Technical Report 55

Learnable Evolutionary Optimization in Autonomous Pickup & Delivery Planning

A Scenario, System Architecture and Initial Results

Tobias Warden
Janusz Wojtusiak
(George Mason University, Fairfax, VA)

TZI, Universität Bremen

TZI-Bericht Nr. 55
2010

Universität Bremen

TZI-Berichte

Herausgeber:
Technologie-Zentrum Informatik und Informationstechnik
Universität Bremen
Am Fallturm 1
28359 Bremen
Telefon: +49-421-218-7272
Fax: +49-421-218-7820
E-Mail: info@tzi.de
http://www.tzi.de

ISSN 1613-3773

Contents

Learnable Evolutionary Optimization in Autonomous Pickup & Delivery Planning

A Scenario, System Architecture and Initial Results

Tobias Warden
warden@tzi.de
Center for Computing and Communication Technologies, University of Bremen

Janusz Wojtusiak
jwojt@gmu.edu
Machine Learning and Inference Laboratory
George Mason University

Abstract

This report describes an evolutionary approach to distributed planning by transport agents. These agents, representing trucks, autonomously decide on their transport orders by interacting with cargo agents representing containers. The agents use a guided evolutionary computation method, called the learnable evolution model, to create transport plans and render optimized decisions on which cargo to transport. The model is implemented within the PlaSMA multiagent simulation platform, and evaluated experimentally.

Keywords: PlaSMA, Multiagent-based Simulation, Autonomous Logistics, Learnable Evolution Model, Evolutionary Computation

1 Introduction

Unprecedented growth of transportation and logistics networks in recent years calls for a shift in planning and control methods. Centralized planning approaches are gradually becoming less suited to handle the complexity of entire logistics networks. This is specifically true with the frequent occurrence of unexpected events which constitute disruptions to existing plans. When centralized approaches are used, often entire global plans need to be re-computed which may be very time consuming, and, in practice, infeasible. To remedy this situation, a highly distributed agent-based approach to logistics has been proposed [HW07]. This approach assumes autonomy in decision-making of agents that are acting on behalf of logistic entities. As a consequence of the renunciation of a centralized planning strategy the approach has been shown to be robust in the sense that it allows for the handling of unexpected irritations at the local level of individual agents.

The problem considered in this report is defined from the perspective of a freight forwarding agency which needs to handle a continuous dynamic flow of transport orders for freight containers. New orders arrive at the system on a regular basis at a certain sub set of storage facilities that are geographically distributed over the federal territory of Germany. The model assumes, that two subsequent transport orders need to be handled for each container introduced in the scenario. The first order comprises the transport of a full container without further distinction of containers with respect to enclosed contents. The second, subsequent order comprises the return transport of the empty container after unloading at its initial delivery target facility. Therefore, the system incorporates aspects of reverse transportation.

In order to handle the transport orders which have already been successfully acquired from customers, the freight forwarding agency is to employ exclusively 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*. For the scope of the experiments presented in this report, the simplifying assumption has been introduced that these trucks can be operated on a continuous 24/7 basis[1].

This report introduces an agent-based distributed control system which autonomously handles the processing of incoming transport orders. The main contribution of this work however lies in the adoption of a domain-agnostic guided evolutionary optimization method for the local formation of pickup and delivery plans for individual trucks in the freight forwarding agency's transport fleet.

The remainder of this report is structured as follows: The next section provides details with regard to order (re)valuation and strategies employed for selection of transport orders. For the latter, the greedy order selection implemented in the baseline transport agents is sketched briefly. Subsequently, the focus is shifted to a detailed description of the transport plan optimization approach adopted by the group of planning transport agents. Section 3 then gives a brief overview of the implementation of the freight forwarding agency as multiagent system and discusses in particular the adoption of the learnable evolution model in the context of the presented application scenario. Hereafter, Section 4 presents experimental results from multiagent based simulation with the PlaSMA simulation system[2] [GOB07, WPG+10]. Three configurations of transport agents have been tested in scenarios that are differentiated by the size of the order inflow over the course of simulation runs. These configurations are: homogeneous deployment of greedy transport management agents, homogeneous deployment of planning transport management agents, and balanced, heterogeneous deployment with equal-sized sub-groups of both types of agents operating in the same scenario. An extensive analysis of the simulation results shows the advantages of the planning approach. Section 5 concludes with a discussion of results.

2 Autonomous Control of Container On-Carriage

The presented multiagent-based autonomous control system for a freight forwarding agency adopts a system design consisting of two parts that jointly determine the emerging system behavior. These are 1) an intra-system approach to the valuation (prioritization) of pending transport orders which are active in the system, and 2) local selection of orders for handling deliveries with particular trucks in the transport fleet.

These two components can be understood as falling into separate areas of responsibility. While the intra-organizational order valuation is part of order management, the operative order handling, which for the purposes of this report subsumes order selection, is part of transport management. This distinction is introduced here as it is reflected in the agent-based implementation of the autonomous control system. In particular, the system design comprises two primary classes of logistic agents which each handle one of the aforementioned management tasks. Transport agents each authoritatively act on behalf of a single truck and manage its intra-company order selection, acquisition and subsequent operative order handling. These agents rely on an adequate valuation of pending transport orders as this constitutes the foundation of their respective approach to order selection, be it a simple greedy approach as employed by the baseline transport agent version or a more sophisticated planning approach. Order management agents, which constitute the second primary agent class, are short-lived representatives for pending orders and as such, are responsible for the valuation - or prioritization - of their respective transport order.

The following sections introduce the mechanics of order valuation by the order management agents and the mechanics of order selection and planning by the transport management agents. One important aspect and to some extent prerequisite of the series of experiments to be described in Section 4 has been to establish an effective interplay of the deployed valuation and planning strategies leading to the emergence of a stable behaviour of the whole freight forwarder multiagent-system under the considered order inflow scenarios. In particular, part of the evaluation of conducted experiments refers to the correlation of system parametrization and the waiting times for the assignment of pending transport orders.

[1] Future experiments will account for down-times that are either due to service and shift changeover or mandated by law and other constraints.

[2] PlaSMA web site: http://plasma.informatik.uni-bremen.de

2.1 (Re-)Valuation of Transport Orders

Since the implementation of transport management agents that act for the freight forwarder assumes that transport agents select the most profitable orders, other orders with lower associated *contractual order value* may never be selected for transport. From the global point of view of the freight forwarding agency, such a system behavior evidently violates contract agreements with customers that the company has committed to obey when the orders where acquired in the first place.

Therefore, the forwarding agency requires an effective mechanism which is designed to ensure a timely handling of *all* pending transport orders. As the orders which are considered in this report do not yet specify a contractually fixed delivery time frame, *timely* in this context denotes that the system should be designed such that all orders should be handled in what would be considered a sensible global time window, e. g. 48 hours. The following text outlines an approach which is based on an autonomous revaluation of transport orders, which emanates from the original real valuation of the orders in terms of monetary value. However, while retaining the monetary pretense, the values whose calculation is introduced hereafter should be understood solely as a means of intra-organizational prioritization of orders. In particular, the approach does not assume additional monetary flows between process stakeholders, i. e. the customers placing orders and the freight forwarding agency.

Having clarified the notion of value, the employed revaluation approach can described in detail.

The transport management agents internally consider the order value, which can be understood as an intra-company priority, for order selection, such that, assuming a sufficiently large number of transport agents and thus managed means of transport, each order will eventually be selected for transport as its value increases with time. The order value of an order to transport a particular container is calculated as follows:

$$value(order, t) = price(order) + prior^{+}(order, t) \tag{2.1}$$

$value(order, t)$ is computed as sum of a first component $price(order)$ which constitutes the initial order price agreed between customer and forwarder (cf. Equation (2.2)) and a second component $prior^{+}(order, t)$ for the oder revaluation based on the period of time the cargo has already been waiting to establish a transport contract (cf. Equation (2.3)).

The initial order price is thereby computed as follows:

$$price(order) = d \cdot (c_0 + r \cdot distance(order)) \tag{2.2}$$

The constant c_0 thereby denotes fixed costs for the operation of a semi-trailer truck, while the constant r denotes variable operating costs per kilometer. In the experiments described in Section 4, 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[3]. Finally, in order to accommodate the two considered 'types' of containers which are associated with transport orders – those with regular content and those which are due to be returned empty to their original storage facility – the dampening factor d has been introduced. It is set to 1 for full and 0.25 for empty containers.

So, for instance, the initial price for transporting a *full* container from Bremerhaven to Bremen (64 km) amounts to 200,88 €.

$prior^{+}(order, t)$ is a function which increases the value of a container depending on the time it has already been waiting. The idea is that 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 significantly such that due to their addition in Equation (2.1), such containers receive preference over those which have only recently been prepared for shipping. While in general the function $prior^{+}(order, t)$ can be very sophisticated and use prediction of travel times, time windows used in contracts, penalties, and the like, a comparatively basic version shown below has been used for the initial set of experiments.

$$prior^{+}(order, t) = w_c(order) \cdot t^{\alpha} \tag{2.3}$$

[3]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

The constant weight w_c has thereby been selected as follows: $w_c(order_{full})$=0.3 for full and $w_c(order_{empty})$ = 0.05 for empty containers. The power is selected as 2.0.

So, 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 exceeds its original price by 691,20 €.

2.2 Selection of Transport Orders

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. Although this report concentrates on a particular transport planning approach based on an evolutionary optimization method, a simple non-planning type of transport agent which effectively employs a greedy order selection strategy has been implemented as a baseline for the measurement of transport management performance. Sections 2.2.1 and 2.2.2 outline both order selection strategies.

A common assumption, which is deemed reasonable within the bounds of a single freight forwarding agency, is that the transport management has access to the entire momentary order situation which comprises 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.

2.2.1 A Baseline Approach to Order Selection

The baseline transport management agents which 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 and identify the most profitable order as calculated by

$$\begin{aligned} bestOrder(t) &= \underset{\text{Orders}(t)}{\arg\max}[\,\overbrace{(value(order,t)}^{\text{gains}} \\ &- \underbrace{(cost(start(order), dest(order)) + cost(pos(truck), start(order))))}_{\text{costs}}\,] \end{aligned} \tag{2.4}$$

where 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 real pickup tour is required in order to subsequently handle the selected transport order.

The choice of the next transport order to be handled is repeated each time a truck that is managed by a greedy transport management agent 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, as without further restrictions, the agents are actually allowed to reconsider their previous delivery choice, either due to the fact that the originally desired orders have been assigned over the course of the pickup tour or other more lucrative options have materialized. Thus, in order to prevent that greedy order selection leads to a behaviour where the agents primarily keep performing pickup tours without ever getting to execute the 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 behaviour.

2.2.2 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 alternative, the options being for each decision point 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 thereby 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 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 planning horizon of size n. Formally, such a plan is defined as:

$$plan^n = (action_1, action_2, \ldots, action_n) : action_i \in Deliveries \cup EmptyRides \tag{2.5}$$

$Deliveries$ thereby 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. $EmptyRides$ by 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 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 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.

   As a first constraint that must hold in admissible transport plans is that:

   $$\forall i = 1 \ldots n : start(action_i) \neq dest(action_i). \tag{2.6}$$

   The rationale here is that hitherto 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 may neither consist of a round-trip nor of a rest or waiting period at a particular storage facility.

2. A further constraint ensures that the tour specified by a valid plan is contiguous which means that short cycles are precluded by this constraint which thus acts as a sub-tour elimination constraint.

   $$\forall i = 1 \ldots (n-1) : dest(action_i) = start(action_{i+1}) \tag{2.7}$$

   where the $action_i$ denote the tuple elements of a plan as defined in Equation (2.5).

   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 short example provided with the last constraint thereby shows, that in contrast to common traveling salesman formulations, admissible plans in our context may but not necessarily need to be round trips beginning and ending at a dedicated home depot.

The value of a particular transport plan as defined above is thereby determined as follows, based on Equation (2.1):

$$value(plan^n) = \sum_{i=0}^{n-1} value(action_{i+1}) \cdot (n-i)^{\alpha} \tag{2.8}$$

$$value(action) = \begin{cases} value(order, t) - cost(start(action), dest(action)) : action \in Deliveries \\ -cost(start(action), dest(action)) : action \in EmptyRides \end{cases}$$

Equation (2.8) 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 concrete weighting scheme. For instance, if $\alpha = 0$, all plan steps are given equal weight in the calculation of the value for the complete plan.

For the scope of the experiments however, the value 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 in particular the plan steps immediately ahead.

2.2.3 The Guided Evolutionary Approach to Planning

The learnable evolution model (LEM) is an evolutionary optimization method that employs machine learning to direct the evolutionary process [Mic98, Mic00]. Specifically, LEM creates general hypotheses indicating regions in the search space that likely contain optimal solutions and then instantiates these hypotheses to generate new candidate solutions. In order to apply machine learning, LEM creates two groups of individuals that are respectively high- and low-performing according to the fitness function being optimized. These individuals 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 hypotheses are then instantiated to create new candidate solutions. In the final step, a new population is assembled from old and new individuals, and the process is repeated until stopping criteria are met.

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 2.1 on the following page. In addition to components found in standard evolutionary computation methods, such as generation of an initial population, evaluation of individuals, and selection of individuals, LEM3 includes several novel components. It dynamically selects one or more innovation methods to create new individuals. These methods are:

1. *Learn & Instantiate*, the aforementioned main mechanism for creating new individuals in LEM3;
2. *Adjust representation*, to change the discretization of numeric attributes;
3. *Probe*, to apply traditional operators such as mutation and crossover;
4. *Search locally*, to apply a user-defined local search method;
5. *Randomize*, to add to the current population a number of randomly created individuals, or restart the evolutionary process.

One of the major novelties of LEM3 is the ability to automatically adjust the representation space through constructive induction [Woj07, Woj08].

Theoretical and experimental work indicates that LEM is particularly suitable for optimization problems in which the fitness evaluation is costly. This is because of the trade-off between significantly shorter evolution length [Mic00, WM06], and more complex learning and instantiation when compared to simple operators used in evolutionary computation. Moreover, the use of machine learning to guide evolutionary computation extends the applicability of LEM. For example, because of the use of AQ21 as a learning module in LEM3, it is able to handle optimization problems naturally described using different types of attributes (nominal, structured, ordinal, cyclic, interval, ratio, and compound) and background knowledge provided to the learning program [Woj09].

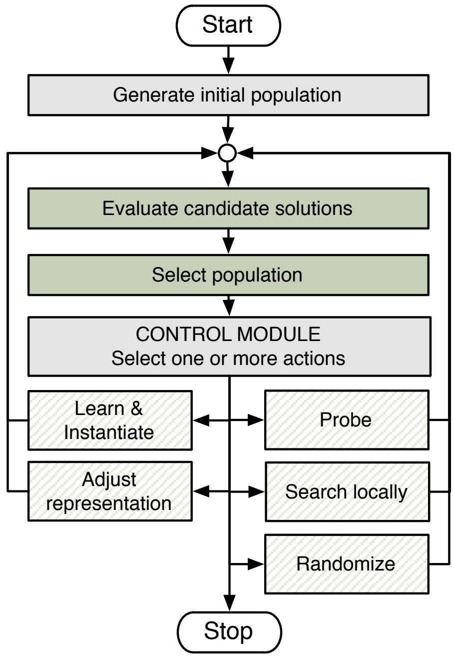

Figure 2.1: A top-level flowchart of LEM3, adapted from [MWK07].

In addition to general LEM implementations, a class of LEM-based systems for heat exchanger optimization has been developed. These include the ISHED system for optimizing evaporators [YDWK10] and ISCOD system for optimizing condensers [DYKM04, KM00]. These specialized systems combine LEM's learning and instantiation operators with specialized probing operators that are specifically designed to work with heat exchangers. Based on the ISHED and ISCOD systems, Michalski and Kaufman proposed a general LEMd methodology for optimizing complex systems [MK06].

2.2.4 Employing LEM3 for Transport Planning

In the presented research, transport agents use LEM3, the newest implementation of the learnable evolution model [WM06], to search plan space. Since LEM3 is a multipurpose library for evolutionary optimization, its application for a particular planning problem in a target application-domain presupposes both an adequate representation of the planning problem and the provision of a problem-specific weighting function which allows LEM3 to determine the significance of candidate solutions which are created over the course of the evolutionary optimization process. The problem definition for the application of LEM3 in the presented context currently incorporates:

1. the storage facility where the truck for whom the planning is conducted 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 effectively unused at the moment), and
3. the size of the plan horizon.

