The Learnable Evolution Model in Agent-based Delivery Optimization

Janusz Wojtusiak[1], Tobias Warden[2], and Otthein Herzog[2]

[1] Machine Learning and Inference Laboratory, George Mason University,
jwojt@mli.gmu.edu
[2] Center for Computing and Communication Technologies,
University of Bremen, Germany
{warden,herzog}@tzi.de

Abstract. The Learnable Evolution Model is a stochastic optimization method which employs machine learning to guide the optimization process. LEM3, its newest implementation, combines its machine learning mode with other search operators. The presented research concerns its application within a multi-agent system for autonomous control of container on-carriage operations. Specifically, LEM3 is used by transport management agents that act on behalf of the trucks of a forwarding agency for the planning of individual transport schedules.

1 Introduction

This paper describes an agent-based control system that autonomously handles the processing of incoming transport orders. The main contributions of this work are two-faced: First, a novel model for the autonomous evaluation of orders is proposed. Second, the Learnable Evolution Model is used for the formation of pickup and delivery plans for trucks in a freight forwarder transport fleet.

The increasing pace of structural change in today's globalized markets bears significant implications for adequate planning and control strategies in transport logistics. Traditional supply chains are evolving into complex networks with numerous stakeholders. Three main effects which characterize these changes have been identified: the goods structure effect, the logistics effect, and the structural effect. The first describes a shift away from mass production towards a buyers' market, which creates a trend towards individual product customization and consequently a noticeable increase in per-unit shipments. The second effect relates to a shift towards road freight transport, which arises from the increasing demands for small shipments along with a high quality of service and due-date reliability. Finally, the structure effect indicates an individualization of transport on the micro-logistics level. Cooperation is needed between otherwise competing logistics service providers to satisfy today's customer requirements.

These three effects lead to a dramatic increase in the complexity and dynamics of today's transport logistics processes, which pose as a growing challenge for the traditional paradigm of centralized process management. Centralized planning approaches are gradually becoming insufficient to handle the entire logistics networks. This is specifically true with the frequent occurrence of unexpected events which constitute disruptions to existing plans. The widespread spatial distribution of transport nodes within the networks, as well as the link-up of competing logistics service providers in large supply-networks, restrict or even inhibit the provision of crucial information for central decision-making instances. Multidisciplinary research in autonomous cooperating logistic processes seeks to identify chances to cope with the highlighted challenges by promoting the paradigm shift towards the decentralized control of logistic processes [13, 12, 8]. This paper focuses on multiagent systems as one particular path to implement the novel control paradigm, which has already been found to yield promising results in multiagent-based simulation experiments [33]. The work compiled in [12, 13] shows that multiagent systems are particularly suitable for large scale logistic processes, including one considered here.

The problem addressed in this research is that of a freight-forwarding agency, which needs to handle a continuous dynamic flow of transport orders for freight containers. New orders are received by the system on a regular basis for a particular subset of storage facilities physically distributed over a large geographic area, such as the territory of Germany. In this considered scenario the requirement is that two subsequent transport orders need to be handled for each managed container. The first order comprises the transport of a full container without further distinction of containers with respect to type[3] or enclosed contents. The second, subsequent order comprises the return transport of the empty container after unloading at its initial delivery target facility (reverse transportation).

In order to handle the transport orders, which have already been successfully acquired from customers, the freight forwarding agency exclusively employs its own fleet of transport vehicles. Therefore, the considered scenarios thus far factor out the delegation of transport orders to sub-contractors as additional handling modality. The freight forwarder operates a homogeneous transport fleet of semi-trailer trucks, which are equipped to carry exactly one freight container at a time. Thus, each operated truck can only be in either of the two states - empty or fully-loaded.

Informed decision-making and planning within a multi-agent system presupposes that the involved agents possess the ability to learn from experience and find their best courses of action. The former has been addressed in part in earlier work by Gehrke and Wojtusiak [9]. Therein, rule-based machine learning was used to autonomously build models in order to predict traffic patterns. The problem of determining an optimized course of action is subsequently addressed in this paper through a form of memetic algorithm called the *Learnable Evolution Model* (LEM). In its core, the model uses machine learning to guide an evolutionary optimization process to arrive more rapidly at high-performance plans. Early experiments on LEM [21, 41] indicated its advantage over more traditional evolutionary optimization methods when applied to a set of test problems. The advantage was shown to be growing with the evaluation function complexity, and problem complexity.

[3] Such as refrigerated, hazardous materials or a standard container.

These findings led to the conclusion that LEM is particularly suitable for problems in which the fitness function evaluation is costly. The presented application to agent-based planning falls into this category.

The remainder of the paper is structured as follows: First, the subsequent section introduces the Learnable Evolution Model and relates it to other selected approaches from the field of memetic and evolutionary computing. Then, a special focus is put on LEM3, the most recent version of the Learnable Evolution Model, and its key improvements over predecessors. Section 3 focuses on the proposed method to address the complex planning problem sketched out in the introduction. In particular, this section outlines how the planning problem can be specified into a problem representation suitable for use in LEM3. Section 4 then introduces the constituents of the multi-agent system for autonomous control of container on-carriage. Subsequently, Section 5 describes the simulation experiments within the multiagent-based simulation systems PlaSMA and provides a comprehensive evaluation. Then, Section 6 relates the proposed MAS for container on-carriage with other existing approaches, particularly from the field of operations research. Finally, a conclusion and opportunities for future research are presented.

2 The Learnable Evolution Model

The Learnable Evolution Model (LEM) is an evolutionary optimization method that employs machine learning to direct the evolutionary process [20, 21]. Specifically, LEM creates general hypotheses that indicate regions in the search space that likely contain optimal solutions and then instantiates these hypotheses to generate new candidate solutions (a.k.a. individuals). In order to apply machine learning, LEM creates two groups of candidate solutions that are respectively high- and low-performing according to the fitness function(s) being optimized. These candidate solutions can be selected from the current population or a combination of current and past populations of individuals. The group of high-performing individuals is called H-Group and the group of low-performing individuals is called L-Group. Once the groups are selected, LEM applies concept learning to create a general hypothesis describing the H-Group in contrast to the L-Group. The hypothesis is then instantiated to create new candidate solutions. In the final step, a new population is assembled from old and new candidate solutions, and the process is repeated until stopping criteria are met.

The LEM3 implementation of the Learnable Evolution Model [41], used in the described work, is an adaptive multistrategy evolutionary optimization method. As such, it falls into category of memetic algorithms. Among other methods closely related to LEM are estimation of distribution algorithms (EDA), cultural algorithms, the wise breeding genetic algorithm, and others. Previous implementations of LEM, although using less sophisticated learning methods, achieved highly competitive results on benchmark optimization problems.

Estimation of distribution algorithms use statistical inference and learning to generate distributions of high-performing candidate solutions selected from one population [25, 18, 32]. This is usually done without using a contrast set of low-performing candidate solutions. The most popular statistical inference methods in EDAs include those that utilize different variants of Gaussian distributions. There are also a number of implementations that use Bayesian and Gaussian networks as models for representing hypotheses.

Another class of methods whose idea is close to the Learnable Evolution Model are cultural algorithms [28–31]. These algorithms use information about candidate solutions to guide mutation and recombination operators. The cultural algorithms perform a constrained optimization process, in which constraints are created in parallel to the evolution process. The constraints, called beliefs, are stored in a belief space that is updated to reflect the fitness profile or the current population. Candidate solutions that are stored in an optimization space are modified (via constrained mutation and recombination) so that they satisfy the beliefs. The belief space is built based on statistical information about candidate solutions, which usually consists of intervals containing the fittest candidate solutions.

An effort to combine ideas from the Learnable Evolution Model and the estimation of distribution algorithms has been made by Llorà and Goldberg [17]. The method is in fact a simplified version of LEM, which uses statistical information about a population to instantiate attributes not included in rules. The method uses a fixed bit representation of candidate solutions and passes high-performing and low-performing examples to the ID3 [27] decision tree learning program. Rules obtained (from trees) by ID3 consist of information that indicate which bits are present in high-performing candidate solutions. For bits not included in the rules, the method uses an idea of the population based incremental learning [1, 2], which uses incrementally updated probabilities of values of bits in the best candidate solutions.

The idea of LEM was also applied in the recently published *Informed Evolutionary Algorithm* (InEA) [6]. The overall algorithm is, in fact, very similar to earlier implementations of LEM [20]. In data mining mode it uses association rule mining to discover areas likely to contain the fittest solutions. The rules are then instantiated (through knowledge based mutation) to generate new solutions. In contrast to LEM, InEA uses association rule mining rather than concept learning of attributional rules. InEA also uses a knowledge-based mutation operator which follows the idea used in Differential Evolution.

