delta_i \in \mathbb{R}^+ \qquad \forall i = 1, \ldots, N \quad (7)$$

$$x_{uv} \in \{0, 1\} \qquad \forall u, v = 1, \ldots, K+N \quad (8)$$

In words, the objective (1) of this model is to minimize the total amount of time spent traveling without a profit generating load. This objective is subject to the following seven constraints:

(2) Each demand and vehicle node must have one and only one arc entering.

(3) Each demand and vehicle node must have one and only one arc leaving.

(4) If demand i is the first demand assigned to vehicle k, then the start time of demand i (δ_i) must be later than the available time of vehicle k plus the time required to travel from the available location of vehicle k to the pick up location of demand i.

(5) If demand i follows demand j then the start time of demand j must be later than the start time of demand i plus the time required to serve demand i plus the time required to travel between demand i and demand j; if however, demand i is rejected, then the pick up time for job i is unconstrained.

(6) The arrival time at the pick up terminal of demand i must be within the specified time windows.

(7) δ_i is a positive real number.

(8) x_{uv} is binary.

Mathematically this model specification serves to find the least-cost (in terms of time) set of cycles that includes all nodes given in the set $\{1, \ldots, K, K+1, \ldots, K+N\}$. We define $x_{uv}, (u, v = 1, \ldots, K+N)$ to indicate whether arc (u, v) is selected in one of the cycles. These tours require interpretation in terms of vehicle routing. This is done by noting that node $k, (1 \leq k \leq K)$ represents the vehicle k and node $K + i, (1 \leq i \leq N)$ corresponds to demand i. Thus, each tour that is formed may be seen as a sequential assignment of demands to vehicles respecting time window constraints.

The model described above is used to provide the optimal (yet realistically unattainable) lower bound for each day of data in each scenario. We denote this approach as the static *a priori* case. In this case, we obtain the route and schedule as if all the jobs are known and we have hours to find the optimal solution. Thus, not only is this lower bound realistically unattainable due to a relaxation on the amount of information available, but also due to a relaxation on the amount of time available to CPLEX for obtaining the optimal solution. In this way, because the problem instances are relatively small (note, using this MIP structure CPLEX can handle a maximum of about 100 jobs and about 50 Trucks, yet our instances are only 34 trucks and 65 jobs) we are able to uncover the optimal solution for all 26 problem instances across all four uncertainty scenarios.

In order to provide a fair comparison with the agent-based approach, the MIP is then manipulated for use in on-line operations. In our on-line approach, this MIP is invoked at 30 second intervals. At each interval, the full and current state of the world is captured, and then encoded in the MIP. This "snapshot" of the world includes information of all jobs that are available and in need of scheduling, as well as the forecasted next available location and time of all trucks. The MIP is then solved via a call to CPLEX. The decision to use 30 second intervals was driven by the desire to be comparable to the agent-based approach while still providing CPLEX enough time to find a feasible solution for each snapshot problem. The solution given by CPLEX is parsed and any jobs that are within two intervals (i.e. 60 seconds) of being late (i.e. missing the time specified by δ_i in the latest plan) are permanently assigned if travel is not commenced in the next interval. Any jobs that were designated for rejection in the solution are rejected only if they are within two intervals of violating a time window;

otherwise they are considered available for scheduling in a subsequent interval. The procedure continues in this fashion until the end of the working day at which point all jobs have been served or rejected.

The test problems and the results from the static *a priori* benchmark, the on-line optimization, and the agent-based solution approach as applied to these test problems are the topic of the next section.

6. COMPUTATIONAL EXPERIMENTS

In this section we report the computational results on the performance of the agent-based approach in comparison to the optimization-based approach. The first subsection (6.1) describes how the test problems were generated and in subsection 6.2 we present the results of these tests.

6.1 Test Problems

The data we used for our experiments was based on data provided to us by the LSP described in Section 3. In all, we were given the execution data from January 2002 to October 2005 as well as the data from January 2006 through March 2006. We could not, however, simply use this data in its raw form. We first had to make multiple corrections to the customer address fields as many addresses referred to postal boxes and not to the physical terminal locations. After cleaning the address fields, we then extracted a random sample of jobs from the original data-set in order to generate a set of 26 days with 65 orders per day. The company from which these data are taken serves between 50 and 80 jobs per day, thus 65 jobs per day represents the average daily job load.