It is very basic for the time being as the additional encoding of further domain knowledge, such as the structure of the transport network where the passed storage facilities are geographically located, has been postponed in order to focus initially on the engineering of a stable and extensible coupling of LEM3 and the system components using the library. Based on the problem definition, as given by a list of possible locations to visit, and background knowledge, LEM3 searches for the best plan as illustrated in Figure 2.1. It starts with an initial population of candidate plans, which is randomly generated. Due to the aforementioned reduced problem definition these candidates constitute what has been referred to as *plan skeleton* rather than a fully-fledged transport plan. This concept can be formalized and related to the definition of proper transport plans in Equation (2.5) as follows, assuming a plan horizon of size n:

$$\begin{aligned} planSkeleton(plan^n) &= (dest(action_1), dest(action_2), \ldots, dest(action_n)) \\ &\equiv (storage_1, storage_2, \ldots, storage_n) \end{aligned} \tag{2.9}$$

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 application in plan optimization, 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 these plan skeletons without additional processing 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 due to 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 $bestAction : SF \times SF \rightarrow Deliveries \cup EmptyRides$ which accepts transport end points as input and returns either 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. Then, based on a plan skeleton, the corresponding plan is:

$$plan^n = (bestAction(pos(truck), storage_1), \ldots, bestAction(storage_{n-1}, storage_n)) \tag{2.10}$$

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 which is used to evaluate candidates and thus drives LEM's search in the space of possible plans. The approach which is currently applied with the integration of LEM3 is basically to out-source the weighting function from the library to the planning agent as user of the library which is also equipped with the required domain-specific knowledge to execute the candidate valuation.

In order to directly calculate the value of a suggested plan skeleton, the following Equation (2.11) can be applied:

$$\begin{aligned} value(planSkeleton^n) &= balance(bestAction(pos(truck), storage_1)) \cdot n^\alpha \\ &+ \sum_{i=1}^{n-1} [balance(bestAction(storage_i, storage_{i+1})) \cdot (n-1)^\alpha] \end{aligned} \tag{2.11}$$

The function $balance : SF \times SF \rightarrow$ Euro thereby takes the current order value (which for all orders that have been kept waiting for some time is higher than the initial order price, cf. Section 2) and subtracts the operation costs for the execution 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).

In *learning mode*, LEM3 employs machine learning in a three-tier process of creating new candidate plans. First, high- and low-scoring candidate plans are selected according to their value to serve as positive examples (H-Group) and negative examples (L-Group) for learning, respectively. Then, machine leaning

Listing 1 Example for the basic ontological modelling of the employed transport agents and their respective managed truck.

```
<!-- http://plasma.informatik.uni-bremen.de/owl/lem_germany.owl#TZI_Logistics -->
<tlo:Company rdf:about="#TZI_Logistics">
    <tlo:isOwnerOf rdf:resource="#Truck_TZI_001"/>
    ...
</tlo:Company>
...
<!-- http://plasma.informatik.uni-bremen.de/owl/lem_germany.owl#TZI_Operator_001 -->
<tlo:SoftwareAgent rdf:about="#TZI_Operator_001">
    <rdf:type rdf:resource="&tlo;ObjectAgent"/>
    <tlo:operatesFor rdf:resource="#TZI_Logistics"/>
    <tlo:representsObject rdf:resource="#Truck_TZI_001"/>
</tlo:SoftwareAgent>
...
<!-- http://plasma.informatik.uni-bremen.de/owl/lem_germany.owl#Truck_TZI_001 -->
<lem_shared:LEMTruck rdf:about="#Truck_TZI_001">
    <rdf:type rdf:resource="&tlo;StorageFacility"/>
    <tlo:storageCapacity rdf:datatype="&xsd;float">1.0
    </tlo:storageCapacity>
    <trans:maximumVelocity rdf:datatype="&xsd;integer">100
    </trans:maximumVelocity>
    <tlo:maximumPossibleVelocity rdf:datatype="&xsd;integer">120
    </tlo:maximumPossibleVelocity>
    <tlo:hasOwner rdf:resource="#TZI_Logistics"/>
    <tlo:positionedAt rdf:resource="&graph;Kassel"/>
</lem_shared:LEMTruck>
```

is applied to induce a general hypothesis differentiating between these two groups of candidate plans. Specifically, LEM3 uses the AQ21 rule learning system [WMKP06] to create a set of rules describing high-value plans. Finally, the rules are instantiated to produce new candidate plans that are likely to have high value. In addition, LEM3 optionally employs a *probing mode*, in which candidate plans are created using traditional evolutionary operators such as mutation and recombination.

The detailed description of the algorithm and its specific elements is presented by Wojtusiak and Michalski [WM06, WM05], and Wojtusiak [Woj09, Woj07].

3 Multiagent-based Implementation with PlaSMA

The following section is dedicated to a description of the multiagent system (MAS) which represents the freight forwarding agency whose operative transport planning is in the focus of this report. The distinct agent types that comprise the multiagent system are introduced and their description is related with the order revaluation and transport planning approaches detailed in Section 2. The multiagent system has been implemented for analysis within the PlaSMA simulation system[4]. For background information on PlaSMA, the reader is referred to [WPG+10] and the PlaSMA user guide [GWB+10]. The following sub sections outline the design of the freight forwarding MAS as used in the experiments, including an in-depth description of the integration of the LEM3 system. Section 5.1 highlights architectural changes to the forwarder implementation in a future revision of the introduced multiagent system.

3.1 Transport Management Agents

As the first type of agent that constitutes the freight forwarder MAS, the transport management agent is designed to authoritatively act on behalf of one particular truck within the transport fleet of the forwarding agency such that there exists a one-to-one management relation between agent and physical

[4] PlaSMA web site: http://plasma.informatik.uni-bremen.de

object. This relation is kept invariant over time. Listing 1 on the preceding page outlines the ontological modelling of the freight forwarding company considered in this report, the modelling of one particular truck from the homogeneous transport fleet and finally the modelling of the transport management agent itself. Agents of this type are in the focus of this report as they decide on an individual basis which transport orders they would like to commit to and subsequently execute. Within this report two sub-types of transport agents have been realized which differ exclusively in their respective strategy for order selection. That is to say, that both 1) *greedy transport agents* which implement order selection according to the specification in Section 2.2.1 and 2) *planning transport agents* which leverage the learnable evolution model (cf. Section 2.2.3 on page 9) for the computation of multi-step transport plans as specified in Section 2.2.2, share a common behaviour architecture whose structure is depicted in Figure 3.1 on the next page. The top-level behaviour for theses agents is derived from the standard JADE finite state machine behaviour where individual behaviours that concern specific agent sub-tasks that frequently occur over the agent life-time, such as the acquisition of information about pending orders to be handled by the freight forwarder, are encapsulated in dedicated behaviours and the overall behaviour of the agent is specified by the result-dependent transitions between these behaviours (cf. the PlaSMA user guide [GWB+10, Chapter 5.5.4] for further details).

The design of the transport management agent can be conceptually partitioned into a *planning stage* and an *operative stage* which are traversed alternatingly. The transport management agents enter the planning stage whenever their current queue of operative tasks, such as performing an empty ride or a delivery, has been worked off. In such a situation, the agent actively needs to find a new engagement. To that end, it starts with an order acquisition process in which it pulls all order-related information that is required for order selection (cf. Section 2.2) from the company-owned order information service (OIS) which is described later in Section 3.3. The communication with the OIS thereby follows the *FIPA Query Interaction Protocol* [fIPAF02b]. The agent then enters its strategy-specific order selection behaviour which has been derived from a common planning behaviour base class (cf. Figure 3.1 on the following page). After a successful planning session which can be conducted without the need for further communication with external sources, the immediately following activity which may be either an empty ride or a delivery has been identified. In the latter case, the truck associated with the transport management agent needs to deliver a container from its current location to a target storage facility as specified by the transport order.

In the current version of the presented system concept, both types of transport agents adopt a lazy commitment strategy with respect to the handling of transport orders. This means, that the actual binding commitment to execute a particular transport order is consciously deferred until the truck arrives at the pickup point, i. e. the storage facility where the container the order refers to is currently taken to stock. This strategy was chosen in order to retain for the transport management agent the flexibility to reconsider and potentially revise earlier transport plans in case more profitable action alternatives arrive. Thus, before taking action, the order must of course been formally assigned to the transport agent. Thus, using a later commitment strategy, the transport agent initiates a *FIPA Propose Interaction Protocol* [fIPAF02a] with the responsible order management agent to have its order assignment confirmed and signed.

Once this operation is completed, the agent can switch from the planning back to the operative stage. Herein, the agent sees to it, that its current queue of operative task defined by its planning stage is carried out properly. This two-tiered behaviour architecture for the transport management agent has been chosen deliberately in order to promote reusability of significant parts of the agent code-base across a variety of scenarios. Even in scenarios where the planning stage of the transport agents needs to be exchanged due to, for instance, another scheme for order acquisition and planning such as an iterative multi-tier process, the operative stage may still be reused. Further versions of the transport management agent architecture will seek to allow an even greater flexibility in the implementation of the planning stage.

The operative stage of the transport management agent also entails all interaction with the simulation world model via appropriate world model actions, as described in [WPG+10]. The set of actions used entails the standard drive action distributed together with the PlaSMA system and additional load- and unload actions.

From a visual inspection of Figure 3.1 on the next page, it is rendered clear that the current generation of transport management agents acts independent from peers which are also active within the same

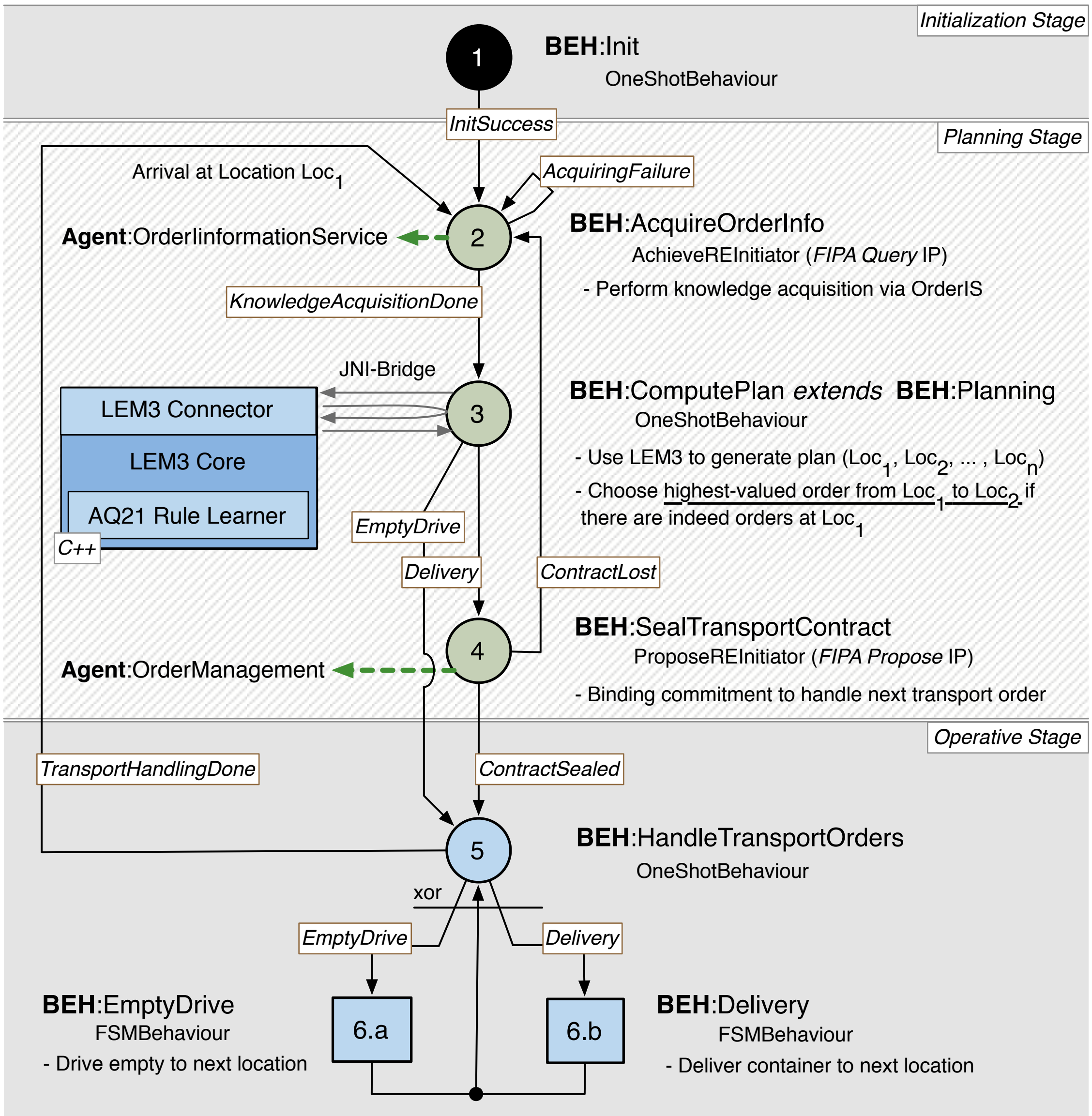

Figure 3.1: Behaviour specification for planning transport management agents.

environment. That is, event though multiple instances of transport management agents may - and in the experiments described in Section 4 do - operate under the umbrella of one and the same freight forwarding agency, these agents can for the time being be thought of as acting in their own best interest. It should be stated clearly that the focus of this report is to investigate the advantages of a planning group of transport agents compared to a non-planning base line. However, effective coordination schemes for the described transport agents are worth studying in the future.

Both implemented types of transport agents may be employed either exclusively for the management of the whole transport fleet or jointly in what may then be considered a competitive setting. Both have been employed for the experiments described in Section 4.

3.1.1 Integrating PlaSMA with the Learnable Evolution Model (LEM)

For the planning-enabled transport management agents, section 2.2.4 on page 10 already described the employment of the multi-purpose evolutionary optimization approach LEM for the specific task of creating transport plans for the trucks of the considered freight forwarding agency. This section will now

Listing 2 Definition of the generalized Rosenbrock function in LEM3 source code. Adapted from [Woj04, p. 21].

```
class rosenbrock_significance : public compute_significance
{
        double compute( Event const & );
};

double rosenbrock_significance::compute( Event const & event )
{
        int i;
        double re, val, val1;
        re = 0;

        event.check_value( 0, val1 );

        for( i = 0; i < event.positions()->size() - 1; i++ ){
                if( event.check_value( i + 1, val ) == true ) {
                        re += 100*(val-val1)*( val-val1 )+(val1-1)*(val1-1);
                        val1 = val;
                }else{
                        cout << "error computing significance" << endl;
                }
        }
        return 1 / ( re + 1 );
};
```

discuss the technical aspects of the coupling of the newest member of the LEM system, named LEM3, with PlaSMA agents.

The user guide for the LEM3 implementation of the Learnable Evolution Model [Woj04] offers two methods which may be applied to employ LEM3 as a general-purpose system in the context of a particular application. Both methods thereby involve the definition of a problem-specific fitness function. The first method relies on an external dedicated program which is used to calculate the fitness function and is due to communicate with LEM3 running as additional standalone program via structured text files. In the second method, the user extends the vanilla LEM3 code-base with an additional class which implements the custom fitness function as shown in Listing 2. This extra class is linked with LEM3 into one executable file. For the LEM3 integration considered for this report, the first option, i. e. a simplistic, file-based communication between standalone programs was considered as to inefficient due to the I/O-overhead, in particular in the light of application scenarios where large groups of transport management agent employ LEM-based planning within the same scenario. The second extension option was not directly applicable in the intended planning scenario as well. This is to some extent due to the technical constraint that the problem-specific significance function which is required for the transport planning problem is by no means as self-contained as for instance the paradigmatic significance functions shown in the LEM3 user guide (cf. [Woj04, p. 21]). It rather depends on domain-specific knowledge supplied by and an interpretation of candidate solutions (candidate plans) performed by the actual users of the LEM3 system in the application scenario, namely the respective transport management agents.

Therefore, a third methodology to embed LEM3 as a planning sub-system into PlaSMA agents has been adopted which retains the general extension scheme introduced by the former methods, that is to out-source the significance function from the main LEM3 code by basically 1) extending the class `compute_significance`, 2) provide the means to configure LEM3 runs using a custom problem description and parameterization, and finally 3) invoke the execution such carefully prepared LEM3 runs, thus obtaining the desired optimized transport plans.