2.1 The LEM Algorithm

The LEM algorithm consists of elements typically found in evolutionary computation, such as generation of initial population, evaluation, and selection. In LEM, however, new candidate solutions are generated in a three-step process based on machine learning:

1. *Example preparation*, is the process in which candidate solutions are split into H- and L-groups. This split is done based on the solutions' performance (fitness values). Several methods for selecting examples have been investigated [21, 41, 39], including combining solutions from the current and previous populations.

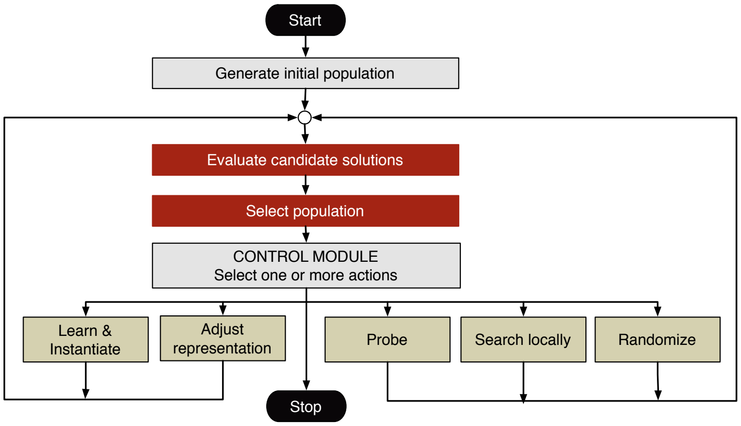


Fig. 1: A top-level flowchart of LEM3, adapted from [22].

For simplicity, one can assume that a selected number of top solutions become H-Group, and bottom ones become L-Group, although in practice more sophisticated methods can be used.

2. *Hypothesis formation*, is the process in which machine learning is applied to form a general hypothesis describing why some candidate solutions perform better than others. In general, any machine learning method that generates easily instantiable hypotheses can be used (see next step). For this reason, LEM3 as well as previous LEM implementations use rule-based hypotheses representation. Specifically, LEM3 applies the AQ21 rule learning system for this task [42]. AQ21 is a highly customizable system which learns attributional rules in the form of Equation (1) for describing high-performing candidate solutions. The CONSEQUENT is a conjunction of attributional conditions built using parameters from the optimization space.

$$[\text{Class} = \text{H-Group}] \Leftarrow \text{PREMISE} \quad (1)$$

3. *Hypothesis Instantiation*, called sampling in statistics, is the process during which new candidate solutions are created to satisfy the learned hypotheses. Attributional rules can be easily instantiated. The default algorithm used in LEM3 uniformly selects values (both numeric and symbolic) for attributes that are included in the PREMISE. For attributes that are not included in the premise, the method selects values from a randomly selected existing non-L-Group candidate solution. Other instantiation methods are discussed by Wojtusiak [41].

2.2 The LEM3 Extensions and Implementation

Very successful initial implementations of the Learnable Evolution Model sparked development of the third generation of LEM software, called LEM3. It extends many ideas found in the original LEM methodologies, some of which are unique in the field of evolutionary computation. The general flow diagram of LEM3's algorithm is presented in Figure 1. In addition to components found in standard evolutionary computation methods, such as generation of an initial population, evaluation of solutions, and selection of candidate solutions, LEM3 includes several additional components. One of its novel features is the ability to dynamically select one or more innovation methods to create new candidate solutions. These innovation methods are:

Learn & Instantiate the aforementioned main mechanism for creating new candidate solutions in LEM3;

Probe to apply traditional operators such as mutation and crossover;

Search locally to apply a user-defined local search method;

Randomize to add to the current population a number of randomly created candidate solutions, or restart the evolutionary process.

One of the important features of LEM3 is its ability to automatically adjust the representation space through *constructive induction* [38, 39]. The original representation space provided to a machine learning, data mining, or evolutionary computation system may be inadequate for performing the desired task for concept learning, pattern discovery, optimization, etc. Constructive induction methods automatically create new representation spaces based on the original representations. The new representations allow the determination of relationships that cannot be represented in the original spaces. Since LEM relies upon machine learning to guide the evolutionary process, finding such relationships is crucial.

The constructive induction process can be characterized by a function $\Psi : E \rightarrow EC$ which maps the original representation space E to a modified representation space, denoted here as EC. New representations are created by removing attributes irrelevant to the considered problem, by modifying domains of attributes (for example by discretizing numeric attributes), and by creating new attributes.

The following section described how the Learnable Evolution Model has been applied as a solver for planning of transport schedules.

3 Autonomous Container On-Carriage

The presented multiagent-based autonomous control system for a forwarding agency includes two components that jointly determine the emerging system behavior. These are local valuation (prioritization) of pending transport orders, and local selection of orders for handling deliveries by trucks in the transport fleet.

These two components involve separate areas of responsibility. While the order valuation is part of order management, the operative order handling (which for the purposes of this research subsumes order selection) is part of transport management. This distinction is introduced in the agent-based implementation of the autonomous control system. Specifically, the system design comprises two primary classes of logistic agents where each handles one of the aforementioned management tasks. Each transport agent acts on behalf of a single truck and manages its intra-company order selection, acquisition, and subsequent operative order handling. These agents rely on adequate valuations of pending transport orders, which constitutes the basis for the order selection. In this study we considered two types of transport management agents that use a simple greedy approach (baseline for comparison), and an evolutionary-based planning approach.

3.1 Problem Definition

The goal of the freight forwarding agency is to deliver the maximum value of cargo at the minimum cost, and therefore maximize its profit:

$$profit = \sum_i (price(order_i) - cost(order_i)) \tag{2}$$

$price(order_i)$ is the shipping price of i-th order agreed between customer and forwarder, and $cost(order_i)$ is the cost of delivering that order. This model assumes that all of the cargo needs to be delivered to its destinations. By increasing value of cargo as described below, it is guaranteed that every container will eventually be delivered. This fact can be shown directly from the order valuation definition shown in equations 3-5.

The presented problem can be viewed as a variant of the Vehicle Routing Problem (VRP) concerns finding the optimal (minimum cost) assignment of cargo and routes to a fleet of trucks delivering goods to customers. In the presented dynamic version of the problem (DVRP), cargo dynamically arrives at different locations during the simulation.

The assumed model, does not explicitly model the profit of the agency, but rather represents a distributed strategy for cargo assignment. The following sections introduce the mechanics of order valuation by the order management agents, and the order selection and planning by the transport management agents.

3.2 (Re-)Valuation of Transport Orders - Cargo Agents

Transport agents select the most profitable orders. Therefore, orders with low *contractual order value* may never be selected for transport. For the freight forwarding agency, such a system behavior evidently violates contract agreements with customers. Therefore, the forwarding agency requires an effectively desgined mechanism that will ensure a timely handling of *all* pending orders. Although this research does not assume contractually fixed delivery times, the system is designed to allow all orders to be handled in a sensible global time window, e. g., 48 hours. The approach used here is based on an autonomous re-valuation of transport orders, which emanates from the original real valuation of the orders in terms of monetary value. The values whose calculation is introduced hereafter should be understood as a means of intra-organizational prioritization of orders. The transport management agents internally consider the order value for order selection which is calculated as follows:

$$value(order, t) = price(order) + prior^{+}(order, t) \tag{3}$$

$value(order, t)$ is computed as sum of a first component $price(order)$ which constitutes the initial order price agreed between customer and forwarder (cf. equation (4)) and a second component $prior^{+}(order, t)$ for the order re-valuation based on the period of time the cargo has already been waiting to establish a transport contract (cf. equation (5)). The initial order price is computed as follows:

$$price(order) = d \cdot (c_0 + r \cdot distance(order)) \tag{4}$$

The constant c_0 denotes fixed costs for the operation of a semi-trailer truck, while the constant r denotes variable operating costs per kilometer. In order to accommodate the two considered 'types' of containers, which are associated with transport orders – with regular content, and empty to be returned to their original storage facility – the dampening factor d has been introduced.

To increase the value of a container depending on the time it has been waiting, the function $prior^{+}(order, t)$ is used. This function is monotonically increasing with t, thus for containers which are kept waiting for a longer period of time the term $prior^{+}(order, t)$ at some point begins to dominate the initial order price. Thus, assuming a sufficiently large number of transport agents, each order will eventually be selected for transport as its value increases with time. While in general, the $prior^{+}(order, t)$ can be sophisticated and account for, amongst others, prediction of travel times, pickup and delivery time windows, and penalties, the basic version shown below has been used for the initial set of experiments. The constant w_c depends only on the fact if a container is full or empty.

$$prior^{+}(order, t) = w_c(order) \cdot t^{\alpha} \tag{5}$$

3.3 Selection of Transport Orders - Transport Agents

The transport management agents, which have been briefly introduced in the preceding sections, are routinely faced with the challenge to autonomously render decisions that determine their respective operative transport planning. This report concentrates on a transport planning approach based on an evolutionary optimization method. Additionally, a simple non-planning type of transport agent which employs a greedy order selection strategy has been implemented as a baseline for the measurement of transport management performance. Sections 3.4 and 3.5 outline both order selection strategies.