Just as discussed before, each order consists of a pickup location, customer location, and return location. To standardize the data for our experimental purposes we specified time windows at all locations as follows: for the terminals (the pickup and return locations) the time windows span a full twelve hour work day from 6am to 6pm and the time windows at the customer locations are always 2 hours. The start of the 65 customer time windows occurs throughout the working day in accordance with the data provided by the LSP, which roughly follows a uniform distribution. Given the variation in customer locations, the workload per day varies similarly. On average each job requires approximately 4.2 hours of loaded distance. When the routing is optimal in the case that all jobs are known at the start of the day the average empty time per job is approximately 25 minutes.

Given our interest in determining how the agent solution performs on this pick-up and delivery problem with time windows and order arrival uncertainty, we further rendered our 26 days of data into four separate scenarios with varying levels of order arrival uncertainty. This was done by altering the arrival times of the orders, i.e., the time at which the order data is revealed to the LSP. We generated these points in times over the day using a uniform distribution. We used such a uniform distribution as the original data did not show a fit with other distributions. The four different scenarios reflecting different levels of order arrival uncertainty were:

Scenario A: All orders (100%) are known at the start of the working day, 6AM.

Scenario B: About half of the orders (50%, selected randomly from the 65 jobs) are known at the start of the
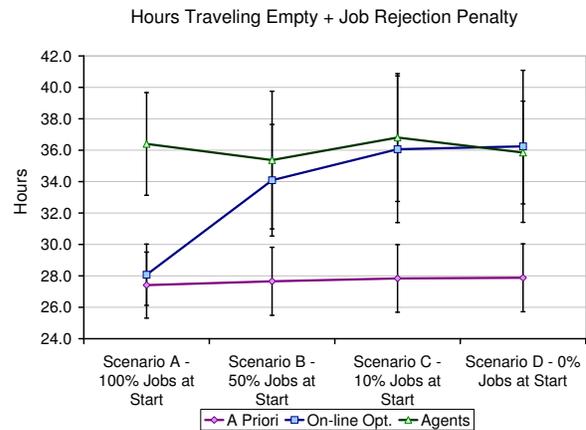

Hours Traveling Empty + Job Rejection Penalty

Figure 4: Mean, over 26 days, of the total cost for the three approaches across the four scenarios. Bars indicate ± standard deviation.

working day, 6AM. The other half of the orders arrive two hours before the start of the customer location time window (i.e., four hours before the end of the customer location time window, leaving slightly less than two hours on average before the latest departure time from the pickup location).

Scenario C: Only seven of the jobs (10%, selected randomly from the 65 jobs) are known at the start of the working day, 6AM. The remaining 58 jobs arrive two hours before the start of the customer location time window.

Scenario D: None of the jobs (0%) are known at the start of the working day. All 65 jobs arrive two hours before the start of the customer location time window.

If we classify these scenarios in terms of the effective degree of dynamism for vehicle routing problems with time windows as developed by Larsen et al. in 2002 [10] then values of dynamism for Scenarios A, B, C, and D are .5, .7, .8, and .9, respectively. Noting that this form of measuring uncertainty may range from 0 to 1 with 1 being the most uncertain, then we may say that our test problems range from partially uncertain to mostly uncertain.

6.2 Computational Results

All three solution approaches were applied to each of the 26 days of data in the four scenarios. The mean cost over the 26 days of these experiments may be seen in Figure 4. From this graphical depiction, the on-line optimization procedure clearly outperforms the agents only in Scenario A. In fact, in Scenario A in which all information is known at the start of the day, the on-line optimization performs at a level quite close to the realistically unattainable benchmark optimal. The on-line optimization does not, however, achieve optimal in Scenario A as the snapshot problem in the first 30 second interval represents the full problem size. A size for which finding the optimal solution in thirty seconds is quite difficult. In all cases, CPLEX does, however, provide a feasible solution which can then be improved in future intervals. In the remaining three scenarios, however,

Table 1: Mean ± standard error over 26 days for on-line optimization and the agent-based approach on the total cost for scenarios A, B, C, and D.