The technical approach has been to compile the LEM3 C++ code base as a shared library which can be dynamically loaded by individual Java-based transport agents upon startup as a native library. Communication with this LEM3 library is thereby established via a custom-tailored API built on top of the Java Native Interface (JNI) [Lia99]. The realized Java/C++ bridge which couples agent and LEM3 code thereby reflects the distinct task areas which have already been outlined briefly in the

Listing 3 Native LEM3 invocation method used in the planning behaviour.

```
public native int runLEM(final String currentPosition,
                         final String[] locationsList,
                         String[] optimizedPlan);
```

preceding paragraph. As a new LEM3-API, it can be effectively distinguished in two sub-APIs for the actual LEM-based optimization of transport plans (*LEM3-PI*) and the preceding configuration of a LEM-invocation which is internally referred to as LEM3Run (*LEM3-Conf*). LEM3-PI comprises 1) a native method which can be called in the planning behaviour of LEM-enabled transport agents; and 2) a callback method which can be invoked from the C++ LEM3 side in order to determine the value of particular candidate plans. Thus, the API allows both calling LEM3 from the agent Java code and - entwined - call agent Java code from LEM3. LEM3-Conf additionally provides a set of getter function that can be used by LEM3 to query details of the LEM3 parameterization. That is, by means of the agent-specific configuration within the scenario configuration as shown in Listing 4 on the next page, it is possible to specify the most important parameters for the LEM3 system.

An invocation of the LEM3 sub-system is realized via a call of a native function within the planning behaviour of the transport management agents. Technically, the LEM3 binding on the Java side is realized via a dedicated class `LEM3Bridge` which encapsulates the native function declaration and exposes callback methods which later allow the LEM3 C++ code to retrieve configuration parameters and call the method which effectively constitutes the custom significance function. The signature for the latter function is thereby defined via a Java interface `LEM3Callback` which is implemented by the planning behaviour that employs the LEM3 sub-system. Upon its instantiation at the very beginning of the agent life cycle, a new `LEM3Bridge` instance is created which is passed the callback for the significance computation and an additional object of class `LEM3Configuration` which stores all the information required to prepare a concrete LEM3 run. In order to describe the LEM3 bridge in more detail, it is probably best to explain step-by-step. At the start of a LEM3 operation, the planning behaviour of the respective transport management agent calls the native method with the signature as shown in Listing 3.

It provides both the current storage facility for the managed truck and an array of storage facilities active within the considered scenario as primary input parameters. Both are encoded as plain string representations of the respective ontology individuals used in PlaSMA to define the physical world model. The additional array `optimizedPlan` is initially empty and is filled by LEM3 with the best computed candidate plan encoded as a plan skeleton as introduced in Section 2.2.4 on page 10. Once the native function has been invoked by the agent, a new `LEM3Run` is instantiated which represents the configuration for a LEM experiment. First an internal representation of the *problem domain* is constructed. The domain contains the names of the storage facilities as nominal values. The next step is to produce a number of *attributes*, one for each step in the candidate plans[5]. The next step in the preparation of the `LEM3Run` is its proper parameterization. In the standalone LEM3 executable detailed in the LEM3 user guide, parameters are passed as command line arguments. In the presented use of the system the respective parameters are rather set directly via the respective functions exposed by the `LEM3Run` object. For a complete list of regular LEM parameters, the reader is referred to the LEM3 user guide [Woj04]. For the concrete optimization problem considered in this report, only a sub set of available public parameters was used[6], namely:

`lem_population_size [50]` the overall *population size* of active candidate solutions (or rather candidate plans) (cf. LEM3 user guide [Woj04, p.6]);

`lem_no_children [30]` the *number of children* which are brought to life in each iteration (cf. LEM3 user guide [Woj04, p.6]);

`lem_no_generations [20]` the maximum number of generations which may be created in finding an optimized candidate solution (cf. LEM3 user guide [Woj04, p.6]);

`lem_selection_method [POPULATION_BASED]` the selection method for choosing members of the groups of high- and low-performing individuals (cf. LEM3 user guide [Woj04, p.9])

[5]These attributes are consequently labeled `Step_1` to `Step_n` for a candidate plan of length n.
[6]Default values are given in square brackets.

Listing 4 Example configuration for a LEM-enabled transport management agent with all optional LEM parameters redefined explicitly. Below, for comparison, a configuration for a simple greedy transport agent.

```
<MAPPINGS>
...
<SIM_OBJECT description="Transport Management Agent (1)" enabled="true">
    <CREATION_DATE>1230883200000<!-- Fri, 02 Jan 2009 08:00:00 GMT --></CREATION_DATE>
    <ONT_INSTANCE>TZI_Operator_001</ONT_INSTANCE>
    <CLASS> org.tzi.plasma.toolkit.truck.agent.Agent_TruckManagement</CLASS>
    <!-- Agent Mode Selection -->
    <ATTRIBUTE description="order selection strategy"
        key="run_mode" value="simulation_lem"/>
    <!-- LEM Parameterization -->
    <ATTRIBUTE key="lem_population_size" value="50"/>
    <ATTRIBUTE key="lem_no_children" value="20"/>
    <ATTRIBUTE key="lem_no_generations" value="30"/>
    <ATTRIBUTE key="lem_selection_method" value="FITNESS_BASED"/>
    <ATTRIBUTE key="lem_sel_not_included" value="SELNI_P"/>
    <ATTRIBUTE key="lem_random_events" value="0"/>
    <!-- Planning Parameterization -->
    <ATTRIBUTE description="Desired length of plan sequence generated by LEM"
        key="lem_plan_length" value="3" />
    <ATTRIBUTE description="Whether or not to log LEM performance to DB"
        key="lem_log_performance" value="true"/>
    <ATTRIBUTE description="Absolute file system path to LEM3 library"
        key="lem3_lib_path" value="~/SFB/scenarios/lem3_lib/Release/liblem3.jnilib"/>
</SIM_OBJECT>
<SIM_OBJECT description="Transport Management Agent (2)" enabled="true">
    <CREATION_DATE>1230883260000<!-- Fri, 02 Jan 2009 08:01:00 GMT --></CREATION_DATE>
    <ONT_INSTANCE>TZI_Operator_002</ONT_INSTANCE>
    <CLASS> org.tzi.plasma.toolkit.truck.agent.Agent_TruckManagement</CLASS>
    <!-- Agent Mode Selection -->
    <ATTRIBUTE description="order selection strategy"
        key="run_mode" value="simulation_greedy"/>
</SIM_OBJECT>
...
</MAPPINGS>
```

In order to have the chance to introduce a more explorative character to LEM's search of the plan space, it was further decided, also to consider the following non-public parameter:

`lem_random_events` [0] The number of random candidate solutions, for diversification of the search process (not documented in LEM3 user guide)

Besides these two sets of parameters which are used to shape the evolutionary search to be executed by the LEM system, the following more general parameters were introduced as well:

`lem_plan_length` the desired length of candidate plans/plan skeletons that are generate throughout the search process; and indirectly

`lem_random_seed` the random seed which passed to LEM3 before the start of the search for an optimized candidate plan.

All of the aforementioned parameters but the `lem_random_seed` can be defined by the experimenter on a per-agent basis via the configuration of the transport management agents in the XML-based scenario configuration (cf. the PlaSMA user guide [GWB+10, Chap. 4] for details). An example for such a configuration is presented in Listing 4. All of the LEM parameters are optional. If skipped in the configuration of a particular planning transport management agents, the LEM3 sub-system will be configured with default parameters. `lem_random_seed` is the only parameter which is effectively configured on a global basis within the scenario configuration. The relevant configuration excerpt is shown in Listing 5. Each PlaSMA agent may query its local container manager for this explicitly configured random seed value. It is worth mentioning here, that this seed is specific to particular simulation runs. That is, for simulations with multiple repeats as shown in the example, a new derived random seed is generated for each subsequent run.

Listing 5 Configuration of the initial global random seed for a particular scenario (line 8).

```
1  <SIMULATION xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2      xmlns:xi="http://www.w3.org/2001/XInclude"
3      xsi:noNamespaceSchemaLocation="config.xsd"
4      name="lem_germany_greedyfleet"
5          simStartTime="1230796800000"
6          repeats="10"
7          maxSimLength="50d"
8          seed="1230796800000">
9  ...
10 </SIMULATION>
```

All parameters that are relevant for a LEM run are stored in the `LEM3Configuration` object which is passed to the `LEM3Bridge` upon instantiation. The bridge then provided a set of callback functions (i. e. the LEM3-Conf API) such that after the initial problem definition LEM3 can actively query the calling transport agent for the desired configuration of the LEM run to be configured. Besides the configuration of the LEM3 system itself, it is also possible to configure the entailed AQ21 rule learning system [WMKP06], which as mentioned before, is employed in the context of LEM to discriminate low- and high-performing candidate plans and thus guides the evolutionary search process. However, for the scope of experiments presented in this report, a standard AQ21 parametrization has been employed.

The final step in the configuration process is to pass the desired problem-specific significance function to be used for the LEM run. For transport planning, the new class `route_significance` has been derived from the the abstract LEM base class `compute_significance`. However, in terms of functionality with regard to the actual computation of the significance of candidate plans, the C++ class constitutes only a thin wrapper which uses the evaluation function

Listing 6 Signature of LEM3 callback method used for valuation of candidate plans.

```
/** The evaluation function called by LEM
 *  @param plan The plan which needs to be evaluated.
 *  @return The value of the plan. */
double evaluatePlan(final String[] plan);
```

declared by the `LEMCallback` and implemented by the calling transport agent planning behaviour to calculate the value of the passed candidate plan or plan skeleton according the the specification presented in Section 2.2.4 on page 10.

The same section also presented prerequisites for admissible candidate plans which need to be enforced by the LEM3 sub system. The system does that by means of a filter process for newly generated candidate plans. To that end, the class `plan_significance` also overrides the default `compute_constraints` method which is passed a new candidate plan and is then by specification supposed to compute its number of domain-specific constraint violations. The filter for candidate plans consequently eliminates all plans for which any constraint violations have been detected. The concrete implementation of the `compute_constraints` method used for the experiments presented in this report is shown in Listing 7 on the next page.

3.2 Order Management Agents

Besides the transport management agents which have been described thus far, the second major type of agent which acts in the context of the freight forwarding agency, is the order management agent. Agents

Listing 7 Constraint computation on candidate plans in LEM3.

```
double route_significance::compute_constraints(Event const & event) {

        double constraint_violations = 0; // returns number of constraints violated

        string val1;

        // Step 1: Initial position should be distinct from first location in
        //         solution sequence.

        if(event.size() >= 1){
                event.check_value(0,val1);
                if(val1 == this->currentPosition){ constraint_violations+=1.0; }
        }

        string val2;

        // Step 2: Consecutive locations in the solution sequence should be distinct.

        for (uint32_t i = 0; i < event.size() - 1; i++) {
                event.check_value(i, val1);
                event.check_value(i + 1, val2);
                if (val1 == val2){ constraint_violations+=1.0; }
        }

        return constraint_violations;
}
```

of this type are short-lived dedicated managers overseeing both the initial and the reverse transport for particular containers. Both transports are thereby represented by self-contained transport orders. Order management agents are instantiated on an on-demand basis whenever a new container is introduced into the system.

They are responsible to register their respective transport orders with a company-wide order information service described in the following section. Subsequently, they control the revaluation of their transport orders as a function of waiting time, as described in Section 2.1, thus ensuring that their orders will be delivered eventually.

Besides actively managing the prioritization of their respective orders via revaluation, the transport management agents are passive in that they wait for transport management agents to initiate a transport contract negotiation via a *FIPA Propose interaction Protocol* [fIPAF02a]. When approached by a transport management agent, it is first checked within the scope of the interaction protocol whether appropriate storage space is currently available at the transport delivery site. This kind of information is acquired from an agent managing the respective storage in a dedicated *FIPA Request Interaction Protocol* [fIPAF02c] interlaced with the enclosing interaction protocol. In case storage space is available, it is reserved immediately which means that the prerequisites for haulage by the enquiring transport agent are fulfilled and a binding transport agreement is reached. If, however, the storage space at the destination site is currently unavailable, the inquiring transport management agent is given a negative response. Also, the transport management agent temporarily suspends its transport order from the list of pending orders held by the order information service of the transport agency and stalls the order revaluation. The order management agent than waits for a notification from the agent managing the destination site that space has become available once more before re-entering its order into the system and resuming normal operation.

During container transport, the load management agent has a monitoring behaviour which in future iterations of the freight forwarder implementation may be extended to supervise the transport progress of the managed container. So far, the transport management agents remains passive during transport. However, once the initial transport of the full container has been completed, the agent initiated the unloading of the container contents by request to the location management agent for the respective container location by means of an *FIPA Request Interaction Protocol* [fIPAF02a]. Once notification from the latter agent has been received, indicating that the unloading has succeeded, the transport management agent internally creates the follow-up transport order for the recirculation of said container. Besides the implications for order-revaluation outlined in Section 2.1, the handling of this new order is analogous to the first case.

3.3 Order Information Service

The order information service, in the following abbreviated as *OIS* constitutes a comparatively simple, reactive agent which provides a company-wide data base of pending transport orders that need to be handled by the forwarding agency.

If an order management agent wants its container to be delivered, it can register with the OIS as explained before. When a shipment contract with a transport management agent to deliver the container has been agreed upon, the aforementioned transport order will be unregistered and taken off the list. Transport management agents can gain access to this list - the momentary oder situation - by means of engaging into a FIPA Request Interaction Protocol [fIPAF02c]. In response, the OIS will inform the truck agent about the momentarily pending transport orders. The data structure which is provided by the OIS is a hash map which uses tuples of 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.

In the initial phase of creating the multiagent-based freight forwarding agency, it was a conscious design decision to indeed implement a dedicated agent as central point of contact for transport management agents that seek to obtain an overview of the global order situation of the freight forwarding agency as their starting point for individual active order acquisition. Since transport management agents, order management agents and the OIS are operated by one and the same company, privacy concerns can be neglected. On the other hand, the information acquisition process performed before each planning step is reduced to a single interaction with the OIS, rather than having to approach the load management agents individually.

3.4 Location Agent

The *location agent* manages *all* storage facilities within the scenario. Upon initialization, this agent creates dedicated handling behaviours for each location. The location management behaviour handles both registration and de-registration requests from load and truck agents when a truck or a container arrives at or departs from the location. New containers will be generated by the load generation behaviour. If a container was successfully delivered and returned, it must be removed from the scenario by the load consumption behaviour. The trans-loading process (unloading content from the container) is simulated by simply wasting some time in the trans-loading behaviour before responding to the request.

Besides acting as representative of storage facilities and thus partaking in the presented scenario as a proper simulation actor, the location agent additionally also plays the role of a load generator. In this role, it is responsible for the fabrication of containers in the simulations, their initial inventarization into the stock of storage facilities, the creation of the respective transport order and the transport management agent to manage the handling of this order.

4 Simulation Experiments and Evaluation

In order to thoroughly evaluate the performance of the transport management agents which employ LEM3 for their respective route planning relative to the greedy transport management agents as baseline, the multiagent-based freight forwarding implementation was tested with different transport planning strategies and two different order inflow scenarios within the multiagent-based simulation environment PlaSMA.

Listing 8 Paradigmatic ontological modelling for a storage facility located at the city of Bremen.

```
<!-- http://plasma.informatik.uni-bremen.de/owl/lem_germany.owl#Bremen_Storage -->
<tlo:StaticStorageFacility rdf:about="#Bremen_Storage">
    <!-- Asserted Concepts -->
    <rdf:type rdf:resource="&trans;ISOContainerStorage"/>
    <!-- Scenario-agnostic Modelling -->
    <tlo:representedBy rdf:resource="#LocationManager"/>
    <tlo:positionedAt rdf:resource="&graph;Bremen"/>
    <tlo:storageCapacity rdf:datatype="&xsd;float">100.0</tlo:storageCapacity>
    <!-- Scenario-specific Modelling -->
    <lem_shared:loadGeneration rdf:datatype="&xsd;string">PERIODIC
    </lem_shared:loadGeneration>
    <lem_shared:loadGeneration rdf:datatype="&xsd;string">ONETIME
    </lem_shared:loadGeneration>
    <lem_shared:stockReceiptRate rdf:datatype="&xsd;int">1
    </lem_shared:stockReceiptRate>
    <lem_shared:loadGenerationInterval rdf:datatype="&xsd;string">6h
    </lem_shared:loadGenerationInterval>
</tlo:StaticStorageFacility>
```

As a common basis for all simulation experiments, where experiment in this context refers to the batch-execution of multiple repetitive runs of a particular simulation with the same environmental setup, order inflow parameterization and forwarder configuration and parametrization, a basic but nevertheless realistic traffic network has been employed which is specified as an annotated directed graph. This graph, which is distributed as part of the PlaSMA releases, covers the federal area of Germany and contains 359 nodes and 1044 edges. The nodes comprise, besides pure traffic junctions and path subdivisions (309 nodes) the major cities of Germany (50 nodes). The edges constitute a traffic network which represents a significant excerpt of the German motorway network (750 edges). Federal roads (152 edges) and inner-city roads (28 edges) are implemented to a much lesser degree in order to connect motorway sections or cities to the motorway network.