A common assumption, which is reasonable within the bounds of a single freight forwarding agency, is that the transport management has access to the entire momentary order situation. These order situation comprise all of the pending orders which are waiting for processing at the distinct storage facilities. The data is conditioned as a lookup table indexed by transport endpoint pairs, which point to a priority queue of orders which belong to the given transport relation. The priority is measured in the respective order value as introduced in the preceding section, which is updated in regular intervals by the order management agents.

3.4 A Baseline Approach to Order Selection

Baseline transport management agents, who have been implemented as part of the freight forwarding agency, employ a *greedy order selection strategy*. Upon initialization, they scan the full set of pending transport orders to identify the most profitable order calculated by:

$$\begin{aligned} bestOrder(t) = \arg\max_{\mathsf{Orders}(t)} \big[\, (value(order, t) - (cost(start(order), dest(order)) \\ + \, cost(pos(truck), start(order)))) \, \big] \end{aligned} \tag{6}$$

where $\mathsf{Orders}(t)$ is the set of all pending transport orders at time t. Once the best order has been identified, the resulting actions which need to be executed by the managed truck can be immediately derived. Two cases can be distinguished:

1. The most profitable order is associated with a transport relation whose starting point corresponds to the momentary position of the truck. In this case, no separate pickup tour is required and it holds that $cost_{Pick}(pos(truck), order) = 0.00\,€$; and
2. a pickup tour is required in order to subsequently handle the selected transport order.

The choice of the next transport order is repeated each time a truck reaches a storage facility, either upon completion of its most recent delivery or a pickup tour. The latter case has significant potential negative implications with regard to the efficiency of the greedy-based operation. The agents are allowed to reconsider their previous delivery choice. This may happen either due to the fact that the originally desired orders have been already assigned to other trucks, or because some better options have materialized. Thus, in order to prevent greedy order selection leading to a behavior where the agents primarily keep performing pickup tours without ever getting to execute the initial transports that led to these pickup tours in the first place, the following restriction has been introduced for the greedy strategy: If a truck has just completed a pickup tour which corresponds to an empty ride between storage facilities and the originally planned delivery cannot be executed from there, choose the most profitable order whose transport starts at the current location of the truck. Only if no such orders exist in the system, fall back to the standard greedy behavior.

3.5 A Planning Approach to Order Selection

Planning transport management agents are routinely faced with the challenge to autonomously render decisions that determine their respective transport plans. These decisions thereby pertain to a choice of adequate action alternatives, the options being for each decision

1. choosing a transport order whose pickup point is the currently considered storage facility, or

2. postponing that choice and relocate to another storage facility.

In essence, the transport agents need to choose which transport order to pick at a specific time and location. That choice is guided by the gains and costs of transporting the container associated with the order. Therefore, by choosing orders in an optimized way, an agent can maximize its financial balance. The behavior of the transport management agents is thus the result of a series of constitutive decisions. This initial situation calls for provident planning, in which a transport agent considers several steps ahead. The transport management agents seek an optimized pickup and delivery plan with a plan horizon of size n. Formally, such a plan is defined as:

$$\begin{aligned} plan^n &= (action_1, action_2, \ldots, action_n) \\ &\text{where}\, action_i \in Deliveries \cup EmptyRides \end{aligned} \tag{7}$$

Deliveries refers to the set of possible delivery actions as determined by the pending transport orders, which have previously been acquired by the transport forwarding agency. *EmptyRides*, in contrast, refers to the set of possible empty journeys between storage facilities. Thus, a transport plan as defined above can blend deliveries and empty drives where the latter can often be interpreted as pickup tours.

The space of *valid* transport plans is specified by means of constraints:

1. Let $start : Deliveries \cup EmptyRides \rightarrow SF$ define a function which returns to the source location of a particular plan step (i. e., in the case of proper orders, the pickup site). Let further $dest : Deliveries \cup EmptyRides \rightarrow SF$ define the complementary function which returns to the target location of a plan step (for proper orders, the delivery site). In both cases, SF thereby constitutes the set of storage facilities in the given scenario.
   The first constraint that must hold in admissible transport plans is defined by:
   $$\forall i = 1 \ldots n : start(action_i) \neq dest(action_i).$$
   The rationale here is that both types of actions that can be carried out as plan steps, i. e., empty relocation from one storage facility to another and execution of a delivery, comprise a non-circular movement of the truck in question. Thus, a single plan step might not consist of a round-trip or of a rest or waiting period at a particular storage facility.
2. Another constraint ensures that the tour specified by a valid plan is contiguous, which means that short cycles are precluded by this constraint thus acting as a sub-tour elimination constraint:
   $$\forall i = 1 \ldots (n-1) : dest(action_i) = start(action_{i+1})$$
   where $action_i$ denotes the tuple elements of a plan as defined in equation (7). It is, however, possible for transport tours to revisit certain locations since loops are allowed by the formulation. For instance, let $Loc_a, Loc_b \in SF$, then:
   $$plan_3 = (\, Del(Loc_a, Loc_b), Empty(Loc_b, Loc_a), Del(Loc_a, Loc_b)\,)$$

is an admissible plan with a first delivery from A to B, followed by an empty return trip and another delivery from A to B.

The value of a particular transport plan as defined above is thereby determined as follows, based on equation (3):

$$val(plan^n) = \sum_{i=0}^{n-1} val(act_{i+1}) \cdot (n-i)^{\alpha} \tag{8}$$

where $val(act) = val(order, t) - cost(start(act), dest(act))$ iff $act \in Deliveries$. For $act \in EmptyRides$, it holds that $val(order, t) = -cost(start(act), dest(act))$.

Equation (8) shows that the value of the complete transport plan is a weighted sum of the values of the respective plan steps $action_i$. The parameter is thereby used to determine a specific weighting scheme. For instance, if $\alpha = 0$, all plan steps are given equal weight in the calculation of the value for the complete plan.

3.6 Employing LEM3 for Transport Planning

In the presented research, transport agents use LEM3 to search plan space. Since LEM3 is a multipurpose system for evolutionary optimization, its application for a particular planning problem requires setting the optimized fitness function as well as representation (search) space. The problem definition for the application of LEM3 in the presented context multi-agent planning incorporates:

1. The storage facility where the truck for whom the planning is being conducted for is located at the time of planning,
2. A complete list of storage facilities where transport orders may be pending (i. e., this list comprises both such facilities where pending orders are momentarily in stock and those which are unused at the moment), and
3. The size of the plan horizon.

Although LEM3 allows for the defining of domain-specific knowledge to help in the optimization process, for simplicity these features have not been used in the presented research.

Due to the reduced problem definition, candidate plans constitute what has been referred to as *plan skeleton* rather than a fully-fledged transport plan. The concept can be formalized and related to the definition of proper transport plans in Equation (7) as follows, assuming a plan horizon of size n:

$$\begin{aligned} planSkel(plan^n) &= (\, dest(action_1), dest(action_2), \ldots, dest(action_n)\,) \\ &\equiv (\, storage_1, storage_2, \ldots, storage_n\,) \end{aligned} \tag{9}$$

A plan skeleton is thus a n-tuple of storage facilities. However, with regard to the planning problem, only a sub-set of the set of all plan skeletons of size n is admissible in terms of compliance with the following constraints:

1. $\forall i = 1 \ldots (n-1) : storage_i \neq storage_{i+1}$ since all actions in proper plans involve a relocation of the truck between two distinct storage facilities, either via an empty ride or a proper delivery of a container.

2. $pos(truck) \neq storage_1$ where $pos(truck), storage_1 \in SF$. In particular, the storage facility where the truck is located at planning time must not be identical with the first storage facility in the plan skeleton.

For the plan optimization application, the aforementioned constraints have been implemented as part of the problem-specific LEM3 integration. As a consequence, after the creation of new candidate solutions, LEM3 is enabled to detect the number of constraint violations and filter inadmissible solutions immediately.