	A	B	C	D
On-line Opt.	28.07 ± .38	34.09 ± .70	36.06 ± .92	36.24 ± .95
Agents	36.4 ± .64	35.37 ± .86	36.81 ± .80	35.85 ± .64

Table 2: Results of the t-test on the null hypothesis that the means of the total cost of the two datasets are equal (with .05 significance).

	A	B	C	D
Calculated t-value	11.16	1.16	.61	.34
Tabulated t-value	2.01	2.01	2.01	2.01
Result	Reject	Fail to Reject	Fail to Reject	Fail to Reject

the agents perform at a level competitive to the on-line optimization.

To fully understand the competitive nature of the agents in the dynamic settings of Scenarios B, C, and D a t-test was performed to determine if the average total cost of the routing solutions were statistically equivalent. The results of these tests may be seen in Tables 1 and 2. From these results we may conclude that for the reality-based datasets used in this study, agent-based solution approaches perform competitively with the on-line optimization when at least half of the jobs is unknown at the start of the day.

While the study of total cost and associated t-test results are promising for the agent approach, we must also look at the portion of this total cost due to the job rejection penalty and the portion of the cost due to empty travel time. Figure 5 depicts the penalty of rejected jobs on the left axis and the number of jobs rejected on the right axis. Note, we do not include the *a priori* optimal in this figure as no jobs were rejected using this approach. While the on-line optimization demonstrates a clear trend in the number of rejections (the more dynamic the setting the more jobs are rejected at a higher penalty), the agent approach does not demonstrate any trend. In comparing Figure 6 and Figure 5, it is clear that this irregular job rejection trend of the agent approach is having a significant impact on the trend in the total cost of the agent approach (see Figure 4).

Figure 6 depicts the average number of hours spent traveling empty in the routing solution provided by each approach in the four scenarios. From this figure, all three approaches show a general trend toward an increased level of empty travel with an increased level of uncertainty. Interestingly, however, the agent approach shows far more stability in this regard. In this sense we may conclude that despite the agents' poor performance in our less uncertain settings, they are, however, less susceptible than on-line optimization to the effects of high uncertainty. Yet, in the end, both systems perform comparatively well in the most uncertain setting.

7. DISCUSSION

In this paper, we studied an on-line truckload vehicle routing problem arising from a real-world case study. We proposed a state-of-the-art agent-based solution approach and compared that approach to a well known on-line optimization approach. The computational results, from 26 days of data spanning four different scenarios representing various levels of job arrival uncertainty, indicate that the agent-
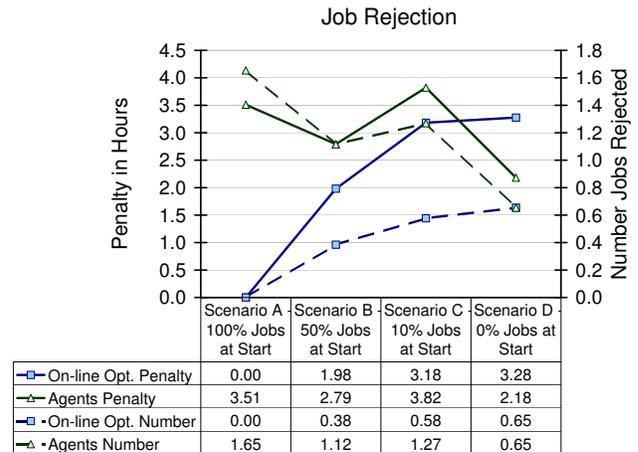

	Scenario A 100% Jobs at Start	Scenario B 50% Jobs at Start	Scenario C 10% Jobs at Start	Scenario D 0% Jobs at Start
On-line Opt. Penalty	0.00	1.98	3.18	3.28
Agents Penalty	3.51	2.79	3.82	2.18
On-line Opt. Number	0.00	0.38	0.58	0.65
Agents Number	1.65	1.12	1.27	0.65

Figure 5: Job rejection in terms of penalty and number of jobs rejected for the on-line optimization and agent approaches across four scenarios.