4.1 Distribution and Configuration of Storage Facilities

Such a generic traffic network can be integrated into specific scenario specifications due to OWL's modularity and import mechanisms. Therefore, in modelling the scenario infrastructure for the simulation experiments in this report, it was rendered possible to concentrate on the specification of stationary logistic resources, in particular the allocation of static storage facilities at the cities modeled by the traffic networks graph. For the scope of the presented experiments, the assumption was made, that each city should host exactly one storage facility. Listing 8 shows the detail modelling of one paradigmatic storage facility which is located at Bremen. As the focus of the simulation experiments was on the analysis of the performance of the LEM3-based planning approach employed by the transport management agents and the system behaviour which emerges in respective settings rather than creating a highly detailed and accurate representation of a particular, real logistic environment, the decision was rendered to keep the modelling of the storage facilities generic with respect to specializations such as inland- or sea ports, pure distribution centers, manufacturing industry sites, warehouses run by end customers and the like.

Instead, the only relevant modelling parameters are the overall storage capacity of a particular storage facility which was specified as 100 container units for all storage facilities in the scenario, and scenario-specific parameters which are used to control the external intake of new orders on a per-facility basis.

`loadGeneration` specifies whether or not there exists an external intake of new transport orders. If an intake actually exists for the storage facility, it is possible to further specify the characteristics of the order intake. Allowed values are:

- `NONE` The storage facility does not feature an external intake of new orders. Thus, it may only act as a destination facility for full container transports and subsequently source facility for the respective empty reverse transports.

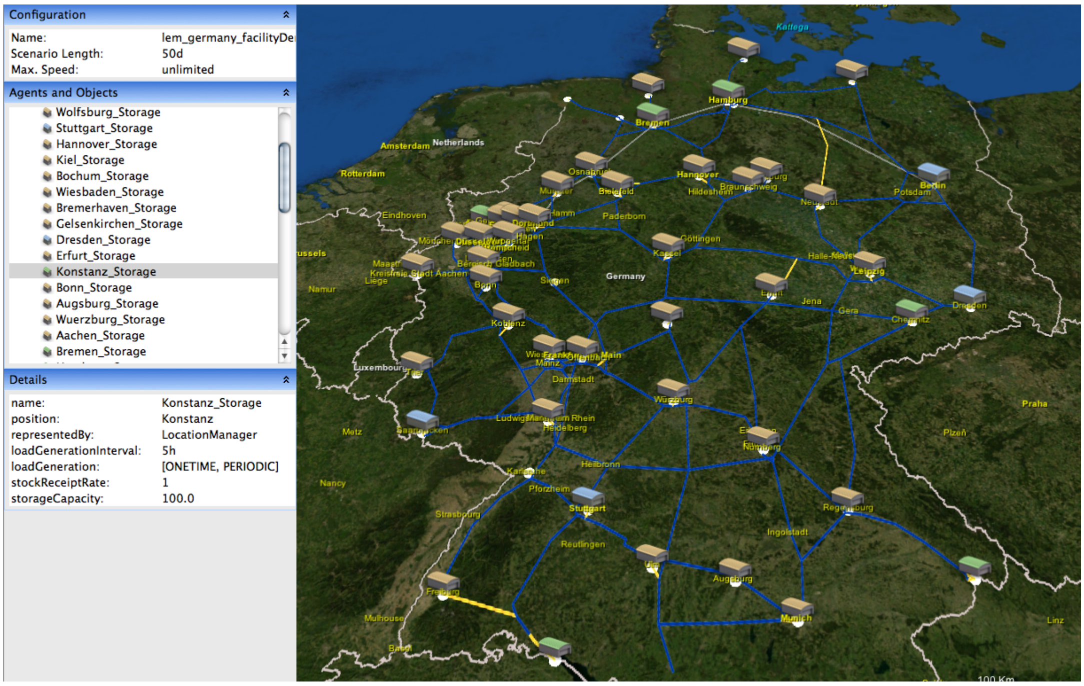

Figure 4.1: Germany-wide traffic network with allocated storage facilities employed in experiments. Facilities with blue or green-colored roofs are *active* facilities with an external intake of transport orders.

`ONETIME` The storage facility features an external one-time intake of new transport orders at the beginning of a simulation scenario.

`PERIODIC` The storage facility features a continuous external intake of new transport orders every n hours.

It is also possible to configure storage facilities such that they have both an `ONETIME` and a `PERIODIC` intake of new transport orders.

`stockReceiptRate` This parameter indicates how many new containers and transport orders should emerge at the storage facility, each time the order generation is invoked.

`loadGenerationInteval` This parameter defines the time interval in hours between back-to-back periodic order generations.

For the concrete experiments which have been conducted for the report, a sub-set of 10 storage facilities has been configured as facilities with an external intake of transport orders. These have been chosen based on their geographic location in the transport network such that a good distribution of order intake could be established in the scenarios. In Figure 4.1, these active storage facilities have been highlighted in the scenario visualization via blue and green roofs instead of the beige roof used for passive storage facilities.

The two different colors for the *active* storage facilities are a visual marker that allows the distinction between such storage facilities with *low* order intake (*blue* roof in Figure 4.1, e.g. `Berlin_Storage`) and *high* order intake (*green* roof in Figure 4.1, e.g. `Bremen_Storage`). For the concrete experiments presented in this report, two different base configurations have been used as shown in Table 4.1 for the respective order intake parameterization which has been encoded in adapted scenario ontology files.

4.2 Configuration of the Forwarder Transport Fleet

Besides the configuration of the storage facilities and their characteristics in the scenarios to be used over the course of the experiments reported here, it is also necessary to configure a concrete transport fleet for the considered freight forwarding agency. All experiments feature a homogeneous transport fleet of 16 trucks which can transport a single container at a time as shown in Listing 1 on page 12. Due to the ontological modelling the trucks are all located in Kassel and start their operations from there.

Parameter	Active, Low Order Intake	Active, High Order Intake	Passive, No Order Intake
Base Scenario 1: `scenario_germany_prod.5h.12h.owl`			
# Facilities	4	6	36
`lem_shared:loadGeneration`	ONETIME,PERIODIC	ONETIME,PERIODIC	NONE
`lem_shared:stockReceiptRate`	1 container/order	1 container/order	–
`lem_shared:loadGenerationInteval`	5 hours	12 hours	–
Base Scenario 2: `scenario_germany_prod.6h.12h.owl`			
# Facilities	4	6	36
`lem_shared:loadGeneration`	ONETIME,PERIODIC	ONETIME,PERIODIC	NONE
`lem_shared:stockReceiptRate`	1 container/order	1 container/order	–
`lem_shared:loadGenerationInteval`	6 hours	12 hours	–

Table 4.1: Employed parameterizations for the external transport order intake at storage facilities.

For the experiments, three types of configuration with respect to the transport management agents that are employed to act on behalf of the trucks in the transport fleet have been considered. First, there is a baseline configuration where all 16 trucks are managed by greedy transport agents. Second, there is a configuration where all 16 trucks are instead managed by transport agents which employ LEM3 for individual route planning. Finally, a third, if you like 'competitive' scenario, has been created where 8 trucks respectively are managed by agents of either type. The latter scenario was chosen specifically to analyze the emergent effects of mixing order selection strategies of different complexity.

With regard to the parameterization of the LEM3 sub-system, the decision was rendered to run the initial experiment with generic default parameters as shown in Listing 9 so as not to have the greedy baseline approach compete with a LEM3-based route-planning which has not been specifically tuned for the task at hand. It is thus possible to focus on 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 [Woj04, Section 2.2, pp. 6].

Listing 9 LEM3 Parametrization which has been used throughout the entire the experiment series.

```
# population_sizes=(20 50 100)
population_sizes=(50);
# no_children=(20 50 100)
no_children=(20);
# no_generations=(10 20 50 100)
no_generations=(30);
# selection_methods=("POPULATION_BASED" "FITNESS_BASED")
selection_methods=("POPULATION_BASED");
# sel_not_included=("SELNI_P" "SELNI_H" "SELNI_D")
sel_not_included=("SELNI_P");
# random_events=(0 5 10)
random_events=(0);
# plan_lengths=(3 5 7)
plan_lengths=(3);
```

4.3 Experiment Configuration and Execution

For the final experiment setup, given the degrees of freedom introduced in the two preceding sections, namely two base scenarios which differ in terms of the amount of orders that need to be handled by the freight forwarding agency, three different configurations of transport management agents, and an invariant set of parameters for the LEM3 sub-system, therefore comprised six individual experiments. Each experiment features ten back-to-back simulation runs where each run simulates the freight forwarder's operation over a time period of 60 days. In the following, the results of these experiment runs is presented which have been conducted via a batch-operation of the PlaSMA simulation system in roughly

Unit: *Euro/60d*	Mixed Transport Fleet	LEM Transport Fleet	Baseline Transport Fleet
LEM3 Route Planning			
Σ Costs ($\mu_s \pm \sigma_s$)	$604.538,72 \pm 739,14$	$1.201.645,21 \pm 1.458,77$	–
Σ Gains ($\mu_s \pm \sigma_s$)	$925.267,01 \pm 11.488,05$	$1.812.161,93 \pm 11.188,17$	–
Σ Balance ($\mu_s \pm \sigma_s$)	$320.728,29 \pm 10.748,91$	$610.516,72 \pm 9.729,40$	–
$p.A.$ Bal. ($\mu_s \pm \sigma_s$)	$40.091,03 \pm 4.158,33$	$38.157,29 \pm 4.398,14$	–
Baseline Order Selection			
Σ Costs ($\mu_s \pm \sigma_s$)	$612.301,91 \pm 353,71$	–	$1.212.916,38 \pm 670,98$
Σ Gains ($\mu_s \pm \sigma_s$)	$878.460,30 \pm 7.207,34$	–	$1.788.508,59 \pm 10.634,67$
Σ Balance ($\mu_s \pm \sigma_s$)	$266.158,39 \pm 6.853,63$	–	$575.592,21 \pm 9.963,69$
$p.A.$ Bal. ($\mu_s \pm \sigma_s$)	$33.269,80 \pm 5.160,81$	–	$35.974,51 \pm 4.794,69$
Fleet Bal. ($\mu_s \pm \sigma_s$)	$586.886,68 \pm 17.602,54$	$610.516,72 \pm 9.729,40$	$575.592,21 \pm 9.963,69$

Table 4.2: Financial results of the employment of different transport management approaches, measured in Euros per 60 days.

ten hours of real time on a standard PC with a 2.8 GHz Intel Core i7 CPU (with four cores) running the Linux-version of PlaSMA under Ubuntu Linux 10.04 (64bit).

4.4 Experiment Results

This section presents experimental results divided by the two considered scenarios that have been introduced above. The evaluation of the particular experiments is based on a sub-set of a comprehensive system of performance indicators[7] which have been logged by the transport management agents (20 indicators), the order management agents (7 indicators) and the order information service agent (3 indicators).

The presentation of the experiment results comprises different tiers. In the first tier, Table 4.2 (Section 4.4.1) and Table 4.6 on page 31 (Section 4.4.1) present a financial view upon the operations of the freight forwarding agency, both from a global company perspective and also the individual agent perspective. Thereby, these tables oppose the three considered transport management configurations introduced in Section 4.2. The data in the tables 4.3 (Section 4.4.1) and 4.7 (Section 4.4.2 on page 30) provide a supplemental analysis which allows for the detailed breakdown of the allocation of possible types of transport orders which is of particular interest in a competitive setting where both transport management approaches that have been introduced in previous chapters are employed concurrently.

In the second tier, the focus of the analysis is shifted towards the pickup times for those containers associated with different types of transport orders. The incentive here was to get a feeling for the processing times given different order inflow scenarios. The results of this part of the evaluation are documented in Table 4.4 (Section 4.4.1) and Table 4.8 (Section 4.4.2 on page 30). Besides the statistical parameters which present highly accumulated results additional graphics plot the inventory levels at the *active* storage facilities as well as a global overview of pending orders and their breakdown in transport orders for full containers and less valuable empty reverse transports, both over the full course of the simulation runs. The plots are thereby examples which belong to a single simulation run.

The final tier of the analysis finally is concerned with the distribution of truck operations, that is empty rides and deliveries, by the transport fleets. The data is presented in Table 4.5 (Section 4.4.1) and Table 4.9 (Section 4.4.2 on page 30).

4.4.1 Experiment Series I: Scenario with Low External Order Inflow

Fleet-Level Financial Analysis As highlighted in the preceding section, the multi-tier analysis of the simulation experiments conducted for this report begins with an examination of the financial performance

[7] internally referred to as *keymeasures* in PlaSMA, cf. [GWB+10, Chapter 5.5.9]

Unit: *Euro*	Mixed Transport Fleet	LEM Transport Fleet	Baseline Transport Fleet
LEM3 Route Planning			
Order Price, full ($\mu_s \pm \sigma_s$)	$677,58 \pm 279,03$	$757,39 \pm 291,97$	–
$CI_\alpha : \alpha = 0.05$	$[672,18\,;682,99]$	$[753,27\,;761,51]$	–
Order Price, empty ($\mu_s \pm \sigma_s$)	$183,5 \pm 70,81$	$188,91 \pm 73,05$	–
$CI_\alpha : \alpha = 0.05$	$[182,26\,;184,73]$	$[187,87\,;189,96]$	–
Baseline Order Selection			
Order Price, full ($\mu_s \pm \sigma_s$)	$847,98 \pm 279,10$	–	$757,98 \pm 291,47$
$CI_\alpha : \alpha = 0.05$	$[842,22\,;853,75]$	–	$[753,85\,;762,11]$
Order Price, empty ($\mu_s \pm \sigma_s$)	$201,58 \pm 76,25$	–	$189,62 \pm 72,95$
$CI_\alpha : \alpha = 0.05$	$[199,60\,;203,57]$	–	$[188,56\,;190,7]$

Table 4.3: Order-level financial results of the employment of different transport management approaches.

of the respective compositions of transport fleets. The data supporting this examination is shown for all tested fleet configurations in Table 4.2 on the preceding page.

When considering the fleet balance (Fleet Bal. ($\mu_s \pm \sigma_s$) in Table 4.2 on the facing page) first, the direct comparison of the performance of the heterogeneous transport fleets shows a significantly higher mean overall revenue when using the planning approach, i. e. 610.516,72 € vs. only 575.592,21 € for the baseline approach where for both the standard derivation falls short of 10.000 € ($< 2\,\%\,\mu_s$). When broken down to the level of single means of transports, this translates on average to an individual revenue of 38.157,29 € for the planning approach compared to 35.974,21 € for the baseline approach. The data for costs and gains also provided in Table 4.2 on the preceding page also shows that planning leads to a reduction of operations costs and at the same time an increase in earnings.

When considering the competitive scenario where both the baseline and the planning approach were employed to equal proportion, the mean overall revenue of 586.886,68 € falls between the homogeneous cases with a tendency towards the weaker fleet performance measured for the baseline fleet. Thus, while the insertion of planning transport management agents is an effective means to increase baseline performance, from a purely macro-financial point of view, the pure planning approach is still clearly preferable. This is especially true as the standard deviation from the mean increases significantly to 17.602,54 € compared to the roughly 10.000 € measured for both other fleet configurations. It is also interesting to note in the competitive scenario that on average, the transport management agents which employ planning outperform their less provident counterparts with a mean revenue of 40.091,03 € compared to 33.269,80 € while the standard deviation is also reduced from 5.160,81 € down to 4.158,33 €.

Order-Level Financial Analysis Having analyzed the economical performance of the respective transport fleets on a global level, the focus is now shifted towards an investigation on the level of earnings from handling individual regular or reverse transport orders. The corresponding data is presented in Table 4.3.