Since without additional processing these plan skeletons do not describe directly any particular plan which can be evaluated, an unequivocal conversion into a proper transport plan needs to be established. This conversion is possible because of two assumptions. First, transport plans always have as origin the current location of the associated truck, which corresponds to a storage facility ($pos(truck)$). In addition, as the transport agent seeks to maximize its financial balance for each plan step, it is rational to choose the most profitable action alternative using a function $bestAct : SF \times SF \rightarrow Deliveries \cup EmptyRides$, which accepts transport end points as input. The function returns either the most profitable real transport order, or, as a fall-back if no orders with the specified transport endpoints currently exists in the system, the empty drive order. Based on a plan skeleton, the corresponding plan is:

$$plan^n = (\, bestAct(pos(truck), storage_1), \ldots, bestAct(storage_{n-1}, storage_n) \,) \tag{10}$$

The transformation from plan skeletons, which constitute the plan suggestions created iteratively by LEM3 into proper candidate plans (from the point of view of the planning agent), is a mandatory prerequisite in order to apply the domain-specific weighting function. This weight function is used to evaluate candidates and thus drives LEM's search in the space of possible plans.

In order to directly calculate the value of a suggested plan skeleton, the following equation (11) can be applied:

$$\begin{aligned} value(planSkeleton^n) &= balance(bestAct(pos(truck), storage_1)) \cdot n^{\alpha} \\ &+ \sum_{i=1}^{n-1} [balance(bestAct(storage_i, storage_{i+1})) \cdot (n-1)^{\alpha}] \end{aligned} \tag{11}$$

The function $balance : SF \times SF \rightarrow$ Euro takes the current order value (which for all orders that have been waiting for some time is higher than the initial order price, cf. Section 3.2) and subtracts operation costs of these orders.

Once plans have been evaluated, LEM3 checks stopping criteria (reached desired value of plan or the maximum length of evolution is reached).

Although the described problem seems to be dynamic optimization [5], it is static from the optimization algorithm perspective. During the LEM execution the fitness function does not change. However, an interesting extension to the proposed method would be to use highly performing previous plans to seed initial population for LEM in the next simulation step. The rationale for the method is that in many cases the plans do not change over time, and reusing old plans may speed-up optimization. This extension, however, has not been used in this study.

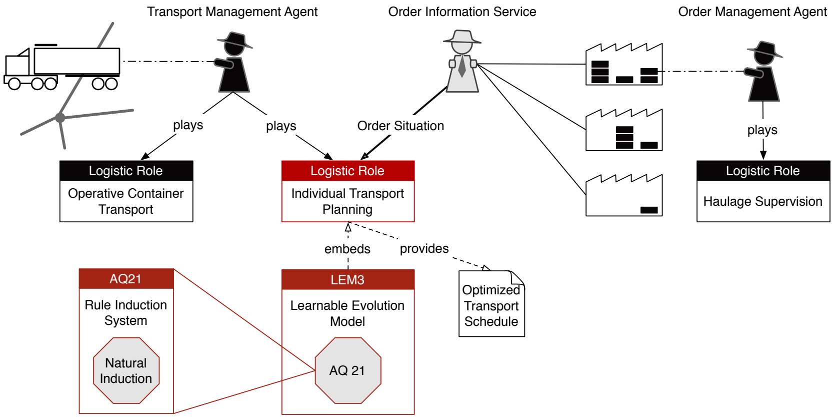


Fig. 2: Agent types for the realization of autonomous container on-carriage.

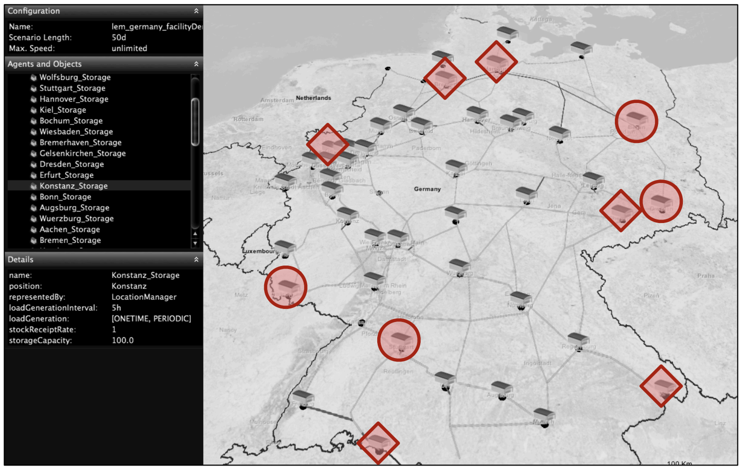
4 System Implementation

Transport management and order management (cargo) agents described in the previous section, have been implemented within a multiagent-system (MAS). Additionally, the system includes the third type of agent, an order information service agent (OIS), which maintains a company-wide database of pending transport orders. An outline of the system architecture is presented in Figure 2. This section briefly describes the system architecture. However, its technical details are out of the scope of this paper and can be found in the technical report [36].

The physical system implementation was done within the PlaSMA multiagent simulation system. PlaSMA (**Pla**tform for **S**imulations with **M**ultiple **A**gents) provides a distributed multiagent-based simulation and demonstration system based on the FIPA-compliant Java Agent Development Framework JADE [3]. The system was developed at the University of Bremen within the CRC 637 ”Autonomous Cooperating Logistic Processes”.[4] Although PlaSMA is a general purpose system, applicable in diverse simulation domains, its primary application domain is transport logistics. A detailed description of this simulation environment is provided in [35].

In the environment, all three types of agents communicate using the FIPA Query Interaction Protocol. At this point there is no direct communication between transport and order agents. All communication is done through order information service agent (OIS). When an order management agent wants its container to be delivered, it registers with the OIS. Transport management agents can gain access to the list of orders querying OIS. In response, the OIS informs the truck agent about pending transport orders. Once a shipment contract is established with a transport management agent to deliver the container, the transport order is unregistered and taken off the list of active orders. The data structure which is provided by the OIS is a hash map which uses tuples of

[4] PlaSMA is available at: `http://plasma.informatik.uni-bremen.de`



Configuration
Name: lem_germany_facilityDe
Scenario Length: 50d
Max. Speed: unlimited

Agents and Objects
- Wolfsburg_Storage
- Stuttgart_Storage
- Hannover_Storage
- Kiel_Storage
- Bochum_Storage
- Wiesbaden_Storage
- Bremerhaven_Storage
- Gelsenkirchen_Storage
- Dresden_Storage
- Erfurt_Storage
- Konstanz_Storage
- Bonn_Storage
- Augsburg_Storage
- Wuerzburg_Storage
- Aachen_Storage
- Bremen_Storage

Details
name: Konstanz_Storage
position: Konstanz
representedBy: LocationManager
loadGenerationInterval: 5h
loadGeneration: [ONETIME, PERIODIC]
stockReceiptRate: 1
storageCapacity: 100.0

Fig. 3: Traffic network with storage facilities employed in experiments. Those not marked explicitly do not have inbound customer orders. Storages marked with a diamond have a larger, those marked with a circle a lower amount of inbound orders.

transport endpoints as keys. Stored for each key is then a priority queue of orders with the transport endpoints specified by the respective key. Priority is thereby determined by the value (prioritization) of the transport orders.

5 Simulation Experiments and Evaluation

To thoroughly evaluate the performance of transport management agents employing LEM3 for route planning relative to the baseline, both strategies were tested with the freight forwarder MAS within PlaSMA simulations.

5.1 Setup of the Simulation Experiments

Traffic Network: A realistic traffic network represented as a directed graph has been used as a common basis for the simulation experiments. The network covers the area of Germany (Figure 3). It contains 359 nodes and 1.044 edges. The nodes comprise in addition to pure traffic junctions and path subdivisions (309 nodes) the major cities of Germany (50 nodes). The edges constitute transport routes. These represent a significant part of the German motorway network (750 edges) and, to a lesser degree in order to connect motorway sections or cities in the motorway network, federal roads (152 edges) and inner-city roads (28 edges).

Distribution and Configuration of Storage Facilities: Modeling of the logistics infrastructure comprises resources that are initially stationary, in particular the allocation of stationary storage facilities at the cities modeled in the traffic network. For the scope of the conducted experiments, each city hosts exactly one facility whose relevant modeling parameters comprise its overall storage capacity, uniformly specified as 100 container units, and simulation-specific parameters to control the external intake of new orders (for details, cf.[36, p. 23]). For the presented experiments, 10 out of 42 storage facilities were configured with an external intake of transport orders. These have been chosen due to their geographic location, such that a reasonable distribution of orders could be established. The experiments were made up of two experiment series which differ in the amount of orders created at high-intake storage facilities, i.e., creation of orders every six hours (experiment series 1) or every five hours (experiment series 2)[5]. Due to space constraints, this article focuses on findings from the latter series.