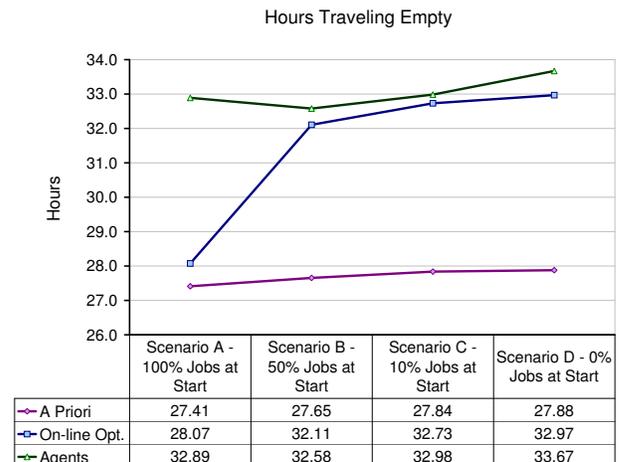|	Scenario A - 100% Jobs at Start	Scenario B - 50% Jobs at Start	Scenario C - 10% Jobs at Start	Scenario D - 0% Jobs at Start
A Priori	27.41	27.65	27.84	27.88
On-line Opt.	28.07	32.11	32.73	32.97
Agents	32.89	32.58	32.98	33.67

Figure 6: Time spent traveling empty for the three approaches across the four scenarios.

based approach is highly competitive in cases where less than 50% of the jobs are known in advance.

Given these results, agents should be considered as a viable decision support mechanism for transportation planners that must cope with uncertain job arrivals. If, however, the job arrival environment is relatively static, that is more than half of the jobs are known at the start of the day, then optimization should remain the tool of choice. Admittedly, this recommendation carries the following caveat. The agents do suffer a certain level of instability as reflected in the lack of a trend in job rejections relative to the level of uncertainty. The reason is that while job rejection is explicitly handled in the optimization model, it is implicit in the agent model. When an agent rejects an order, it has no way of knowing whether other agents will reject it too. In general, it is therefore more difficult to implement a global notion such as the number of rejected orders in an agent approach. In practice, a transportation provider must be very explicit about routing priorities. If a consistent or predictable level of job rejections is important then on-line optimization is a better choice.

One of the reasons that the agent-based solution performs consistently in terms of empty distance traveled is because of the sequential auction method used to handle jobs that arrive simultaneously. Thus, in Scenario A, in which the uncertainty is low, the agents must run many auctions at the start of the day; on-line optimization on the other hand may exploit all of this information at once to obtain a near optimal solution. In Scenario D, on the other hand, the agents approach the auctions in very much the same way as in Scenario A except that they are spread more evenly over time. In contrast the on-line optimization is forced to adopt job assignments that may preclude the assignment of jobs arriving late in the day.

In short, agent-based systems perform well in settings where less than half of all jobs are known in advance. Agents do, however, present issues concerning tractability in terms of rejected jobs. The number and penalty of rejected jobs is particularly variable with no clear trend across the four scenarios. Finally, in steep contrast to the online optimization, the agents used in this study are not well suited to exploit large batches of job arrivals; agents tend to perform better when a small number of jobs arrive evenly spaced through out the planning horizon.

Noting from these cases the impact of clumped job arrivals on the two approaches brings us to our first extension of this work. We recommend that both systems be tested across several problem sizes and a variety of uncertain job arrival patterns to truly understand the effect of clumped job arrivals.

Turning now to the theme of uncertainty, job arrival uncertainty as studied here represents only one narrow definition of uncertainty. A simple extension to this definition by including variable numbers of jobs across the days (i.e. each day would have a different number of jobs taken from the range 50 to 80) will provide additional insight on the strengths and weaknesses of agents in handling uncertainty. Furthermore examining other sources of uncertainty in the transportation domain, such as loading, unloading, and travel time variability, will not only add realism to the study, but will also yield a more robust view on the benefits and drawbacks of an agent approach as compared to centralized approaches.