In a side-by-side comparison of the two cases of heterogeneous transport fleets, the data shows that the mean per revenues for orders related to the handling of regular and reverse transports are nearly identical as should be expected given one and the same applied order generation mechanism throughout the simulation experiments. The mean order price resides at $\sim$757 € with a high standard deviation of $\sim$291 € for regular transports. The numbers for reverse transports are $\sim$189 € in the mean with a standard deviation of $\sim$73 €.

In the competitive scenario, the data in Table 4.3 shows that the competition for transport orders across the two distinct transport sub fleets effectuates an unequal distribution of acquired orders such that the mean per-order revenue for both regular and reverse transports is significantly higher for the group of agents that employ the baseline order selection. To be more precise, while the agents that employ LEM-based planning pocket 677,58 € in the mean per regular and 183,50 € per reverse transport – numbers largely consistent with those measured in the planning-only case, the baseline agents manage to increase their per-order performance significantly compared to the baseline-only case, pocketing

Unit: *hours*	Mixed Transport Fleet	LEM Transport Fleet	Baseline Transport Fleet
Wait for Pickup, full ($\mu_s \pm \sigma_s$)	$7,317 \pm 8,011$	$6,556 \pm 7,359$	$14,715 \pm 12,960$
$CI_\alpha : \alpha = 0.05$	$[7,204 ; 7,431]$	$[6,452 ; 6,661]$	$[14,532 ; 14,900]$
Wait for Pickup, empty ($\mu_s \pm \sigma_s$)	$68,565 \pm 34,092$	$40,259 \pm 25,935$	$98,730 \pm 32,640$
$CI_\alpha : \alpha = 0.05$	$[68,070 ; 69,062]$	$[39,886 ; 40,632]$	$[98,254 ; 99,217]$

Table 4.4: Waiting times until a container is picked up.

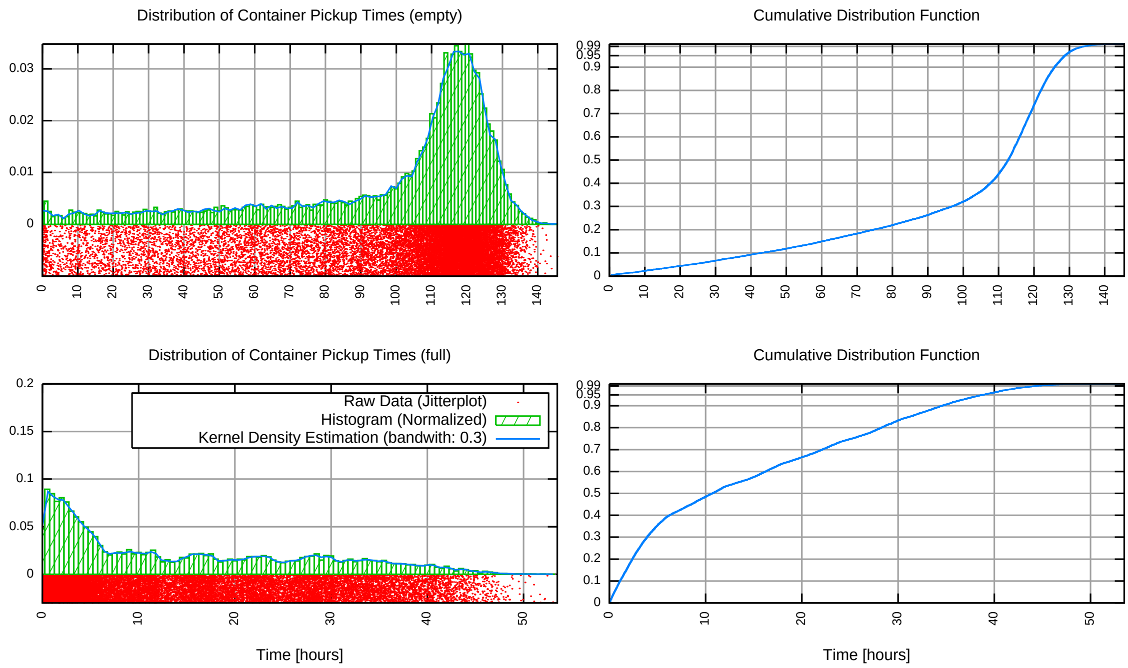

Figure 4.2: Distribution of pickup times for containers for the 10 simulation runs with a homogeneous greedy transport fleet.

847,98 € in the mean per regular and 201,58 € per reverse order. Taking into consideration the results presented in Table 4.2 which have been discussed in the preceding paragraph, it is highlighted, that planning amounts to higher overall revenues, yet lower revenues on the level of individual orders. This is in fact an effect of the planning evaluating complete transport plans with their multiple tiers in which it may be acceptable to tolerate lower revenues for certain plan steps when this in return leads to more attractive follow-up orders.

Pickup Times The data in Table 4.4 and the supplementary plots in figures 4.2, 4.4, 4.3 show that the LEM-based planning approach leads to a significant improvement of waiting times both for the pickup of containers for regular and reverse transports.

For the pickup of containers for regular transports, the mean waiting time and the standard deviation is roughly cut in half, i. e. $6,556\,h \pm 7,359\,h$ for the planning approach compared to $14,715\,h \pm 12,96\,h$ for the baseline approach. The plots of the distribution of pickup times show in addition, that the planning has the effect of sliding the distribution of pickup times towards desirable shorter response times. In particular, the tail of the distribution is rendered much less pronounced. The effect is also clearly visible in the curve progression of the respective cumulative distribution functions. While in general, the measured effect of the proposed planning approach is thus substantiated. Figure 4.3 on the next page also shows that single outlier cases exist for which the pickup times are worse than for those measured in the baseline (Figure 4.2) or mixed case (Figure 4.4 on the next page). This effect, namely a significant improvement for the vast majority of cases and, at the same time, the emergence of detached particularly bad cases, is replicated as well with the pickup times for reverse transports analyzed next.

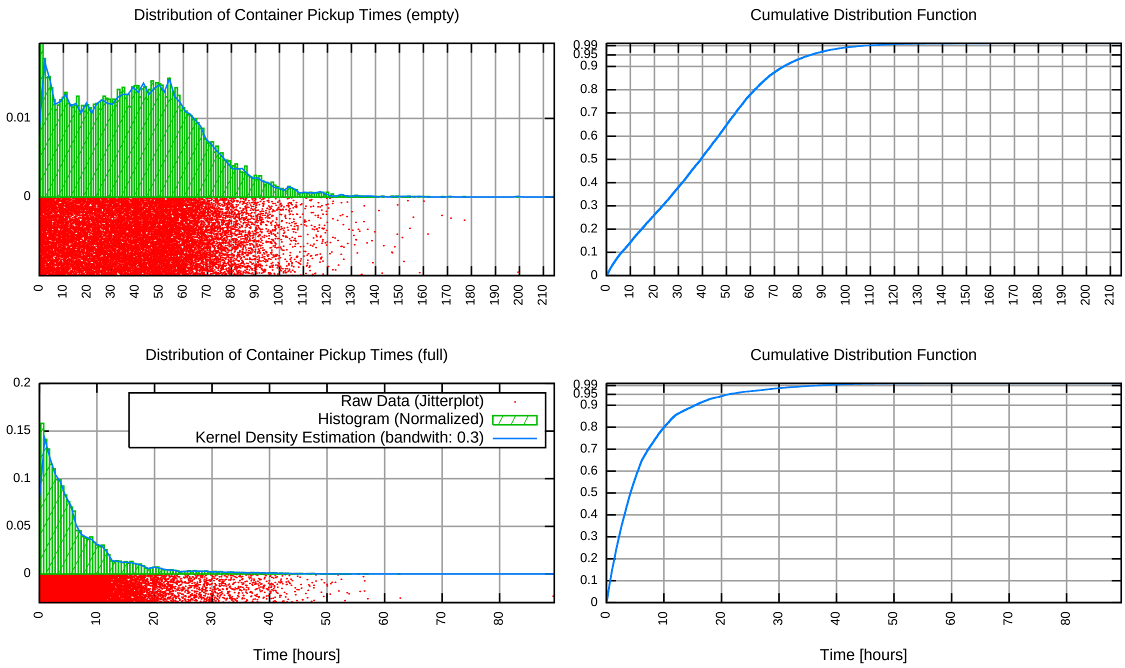

Figure 4.3: Distribution of pickup times for containers for the 10 simulation runs with a homogeneous planning transport fleet.

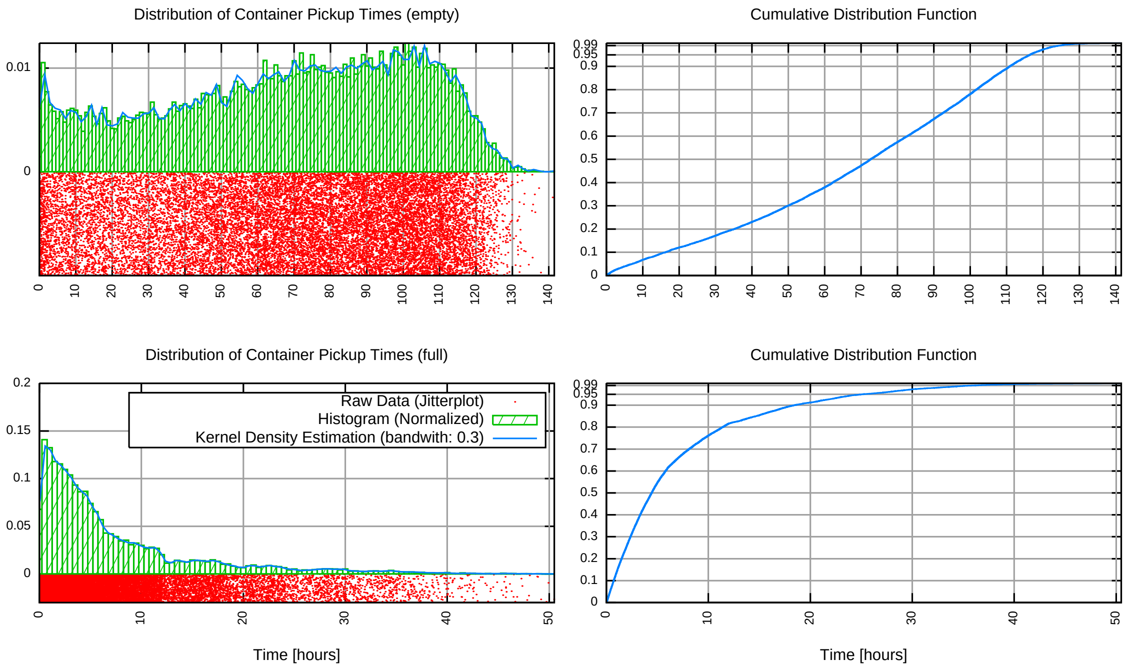

Figure 4.4: Distribution of pickup times for containers for the 10 simulation runs with a heterogeneous transport fleet.

First, consulting Table 4.4 on the facing page, it is shown that the reduction in waiting time is pronounced with respect to the mean $(40,259\,h$ vs. $98,73\,h)$, yet less pronounced for the respective standard deviation $(25,935\,h$ vs. $32,64\,h)$ – a result backed by the data from the scenario with higher order inflow, as shown in Table 4.4 on the preceding page. The introduction of planning agents into the system both erodes and moves the pronounced peak in the distribution of pickup times of $\sim 120\,h$ for the homogeneous baseline fleet down to $\sim 90\,h$ for the heterogeneous transport fleet and finally

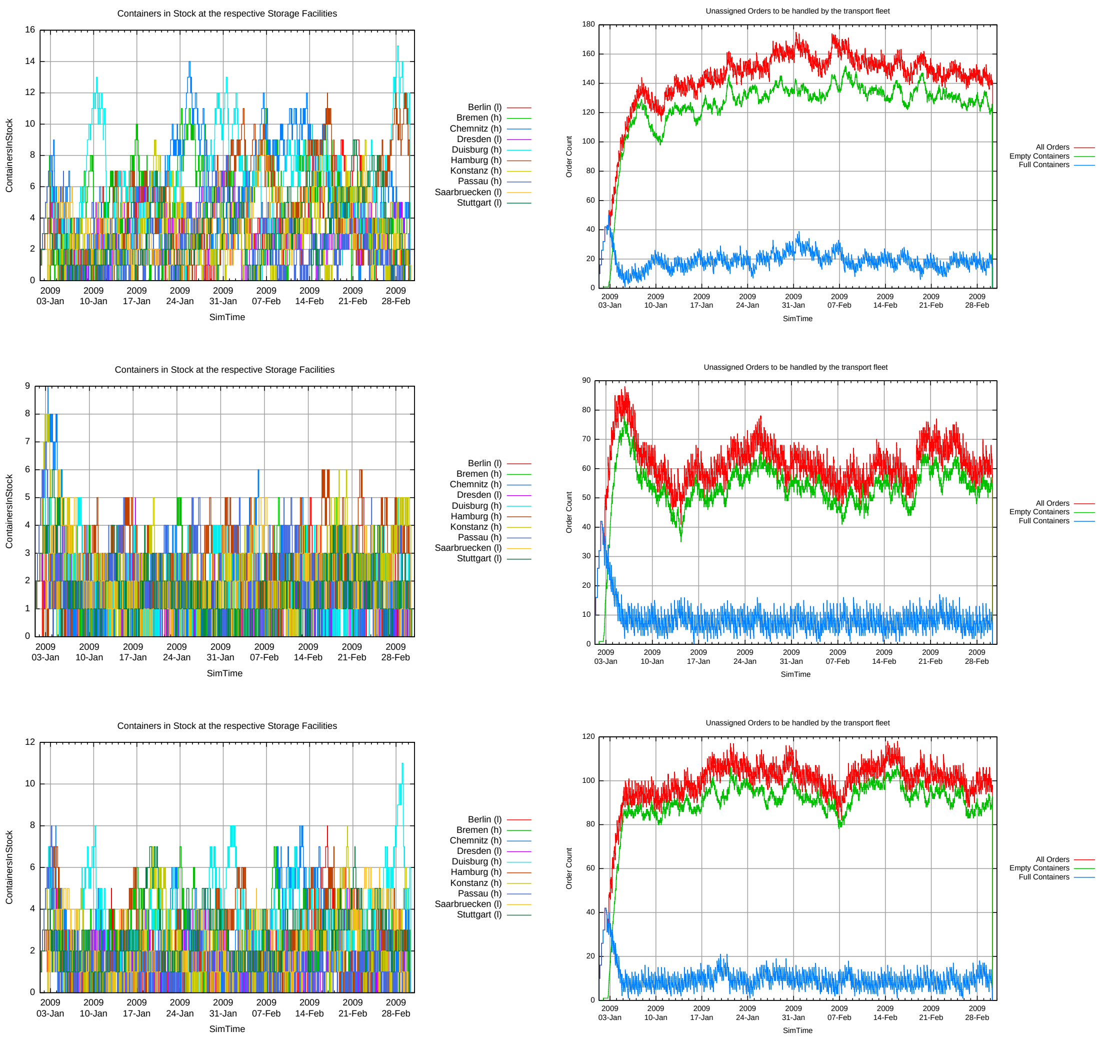

Figure 4.5: *Top row*: Inventory Levels of pending orders at *active* storage facilities (where *h* means high order inflow and *l* low order inflow) on the left, and global development of pending orders in the system over the course of a simulation run, here for the greedy transport management approach. *Middle row*: The same plots for the mixed transport management, and finally in the *bottom row*: the results using LEM3-based transport planning exclusively. For all scenarios, the plotted results are taken from the first out of ten simulation runs.

$\sim 60\,h$ for the homogeneous planning fleet. In the process, the pickup times become much more evenly distributed and a peak for very low pickup times is taking shape. Therefore, the planning shows a significant positive effect on the handling times of the transport fleets which is again highlighted in the curve progression of the cumulative distribution functions. Yet, pure planning produces even more pronounced cases, seldom as they may be, with vastly substandard pickup times.

The data from the simulation experiments therefore seems to suggest, that actually a well-adjusted combination of both employed approaches to transport management yields a satisfactory system behaviour with regard to the pickup times. Otherwise, the planning approach would need to be enhanced such that outlier cases can be effectively handled.

Stock Levels and Active Orders In contrast to the evaluation of the conducted experiment with respect to the total revenue, the pickup times or the handling operations which are based on a table-based presentation of condensed data, 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 4.5, sampled from single simulation runs.