Configuration of the Forwarder Transport Fleet: Besides the configuration of storage facilities and their characteristics, it was also necessary to configure the transport fleet for the freight forwarder. The experiments feature a homogeneous fleet of 16 trucks, which can transport a single container at a time. Three configurations of the transport management agents were employed: first, a configuration where all trucks are managed by baseline transport agents; second, a configuration featuring LEM-enabled transport agents; and a third 'competitive' scenario, where eight trucks respectively are managed by agents of either type. That last scenario was chosen specifically to analyze the emerging effects of intermixing transport planning strategies of different complexities.

Model Parameterization: In the following, with regard to the conducted experiments, invariant parameterizations for the multiagent-based planning approach are specified.

First, it is necessary to determine both the fixed costs c_0 for operation of a semi-trailer truck and the variable operating costs r per kilometer, that are used in Equation (4) on page 8 in Section 3.2. These parameters have been chosen as c_0=110,00 € and r=1,42 €. The length of a transport tour, i. e., the $distance(order)$ from pickup to delivery point is measured in kilometers[6]. The dampening factor d also used in Equation (4) is set to 1 for full and 0.25 for empty containers. These values have been experimentally selected. For instance, the initial price for transporting a *full* container from Bremerhaven to Bremen (64 km) amounts to 200,88 €.

For the time-dependent internal revaluation of transport orders calculated with Equation (5) on page 8 (Section 3.2), the constant weight w_c has been

[5] In both cases, orders are created every 12 hours for low-intake facilities. Details are provided again in [36, p. 23]

[6] The distances between vertices in the transport network used in simulation experiments are based on Google StreetMap distances which have been entered as attributes of edges in the graph. Thus, even though the employed transport network uses a simplified model of the German motorway system, the distances between modeled track segments are a close approximation of the real distances.

selected as follows: $w_c(order_{full})$=0.3 for full and $w_c(order_{empty}) = 0.05$ for empty containers. The power α is selected as 2.0. These values have been chosen experimentally to achieve stale system behavior. For example, the *value* of an order for the transport of a full container which was left waiting 10 hours exceeds the real order price by about 30,00 €. However, if the container is left waiting for 2 days (48 hours), its value already will its original price by 691,20 €.

For the computation of a particular transport plan in Equation (8) on page 11 (Section 3.5), the weighting factor for respective plan steps has been chosen as $\alpha = 2.0$. As a consequence, the initial steps of the plan are given a much higher weight. This is due to the fact that the transport agents reconsider their current transport plan each time they reach another storage facility during their pickup and delivery tours. Thus it is expected, that plans are changed on a regular basis, which increases the importance to optimize particularly the plan steps immediately ahead.

With regard to the parameterization of the LEM3 sub-system, the decision was rendered to run the initial experiment with default parameters. Thus, the greedy baseline approach does not compete with a LEM3-based route-planning, which has been specifically tuned for the task at hand. It is thus possible to focus on a standard off-the-shelf LEM performance for the first iteration of experiments. Details of all LEM3 parameters are described in the program's user's guide [37, Section 2.2, pp. 6].

5.2 Experimental Results

This section presents results from three individual experiments. Each is thereby made up of ten repeated simulation runs. Each run simulates the freight forwarder's operation over a time period of 60 days. The experiments have been conducted with the PlaSMA system in roughly ten hours of real time on a PC with a 2.8 GHz Intel Core i7 CPU (quad-core) running Ubuntu 10.04.

The evaluation is based on a system of key performance indicators (KPIs), collected by the transport management agents (20 KPIs), the order management agents (7 KPIs) and the order information service agent (3 KPIs). The presentation of the experiment results comprises different tiers. The first takes a financial view on the operations of the freight forwarder from a global company perspective. In the second tier, focus is shifted to the pickup times for containers associated with the transport orders. The incentive here was to develop a sense for the processing times given different order inflow scenarios. The conducted evaluation considers both statistics and additional plots of pending orders that need to be handled. The final tier of the analysis is concerned with the distribution of truck operations, i. e., empty rides and deliveries.

Fleet-Level Financial Analysis: When considering the fleet balance (Fleet Bal. ($\mu_s \pm \sigma_s$) in Table 1) first, the direct comparison of the performance of the heterogeneous transport fleets shows a significantly higher mean overall revenue

Table 1: Financial results of the employment of different transport management approaches, measured in €/60 days ($\mu_s \pm \sigma_s$).

€/60d	Mixed Fleet	LEM Fleet	Baseline Fleet
LEM3 Route Planning			
Σ Costs	$606.931,42 \pm 415,90$	$1.213.175,1 \pm 930,35$	–
Σ Gains	$1.096.546,4 \pm 15.878,41$	$2.079.214,0 \pm 10.981,84$	–
Σ Balance	$489.614,98 \pm 15.462,51$	$866.038,94 \pm 10.051,49$	–
p.A. Balance	$61.201,87 \pm 3.856,59$	$54.127,43 \pm 4.415,25$	–
Baseline Order Selection			
Σ Costs	$609.229,46 \pm 920,48$	–	$1.212.536,1 \pm 885,16$
Σ Gains	$966.137,90 \pm 10.248,10$	–	$1.920.946,1 \pm 22.450,69$
Σ Balance	$356.908,44 \pm 9.327,62$	–	$708.409,97 \pm 21.565,53$
p.A. Balance	$44.613,56 \pm 5.062,45$	–	$44.275,62 \pm 4.347,73$
Fleet Bal.	$846.523,42 \pm 24.790,13$	$866.038,94 \pm 10.051,49$	$708.409,97 \pm 21.565,53$

when using the planning approach, i. e., 866.039 € vs. only 708.410 € for the baseline approach. The standard deviation – even though still residing at ∼3% is notably higher for the baseline approach (i. e., 10.051 € for the planning fleet and more than double that value, namely 21.566 € for the baseline fleet). The data for costs and benefits (gains), which is also provided in Table 1 shows that this increase in the standard deviation is caused by more variation in the revenue generated by handling transport orders. Therefore, the data seems to suggest that the planning approach exhibits better stability in a more demanding market situation. Further, it should be noted that the planning approach generates slightly higher costs than the baseline approach which are, however, compensated by the higher revenues.

When considering the competitive scenario where both the baseline and the planning approach were employed in equal proportion, the mean overall revenue of 846.523 € comes very close to the 866.039 € for the homogeneous planning fleet. Compared to the results measured for a scenario with a lesser order generated throughout, the mixed fleet yields even more notable positive results in a scenario with a comparatively strong inflow of transport orders. However, the effect of an increased standard deviation compared to a fleet operated by planning transport agents alone which has been observed before is evident once more in the data presented in Table 1.

Similarly to the previously discussed set of experiments, it is the case that in the competitive scenario the transport management agents which employ LEM-based planning on average significantly outperform their less provident counterparts with a mean revenue of 61.202 € compared to 44.614 € and a lower standard deviation of 3.857 € compared to 5.062 €.

More interesting than the presented raw numbers, is a comparison of the per agent performance between the homogeneous and the mixed setting. Here, the data shows that the baseline agents retain their performance values with respect to mean revenue while the standard deviation rises, from 4.348 € up to 5.062 €. The planning agents, however, manage to thrive in a situation with less peer competition and the insertion of baseline agents to compete against. This is documented by the fact that the per agent revenues rise significantly from a mean of 54.127 € up to 61.202 € while at the same time the standard deviation is reduced from 4.415 € down to 3.857 €.

Table 2: Waiting times until a container is picked up.

Unit: *hours*	Mixed Fleet	LEM Fleet	Baseline Fleet
Pickup wait, full ($\mu_s \pm \sigma_s$)	$12,597 \pm 11,899$	$7,926 \pm 8,706$	$40,820 \pm 24,855$
Pickup wait, empty ($\mu_s \pm \sigma_s$)	$90,029 \pm 33,761$	$53,832 \pm 30,718$	$141,564 \pm 46,675$

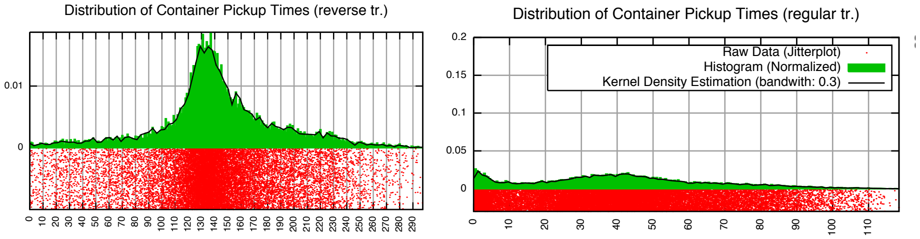


Fig. 4: Pickup times for 10 simulations with a *baseline* fleet.