Another extension of this work is the introduction of optimization into the agent approach. In this way, the agents may be able to capitalize on the benefit of optimization in less uncertain situations and the benefit of local heuristics in more uncertain situations. We conclude by stating that agent-based approaches may have even greater benefits when we consider modeling other forms of uncertainty such as travel time uncertainty, loading and unloading time uncertainty, and so forth. The field for agent-based approaches to the VRP is wide open, but should also be carefully explored to ensure that the practical everyday needs of real-world transport planners are met.

Acknowledgments

8. REFERENCES

[1] H.-J. Bürckert, K. Fischer, and G. Vierke. Holonic transport scheduling with Teletruck. *Applied Artificial Intelligence*, 14(7):697–725, 2000.

[2] J.-F. Cordeau, G. Desaulniers, J. Desrosiers, M. M. Solomon, and F. Soumis. VRP with time windows. In *The vehicle routing problem*, pages 157–193. Society for Industrial and Applied Mathematics, 2001.

[3] P. Davidsson, L. Henesey, L. Ramstedt, J. Törnquist, and F. Wernstedt. An analysis of agent-based aproaches to transport logistics. *Transportation Research Part C*, 13(4):255–271, 2005.

[4] K. Dorer and M. Calisti. An adaptive solution to dynamic transport optimization. In *Proc. of 4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2005)*, pages 45–51, New York, NY, USA, 2005. ACM Press.

[5] K. Fischer, J. P. Muller, M. Pischel, and D. Schier. A model for cooperative transportation scheduling. In *Proc. of the 1st Int. Conf. on Multiagent Systems*, pages 109–116, Menlo park, California, 1995. AAAI Press / MIT Press.

[6] P. J. Hoen and J. A. L. Poutré. A decommitment strategy in a competitive multi-agent transportation setting. In *Proc. of 2nd Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2003)*, pages 1010–1011, New York, NY, USA, 2003. ACM Press.

[7] C. Incorporated. Using the CPLEX Callable Library and CPLEX Mixed Integer Library, 1992.

[8] N. Jennings and S. Bussmann. Agent-based control systems: Why are they suited to engineering complex systems? *Control Systems Magazine, IEEE*, 23(3):61–73, 2003.

[9] R. Kohout and K. Erol. In-time agent-based vehicle routing with a stochastic improvement heuristic. In *AAAI '99/IAAI '99: Proc. of the 16th national conference on Artificial Intelligence and the 11th on Innovative Applications of Artificial Intelligence*, pages

864–869, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[10] A. Larsen, O. Madsen, and M. Solomon. Partially dynamic vehicle routing-models and algorithms. *Journal of the Operational Research Society*, 53:637–646, 2002.

[11] H. W. Leong and M. Liu. A multi-agent algorithm for vehicle routing problem with time window. In *Proc. of the ACM Symposium on Applied Computing (SAC 2006)*, pages 106–111, New York, NY, USA, 2006. ACM Press.

[12] M. Mes, M. van der Heijden, and A. van Harten. Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research*, 181(1):59–75, 2007.

[13] J. A. Persson, P. Davidsson, S. J. Johansson, and F. Wernstedt. Combining agent-based approaches and classical optimization techniques. In *Proc. of the European workshop on Multi-Agent Systems (EUMAS 2005)*, pages 260–269, 2005.

[14] D. Perugini, D. Lambert, L. Sterling, and A. Pearce. A distributed agent approach to global transportation scheduling. In *IEEE/WIC Int. Conf. on Intelligent Agent Technology (IAT 2003)*, pages 18–24, 2003.

[15] T. W. Sandholm and V. R. Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behaviour*, 35:212–270, 2001.

[16] M. Schillo, C. Kray, and K. Fischer. The eager bidder problem: a fundamental problem of DAI and selected solutions. In *Proc. of 1st Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2002)*, pages 599–606, New York, NY, USA, 2002. ACM Press.

[17] R. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

[18] P. Thompson and H. Psaraftis. Cyclic transfer algorithms for multivehicle routing and scheduling problems. *Operations research*, 41(5), 1993.

[19] P. Toth and D. Vigo, editors. *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and its Applications. Society for Industrial and Applied Mathematic, 2002.

[20] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

[21] J. Yang, P. Jaillet, and H. Mahmassani. On-line algorithms for truck fleet assignment and scheduling under real-time information. *Transportation Research Record*, 1667:107–113, 1999.