The plots clearly show that with the planning approach the inventory levels of the storage facilities can be kept significantly lower than as compared to the baseline approach. In that planning case, in the beginning of the scenario, the stock levels reach the highest values which coincides with the initial high amount of regular transports in the system generated at the begin of the simulation scenarios. Hereafter, the system settles down quickly and does not produce further notable spikes over the remaining course of the scenario. For the baseline case, the stock levels settle down at significantly higher values and the variability in stock is far more pronounced. The plot for the mixed scenario is basically a middle ground between the aforementioned cases which underlines the stabilizing effect of the planning transport agents on the emerging overall system behaviour.

The right set of plots in Figure 4.5 on the preceding page depicts the number of unassigned orders which need to be handled by the respective transport fleets over the course of a simulation. For all three configurations of the transport fleet, it holds that the high amount of unassigned orders for regular transports of full containers is soon, i. e. within the first few days of the simulation, dissipated while at the same time the number of unassigned reverse transport grows to a multitude of the remaining regular transports. This is due to the prioritization of the former type of transport via the invariant experiment parameterization. It was assumed that delivery of goods has premium priority for the freight forwarding agency while the recirculation of empty containers is less time-critical.

The plots show, that the planning approach to operative transport management leads to a significantly lower stack of unassigned orders oscillating around 60 compared to roughly 145 for the baseline case. Also, only in the former case, after the steep initial rise in the number of unassigned orders for empty container transports, the number can be reduced once more before settling down on a lower level. The baseline approach leads to significant oscillations for both order types while the planning approach effectively mitigates to some extent the oscillations in the case of transport orders for full containers, an effect which is also visible in the mixed scenario.

In all the considered cases, the plots suggest that the order situation which was taken as a basis for the first conducted set of experiments could be handled effectively in that the amount of unassigned orders settles down on a certain, although different level rather than growing unboundedly. However, the planning approach must be considered more efficient in handling orders and thus seems to be able to handle larger amounts of orders; a hypothesis, which could be confirmed in the second set of conducted experiments which featured a higher order inflow (cf. Figure 4.9 on page 35).

Transport Operations The data presented in Table 4.5 on the next page shows that the LEM-based planning approach let to a significant increase in the overall amount of successfully operated container transports $(3.761, 80 \pm 5, 02)$ compared to the baseline approach $(3.661, 20 \pm 5, 49)$. Even though both regular and reverse transports were handled by the planning fleet, the data also suggests a tendency of this approach towards a reduced fraction of regular transports ($50, 8\,\%$ compared to $51, 8\,\%$) which becomes more pronounced in more buzzing order situations (cf. Table 4.9 on page 36 in Section 4.4.2).

In spite of the dominance in the raw amount of handled transports highlighted thus far and a slightly increased total length of delivery tours, the fraction of container transports on all truck operations is in the mean notably smaller for the planning transport fleet $(47, 2\,\%)$ than for the baseline fleet $(54, 9\,\%)$. These pieces of data seem to suggest that due to the foresight (i. e. always planning three steps ahead), the agents of the planning fleet are more inclined to interleave regular transport operations with empty relocations if such a mode of action leads to an increase in individual revenue as shown in Table 4.2 on page 24. Therefore, the results affirm the expectations given the considered order selection strategies.

When shifting the focus from the analysis of the homogeneous settings where either only baseline or planning agents were operating in the scenario towards the mixed setting with an equal amount of both agent types, the first thing to notice in the data presented in the left column of Table 4.5 is the significant difference in overall operated container transports across both fleets. Both with regard to regular and reverse container transports, the planning fleet clearly outperformed the baseline, resulting in $2.267 \pm 15, 93$ deliveries for the former vs. only $1.452, 3 \pm 15, 12$ deliveries for the latter.

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

	Mixed Transport Fleet	LEM Transport Fleet	Baseline Transport Fleet
LEM3 Transport Operations			
Σ Deliv ($\mu_s \pm \sigma_s$)	$2.267,00 \pm 15,93$	$3.761,80 \pm 5,02$	–
$CI_\alpha : \alpha = 0.05$	$[2.255,59\,;2.278,40]$	$[3.758,20\,;3.765,39]$	–
Σ Deliv $_{\text{full}}$ ($\mu_s \pm \sigma_s$)	$1.016,80 \pm 14,67$	$1.911,00 \pm 1,71$	–
$CI_\alpha : \alpha = 0.05$	$[1.006,31\,;1,027.29]$	$[1.909,78\,;1.912,22]$	–
Σ Deliv $_{\text{empty}}$ ($\mu_s \pm \sigma_s$)	$1.250,20 \pm 12,59$	$1.850,80 \pm 3,88$	–
$CI_\alpha : \alpha = 0.05$	$[1.241,19\,;1.259,21]$	$[1.848,02\,;1.853,58]$	–
Σ Fraction Full (μ_s)	$0,448$	$0,508$	–
Σ Fraction: All (μ_s)	$0,657$	$0,472$	–
Σ Length Del. ($\mu_s \pm \sigma_s$) [km]	$960.992,42 \pm 7.490,42$	$1.725.801,60 \pm 13.041,15$	–
$CI_\alpha : \alpha = 0.05$	$[955.634,1\,;966.350,7]$	$[1.716.472,5\,;1.735.130,7]$	–
Baseline Transport Operations			
Σ Deliv ($\mu_s \pm \sigma_s$)	$1.452,30 \pm 15,12$	–	$3.661,20 \pm 5,49$
$CI_\alpha : \alpha = 0.05$	$[1.441,49\,;1.463,11]$	–	$[3.657,27\,;3.665,13]$
Σ Deliv $_{\text{full}}$ ($\mu_s \pm \sigma_s$)	$893,10 \pm 13,98$	–	$1.898,60 \pm 3,22$
$CI_\alpha : \alpha = 0.05$	$[883,10\,;903,10]$	–	$[1.896,29\,;1.900,91]$
Σ Deliv $_{\text{empty}}$ ($\mu_s \pm \sigma_s$)	$559,20 \pm 12,36$	–	$1.762,60 \pm 7,28$
$CI_\alpha : \alpha = 0.05$	$[550,36\,;568,04]$	–	$[1.757,39\,;1.767,81]$
Σ Fraction: Full (μ_s)	$0,615$	–	$0,518$
Σ Fraction: All (μ_s)	$0,495$	–	$0,549$
Σ Length Del. ($\mu_s \pm \sigma_s$) [km]	$746.629,48 \pm 7.348,29$	–	$1.684.940,17 \pm 11.136,89$
$\langle CI_\alpha : \alpha = 0.05$	$[741.372,8\,;751.886,1]$	–	$[1.676.973,3\,;1.692.907,0]$

Table 4.5: Overview of transport operations for trucks managed by different transport agents.

from the mixed scenario, the data shows that the mean fraction of deliveries on all truck operations is higher for the planning sub fleet ($65,7\,\%$) than for the baseline analogon ($49,5\,\%$). This suggests that the former fleet actually profits from the competitive setting at the expense of the baseline fleet. A related result is the planning fleet steals a significant amount of individually less profitable reverse transports from the baseline fleet. However, as previously documented in Table 4.2 on page 24, given the specification of the considered simulation model, 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 capacity utilization and profitability.

4.4.2 Experiment Series II: Scenario with High External Order Inflow

The preceding section discussed the results for the deployment of all three considered approaches to operational transport planning for a first particular order inflow scenario as introduced in Table 4.1 on page 23, the focus is now turned toward the alternative order inflow scenario specified in that same table. The analysis will highlight in particular divergencies from the observations highlighted hitherto which should provide an understanding of the overall system behaviour under changing economic conditions.

Fleet-Level Financial Analysis Taking up the analysis approach from Section 4.4.1, the multi-tier analysis of the simulation experiments conducted for this report begins again with an examination of the financial performance of the respective compositions of transport fleets as a whole. The data supporting this examination is shown for all tested fleet configurations in Table 4.6 on the facing page.

When considering the fleet balance (Fleet Bal. ($\mu_s \pm \sigma_s$) in Table 4.6 on the next page) first, the direct comparison of the performance of the heterogeneous transport fleets shows a significantly higher mean overall revenue when using the planning approach, i. e. 866.038,94 € vs. only 708,409,97 € for the baseline approach. In comparison to the results for the first set of experiments outlined in Section 4.4.1 on page 24 however, this time the standard deviation – even though still residing at $\sim 3\%$ is notably higher for the baseline approach (i. e. 10.051,49 € for the planning fleet and more than double that

Unit: *Euro/60d*	Mixed Transport Fleet	LEM Transport Fleet	Baseline Transport Fleet
LEM3 Route Planning			
Σ Costs ($\mu_s \pm \sigma_s$)	$606.931,42 \pm 415,90$	$1.213.175,10 \pm 930,35$	–
Σ Gains ($\mu_s \pm \sigma_s$)	$1.096.546,40 \pm 15.878,41$	$2.079.214,04 \pm 10.981,84$	–
Σ Balance ($\mu_s \pm \sigma_s$)	$489.614,98 \pm 15.462,51$	$866.038,94 \pm 10.051,49$	–
$p.A.$ Balance ($\mu_s \pm \sigma_s$)	$61.201,87 \pm 3.856,59$	$54.127,43 \pm 4.415,25$	–
Baseline Order Selection			
Σ Costs ($\mu_s \pm \sigma_s$)	$609.229,46 \pm 920,48$	–	$1.212.536,08 \pm 885,16$
Σ Gains ($\mu_s \pm \sigma_s$)	$966.137,90 \pm 10.248,10$	–	$1.920.946,05 \pm 22.450,69$
Σ Balance ($\mu_s \pm \sigma_s$)	$356.908,44 \pm 9.327,62$	–	$708.409,97 \pm 21.565,53$
$p.A.$ Balance ($\mu_s \pm \sigma_s$)	$44.613,56 \pm 5.062,45$	–	$44.275,62 \pm 4.347,73$
Fleet Bal. ($\mu_s \pm \sigma_s$)	$846.523,42 \pm 24.790,13$	$866.038,94 \pm 10.051,49$	$708.409,97 \pm 21.565,53$

Table 4.6: Financial results of the employment of different transport management approaches, measured in Euros per 60 days.

value, namely 21.565,53 € for the baseline fleet). The data for costs and gains which is also provided in Table 4.6 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. It should be noted further, that in the considered scenario 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 to equal proportion, the mean overall revenue of 846.523,42 € comes very close to the 866.038,94 € for the homogeneous planning fleet. Thus, compared to the results for the first sets of experiments, discussed in Section 4.4.1 on page 24, the insertion of planning transport management agents into an otherwise non-planning baseline 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 management agents alone which has been observed before is evident once more in the data presented in Table 4.6.

Also in line with the previously discussed set of experiments, it is again 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 61.201,87 € compared to 44.613,56 € and a lower standard deviation of 3.856,59 € compared to 5.062,45 €.

More interesting than those 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 4.347,73 € up to 5.062,45 €. 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 in the fact that the per agent revenues rise significantly from a mean of 54.127,43 € up to 61.201,87 € while at the same time the standard deviation is reduced from 4.415,25 € down to 3.856,59 €.

Order-Level Financial Analysis Having analyzed the economical performance of the respective transport fleets on a global level, the focus is now shifted towards an investigation on the level of earnings from handling individual regular and reverse transport orders. The corresponding data is presented in Table 4.7 on the following page.

The side-by-side comparison of the two cases of heterogeneous transport fleets can be neglected as the data measured for this second set of experiments is concurrent with the corresponding data discussed in Section 4.4.1 on page 31.

In the competitive scenario, the data in Table 4.7 on the next page shows that the competition for transport orders across the two distinct transport sub fleets effectuates an unequal distribution of acquired orders such that the mean per-order revenue for both regular and reverse transports is higher for

Unit: *Euro*	Mixed Transport Fleet	LEM Transport Fleet	Baseline Transport Fleet
LEM3 Route Planning			
Order Price, full ($\mu_s \pm \sigma_s$)	$695,20 \pm 272,49$	$758,93 \pm 291,44$	–
$CI_\alpha : \alpha = 0.05$	$[690,40 ; 700,00]$	$[755,09 ; 762,77]$	–
Order Price, empty ($\mu_s \pm \sigma_s$)	$187,19 \pm 71,73$	$189,18 \pm 72,95$	–
$CI_\alpha : \alpha = 0.05$	$[185,93 ; 188,44]$	$[188,20 ; 190,16]$	–
Baseline Order Selection			
Order Price, full ($\mu_s \pm \sigma_s$)	$843,85 \pm 295,08$	–	$758,87 \pm 291,99$
$CI_\alpha : \alpha = 0.05$	$[837,96 ; 849,75]$	–	$[754,91 ; 762,83]$
Order Price, empty ($\mu_s \pm \sigma_s$)	$193,57 \pm 75,28$	–	$189,50 \pm 73,24$
$CI_\alpha : \alpha = 0.05$	$[191,92 ; 195,23]$	–	$[188,42 ; 190,58]$

Table 4.7: Order-level financial results of the employment of different transport management approaches.

Unit: *hours*	Mixed Transport Fleet	LEM Transport Fleet	Baseline Transport Fleet
Wait for Pickup, full ($\mu_s \pm \sigma_s$)	$12,597 \pm 11,899$	$7,926 \pm 8,706$	$40,820 \pm 24,855$
$CI_\alpha : \alpha = 0.05$	$[12,439 ; 12,755]$	$[7,812 ; 8,042]$	$[40,482 ; 41,158]$
Wait for Pickup, empty ($\mu_s \pm \sigma_s$)	$90,029 \pm 33,761$	$53,832 \pm 30,718$	$141,564 \pm 46,675$
$CI_\alpha : \alpha = 0.05$	$[89,566 ; 90,493]$	$[53,418 ; 54,246]$	$[140,874 ; 142,254]$

Table 4.8: Waiting times until a container is picked up.

the group of agents that employ the baseline order selection. To be more precise, while the agents that employ LEM-based planning pocket 695,20 € in the mean per regular and 187,19 € per reverse transport, the baseline agents manage to increase their per-order performance compared to the baseline-only case, pocketing 843,85 € in the mean per regular and 193,57 € per reverse order. Taking into consideration the results presented in Table 4.6 which have been discussed earlier paragraph, it is highlighted that planning amounts to higher overall revenues, yet lower revenues on the level of individual orders. Thus, the effect already found in the analysis of the low order inflow experiments is affirmed and, given the increased mean per agent revenues for planning transport management agents, rendered more pronounced. However, in a market situation with a higher overall inflow of transport orders, the effect of baseline agents succeeding to earn higher mean revenues from reverse transports is mitigated, while the tendency of the planning agents to prefer lower-priced regular transports is substantiated and should therefore be investigated in greater detail in future analyses of the overall system behaviour.

Pickup Times The data in Table 4.8 and the supplementary plots in figures 4.6, 4.8, 4.7 show that the LEM-based planning approach leads to a significant improvement of waiting times both for the pickup of containers for regular transports and reverse transports.