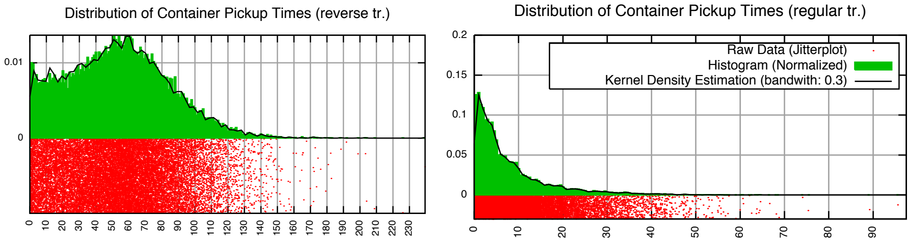


Fig. 5: Pickup times for 10 simulations with a *planning* fleet.

Pickup Times: The data in Table 2 and the supplementary plots in Figures 4 and 6 show that the LEM-based planning approach leads to a significant reduction of waiting times both for the pickup of containers for regular and reverse transports.

The plots of the distribution of pickup times depict vastly different characteristics. In Figure 4, it is shown that for the baseline approach, the distribution of pickup times is quite regular with only two shallow peaks. The first shallow peak is around minimal pickup times (orders handled just-in-time), and the second peak around the measured mean. Both peaks are also reflected in the respective cumulative distribution function. For the planning case, by contrast, there is only a single peak at ∼ 1 hour, which is the followed by a very steep incline that leads to 95 % of the regular transports being picked up within 30 hours and thus before the mean measured for the baseline approach. The surplus value generated through the employment of a planning approach is thus evident in the plots. What is also rendered obvious in the plot for the planning transport fleet is the long tail of the distribution due to a very low number of outlier cases for whom the pickup takes significantly longer than for the vast majority of cases. This effect has already been described in the analysis of the first set of experiment and is also replicated for the case of reverse transports. It is thus a general side-characteristic of the planning approach used in the experiments, which calls for in-depth research in order to enhance the approach for future experiments.

When considering the case of a mixed transport fleet, the plot in Figure 6 shows that the insertion of planning agents has the effect of ablating the second peak in the

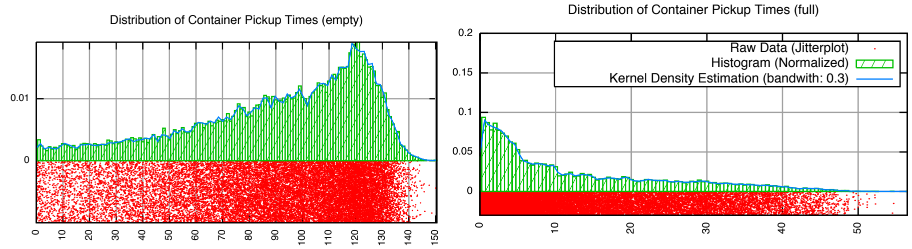


Fig. 6: Pickup times for 10 simulations with a *mixed* fleet.

baseline-only distribution and accumulating more weight at the initial peak. Ultimately, this brings the distribution closer to the planning-only case. It is also shown, that the effective value range of the distribution is cut in half.

As a next step of the analysis, focus is shifted from the examination of regular transport towards the reverse transports. Here, the data presented in Table 2 shows that in the baseline case, the mean pickup time for reverse transport is now $141,56\,h$ – an increase of 43 % from experiments with lower order inflow as documented in [36, Section 4.4.1,ṗp.26], with the standard deviation growing significantly as well. The plot in Figure 4 shows also, that in the face of the higher order inflow, the characteristics of the distribution of pickup times for reverse transports deviates from its equivalent in a low order situation as examined in [36, pp. 26]. The latter was characterized by a single significant peak around 120 hours with highly uneven slopes, the left featuring a particular long outlet while the right measured only a fraction thereof. In Figure 4 however, the single peak of the distribution is shifted to the right and the width of both slopes, left and right, amounts to more than 120 hours. What is more, the right outlet bears an even greater weight than the left. This data is an indication that the number of orders which needed to be handled in this set of experiments exceeded the capacities of the baseline approach. This assessment is backed by the ever growing stock levels and pending orders within the system (cf. Figure 7).

The effect of combining both approaches to equal fractions in the mixed transport fleet can be succinctly described as follows according to the plot in Figure 7c: While retaining the single peak at about 120 hours also found in the baseline case, the majority of the slope from that case is cleared away, the respective weight of the distribution being transferred to the left slope. As noted in the analysis of the first set of experiments, the mixture of approaches also helps to suppress the undesirable outlier cases which were once again observed when employing only planning transport management agents.

Stock Levels and Active Orders: In contrast to the evaluation, with respect to the total revenue, the pickup times, or the handling operations which are all based on statistical key measures, the analysis of both the inventory levels at active storage facilities and the total amount of active orders within the system is based on paradigmatic plots in Figure 7, sampled from single simulation runs.

For the inventories of the storage facilities in the planning case, the experiment data shows that low inventory levels can be maintained throughout the simulation runs. Specifically, stock levels max out at 6–8 containers at a time. The plot for the baseline approach in the top left of Figure 7 paints a different picture. In this case, the transport

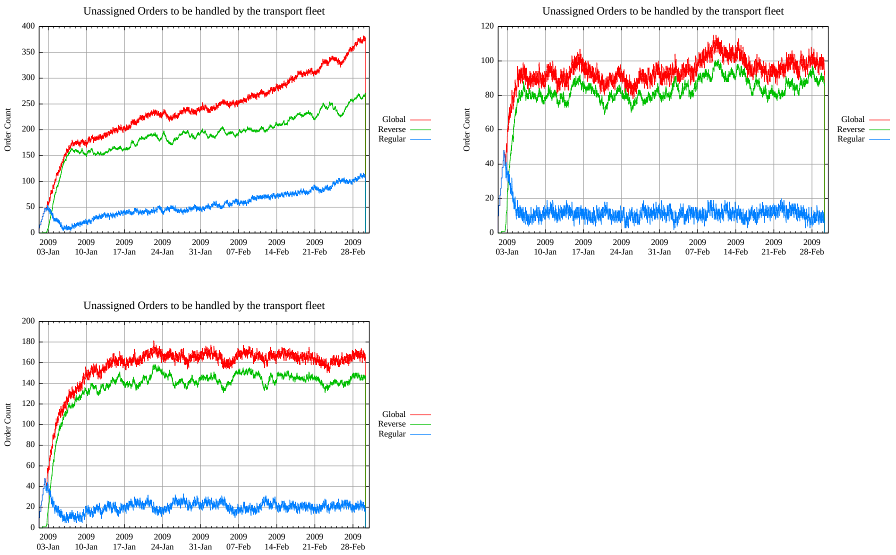


Fig. 7: Inventory Levels of pending orders at *active* storage facilities for the greedy transport management approach (*top left*), the mixed transport management (*bottom left*), and the LEM3-based transport planning (*top right*). For all scenarios, the plotted results are taken from the first out of ten simulation runs.

fleet fails to manage the amount of orders introduced into the system. This leads to a linear growth in stock levels. This finding is backed by the companion plot of the global number of pending transport orders within the system: the steady growth of orders is replicated for both regular and reverse transports. Taking also into account the findings from the preceding section, it can be concluded that the applied load on the forwarding agency was too much and could not be handled effectively when employing the baseline approach to transport planning.

It is interesting to note that the mixed transport fleet, in which baseline and planning transport management agents' work side-by-side, leads to a stable system where both the stock levels at the individual storage facilities and the total amount of pending transport orders tune in on a certain level. However, the plot of the stock levels suggests significantly increased peak amplitudes compared to the planning-only fleet. In addition, although the total amount of pending transport orders levels off after the initial phase of the simulation, it does so at higher levels, i. e., at ∼20 regular transport orders (compared to ∼15 in the planning case) and ∼140 reverse transport orders (compared to ∼90).

With regard to pending orders and inventory levels, the superiority of the planning approach to operative transport planning can be affirmed. While this result is not surprising, with the presented experiments it could be shown where the baseline approach used in the experiments has its limits. However, these limits may be extruded by mixing in planning transport management agents into a baseline transport fleet.

Table 3: Transport operations for trucks managed by different transport agents.