For the pickup of containers for regular transports, the mean waiting time and the standard deviation are both reduced significantly, with $7,926\,h \pm 8,706\,h$ for the planning approach compared to $40,820\,h \pm 24,855\,h$ for the baseline approach. Thus, while the higher inflow of transport orders in this second set of experiments amounts to an increase in the mean pickup time of under two hours for the case of a planning transport fleet, compared to the situation with lower order inflow (cf. Section 4.4.1, page 26), the measured mean time is increased by a factor of $\sim 2.8\%$ for the baseline transport fleet. The plots of the distribution of pickup times also depict vastly different characteristics. In Figure 4.6 on the next page, it is shown that for the baseline approach, the distribution of pickup times is quite regular with only two shallow peaks, the first 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. In Figure 4.7 on the facing page, by contrast, there is only a single peak at ~ 1 hour which is the followed by a very steep incline which 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 amount 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

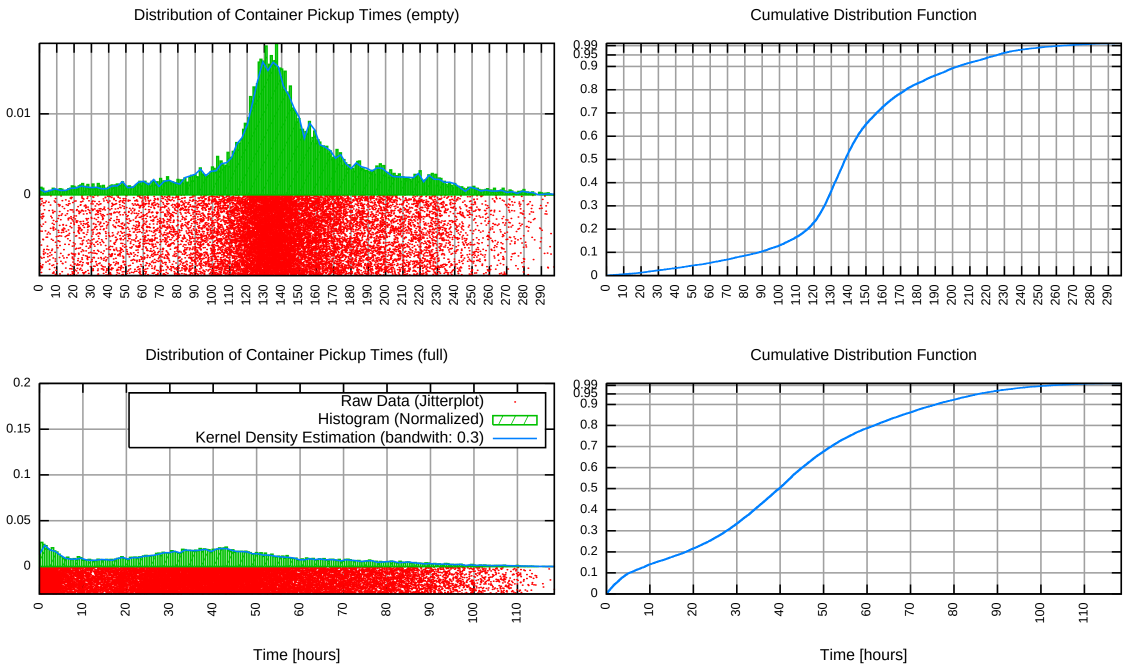

Figure 4.6: Distribution of pickup times for containers for the 10 simulation runs with a homogeneous greedy transport fleet.

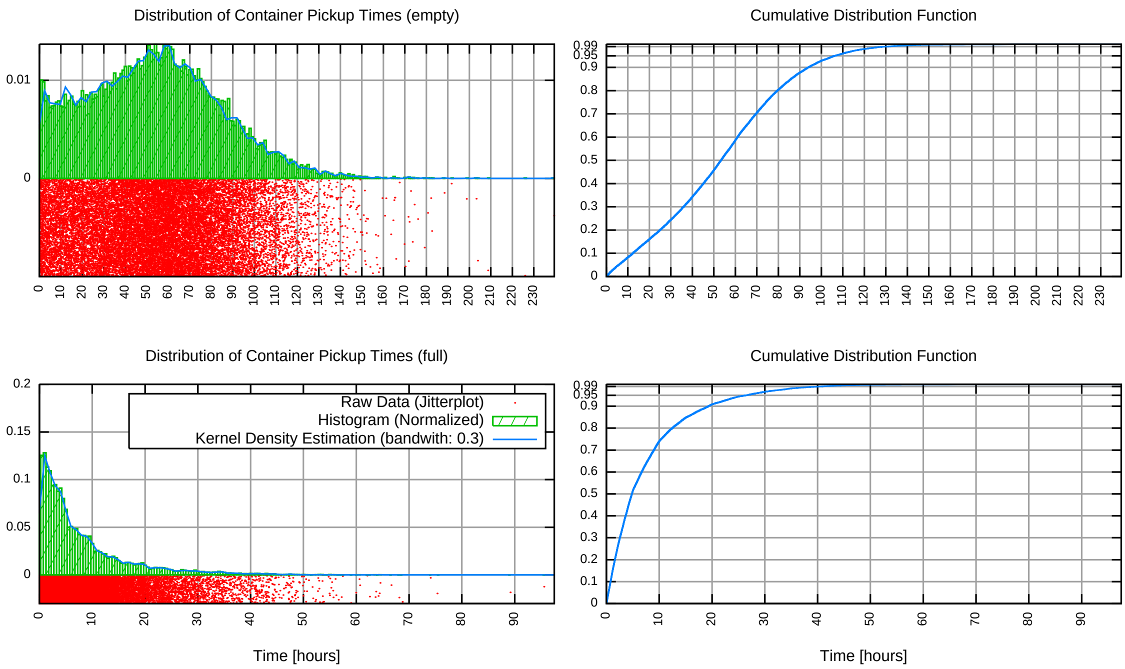

Figure 4.7: Distribution of pickup times for containers for the 10 simulation runs with a homogeneous planning transport fleet.

and it is also replicated for the case of reverse transports. It is thus a general side-characteristic with the planning approach used in the experiments which calls for in-depth causal research in order to enhance the approach for future experiments. When considering the case of a mixed transport fleet, the plot in Figure 4.8 on the next page shows that the insertion of planning agents has the effect of ablating the second peak in the baseline-only distribution and accumulating more weight at the initial peak, thus bringing the distribution closer to the planning-only case. It is also shown, that the effective value range

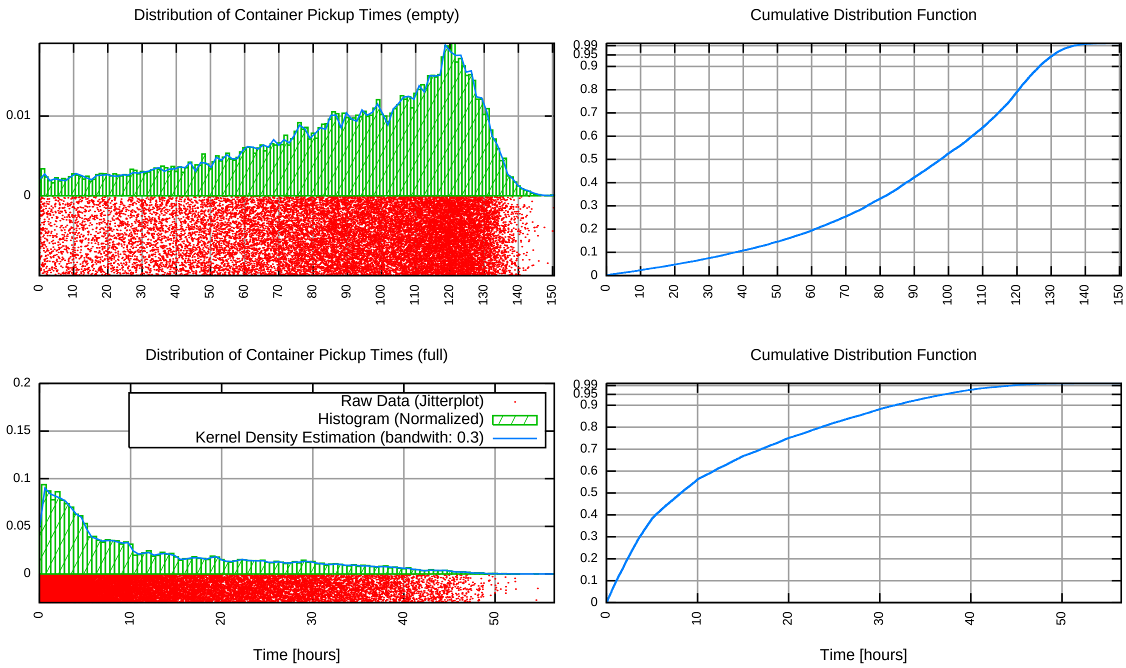

Figure 4.8: Distribution of pickup times for containers for the 10 simulation runs with a heterogeneous transport fleet.

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 4.8 on page 32 shows that in the baseline case, the mean pickup time for reverse transport is now $141,564\,h$ – an increase of $43\,\%$ from the first set of experiments, with the standard deviation growing significantly as well. The plot in Figure 4.6 on the previous page also shows, that in the face of the higher order inflow used for the second set of experiments, the characteristics of the distribution of pickup times for reverse transports deviates from its analog in the first set of experiments (cf. Figure 4.2 on page 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 4.6 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 order 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, shown paradigmatically in the plots in Figure 4.9 on the next page.

For the case of the planning transport fleet, the mean time for the pickup of reverse transports is $53,83\,h$ – an increase of $33\,\%$ from the first set of experiments. The plot in Figure 4.7 on the preceding page thereby shows that the basic characteristics of the distribution of pickup times is retained.

The effect of combining both approaches – baseline and planning – to equal fractions in the mixed transport fleet can, according to the plot in Figure 4.8 be succinctly described as follows: 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 already 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 of the conducted experiment with respect to the total revenue, the pickup times or the handling operations which are based on a table-based presentation of condensed data, for the analysis of both the inventory levels at active storage facilities

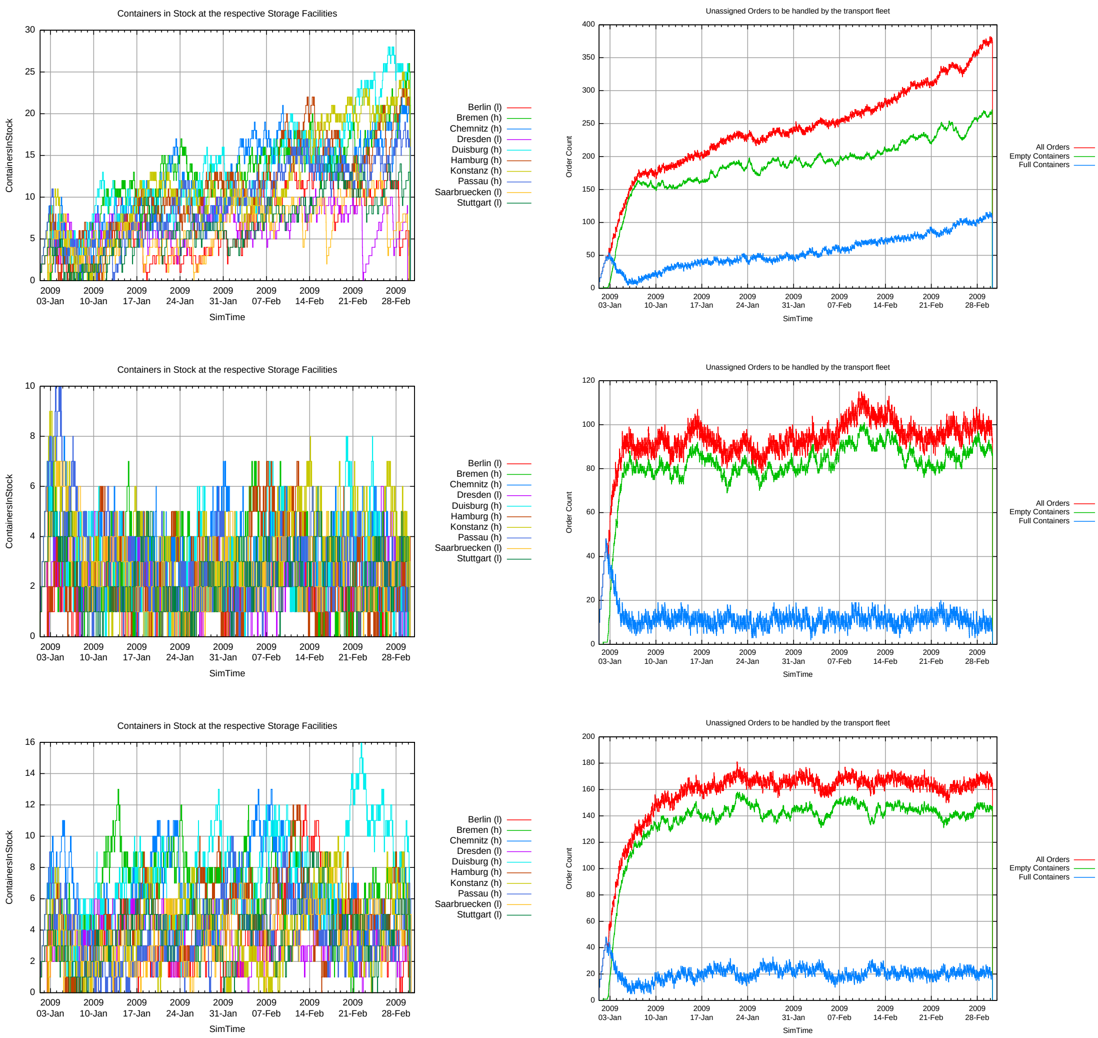

Figure 4.9: *Top row*: Inventory Levels of pending orders at *active* storage facilities (where *h* means high order inflow and *l* low order inflow) on the left, and global development of pending orders in the system over the course of a simulation run, here for the greedy transport management approach. *Middle row*: The same plots for the mixed transport management, and finally in the *bottom row*: the results using LEM3-based transport planning exclusively. For all scenarios, the plotted results are taken from the first out of ten simulation runs.

and the total amount of active orders within the system is based on paradigmatic plots in Figure 4.9, taken from single simulation runs.

For the inventory levels of the storage facilities, the plots clearly show that for the case of planning, these levels can be kept low. While in the scenario with high order inflow considered here, stock levels max out at 6–8 containers at a time (cf. Figure 4.5 on page 28), in the low order case the max stock levels after the initial level-off are at 5–6 containers. In addition, the plots for the respective scenarios are very similar in their overall characteristics. The plot for the baseline approach in Figure 4.9 shows a different picture. The transport fleet cannot manage the amount of orders introduced into the system which leads to linear growth in stock levels at the considered storage facilities. This finding is also backed by plot of the global number of pending transport orders within the system where the steady growth of orders is replicated both for regular and reverse transports. Taking also into account the findings from the preceding section it can be said that the applied load on the forwarding agency was too much to be handled effectively when employing the baseline approach to transport planning.

With these results in mind, it is interesting to see in Figure 4.9, that the mixed transport fleet in which

	Mixed Transport Fleet	LEM Transport Fleet	Baseline Transport Fleet
LEM3 Transport Operations			
Σ Deliv ($\mu_s \pm \sigma_s$)	$2.478,90 \pm 17,06$	$4.301,30 \pm 5,88$	–
$CI_\alpha : \alpha = 0.05$	$[2.466,70\,;2.491,10]$	$[4.297,09\,;4.305,51]$	–
Σ Deliv$_{\text{full}}$ ($\mu_s \pm \sigma_s$)	$1.231,20 \pm 12,93$	$2.194,50 \pm 1,92$	–
$CI_\alpha : \alpha = 0.05$	$[1.221,94\,;1.240,45]$	$[2.193,13\,;2.195,87]$	–
Σ Deliv$_{\text{empty}}$ ($\mu_s \pm \sigma_s$)	$1.247,70 \pm 17,60$	$2.106,80 \pm 5,29$	–
$CI_\alpha : \alpha = 0.05$	$[1.235,11\,;1.260,29]$	$[2.103,02\,;2.110,58]$	–
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]	$1.074.057,25 \pm 8.437,06$	$1.974.968,34 \pm 12.396,29$	–
$CI_\alpha : \alpha = 0.05$	$[1.068.021,7\,;1.080.092,8]$	$[1.966.100,6\,;1.983.836,1]$	–
Baseline Transport Operations			
Σ Deliv ($\mu_s \pm \sigma_s$)	$1.740,70 \pm 12,80$	–	$3.824,70 \pm 80,48$
$CI_\alpha : \alpha = 0.05$	$[1.731,54\,;1.749,86]$	–	$[3.767,13\,;3.882,27]$
Σ Deliv$_{\text{full}}$ ($\mu_s \pm \sigma_s$)	$954,80 \pm 12,54$	–	$2.074,20 \pm 22,40$
$CI_\alpha : \alpha = 0.05$	$[945,83\,;963,77]$	–	$[2.058,17\,;2.090,23]$
Σ Deliv$_{\text{empty}}$ ($\mu_s \pm \sigma_s$)	$785,90 \pm 15,18$	–	$1.750,50 \pm 58,53$
$CI_\alpha : \alpha = 0.05$	$[775,04\,;796,76]$	–	$[1.708,63\,;1.792,37]$
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]	$868.476,87 \pm 6.993,96$	–	$1.759.706,48 \pm 30.745,97$
$CI_\alpha : \alpha = 0.05$	$[863.473,7\,;873.480,0]$	–	$[1.737.712,1\,;1.781.700,8]$

Table 4.9: Overview of transport operations for trucks managed by different transport agents.

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 tunes in on a certain level. However, the plot of the stock levels suggests significantly increased peak amplitudes compared to the planning case. In addition, although the total amount of pending transport orders levels of after the initial phase of the simulation it does so as already seen in the first set of experiments at higher levels, i. e. at ~ 20 regular transport orders (compared to ~ 15 in the planning case) and ~ 140 reverse transport orders (compare to ~ 90).

Concluding the analysis with regard to pending orders and inventory levels, the superiority of the planning approach to operative transport planning could be affirmed. While this result for itself bears little surprise, with this second set of experiments it could also be shown where the baseline approach used in the experiments has its limits and, in particular, how these limits may be extruded by mixing in planning transport management agents into a baseline transport fleet.

Transport Operations The data presented in Table 4.9 shows that the planning approach led 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 both more regular and reverse transports were 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 dominance in the raw amount of handled transports highlighted thus far and an slightly increased total length of delivery tours, the fraction of container transports on all truck operations is in the mean notably higher for the planning transport fleet ($65,5\,\%$) than for the baseline fleet ($57,4\,\%$). When comparing these results with their counterparts measured in the first series of experiments with a lower order inflow (cf. Table 4.5 on page 30), the data suggests that given a higher amount of orders in the system, the planning transport fleet can substantially improve its truck utilization. To be more precise, the former fraction of $47,2\,\%$ of transports on all truck operations – the rest being empty rides – is brought up to $65,7\,\%$. The data allows for the formation of the assumption, which needs to be affirmed or disproved by further experiments, that the planning fleet performs particularly well in scenarios with a high order inflow, potentially even a higher inflow as simulated in the experiments conducted for this report. However, the data acquired from the two considered series of experiments also shows that the goals of high truck 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.

When shifting the focus from the analysis of the homogeneous settings where either only baseline or planning agents were operating in the scenario towards the mixed setting with an equal amount of both agent types, the first thing to notice in the data presented in the left column of Table 4.9 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, the data 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 steals a significant amount of individually less profitable reverse transport from the baseline fleet. However, as documented in Table 4.6 on page 31, given the specification of the considered simulation model, 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 capacity utilization and profitability.

4.4.3 LEM3 in Transport Planning

For both sets of experiments which have been conducted for this report, the LEM3 library has been employed for the optimization of transport plans as outlined in Section 2.2.4 on page 10 and, with regard to the technical realization, in Section 3.1.1 on page 14. In particular, as stated in the description of the experimental setup, LEM3 was run with a standard parameterization throughout the experiments, i. e. the experiments were not designed as a means to identify a particularly optimized parameterization for LEM3 but rather to establish the basic feasibility of the approach for autonomous route planning as new application domain.

As this could be affirmed in the series of experiments conducted for the report, the next step with regard to the deeper integration of LEM3 is an in-depth analysis of the evolutionary searches which are performed in order to find optimized route plans for specific means of transport. Figure 4.10 on the following page shows plots of several showcase searches which have been sampled from a single simulation run for the high order inflow scenario and a planning transport fleet. These plots render the computed *values* for new solution hypotheses (denoted as *plan skeletons* in Section 2.2.4 of this report) which are generated by the LEM system during a single optimization process on the Y-axis, with the X-axis referring to the respective generations, starting at 0. The individuals drawn at an x-value of -1 thereby constitute the initial population created right at the beginning of the search process.

The plots seem to suggest that in all sampled cases, relatively good solutions are found immediately with the initial instantiation of the solution population. Throughout the following generations improvements occur, yet they do so very infrequently and seldom bring about significant gains as in the two examples in the upper right of Figure 4.10. Follow-up experiments must now clarify whether LEM3 actually efficiently manages to search the solution space or whether it gets stuck with local optima and thus consistently fails to find better solutions. Another open question is raised by the strong persistence in creating the same solution hypotheses across back-to-back generations. It needs to be investigated whether this is due to the application domain or rather an indicator that more variation must be introduced in the search process by recourse to, for instance, classical mutation operators. Alternatively, it can be an indicator, that the representation space needs to be extended, using for instance, structured attributes.

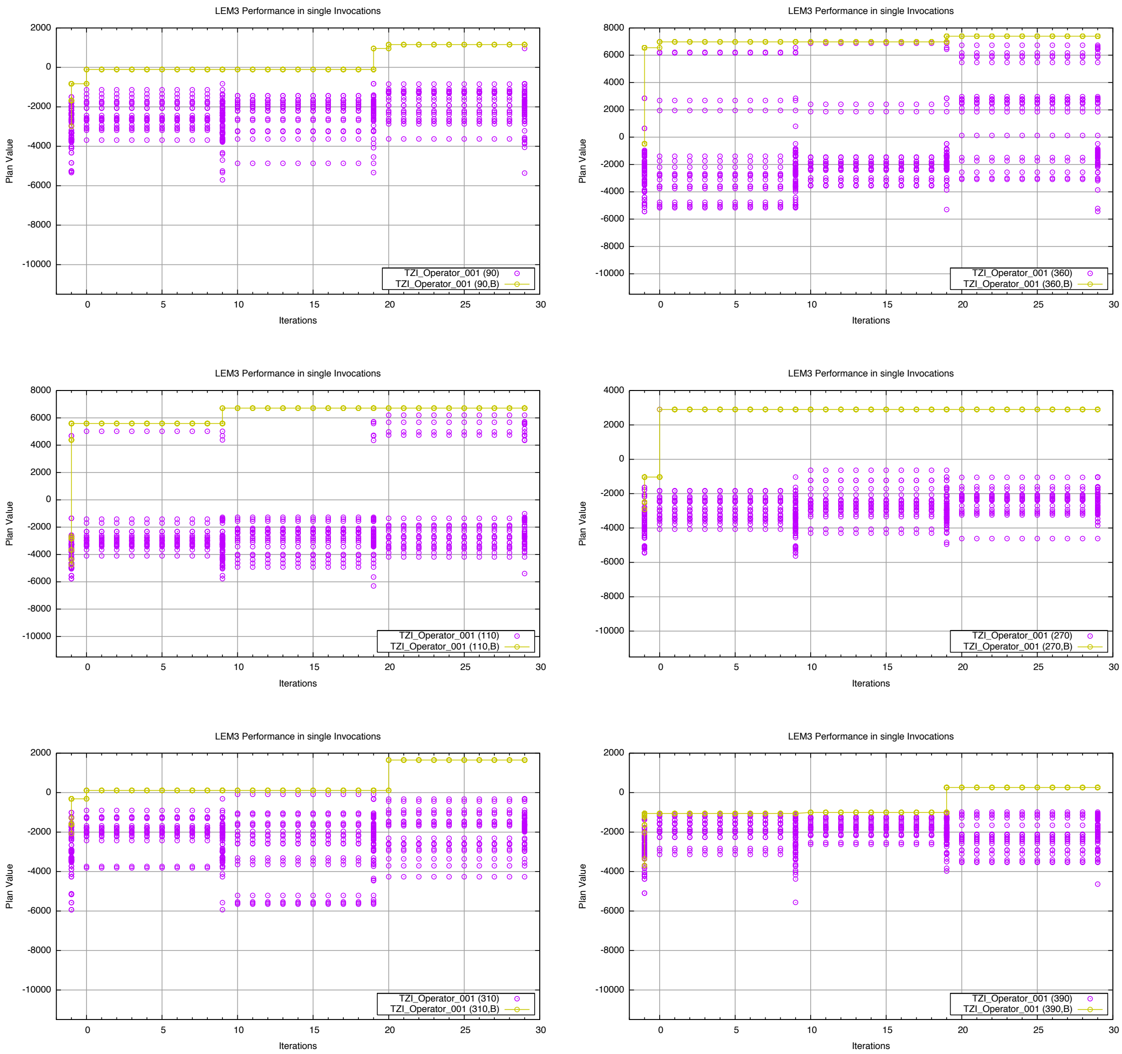

Figure 4.10: Paradigmatic examples for the LEM3 search for an optimized pickup and delivery plan.

5 Conclusion and Future Work

This report presented a methodology for order pickup and delivery planning by autonomous agents. The agents use the learnable evolution model to create sequences of the most optimal container deliveries. In the presented model, each container has an assigned value which is updated based on waiting time of containers. Implementation details of the method also have been presented.

Experimental results indicate that the method performs superior when compared to greedy 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. These experiments were performed using the PlaSMA multiagent system. The modeled scenario included 50 major cities in Germany, and a German motorway network with 750 edges.

5.1 Directions for Future Research and Development

Revision of LEM Planning and Provision of Additional Knowledge Besides an in-depth analysis of LEM3 performance as discussed in Section 4.4.3 on the previous page, future research in the integration and exploitation of the LEM3 system for the application domain of individual transport planning will consider a revision of calculation of the value of transport plans as introduced in Section 2.2.2 on page 8, namely a normalization of plans with respect to 1) (estimated) plan execution time, or 2) haulage distance.

In addition, further research should investigate possibilities to change the format of the solution which are generated by the LEM3 library such that they are more expressive than the *plan skeletons* used thus far. For instance, one plan step could be represented as a tuple $\langle \mathrm{Loc}_a, action, \mathrm{Loc}_b \rangle$. The concrete embodiment of this revision will be determined by the representational means of the LEM3 system.

Also, as pointed out in Section 2.2.3 on page 9, LEM3 excels in comparison to other systems for evolutionary optimization when additional domain knowledge can be provided which is suitable 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. In a comparative evaluation, it is then rendered possible to measure the influence of the availability of the aforementioned knowledge on the plan quality and thus on the logistics performance of individual transport management agents and the freight forwarding agency as a whole.

Architectural Changes to the Forwarder Implementation The multiagent-based realization of the freight forwarding agency as described in Section 3 on page 12 has proven to be a suitable basis to conduct series of experiments to compare approaches to autonomous compilation of transport routes as shown in Section 4 on page 20. However, the implementation will be adapted in the future in order to better encapsulate agent roles and provide for a more explicit modeling of the outer boundary of the forwarding agency.

To that end, the existing location management agent (cf. Section 3.4) will be partitioned such that both the storage facility management role and the - simulation-specific - load generation role are played by dedicated agents. The former agents may thereby operate either for the forwarding agency (management of company-owned storage facilities) or third parties. The latter agent(s) become simulation-specific infrastructure agents that do not belong to the freight forwarding agency.

The instantiation of order management agents will be detached from the container and respective order generation process by the dedicated order generation agent. This is due to the fact, that the aforementioned delegation of the supervision of order handling to the order management agents lies within the power of decision of the freight forwarding agency. Therefore, the model of the freight forwarding agency will be extended with what may be thought of as an explicit company boundary. To that end, the order information service (cf. Section 3.3) will be extended to provide an interface for the receipt of transport orders either from the load generation agent or explicitly modeled customer agents. Therefore, the agent so far referred to simply as order information service - that being its sole function - will play an important additional role whose importance is envisioned to grow if, for instance, real contract negotiations between customers and the forwarding agency are enabled. The OIS is thus turned into a forwarding agency representative similar to the TSA concept described by Bloos et al. in [BSK09].

Side by Side Comparison of LEM Planning with a Centralized Approach In order to more accurately appraise the performance of approaches, whether or not they may be based on LEM3, for agent- rather than the still more common fleet-level planning of transport routes, the forwarder implementation will also be extended in such a way that it is rendered possible to configure both centralized or decentralized planning and test both approaches in the same scenarios. It is expected that the execution of such experiments would allow to pinpoint the performance of LEM3-based planning between the extremes of the presented baseline order selection on a per-agent basis and traditional centralized planning.

6 Acknowledgements

The authors would like to thank Christoph Greulich and Malte Humann for valuable discussions that helped to shape the implementation of the freight forwarding MAS and their efforts in implementing the system in the PlaSMA platform. This research has been funded by the German Research Foundation(DFG) within the Collaborative Research Centre (CRC) 637 *Autonomous Cooperating Logistic Processes – A Paradigm Shift and its Limitations*[8] at the Universität Bremen, Germany. Development of the LEM3 system has been supported in part by the National Science Foundation Grants. Current research on the learnable evolution model is supported by the National Institute of Standards and Technology grant.

[8] CRC 637 web site: http://sfb637.uni-bremen.de

List of Figures

List of Tables

References

[BSK09] M. Bloos, J. Schönberger, and H. Kopfer. Supporting cooperative demand fulfillment in supply networks using autonomous control and multi-agent-systems. In *INFORMATIK 2009. Beiträge der 39. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, 2009.

[DYKM04] P.A. Domanski, D. Yashar, K. Kaufman, and R.S. Michalski. An Optimized Design for Fine-Tube Evaporators Using the Learnable Evolution Model. *International Journal of Heating, Ventilation, Air-Conditioning and Refridgerating Research*, 10:201–211, 2004.

[fIPAF02a] Foundation for Intelligen Physical Agents (FIPA). FIPA Propose Interaction Protocol Specification, December 2002.

[fIPAF02b] Foundation for Intelligen Physical Agents (FIPA). FIPA Query Interaction Protocol Specification, December 2002.

[fIPAF02c] Foundation for Intelligen Physical Agents (FIPA). FIPA Request Interaction Protocol Specification, December 2002.

[GOB07] J. D. Gehrke and C. Ober-Blöbaum. Multiagent-based Logistics Simulation with PlaSMA. In *Informatik 2007. Informatik trifft Logistik. Beiträge der 37. Jahrestagung der Gesellschaft für Informatik*, number 109 in GI Proceedings, pages 416–419, Bremen, Germany, September 24–27 2007. Gesellschaft für Informatik (GI).

[GWB+10] J. D. Gehrke, T. Warden, J.-O. Berndt, et al. *PlaSMA Multiagent Simulation – User Guide*, 2010.

[HW07] M. Hülsmann and K. Windt, editors. *Understanding Autonomous Cooperation & Control in Logistics: The Impact on Management, Information and Communication and Material Flow*. Springer, Berlin, Germany, 2007.

[KM00] K. Kaufman and R.S. Michalski. Applying the LEM Methodology to Heat Exchanger Design. Reports of the Machine Learning and Inference Laboratory MLI 00–2, George Mason University, Fairfax, VA, 2000.

[Lia99] Sheng Liang. *The Java Native Interface: Programmer's Guide and Specification*. Prentice Hall International, Inc., 1999.

[Mic98] R.S. Michalski. Learnable Evolution: Combining Symbolic and Evolutionary Learning. In *Proceedings of the Fourth International Workshop on Multistrategy Learning (MSL'98)*, pages 14–20, Desenzano del Garda, Italy, June 11–13 1998.

[Mic00] R.S. Michalski. Learnable Evolution Model: Evolutionary Processes Guided by Machine Learning. *Machine Learning*, 38:9–40, 2000.

[MK06] R.S. Michalski and K. Kaufman. INTELLIGENT EVOLUTIONARY DESIGN: A New Approach to Optimizing Complex Engineering Systems and its Application to Designing Heat Exchangers. *Internatinal Journal of Intelligent Systems*, 21(12), 2006.

[MWK07] R.S. Michalski, J. Wojtusiak, and K. Kaufman. Progress Report on the Learnable Evolution Model. Reports of the Machine Learning and Inference Laboratory MLI 07–2, George Mason University, Fairfax, VA, 2007.

[WM05] J. Wojtusiak and R.S. Michalski. 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.

[WM06] J. Wojtusiak and R.S. Michalski. 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.

[WMKP06] J. Wojtusiak, R.S. Michalski, K.A. Kaufman, and J. Pietrzykowski. 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*, pages 523–526, Los Alamitos, CA, November 2006. IEEE Computer Society.

[Woj04] J. Wojtusiak. 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.

[Woj07] J. Wojtusiak. *Handling Constrained Optimization Problems and Using Constructive Induction to Improve Representation Spaces in Learnable Evolution Model*. PhD thesis, College of Science, George Mason University, Fairfax, VA, November 2007.

[Woj08] J. Wojtusiak. 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.

[Woj09] Janusz Wojtusiak. The LEM3 System for Multitype Evolutionary Optimization. *Computing and Informatics (28)*, pages 225–236, 2009.

[WPG+10] T. Warden, R. Porzel, J. D. Gehrke, O. Herzog, H. Langer, and R. Malaka. Towards ontology-based multiagent simulations: The plasma approach. In Andrzej Bargiela, Sayed Azam Ali, David Crowley, and Eugene J. H. Kerckhoffs, editors, *24th European Conference on Modelling and Simulation (ECMS 2010)*, pages 50–56. European Council for Modelling and Simulation, 2010.

[YDWK10] D. Yashar, P.A. Domanski, J. Wojtusiak, and K. Kaufman. Evolutionary Computation Approach to Heat Exchanger Optimization. In *Proceedings of the American Society or Heating, Refrigerating and Air-Conditioning Engineers Annual Conference*, Albuquerque, NM, June 26–30 2010.