	Mixed Fleet	LEM Fleet	Baseline Fleet
LEM3 Transport Operations			
Σ Deliv $(\mu_s \pm \sigma_s)$	$2.478,90 \pm 17,06$	$4.301,30 \pm 5,88$	–
Σ Deliv$_{\text{full}}$ $(\mu_s \pm \sigma_s)$	$1.231,20 \pm 12,93$	$2.194,50 \pm 1,92$	–
Σ Deliv$_{\text{empty}}$ $(\mu_s \pm \sigma_s)$	$1.247,70 \pm 17,60$	$2.106,80 \pm 5,29$	–
Σ Fraction Full (μ_s)	$0,497$	$0,510$	–
Σ Fraction: All (μ_s)	$0,808$	$0,655$	–
Σ Length Del. $(\mu_s \pm \sigma_s)$ [km]	$1.074.057 \pm 8.437$	$1.974.968 \pm 12.396$	–
Baseline Transport Operations			
Σ Deliv $(\mu_s \pm \sigma_s)$	$1.740,70 \pm 12,80$	–	$3.824,70 \pm 80,48$
Σ Deliv$_{\text{full}}$ $(\mu_s \pm \sigma_s)$	$954,80 \pm 12,54$	–	$2.074,20 \pm 22,40$
Σ Deliv$_{\text{empty}}$ $(\mu_s \pm \sigma_s)$	$785,90 \pm 15,18$	–	$1.750,50 \pm 58,53$
Σ Fraction: Full (μ_s)	$0,549$	–	$0,542$
Σ Fraction: All (μ_s)	$0,562$	–	$0,574$
Σ Length Del. $(\mu_s \pm \sigma_s)$ [km]	868.476 ± 6.993	–	$1.759.706 \pm 30.745$

Transport Operations: The data in Table 3 shows that the planning approach leads to a significant increase in the overall amount of successfully operated container transports $(4.301,30 \pm 5,88)$ compared to the baseline approach $(3.824,70 \pm 80,48)$. Even though a higher number of both regular and reverse transports are handled by the planning fleet, the data also suggests a tendency of this approach towards a reduced fraction of regular transports (51,0 % compared to 54,2 %).

In addition to the superiority, with regard to the raw amount of handled transports and an slightly increased total length of delivery tours, the fraction of container transports on all truck operations is in the average notably higher for the planning transport fleet (65,5 %) than for the baseline (57,4 %). When comparing these results with their counterparts measured in experiments with a lower order inflow (cf. [36, Section 4.4.1, pp. 24]), the data suggests that given a higher amount of orders in the system, the planning fleet can substantially improve its truck utilization. To be more precise, the fraction of 47,2 % of transports on all truck operations, when running the same experiment presented here with high intake storage facilities receiving new orders only every 6 and not every 5 hours, is brought up to 65,7 %. The data raises the assumption, to be affirmed or disproved by further experiments that the planning fleet performs particularly well in scenarios with a high order inflow. However, the data acquired from the experiments also shows that the goals of high capacity utilization, increasing the freight forwarders profitability, and low pickup times, improving the quality of offered services, are related such that one is faced with a multi-criterial optimization problem.

Shifting the focus from the analysis of the homogeneous settings towards the mixed case with an equal amount of both agent types one can notice the following. In the data presented in Table 3 is the significant difference in overall operated container transports for both fleets. Both with regard to regular and reverse container transports, the planning fleet clearly outperformed the baseline fleet, resulting in $2.478 \pm 17,06$ deliveries for the former vs. only $1.740,70 \pm 12,80$ deliveries for the baseline fleet. More interesting than the plain amounts of deliveries are the results for the respective fractions of regular transports on all transport operations and of transports on all truck operations. If only considering data from the mixed scenario, it shows that the mean fraction of deliveries on all truck operations is higher for the planning sub fleet (80, 8%) than for the baseline sub fleet (56, 2%). This suggests that the former sub fleet actually profits from the competitive setting at the expense of the baseline sub fleet. A related result is that the planning fleet captures a significant amount of individually less profitable reverse transports not carried by the baseline fleet. However, as documented in Table 1, this does not have a negative effect on the revenue generated by the planning fleet. On the contrary, it documents the positive effect of planning in a competitive setting with regard to both truck utilization and profitability.

6 Related Work

A significant body of research on variants of the vehicle routing problem has been published by researchers tackling the problem from the perspective of such diverse fields as operations research [11, 23, 14, 4], distributed artificial intelligence including in particular multi-agent systems [43, 34], and mimetic computation [19]. The vehicle routing problem itself is, for instance, defined in [7]. A recent extensive survey on variants of VRP and approaches to solve them were compiled by Parragh and colleagues in [26]. Many authors focused on solving a dynamic version of VRP. In [16, p. 5], Larsen defines the dynamic VRP as follows: In dynamic VRP, "*not all information relevant to the planning of the routes is known by the planner when the routing process begins*", and "*information can change after the initial routes have been constructed*". This setting is also given in the scenario that has been addressed, from a hybrid multi-agent and machine learning perspective, in this paper.

In [16], the author proposed an approach to DVRPTW (DVRP with time windows) using an insertion heuristic. Kilby used this heuristic to solve the DVRP without time windows [15]. The work presented in [15] is mainly a study on how the cost of the system is affected by the degree of dynamicity. This problem was also treated by Montamenni [24], who examined the DVRP and proposed a solving strategy based on the Ant Colony System paradigm and the time slot concept used in [15]. In [10], Gendreu proposed a solution approach based on a tabu search algorithm.

The DVRP solving methods highlighted above are exclusively appropriate to a special definite problem, which usually contain only one kind of change. The proposed system however, is designed to also handle situations that are characterized by the frequent occurrence of events which constitute a change in the momentary order situation. Due to such frequent changes, an adaptation of transport plans must be considered in everyday business. The proposed approach refrains from computing complete transport routes for the employed transport vehicles. Instead, transport plans are compiled at the local level of individual agents with a certain planning horizon, and are routinely adapted each time the truck reaches another storage facility. This local optimization makes the system more robust and capable of better handling disruptions than global optimization methods. Although many other optimization methods can be used in place of the LEM3 system, previous research has demonstrated LEM's very good performance [21, 41].

7 Conclusion and Future Work

This paper presented a methodology for decentralized order pickup and delivery planning by autonomous agents, which use the Learnable Evolution Model to create optimized sequences of container deliveries. In the presented model each order, be it a regular or reverse transport, is assigned an intra-organizational prioritization which is updated by dedicated order management agent as a function of pickup waiting time. Experimental results indicate that the method performs better when compared to a baseline approach, in which each agent selects the best container at its given location. This result was obtained when greedy and LEM agents were simulated separately, and in a mixed scenario that combined both types of agents. All experiments were performed with the multiagent-based simulation system PlaSMA [35].

The future research will address several open aspects: in-depth analysis of optimization processes for LEM3 invocations, and normalization in calculation of the value of transport plans with respect to haulage distances. In addition, potential benefits of the introduction of a structured problem representation for LEM3 as suggested in [40] will be investigated. For instance, a single plan step could be represented as a tuple $\langle \mathrm{Loc}_a, action, \mathrm{Loc}_b \rangle$. Also, as pointed out in Section 3.5, LEM3 excels in comparison to other systems for evolutionary optimization when additional domain knowledge can be provided to guide and focus the search process. Therefore, it is worthwhile to identify such knowledge in the given application domain and provide it to the LEM system. It is then possible to measure the influence of the aforementioned knowledge on plan quality and thus on the logistics performance of individual transport management agents and the freight forwarder.

Acknowledgment The presented research was partially funded by the German Research Foundation (DFG) within the Collaborative Research Centre 637 "Autonomous Cooperating Logistic Processes: A Paradigm Shift and its Limitations" at the University of Bremen, Germany. Development of LEM3 in the GMU Machine Learning and Inference Laboratory has been supported in part by the U.S. National Science Foundation grants. Current research on the Learnable Evolution Model and its applications is supported by the U.S. National Institute of Standards and Technology grants.

References

1. Baluja, S.: Population Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning. Technical Report (CMU-CS-94-63), Carnergie Mellon University (1994)
2. Baluja, S., Caruana, R.: Removing the Genetics from the Standard Genetic Algorithm. Proceedings of the 12th International Conference on Machine Learning pp. 38–46 (1995)
3. Bellifemine, F., Caire, G., Greenwood, D.: Developing Multi-agent Systems with JADE. Wiley Series in Agent Technologies, Wiley Inter-Science (2007)
4. Bent, R., Van Hentenryck, P.: Scenario-based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers. Operations Research pp. 977–987 (2004)
5. Branke, J., Kaussler, T., Schmidt, C., Schmeck, H.: A multi-population approach to dynamic optimization problems. In: In Adaptive Computing in Design and Manufacturing. pp. 299–308. Springer (2000)

6. Chia, J., Goh, C., Tan, K., Shim, V.: Memetic informed evolutionary optimization via data mining. Memetic Computing 3, 73–87 (2011), http://dx.doi.org/10.1007/s12293-011-0058-7, 10.1007/s12293-011-0058-7
7. Christofides, N., Mingozzi, A., Toth, P.: The Vehicle Routing Problem. In: Christofides, N., Mingozzi, A., Sandi, C. (eds.) Combinatorial Optimization, pp. 315–338. John Wiley & Sons, Inc. (1989)
8. Davidsson, P., Henesey, L., Ramstedt, L., T
"ornquist, J., Wernstedt, F.: An Analysis of Agent-based Approaches to Transport Logistics. Transportation Research part C: emerging technologies 13(4), 255–271 (2005)
9. Gehrke, J.D., Wojtusiak, J.: Traffic prediction for agent route planning. In: Bubak, M., van Albada, G., Dongarra, J., Sloot, P. (eds.) 8th International Conference on Computational Science 2008, vol. 3. pp. 692–701. Springer, Berlin (2008)
10. Gendreau, M., Guertin, F., Potvin, J.Y., Taillard, E.: Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching. Transportation Science 33(4), 381–390 (1999)
11. Gendreau, M., Guertin, F., Potvin, J., Séguin, R.: Neighborhood Search Heuristics for a Dynamic Vehicle Dispatching Problem with Pick-ups and Deliveries. Transportation Research Part C: Emerging Technologies 14(3), 157–174 (2006)
12. Hülsmann, M., Scholz-Reiter, B., Windt, K. (eds.): Autonomous Cooperation and Control in Logistics. Springer Berlin / Heidelberg (2011)
13. Hülsmann, M., Windt, K. (eds.): Understanding Autonomous Cooperation and Control in Logistics. The Impact on Management, Information and Communication and Material Flow. Springer (2007)
14. Hvattum, L., Lokketangen, A., Laporte, G.: Solving a Dynamic and Stochastic Vehicle Routing Problem with a Sample Scenario Hedging Heuristic. Transportation Science 40(4), 421–438 (2006)
15. Kilby, P., Prosser, P., Shaw, P.: Dynamic VRPs: A Study of Scenarios. APES Technical Report APES-06-1998 (September 1998)
16. Larsen, A.: The Dynamic Vehicle Routing Problem. Ph.D. thesis, Technical University of Denmark (2000)
17. Llorà, X., Goldberg, D.: Wise Breeding GA via Machine Learning Techniques for Function Optimization. In: Cantú-Paz, E., Foster, J., Deb, K., et al. (eds.) Genetic and Evolutionary Computation — GECCO 2003, LNCS, vol. 2723, pp. 1172–1183. Springer Berlin / Heidelberg (2003)
18. Lozano, J.A., Larrañaga, P.: Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, Genetic Algorithms and Evolutionary Computation, Vol. 2, vol. 2. Kluwer Academic Publishers (2002)
19. Machado, P., Tavares, J., Pereira, F., Costa, E.: Vehicle routing problem: Doing it the evolutionary way. In: Proceedings of the Genetic and Evolutionary Computation Conference. p. 690. Morgan Kaufmann Publishers Inc. (2002)
20. Michalski, R.: Learnable Evolution: Combining Symbolic and Evolutionary Learning. In: Proceedings of the Fourth International Workshop on Multistrategy Learning (MSL'98). pp. 14–20. Desenzano del Garda, Italy (June 11–13 1998)
21. Michalski, R.: Learnable Evolution Model: Evolutionary Processes Guided by Machine Learning. Machine Learning 38, 9–40 (2000)
22. Michalski, R., Wojtusiak, J., Kaufman, K.: Progress Report on the Learnable Evolution Model. Reports of the Machine Learning and Inference Laboratory MLI 07–2, George Mason University, Fairfax, VA (2007)

23. Mitrovi-Mini, S., Laporte, G.: Waiting Strategies for the Dynamic Pickup and Delivery Problem with Time Windows. Transportation Research Part B: Methodological 38(7), 635–655 (2004)
24. Montamenni, R., Gambardella, L., Rizzoli, A., Donati, A.: A new Algorithm for a Dynamic Vehicle Routing Problem Based on Ant Colony System. In: Second International Workshop on Freight Transportation and Logistics (2003)
25. Mühlenbein, H., Paaß, G.: From Recombination of Genes to the Estimation of Distributions I. Binary Parameters. In: Voigt, H.M., Ebeling, W., Rechenberg, I., Schwefel, H.P. (eds.) Parallel Problem Solving from Nature — PPSN IV, LNCS, vol. 1141, pp. 178–187. Springer Berlin / Heidelberg (1996)
26. Parragh, S., Doerner, K., Hartl, R.: A Survey on Pickup and Delivery Problems. Journal für Betriebswirtschaft 58, 81–117 (2008)
27. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (1986)
28. Reynolds, R.G., Zhu, S.: Knowledge-based Function Optimization Using Fuzzy Cultural Algorithms with Evolutionary Programming. IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics : A Publication of the IEEE Systems, Man, and Cybernetics Society 31(1), 1–18 (Jan 2001)
29. Reynolds, R., Peng, B.: Cultural Algorithms: Modeling of How Cultures Learn to Solve Problems. In: 16th IEEE International Conference on Tools with Artificial Intelligence. pp. 166–172. IEEE Comput. Soc (2004)
30. Reynolds, R.G.: An Introduction to Cultural Algorithms. In: Sebald, A.V., Fogel, L.J. (eds.) Proceedings of the Third Annual Conference on Evolutionary Programming. pp. 131–139. Rvier Edge, New Jersey: World Scientific Publishing (1994)
31. Saleem, S., Reynolds, R.G.: Function Optimization with Cultural Algorithms in Dynamic Environments. In: Proc. of the Workshop on Particle Swarm Optimization 2001. Purdue School of Engineering and Technology, Indianapolis, IN (2001)
32. Santana, R., Larrañaga, P., Lozano, J.: Research topics in discrete estimation of distribution algorithms based on factorizations. Memetic Computing 1, 35–54 (2009), http://dx.doi.org/10.1007/s12293-008-0002-7, 10.1007/s12293-008-0002-7
33. Schuldt, A.: Multiagent Coordination Enabling Autonomous Logistics. Springer-Verlag, Heidelberg, Germany (2011)
34. Vokřínek, J., Komenda, A., Pěchouček, M.: Agents towards vehicle routing problems. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1 - Volume 1. pp. 773–780. AAMAS '10, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2010)
35. Warden, T., Porzel, R., Gehrke, J.D., Herzog, O., Langer, H., Malaka, R.: Towards ontology-based multiagent simulations: The plasma approach. In: Bargiela, A., Azam Ali, S., Crowley, D., Kerckhoffs, E.J.H. (eds.) 24th European Conference on Modelling and Simulation (ECMS 2010). pp. 50–56. European Council for Modelling and Simulation (2010)
36. Warden, T., Wojtusiak, J.: Learnable Evolutionary Optimization in Autonomous Pickup & Delivery Planning: A Scenario, System Architecture and Initial Results. TZI Technical Report 55, Center for Computing and Communication Technologies, Universität Bremen (September 2010)
37. Wojtusiak, J.: The LEM3 Implementation of Learnable Evolution Model: User's Guide. Reports of the Machine Learning and Inference Laboratory MLI 04–5, George Mason University, Fairfax, VA (November 2004)
38. Wojtusiak, J.: Handling Constrained Optimization Problems and Using Constructive Induction to Improve Representation Spaces in Learnable Evolution Model. Ph.D. thesis, College of Science, George Mason University, Fairfax, VA (November 2007)

39. Wojtusiak, J.: Data-Driven Constructive Induction in the Learnable Evolution Model. In: Proceedings of the 16th International Conference on Intelligent Information Systems. Zakopane, Poland (June 16–18 2008)
40. Wojtusiak, J., Michalski, R.: The LEM3 System for Non-Darwinian Evolutionary Computation and Its Application to Complex Function Optimization. Reports of the Machine Learning and Inference Laboratory MLI 05–2, George Mason University, Fairfax, VA (October 2005)
41. Wojtusiak, J., Michalski, R.: The LEM3 Implementation of Learnable Evolution Model and Its Testing on Complex Function Optimization Problems. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2006). Seattle, WA (Juli 8–12 2006)
42. Wojtusiak, J., Michalski, R., Kaufman, K., Pietrzykowski, J.: The AQ21 Natural Induction Program for Pattern Discovery: Initial Version and its Novel Features. In: Proceedings of the 18th IEEE Internation Conference on Tools with Artificial Intelligence, Washington, DC. pp. 523–526. IEEE Computer Society, Los Alamitos, CA (November 2006)
43. Zeddini, B., Temani, M., Yassine, A., Ghedira, K.: An Agent-Oriented Approach for the Dynamic Vehicle Routing Problem. Advanced Information Systems for Enterprises, 2008 International Workshop on 0, 70–76 (2008)